جمهورية العراق
وزارة التعليم العالي والبحث العلمي
الجامعة التكنولوجية
قسم علوم الحاسبات

الربط بين قواعد البيانات بأستخدام الأكسس والفيجوال بيسك

أعداد
عمار فخري مهدي الأنباري

أشراف
د. عماد كاظم جبار

2008

فهرس المواضيع

الفصل الأول تعلم البرمجة بلغة الفيجوال بيسك	
الموضوع	رقم الصفحة
1 – 1 المقدمة	1
1 – 2 ماهو الفيجوال بيسك	1
1 – 3 الفرق بين لغة فيجوال بيسك ولغة بيسك	2
1 – 4 لماذا نستخدم لغة فيجوال بيسك	2
1 – 5 تشغيل برنامج لغة فيجوال بيسك	2
1 – 6 بيئة لغة فيجوال بيسك	4
1 – 7 أنواع أدوات التحكم	11
1 – 8 اضافة أدوات تحكم جديدة	14
1 – 9 اضافة وازالة عناصر التحكم الى ومن النموذج	17
1 – 10 تسمية آدوات التحكم	18
1 – 11 حُفظ المشروع	19
1 – 12 الخصائص والأحداث	20
1 – 13 التعامل مع القوائم	22
1 – 14 البرمجـة بلغة فيجوال بيسك	24
1 – 15 تحليل الأجراء الحدثي	25

1 – 16 العناصر الأساسية لجمل لغة فيجوال بيسك	27
1 – 16 – 1 البيانات	27
1 – 16 – 2 التعابير	29
1 – 16 – 3 المشغلات	29
1 – 16 – 4 الدوال	30
1 – 17 أنواع الجُمل في لُغة فيجوال بيسك	31
1 – 17 – 1 جُمل التعليق	31
1 – 17 – 2 جُمل التصريح	32
1 – 17 – 3 جُمل التخصيص	32
1 – 17 – 4 الجُمل الشرطية	32
1 – 17 – 5 جُمل التكرار أوالدوران	38
1 – 18 العمليات التي تُطبق على النماذج	41
1 – 19 صندوق الرسائل وصندوق المُدخلات	42
1 – 19 – 1 دالة صندوق الرسائل	43
1 – 19 – 2 دالة صندوق المدخلات	49
1 – 20 تكوين تطبيقات لُغة فيجوال بيسك	52
1 – 21 دراسة بعض الخصائص المُهمة	57

الفصل الثاني

تعلم قواعد البيانات في مايكروسوفت آكسس

2 – 1 المقدمة	61
2 – 2 ماهي قاعدة البيانات	61
2 – 3 مقدمة لبرنامج مايكروسوفت أكسس	62
2 – 4 مُصطلحات خاصة بقواعد البيانات	62
2 – 5 تشغيل برنامج مايكروسوفت أكسس	63
2 – 6 مُكونات مايكروسوفت آكسس	64
2 – 7 فتح قاعدة بيانات موجودة	66
2 – 8 الجداول	66
2 – 9 انشاء جدول جديد	67
2 – 10 تكوين جدول بطريقة عرض التصميم	67
2 – 10 – 1 ادراج الحقول وتسميتُها	68
2 – 10 – 2 تحديد أنواع البيانات في حقول الجدول	68
2 – 10 – 3 تحديد خصائص للحقول المُدرجة مع بياناتها	72
2 – 10 – 4 حفظ الجدول مع انشاء مفتاح أساسي اذا تطلب الأمر	76
2 – 11 مُعالجة البيانات	77
2 – 12 عرض البيانات في الجدول	77
2 – 13 الأنتقال بين نافذة عرض التصميم ونافذة عرض صفحة البيانات	78
2 – 14 تغيير أتجاه نافذة عرض صفحة البيانات	78
2 – 15 ترتيب السجلات في الجدول	79

2 – 16 تصفية البيانات	79
2 – 17 العلاقات بين الجداول في قاعدة البيانات	81
2 – 17 – 1 تعريف العلاقات	82
2 – 17 – 2 أنواع العلاقات	82
2 – 18 البحث والأستبدال	85
2 – 19 التنقُل عبر قيود (سجلات) الجدول	85
الفصل الثالث **تطبيق عملي للربط بين قواعد البيانات** **(أكسس & فيجوال بيسك)**	
3 – 1 المقدمة	87
3 – 2 انشاء برامج قواعد البيانات	87
3 – 3 عُنصر التحكم المُرتبط بالأدخال	87
3 – 4 أزرار التنقل عبر السجلات	89
3 – 5 تحديد موقع السجل الحالي	90
3 – 6 اضافة سجل جديد	91
3 – 7 حفظ السجل الحالي	91
3 – 8 الغاء تحديث السجل الحالي	92
3 – 9 حذف السجل الحالي	92
3 – 10 البحث عن بيانات الحقول	93
3 – 11 تحديد الكتابة الرقمية	95
3 – 12 كائن المُعطيات في قواعد البيانات	96

3 – 12 – 1 كائن المعطيات من نوع Dynaset	97
3 – 12 – 2 كائن المعطيات من نوع Table	100
3 – 13 كائن البيانات ADO Data Control	103
3 – 14 التقارير في قواعد البيانات	113
3 – 15 قاعدة البيانات بأستخدام المعالج	119
3 – 16 لغة الأستفسارات المهيكلة SQL	123
3 – 16 – 1 أنماط لغة الأستفسارات المهيكلة (البنيوية)	123
3 – 16 – 2 برمجة أنماط لغة الأستفسارات المُهيكلة	124

فهرس الأشكال

الفصل الأول تعلم البرمجة بلغة الفيجوال بيسك	
الشكل	**رقم الصفحة**
(1 – 1) قائمة البدأ بالفيجوال بيسك في نظام التشغيل	3
(1 – 2) نافذة مشروع جديد	3
(1 – 3) النافذة الرئيسية لبرنامج لغة فيجوال بيسك	4
(1 – 4) أيقونات نافذة مستكشف المشروع	6
(1 – 5) خصائص النموذج	7
(1 – 6) نافذة موقع عرض النموذج	8
(1 – 7) نافذة تصميم النموذج	9
(1 – 8) نافذة البرمجة في الفيجوال بيسك	10
(1 – 9) شريط أدوات التحكم	10
(1 – 10) اضافة أدوات تحكم جديدة	15
(1 – 11) قائمة صندوق أدوات جديدة	16
(1 – 12) حذف قائمة صندوق الأدوات	16
(1 – 13) ازالة عناصر التحكم من النموذج	17
(1 – 14) محرر القوائم	22
(1 – 15) برمجة الأوامر داخل محرر القوائم	23

(1 – 16) نافذة كتابة الشفرة	24
(1 – 17) تنفيذ برنامج لغة فيجوال بيسك	25
(1 – 18) برمجة دالة صندوق الرسائل	46
(1 – 19) برمجة أزرار تحكم دالة صندوق الرسائل	47
(1 – 20) برمجة أزرار تحكم دالة صندوق الرسائل مع أضافة صورة	48
(1 – 21) برمجة دالة صندوق المدخلات	50
(1 – 22) برمجة أزرار دالة صندوق المدخلات	51
(1 – 23) عمل حاسبة بسيطة	55
(1 – 24) حساب مضروب أي عدد	56
(1 – 25) اختبارعدداً ما أولي آم غير أولي	57
(1 – 26) استدعاء القيم أو العناصر ضمن قائمة الكائن Combo برمجياً	58
(1 – 27) برمجة الكائن Option Button	59

الفصل الثاني

تعلم قواعد البيانات في مايكروسوفت آكسس

(2 – 1) فتح أو تكوين قاعدة بيانات جديدة	63
(2 – 2) خزن قاعدة بيانات جديدة	64
(2 – 3) نافذة قاعدة البيانات الرئيسية	65
(2 – 4) حقول وسجلات الجدول	66
(2 – 5) تكوين جدول بطريقة عرض التصميم	67
(2 – 6) نوع بيانات حقول الجدول	69
(2 – 7) مُحرر أو مُنشأ التعابير	76

(2 – 8) الأنتقال بين نافذة عرض التصميم ونافذة عرض صفحة البيانات	78
(2 – 9) أنواع تصفية البيانات	79
(2 – 10) تصفية بيانات الجدول بأستخدام شرط Filter For	80
(2 – 11) تصفية حسب النموذج لبيانات الجدول	80
(2 – 12) العلاقة بين جدول الطلاب وجدول الدرجات من خلال حقل الربط	81
(2 – 13) تعريف العلاقات بين الجداول	82
(2 – 14) علاقة واحد الى مجموعة بين جدولين في قاعدة البيانات	83
(2 – 15) علاقة واحد الى واحد بين جدولين في قاعدة البيانات	83
(2 – 16) تصميم العلاقات بين الجداول	84
(2 – 17) نافذة البحث والأستبدال	85
(2 – 18) التنقل عبر سجلات الجدول	86

الفصل الثالث

تطبيق عملي للربط بين قواعد البيانات

(أكسس & فيجوال بيسك)

(3 – 1) تنفيذ برنامج ملف قاعدة بيانات الطلاب	95
(3 – 2) تحميل مكتبة البرمجة DAO Object Library	96
(3 – 3) برمجة ملف قاعدة البيانات من خلال الكائن مجموعة السجلات Dynaset	98
(3 – 4) تصميم قاعدة البيانات من داخل الفيجوال بيسك	101
(3 – 5) برمجة ملف قاعدة البيانات من خلال الكائن مجموعة السجلات Table	102
(3 – 6) اضافة كائن الوصول للبيانات ADO	104
(3 – 7) تصميم برنامج قاعدة البيانات بأستخدام الكائن ADO	104

(3 – 8) خصائص كائن البيانات ADO	105
(3 – 9) صفحة خصائص كائن البيانات ADO	106
(3 – 10) ربط برنامج الفيجوال بيسك بقاعدة بيانات أكسس بأستخدام كائن بيانات ADO	106
(3 – 11) تحديد نوع مصدر البيانات	107
(3 – 12) تحديد قاعدة البيانات من نوع مايكروسوفت أكسس	107
(3 – 13) تحديد مكان قاعدة بيانات أكسس وأسم مصدر بياناتها	108
(3 – 14) أختيار أسم مصدر بيانات القاعدة	109
(3 – 15) تحديد جدول قاعدة البيانات	109
(3 – 16) نافذة خصائص ربط البيانات	111
(3 – 17) ربط كائن البيانات ADO مع القاعدة وأختبار الربط	111
(3 – 18) ربط كائن البيانات ADO بجدول القاعدة من خلال مصدر البيانات	112
(3 – 19) أجراء العمليات المختلفة على بيانات القاعدة من خلال كائن Data Grid	112
(3 – 20) ربط كائن البيانات ADO بالقاعدة من خلال كائن Data Grid	113
(3 – 21) اضافة بيئة البيانات Data Environment الى المشروع	113
(3 – 22) اضافة تقرير لبيانات القاعدة الى المشروع	114
(3 – 23) تحديد نوع خاصية ربط التقرير	115
(3 – 24) تحديد أسم قاعدة البيانات المصممة مسبقاً	115
(3 – 25) عرض التقرير من خلال كائن زر الأمر	116
(3 – 26) ربط كائن زر الأمر مع حقول الجدول	116
(3 – 27) تصميم واجهة عرض التقرير	117
(3 – 28) تنفيذ برنامج قاعدة البيانات مع عرض تلخيص لورقة العمل	118
(3 – 29) طباعة ملخص ورقة العمل الخاصة ببيانات القاعدة	118

(3 – 30) تكوين قاعدة البيانات بأستخدام المعالج	119
(3 – 31) معالج نموذج البيانات	119
(3 – 32) تحديد نوع الربط مع قاعدة البيانات	120
(3 – 33) تحديد أسم قاعدة بيانات أكسس ومسارها	120
(3 – 34) تحديد أسم نموذج قاعدة البيانات بأستخدام المعالج	121
(3 – 35) أختيار أسم جدول قاعدة البيانات والحقول المطلوبة	121
(3 – 36) تحميل أزرار التحكم الخاصة بقاعدة البيانات بأستخدام المعالج	122
(3 – 37) تنفيذ برنامج قاعدة البيانات بأستخدام المعالج	122
(3 – 38) تحديد النموذج الأول أثناء تنفيذ المشروع	123
(3 – 39) جدول قاعدة بيانات أكسس	124
(3 – 40) قاعدة البيانات بأستخدام عبارات الـ SQL	125
(3 – 41) عرض بيانات حقل الأسم فقط بأستخدام عبارة Select Name	127
(3 – 42) عرض بيانات حقلي الرقم والأسم فقط مع بعض السجلات	128
(3 – 43) عرض جميع حقول القاعدة مع بعض سجلاتها	129
(3 – 44) ترتيب بيانات القاعدة تنازلياً بأستخدام عبارة SQL الشرطية	130
(3 – 45) عرض سجل واحد مع جميع حقول القاعدة	131
(3 – 46) عرض جميع حقول القاعدة مع سجلين فقط	132
(3 – 47) عرض جميع حقول القاعدة مع سجلين فقط أعتماداً على حقل الأسم	132

فهرس الجداول

الفصل الأول

تعلم البرمجة بلغة الفيجوال بيسك

الجدول	رقم الصفحة
(1 – 1) أنواع أدوات التحكم ووظيفتها	11
(1 – 2) البادئات الأكثر شيوعاً لأدوات التحكم	19
(1 – 3) الأحداث المشتركة للعديد من أدوات التحكم	20
(1 – 4) الألوان الستة عشر والأرقام المُخصصة لها	21
(1 – 5) أنواع البيانات الرئيسية في برنامج لغة فيجوال بيسك مع وصفها	28
(1 – 6) الدوال الشائعة الأستخدام مع برامج لغة فيجوال بيسك	30
(1 – 7) العمليات الرئيسية التي تُطبق على النماذج	41
(1 – 8) أزرار التحكم المُستخدمة مع صندوق الرسائل وأرقامها	44
(1 – 9) أزرار التحكم المُستخدمة مع صندوق الرسائل والصور الخاصة بها	45
(1 – 10) أزرار الأمر الخاصة بدالة صندوق الرسائل مع أسم الأمر	47
(1 – 11) عناصر التحكم والخصائص المستخدمة فيها	52
(1 – 12) القيم العددية المُتاحة للكائن صندوق الفحص والأختبار	59
(1 – 13) القيم العددية المُتاحة لكائن رسم الشكل الهندسي	60

الفصل الثاني تعلم قواعد البيانات في مايكروسوفت آكسس	
(2 – 1) أنواع بيانات الحقول وخصائصها وحجم تخزينها	69
(2 – 2) خاصية حجم الحقل المُحددة للبيانات في الحقول الرقمية	72
(2 – 3) الحروف والعلامات الخاصة بقناع الأدخال	73
(2 – 4) أزرار التنقل عبر سجلات الجدول مع وظائفها	86
الفصل الثالث **تطبيق عملي للربط بين قواعد البيانات** **(أكسس & فيجوال بيسك)**	
(3 – 1) العمليات المرتبطة بكائن عنصر الوصول للبيانات	89
(3 – 2) تغيير خصائص الكائنات	105
(3 – 3) تغيير خصائص صناديق النص	110

المصـــادر

1 – 1 المقدمة : Introduction

تُعتبر الفيجوال بيسك احدى لغات البرمجة المتطورة والتي تُستخدم في خلق (تكوين) التطبيقات التي تستثمر النوافذ الرسومية لواجهة المستخدم . هذه التطبيقات يتم تكوينها بأستخدام مجموعة من الكائنات (Objects) مثل كائن صندوق النص (Text Box) وكائن مفتاح أو زر الأمر (Command Buttons) ، حيث تتم اضافة (رسم) هذه الكائنات الى نافذة الفيجوال بيسك والتي تسمى بالنموذج (Form) ومن ثم كتابة البرنامج أو الشفرة (Code) داخل الكائن أو في بعض الأحيان داخل النموذج .

مما تقدم أعلاه نستنتج أن الفيجوال بيسك عبارة عن تطبيقات وبرمجة التطبيقات (Application & Programming) .

1 – 2 ماهو الفيجوال بيسك: What is VB ?

هي اللغة المساعدة في تطوير واجهات البرامج ، حيث كانت اللغات البرمجية بكافة أنواعها ضعيفة في انشاء واجهات مقبولة للمستخدم من ناحية الرسوم والمخططات والصور (Graphic User Interface-GUI) ، اذ أن انشاء واجهات (Interface) تلائم نوعاً ما رغبة المستخدم يتطلب من المبرمج كتابة الاف من الأسطر لغرض تصميمها ، مما دفع الشركات ومن بينها شركة مايكروسوفت بأن تطور اللغات البرمجية وخصوصاً بعد اصدار أنظمة تشغيل تدعم بيئة المستخدم والتي تسمى بنظام النوافذ (Windows)

من المهم معرفة أن أصل لغة فيجوال بيسك هي لغة بيسك (Basic) المشتقة من الحروف الأولى للعبارة التالية : (Beginners All-purpose Symbolic Instruction Code) والمقصود بها لغة التعليمات الرمزية لكافة الأغراض للمبتدئين .

تم انشاء لغة فيجوال بيسك من قبل شركة مايكروسوفت ، حيث أن الجزء الأول منها " Visual " يشير الى الطريقة المستخدمة في خلق أو تكوين التطبيق ، أما الجزء الثاني منها " Basic " فيشير الى لغة البرمجة المستخدمة في التطبيق .

أول ظهور للغة فيجوال بيسك كان عام 1991 حيث تم دمج قدرات لغة بيسك مع أدوات التصميم المرئي ثم توفير سهولة الأستخدام وبساطتها . ثم كان الأصدار الثاني منها عام 1992 ، فالأصدار الثالث عام 1993 ، ومن ذلك الوقت تعاقبت الأصدارات ومنها الأصدار السادس (Visual Basic 6.0) والذي سنتناول الحديث عنه .

1 – 3 الفرق بين لغة فيجوال بيسك ولغة بيسك: VB language Vies Basic language

يتم في لغة بيسك تنفيذ البرنامج بشكل متسلسل (Sequentially) أي دفعة واحدة ، بينما في لغة فيجوال بيسك يتم التنفيذ جزء بعد أخر وحسب الأختيار حيث يقوم المبرمج بأختيار احدى الكائنات ورسمها على نافذة الفيجوال بيسك ومن ثم برمجة الكائن ، حيث يمكن للمبرمج اختيار أكثر من كائن وكل كائن تتم فيه كتابة البرنامج بشكل مستقل عن الكائن الأخر وينفذ كل كائن مستقلاً عن الكائن الأخر ، كما يمكن ربط كائنين أو أكثر ببرنامج واحد ولذلك تعتبر الفيجوال بيسك من اللغات المرنة وسهلة الأستخدام .

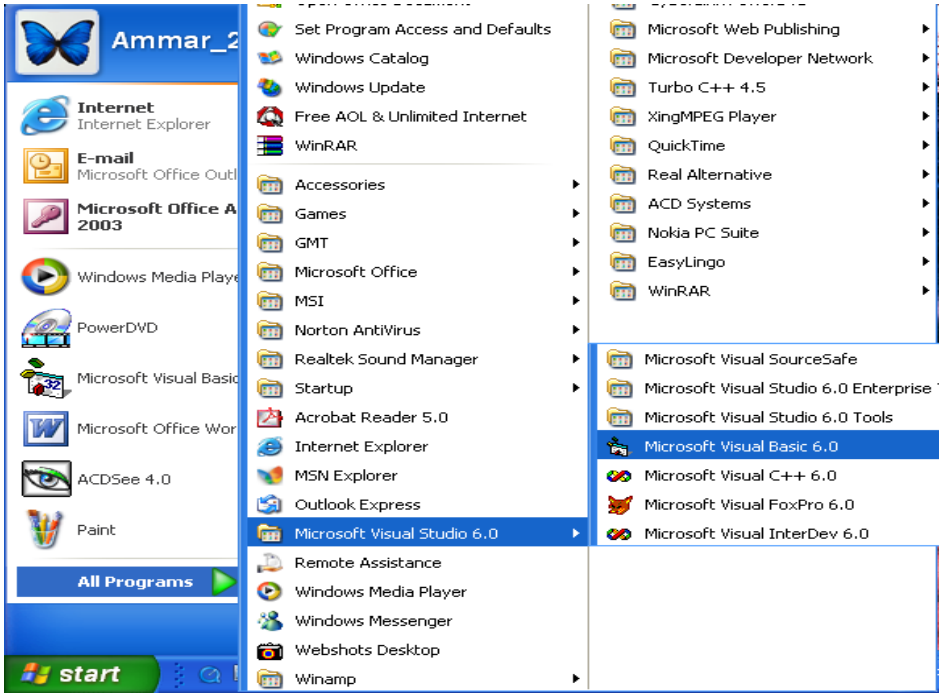
1 – 4 لماذا نستخدم لغة فيجوال بيسك : ? Why we use VB

تعتبر لغة فيجوال بيسك من اللغات الأكثر قبولاً بين شرائح المبرمجين وعلى كافة مستوياتهم ، ويتجلى ذلك في النقاط التالية :

1. انشاء تطبيقات كبيرة باستخدام برامج صغيرة .
2. سهولة استخدام البرامج وسرعة تنفيذها .
3. جمالية الواجهات مع امكانية ادخال المؤثرات الصوتية والأفلام (Multi Media) بحيث تُتيح للمستخدم التمتع والتفاعل مع البرنامج .
4. امكانية ربط لغة فيجوال بيسك مع تطبيقات وبرامج أُخرى .
5. امكانية أستدعاء الكثير من الوظائف الموجودة في نظام التشغيل وأمكانية الأستفادة من بعض البرامج التي تعمل في بيئة نظام النوافذ وتسخيرها لصالح البرنامج المُنشأ بلغة فيجوال بيسك .

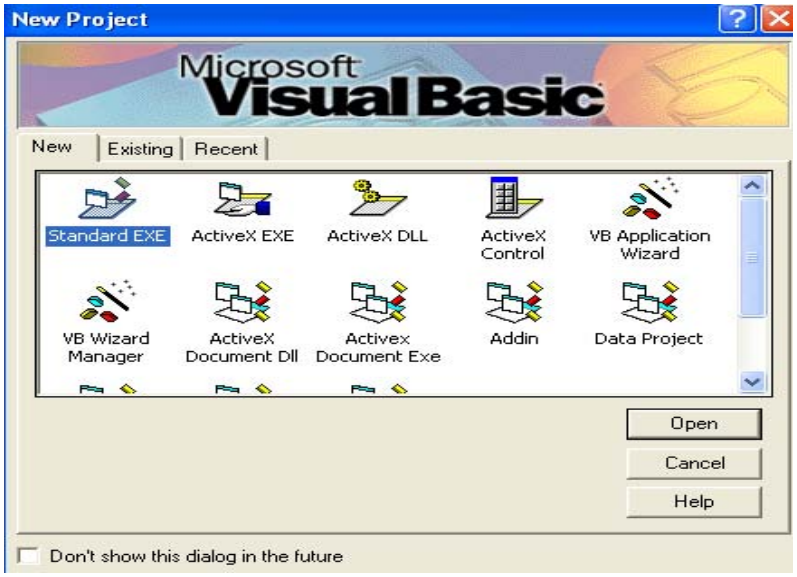
1 – 5 تشغيل برنامج لغة فيجوال بيسك :Starting Visual Basic

يتم تشغيل برنامج لغة فيجوال بيسك من خلال اختيار الأمر التالي من قائمة البدأ Start الموجودة في نظام التشغيل وكما موضح ذلك في الشكل رقم (1- 1) .



شكل رقم (1 – 1) : قائمة البدأ بالـفيجوال بيسك في نظام التشغيل

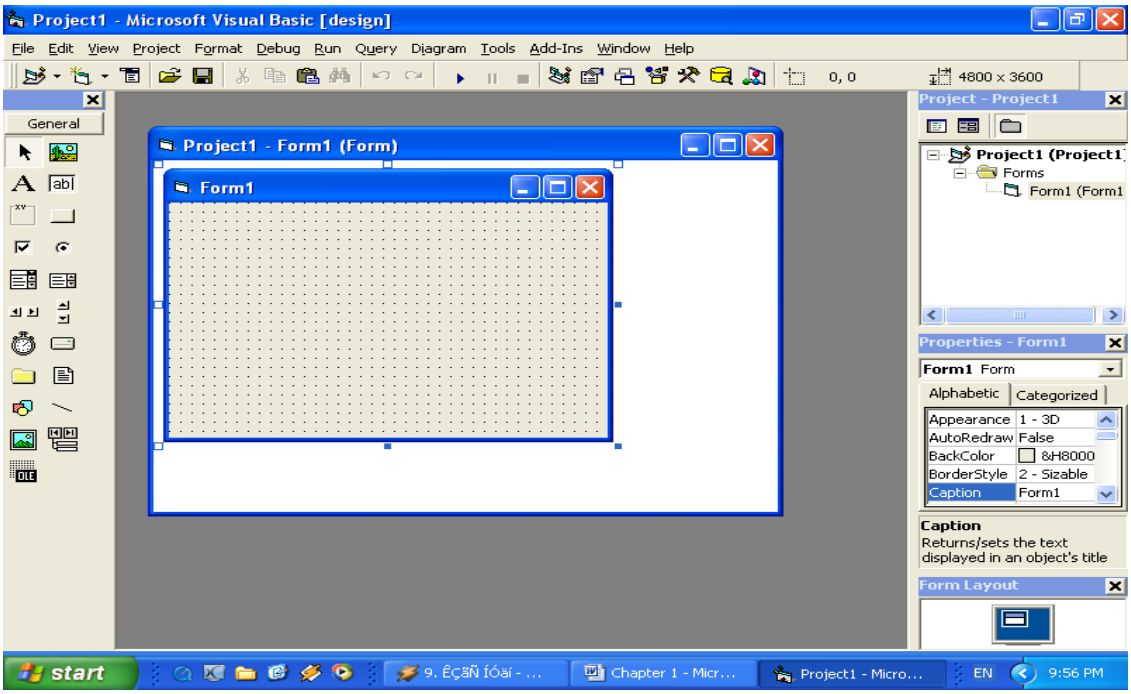
بعد أختيـار الأمـر أعـلاه (Microsoft Visual Basic 6.0) سـتظهـر نافـذة مشـروع جديـد(New Project) والتي تحتوي على ثلاثة خيارات وكما موضح ذلك بالشكل رقم (1 – 2).



شكل رقم (1 – 2) : نافذة مشروع جديد

1. New : وتعني مشروع جديد لم يسبق العمل به (New Project) .
2. Existing : مشروع موجود (تم أنشاءه مُسبقاً) وتم تخزينه سواءاً على جهاز الحاسبة أو على وسيلة أخرى مثل القرص المرن (Existing Project) .
3. Recent : تُظهر مجموعة من المشاريع التي تم العمل بها مُؤخراً وموقع تخزينها (Recent Project) .

تمكن الصفحة (New) من اختيار نوع المشروع الذي تُريد انشاءه ، حيث يوفر الفيجوال بيسك عدة قوالب جاهزة لأنماط مُختلفة من المشاريع يتم اختيار مشروع من النوع القياسي (الأفتراضي) Standard EXE وهو الأختيار الأفتراضي لأغلب المشاريع التي يتم تصميمُها بعدها نختار الأمر أفتح (Open) . عندها ستظهر لنا النافذة الرئيسية لبرنامج لغة (VB) والتي تُسمى ببيئة التطوير المُتكاملة (Integrated Development Environment - IDE) والموضحة بالشكل رقم (1 – 3) .



شكل رقم (1 – 3) : النافذة الرئيسية لبرنامج لغة فيجوال بيسك

1 – 6 بيئة لغة فيجوال بيسك : The visual basic environment

تتكون بيئة الفيجوال بيسك من مجموعة من العناصر هي :

1. شريط العنوان (Title Bar) : وهو الشريط الأول الذي يظهر في شكل رقم (1 – 3) ، حيث يظهر داخل هذا الشريط عنوان المشروع .
2. شريط القوائم (Menus Bar) : وهو الشريط الثاني الذي يظهر في شكل رقم (1 – 3) ، حيث يحتوي على مجموعة من القوائم وكل قائمة تحتوي على مجموعة من الوظائف (Functions) مثل ..,File , Edit , View والتي تساعدنا في بناء التطبيق .
3. شريط الأدوات (Tools Bar) : وهو الشريط الثالث الذي يظهر في شكل رقم (1 – 3) ، حيث يحتوي على مجموعة من الوظائف الموجودة داخل شريط القوائم مثل فتح ملف ، حفظ الملف الحالي ، ... الخ .
4. اضافة الى مجموعة من النوافذ الرئيسية الداخلة في بيئة الفيجوال بيسك والتي تُشكل بمجموعها الواجهة البينية (Interface) لبيئة الفيجوال بيسك ، وهذه النوافذ هي :

أ- نافذة محتويات المشروع (Project Window) : وتظهر على جهة اليمين من شكل رقم (1 – 3) محتويات التطبيق من نماذج (Forms) ووحدات نمطية (Modules) وغيرها من العناصر الموجودة في المشروع ، وتسهل عملية الأنتقال بين هذه المحتويات لغرض تعديلها أو تنفيذها أو اضافة نماذج ووحدات نمطية جديدة وتشبه المُستكشف في أسلوب التعامل مع محتوياتها لذلك يطلق عليها أيظاً بأسم مستكشف المشروع (Project Explorer) حيث يتم استعراض النماذج على شكل مجلدات (Folders) وكذلك بالنسبة للوحدات النمطية وغيرها من المحتويات الأخرى الموجودة داخل هذه النافذة .

عند النقر R.Click على نافذة محتويات المشروع ستظهر قائمة تسمح لنا القيام بعدة مهام هي :

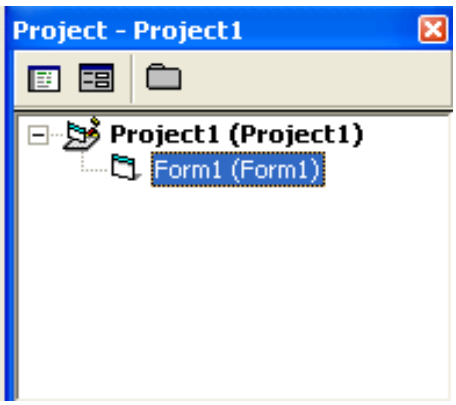
- اظهار شاشة تصميم واجهات البرنامج من خلال شاشة النموذج (View Object) ، وكذلك اظهار شاشة كتابة البرنامج من خلال شاشة الشفرة (View Code) .
- اظهار الخصائص لكل ملف (Properties) أي خصائص محتويات المشروع .
- اضافة نموذج أو وحدة نمطية للمشروع .

ملاحظة : يتم اضافة نماذج جديدة وغيرها من المحتويات الأخرى الى المشروع من خلال عمل الأجراء التالي :

R.Click (on project window) ⟶ Add ⟶ Form
⟶ Module

- حفظ الملف الحالي Save Form1.frm .
- طباعة الملف Print .
- اظهار نافذة مستكشف المشروع بوصفه جزءاً من بيئة التطوير المتكاملة Dockable .
- اخفاء نافذة مستكشف المشروع Hide .

يوجد في نافذة مستكشف المشروع شريط أدوات صغير يحتوي على ثلاثة أيقونات وكما موضح ذلك بالشكل رقم (1 – 4) .



شكل رقم (1 – 4) : أيقونات نافذة مستكشف المشروع

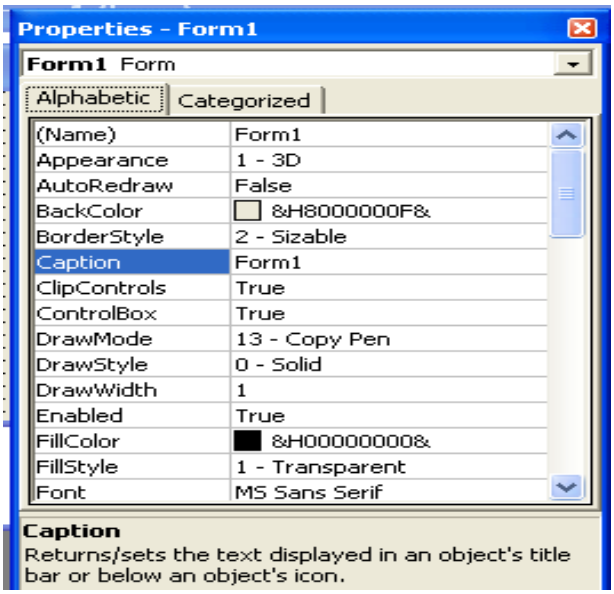
وهذه الأيقونات ابتداءاً من اليسار تمثل :

1. View Code : تُظهر نافذة البرمجة (الشفرة) للكائن أو الملف المحدد .
2. View Object : تُظهر النموذج المحدد .
3. Toggle Code : تُظهر أو تُخفي المجلدات .

ب- نافذة الخصائص (Properties Window) : تحتوي على خصائص مستقلة لكل نموذج ولكل كائن من كائنات صندوق الأدوات ، ويمكن تغيير هذه الخصائص بسهولة ونلاحظ تأثيرها مباشرةً على النموذج أو الكائنات التي يحتويها النموذج . تختلف هذه الخصائص حسب نوع الكائن ، الا أن الكثير منها يتكرر لجميع أنواع عناصر التحكم (الكائنات) . تحتوي نافذة الخصائص في أعلاها على صندوق قائمة يحتوي على قائمة بأسماء عناصر التحكم الموجودة على النموذج ، اذ يُمكن أختيار أي عنصر من عناصر القائمة لأستعراض خصائصه مُباشرةً . أما في أسفل نافذة الخصائص فيوجد اطار يُدعى الوصف (Description) يحوي معلومات توضيحية عن وظيفة

الخاصية المُحددة ، ويُمكن اظهار أو اخفاء هذا الأطار بالنقر بالزر الأيمن للفأرة على أطار نافذة الخصائص وتحديد أو الغاء تحديد القائمة (الوصف) .

يمكن استعراض الخصائص اما وفق الترتيب الأبجدي للخصائص Alphabetic أو على شكل مجاميع Categorized أي وفق فئات الخصائص ، حيث تُقسم الخصائص الى عدة فئات تختلف حسب نوع الكائن المحدد . مثلاً بالنسبة للنموذج تظهر الخصائص التالية وكما موضحة بالشكل رقم (1 – 5) .



شكل رقم (1 – 5) : خصائص النموذج

عند أستعراض خصائص النموذج على شكل مجاميع Categorized ، ستظهر لنا قائمة تحتوي على

- الظهور Appearance : تحتوي هذه الفئة على مجموعة من الخصائص المُتعلقة بطرق الأظهار والشكل العام للكائن مثل اللون والعنوان والخصائص الرسومية .

- السلوك Behavior : تصف قائمة الخصائص الموجودة ضمن هذه الفئة طريقة عمل الكائن بالنسبة لباقي العناصر في التطبيق .

- DDE أو (Dynamic Data Exchange) : ويُقصد بها التبادل الديناميكي وهو وسيلة الأتصال بين النماذج في التطبيق . هذه الفئة خاصة بالنماذج فقط ، اذ لاتمتلك باقي كائنات التطبيق الخصائص الموجودة في هذه الفئة .

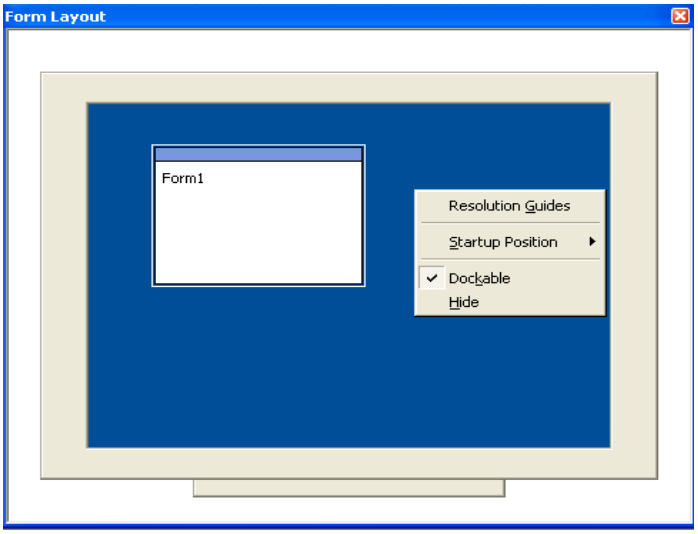
- الخط Font : تحتوي هذه الفئة على الخصائص المتعلقة بنمط الخط الذي سيظهر في الكائن مثل النوع والحجم والتأثيرات المتعلقة بالخط .

- مُتفرقات Misc : ويُقصد بها مواصفات مختلفة ،اذ تحتوي هذه الفئة على مجموعة من الخصائص ذات تأثيرات متنوعة . أغلب الخصائص الموجودة في هذه الفئة كثيرة الأستخدام أثناء تطوير التطبيقات .

- الموضع Position : تقوم الخصائص الواردة في هذه الفئة بتحديد حجم الكائن من أرتفاع وعرض وأحداثيات على النموذج .

- المقياس Scale : تُحدد هذه الخصائص مقياس الأحداثيات المستخدمة في النموذج .

ت- نافذة موقع عرض النموذج (Form Layout Window) : تُستخدم هذه النافذة لمشاهدة أو عرض موقع أو موضع ظهور النماذج على شاشة المستخدم عند تنفيذ البرنامج (التطبيق) وذلك من خلال أستخدام الزر الأيسر للفأرة ، وعند نقر الزر الأيمن للفأرة Mouse على هذه النافذة تظهر لنا القائمة التالية وكما موضح في الشكل رقم (1 – 6) ، وفي حالة عدم وجودها على النافذة الرئيسية لبرنامج لغة فيجوال بيسك فيتم تفعيلها (الوصول اليها) من خلال الأمر View والذي يظهر في شريط القوائم ومن ثم أختيار الأمر Form Layout Window .



شكل رقم (1 – 6) : نافذة موقع عرض النموذج

- خطوط الأرشاد Resolution Guides : وتحدد الحجم الأقصى للنوافذ كي لاتتجاوز حجم شاشة المستخدم . فمثلاً عند العمل على شاشة أبعادها (640 * 480) يجب ان لايزيد حجم

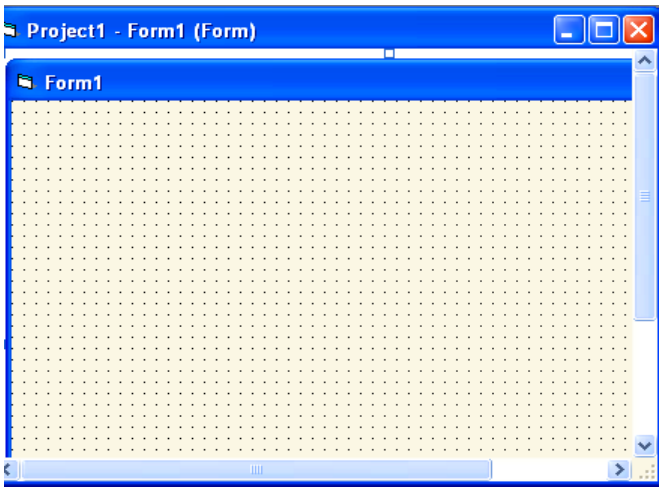
نوافذ التطبيق عن الحجم (640 * 480) لأنه في حالة زيادة حجم النافذة عن هذا الرقم فأن جزء من نافذة التطبيق لن يظهر على شاشة المستخدم أثناء التنفيذ .

- تحتوي القائمة Startup Position على عدة خيارات ، يستطيع المبرمج من خلالها تحديد موضع النموذج بشكل دقيق على شاشة المستخدم عند التنفيذ . فمثلاً لعرض النموذج أثناء التنفيذ في مركز شاشة المستخدم مهما أختلف قياسُها أو نوعها نختار الأمر Center Screen .

ث- نافذة تصميم النموذج (Project- Form والنموذج Form) : نافذة النموذج Project Form هي منطقة العمل الرئيسية لبرنامج لغة فيجوال بيسك ، اذ تحتوي على النموذج الذي يتم تصميمه والعمل عليه والذي سيكون أساس للمشروع الذي نعمل به .

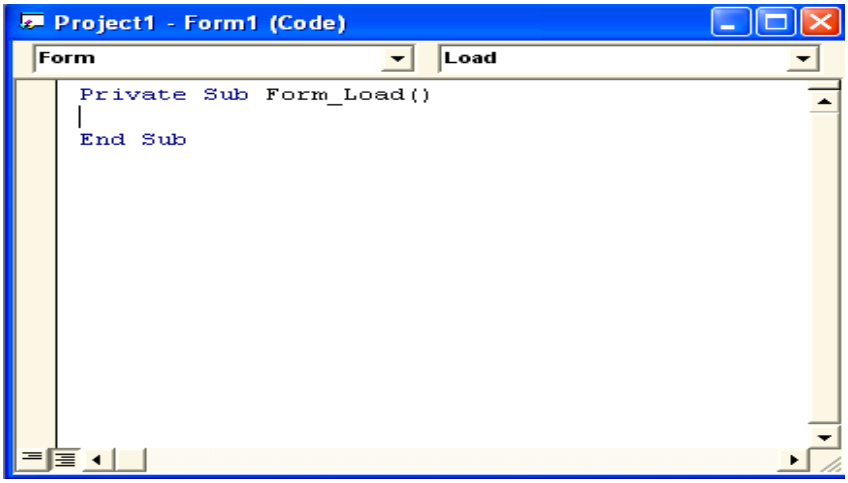
أما النموذج (Form) فهو عبارة عن نافذة فارغة وفيها نضع كُل مانُريد أن يحويه النموذج من أشكال ورسوم وأزرار وغيرها من الأدوات التي يتم أستخدامها أثناء تصميم النموذج ، وهو يشبه لوحة رسم حيث نستطيع الرسم عليه بسهولة وأيظاً تغيير حجمه ولونه وشكل الأطار الخارجي له . عنوان هذه النافذة يظهر في شريط العنوان (Title Bar) بالأسم الأفتراضي له وهو (Form1) حيث يمكن تغيير هذا العنوان من خلال تغييره من نافذة الخصائص وأختيار خاصية العنوان Caption الخاصة بالنموذج و كما موضح ذلك بالشكل رقم (1 – 7) .

ملاحظة : يُمكن أن يتألف المشروع من عدد كبير من النماذج ، كما يُمكن فتح أكثر من نموذج داخل نافذة تصميم النموذج والتنقل بين هذه النماذج عن طريق الربط بينهم .



شكل رقم (1 – 7) : نافذة تصميم النموذج

ج- نافذة البرمجة Code Window : تُستخدم هذه النافذة لكتابة الأيعازات البرمجية التي تُحدد سلوك وأسلوب عمل النماذج وعناصر التحكم . يُمكن فتح نافذة البرمجة بالنقر المزدوج لزر الفأرة الأيسر على أي عنصر تحكم داخل النموذج أو بالنقر المزدوج على النموذج نفسه ، كما يُمكن فتح نافذة البرمجة عن طريق نقر الآيقونة (View Code) الموجودة في نافذة المشروع (Project Window) أو من خلال ضغط المفتاح F7 من لوحة المفاتيح Keyboard ، فتظهر نافذة كما موضح ذلك في الشكل رقم (1 – 8) .



شكل رقم (1 – 8) : نافذة البرمجة في الفيجوال بيسك

ح- نافذة (شريط) صندوق أدوات التحكم The Toolbox Window : يحتوي هذا الصندوق على عناصر التحكم (أدوات التحكم) الأساسية التي يُمكن وضعها أو رسمها على النموذج في مرحلة التصميم والموضحة بالشكل رقم (1 – 9) . وهذه الأدوات تكون على شكل رسوم تُسمى بالكائن Object والتي تُستخدم للأستغلال المرئي والبرمجي (Visual & Code) .

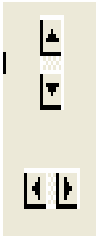
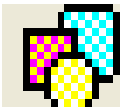
شكل رقم (1 – 9) : شريط أدوات التحكم

1 – 7 أنواع أدوات التحكم (Types of Object) : يتضمن برنامج لغة فيجوال بيسك أنواع أدوات التحكم الأساسية التالية والموضحة مع وظيفتها بالجدول رقم (1 – 1) .

جدول رقم (1 – 1) : أنواع أدوات التحكم ووظيفتها

الأداة Tool	أسم أداة التحكم Object Name	الوظيفة والوصف Description
	أداة المؤشر Pointer	وهي الأداة الوحيدة في صندوق الأدوات التي لايتم رسمها مثل باقي الأدوات . وتستخدم للتأشير على النموذج لأختيار أداة أو أكثر من أداة عند التصميم لأعطائها خصائص معينة وبرمجتها أو تحريكها أو تغيير حجمها .
	أداة صندوق الصور Picture Box	هذه الأداة بمثابة نموذج داخل نموذج ، حيث تُستخدم لوضع الصور في موقع مُحدد على النموذج ويمكن استخدامها كحاوية للأدوات الأخرى . ويتم وضع (تخصيص) الصورة في هذه الأداة عن طريق الخاصية Picture من نافذة الخصائص ، أو برمجياً بأستخدام ايعازات تحميل الصورة .
	أداة العنوان Label	تُستخدم لعرض العنوان داخل النموذج مع عدم الأمكانية للتعديل على هذا العنوان أثناء تنفيذ البرنامج ، وتُستخدم في كثير من الأحيان كعنوان لعُنصر تحكم أخر مثل Textbox . أهم خاصية لهذا العنصر هي خاصية العنوان Caption التي تحوي النص الذي ستُظهره التسمية .
	أداة صندوق النص Text Box	تُستخدم لأدخال النص والتعديل عليه من قبل المستخدم . ومن أهم الخصائص المستخدمة لهذا العنصر هي خاصية الـ Text حيث تحتوي هذه الخاصية على أسم الكائن المستخدم Text1 أي الأسم الأفتراضي والذي يمكن تبديله (أو حذفه) حسب الحاجة قبل التنفيذ .
	أداة الأطار Frame	يُستخدم الأطار كحاوية Container للأدوات الأخرى ، حيث يتم وضع عناصر التحكم داخله ومن أهم خصائص هذا الكائن هي خاصية العنوان Caption التي تكون عادةً بمثابة عنوان لمجموعة العناصر الموجودة داخل الأطار .

	أداة زر الأمر Command Button	تُستخدم هذه الأداة لتنفيذ العديد من الأوامر والأيعازات البرمجية عند النقر عليها ومن أهم الخصائص المستخدمة في هذه الأداة هي خاصية العنوان Caption التي من خلالها نكتب أسم العملية التي سنُنفذها برمجياً .
	أداة صندوق الفحص أو صندوق التحقق Check Box	تُمكن هذه الأداة من عرض عدة صناديق للفحص وتُعطي المستخدم الحرية ليختار واحداً أو أكثر من تلك الصناديق ومن أهم الخصائص المستخدمة لهذه الأداة هي خاصية العنوان الذي سيظهر جانب صندوق الفحص (التحقق) ، كذلك خاصية المحاذاة Alignment والتي تُحدد فيما اذا كان العنوان يظهر عن يمين الصندوق أو يساره أضافةً الى خاصية القيمة Value والتي تأخذ احدى القيم التالية : - 0-Unchecked : الأداة غير محددة . - 1-Checked : الأداة محددة . - 2-Grayed : مظللة بلون رمادي باهت .
	أداة زر الخيار Option Button	تُستخدم هذه الأداة في التطبيقات بشكل مجموعة داخل اطار Frame تتألف من زرين أو أكثر ، ويمكن للمُستخدم أختيار واحد فقط من الخيارات المعروضة أمامه . ومن أهم خصائص هذه الأداة هي خاصية العنوان والمحاذاة اضافةً الى خاصية القيمة Value والتي تأخذ احدى القيم التالية : - False : اذا كان الزر غير محدد . - True : اذا كان الزر محدد .
	أداة السرد والحوار Combo Box	تُستخدم لأختيار البدائل عن طريق القوائم المعروضة وتمكن المستخدم من الكتابة وتجمع بين مزايا صندوق النص وصندوق القائمة ، ويتم اضافة عناصر الى القائمة من خلال خاصية List ، اضافةً الى خاصية Text والتي من خلالها يتم مسح عنوان الأداة Combo1 لتظهر فارغة العنوان عند التنفيذ كما هو الحال بالنسبة لأداة صندوق النص .
	أداة صندوق القائمة List Box	تقوم هذه الأداة بعرض قائمة من الخيارات أمام المستخدم ليقوم بالأختيار ويتم ادخال عناصر أو قيم الى القائمة من خلال خاصية list .

	أشرطة الأنزلاق الأفقي والعمودي H Scroll Bar, V Scroll Bar	يُستخدم شريط الأنزلاق عادةً لتوليد قيمة محصورة بين حد أدنى وحد أعلى ، حيث يُتيح للمستخدم أختيار قيمة من بين مدى من القيم بشكل عمودي أو افقي . ومن أهم الخصائص المستخدمة في هذا الكائن هي : - Min : الحد الأدنى لقيمة شرط الأنزلاق - Max : الحد الأعلى لقيمة شرط الأنزلاق . - Value : القيمة الحالية لقيمة شرط الأنزلاق .
	أداة المؤقت Timer	تُمكن هذه الأداة المبرمج من القيام بمهمات مُحددة ضمن فترات زمنية مُنتظمة . ومن أهم الخصائص المستخدمة بها خاصية Interval أي الفواصل الزمنية ، حيث تُحدد في هذه الخاصية الفترة الزمنية للمؤقت ، هذا يعني أنه عند كل مرور لهذه الفترة سيتم تنفيذ اجراء المؤقت . ومن الملاحظ في هذه الأداة أنها تكون غير مرئية عند تنفيذ البرنامج .
	صندوق مُحرك الأقراص أو صندوق قائمة السواقات Drive List Box	هو عبارة عن صندوق قائمة مُنسدلة Combo Box تُعرض في قائمتها وبشكل تلقائي مُحركات الأقراص الموجودة في حاسبة المستخدم .
	صندوق الدلائل أو صندوق قائمة المجلدات Dir List Box	يُمكن المستخدم من الأختيار والبحث عن الدلائل بأستخدام المسار Path داخل مشغل (محرك) الأقراص ، وله مُعظم خصائص صندوق القائمة List Box .
	صندوق قائمة الملفات File List Box	تقوم بعرض قائمة الملفات الموجودة في المُجلد المُحدد بالخاصية Path . ومن أهم خصائصه المستخدمة هي الخاصية Pattern حيث تُحدد نوع الملفات المعروضة في القائمة ، القيمة الأفتراضية لهذه الخاصية هي *.* أي يتم عرض جميع أنواع الملفات واذا وضعنا في هذه الخاصية القيمة *.bmp فسيتم عرض الملفات ذات اللاحقة bmp.
	أداة الأشكال Shape	وهي أداة لعرض الأشكال الهندسية كالمربعات والدوائر مثلاً على النموذج . وتمتلك هذه الأداة مجموعة من الخصائص المُتعلقة بالأطار وهي : - Border Width : سماكة خط الرسم . - Border Color : لون الأطار .

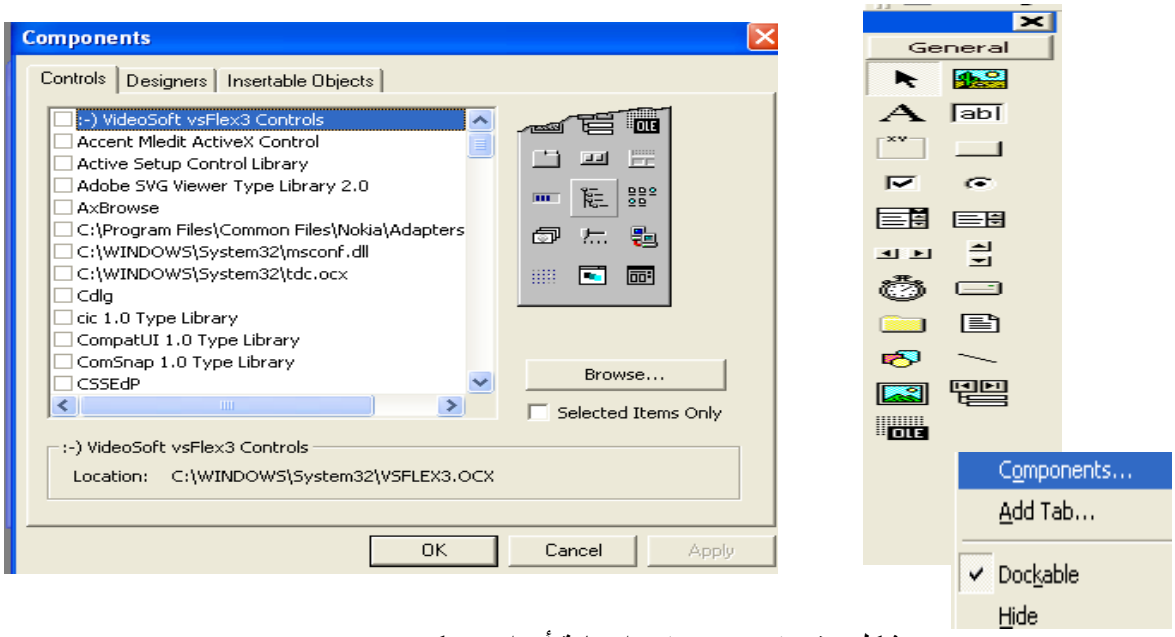
		- Back Color: لون الخلفية ، مع الخاصية Back Style حيث نجعل قيمتُها تساوي 1-Opaque .
	أداة الخط Line	تُتيح للمستخدم رسم الخطوط على النموذج ، حيث تُستخدم لفصل عناصر التحكم عن بعضها البعض أي عمل فواصل بين العناصر بحيث لاتكون مُتداخلة خصوصاً في النماذج التي تحتوي على عدد كبير من العناصر ، حيث تقوم هذه الأداة بعمل حاجز بين كل مجموعة من العناصر المُترابطة .
	أداة الصورة Image	تقوم بعرض الصور على النموذج ولايُمكن استخدامُها كحاوية للأدوات الأخرى . ومن أهم خصائصها خاصية التمدد أو التوسع Stretch حيث نجعل قيمتها تساوي True مما يجعل الصورة تتمدد حسب حجم عنصر الصورة المحددة والمختارة بالكامل بأستخدام خاصية Picture .
	عُنصر التحكم بقاعدة البيانات أو عُنصر الوصول للبيانات Data	تسمح هذه الأداة بالقيام بعملية الربط بين برنامج لغة فيجوال بيسك وقاعدة البيانات Data Base ، حيث تُستخدم هذه الأداة للتعامل مع جدول في قاعدة البيانات (قاعدة بيانات الأكسس) .
	أداة التحكم بالتطبيقات الكائن OLE	يستخدم هذا الكائن لربط وتضمين الكائنات أو العناصر داخل تطبيق الفيجوال بيسك Object Linking and Embedding ، حيث يستخدم لأدراج كائنات خارجية في التطبيق مثل ادراج جدول Excel أو برنامج الرسام . فمثلاً نقوم بربط كائن OLE ببرنامج الرسام Paint ففي مرحلة التنفيذ سيؤدي النقر المزدوج على آيقونة الرسام بفتح المُستند الذي قُمنا بأنشاءه .

1 – 8 اضافة أدوات تحكم جديدة : Adding New Controls

يحتوي صندوق الأدوات Toolbox على عناصر التحكم الأساسية التي يُمكن وضعها على النموذج في مرحلة التصميم ، كما يُمكن اضافة عناصر تحكم أُخرى تُدعى ActiveX Control ويتم ذلك من خلال أختيار الأمر التالي :

تحديد الأداة المطلوب اضافتها الى صندوق الأدوات ⟵ Components ⟵ Project

أو من خلال نقر زر الفأرة الأيمن على شريط أدوات التحكم (صندوق الأدوات) وأختيار الأمر Components من القائمة المُنبثقة وكما موضح في الشكل رقم (1 – 10) .



شكل رقم (1 – 10) : اضافة أدوات تحكم جديدة

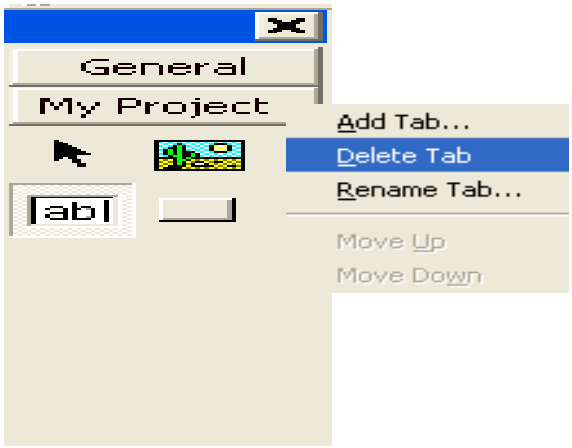
يتم تحديد الأداة أو العنصر المطلوب اضافته الى صندوق الأدوات وذلك بوضع علامة (✓) في المربع المُخصص له والضغط على الأمر (OK) وفي حالة اضافة أكثر من عنصر يتم تحديد العناصر المطلوبة جميعها والنقر على مفتاح (Apply) ومن ثم النقر على مفتاح (OK) للخروج من هذه النافذة (Components) . وأخيراً ستظهر الأداة الجديدة المضافة داخل صندوق أدوات التحكم .

في الحالة الأفتراضية توجد قائمة واحدة هي General ، ولأضافة قائمة أخرى الى صندوق الأدوات ننقر بالزر الأيمن للفأرة فوق صندوق الأدوات ونختار الأمر Add Tab ثُم نكتب أسم القائمة الجديدة في نافذة الحوار التي ستظهر لنا ثم ننقر فوق الأمر OK فنجد أن الفيجوال بيسك قد أضاف قائمة أخرى فارغة الى صندوق الأدوات لاتحوي الا على عنصر المؤشر Pointer ،

ويمكن نقل عدة عناصر من القائمة General الى القائمة الجديدة عن طريق السحب (من عناصر القائمة الرئيسية General) والأفلات (الى القائمة الجديدة التي تحمل الأسم الذي تم تحديده مُسبقاً) ، وكما موضح ذلك بالشكل رقم (1 – 11) .

شكل رقم (1 – 11) : قائمة صندوق أدوات جديدة

كذلك يمكن حذف القائمة الجديدة التي تم أنشاؤها عن طريق النقر بالزر الأيمن للفأرة على رأس القائمة الجديدة الموجودة في صندوق الأدوات ثُم أختيار الأمر (Delete Tab) من القائمة التي ستظهر ، وكما موضح في الشكل رقم (1 – 12) .



شكل رقم (1 – 12) : حذف قائمة صندوق الأدوات

كما يُمكن اعادة تسمية القائمة بأختيار الأمر (Rename Tab) ، مع مُلاحظة أنه لايُمكن حذف أو تغيير أسم القائمة الأساسية General .كذلك يمكن أيظاً عمل قائمة جديدة أخرى وبالطريقة المذكورة أعلاه والموضحة بالشكل رقم (1 – 11) .

1 – 9 اضافة وازالة عناصر التحكم الى ومن النموذج :

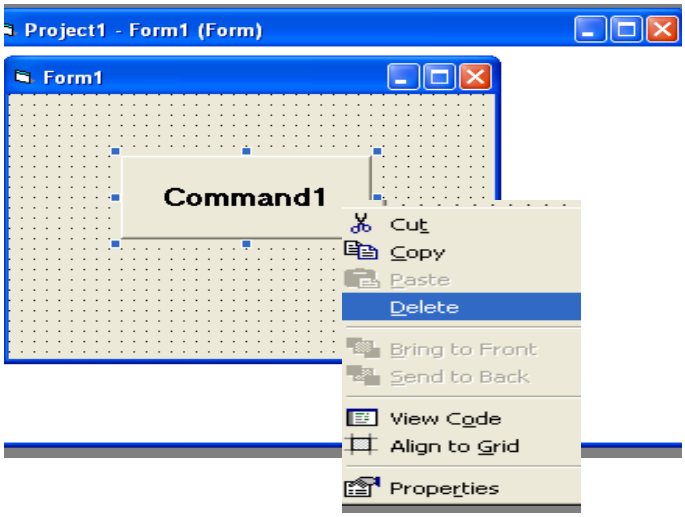
Adding & Removing Controls to / from Form

- تتم اضافة أدوات (عناصر) التحكم المُختلفة الى النموذج بأتباع مايلي :

1. النقر المزدوج بأستخدام الزر الأيسر للفأرة على آيقونة العنصر الذي نُريد اضافته ، حيث سيقوم الفيجوال بيسك بوضع العنصر في مُنتصف النموذج .
2. أو من خلال أختيار العنصر الذي نريد اضافته الى النموذج من صندوق الأدوات وذلك بالنقر مرة واحدة على آيقونة العنصر ثُم نقوم برسم العنصر في أي مكان على النموذج وذلك بأستخدام الزر الأيسر للفأرة بحيث يأخذ العنصر (الكائن) الحجم الذي نُريده ، حيث نستطيع التحكم بحجم أداة التحكم من خلال النقر عليها عندها ستظهر مُربعات حول الأداة نستطيع من خلالها تكبير أوتصغير حجم الأداة .

- يتم حذف أداة التحكم من النموذج والموضح بالشكل رقم (1 – 13) من خلال :

1. نُحدد الأداة المُراد حذفها من النموذج وذلك بالنقر عليها نقرة واحدة بأستخدام الزر الأيسر للفأرة حيث ستظهر مربعات على كُل زاوية من زوايا الآداة .
2. الضغط على مفتاح الحذف Delete من لوحة المفاتيح ، أو النقر فوق العنصر الذي نريد حذفه بالزر الأيمن للفأرة ونختار الأمر Delete من قائمة الأوامر التي ستظهر عند النقر .



شكل رقم (1 – 13) : ازالة عناصر التحكم من النموذج

1 – 10 تسمية آدوات التحكم : Naming Of Controls Tools

يُعتبر أسم آداة التحكم من أهم الخصائص التي تُميز تلك الأداة ، حيث يقوم برنامج لغة الفيجوال بيسك بأعطاء أسم آفتراضي (Default) لكل كائن نقوم برسمه أو وضعه على النموذج . فمثلاً عند اضافة زر تحكم Command button الى النموذج فأن البرنامج يُعطيه أسم تلقائي أو أفتراضي هو Command1 وعند اضافة زر تحكم ثاني فيُعطيه الأسم Command2 وهكذا بالنسبة للكائنات الأخرى مثل Text1 و Picture1 ... الخ .

في بعض الأحيان الأعتماد على التسمية التلقائية التي يقترحها البرنامج مُربك بعض الشئ خاصة اذا كان المشروع يحتوي مثلاً على (100) أداة تحكم ، حيث أن آخر آداة سيكون آسمها بالنسبة لكائن زر الأمر هو Command100 وهذه الأسماء الأفتراضية لاتدلل على محتوياتها ، لذلك في هذه الحالة لابُد من القيام بأعادة تسمية Rename لأدوات التحكم تبعاً للغرض الذي تقوم به . وتتم عملية التسمية كما يلـي :

1. اضافة آداة التحكم الى النموذج بأستخدام احدى الطريقتين التي تم ذكرهما سابقاً .
2. من خصائص الأداة أو الكائن Properties Window نختار خاصية الأسم Name الموجودة في بداية قائمة الخصائص .
3. حذف الأسم التلقائي الموجود وآضافة الأسم الجديد ثُم الضغط على مفتاح الأدخال Enter من لوحة المفاتيح .

ملاحظـة : كُل كائن من كائنات أداة التحكم له خاصية الأسم Name أي الأسم البرمجي للكائن ، وهذه الخاصية عبارة عن كلمة واحدة اذ لايمكن أن تحتوي على فراغ أو فاصل Space ، كما يُفضل أن يكون الأسم صغيراً كي يسهُل التعامُل معه برمجياً .

يستخدم الأشخاص الذين يتعاملون مع برنامج لغة الفيجوال بيسك مايُسمى بالبادئة (Prefix) ، وهو اعطاء ثلاثة أحرُف من بداية آسم كُل آداة من آدوات التحكم ، ثُم يُتبع هذه الأحرف آية أحرف اضافية لتدُل على وظيفة آداة التحكم تُسمى بالاحقة (Suffix) . فمثلاً اذا أردنا تسمية صندوق نص يحوي بداخله أسم الزبون فالأسم المُقترح لآداة التحكم في هذه الحالة هو (TxtCustName) حيث أن البادئة (Txt) تدل على نوع عُنصر التحكم (Text Box) والاحقة (CustName) تدل على أن صنوق النص مُخصص لوضع أسم الزبون فيه . وكذلك أداة التحكم التي تكون وظيفتها انهاء البرنامج قد يتم تسميتها (CmdClose) .

الجدول رقم (1 – 2) يُبين البادئات الأكثر شيوعاً لأدوات التحكم التي يستخدمُها الفيجوال بيسك للعناصر المختلفة (البادئة المُقترحة) .

جدول رقم (1 – 2) : البادئات الأكثر شيوعاً لأدوات التحكم

أداة التحكم Control	البادئة Prefix	أداة التحكم Control	البادئة Prefix
Command Button	cmd	DirListBox	dir
List Box	lst	Shape	shp
Text Box	txt	Image	img
Combo Box	cbo	Option Button	opt
Check Box	chk	Line	lin
Picture Box	pit	H Scroll Bar	hsb
Form	frm	V Scroll Bar	vsb
Frame	fra	OLE Container	oie
Label	lbl	Drive List Box	drv
Timer	tmr	File List Box	fil

1 – 11 حُفظ المشروع : Saving the New Project

المشروع Project في برنامج لغة فيجوال بيسك هو مجموعة من الملفات التي تُشكل بمجموعها في النهاية البرنامج . هذه الملفات هي عبارة عن مجموعة من النماذج Forms والبرامج الفرعية Modules والرسومات البيانية Graphics . الشئ المُهم ذكره هُنا عن المشروع أن عملية الحُفظ لكُل من النماذج والبرامج الفرعية والرسومات البيانية تتم في ملفات مُنفصلة . يتم خزن كُل المعلومات عن ملفات البرنامج Program Files في ملف المشروع Program File ، حيث يتم عرض جميع الملفات الموجودة في البرنامج (عند الحفظ يجب حفظ كُل النماذج والبرامج الفرعية والرسومات البيانية ان وجدت ليكون المشروع مُتكاملاً وجاهزاً للأستخدام أثناء التشغيل مرة ثانية) . أيظاً تتم الأشارة الى أسم المشروع ، حيث أن آمتداد ملفات المشروع هـي (vbp.) .

عملية حفظ المشروع تتم بنقل المؤشر الى قائمة ملف File ثُم أختيار الأمر حُفظ المشروع Save Project .

File ⟶ Save Project ⟶ Save in

Save in : يُستخدم لتحديد الموقع أو المكان الذي سيُحفظ فيه المشروع (Directory) وهذا الأمر يظهر للمرة الأولى فقط عند حفظ المشروع .

نُلاحظ أنه عند حفظ المشروع للمرة الأولى فأن البرنامج سيُعطي أسم تلقائي للمشروع هو Project1 ، حيث نستطيع تغيير هذا الأسم بسهولة وأعطاء أسم جديد للمشروع وبعد الأنتهاء من عملية الحفظ سيظهر أسم المشروع الجديد في نافذة مُستكشف المشروع(Project Window).

1 – 12 الخصائص والأحداث : Properties & Events

1. الخصائص (Properties) : هي وصف للعُنصر أو الأداة ، اذ أن لكُل آداة من صندوق الأدوات Tool Box مجموعة من الخصائص تصف الآداة . فبالرغم من أن الخصائص غير مُتطابقة لكُل الأدوات الا آنه توجد مجموعة شائعة في أغلب الخصائص مثل (Name,Enabled,Visible) وغيرها من الخصائص المُشتركة .
2. الأحداث (Events) : هـي مايحدث عندما تقوم آداة بعمل مُعين ، أو هي عملية يتعرف عليها النموذج أو عنصر التحكم ، حيث تُنشأ الأحداث أستجابةً لفعل مُعين قام به المستخدم مثل النقر بالفأرة أو الضغط على أحد مفاتيح لوحة المفاتيح وهكذا .

مُلاحظة : لاتمتلك جميع أدوات التحكم نفس الأحداث ، ولكن هُناك مجموعة من الأحداث المُشتركة بين العديد من الأدوات وكما موضح بالجدول رقم (1 – 3) .

جدول رقم (1 – 3) : الأحداث المشتركة للعديد من أدوات التحكم

الحدث Event	وقوعه Occurrence
Click	يحدث عند النقر المفرد فوق الكائن (الأداة)
Dbclick	يحدث عند النقر المزدوج فوق الكائن (الأداة)
LostFocus	يحدث عندما يفقد كائن ما التركيز
GotFocus	يحدث عندما يحصل كائن ما التركيز
KeyPress	يحدث عند الضغط على أحد مفاتيح لوحة المفاتيح

مُلاحظة : يُمكن أستخدام خصائص الكائن لأستدعائه برمجياً وتنفيذ عمليات مُعينة ، وذلك يتم في نافذة البرمجة (Code Window) . فمثلاً لتغيير لون خلفية النموذج برمجياً نقوم بكتابة الشفرة (Code) التالية :

```
Private Sub cmdcolor_Click()
Form1.BackColor = QBColor(1)
End Sub
```

- الدالة QBColor تُعطي لوناً مختلفاً حسب القيمة الرقمية الموضوعة بين القوسين ، وهذه القيمة يكون مجالُها محصوراً بين (0- 15) وبهذا فأن هذه الدالة تولد (16) لوناً مُختلفاً . لاحظ الجدول رقم (1 – 4)
- أن الرقم (1) داخل القوسين يعني ظهور اللون الأزرق .

الجدول رقم (1 – 4) : الألوان الستة عشر والأرقام المُخصصة لها .

Number	Color	Number	Color
0	Black	3	Cyan
1	Blue	4	Red
2	Green	5	Magenta
6	Yellow	11	Light Cyan
7	White	12	Light Red
8	Gray	13	Light Magenta
9	Light Blue	14	Light Yellow
10	Light Green	15	Bright White

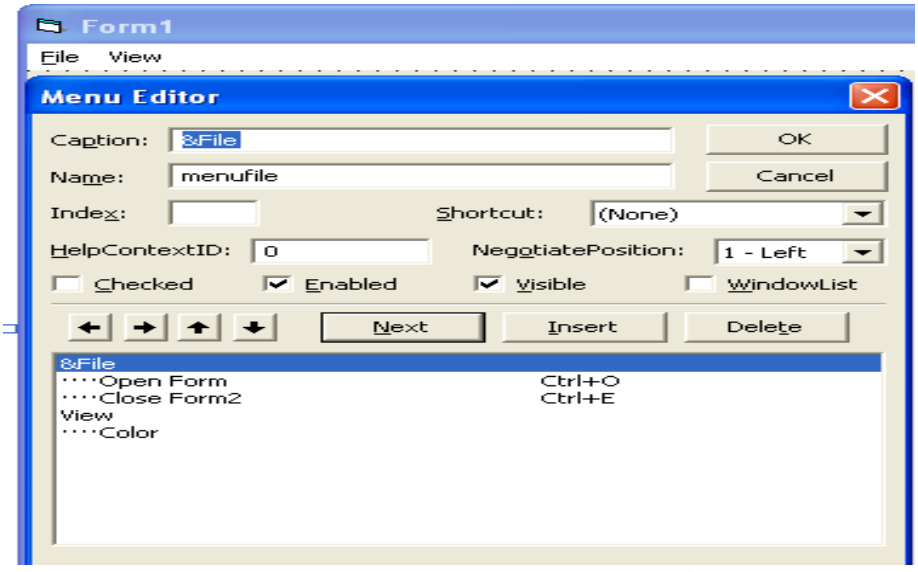
1 – 13 التعامل مع القوائم : Menu Editor

تُعتبر القوائم من العناصر الأساسية في بناء التطبيقات بأستخدام الفيجوال بيسك خصوصاً عندما يكون لدينا عدد كبير من الأوامر ، حيث يُمكن اعتبار كُل عنصر من عناصر القائمة كزر آمر نكتُب عليه اجراء (Code) يُنفذ عند النقر على العنصر .

لغرض تصميم مُحرر القوائم Menu Editor نقوم بالخطوات التالية :

1. نضغط المفتاحين (Ctrl + E) من لوحة المفاتيح أو من النقر على الأيقونة الموجودة في أقصى يسار شريط الأدوات Toolbar أو من خلال أختيار الأمر Tools من شريط القوائم Menu bar ثم النقر على الأمر مُحرر القوائم Menu Editor .
2. نقوم بأدخال أوامر الى القائمة وليكن على سبيل المثال الأمر ملف File وذلك بوضعه داخل حقل العنوان Caption مع اعطاء أسم خاص به في حقل الأسم Name ثم النقر على الأمر Ok فيتم تكوين قائمة الملف File . كذلك يمكن اضافة قوائم أخرى وبنفس الطريق السابقة بحيث تتكون لدينا مجموعة من القوائم تظهر جميعها على النموذج ويمكن عمل برنامج لكل قائمة من هذه القوائم بحيث يؤدي وظيفة مُعينة ، حيث يتم التعامل مع هذه القوائم بأعتبارها كائنات يمكن برمجتُها كما هو الحال في أدوات التحكم وذلك بالنقر عليها .

مما تقدم أعلاه مُمكن بأستخدام محرر القوائم Menu Editor الموضح بالشكل رقم (1 – 14) تكوين شرط قوائم Menu bar وكل قائمة من هذا الشريط تحتوي على مجموعة من الأوامر التي تُنفذ عند النقر عليها كما هو الحال في شريط القوائم الموجود في برنامج لغة فيجوال بيسك .

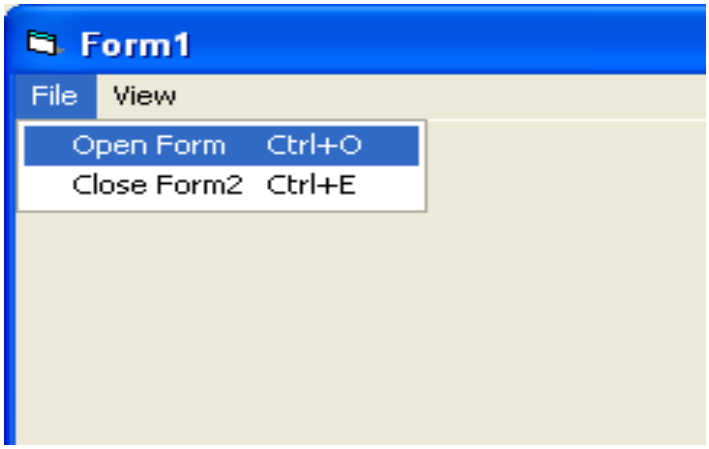


شكل رقم (1 – 14) : محرر القوائم

ملاحظات :

- الخاصية Caption : تُستخدم لكتابة النص الذي سيُعرض كعنوان للأمر المُراد اظهاره داخل القائمة أمام المستخدم .
- الخاصية Name : تعني تخصيص أسم برمجي لكل أمر (أو عنصر) في القائمة .
- السهم المُتجه نحو اليمين في نافذة مُحرر القوائم يعني اضافة أربع نقاط أمام الأمر (Open Form) مما يُعطي قائمة فرعية واحدة من قائمة رئيسية (أي قائمة الملف File) ، أما السهم المتجه نحو اليسار فيمحو التأشير أي الغاء القائمة الفرعية .
- يُمكن تعيين لكل عنصر من عناصر القائمة الفرعية (فقط) مفتاح أختصار لجعل أستخدام الأمر يبدو أكثر سهولة وبشكل مُباشر من خلال أستخدام لوحة المفاتيح وذلك يتم من خلال الخاصية طريق مُختصر (Shortcut) .

سنقوم الآن ببرمجة الأوامر التي تم تكوينها داخل محرر القوائم والموضحة بالشكل رقم (1 – 15).



شكل رقم (1 – 15) : برمجة الأوامر داخل محرر القوائم

```
Private Sub menuclose_Click()
Form2.Hide
End Sub
```

```
Private Sub menucolor_Click()
Me.BackColor = QBColor(Rnd * 15)
End Sub
Private Sub menuopen_Click()
```

```
Form2.Show
End Sub
```

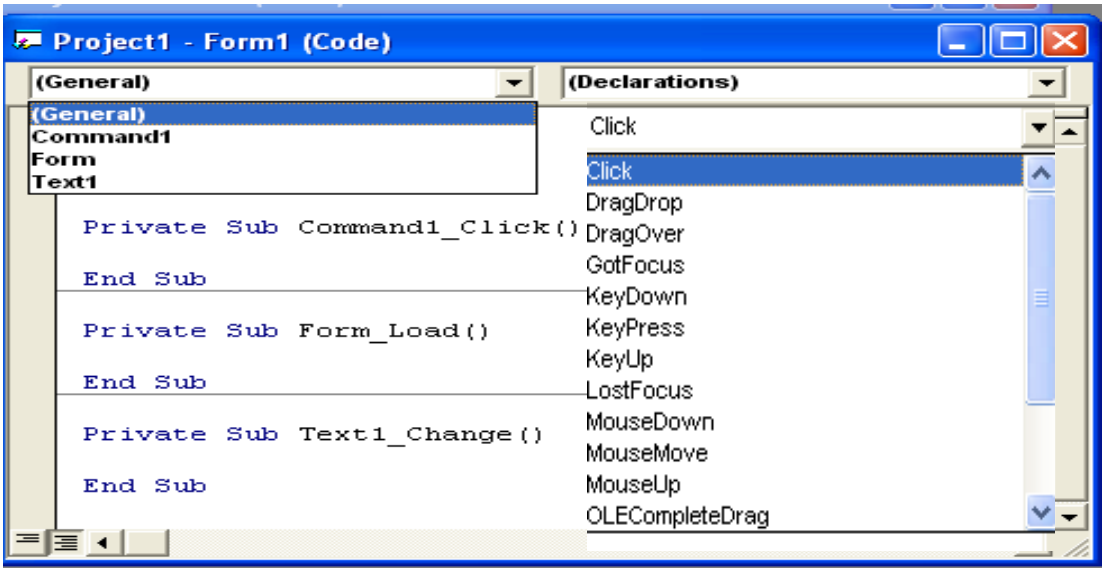
1 – 14 البرمجـة بلغة فيجوال بيسك : Programming In VB

عملية كتابة أي برنامج بلغة فيجوال بيسك تمرُ بمرحلتين :

1. مرحلة البرمجة المرئية Visual Programming .
2. مرحلة كتابة الشفرة Code Programming .

حيث يتم في مرحلة البرمجة المرئية تصميم البرنامج عن طريق أستخدام الأدوات البرمجية الموجودة مع حُزمة الفيجوال بيسك وكما تم شرحها مسبقا . أما مرحلة كتابة الشفرة فتتم عن طريق كتابة البرنامج الفرعي (Code) داخل نافذة الشفرة (Code Window) والتي تكون عادةُ مخفية عند التصميم ، وتظهر عند النقر المزدوج بزر الفأرة الأيسر في أي مكان من على النموذج أو على أي كائن (آداة) داخل النموذج أو من خلال النقر على الآيقونة الموجودة في نافذة المشروع (Project Window) .

عند الدخول الى نافذة الشفرة نُلاحظ وجود مُربيعين لقائمتين مُنسدلتين وكما موضح في الشكل رقم (1 – 16) .

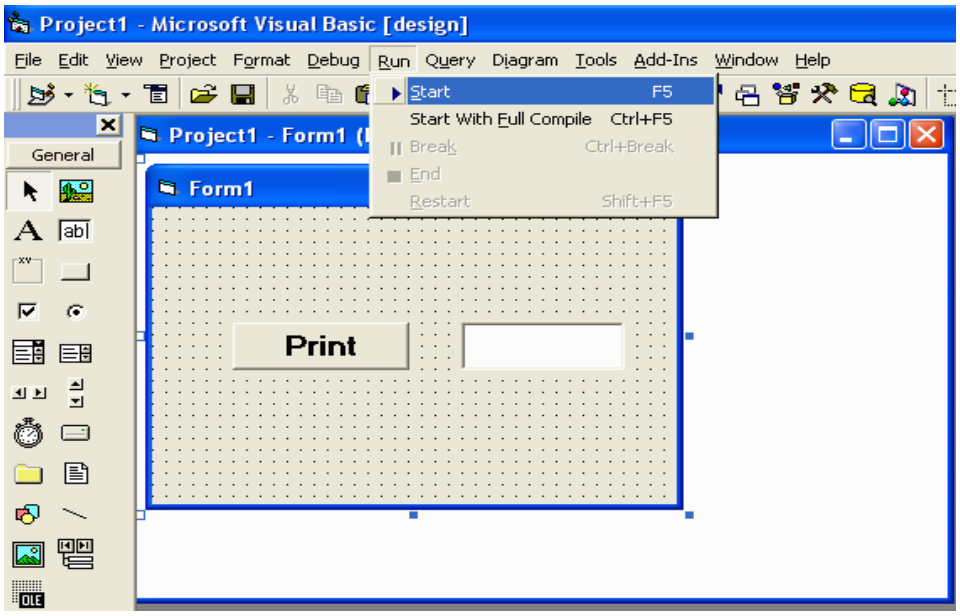


شكل رقم (1 – 16) : نافذة كتابة الشفرة

1. قائمة General : تحتوي هذه القائمة على أسماء جميع الكائنات التي يحتويها النموذج ، بالأضافة الى النموذج نفسه . ويُمكن من خلال تحديد الكائن المطلوب عمل برمجة عليه وبشكل مُباشر من غير النقر على الكائن مرتين في نافذة التصميم .

2. قائمة Declarations : تضم هذه القائمة جميع الأحداث التي يُمكن كتابة الشفرة فيها لكُل كائن ، فمثلاً الحدث Click يقوم بتنفيذ الخطوات البرمجية Code الموجودة داخل الكائن وذلك عند النقر بزر الفأرة على الكائن عند التنفيذ .

ملاحظة : يتم تنفيذ برنامج لغة فيجوال بيسك من خلال أختيار الأمر (نفذ Run) من شريط القوائم ثم أختيار الأمر (أبدأ Start_) ، أو من خلال الضغط على مفتاح F5 من لوحة المفاتيح وكما موضح ذلك في الشكل رقم (1 – 17) .



شكل رقم (1 – 17) : تنفيذ برنامج لغة فيجوال بيسك

1 – 15 تحليل الأجراء الحدثي : Analyzing the event procedure

يتكون الأجراء الحدثي في برنامج لغة فيجوال بيسك من الأجزاء التالية :

- Private Sub : الأجراء الفرعي ، يمثل بداية الأجراء .
- Private : يعني أن هذا الأجراء لايُنفذ الا من نفس النموذج .
- Sub : تُحدد بداية الأجراء ويليها أسم الأجراء .
- Command1 : أسم آداة التحكم المُراد برمجتها .

- Click()_ : نوع الحدث المُستخدم لتنفيذ أداة التحكم وفي هذه الحالة يعني النقر المفرد بالفأرة .
- End Sub : تُمثل نهاية الأجراء الحدثي .
- ويتم حصر جميع الشفرة المكتوبة لكل حدث للكائن بين السطرين التاليين :

```
Private Sub menuclose_Click()

End Sub
```

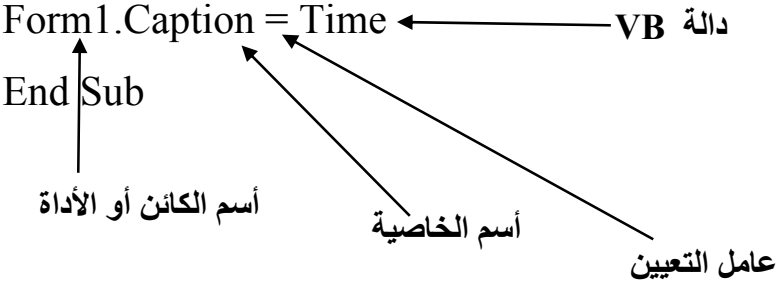
ملاحظـة : يُسمى كُل سطر من الشفرة في برنامج لغة فيجوال بيسك بجملة برمجية ، والجملة البرمجية في هذه الحالة تُمثل تركيبة من كلمات فيجوال بيسك ومن الخصائص والدالات والعوامل والرموز .

يمكن أن تكون الجُملة البرمجية بأكملها مُجرد كلمة دليلية بسيطـة ، كما في المثال التالي :

```
Private Sub Command1_Click()
Beep
End Sub
```

والتي تُصدر نوتة (صوت) من مُكبر الصوت في الحاسبة ، أو قد تكون الجملة البرمجية تركيبة من عدة عناصر كالجُملة التي تُعين الوقت الحالي الى الخاصية Caption وكما موضح ذلك في المثال الأتي :

```
Private Sub  Form_Load()
Form1.Caption = Time
End Sub
```




```
Private Sub Command1_Click()
Label1.Caption = Date
End Sub
```

1 – 16 العناصر الأساسية لجمل لغة فيجوال بيسك : Basic elements in VB

أي جملة برمجية في لغة الفيجوال بيسك تتكون من العناصر الرئيسية التالية :

1 – 16 – 1 البيانات : Data

تُعتبر البيانات أساس آي لُغة برمجية وتكون على نوعين المُتغيرات والثوابت .

أ-المُتغيرات (Variables) : المُتغير هو عنوان منطقة محجوزة في الذاكرة ، حيث يُمكن الكتابة فيها والقراءة منها . يُعتبر المُتغير مكان تخزيين مؤقت للبيانات في البرنامج ، يُمكن أستخدام مُتغير واحد أو أكثر في البرنامج ومن المُمكن أن يحتوي المُتغير على كلمات أو أرقام أو تواريخ . المُتغيرات مُفيدة لأنها تُتيح اعطاء أسم صغير من السهل تذكره .

يتم التصريح عن المُتغيرات بطريقتين :-

- التصريح العلني (الصريح) Explicit : يتم التصريح عن المُتغير (تعريفه) علنياً قبل أن نستعمله(عادةً في بداية الأجراء الحدثي) وذلك بكتابة اسم المُتغير بعد العبارة Dim أو Private أو Public ، حيث يتم حجز مساحة (مكان) في الذاكرة لهذا المُتغير عندما يشتغل البرنامج .

Example :

```
Dim X As Integer, Y As String
```

ملاحظة : في حالة عدم ذكر نوع المُتغير سيقوم الفيجوال بيسك افتراضياً بحجز مساحة لنوع متغيرات يُدعى (مُتنوع Variant) يُمكنه احتواء بيانات من أي نوع حجم أو أي تنسيق .

بعد كتابة أسم المُتغير يُمكن تحديد نوعه لنتمكن من التحكم بكمية الذاكرة التي سيستعملها البرنامج، مثلاً اذا كان المُتغير سيحتوي على رقم صغير خال من آية مرتبة عشرية (أي عدد صحيح) يُمكن تصريحه (تعريفه) على آنه عدد صحيح وبذلك سنوفر من مساحة الذاكرة

```
Dim X As Integer
```

- التصريح الضمني Implicit : يتم التصريح عن المُتغير ضمنياً وذلك بأستعمال المُتغير لوحده مُتجاهلين الجُملة Dim كُلياً .

```
Y = "Ali"
```

يتميز التصريح الضمني بالسُرعة اذ لايتم تضييع الوقت في كتابة الجملة Dim ، حيث أن هذه الطريقة لاتُجبرنا على تنظيم المُتغيرات وسردها مُسبقاً .

ملاحظـة : يُمكن تحديد بعض أنواع البيانات من خلال كتابة حرف تصريح النوع بعد أسم المُتغير . مثلاً يُمكن تصريح المُتغير X على آنه عدد صحيح من خلال اضافة الحرف % الى نهاية أسمه لاحظ الجدول رقم (1 – 5) . لذا يعتبر برنامج لغة الفيجوال بيسك جُملتي التصريح التاليتين مُتشابهتين :

Dim x As Integer

OR

Dim X%

جدول رقم (1 – 5) : أنواع البيانات الرئيسية في برنامج لغة فيجوال بيسك مع وصفها .

Data Types	Declaration
Integer عدد صحيح -32768<X<32768	Dim X As Integer Dim X% X% =20
Long عدد صحيح طويل يحتوي على كسور	Dim X As Long Dim X& X& =200000
Byte عدد صحيح 0-255	Dim X As Byte
Single عدد حقيقي يحتوي على كسور	Dim X As Single Dim X! X! =368.9
Double عدد حقيقي طويل	Dim X As Double Dim X# X# =3.14
Currency عملة	Dim X As Currency Dim X@ X@ =20
String سلسلة حروف	Dim X As String Dim X$ X$ =”Ali”

Dim X	Variant متنوع تضم جميع أنواع البيانات المذكورة أعلاه
Dim X As Boolean X = True	Boolean قيمة منطقية
Dim X As Date X = #5-3-1970#	Date تأريخ

ب - الثوابت (Constants) : هي القيم التي تبقى ثابتة خلال تنفيذ البرنامج . الثوابت مُفيدة لأنها تُزيد من سهولة قراءة شفرة البرنامج ، كما توفر من كمية الذاكرة المُستعملة وتُسهل تنفيذ المُتغيرات العامة . تعمل الثوابت كالمُتغيرات الى حد كبير ، الا آنه لايُمكن تغيير قيمتها خلال التنفيذ. يتم الأعلان أو التصريح عن الثوابت بواسطة الكلمة Const .

Private Const Pi = 3.14 ، لجعل الثابت مُتوفراً لكل الكائنات والأجراءات وذلك من خلال وضعه في قسم التصاريح من النموذج (السطر الأول في نافذة الشفرة) .

Public Const Pi = 3.14 ، لجعل الثابت مُتوفراً في كُل النماذج والوحدات النمطية من البرنامج (أي ليس في النموذ ج الحالي فقط) .

1 – 16 – 2 التعابيـر : Expressions

وتُقسم الى قسمين ، همـا :

أ- التعابير الحسابيـة (Mathematic Expression) : وهي التعابير التي يكون ناتجها قيمة رقمية ، مثل Z = A + B .

ب- التعابير المنطقية (Logical Expression) : هي التعابير التي يكون ناتجها قيمة منطقية (True or False) ، مثل Z = A Or B .

1 – 16 – 3 المُشغلات : Operators

وهي العمليات التي يُمكن أجراءها في برنامج لغة الفيجوال بيسك ، وهناك عدة أنواع من المُشغلات وهـي :

أ- المُشغلات الحسابية (Arithmetic Operators) : وتشمل العمليات الرياضية من الجمع والطرح والضرب والقسمة .

ب- المُشغلات العلائقية (Relational Operators) : وتشمل علامة المُساواة والأصغر والأكبر والأصغر من والأكبر من واللامُساواة .

ت- المُشغلات المنطقية (Logical Operators) : وتشمل علامة AND, OR, XOR, Not

ث- مُشغلات البيانات (String Operators) : وتقوم بعملية الدمج بين سلسلتين وذلك بأستخدام علامة الجمع (&) .

1 – 16 – 4 الدوال : Functions

تقوم الدوال بأجراء بعض التعديلات في البرنامج كتحويل قيمة من نوع الى نوع أخر أو حساب نتيجة تعبير رياضي مُعقد أو أعتماد مبدأ العشوائية في البرنامج ، اضافةً الى امكانية العمل على الأرقام في الجُمل . يتم أستعمال الدالات الرياضية في جُملة برمجية ، حيث انها ستُعيد قيمة الى البرنامج . لاحظ الجدول رقم (1 – 6) ، حيث تُمثل (X) الرقم أو المُتغير الذي نُريد أستخراج قيمتهُ من الدالة .

جدول رقم (1 – 6) : الدوال الشائعة الأستخدام مع برامج لغة فيجوال بيسك .

Function	Value Returned	Function	Value Returned
Int(X)	Integers of X	Cstr(X)	Convert X variable to string
Abs(X)	Absolute of X	Chr(X)	Convert number X to one character
Fix(X)	Take the integer part	Asc(X)	Convert first character of X to ASCII number
Rnd(X)	Create random number value between 0 and 1	Val(X)	Numeric value of a given string
Sqr(X)	Square root of X	Len(X)	Number of character of variable X

Sgn(X)	= -1 if X<0 = 0 if X=0 =1 if X>0	Z = Lcase(X)	Change to small letters
Exp(X)	e^{x}	Z = Ucase(X)	Change to capital letters
Log(X)	Natural Logarithms	StrReverse(X)	Reverse String
Sin(X), Cos(X)	Trigonometric Function	Mid(X,L,R)	Take only characters between L and R
Str(X)	Change X numeric to string	Replace (X,X1,X2)	Replace X1 by X2

1 – 17 أنواع الجُمل في لُغة فيجوال بيسك : Types of Statements In VB

يتم كتابة جُملة لُغة فيجوال بيسك (Code) داخل نافذة الشفرة (Code Window) ، وتتألف شفرة البرنامج من الخطوات التالية :-

1. تعريف المُتغيرات ان وجدت (Declaration Variables) .
2. ادخال القيم للمُتغيرات (Input) .
3. اجراء العمليات الحسابية والمنطقية اللازمة لتنفيذ البرنامج (Processing) .
4. طباعة النتائج (Output) .

ومن أنواع جُمل لُغة فيجوال بيسك هـي :-

1 – 17 – 1 جُمل التعليق : Comment Statements

تستعمل هذه الجُمل في أي مكان من البرنامج لشرح وتوضيح مايتضمنه البرنامج ، أو لذكر مُلاحظات أثناء عمل البرنامج . وهذه الجُمل تكون غير قابلة للتنفيذ ، حيث يتم وضع علامة أقتباس مُفردة أمام سطر البرنامج وبهذه الحالة سيتم توقيف أو تعطيل عمل هذا السطر أثناء تنفيذ البرنامج ،

كذلك مُمكن أن تُستخدم جُملة التعليق في نهاية السطر ، حيث يتم وضع علامة أقتباس مُفردة وتليها جُملة التعليق (المُلاحظات) .

Example :

```
‘ X = X MOD I
```

OR

```
I = I + 1  ‘ Increment the counter
For I = 1 to 10  ‘ Loop 10 times
```

1 – 17 – 2 جُمل التصريح : Declaration Statements

تُستعمل هذه الجُمل للتصريح عن المُتغيرات وتحديد أنواعها لنتمكن من التحكم بكمية الذاكرة التي سيستعملُها البرنامج .

Example :

```
Dim X As Integer, Y As String
```

1 – 17 – 3 جُمل التخصيص : Assignment Statements

تُستعمل هذه الجُمل لوضع (تخصيص) القيمة وتخزينها للمُتغيرات بكافة أنواعها .

Example :

```
Sum = 500
```

المتغير — قيمة المُتغير

1 – 17 – 4 الجُمل الشرطية : Conditional Statements

وتُسمى أيظاً بالبُنى الشرطية أو جُملة القرار (Decision Statement) حيث تُستخدم للتحكم في تنفيذ أجزاء من البرنامج مُعتمدة على شروط مُعينة ، أي تحديد فيما اذا كانت مجموعة من التعليمات ستُنفذ أم لا وذلك حسب قيمة تعبير مُعين .

هُناك نوعان رئيسيان من جُمل القرار هُما : (جُملة If … Then وجُملة Select Case) .

أ- جُملة If … Then Statement : وتُستخدم لتحديد فيما اذا كانت مجموعة من التعليمات ستُنفذ أم لا وذلك وفق قيمة تعبير مُعين ، أو وفق شرط مُعين وتُدعى بالمُعاملات العلائقية والتي تقوم بأختبار شروط للتأكد من آنها صحيحة أو خاطئة (True or False) . هذا النوع من الجُمل يستخدم مُعاملات المُقارنة (Conditional Operators) .

هُناك عدة صيغ لكتابة جُملة If … Then ، ومنها :-

1. Simple If … Then Statement : وتستخدم لتنفيذ آمر واحد فقط في حالة تحقق الشرط (أي تنفيذ تعليمة أو مجموعة تعليمات وفق شرط مُعين ، والصيغة العامة لهذه الصيغة كالأتي :

```
If < Condition > Then < Statement >
```

```
If  X = 3 Then
   Y = X * 2
End If
```

كما ويمكن أيظاً كتابة هذه الصيغة على سطر واحد فقط ، وكالأتي :

```
If Condition Then Statement1 : Statement2 : Statement3
```

```
If  X<Y Then X = Y : Y = Z
```

في مثل هذه الحالة لانحتاج لأضافة العبارة End If لأن صيغة If … Then جاءت على سطر واحد فقط .

```
If X = 3 Then Y = X * 2
```

2. Block If … Then Statement : تُستخدم لتنفيذ مجموعة آوامر في حال تحقق الشرط . الصيغة العامة لها تكون كالأتي :

```
If < Condition > Then
   < Statement1 >
   < Statement2 >
         .
         .
End If
```

في مثل هذه الحالة يجب اضافة عبارة End If لأن صيغة If … Then جاءت على أكثر من سطر .

```
If  X = 3 Then
    Y = X * 2
    Z = X + Y
End If
```

مما تقدم في آعلاه نُلاحظ أن المثالين التالين مُتكافئين :

```
If  X < Y Then X = Y : Y = Z
```

OR

```
If  X < Y Then
   X = Y
   Y = Z
End If
```

3. If … Then … Else : في هذه الحالة لدينا شرط مُعين نُريد من خلاله أن نُنفذ تعليمات مُعينة ، وفي حالة عدم تحقق الشرط نُنفذ تعليمات أُخرى . الصيغة العامة هي كالأتي :

```
If < Condition > Then
  < Staetment1 > [ If True ]
Else
  < Statement2 > [ If False ]
End If
```

Example :

```
Private Sub Command1_Click()
X = InputBox("Enter the number", "Result")
If X < 0 Then
  y = X ^ 2
  Print y
Else
   y = Sqr(X)
    Print y
End If
End Sub
```

4. Multiple If Statement : يتم في مثل هذا النوع من الجُمل القيام بفحص عدة شروط حتى الوصول الى الجواب (الخيار) الأفضل . أي توجد عدة كُتل من التعليمات ، كُتلة واحدة من بينها ستُنفذ فقط . الصيغة العامة لهذه الجُمل هي :

```
If  < Condition1 > Then
    < Statement1 >
Else If < Condition2 > Then
        < Statement2 >
Else
        < Condition3 >
End If
```

Example :

```
Private Sub Command1_Click()
x = InputBox("Enter the number", "Result")
If x > 0 Then
  y = Sqr(x)
  Print y
  ElseIf x = 0 Then
  y = MsgBox("Enter another number", , "No Result")
  Else
  y = x ^ 2
  Print y
End If
End Sub
```

نُلاحظ من المثال أعلاه أن الفيجوال بيسك سيقوم بأختبار أول شرط (Condition1) فأن وجده مُحقق (True) سيقوم بتنفيذه ، وفي حالة كون الشرط غير مُحقق (False) سينتقل لأختبار الشرط الثاني (Condition2) وهكذا حتى يجد شرطاً مُحققاً عندها سينفذ كُتلة التعليمات الموافقة لهذا الشرط ثُم ينتقل للتعليمات التي تلي عبارة End If (أي أنه لا يختبر بقية الشروط الأخرى بل يُنفذ شرط واحد فقط) .

ب- جُملة الأختيار Select Case : تُستخدم هذه الجُملة عندما نُريد أن نُنفذ مجموعة من الجُمل والتي تعتمد على مُتغير واحد ، أي أختيار كُتلة تعليمات من بين مجموعة من الكُتل وهذا النوع من جُمل الأختيار يشبه الأنواع السابقة الى أنه يجعل التعليمات البرمجية تبدو أكثر وضوحاً وأسرع في التنفيذ . ويتم تنفيذ هذا النوع من الجُمل بالشكل التالي :

1. يتم فحص أو تحديد قيمة المُتغير .
2. اجراء عملية المُقارنة مع جميع جَمل الـ Select فأن تساوت قيمة المُتغير مع احدى قيم الـ Select فسيتم تنفيذ الجُمل التابعة لها .

الصيغة العامة لهذه الجُمل هي كالآتـي :

```
Select Case test expression
        [ Case expression list1
                [ Statement block-1]]
        [ Case expression list2
                [ Statement block-2]] …
        [ Case Else
                [ Statement block –n]]
End Select
```

مُلاحظـة : العبارة (test expression) مُمكن أن تكون قيمتُها أو نوعها عددي أو حرفي (أي حسب طبيعة البرنامج) .

Example :

```
Private Sub Command1_Click()
Dim x As Integer
x = InputBox(" Enter the degree", "Degree between 0-100")
Select Case x
    Case 90 To 100
       MsgBox "Excellent", vbOKOnly, "Result"
    Case 80 To 89
       MsgBox "Very Good", vbOKOnly, "Result"
```

```
    Case 70 To 79
        MsgBox "Good", vbOKOnly, "Result"
    Case 60 To 69
        MsgBox "Accept", vbOKOnly, "Result"
    Case 50 To 59
        MsgBox "Bad", vbOKOnly, "Result"
    Case Else
        MsgBox "Fail", vbOKOnly, "Result"
    End Select
End Sub
```

تقوم جُملة Select Case بفحص تعبير واحد فقط (test expression) والموجود في رأس التعليمة ، ومن ثُم مُقارنة نتيجة التعبير مع مجموعة التعابير الأخرى (أي مع كُل تعبير مُرتبط مع عبارة Case) . وعند تطابق قيمة التعبير (test expression) مع أحد التعابير (expression list) فأنها ستُنفذ كُتلة التعليمات المُرتبطة مع عبارة Case الموافقة (True) .

مُلاحظـة : اذا طابقت عدة تعابير (expression list) مع قيمة التعبير (test expression) ، فأن كُتلة التعليمات الموافقة للتعبير الأول (expression list) هي فقط ستُنفذ .

يقوم برنامج لغة فيجوال بيسك بتنفيذ التعليمات الموجودة في عبارة Case Else (والتي تكون خيارية) في حال عدم مُطابقة قيمة التعبير لأي من التعابير الموجودة ، وكما موضح ذلك في البرنامج التالي :

Example :

```
Private Sub Command1_Click()
Dim x As Integer
Dim y As Integer
x = InputBox("Enter the number")
Select Case x
Case Is < 10
   Label1.Caption = 1
```

```
Case Is < 100
   Label1.Caption = 2
Case Else
    Label1.Caption = 3
End Select
End Sub
```

1 – 17 – 5 جُمل التكرار أو الدوران : Loop Statements

تُستخدم هذه الجُمل لتكرار تنفيذ عدد من التعليمات في برنامج لغة فيجوال بيسك ولعدة مرات . ولغرض تكرار تنفيذ بعض الجُمل لعدة مرات يجب تحديد تلك التعليمات المطلوب تنفيذها وذلك من خلال وضع بداية ونهاية لها ، وهذا يُسمى بالدوران .

هُناك عدة أنواع من هذه الجُمل ، هـي :

أ- جُملة For … Next : وتقوم بتكرار تنفيذ مجموعة من التعليمات وبشكل مُحدد أي أن عدد التكرار مُحدد مُسبقاً ، ويُطلق عليها أحياناً بأسم (العداد الآتوماتيكي) وذلك لأن الزيادة لعدد الدوران تتم مُباشرةً من خلال لُغة الفيجوال بيسك ، ويتم تحديد بداية الدوران ونهايته .

الصيغة العامة لهذه الجُمل هـي :

```
For Counter Var = Start Num To End Num Step [ Step Num ]
     Statements
Next [ Counter Var]
```

Example : Find the summation of integer number (0 – 10)

```
Private Sub Command1_Click()
Dim i As Integer      ' i is a counter
Dim s As Integer     ' s is a summation of i
s = 0
For i = 1 To 10
s = s + i
Next i
```

```
Print "sum="; s
End Sub
```

مُلاحظة : الأيعاز Print يستخدم لعرض قيم معينة على النموذج أثناء التنفيذ ، ويمكن اختصاره بالرمز (علامة الأستفهام " ? " من لوحة المفاتيح) حيث بعد الضغط على مفتاح Enter للأنتقال الى سطر برمجي جديد ستقلب علامة الأستفهام تلقائياً الى كلمة Print . وهذه تعتبر واحدة من ميزات وأسرار برنامج لغة فيجوال بيسك (أي السهولة في تمثيل الأيعازات) .

- يُمكن أن تكون صيغة العداد (Counter Var) كالآتي : For I = 2 to 10 Step 2 ، أي قراءة الأعداد الزوجية المحصورة من (2 الى 10) . أي أن القيمة الأبتدائية للعداد (I) أكبر من واحد .

- يُمكن أن تكون جملة الدوران معكوسة ، وكما موضح بالصيغة التالية :

For I = 10 to 2 Step -2 ، أي تكون قيمة الخطوة Step سالبة ولكن بشرط أن تكون القيمة الأبتدائية أكبر من القيمة النهائية للعداد. يتم تنفيذ الأيعازات الموجودة داخل حلقة التكرار جميعها حتى نهاية الدوران ومن ثم الخروج من حلقة التكرار لتنفيذ باقي ايعازات (خطوات) البرنامج . كذلك من المُمكن أن يحتوي البرنامج على مجموعة من حلقات التكرار المُتداخلة (Nested Loop) على شرط أن لاتكون مُتقاطعة . وأيظاً لايجوز استخدام نفس العداد لحلقتي تكرار مُتداخلة .

- يُمكن وضع بُنية تحكم داخل بُنية تحكم أُخرى وهذا مايُطلق عليه ببُنى التحكم المُهيكلة (Nested Control Structures) . فمثلاً نضع البُنية If … Then داخل البُنية For … Next ، وكما موضح ذلك في المثال التالي :

Example : Find the summation of the even number (1 – 10)

```
Private Sub Command1_Click()
Dim i As Integer, s As Integer
For i = 1 To 10
If i Mod 2 = 0 Then
s = s + i
End If
Next i
```

```
Print "sum="; s
End Sub
```

ب- جُملة Do … While : لايُحدد فيها عدد الدوران وانما تعتمد على شرط مُعين ، حيث يتم فحص الشرط فأذا كانت قيمته صحيحة يتم تنفيذ جُمل الدوران ، أما اذا كانت قيمة الشرط خاطئة فسيتم أنهاء جُمل الدوران مُباشرةً . الصيغة العامة لهذا الشرط كالآتي :

```
Do While < Condition >
    Statements
Loop
```

Example : Find the summation of integer number (1 – 10)

```
Private Sub Command1_Click()
Dim i As Integer
Dim s As Integer
i = 1
Do While i <= 10
  s = s + i
  i = i + 1
Loop
Print "sum="; s
End Sub
```

ت- جُملة Do … Until : أيظاً لايُحدد فيها عدد الدوران وأنما تعتمد على شرط مُعين ، حيث يتم فحص الشرط فاذا كانت قيمتهُ خاطئة يتم تنفيذ جُمل الدوران ، أما اذا كانت قيمة الشرط صحيحة فسيتم أنهاء جُمل الدوران مُباشرةً (عكس جُلة Do … While) . أي بمعنى آخر يستمر الدوران طالما كان الشرط خاطأ . الصيغة العامة لهذه الجُملة كالأتي :

```
Do Until < Condition >
    Statements
Loop
```

Example : Find the summation of integer number (1 – 10)

```
Private Sub Command1_Click()
Dim i As Integer
Dim s As Integer
i = 1
Do Until i > 10
  s = s + i
  i = i + 1
Loop
Print "sum="; s
End Sub
```

1 – 18 العمليات التي تُطبق على النماذج : Methods applying on Form

هُناك عدد من العمليات التي يُمكن اجراءها على النموذج أثناء تنفيذ المشروع ، ومنهـا كما موضحة بالجدول رقم (1 – 7) .

جدول رقم (1 – 7) : العمليات الرئيسية التي تُطبق على النماذج .

Method	Description
Cls مسح الشاشة	Clears all graphics and text from form . Does not clear any object
Print طباعة النتائج	Print text string on the form
Hide اخفاء النموذج الحالي	Hide the form
Show اظهار النموذج	Show the form

Example **:**

```
Private Sub cmdprint_Click()
Form1.Print "ali"
End Sub
```

.....

```
Private Sub cmdclear_Click()
Form1.Cls
End Sub
```

.....

```
Private Sub cmdshow_Click()
Form2.Show
End Sub
```

.....

```
Private Sub cmdhide_Click()
Form2.Hide
End Sub
```

19 – 1 صندوق الرسائل وصندوق المُدخلات : ( )MsgBox( ) & InputBox

تُعتبر صندوق الرسائل وصندوق المُدخلات واحدة من أهم دوال لُغة فيجوال بيسك ، ويُطلق عليها بأسم الدوال الداخلية (Internal Functions) .

The main purpose of the functions is to accept certain input and pass them on to the main program to finish the execution .

The general format of a functions is : الصيغة العامة لهذه الدوال هـي

Function Name (arguments) .

Where arguments are values that are passed on to the functions .

يتُم استخدام دالتي صندوق الرسائل وصندوق المُدخلات عندما يُراد من البرنامج أن يقوم بسؤال المُستخدم User عن أسئلة معينة أثناء تنفيذ البرنامج ، أو لأظهار رسائل خاطئة ، أو لتقديم نصيحة مُعينة للمُستخدم .

صندوق الرسائل : هو عبارة عن صندوق حوار يُستخدم لعرض معلومات مُعينة للمُستخدم .

A message box : is a dialog box you display to give the user information.

صندوق المُدخلات : هو عبارة عن صندوق حوار يُستخدم لسؤال المُستخدم عن معلومات مُعينة .

An input box : is a dialog box you display to ask the user questions .

1 – 19 – 1 دالة صندوق الرسائل : MsgBox( ) Function

تعتبر هذه الدالة نوعاً ما بمثابة ايعاز الطباعة Print ، حيث تستخدم لعرض ناتج تنفيذ خطوات البرنامج سواءً أكان البرنامج يعمل بشكل صحيح أم خاطئ من خلال اظهار رسالة معينة تشير الى حالة البرنامج . هُناك صيغتان رئيسيتان لكتابة (لتنفيذ صندوق الرسائل) ، وهي كالآتي :-

أ- MsgBox Return Value : حيث يتم أستخدامُها داخل مُتغير لتنفيذ أو لأظهار قيمة مُعينة ، والصيغة العامة لها هـي :

```
Your Msg = MsgBox ( Prompt , Style Value , Title )
                    OR
X = MsgBox ( argument )
```

أسم المُتغير دالة الـفيجوال بيسك

- The first argument(Prompt): Will display the message in the message box.
- The second argument (Style Value) : Is an optional numeric value or constant name that will determine what type of command button appear on the message box and any icon to show .
- The third argument (Title) : Is an optional string that represents the text in the message box's title bar .

الجدول رقم (1 – 8) يُظهر أزار التحكم (Command button) المُستخدمة مع صندوق الرسائل ، حيث يُمكن الأستعانة بقيم رقمية (Integer) بدلاً من أسماء ثوابت .

جدول رقم (1 – 8) : أزرار التحكم المُستخدمة مع صندوق الرسائل وأرقامها .

Style Value	Named Constant	Buttons Display
0	VbOkOnly	Ok button
1	VbOkCancel	Ok and Cancel buttons
2	VbAbortRetryIgnore	Abort , Retry and Ignore buttons
3	VbYesNoCancel	Yes , No and Cancel buttons
4	VbYesNo	Yes and No buttons
5	VbRetryCancel	Retry and Cancel buttons

Example :

```
Private Sub Command1_Click()
x = MsgBox("hello", 0, "information")
End Sub
OR
Private Sub Command2_Click()
x = MsgBox("hello", vbOKOnly, "Information")
End Sub
```

مُلاحظة : لجعل صندوق الرسائل Msg Box يبدو أكثر وضوحاً ، يُمكن اضافة صورة (Icon) الى جانب الرسالة بحيث تُشير هذه الصورة الى وظيفة أو مُهمة الرسالة الواجب تنفيذُها .

لُغة الفيجوال بيسك تُجهزُنا بأربعة أنواع من الصور وكما موضحة في الجدول رقم (1 – 9) .

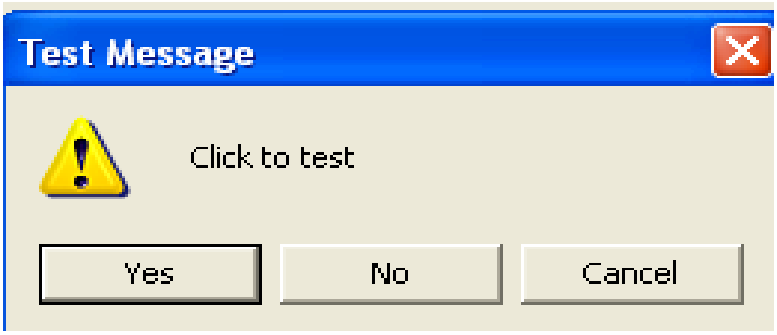
جدول رقم (1 – 9) : أزرار التحكم المُستخدمة مع صندوق الرسائل والصور الخاصة بها .

Value	Named Constant	Icon
16	VbCritical	
32	VbQuestion	
48	VbExclamation	
64	VbInformation	

Example :

```
Private Sub Command1_Click()
x = MsgBox("Click to test", vbYesNoCancel + vbExclamation, "Test Message")
End Sub
OR
Private Sub Command2_Click()
x = MsgBox("Click to test", vbYesNoCancel + 48, "Test Message")
End Sub
OR
Private Sub Command2_Click()
x = MsgBox("Click to test", 3 + 48, "Test Message")
End Sub
```

عند التنفيذ سيظهر لنا البرنامج بالشكل رقم (1 – 18) .



شكل رقم (1 – 18) : برمجة دالة صندوق الرسائل

أن المُتغير (X) يحمل قيمة مُعينة يُمكن أسترجاعها من دالة صندوق الرسائل Msgbox( ) ، وهذه القيمة تُحدد من خلال نوع الأمر (button) المُستخدم والذي يتم نقره بواسطة المُستخدم . ويتم تعريف هذا المُتغير كرقم صحيح Integer داخل البرنامج الفرعي Procedure أو في خانة التعاريف العامة (General Declaration) .

مُلاحظة: يُمكن برمجة أزرار الأمر (Button Clicked) التي تظهر لنا أثناء أستخدام دالة صندوق الرسائل من خلال أعطائها قيم تعريفية (Value) أو من خلال أستخدام أسم الأمر نفسه (Named Constant) وكما موضح ذلك في المثال التالي :-

Example : Use the two control box (one Command button & one Label button) on the form (Form1) , and set the properties as follows :

Command1: **Name : Cmd1** **Caption : Test** **Font : Bold - 14**	**Label1 :** **Name : Lbl1** **Caption :** **Appearance : 0-flat** **Alignment : 2-Center** **ForeColor : System – Active Title Bar**

```
Private Sub Cmd1_Click()
Dim x As Integer
x = MsgBox("Click to test", 1, "Test Message")
```

```
If x = 1 Then
Lbl1.Caption = "Testing Successful"
Else
Lbl1.Caption = "Testing Fail"
End If
End Sub
```

عندما يقوم الـ User بالنقر على زر الأمر Command button والذي يحمل العنوان Test فستظهرلنا شاشة البرنامج بالشكل رقم (1 – 19) والتي تحتوي على خيارين ، الخيار الأول (Ok) والذي عند النقر عليه ستظهر الرسالة (Testing Successful) ، وعند النقر على الخيار الثاني (Cancel) ستظهر الرسالة (Testing Fail) .

شكل رقم (1 – 19) : برمجة أزرار تحكم دالة صندوق الرسائل

الجدول رقم (1 – 10) يُبين أزرار الأمر (Button Clicked) التي تظهر لنا أثناء أستخدام دالة صندوق الرسائل مع أسم الأمر نفسه (Named Constant) .

جدول رقم (1 – 10) : أزرار الأمر الخاصة بدالة صندوق الرسائل مع أسم الأمر .

Value	Named Constant	Button Clicked
1	VbOk	Ok button
2	VbCancel	Cancel button
3	VbAbort	Abort button
4	VbRetry	Retry button
5	VbIgnore	Ignore button

6	VbYes	Yes button
7	VbNo	No button

كذلك يُمكن اضافة صورة (Icon) الى المثال أعلاه تظهر عند تنفيذ البرنامج كما مبين ذلك في الشكل رقم (1 – 20) ، وذلك من خلال اضافة الشفرة (Code) التالي الى البرنامج ليكون كالآتي :

```
Private Sub Cmd1_Click()
Dim x As Integer
x = MsgBox("Click to test", vbYesNoCancel + vbExclamation, "Test Message")
If x = 6 Then
Lbl1.Caption = "Testing Successful"
ElseIf x = 7 Then  ' Or if x = vbNo
Lbl1.Caption = "Testing Fail"
Else
Lbl1.Caption = "Try again"
End If
End Sub
```

شكل رقم (1 – 20) : برمجة أزرار تحكم دالة صندوق الرسائل مع أضافة صورة

ب- MsgBox Return No Value : في هذا النوع يتُم كتابة المُتغير داخل صندوق الرسائل ، حيث يتم تنفيذ MsgBox بشكل مُباشر دون الحاجة لوضعها داخل مُتغير كما هو الحال في النوع السابق . والصيغة العامة تكون كالآتي :

```
MsgBox  Message , Type , Title
```

Where :

Message : Is a text message to be displayed .

Type : Determine what type of command buttons appear on the message box and any icon to show .

Title : Is a text in title bar of message box .

مُلاحظة : مُمكن أن يكون حقل الـ Type فارغ ، وذلك حسب مامطلوب تنفيذه داخل صندوق الرسائل وكما موضح في المثال التالي :

Example :

```
Private Sub Form_Activate()
For i = 1 To 10
Form1.Caption = "X =" & i
x = i ^ 2
MsgBox "The value of X =" & x, , "The square Values of variable"
Next
Form1.Caption = "Thank You"
End Sub
```

1 – 19 – 2 دالة صندوق المدخلات : Input Box ( ) Function

تقوم هذه الدالة بعرض رسالة (Message box) للمُستخدم تسمح له بأدخال قيمة مُعينة (رقمية أو نصية) . والصيغة العامة لهذه الدالة كالأتي :

N = InputBox (Prompt , Title , default_text , X-position , Y-position)

- Prompt : The message displayed normally as a question asked .

- Title : The title of the input box .

- default_text : It’s a appears in the input field where users can use it as his intended input .

- X-position and Y-position : The position or the coordinate of the input box .

Example1 : Draw two Labels and one Command buttons on the form and then set the following properties as follows :

Label1 :
Caption : Your Message

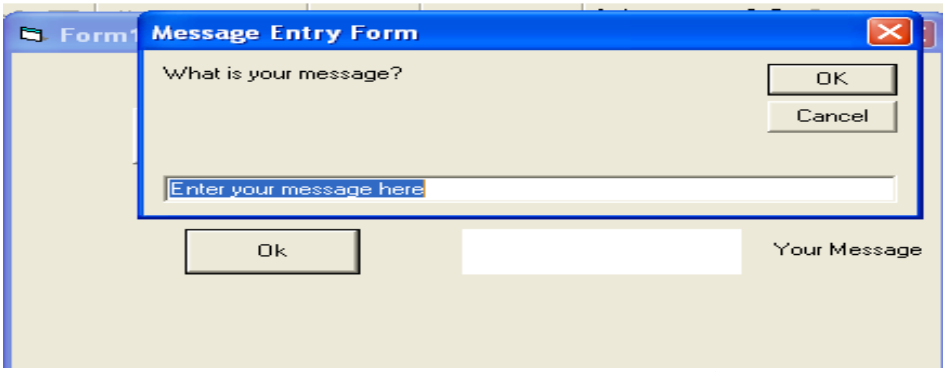
Label2 :
Caption :
Appearance : 0-flat

Command1 :
Name : cmdok
Caption : Ok

Then the procedure for the Command1 , is :

```
Private Sub cmdok_Click()
Dim n As String
n = InputBox("What is your message?", "Message Entry Form", "Enter your message here", 500, 50)
If n <> "" Then
Label2.Caption = n
Else
Label2.Caption = "No Message"
End If
End Sub
```

عند التنفيذ سيظهر لنا البرنامج بالشكل رقم (1 – 21).



شكل رقم (1 – 21) : برمجة دالة صندوق المدخلات

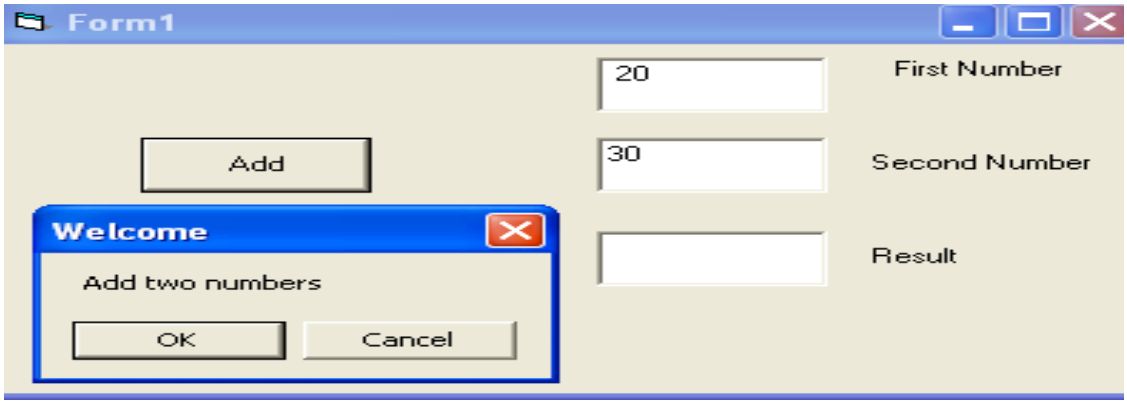
Example2: Use the object bellow to add two numbers :

TextBox1 : **Name : txt1** **Text :**	**TextBox2 :** **Name : txt2** **Text :**	**TextBox3 :** **Name : txtresult** **Text :**

Label1 : **Caption : First Number**	**Label2 :** **Caption : Second Number**
Label3 : **Caption : Result**	**Command1 :** **Name : cmdadd** **Caption : Add**

```
Private Sub cndadd_Click()
Dim n As Integer
n = MsgBox("Add two numbers", vbOKCancel, "Wellcome")
If n = vbOK Then
txtresult.Text = Val(txt1.Text) + Val(txt2.Text)
Else
txtresult.Text = ""
End If
End Sub
```

عند التنفيذ سيظهر لنا البرنامج بالشكل رقم (1 – 22) .



شكل رقم (1 – 22) : برمجة أزرار دالة صندوق المدخلات

نستنتج مما تقدم أعلاه أن صندوق الرسائل Msgbox يُستخدم لأخراج أو لعرض النتائج ، بينما صندوق المُدخلات Inputbox فيستخدم لأدخال البيانات .

1 – 20 تكوين تطبيقات لُغة فيجوال بيسك : Creating an application

هُناك أربع خطوات رئيسية تُستخدم لتكوين تطبيقات أو برامج لُغة فيجوال بيسك وهي:

1. تصميم واجهة البرنامج .

Designing the interface for a new program .

2. ضبط خصائص الكائنات المُستخدمة في واجهة البرنامج .

Setting the properties of objects in the interface .

3. كتابة شفرة البرنامج .

Writing the program code .

4. حُفظ وتنفيذ البرنامج

Save and run the program (project) .

مثال 1 : سيتم عمل حاسبة بسيطة وكما موضحة بالشكل رقم (1 – 23) ، مانحتاج اليه هو :

1. Creating the interface : سنحتاج هنا ادراج (رسم) ثلاثة صناديق نص Text Box الى النموذج ، وأربعة أزرار أمر Command Button مع ثلاثة صناديق عنوان Label Box، حيث سيتُم اجراء أربع عمليات حسابية وهي (الجمع والطرح والضرب والقسمة) لاحظ جدول رقم (1 – 11) .

2. Setting the properties : هُنا سيتم أستخدام وتحديد الخصائص الضرورية التي يتطلبُها المشروع للكائنات المُدرجة على النموذج .

جدول رقم (1 – 11) : عناصر التحكم والخصائص المستخدمة فيها .

Object	Property	Setting
Label1	Caption Font	First Number Bold – 12 Size
Label2	Caption Font	Second Number Bold – 12 Size

Label3	Caption Font	Result Bold – 12 Size
Text1	Name Text	txtinput1
Text2	Name Text	txtinput2
Text3	Name Text	txtresult
Command1	Name Caption	cmdadd Addition
Command2	Name Caption	cmdsub Subtraction
Command3	Name Caption	cmddiv Division
Command4	Name Caption	cmdmultiplay Multiplication

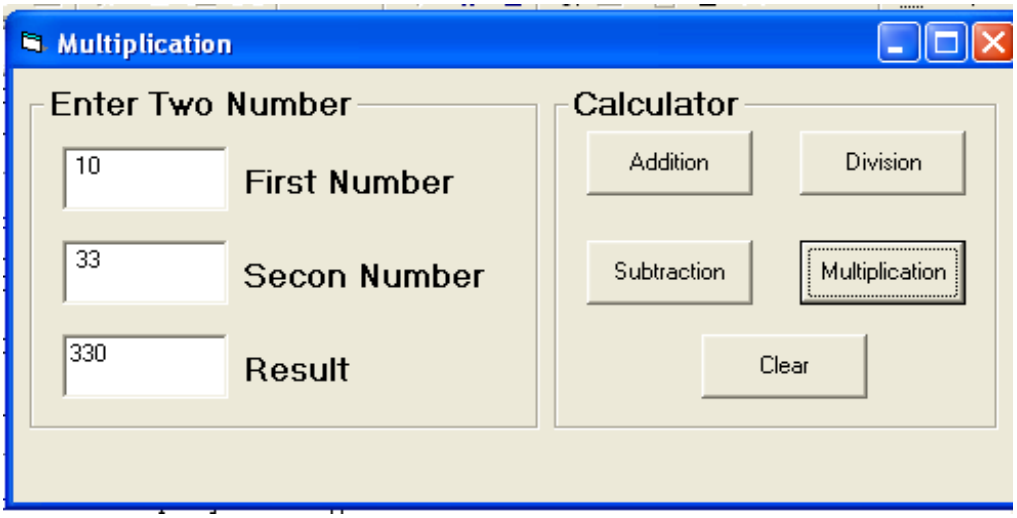
3. Writing the program code : عندما يقوم المُستخدم بالنقر على زر الأمر سيتم تنفيذ البرنامج من خلال الشفرة المكتوبة داخل الـ Command والتي هي طبقاً لهذا المشروع ستكون كالأتي :

```
Private Sub cmdadd_Click()
txtresult.Text = Val(txtinput1.Text) + Val(txtinput2.Text)
Form1.Caption = "Addition"
End Sub
```

```
Private Sub cmddiv_Click()
If Val(txtinput2.Text) = 0 Then ' هنا يجب اختبار أن المقسوم عليه لايساوي صُفر
txtresult.Text = "Error"
Else
txtresult.Text = Val(txtinput1.Text) / Val(txtinput2.Text)
End If
Form1.Caption = "Division"
End Sub
Private Sub cmdmultiplay_Click()
txtresult.Text = Val(txtinput1.Text) * Val(txtinput2.Text)
Form1.Caption = "Multiplication"
End Sub
Private Sub cmdsub_Click()
txtresult.Text = Val(txtinput1.Text) - Val(txtinput2.Text)
Form1.Caption = "Subtraction"
End Sub
```

يُمكن مسح محتويات الكائن Textbox من خلال كتابة الشفرة التالية على زر الأمر (Clear) .

```
Private Sub cmdclear_Click()
txtinput1.Text = ""
txtinput2.Text = ""
txtresult.Text = ""
Form1.Caption = "Mathematics"
End Sub
```



شكل رقم (1 – 23) : عمل حاسبة بسيطة

مثال2 : صمم نافذة مشروع يتم من خلالها :

أ- حساب مضروب أي عدد (Factorial) ، كما موضح بالشكل رقم (1 – 24) .

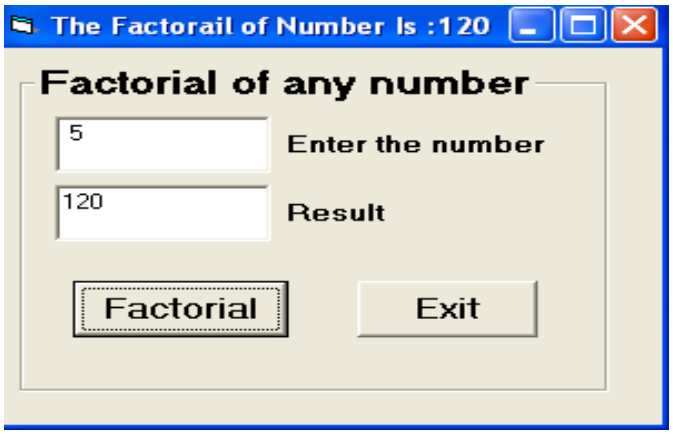
ب- اختبارعدداً ما آولي آم غير آولي .(Prime Number) ، كما موضح بالشكل رقم (1 – 25) .

- أن مضروب العدد N هو حاصل ضرب الأعداد من 1 الى العدد N ، وكالآتي :

N! = 1*2*3*…*N

```
Private Sub cmdfac_Click()
Dim n As Integer, I As Integer
Dim f As Double
n = Val(Text1.Text)
f = 1
For I = 1 To n
   f = f * I
Next I
Text2.Text = f
Form1.Caption = "The Factorial of Number Is :" & Val(Text2.Text)
End Sub
```

```
Private Sub cmdexit_Click()
End
End Sub
Private Sub Form_Load()
Form1.Caption = "Welcome"
End Sub
```



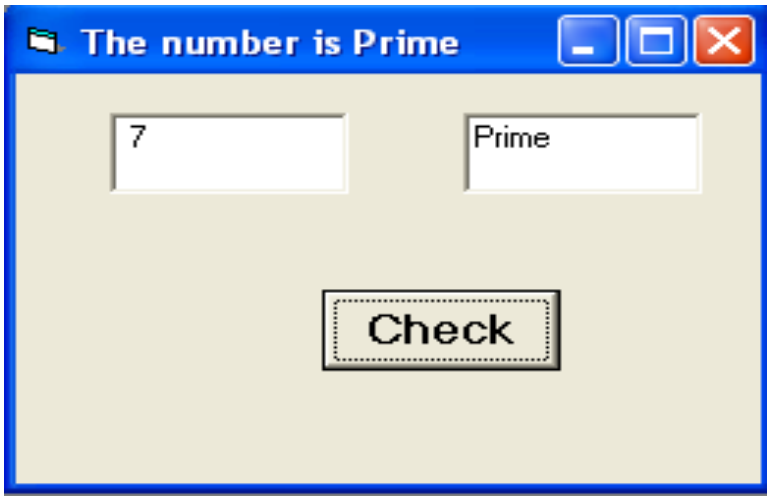
شكل رقم (1 – 24) : حساب مضروب أي عدد

- العدد الأولي هو العدد الذي يقبل القسمة على نفسه وعلى الواحد فقط . ولأختبار أن العدد N آولي نُقسمهُ على كُل الأعداد من 2 الى N-1 :

1. اذا كان هُناك باقي لقسمة العدد N على جميع الأعداد من 2 الى N-1 ، يكون اذن العدد N آولياً
2. آما اذا قبل العدد N القسمة على أحد الأعداد من 2 الى N-1 ، فيكون اذن العدد N غير آولي .

```
Private Sub Command1_Click()
Dim n As Integer
n = Val(Text1.Text)
Dim i As Integer
Text2.Text = "Prime "
For i = 2 To n - 1
If n Mod i = 0 Then
   Text2.Text = "Not Prime"
Exit For
End If
Next i
```

```
Form1.Caption = "The number is " & Text2.Text
End Sub
```



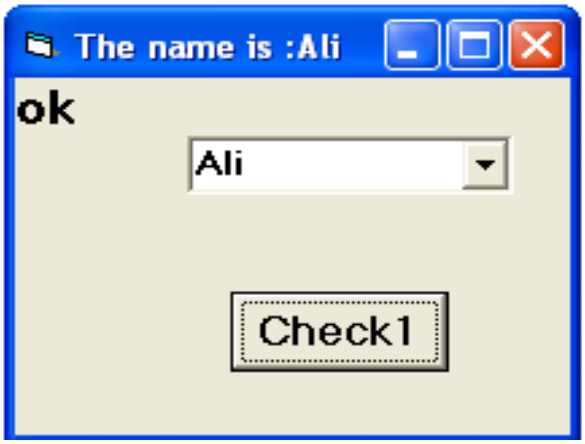
شكل رقم (1 – 25) : اختبار عدداً ما أولي آم غير أولي

1 – 21 دراسة بعض الخصائص المُهمة : Studying the object's properties

الكائن Combo Box يحتوي على خاصية مُهمة تُدعى بالقائمة List ، والتي من خلالها يتم ملئ القائمة . ويتم ملئ قائمة الكائن من خلال اضافة العناصر (أو القيم) الى القائمة وبشكل مُباشر من نافذة الخصائص وضغط مفتاح الأدخال Enter من لوحة المفاتيح . وفي حالة ادخال أكثر من عُنصر الى القائمة ، فيتُم من خلال اضافة القيمة الأولى ومن ثُم الضغط على مفتاحي (Ctrl + Enter) من لوحة المفاتيح حيث سيسمح لنا بأدخال أكثر من قيمة داخل قائمة الكائن ضمن خاصية List .

ولغرض استدعاء القيم أو العناصر الموجودة ضمن قائمة الكائن Combo برمجياً ، فيتَم ذلك من خلال كتابة الشفرة أدناه والموضحة عند التنفيذ في الشكل رقم (1 – 26) .

```
Private Sub Command1_Click()
If Combo1.Text = "Ali" Then
Print "ok"
End If
Form1.Caption = "The name is :" & Combo1.Text
End Sub
```



شكل رقم (1 – 26) : استدعاء القيم أو العناصر ضمن قائمة الكائن Combo برمجياً

كذلك يُمكن اختبار الخاصية (List Index) لصندوق القائمة المُنسدلة Combo Box عوضاً عن الخاصية (Text) ، حيث تأخذ هذه الخاصية القيمة (0) للعُنصر الأول في القائمة ، والقيمة (1) للعُنصر الثاني وهكذا .... الخ . وبذلك يتم تطبيق العبارة التالية :

```
Private Sub Command1_Click()
If Combo1.ListIndex = 0 Then
Print "ok"
End If
Form1.Caption = "The name is :" & Combo1.Text
End Sub
```

الخاصيةBorder Style الموجودة في النموذج Form تُستخدم لتحديد شكل اطار النموذج ، والقيمة الأفتراضية لهذه الخاصية هي (2-Sizable) ، أي أنه يُمكن للمُستخدم أن يقوم بتغيير أبعاد النموذج . أما اذا وضعنا القيمة (3-FixedDialog) فسيصبح النموذج غير قابل للتمدُد . أما الخاصية Min Button تُستخدم لألغاء أو للأبقاء على زر تصغير النموذج .

يحتوي الكائن Text Box على خاصية مُهمة جداً تُدعى (Password Char) ، والتي تُستخدم لجعل حرفاً (أو علامةً) ما يظهر عوضاً عن الأحرُف المُدخلة (عادةً نضع العلامة *) . وتُستخدم هذه الخاصية عادةً في حالة كون البرنامج يحتوي على كلمة سر .

الكائن Option Button يحتوي على خاصية مُهمة جداً وهي (Value) ، فاذا كان زر الخيار مًحدد فتأخذ الـ Value قيمة True ، واذا كان غير مُحدد فتأخذ القيمة False ، لاحظ الشكل رقم (1 – 27) . كذلك يُمكن أستخدام العبارة Me للدلالة على النموذج الحالي ، وكما موضح ذلك في المثال التالي :

```
Private Sub Command1_Click()
If Option1.Value = True Then
Label1.Caption = "Welcome"
Else
If Option2.Value = True Then
Label1.Caption = "Hi"
End If
End If
Me.Caption = "Testing is : " & Label1.Caption
End Sub
Private Sub Form_Load()
Me.Caption = "Testing is : "
End Sub
```

شكل رقم (1 – 27) : برمجة الكائن Option Button

الكائن Check Box يحتوي على خاصية مُهمة جداً وهي (Value) ، وهذه الخاصية تُتيح لعنصر التحكم أن يأخُذ احدى القيم الثلاث التالية وكما موضح ذلك بالجدول رقم (1 – 12) .

جدول رقم (1 – 12) : القيم العددية المُتاحة للكائن صندوق الفحص والأختبار.

Value	Description
0-Unchecked	غير مُحدد
1-Checked	مُحدد
2-Grayed	رمادي

Example :

```
Private Sub cmdtest_Click()
If Check1.Value = 1 Then ‘ ( OR ) If Check1.Value = vbChecked Then
Print "True"
Else
Print "false"
End If
End Sub
```

يُستخدم عُنصر التحكم Shape لرسم شكل هندسي على النموذج ، حيث يحتوي هذا الكائن على خاصية مُهمة وهي المظهر Shape والتي تُحدد الشكل الهندسي الناتج وكما موضح بالجدول رقم (1 – 13) .

جدول رقم (1 – 13) : القيم العددية المُتاحة لكائن رسم الشكل الهندسي .

Value	Description
0-Rectangle	مستطيل
1-Square	مربع
2-Oval	قطع ناقص
3-Circle	دائرة
4-Round Rectangle	مستطيل ذو حواف مُستديرة
5-Round Square	مربع ذو حواف مُستديرة

```
Private Sub cmddraw_Click()
Shape1.Shape = 3
End Sub
```

2 – 1 المقدمة : Introduction

زادت أهمية المعلومات في وقتنا الحاضر من أهمية نُظم قواعد البيانات لتنظيم المعلومات التي نحتاجُها وتوفيرُها للأستخدام بالطريقة المُناسبة وفي الوقت المُناسب . هكذا تطور علم أدارة نُظم قواعد البيانات من موضوع فرعي في تطبيقات الحاسوب الى موضوع أساسي ومكون رئيسي من علم الحاسوب الحديث ، وبذلك فقد أصبحت نُظم قواعد البيانات جُزءاً رئيسياً في منهاج علم الحاسوب .

2 – 2 ماهي قاعدة البيانات : What is a database

قاعدة البيانات عبارة عن مجموعة من البيانات أو المعلومات المُتصلة وذات العلاقة المُتبادلة فيما بينها والمخزونة بطريقة نموذجية أي من دون تكرار غير مُبرر . وأهم مايُميزُها أنها تُخزن بطريقة تُحقق نوعاً من الأستقلالية والمناعة ضد التغيير من البرامج التي تقوم بأستخدامُها حاضراً أو مُستقبلاً . حيث يتم انشاء قواعد البيانات وخزنها مع امكانية التعامل معها وأجراء كافة العمليات المطلوبة عليها من عمليات حذف ، أضافة ، تحديث ، ... الخ وذلك بأستخدام نظام ادارة قواعد البيانات (Database Management Systems (DBMS . والذي هو عبارة عن برنامج أو مجموعة برامج والتي من خلالها يُمكن الوصول الى قواعد البيانات ومن ثم اجراء كافة العمليات المطلوبة عليها .

بشكل عام يتم تمثيل البيانات داخل قاعدة البيانات على هيئة نماذج مختلفة ، كأن تكون نماذج علائقية relational model أو شبكية network model أو هرمية hierarchical model (منظمة في مراتب مُتسلسلة) أو على شكل رسوم على الحاسبة ممثلة بمعادلات رياضية object-oriented data model.

هذه النماذج تُعرف الطريقة التي يتم من خلالها تمثيل البيانات داخل قواعد البيانات ، ويعتبر نظام أدارة قواعد البيانات واحد من أهم هذه النماذج .

برنامج مايكروسوفت أكسس MS-Access 2003 يعتمد على النموذج العلائقي في تمثيله للبيانات ضمن قاعدة البيانات لذلك فأن النظام المُستخدم فيه يُسمى بنظام أدارة قواعد البيانات العلائقية Relational DBMS أو RDBMS .

2 – 3 مقدمة لبرنامج مايكروسوفت أكسس : An introduction to Ms-Access

مايكروسوفت أكسس (Ms-Access) هو عبارة عن برنامج يُستخدم لحفظ البيانات والمعلومات الخاصة بالشركات في صيغة قواعد بيانات مُكونة من مجموعة من الجداول المُنظمة بصورة مُعينة وعرضها بطريقة تُسهل الأستفادة من تلك المعلومات وتخزين بيانات جديدة والأستعلام عن هذه البيانات وطباعتها .

Microsoft Access lets you collect, organize, find, display and print information about your personal and business life.

كذلك يُمكن تعريف برنامج مايكروسوفت أكسس على أنه أحد تطبيقات الحاسبة والتي تُستخدم لخلق (تكوين) وأدارة قواعد البيانات الموجودة في حاسبة واحدة أو في عدة حواسيب مُرتبطة مع بعضها البعض عن طريق الشبكة ، حيث يُستخدم هذا التطبيق في ادارة المعلومات الشخصية (Personal information management (PIM) (مثل المعلومات التجارية) من حيث تنظيم وأدارة كافة البيانات الداخلة والمُكونة لقاعدة البيانات .

يُمكن بأستخدام Ms-Access ادارة كافة المعلومات من ملف قاعدة بيانات واحد ، حيث نستطيع في هذا الملف أن نقوم بتقسيم البيانات الى حاويات تخزين مُنفصلة (Tables) ، ثُم عرض بيانات الجداول ، اضافتُها ، وتحديثُها بأستخدام النماذج الفورية (Forms) ، ثم البحث عن البيانات التي نريدها فقط ، وأستردادها بأستخدام الأستعلامات (Queries) ، كما نستطيع تحليل البيانات أو طباعتُها بتخطيط مُعين بأستخدام التقارير (Reports) .

مُلاحظة : سُمي الأكسس بقاعدة البيانات لأن كلمة قاعدة تعني ترتيب البيانات وفق الموضوع ، وهذا الترتيب يكون على شكل أعمدة وصفوف (جدول) .

2 – 4 مُصطلحات خاصة بقواعد البيانات :

- الحقل Field : هو المكان المُخصص في الجدول لحفظ بيانات مُعينة .

Fields : Are places in a table where you store information .

- السجل Record : مجموعة الحقول (جميع المعلومات المخزونة في صف واحد) .

A record : Is all the fields contained in one row of an access datasheet .

- الملف File : جميع السجلات لقاعدة البيانات تُخزن في الملف .

File : Is all the records of a database .

- المُجلد Folder : تُخزن الملفات المُختلفة في مُجلدات مُنفصلة .

Folder : Different files are stored in separate folders .

2 – 5 تشغيل برنامج مايكروسوفت أكسس : Starting MS-Access

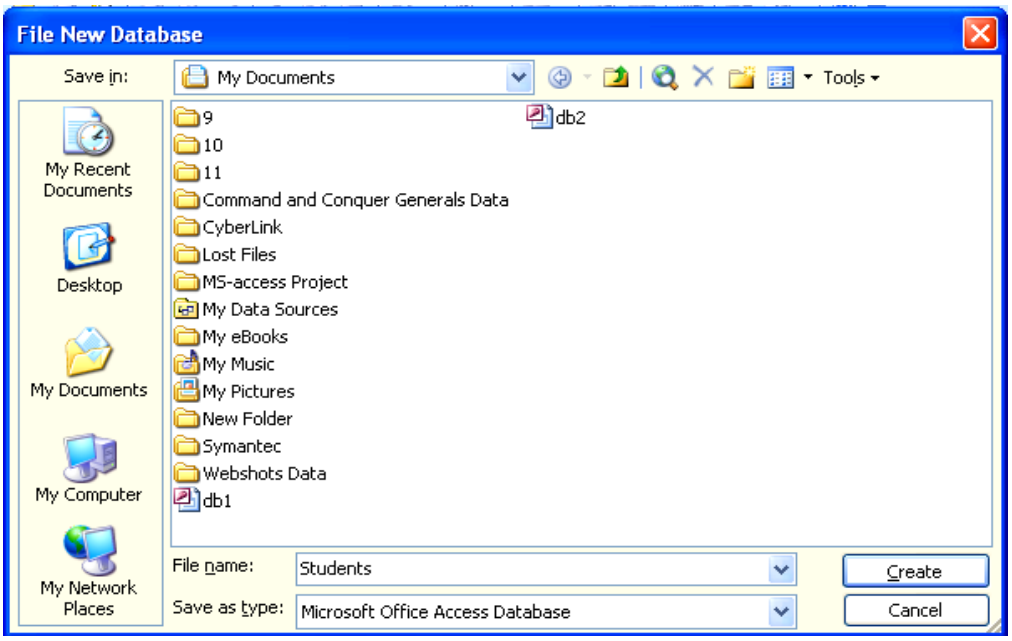
لتشغيل برنامج مايكروسوفت أكسس Ms-Access 2003 يتم النقر فوق الأمر برامج Programs من قائمة البدأ Start ، ثم أختيار برنامج Microsoft Office Access 2003 الموجود مع مجموعة تطبيقات برامج Microsoft Office .

بعد تشغيل البرنامج نقوم بتكوين قاعدة بيانات جديدة أو فتح قاعدة بيانات موجودة سابقاً وذلك من خلال تنفيذ الأمر التالي ، وكما موضح ذلك بالشكل رقم (2 – 1) .

Task pane ⟶ create a new file ⟶ Blank database

شكل رقم (2 – 1) : فتح أو تكوين قاعدة بيانات جديدة

سيتم هُنا تكوين قاعدة بيانات فارغة (جديدة) بأسم مُعين وخزنها بموقع مُعين . عند اختيار الأمر Blank database يظهر مُربع حوار والموضح بالشكل رقم (2 – 2) عنوانه ملف قاعدة بيانات جديد File New Database حيث يتم تحديد الدليل (الموقع) الذي نُريد حفظ ملف قاعدة البيانات فيه وذلك في مربع حُفظ في Save In ، بعدها يتم تحديد أسم لملف قاعدة البيانات في مُربع أسم الملف File Name ، ثم أختيار الأمر انشاء Create فتظهر قاعدة البيانات التي تم انشائُها .



شكل رقم (2 – 2) : خزن قاعدة بيانات جديدة

مُلاحظة : تُعتبر هذه القاعدة Access 2003 من نمط Access 2000 ، ويُمكن تحويلها الى Access 97 أو Access 2002 أو Access 2003 من خلال أختيار الأمر التالي :

Tools ⟶ Database Utilities ⟶ Convert Database
⟶ To Access 2002-2003 File Format

مُلاحظة : قبل البدأ بأنشاء قاعدة بيانات جديدة لابُد من :

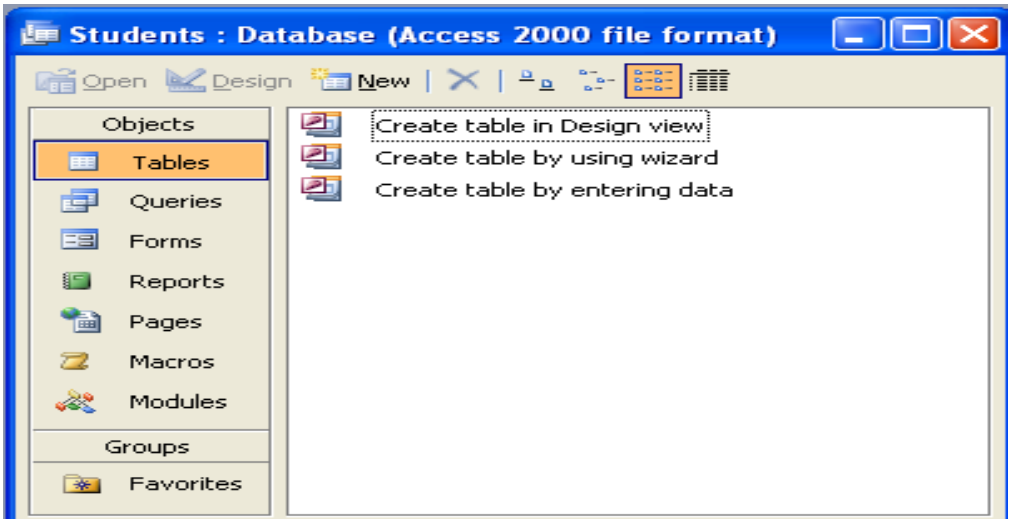
1. تحديد الغرض من قاعدة البيانات وبالتالي تحديد المعلومات التي سنحتاجُها .
2. تحديد الجداول اللازمة ، بحيث تحدد المعلومات التي نود حفظها في كل جدول .
3. تحديد الحقول اللازمة في الجدول .

2 – 6 مُكونات مايكروسوفت آكسس : Access Components

بعد تكوين قاعدة البيانات تظهر لنا نافذة قاعدة البيانات الرئيسية (Database window) ، والتي تحتوي على الكائنات (Objects) التالية والموضحة بالشكل رقم (2 – 3) .

1. الجداول (Tables) : وتُستخدم لتخزين البيانات في صفوف وأعمدة بحيث أن كل سجل يُمثل صفاً Row وكل حقل يُمثل عموداً Column ، حيث يتُم تحديد حقول بيانات القاعدة فقط من خلال الجداول .
2. الأستعلامات (Query) : تُستخدم لتجميع البيانات التي نطلُبها من الجدول (كائن الأستعلام عن البيانات وربط الجداول بمُساعدة العلاقات بين الجداول) .
3. النماذج (Forms) : تُستخدم لأدخال البيانات ومن ثم عرضها وتحليلها (عرض الواجهات والصور) .
4. التقارير (Reports) : ورقة تُستخدم لتلخيص البيانات (عرض التقارير وتنسيقها) .
5. الصفحات (Pages) : هي عناصر أو كائنات جديدة أُضيفت الى Ms-Access أبتداءاً من الأصدار 2000 ، وتعني تكوين واجهات أو صفحات آنترنت تقوم بعرض بيانات القاعدة .
6. وحدات ماكرو (Macro) : تقوم بأتمتة قاعدة البيانات الخاصة بـك (تنفيذ أكثر من عمل في وقت واحد) .
7. وحدات نمطية (Modules) : عبارة عن برامج تُساعد المُبرمج في تكوين لُغات البرمجة المُخصصة لقاعدة البيانات .

ملاحظـة : سنتطرق في هذا الفصل الى دراسة كيفية تصميم الجداول وأنواع الحقول المستخدمة فيها مع الخصائص المهمة التي تحويها ، والتي سنستخدمها لاحقاً في عملية الربط مع برنامج لغة فيجوال بيسك والموضح ذلك في الفصل الثالث .



شكل رقم (2 – 3) : نافذة قاعدة البيانات الرئيسية

من المهم معرفة أن البرمجة المُستخدمة في مايكروسوفت آكسس تُسمى بالـ (Visual Basic for Application VBA) . كذلك فأن النماذج الموجودة في الآكسس تشبه الى حد ما النماذج الموجودة في برنامج لغة فيجوال بيسك ، لكنها تحتوي على خاصية الربط المُباشر مع البيانات لأنها موجودة ضمن قاعدة البيانات .

2 – 7 فتح قاعدة بيانات موجودة : Open an existing file

لفتح قاعدة بيانات قد تم انشاءها مُسبقاً ، فيتُم من خلال الأمر التالي :

Task Pane ⟶ Open
OR :
File ⟶ Open

حيث يتُم تحديد الملف (ملف قاعدة البيانات) المُراد فتحه من قائمة الملفات ، ثم اختيار الأمر Open والوضحة بالشكل رقم (2 – 1) .

2 – 8 الجداول : (Tables)

الجدول عبارة عن تنظيم مجموعة من المعلومات في حقول حول موضوع مُعين في قواعد البيانات ، فمثلاً قد يحتوي الجدول على بيانات حول العُملاء . يُنظم الجدول البيانات في أعمدة تُدعى حقول Fields وصفوف تُدعى سجلات Records . يحتوي كل حقل على معلومة حول أحد العُملاء ويحتوي السجل على كُل المعلومات (جميع الحقول) المُتعلقة بأحد العُملاء وكما موضح ذلك بالشكل التخطيطي رقم (2 – 4).

بواسطة الجداول يُمكن اضافة بيانات ، تحريرها أو عرضها في الجدول . كذلك يُمكن اجراء تدقيق املائي للبيانات الموجودة في الجدول وطباعتها ، وفرز السجلات أو تصفيتها . ويُمكن كذلك تغيير مظهر " صفحة البيانات " أو تعديل هيكل الجدول بواسطة اضافة أعمدة أو حذفها .

Field1-Code	Field2-Name	Field3-Phone Number	Field4-Email Address
(Record)Row	صف ◄		

حقل Column (Field)

شكل رقم (2 – 4) : حقول وسجلات الجدول

2 – 9 انشاء جدول جديد : Creating New Table

هُناك ثلاث طرق لتكوين جدول جديد في ملف قاعدة البيانات (Ms-Access) ، وهـي :

الطريقة الأولى : تكوين جدول بطريقة عرض التصميم (Create table in Design view) ، حيث في هذه الطريقة يتُم انشـاء جدول كامل من لاشـئ ، أو اضافـة حقـول جـديدة ، أو حذفهـا أو اعـادة تخصيـصها وكما موضح ذلك بالشكل رقم (2 – 5) ، حيث تُعتبر هذه الطريقة من أهم الطرق وأكثرُها أستخداماً لذلك سيتم التطرق لها لاحقاً وبالتفصيل.

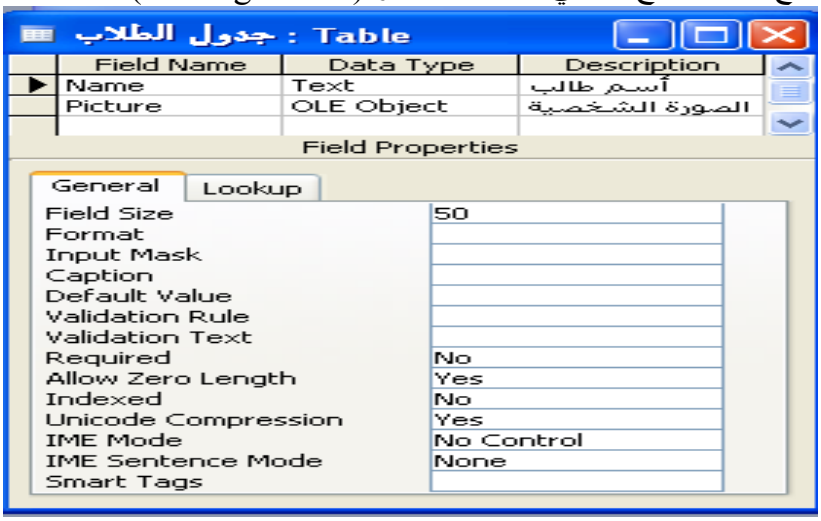
الطريقة الثانية : تكوين جدول بأستخدام المعالج (Create table by using wizard) ، حيث يوفر لنا برنامج مايكروسوفت آكسس جداول جاهزة مع حقولها يُمكن الأستعانة بها لتكوين جدول جديد .

الطريقة الثالثة : تكوين جدول جديد بأدخال البيانات (Create table by entering data) ، حيث تكون الحقول موجودة مُسبقاً بأسماء آفتراضية (.... , Field1 , Field2) لذلك نقوم فقط بأدخال البيانات تحت كل حقل مع أمكانية تغيير أسم الحقل الأفتراضي ، وتخصيصه بنوع بيانات معينة .

2 – 10 تكوين جدول بطريقة عرض التصميم : Create table in Design view

أن عملية انشاء جدول جديد في وضع التصميم تُقسم الـى :

- ادراج الحقول وتسميتُها (Field Name) .
- تحديد أنواع البيانات في حقول الجدول (Data Type) .
- تحديد خصائص للحقول المدُرجة مع بياناتها (Field Properties) .
- حُفظ الجدول مع انشاء مفتاح أساسي اذا تطلب الأمر (Saving Table) .



شكل رقم (2 – 5) : تكوين جدول بطريقة عرض التصميم

2 – 10 – 1 ادراج الحقول وتسميتُها : Field Name

أن أسم الحقل Field Name هو المكان المُخصص لأدخال أسماء معينة للحقول المُدرجة في الجدول (الأسم عادة يكون اختياري ، بحيث يُشير الى نوع البيانات التي سيتُم ادخالُها) ، ويحتوي على سعة 64 حرفاً كحد أقصى مع الفراغات . فعلى سبيل المثال يتم تخصيص أسم الحقل الأول بعنوان " Name " الذي سيضم أسماء الطلاب ضمن جدول الطلاب ، وكذلك تخصيص أسم الحقل الثاني بعنوان " Picture " والذي سيضم صور شخصية لكل طالب ضمن جدول الطلاب وكما موضح ذلك بالشكل رقم (2 – 5) .

يقوم برنامج مايكروسوفت أكسس تعريف كل حقل في الجدول من خلال أسم الحقل والذي سيُستخدم بعد ذلك في الأجراءات الفرعية للبرامج بلغة الفيجوال بيسك Visual Basic Procedures ، وكذلك في تصريحات لغة الأستعلامات المهيكلة Structured Query Language (SQL) .

ملاحظة : أسم الحقل لايُمكن أن يُكرر في الجدول ، لأن تكراره سيؤدي الى حدوث تضارب conflict أثناء تصميم قاعدة البيانات .

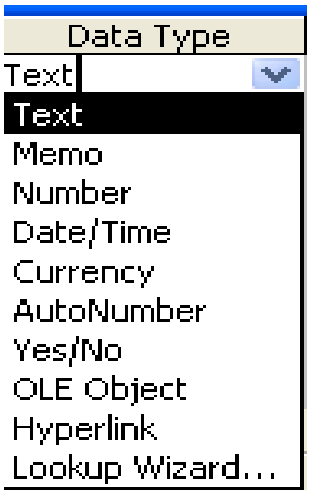
Procedure: A sequence of declarations and statements in a module that are executed as a unit. Procedures in Visual Basic include both Sub and Function procedures .

SQL string/statement: An expression that defines an SQL command, such as SELECT, UPDATES, or DELETE, and includes clauses such as WHERE and ORDER BY. SQL strings/statements are typically used in queries and in aggregate functions.

2 – 10 – 2 تحديد أنواع البيانات في حقول الجدول : Data Types

نوع البيانات Data Type هو المكان المُخصص لتحديد نوع البيانات التي سيتم ادخالُها في كل حقل ضمن الجدول ، مع مُلاحظة أن لكل حقل يُخصص نوع بيانات واحد فقط .

بعد الأنتهاء من اضافة اسم الحقل الى الجدول لابُد من تحديد نوع بيانات كُل حقل ، فعند الأنتقال الى عمود نوع البيانات يظهر سهم بجانب كُل حقل ، وعند النقر على ذلك السهم تنسلُ (تظهر) قائمة تحتوي على أنواع مُختلفة من البيانات ، وكما موضح ذلك في الشكل رقم (2 – 6) حيث أن هنالك عشرة أنواع من بيانات الحقول لكل منها خصائصه الخاصة به وكما موضح ذلك بالجدول رقم (2 – 1) .



شكل رقم (2 – 6) : نوع بيانات حقول الجدول

جدول رقم (2 – 1) : أنواع بيانات الحقول وخصائصها وحجم تخزينها .

Setting نوع الحقل	Type of data الوصف	Size حجم التخزين
Text نص	(Default) Text or combinations of text and numbers , as well as numbers that don't require calculations , such as phone numbers . It's also called Alphanumeric data .	Up to 255 characters or the length set by the field size property , whichever is less . الحد الأعلى 255 حرفاً أبجدياً ، والبرنامج يُعطي بشكل تلقائي 50 حرفاً .
Memo مُذكرة	Lengthy text or combination of text and numbers نص مفتوح مع أمكانية ادخال حروف وأرقام وعلامات .	Up to 65,535 characters

Number رقم	Numeric data used in mathematical calculations. قيم عددية (رقمية) يمكن أستخدامها في أجراء الحسابات الرياضية وحسب الحاجة . (رقم صحيح أو عشري حسب الخصائص المستخدمة) .	1 , 2 , 4 < or 8 bytes
Date / Time تأريخ / وقت	Date and time values for the years 100 through 9999	8 bytes
Currency العُملة	Currency values and numeric data used in mathematical calculations involving data with one to four decimal places (Numbers with a leading dollar sine. Use this data type for currency values such as dollars, Frances, or yens).	8 bytes Accurate to 15 digits on the left side of the decimal separator قيم نقدية تصل الى 15 خانة ، بالأضافة الى الفاصلة العُشرية (رقم + نوع العُملة المُستخدمة)
Auto Number ترقيم تلقائي	A unique sequential (incremented by 1) number or random number assigned by Microsoft Access whenever a new record is added to a table. AutoNumber fields can't be updated . في حالة حذف أي قيد (سجل) يستمر الترقيم تصاعدياً أي بدون اعادة الترقيم .	4 bytes (16 bytes if the Field Size property is set to Replication ID).
Yes / No نعم / لا	Yes and No values are fields that contain only one of two values (Yes/No, True/False, or On/Off). Logical or Boolean values .	1 bit
OLE Object كائن ربط وتضمين العناصر	An object (such as a Microsoft Excel spreadsheet, a Microsoft Word document, graphics, sounds, or other binary data) linked to or embedded in a Microsoft Access table.	Up to 1 gigabyte (limited by available disk space) OLE : Object Linking and Embedding

Hyperlink أرتباط تشعُبي	Text or combinations of text and numbers stored as text and used as a hyperlink address (hyperlink address: The path to a destination such as an object, document, or Web page). A hyperlink address can be a URL (address to an Internet or intranet site) or a UNC network path (address to a file on a local area network).). A hyperlink address can have up to four parts: Text to display— the text that appears in a field or control. Address—the path to a file (UNC path) or page (URL). Sub address—a location within the file or page . Screen tip—the text displayed as a tool tip .	Each part of the three parts of a Hyperlink data type can contain up to 2048 characters ربط الخلية الحالية بملف خارجي ، حيث يُستخدم كأرتباط تشعُبي لعنوان مُعين
Lookup Wizard مُعالج بحث	Creates a field that allows you to choose a value from another table or from a list of values by using a list box or combo box. Clicking this option starts the Lookup Wizard, which creates a Lookup field (Lookup field: A field, used on a form or report in an Access database, that either displays a list of values retrieved from a table or query, or stores a static set of values.). After you complete the wizard, Microsoft Access sets the data type based on the values selected in the wizard.	The same size as the primary key field used to perform the lookup, typically 4 bytes. يُستخدم عند ادخال بيانات أو قيم مُكررة في الحقل المُخصص لها ، مثلاً حقل أسم المدينة أو حقل الجنس (ذكر / أنثى) . أي تكوين حقل لغرض أختيار قيم من جدول آخر أو من قائمة من القيم (اعداد قائمة من الخيارات) .

الوصف (Description) : هذا الحقل اختياري يُستخدم لوصف محتويات الحقل وأي مُلاحظات أخرى ، وبالتالي يُسهل على المُستخدم فهم هدف الحقل بعد مرور فترة زمنية . ويحتوي هذا الحقل على سعة 255 حرفاً أبجدياً كحد أقصى . فمثلاً يتم وصف حقل الأسم المُمثل بالحقل

الأول بعبارة " أسم الطالب " وذلك للدلالة على أن بيانات هذا الحقل تحمل أسماء الطلاب ، وكذلك الحال بالنسبة للحقل الثاني فسيتم وصف حقل الصور بعبارة " الصورة الشخصية " للدلالة على أن بيانات هذا الحقل تحمل الصورة الشخصية لكل طالب من الطلاب المُمثليين بجدول الطلاب وكما موضح ذلك بالشكل رقم (2 – 5) .

2 – 10 – 3 تحديد خصائص للحقول المُدرجة مع بياناتها : Field Properties

خصائص الحقل هو الجُزء السُفلي من نافذة تصميم الجدول ، والذي يُستخدم لعمل الأعدادات الخاصة بكل عُنصر ولكل الحقول المُستخدمة في الجدول (ليس بالضرورة أن تحتوي جميع أنواع البيانات المُخصصة للحقول على نفس الخصائص) ، مثل :

1. حجم الحقل (Field Size) : يستخدم لتحديد حجم البيانات المُدخلة في الحقل (أي تحديد الحد الأقصى لبيانات الحقل المدُخلة سواءً أكانت بيانات نصية أو رقمية)، ففي حالة كون البيانات المدخلة نصية فيتم تحديد حجمها field size بالمدى (0 – 255) حرفاً أبجدياً . في الحالة الأفتراضية default فتكون محددة بحجم (50) حرفاً أبجدياً . أما في حالة كون البيانات المدخلة في حقل من نوع ترقيم تلقائي ، فأن حجم الحقل سيكون مُحدد بالخاصية عدد صحيح طويل long integer . أما في حالة كون البيانات المُدخلة في حقول نوعها رقمية number ، فأن حجم الحقل سيكون مُحدد بأحدى الخصائص التالية والموضحة أدناه بالجدول رقم (2 – 2) .

جدول رقم (2 – 2) :خاصية حجم الحقل المُحددة للبيانات في الحقول الرقمية .

Setting	Description	Storage size
Byte	Stores numbers from 0 to 255 (no fractions).	1 byte
Integer	Stores numbers from –32,768 to 32,767 (no fractions).	2 bytes
Long Integer	(Default) Stores numbers from –2,147,483,648 to 2,147,483,647 (no fractions).	4 bytes
Single	Stores numbers from –3.402823E38 to –1.401298E–45 for negative values and from 1.401298E–45 to 3.402823E38 for positive values (with fraction).	4 bytes

Double	Stores numbers from –1.79769313486231E308 to –4.94065645841247E–324 for negative values and from 4.94065645841247E–324 to 1.79769313486231E308 for positive values.	8 bytes

2. التنسيق (Format) : تُستخدم هذه الخاصية لعمل اعداد للحقل الذي نتعامل معهُ وأظهار المعلومات بشكل مُعين .

Allows you to specify how numbers , dates , times and text are displayed and printed .

3. قناع الأدخال (Input Mask) : يُستخدم لعمل أو تحديد طريقة ادخال البيانات التي نتعامل معها بشكل مُعين وكما موضحة بالجدول رقم (2 – 3) ، حيث تُسيطر بها على الأدخال من خلال الضغط على مفتاح F1 من لوحة المفاتيح .

Allows you to simplify the data entry for fields that have the same format . Such as phone number field by creating an input mask .

جدول رقم (2 – 3) : الحروف والعلامات الخاصة بقناع الأدخال .

Character	Description
0	Digit (0 to 9, entry required, plus [+] and minus [–] signs not allowed).
9	Digit or space (entry not required, plus and minus signs not allowed).
#	Digit or space (entry not required; spaces are displayed as blanks while in Edit mode, but blanks are removed when data is saved; plus and minus signs allowed).
L	Letter (A to Z, entry required).
?	Letter (A to Z, entry optional).

A	Letter or digit (entry required).
a	Letter or digit (entry optional).
&	Any character or a space (entry required).
C	Any character or a space (entry optional).
. , : ; - /	Decimal placeholder and thousand, date, and time separators. (The actual character used depends on the settings in the Regional Settings Properties dialog box in Windows Control Panel).
<	Causes all characters to be converted to lowercase.
>	Causes all characters to be converted to uppercase.
!	Causes the input mask to display from right to left, rather than from left to right. Characters typed into the mask always fill it from left to right. You can include the exclamation point anywhere in the input mask.
\	Causes the character that follows to be displayed as the literal character (for example, \A is displayed as just A).

في حالة جعل قناع الأدخال يأخذ الخاصية " Password " في الحقل الذي بياناته من نوع قيم نصية text فسيتم طباعة أي قيمة نصية يتم ادخالها ضمن هذا الحقل فقط على شكل علامة النجمة (*) ، حيث تُستخدم هذه الخاصية لجعل المعلومات تبدو سرية وغير مفهومة للأشخاص الغير مخولين .

4. المراتب العُشرية (Decimal Places) : تُستخدم لتحديد عدد المراتب بعد الفاصلة العُشرية

The number of digits that allows you to specify the number of places to display to the right of the decimal , it's for numeric and currency field only .

5. العنوان (Caption) : هو الأسم الذي يظهر في رأس الجدول لكل حقل يتم اضافته .

Allows you to define a default field label to appear on forms and reports

6. قاعدة التحقق من الصحة (Validation Rule) : هي عبارة عن شرط يُوضع في تلك الخانة بحيث لاتتعدى القيمة الموضوعة بالحقل ذلك الشرط .

7. نص التحقق من الصحة (Validation Text) : هي الرسالة التي تظهر في حالة ما اذا كانت القيمة الموضوعة (المُدخلة) مًخالفة للشرط الذي يتم التعامُل معه في قاعدة التحقق من الصحة .

8. القيمة الأفتراضية (Default Value) : هي القيمة التي يتم ادخالها في خصائص الحقل ، والتي ستظهر تلقائياً عند التنفيذ .

9. مطلوب (Required) : يوجد بها خياران (نعم / لا) ، في حالة اختيار (نعم) يجب ادخال القيمة (البيانات) داخل الحقل ولايجب تركها خالية . ففي حالة أرقام التلفونات مثلاً ليس بالضرورة أن يكون لدى كل موظف رقم تلفون ففي هذه الحالة نختار (لا) . القيمة المُدخلة مُمكن أن تكون بضغط مفتاح (Space) من لوحة المفاتيح (في حالة نوع حقل البيانات نص) .

10. مُفهرس (Index) : المقصود بها عمل فهرسة للحقل بالكامل ، حيث تُساعدنا في سُرعة ترتيب Sorting والبحث Searching عن بيانات الحقل . وتوجد فيها ثلاثة خيارات :

- نعم مُكرر (Yes Duplicate Ok) : تسمح بعمل فهرسة وتسمح بتكرار بعض المعلومات .
- نعم غير مُكرر ((Yes (No Duplicates : تسمح بعمل فهرسة ولاتسمح بتكرار المعلومات.
- لا (No) : لاتسمح بعمل فهرسة .

مُلاحظـة :

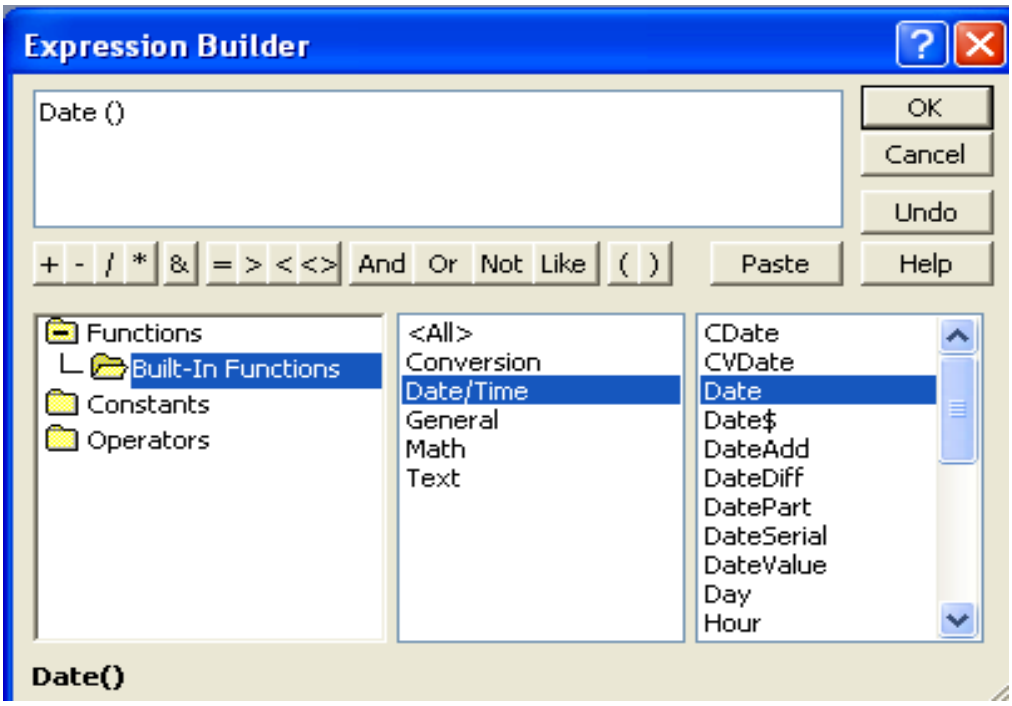
- اذا كان نوع بيانات الحقل (Memo) ، فنستطيع جعل النصوص المكتوبة بداخله تظهر بشكل مُتدرج من خلال ضغط مفتاحي (Ctrl + Enter) من لوحة المفاتيح .
- يُمكن بأستخدام كائن (OLE) ادراج صورة في الحقل عند التنفيذ ، وذلك من خلال الأمر التالي

R.Click على حقل البيانات عند التنفيذ ← Insert Object ← Create from File

- لأظهار الأرتباط التشعُبي على بيانات الحقل (Hyperlink) ، يتم ذلك من خلال الأمر التالي

R.Click على حقل البيانات عند التنفيذ ← Hyperlink ← Edit Hyperlink

- يُوجد مُعالج يُطبق على خصائص الحقول ، يُسمى بـ (Expression Builder) : مُحرر أو مُنشأ التعابير ، حيث توضع داخله شروط مُعينة . مثلاً في حقل البيانات من نوع (Date/Time) يُمكن جعل خاصية القيمة الأفتراضية (Default Value) بالشكل رقم (2 – 7) .



شكل رقم (2 – 7) : مُحرر أو مُنشأ التعابير

حيثُ أن هذا الأمر سيُظهر لنا التأريخ الحالي (لليوم الحالي حسب تأريخ الحاسبة المُستخدمة) تلقائياً عند التنفيذ ولجميع بيانات الحقل دون الحاجة لملئها يدوياً ، أي السرعة في ملئ بيانات الحقول المكررة .

2 – 10 – 4 حُفظ الجدول مع انشاء مفتاح أساسي اذا تطلب الأمر: Saving Table

بعد الأنتهاء من اضافة حقول الى الجدول لابُد من حُفظ تصميمه قبل أن نتمكن من اضافة سجلات (بيانات الحقول) اليه . وبعد حفظ الجدول سيظهر عنوانه في قائمة جدول في نافذة قاعدة البيانات الرئيسية . كما يُمكن تغيير أسم الجدول الذي تم تصميمه وحسب الرغبة .

المفتاح الأساسي : Primary Key

بعد الأنتهاء من تعيين أسماء الحقول وتحديد نوع البيانات المُدخلة فيها مع خصائصها يتم الأنتقال الى مرحلة حُفظ الجدول . أثناء عملية الحُفظ سيطلب منا برنامج مايكروسوفت أكسس تحديد فيما اذا كانت هناك حاجة لأنشاء مفتاح أساسي أم لا . لذلك من المُهم جداً معرفة ماهو المفتاح الأساسي ؟ حيث يُعرف كل سجل في الجدول بشكل مُنفرد من خلال البحث عن مفتاح أساسي لتسهيل

عملية البحث عن البيانات ودمجها وأسترجاع السجلات من الجدول . ولتعيين المفتاح الأساسي يتم من خلال تحديد الحقل المطلوب جعل مفتاح أساسي له ونقرة يمين بالفأرة وأختيار الأمر Primary Key ، أو من قائمة تحرير (Edit) أو تحديد الأيقونة الموجودة في شريط الأدوات .

أما في حالة نسيان مُصمم قاعدة البيانات تعريف (مفتاح أساسي) لأي جدول ، فسوف يقوم برنامج مايكروسوفت أكسس بتذكيره عند الأنتهاء من التصميم .

مُلاحظة : ليس بالضرورة أن نعمل مفتاح أساسي في الجدول ، حيث تكمُن أهمية المفتاح الأساسي في حالة وجود علاقات بين الجداول (Relationships) .

A primary Key doesn't allow Null values and must always have a unique index. A primary Key is used to relate a table to Foreign Key in other tables.

2 – 11 مُعالجة البيانات : Manipulating Data

هُناك عدة عمليات يُمكن اجرائها على البيانات (سجلات الحقول) أثناء عرض صفحة البيانات ، وهذه العمليات هـي :

1. اضافة قيد جديد (Adding a new record) .

Simply drop down to a new line and enter the information .

2. تحديث قيد (Updating a record) .

Simply select the record and field you want to update , and change it's data with what you want .

3. حذف قيد (Deleting a record) .

Simply select the entire row and hit the Delete Key on the keyboard .

2 – 12 عرض البيانات في الجدول : Viewing data in table

تُعرض البيانات في الجدول بطريقتين ، هما :

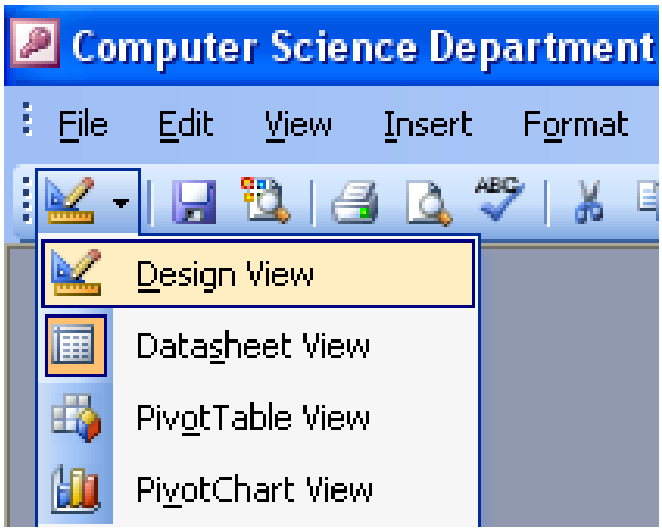
1. طريقة عرض التصميم (Design View) :حيث يتم عرض هيكلية الجدول ضمن قاعدة البيانات مع أمكانية اضافة حقول أو اجراء تعديلات عليها .

2. طريقة عرض صفحة البيانات (Datasheet View) : هنا يتم ادخال البيانات وتحريرها والبحث عنها .

2 – 13 الأنتقال بين نافذة عرض التصميم ونافذة عرض صفحة البيانات:Switching Views

نافذة صفحة البيانات Datasheet View هي النافذة التي يتم فيها ادخال السجلات والبيانات الى الجدول ، وللتنقل مابين صفحة البيانات وطريقة عرض تصميم الجدول وكما موضحة بالشكل رقم (2 – 8) نقوم بالتالي :

من نافذة صفحة البيانات يتم النقر على تصميم الجدول Design View من قائمة عرض View الموجودة في شريط القوائم ، أو بالنقر على الأيقونة الموجودة في أقصى يسار شريط الأدوات فننتقل بذلك الى نافذة عرض التصميم وبنفس الطريقة يتم الأنتقال بالعكس .



شكل رقم (2 – 8) : الأنتقال بين نافذة عرض التصميم ونافذة عرض صفحة البيانات

2 – 14 تغيير أتجاه نافذة عرض صفحة البيانات : Direction of data view

يُمكن تغيير اتجاه عرض صفحة البيانات (Datasheet View) ليكون من اليمين الى اليسار كما هو الحال في الجداول العربية ، ومن اليسار الى اليمين في الجداول الأنكليزية وذلك من خلال اتباع الخطوات التالية . (تُنفذ هذه الخطوات قبل البدء بتصميم الجدول في صفحة Design View).

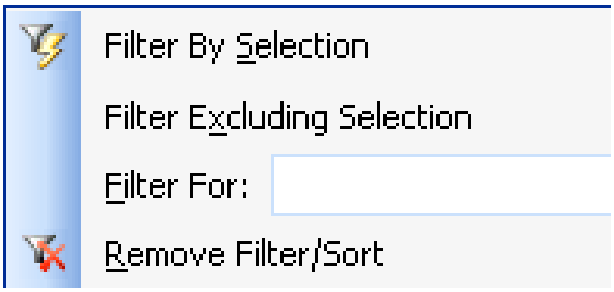
Tools ⟵ Options ⟵ International ⟵ Default Direction
⟵ Left-to-right
⟵ Right-to-left

2 – 15 ترتيب السجلات في الجدول : Sorting record in a table

يُمكن (فرز) أو ترتيب السجلات في الجدول تصاعُدياً أو تنازُلياً ، فمثلاً يُمكن اجراء فرز حسب ترتيب الحروف الأبجدية للأسماء وذلك بوضع مؤشر الفأرة في حقل أسماء الطُلاب ثُم نقر آيقونة فرز تصاعدي (Sort Ascending) الموجودة في شريط الأدوات فيتم بذلك ترتيب الأسماء أبتداءاً من (A) وأنتهاءاً الى (Z) والعكس في حالة أختيار الفرز التنازلي (Sort Descending)

2 – 16 تصفية البيانات : Data Filtering

هُناك ثلاثة أنواع من التصفية والتي يتم من خلالها تقليص حجم عرض البيانات وذلك بالأعتماد على نوع التصفية المُستخدمة . وهذه الأنواع الثلاثة من التصفية تظهر عند النقر بزر الفأرة الأيمن على حقل البيانات المُراد اجراء تصفية لبياناته فتظهر لنا أنواع التصفية الموجودة حيث يتم أختيار احداها ومن ثم اختيار الأمر (Apply Filter) ، كما ويمكن الغاء عامل التصفية وذلك من خلال أختيار الأمر(Remove Filter) وكما موضح ذللك بالشكل رقم (2 – 9) وهي :

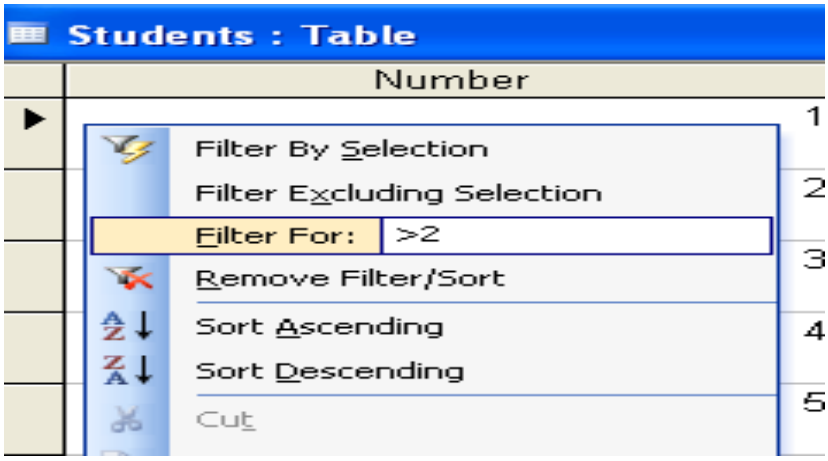


شكل رقم (2 – 9) : أنواع تصفية البيانات

1. تصفية حسب الأختيار (Filter By Selection) : أي تصفية (الغاء) جميع بيانات الجدول عدا بيانات السجل (الصف) المُحددة احدى بيانات حقوله .

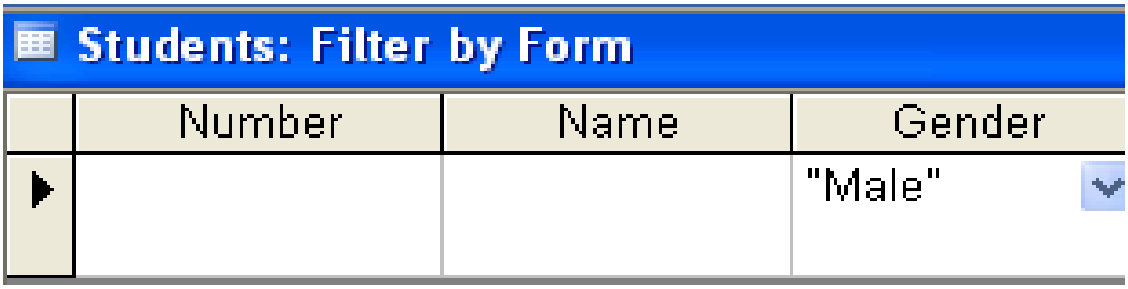
2. تصفية بأستثناء الأختيار (Filter Excluding Selection) : هذا النوع من التصفية يعمل عكس النوع الأول ، حيث يتم فقط تصفية السجل المُحدد احدى بيانات حقوله مع اظهار جميع السجلات الأخرى (الغير مُحددة) في الجدول مع بياناتها .

3. تصفية لـ (Filter For) : هنا يتم عمل تصفية لبيانات الجدول بالأعتماد على شرط مُعين يوضع داخل المستطيل الظاهر أمام هذا النوع من التصفية والموضح بالشكل رقم (2 – 10) ، ومن ثم الضغط على مفتاح (Enter) من لوحة المفاتيح .



شكل رقم (2 – 10) : تصفية بيانات الجدول بأستخدام شرط Filter For

4. اضافة الى ذلك هنالك نوع رابع من التصفية الخاصة والذي يظهر في شريط الأدوات ويُسمى تصفية حسب النموذج (Filter by Form) : ويتم فيه تحديد الشرط المراد تنفيذه لأي حقل من حقول الجدول وبشكل مُباشر وكما موضح بالشكل رقم (2 – 11) . هذا النوع من التصفية يُعتبر مهم جداً لأنه يُمكن خزن أو حفظ ناتج تطبيق عامل التصفية على هيئة أستعلام Query ، وهذا مايُميزه عن باقي الأنواع الأخرى .



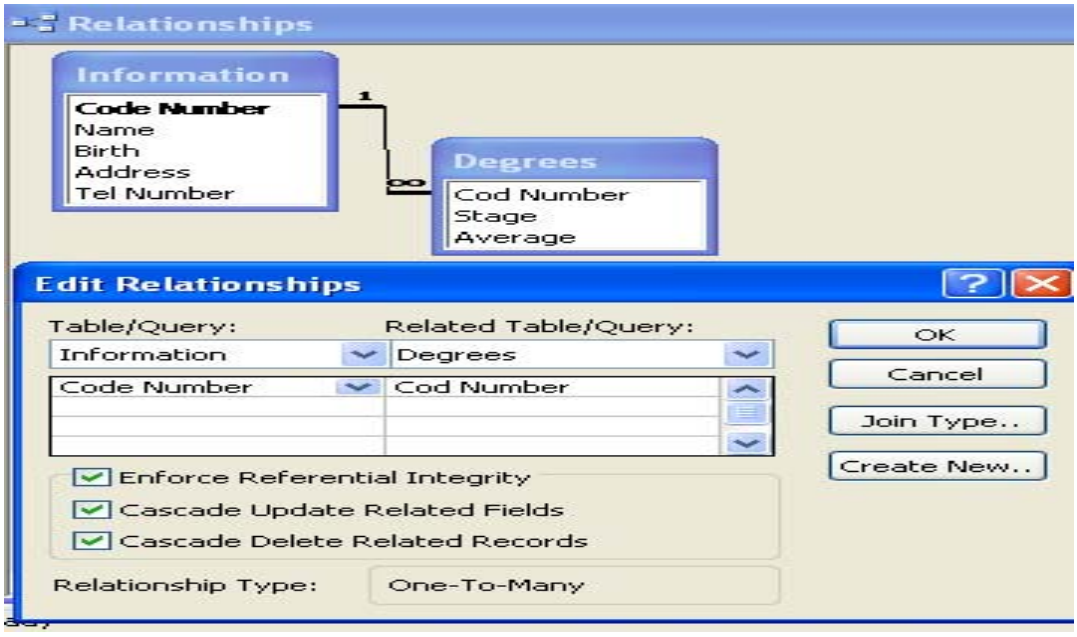
شكل رقم (2 – 11) : تصفية حسب النموذج لبيانات الجدول

2 – 17 العلاقات بين الجداول في قاعدة البيانات: Relationships between tables

العلاقات هي عملية ربط بين الجداول حيث تُفيد في الربط المُباشر بين جدولين مُختلفين ولكنهُما يشتركان بحقل يكون هو حلقة الوصل بينهُما ، حيث يُمكن بواسطة العلاقات جعل برنامج مايكروسوفت أكسس يقوم بجلب المعلومات المهمة التي نحتاجُها من جدولين أو أكثر . فمثلاً يرتبط جدول الطُلاب بجدول الدرجات من خلال حقل رمز الطالب والذي يكون مُشترك لكلا الجدولين وكما موضح بالشكل رقم (2 – 12) ، ويرتبط جدول العُملاء بجدول المبيعات من خلال حقل أسم العميل . ان العلاقات بين الجداول تختصر الزمن في ايجاد علاقة مُباشرة بين جدولين أو أكثر وبالتالي تقليص التأخير في ايصال الناتج بأختصار دقيق .

يُمكن أن نستفيد من العلاقات لاحقاً في ربط أكثر من جدول داخل نموذج البيانات (Form) والذي تتُم فيه غالبية عمليات البرمجة (VBA) ، اضافةً الى امكانية الربط بين الجداول داخل تصميم التقرير (وهو ما نُسميه في التقرير بالتقارير الفرعية) .

بعد تكوين العلاقات بين الجداول يُمكن عمل استعلام أو نموذج أو تقرير ، حيث سيتم عرض معلومات عن عدة جداول في الوقت نفسه . كما توفر العلاقات امكانية ربط الجدول الرئيسي بأكثر من جدول فرعي ، كذلك امكانية الحذف والتحديث التلقائي بين بيانات الجدول الرئيسي والفرعي من خلال حقل الربُط فقط .



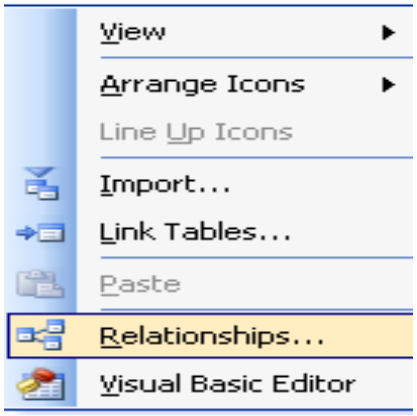
شكل رقم (2 – 12) : العلاقة بين جدول الطلاب وجدول الدرجات من خلال حقل الربط

يتم عمل العلاقات بين الجداول في قاعدة البيانات من خلال مُطابقة البيانات (matching) بين حقول الجداول وعن طريق مفتاح الربط primary key .

In most cases , these matching fields are the primary key from one table , which provides a unique identifier for each record , and a foreign key in the other table .

2 – 17 – 1 تعريف العلاقات : Defining the relationships

يتم تعريف العلاقة بين جدولين من خلال اضافة الجدولين الى شاشة العلاقات (Relationships Window) من خلال نقر بزر الفأرة الأيمن على نافذة قاعدة البيانات الرئيسية Database Window وأختيار الأمر Relationships وكما موضح ذلك بالشكل رقم (2 – 13) ، أو من خلال قائمة أدوات Tools الموجودة في شريط القوائم ، ثم اجراء عملية السحب لأحد حقول الجدول الأول والأفلات الى حقل مُطابق له في الجدول الثاني .

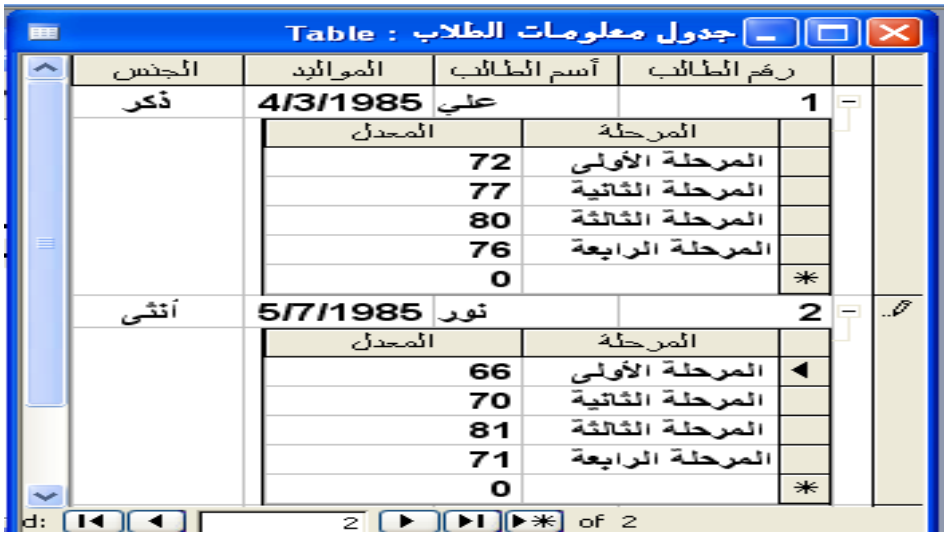


شكل رقم (2 – 13) : تعريف العلاقات بين الجداول

2 – 17 – 2 أنواع العلاقات : Types of Relationships

1. علاقة ($1 \longrightarrow \infty$) : A One-to-Many Relationship

هي الطريقة الأكثر شيوعاً وأستخداماً ، حيث أن كل قيد في الجدول A يمتلك عدة قيود في الجدول B لكن كل قيد في الجدول B يمتلك قيد واحد فقط في الجدول A . لاحظ الشكل رقم (2 – 14) .



شكل رقم (2 – 14) : علاقة واحد الى مجموعة بين جدولين في قاعدة البيانات

2. علاقة (1 ⟶ 1) : A One-to-One Relationship

كل قيد في الجدول A له قيد واحد فقط في الجدول B وكل قيد في الجدول B له قيد واحد فقط في الجدول A . هذا النوع من العلاقات لايُستخدم بشكل واسع لأن جميع البيانات فيه تكون مربوطة بجدول واحد فقط . الغرض من استخدام هذا النوع من العلاقات هو لتقسيم الجدول لعدة جداول ، ولتوفير الأمنية . لاحظ الشكل رقم (2 – 15)

شكل رقم (2 – 15) : علاقة واحد الى واحد بين جدولين في قاعدة البيانات

3. علاقة (∞ ⟶ ∞) : A Many-to-Many Relationship

كل قيد في الجدول A يمتلك عدة قيود في الجدول B ، وكذلك كل قيد في الجدول B يمتلك عدة قيود في الجدول A . هذا النوع من العلاقات يتم العمل به فقط من خلال تعريف جدول ثالث يُسمى جدول الربط أو جدول الأتصال (Junction Table) .

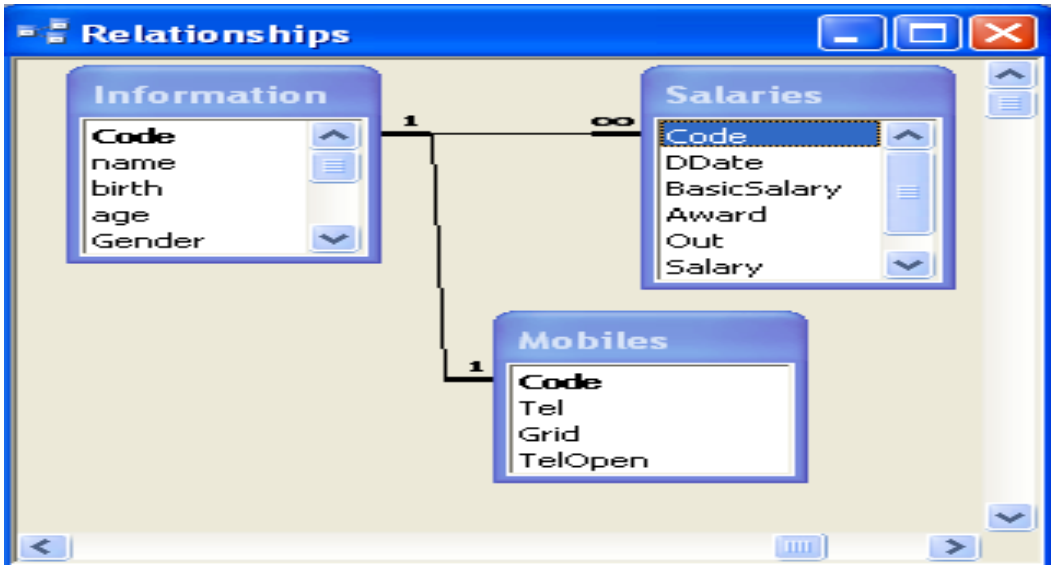
ولغرض تكوين العلاقة بين الجداول لابُـد من توفر الشروط التالية ، لاحظ الشكل رقم (2 – 16) :

The kind of relationships that Microsoft Access creates depends on how the related files are defined :

1. A One-to-Many Relationship is created if only one of the related fields is a primary key or has a unique index .

2. A One-to-One Relationship is created if both of the related fields are primary keys or have unique indexes .

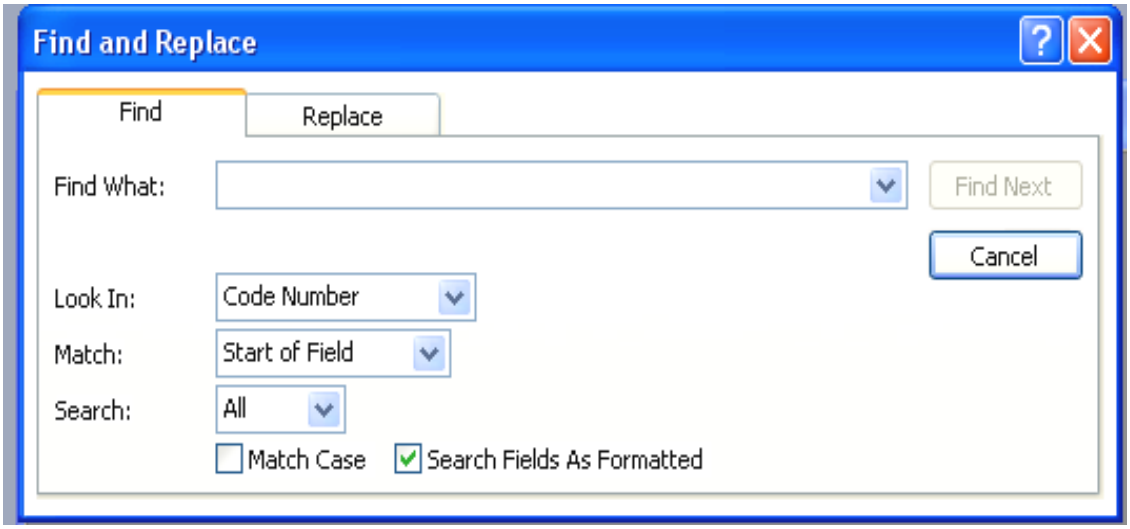
3. A Many-to-Many Relationship is really two (One-to-Many Relationships) with a third table called Junction table whose primary key consists of two fields —— the foreign keys from the two other tables .



شكل رقم (2 – 16) : تصميم العلاقات بين الجداول

2 – 18 البحث والأستبدال : Find & Replace

تُستخدم للبحث عن حرف أو نص أو رقم ، وأستبدالها بحرف أو نص أو رقم أخر ضمن بيانات السجلات التي تم انشاءها . ويتم اجراء عمليتي البحث والأستبدال من خلال النقر على الأمر تحرير Edit من شريط القوائم وأختيار الأمر بحث Find ، أو من خلال النقر على الآيقونة الموجودة في شريط الأدوات فتظهر لنا النافذة بالشكل رقم (2 – 17) .



شكل رقم (2 – 17) : نافذة البحث والأستبدال

يتم في (مُربع البحث عن Find What) كتابة كلمة أو حرف أو رقم معين والذي نُجري البحث عنه ، ثم النقر على الأمر Find Next (ابتداءاً يتم تحديد الحقل الذي نبحث عن بياناته) .

وفي حالة استبدال كلمة أو حرف بأخر فيتم الضغط على الأمر Replace ، وفي حالة استبدال الكلمة آينما وجدت في الحقل بكلمة أخرى فيتم النقر على الأمر Replace All . (البحث عادةً يتم من بداية الحقل Start of Field) .

2 – 19 التنقُل عبر قيود (سجلات) الجدول : Table Navigation Buttons

لغرض التنقل عبر حقول الجدول (من سجل الى آخر) ، فيتم بأستخدام شريط التمرير (Navigation bar) الموضح بالشكل رقم (2 – 18) والموجود في أسفل يسار الجدول (Datasheet View) والذي يتكون من خمسة أوامر والموضحة في الجدول رقم (2 – 4) .

شكل رقم (2 – 18) : التنقل عبر سجلات الجدول

جدول رقم (2 – 4) : أزرار التنقل عبر سجلات الجدول مع وظائفها .

Object	Role
	First Record : allows moving to the first record of the table .
	Previous Record : allows to move one record back (if there is one).
	Record Indicator : Displays the number representing the current record .
	Next Record : allows moving you one record ahead .
	Last Record : allows moving you to the last record of the table .
	New Record : use to enter a new record on the table .

3 – 1 المقدمة : Introduction

في هذا الفصل سيتُم التعرُف على كيفية عمل وتكوين ملف قاعدة بيانات وذلك من خلال أستخدام احدى كائنات الفيجوال بيسك والتي تسمح لنا بالتعامُل مع ملفات القاعدة . الشئ المهم ذكره هنا هو أن برنامج الفيجوال بيسك مُصمم لتنفيذ العديد من المهام البرمجية ومنها تصميم قواعد البيانات Databases ، الا انه وبالرغم من الأمكانيات التي يوفرها في تصميم قواعد البيانات يُمكننا الحصول على دعم أفضل من خلال أستغلال الأمكانيات التي يوفرها لنا برنامج مايكروسوفت أكسس حيث أنه مُصمم أصلاً لتنفيذ كل مانحتاجُ اليه في تصميم قواعد البيانات . لذلك ومن أجل الحصول على تصميم أفضل سنقوم بتصميم قاعدة بيانات بأستخدام برنامج مايكروسوفت أكسس وذلك من خلال تصميم جدول لملف معين يحوي كافة الحقول المطلوبة عن بيانات الملف مع تحديد نوع بيانات كل حقل من الحقول المستخدمة وتحديد الخصائص المُتاحة لكل حقل من حقول الجدول ، ومن ثم ربط هذه القاعدة مع برنامج فيجوال بيسك من خلال استخدام كائن الربط ، وبذلك سيتم أستغلال جميع الأمكانيات المُتوفرة لدى كُل من برنامج الفيجوال بيسك وبرنامج مايكروسوفت أكسس ، وهذا هو المطلوب .

3 – 2 انشاء برامج قواعد البيانات : Creating Database Programs

يعتبر برنامج لغة فيجوال بيسك من البرامج الكفوءة في التعامل مع قواعد البيانات ، حيث يوفر لنا مرونة عالية للربط مع قواعد البيانات المختلفة وبعدة طرق مختلفة . وهذه الطرق ، هــي :

1. عنصر التحكم المرتبط بالأدخال Data Control .
2. كائن مجموعة السجلات (Record Set Object) ، أي تحميل مكتبة الكائنات (DAO object library) .
3. كائن البيانات ADO Data Control .

3 – 3 عُنصر التحكم المُرتبط بالأدخال : The Bound Input Control

يُسمى الكائن Data بعنصر الوصول للبيانات حيث يُعتبر بمثابة حلقة الوصل بين الفيجوال بيسك وقواعد البيانات مثل Access أو SQL Server أو Oracle أو FoxPro . يُمكن اعتبار الكائن Data هو الأساسي في عملنا حالياً للربط مع قواعد البيانات من نوع MS-Access 97 . والخصائص التي يرتبط بها هذا الكائن مع القاعدة ، هــي :

1. Database Name : اسم القاعدة المُختارة ومسارها (تحديد المسار الكامل لقاعدة البيانات التي نُريد الأتصال معها) .
2. Record Source : جدول المُعطيات (جدول البيانات) المُختار من القاعدة ، أي تحديد مصدر السجلات (تحديد اسم الجدول المُعين من قاعدة البيانات الذي نُريد التعامُل معه) .
3. Connect : نوع قاعدة البيانات المُرتبط بها .

أما عُنص التحكم أو الكائن الذي يرتبط بحقل البيانات الموجودة في الجدول فهو مُربع النص Text Box . والخصائص التي يرتبط بها مع الحقل ، هــي :

1. Data Source : تحديد اسم عنصر التحكم من النوع Data والمرتبط مع جدول في قاعدة البيانات .
2. Data Field : تحديد اسم الحقل من الجدول والذي نُريد لعنصر التحكم أن يرتبط معه .
3. Caption : العنوان (الأسم المعروض لعنصر التحكم .
4. Name : الأسم البرمجي لعنصر التحكم (Text Box) .

ومن أهم الطرق والعمليات التي يتعامل معها كائن عنصر الوصول للبيانات (Data) والموضحة بالجدول رقم (3 – 1) ، هــي :

1. اضافة سجل جديد : Add New Record .
2. حفظ السجل الحالي : Save Current Record .
3. الغاء تحديث السجل الحالي : Cancel Update Current Record .
4. مسح السجل الحالي : Delete Current Record .
5. الأنتقال الى السجل الأول : Move to First Record .
6. الأنتقال الى السجل الأخير : Move to Last Record .
7. الأنتقال الى السجل السابق : Move to Previous .
8. الأنتقال الى السجل التالي : Move to Next .
9. اختبار الوصول لبداية السجل : Is Begin Of File .
10. اختبار الوصول لنهاية السجلات : Is End Of File .
11. البحث عن أول سجل يُحقق الشرط : Find First-criteria .

12.البحث عن السجل التالي الذي يُحقق الشرط : Find Next-criteria .

جدول رقم (3 – 1) : العمليات المرتبطة بكائن عنصر الوصول للبيانات .

Data.RecordSet.AddNew	Data.RecordSet.MovePrviouse
Data.UpdateRecord	Data.RecordSet.MoveNext
Data.UpdateControls	Data.Record.BOF
Data.RecordSet.Delete	Data.RecordSet.EOF
Data.RecordSet.MoveFirst	Data.RecordSet.FindFirst criteria
Data.RecordSet.MoveLast	Data.RecordSet.FindNext criteria

3 – 4 أزرار التنقل عبر السجلات : Records Navigation

ان الكائن Data Control ينقُلنا الى أي موقع ضمن الجدول المربوط به ، ولكن وفي حالة استخدام أزرار الأوامر Command Buttons فان ذلك يكون من خلال الشفرات التالية :

1. Move to the First Record

```
Private Sub cmdFirst_Click()
On Error GoTo Error
Data1.Recordset.MoveFirst
Error:
If Err.Number = 30 Then
  Exit Sub
End If
End Sub
```

2. Move to the Last Record

```
Private Sub cmdLast_Click()
On Error Resume Next
```

```
Data1.Recordset.MoveLast
End Sub
```

3. Move to the Next Record

```
Private Sub cmdNext_Click()
On Error Resume Next
Data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
   Data1.Recordset.MoveLast
End If
End Sub
```

4. Move to the Previous Record

```
Private Sub cmdPrevious_Click()
On Error Resume Next
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
   Data1.Recordset.MoveFirst
End If
End Sub
```

3 – 5 تحديد موقع السجل الحالي : Detect Current Record Position

يتم تحديد موقع السجل الحالي بأستخدام الأيعاز Absolute Position والذي يُعطينا القيمة الفعلية لموقع السجل الحالي مُضافاً اليه الرقم (رقم السجل) لكي يكون واضح أمام المُستخدم ، هذا وبمُساعدة الأيعاز Record Count .

```
Private Sub Data1_Reposition()
Data1.Caption = "Record : " & Data1.Recordset.AbsolutePosition + 1 _
```

```
& " of " & _
Data1.Recordset.RecordCount
End Sub
```

هذه العلامة تعني تكملة سطر البرنامج بالسطر الثاني

كذلك يجب كتابة الشفرة التالية على النموذج عندما يكون حدث النموذج نشطاً ، وكالأتي :

```
Private Sub Form_Activate()
On Error GoTo Error
DU = 0
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Error:
If Err.Number = 30 Then
  Exit Sub
End If
End Sub
```

3 – 6 اضافة سجل جديد : Adding New Record

سيتم اضافة قيد جديد مع ملاحظة أنه بعد كل عملية اضافة جديدة يجب زيادة العداد الخاص بمؤشر كائن عنصر الوصول للبيانات (Data) .

```
Private Sub cmdNewRecord_Click()
Data1.Recordset.AddNew
Data1.Caption = Data1.Recordset.RecordCount + 1 _
& " of " & _
Data1.Recordset.RecordCount + 1
End Sub
```

3 – 7 حفظ السجل الحالي : Saving Current Record

بعد تكوين سجل جديد يُمكن حفظ بياناته من خلال كتابة الشفرة التالية :

```
Private Sub cmdSaveRecord_Click()
Data1.UpdateRecord
Data1.Recordset.MoveLast
End Sub
```

OR

```
Private Sub cmdSaveRecord_Click()
Data1.UpdateRecord
Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub
```

حيثُ أن Bookmark تُمثل تأشيرة أو دالة لبيان الموقع الحالي (الأخير) للسجل .

3 – 8 الغاء تحديث السجل الحالي : Cancel Saving

في هذا المطلب نستخدم الأيعاز UpdateControls مع مُلاحظة أنه يعمل فقط عند بقاء المؤشر على السجل الحالي .

```
Private Sub cmdCancelUpdate_Click()
Data1.UpdateControls
End Sub
```

3 – 9 حذف السجل الحالي : Deleting Current Record

حذف سجل يعني حذف قيد كامل مع جميع بياناته . بعد اجراء عملية الحذف يتم الرجوع الى السجل السابق وفي حالة الأستمرار بعملية الحذف سنصل الى أول سجل ومن ثم مسح جميع سجلات الجدول .

```
Private Sub cmdDelete_Click()
On Error GoTo Error
Data1.Recordset.Delete
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
```

```
  Data1.Recordset.MoveFirst
End If
Error:
If Err.Number = 30 Then
Exit Sub
End If
End Sub
```

ملاحظة : في البرنامج التالي قد تم استخدام متغيرات وهي (n , NA , DU , DA) لذلك لابد من تعريفها وذلك في قائمة General وكالأتي :

```
Dim n As Integer
Dim NA As String
Dim DU As Integer
Dim DA As Date
```

3 – 10 البحث عن بيانات الحقول : Searching

من الوظائف المهمة التي تجرى على قاعدة البيانات هي البحث عن بيانات معينة داخل حقول القاعدة وفقاً لشروط معينة توضع مُسبقاً من قبل مصمم قاعدة البيانات ، والتي تُساعدنا في سرعة عرض البيانات واجراء التعديلات اللازمة عليها ، كذلك لمعرفة فيما اذا كان عنصر ما موجود ضمن القاعدة أم لا . سيتم استخدام ايعازات البحث التابعة للكائن Data Control وهي ايعاز (FindFirst و ايعاز FindNext) . حيث يُستخدم الأيعاز الأول للبحث عن أول قيمة ضمن حقل البيانات . أما الأيعاز الثاني فيُستخدم للبحث عن القيمة التالية لنفس بيانات الحقل مع امكانية معرفة مدى تكرار هذه القيمة ضمن الحقل نفسه ، وذلك من خلال كتابة الشفرات التالية :

Searching for Number Field

```
Private Sub cmdSearchNo_Click()
On Error Resume Next
n = InputBox("Enter the Number")
Data1.Recordset.FindFirst "[Code]=" & n
If Data1.Recordset.NoMatch = True Then
  MsgBox "Try Again", vbOKOnly, "Error Number"
End If
End Sub
```

أسم حقل الرقم المعرف في قاعدة بيانات أكسس

Searching For Birth Field

```
Private Sub cmdSearchDate_Click()
On Error Resume Next
DA = InputBox("Enter the Birth")
Data1.Recordset.FindFirst "[birth]=#" & DA & "#"
If Data1.Recordset.NoMatch = True Then
  MsgBox "Try again", vbOKOnly, "Error Birth"
End If
End Sub
```

أسم حقل المواليد المعرف في قاعدة بيانات أكسس

Searching For Text Field

```
Private Sub cmdSearchText_Click()
DU = 0
NA = InputBox("Enter The Name")
Data1.Recordset.FindFirst "[name]=' " & NA & " ' "
If Data1.Recordset.NoMatch = True Then
   MsgBox "Try again", vbOKOnly, "Error Name"
Else
   DU = DU + 1
End If
End Sub
```

أسم حقل الأسم المعرف في قاعدة بيانات أكسس

Searching For Duplicate Text Field

```
Private Sub cmdDuplicate_Click()
Data1.Recordset.FindNext "[name]=' " & NA & " ' "
If Data1.Recordset.NoMatch = False Then
```

```
   DU = DU + 1
Else
   MsgBox "Duplicates of name are :" & DU, vbOKOnly, "Name Repeate"
End If
End Sub
```

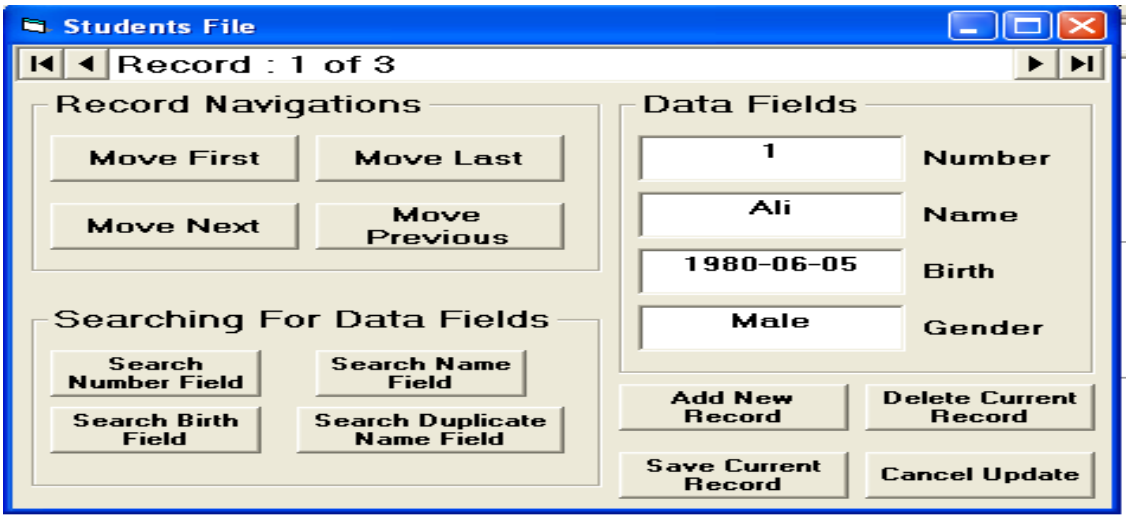
3 – 11 تحديد الكتابة الرقمية : Only Numeric Value

نقصُد بتحديد الكتابة الرقمية هو جعل الخانة الخاصة بحقل الرقم أو التسلسل والمُمثلة من خلال كائن صندوق البيانات (Text Box) أن يسمح لنا بأدخال قيم رقمية فقط داخل الصندوق وقفل باقي أزرار لوحة المفاتيح Keyboard .

البرنامج أدناه سيسمح لنا بأدخال أرقام فقط الى الصندوق الخاص بحقل الأرقام مع تفعيل مفتاح الحذف Delete فقط .

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If Chr(KeyAscii) Like "[0-9]" Or KeyAscii = 8 Then
Else
  KeyAscii = 0
End If
End Sub
```

وبعد تنفيذ البرنامج سيظهر لنا ملف قاعدة البيانات بالشكل رقم (3 – 1) .



شكل رقم (3 – 1) : تنفيذ برنامج ملف قاعدة بيانات الطلاب

3 – 12 كائن المُعطيات في قواعد البيانات : Visual Basic Database Object

ان كائن المُعطيات الرئيسي الذي يتعامل مع قواعد البيانات في الفيجوال بيسك برمجياً هو كائن مجموعة السجلات (Record Set Object) . وهو العنصر الذي يُحدد سجلات المعطيات المستخدمة في برامج الفيجوال بيسك . أن مجموعة السجلات أو القيود تكون فيها البيانات مرتبة على هيئة أعمدة وصفوف ، حيث تكون اما على شكل جدول كامل أو جزء من جدول (أستعلام) .

بشكل عام هناك أنواع مختلفة لكائن مجموعة السجلات (Record Set) ، ومن أهمها :

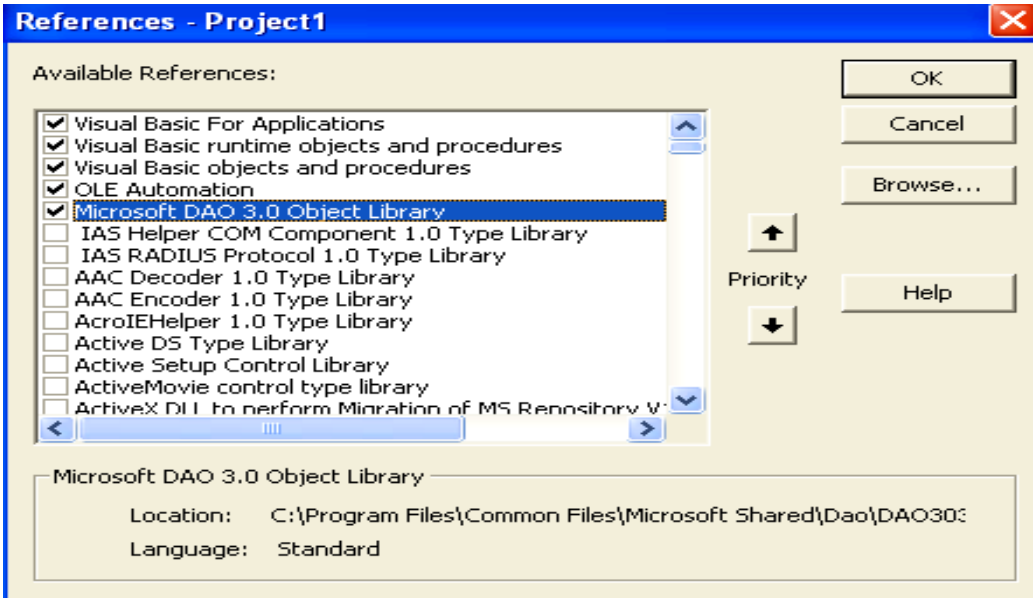
1. مجوعة سجلات من نمط مجموعة ديناميكية (Dynaset) .
2. مجموعة سجلات من نمط جدول (Table) .

ملاحظة : لغرض ربط برنامج لغة فيجوال بيسك مع قواعد البيانات برمجياً (أي من غير استخدام الكائن Data) ، نحتاج الى مايُسمى بمكتبة البرمجة وذلك عن طريق اختيار الأمر التالي :

Project ⟶ References ⟶ Microsoft DAO 3.51 Object Library

والذي يُستخدم للربط مع قواعد البيانات من نوع MS- Access 97 . لاحظ الشكل رقم (3 – 2) كما ويمكن الربط مع أنواع اخرى مثل Ms- Office XP من خلال اختيار الأمر التالي :

Project ⟶ References ⟶ Microsoft DAO 3.6 Object Library



شكل رقم (3 – 2) : تحميل مكتبة البرمجة DAO Object Library

3 – 12 – 1 كائن المعطيات Dynaset – Type Record Set Data Object

ان المجوعة الديناميكية (Dynaset) هي كائن المعطيات الأكثر شيوعاً وأستخداماً في الفيجوال بيسك ، وتُستخدم في عمليات البحث والحذف والتحديث والأضافة والأنتقال عبر سجلات القاعدة وغيرها من الوضائف الأخرى . هذه المجموعة تقوم بحجز جُزءاً من الذاكرة الحية (RAM) وأحياناً من القرص الصلب اذا كان حجم السجلات المطلوبة فيها أكبر من حجم الذاكرة الحية . وتعتبر من الكائنات الوحيدة القابلة للتحديث والمستخدمة في قواعد البيانات المترابطة عبر نموذج ODBC .

ماالمقصود بالـ DAO والـ ODBC ؟

يُعتبر كلاً من كائن الوصول للبيانات (Data Access Object (DAO و قواعد البيانات المترابطة (Open Database Connectivity (ODBC واجهات بينية تستخدم لبرمجة التطبيقات بلغة الفيجوال بيسك (Application Programming Interfaces (APIs والتي تمكننا من كتابة وتنفيذ برامج قواعد البيانات وبشكل مستقل عن نظام أدارة قواعد البيانات (DBMS) .

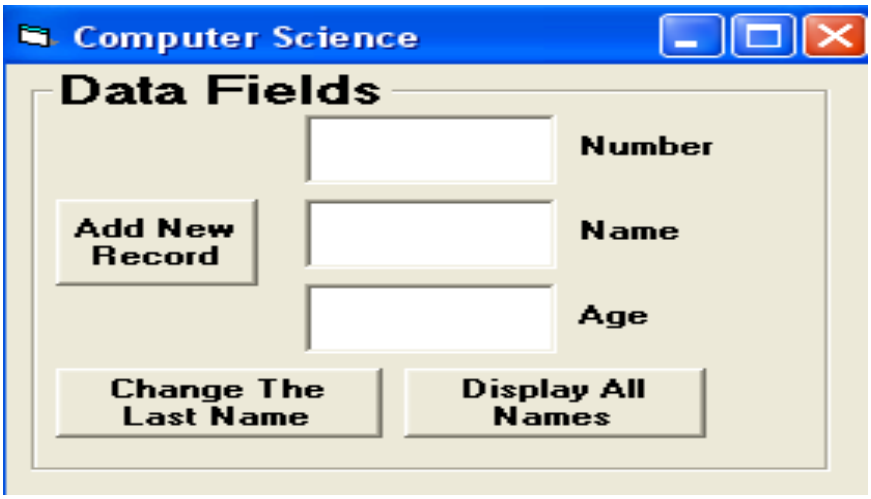
بشكل عام يقوم مُبرمجي قواعد البيانات بأستخدام كائن الوصول للبيانات DAO للربط بين قواعد البيانات المُصممة ببرنامج مايكروسوفت أكسس مع برنامج لغة فيجوال بيسك وذلك من خلال أستخدام مُحرك قواعد البيانات (Microsoft Jet database engine) والذي بدوره سيقوم بتجهيز مجموعة من كائنات الوصول للبيانات مثل database objects و tabledef and querydef object و recordset objects ، حيث أن كائن الوصول للبيانات DAO يعمل بشكل أفضل مع ملفات قواعد البيانات التي يتم تصميمها من خلال برنامج مايكروسوفت أكسس . كذلك يمكن الوصول الى مصادر قواعد البيانات المترابطة ODBC من خلال كائن الـ DAO ومحرك قواعد البيانات .

ملاحظة: عند تصميم برامج قواعد البيانات بأستخدام كائن مجموعة السجلات (Record Set Object) فان الكائن Data Control لن يُستخدم . في هذه الحالة نحتاج الى تعريف مُتغيرين وهما على التوالي :

Dim DB As Database ‘تحديد أسم قاعدة بيانات أكسس

Dim RS As Recordset ‘ تحديد أسم الجدول من القاعدة

البرنامج آدناه والموضح بالشكل رقم (3 – 3) يبين كيفية التعامل مع قاعدة البيانات من خلال كائن البرمجة (Record Set) من نوع Dynaset .



شكل رقم (3 – 3) : برمجة ملف قاعدة البيانات من خلال الكائن مجموعة السجلات Dynaset

أيعاز يُستخدم لتحميل الدليل Directory أو المسار Path الخاص بقاعدة بيانات أكسس والموجودة ضمن نفس المجلد Folder والذي يحوي كُلاً من برنامج لغة فيجوال بيسك و قاعدة بيانات أكسس .

أسم قاعدة بيانات أكسس

Add New Record :

```
Private Sub cmdAddUpdate_Click()
Set DB = DBEngine.OpenDatabase(App.Path + "\new.mdb")
Set RS = DB.OpenRecordset("info", dbOpenDynaset)
RS.AddNew
```

أسم الجدول في قاعدة بيانات أكسس

```
If Text1.Text = " " Or Text2.Text = " " Or Text3.Text = " " Then
    MsgBox "Try Again", vbCritical, "Fields Are Empty"
    Exit Sub
End If
RS.Fields("number") = Text1
RS.Fields("name") = Text2
RS.Fields("age") = Text3
RS.Update
MsgBox "Thank You", vbOKOnly, "Record Is Created"
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
RS.Close
End Sub
```

Change The Last Name :

```
Private Sub cmdAllUpdateEnd_Click()
On Error Resume Next
Set DB = DBEngine.OpenDatabase(App.Path + "\new.mdb")
Set RS = DB.OpenRecordset("info", dbOpenDynaset)
NA = InputBox("Enter The New Name", "Replace Name")
RS.MoveLast
RS.Edit
RS.Fields("name") = NA
RS.Update
RS.MoveFirst
```

```
Do Until RS.EOF = True
    Print RS.Fields("name")
    RS.MoveNext
Loop
RS.Close
End Sub
```

Display All Names :

```
Private Sub cmdAllNames_Click()
Set DB = DBEngine.OpenDatabase(App.Path + "\new.mdb")
Set RS = DB.OpenRecordset("info", dbOpenDynaset)
RS.MoveFirst
Cls
Do Until RS.EOF = True
    Print RS.Fields("name")
    Print
    RS.MoveNext
Loop
RS.Close
End Sub
```

3 – 12 – 2 كائن المعطيات The Table – Type Record Set Data Object

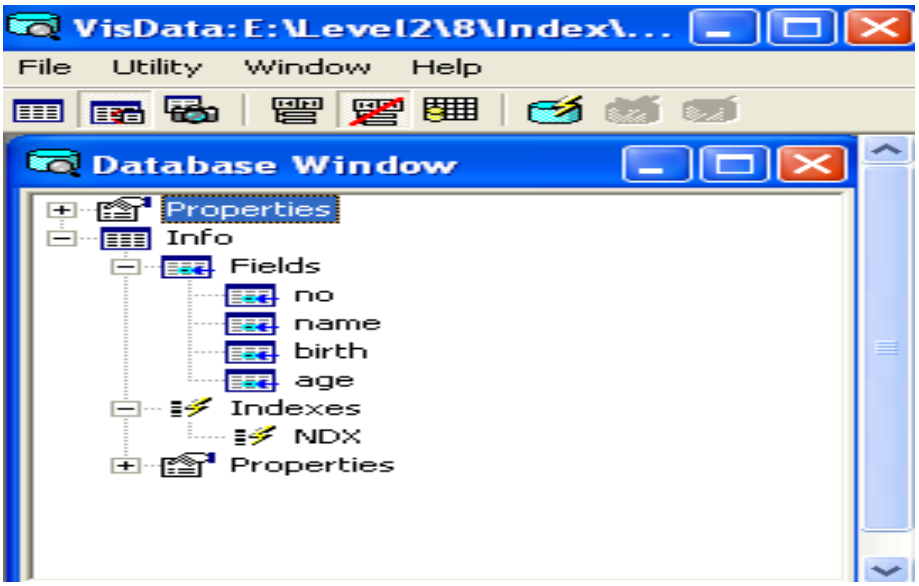
هذا النمط يستخدم ايعاز البحث (Seek) ويحتاج بذلك الى تعريف دلائل البحث والفهرسة (Index) حيثُ يُستخدم في عمليات البحث المُفهرس .

البرنامج التالي يقوم بقراءة والبحث عن البيانات الموجودة ضمن القاعدة وعرضها بأستخدام خاصية Index Field .

خطوات عمل البرنامج :

1. نقوم بتصميم قاعدة البيانات ومن ثم نُخصص لأحد حقول القاعدة دليل الفهرسة Index والذي من خلاله سيتم التعامُل مع القاعدة عند ربطها مع برنامج لغة فيجوال بيسك برمجياً .
2. يُمكن تصميم قاعدة البيانات من داخل الفيجوال بيسك وكما موضح بالشكل رقم (3 – 4) وذلك من خلال الأمر التالي :

Add-Ins ⟶ Visual Data Manager ⟶ File ⟶ New ⟶ Microsoft Access ⟶ Version 7.0 MDB

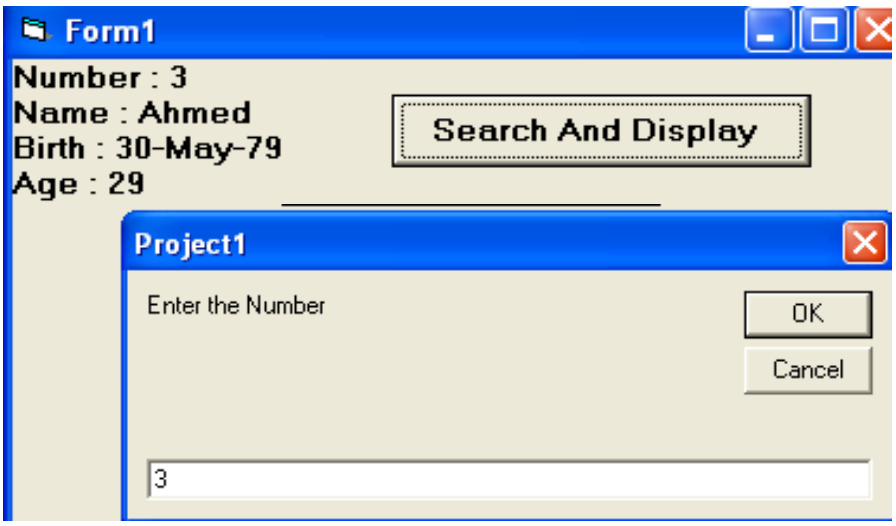


شكل رقم (3 – 4) : تصميم قاعدة البيانات من داخل الفيجوال بيسك

سنقوم الأن بكتابة البرنامج التالي وذلك بأستخدام كائن البرمجة مجموعة السجلات من نوع Table والذي سيظهر عند التنفيذ بالشكل رقم (3 – 5) .

```
Dim N As Integer
Dim DB As Database
Dim RS As Recordset
```

```
Private Sub cmdSearch_Click()
Cls
On Error GoTo error
Set DB = DBEngine.OpenDatabase(App.Path + "\new.mdb")
Set RS = DB.OpenRecordset("info", dbOpenTable)
N = InputBox("Enter the Number")
error:
If Err.Number = 13 Then
Exit Sub
End If
RS.Index = "ndx"   ' تخصيص دليل الفهرسة على حقل الرقم
RS.Seek "=", N
If RS.NoMatch = False Then
   Print "Number : " & RS.Fields("no")
   Print "Name : " & RS.Fields("Name")
   Print "Birth : " & RS.Fields("Birth")
   Print "Age : " & RS.Fields("Age")
End If
End Sub
```



شكل رقم (3 – 5) : برمجة ملف قاعدة البيانات من خلال الكائن مجموعة السجلات Table

3 – 13 كائن البيانات ADO Data Control

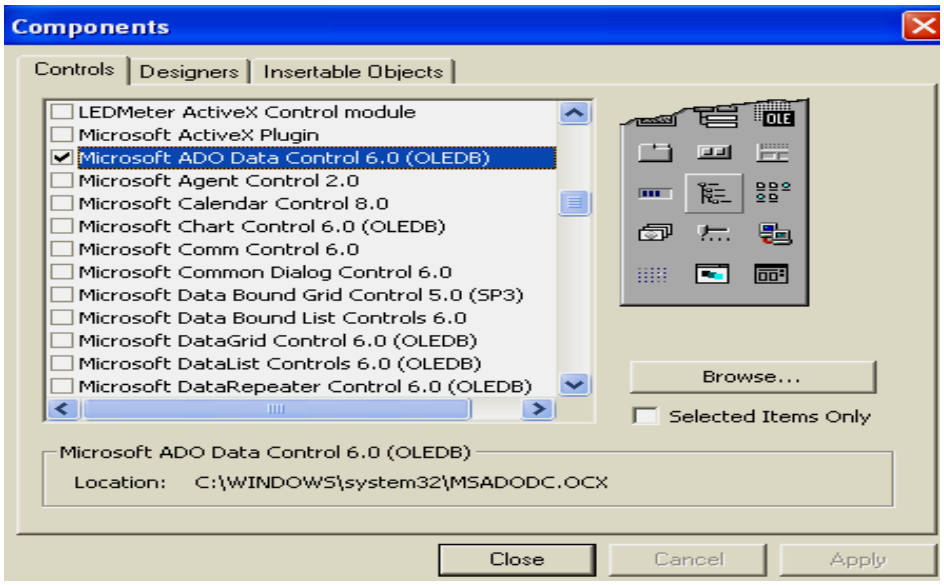
يشبه هذا الكائن عنصر التحكم المرتبط بالأدخال Data Control ، حيث يساعدنا في الربط المباشر والسريع مع قواعد البيانات وفتح القاعدة ومن ثم اجراء كافة العمليات المطلوبة من (اضافة قيد ، حذف قيد ، تحديث قيد ، الأنتقال من قيد لأخر) وذلك من خلال مجهز قواعد البيانات OLE DB provider ، كذلك امكانية عمل تقرير Report لتوضيح خلاصة العمل ، وغيرها من الأمكانيات الأخرى .

يتميز كائن الـ (ADO) Microsoft ActiveX Data Object بسهولة أستخدامه وسرعته في الربط مع قواعد البيانات وحجزه مساحة صغيرة من الذاكرة مع الأمكانيات التي يوفرها في بناء التطبيقات الخاصة بالشبكات client / server and web-based application ، حيث يقوم بخدمة البيانات عن بعد (RDS) Remote Data Service من خلال نقل البيانات من الـ Server الى الـ Client أو عبر صفحة الشبكة Web page ليتم معالجة البيانات وأجراء العمليات المطلوبة عليها update من قبل الـ Client ثم أرجاعها الى الـ Server ضمن دورة واحدة

Connect the ADO Data Control to a data-bound control such as the Data Grid, Data Combo, or Data List control by setting the Data Source property to the ADO Data Control.

يتم اضافة كائن (ADO) الى المشروع والموضح بالشكل رقم (3 – 6) من خلال اختيار الأمر التالي :

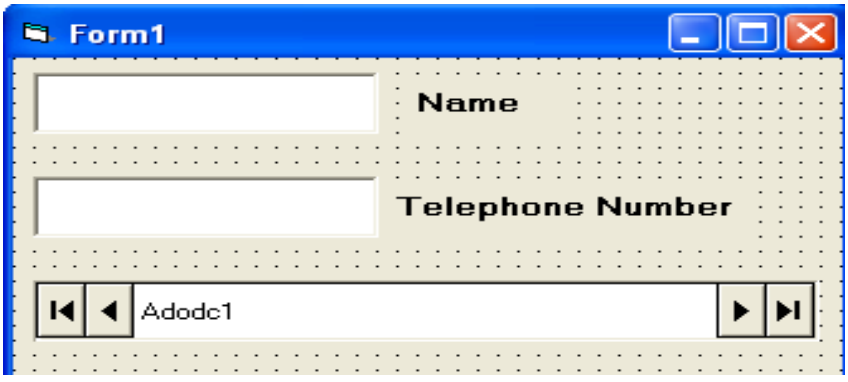
Project ⟶ Components ⟶ Microsoft ADO Data Control 6.0 (OLEDB)



شكل رقم (3 – 6) : اضافة كائن الوصول للبيانات ADO

بعد الضغط على الأمر Apply سيتم اضافة الكائن ADO الى النموذج حيث سيظهر لنا الكائن في صندوق الأدوات بهذا الشكل وبأمكاننا استخدامه في الربط مع قاعدة بيانات أكسس وبطريقتين :

الطريقة الأولى : يتم اضافة الكائن ADO الى النموذج وكذلك اضافة مربعين نص TextBox وصندوقين للعنوان Label كما في الشكل رقم (3 – 7) .



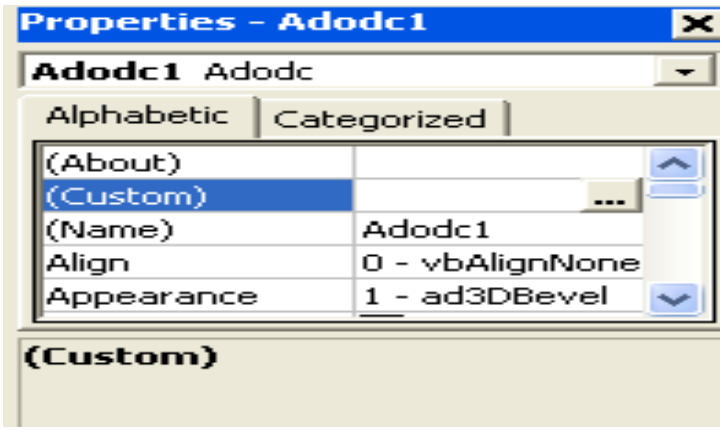
شكل رقم (3 – 7) : تصميم برنامج قاعدة البيانات بأستخدام الكائن ADO

نقوم بتغيير خصائص الكائنات (الأدوات) التي تم اضافتها الى النموذج وكما موضح بالجدول رقم (3 – 2) .

جدول رقم (3 – 2) : تغيير خصائص الكائنات .

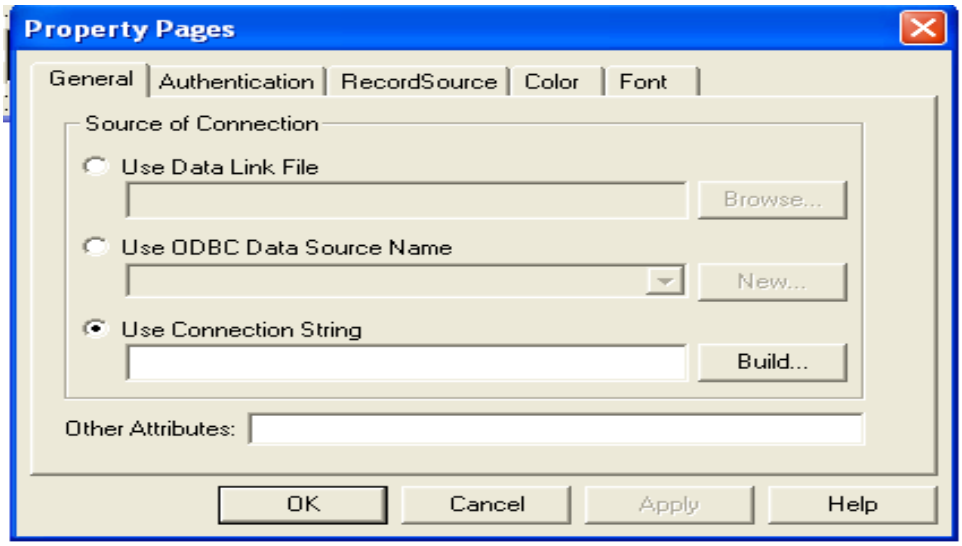
الأداة	الخاصية	القيمة
Label1	Caption	Name
Label1	Font	Bold- Size 10
Label2	Caption	Telephone Number
Label2	Font	Bold- Size 10
Text1	Text	فارغ
Text1	Alignment	Right justify
Text2	Text	فارغ
Text2	Alignment	Right justify

نختار الأداة ADO ونختار من صندوق الخصائص الخاصية Custom كما بالشكل رقم (3 – 8).



شكل رقم (3 – 8) : خصائص كائن البيانات ADO

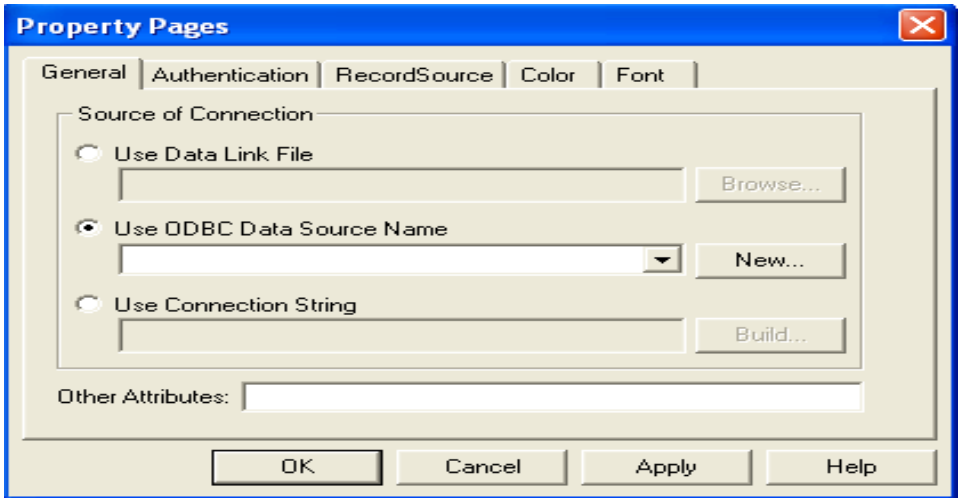
ستظهر لنا صفحة الخصائص الموضحة بالشكل رقم (3 – 9) ، أو من خلال نقر زر الفأرة الأيمن على الكائن ADO وأختيار الأمر ADODC Properties .



شكل رقم (3 – 9) : صفحة خصائص كائن البيانات ADO

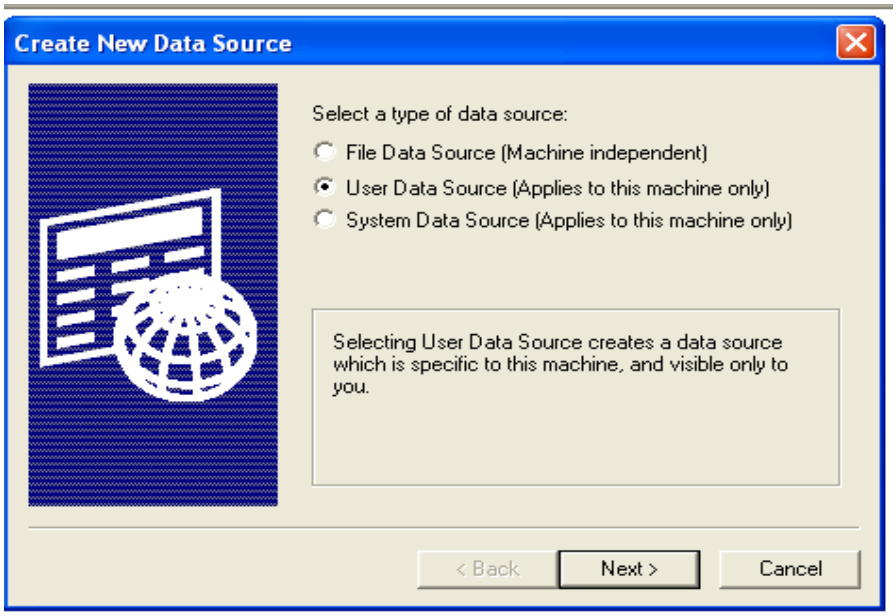
سنقوم الأن بربط قاعدة بيانات أكسس بالفيجوال بيسك عن طريق الأداة ADO ومن ثم ربط حقول القاعدة مع الفيجوال بيسك ايظاً من خلال الأداة ADO ، وكالأتـي :

نختار من نافذة Property Pages الأمر الثاني Use ODBC Connection String ، ومن ثم الضغط على كلمة New التي بجوارها كما بالشكل رقم (3 – 10).



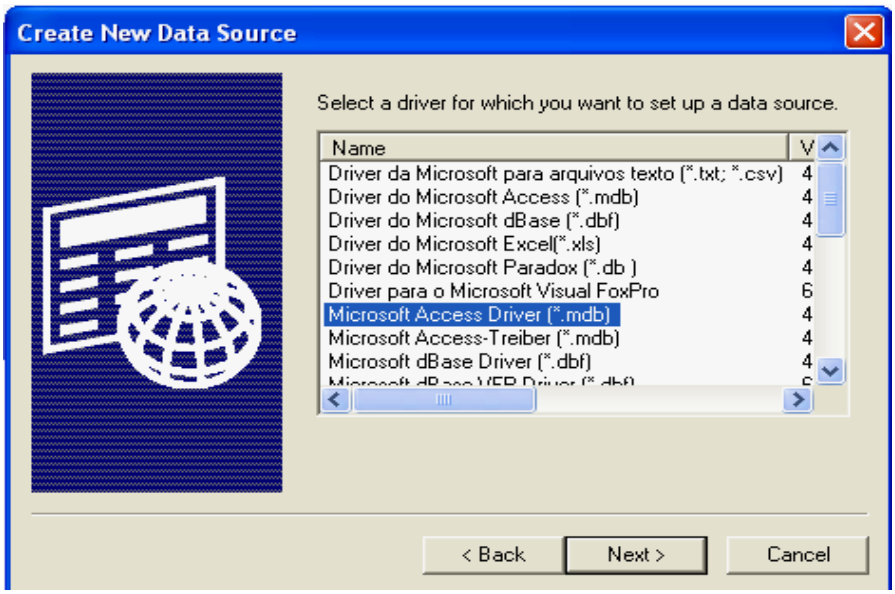
شكل رقم (3 – 10) : ربط برنامج الفيجوال بيسك بقاعدة بيانات أكسس بأستخدام كائن بيانات ADO

ستظهر لنا نافذة أخرى نختار منها الأختيار الثاني والذي من خلاله سيتم تحديد نوع مصدر البيانات وكما موضح بالشكل رقم (3 – 11) والضغط على الأمر Next .



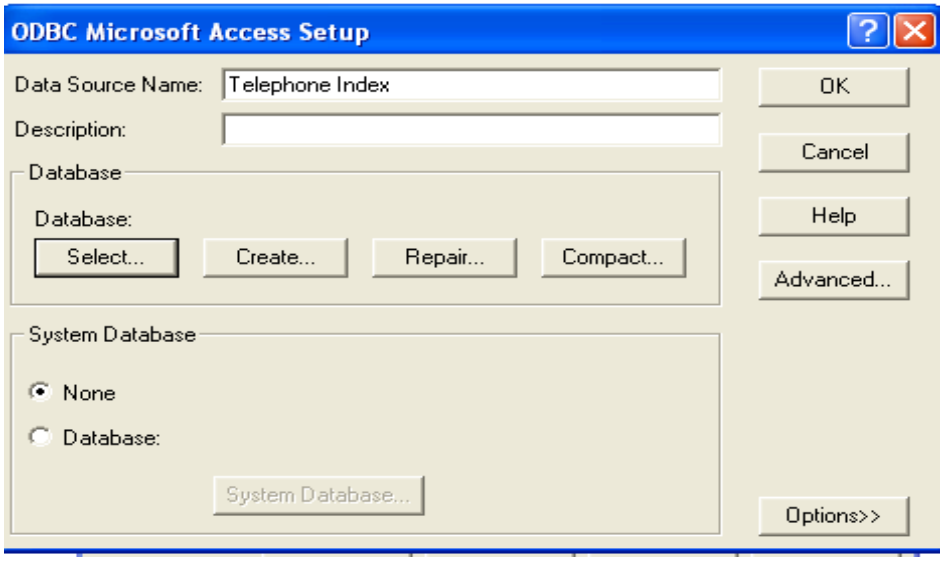
شكل رقم (3 – 11) : تحديد نوع مصدر البيانات

ستظهر لنا نافذة أخرى لأختيار نوع قاعدة البيانات كما في الشكل رقم (3 – 12) نختار منها (*.mdb) Microsoft Access Driver .



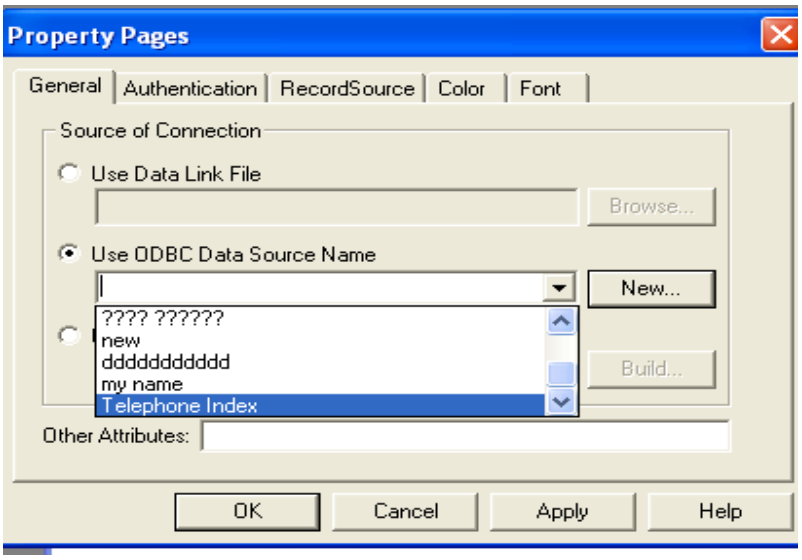
شكل رقم (3 – 12) : تحديد قاعدة البيانات من نوع مايكروسوفت أكسس

نضغط Next ستظهر نافذة أخرى ، و نضغط على الزر Finish . ستظهر لنا نافذة أخرى كما في الشكل (3 – 13) يتم فيها تحديد مكان قاعدة البيانات على الجهاز مع أعطاء أسم لمصدر بيانات القاعدة .



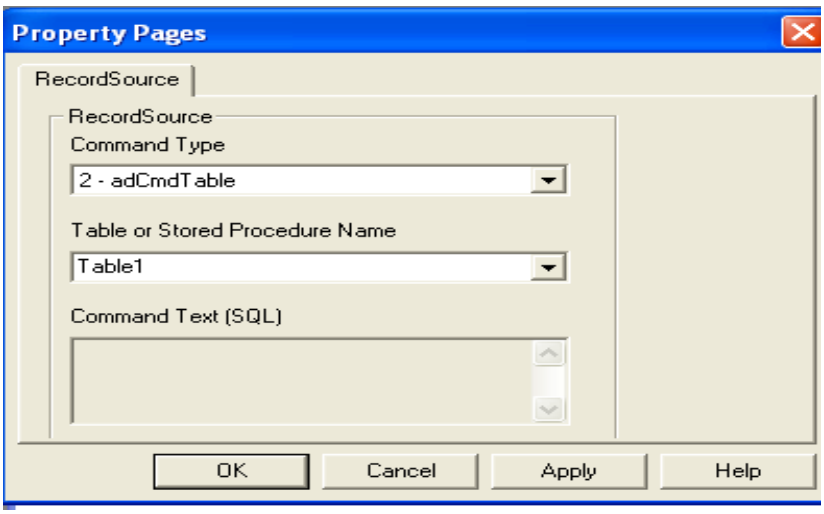
شكل رقم (3 – 13) : تحديد مكان قاعدة بيانات أكسس وأسم مصدر بياناتها

في خانة الـ Data Source Name نكتب العبارة Telephone Index (عبارة أختيارية) ، ثم الضغط على الأمر Select الموضح بالشكل رقم (3 – 13) وأختيار قاعدة البيانات المصصمة مسبقاً والتي تحتوي على جدول مكون من حقلين (الأسم ، رقم الهاتف) ثم الضغط على الأمر موافق . بعد ذلك سنعود الى نافذة Custom الرئيسية ، نقوم بأختيار العبارة " Telephone Index " من القائمة المنسدلة الموجودة في الأختيار الثاني ، وكما في الشكل رقم (3 – 14) ، حيث سيتم أختيار أسم مصدر البيانات والمحدد بالعبارة الأختيارية " Telephone Index "، وأخيراً يتم الضغط على الأمر موافق OK .



شكل رقم (3 – 14) : أختيار أسم مصدر بيانات القاعدة

الأن قمنا بربط قاعدة بيانات أكسس بالفيجوال بيسك عن طريق الأداة ADO . يأتي الأن الدور لتحديد الجدول الذي سنقوم بأحضار البيانات منه ، وذلك بتغيير الخاصية Record Source للأداة ADO والتي ستظهر بالشكل رقم (3 – 15) حيث سيتم أختيار أسم الجدول المصمم مُسبقاً ضمن قاعدة البيانات .



شكل رقم (3 – 15) : تحديد جدول قاعدة البيانات

بعد ذلك يأتي دور صناديق النصوص والموضحة بالجدول رقم (3 – 3)، حيث سنجعلها تقوم بعرض بيانات القاعدة وذلك من خلال تغيير بعض خصائصها .

جدول رقم (3 – 3) : تغيير خصائص صناديق النص .

الأداة	الخاصية	القيمة
Text1	Data Source	Adodc1
Text1	Data Field	Name
Text2	Data Source	Adodc1
Text2	Data Field	Telephone Number

الى الأن لم يتم كتابة سطر برمجي واحد ولكن مع ذلك قمنا بعمل برنامج وربطناه بقاعدة البيانات . نقوم أخيراً بتنفيذ البرنامج ونختبره حيث نستطيع تحريك الأسهم الموجودة على الأداة Adodc1 لكي نبدل بين السجلات .

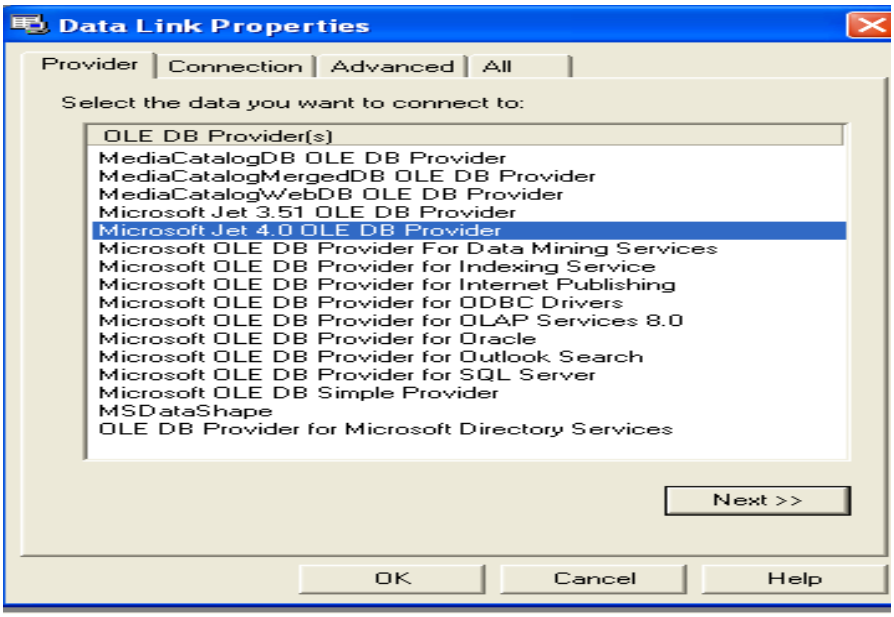
ملاحظة : بأستخدام الأداة ADO نستطيع ربط الفيجوال بيسك مع قاعدة بيانات Access 2003 ، اضافة الى قاعدة بيانات Access 97 وهذا مايميزها عن الأداة Data Control .

الطريقة الثانية : يتم اضافة الكائن ADO الى النموذج وعرض حقول جدول القاعدة مع بياناتها دون الحاجة لأستخدام كائن صندوق النص ، حيث سيتم أستخدام الكائنData Grid Control والذي سيظهر في صندوق الأدوات Tool Box وذلك من خلال أختيار الأمر التالي :

Project ⟶ Components ⟶ Microsoft Data Grid Control 6.0

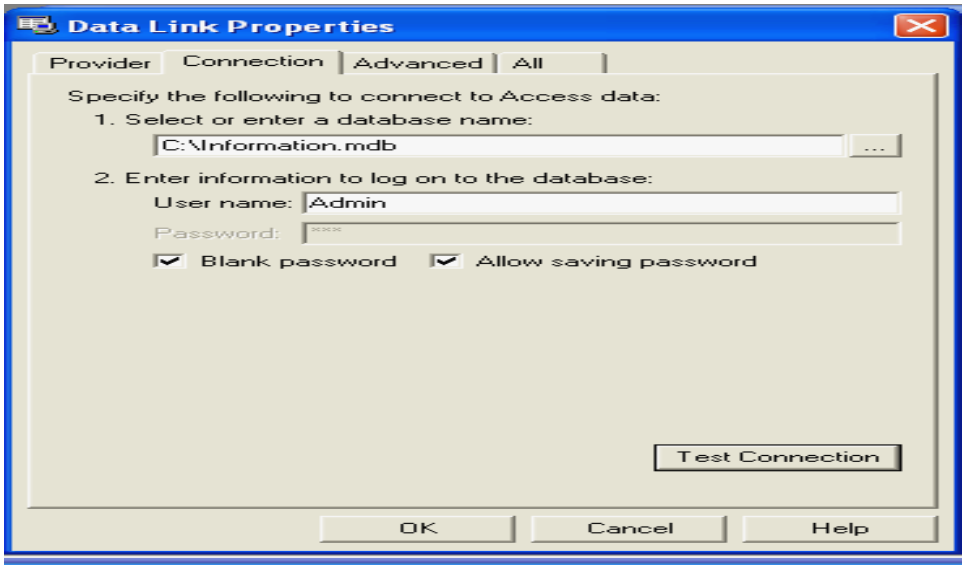
بعد ذلك نقوم بأختيار الكائن ADO ونختار من صندوق الخائص الخاصية Custom فتظهر لنا نافذة Property Pages الموضحة بالشكل رقم (3 – 9) ، نختار منها الأمر الثالث Use Connection String ثم نضغط فوق الأمر Build فتظهر لنا نافذة خصائص الربط

الموضحة بالشكل رقم (3 – 16) حيث نختار منها الخاصية Microsoft jet 4.0 OLE DB Provider .



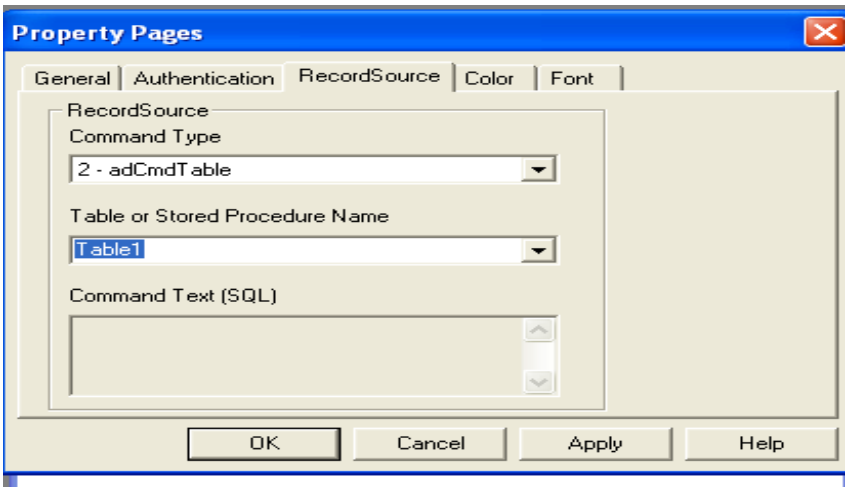
شكل رقم (3 – 16) : نافذة خصائص ربط البيانات

ثم نضغط Next ونقوم بربط الكائن ADO مع القاعدة المصممة مسبقاً من خلال أختيار الأمر Connection وكما موضح بالشكل رقم (3 – 17) ، مع امكانية اختبار الربط من خلال الأمر Test Connection .



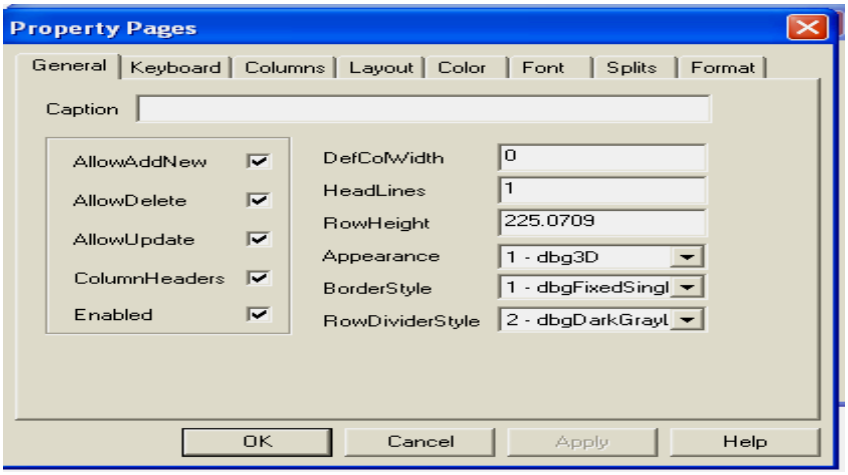
شكل رقم (3 – 17) : ربط كائن البيانات ADO مع القاعدة وأختبار الربط

بعد ذلك نقوم بربط الكائن مع الجدول الموجود في قاعدة البيانات كما في الشكل رقم (3 – 18) وذلك من خلال أختيار الأمر Record Source ومن ثم ضغط موافق . بهذه الحالة قد تم ربط الكائن ADO مع القاعدة .



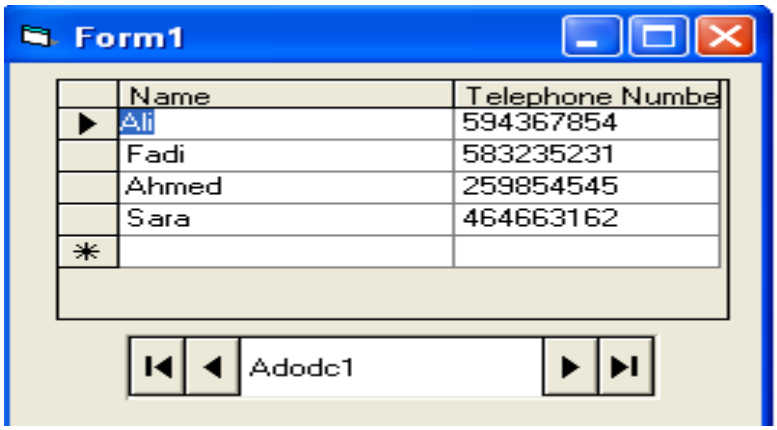
شكل رقم (3 – 18) : ربط كائن البيانات ADO بجدول القاعدة من خلال مصدر البيانات

ولغرض ربط الكائن Data Grid مع الأداة ADO يتم من خلال خاصية Data Source ومن ثم نقر زر الفأرة الأيمن على كائن الشبكة وأختيار الأمر Retrieve Fields بذلك سيتم عرض الحقول وبياناتها مع أمكانية أجراء عمليات الأضافة (أضاقة قيد جديد) والحذف والتحديث ، وكما موضح ذلك بالشكل رقم (3 – 19) .



شكل رقم (3 – 19) : أجراء العمليات المختلفة على بيانات القاعدة من خلال كائن Data Grid

وبعد التنفيذ سيظهر لنا البرنامج بالشكل رقم (3 – 20) .



شكل رقم (3 – 20) : ربط كائن البيانات ADO بالقاعدة من خلال كائن Data Grid

3 – 14 التقارير في قواعد البيانات : Creating Reports in Databases

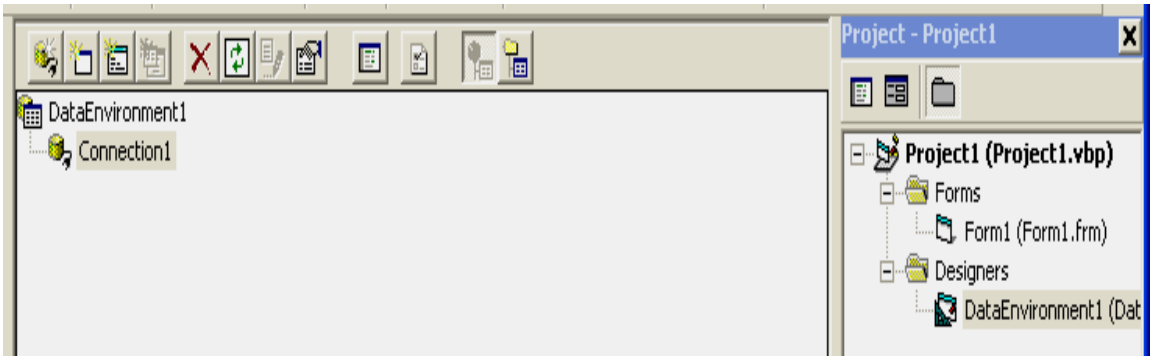
بعد الأنتهاء من ربط قاعدة بيانات أكسس ببرنامج لغة فيجوال بيسك نستطيع عمل ورقة لتلخيص البيانات مع أمكانية تنسيقها بما يتلائم وحسب رغبة المصمم وطباعتها ، وهذا مانطلق عليه أسم " التقرير " .

ولغرض عمل التقرير نتبع الخطوات التالية :

1. من شريط القوائم والذي يظهر في برنامج لغة فيجوال بيسك نختار الأمر التالي :

Project ⟶ Add Data Environment

سيظهر لنا الشكل رقم (3 – 21) .

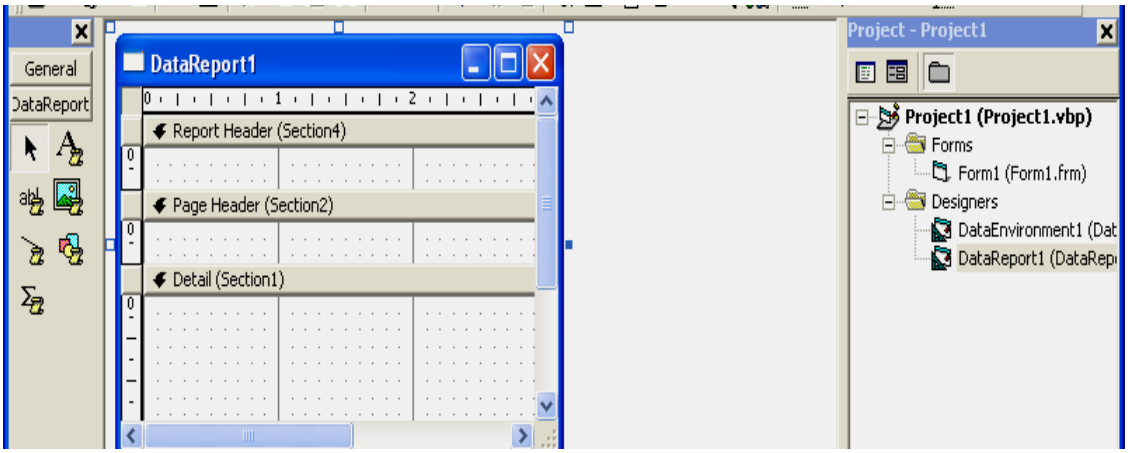


شكل رقم (3 – 21) : اضافة بيئة البيانات Data Environment الى المشروع

2. من شريط القوائم نختار الأمر التالي :

Project ⟶ Add Data Report

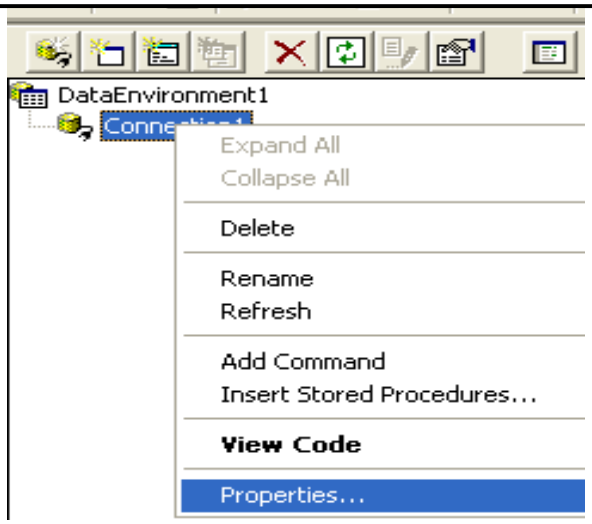
سيظهر لنا الشكل رقم (3 – 22) .

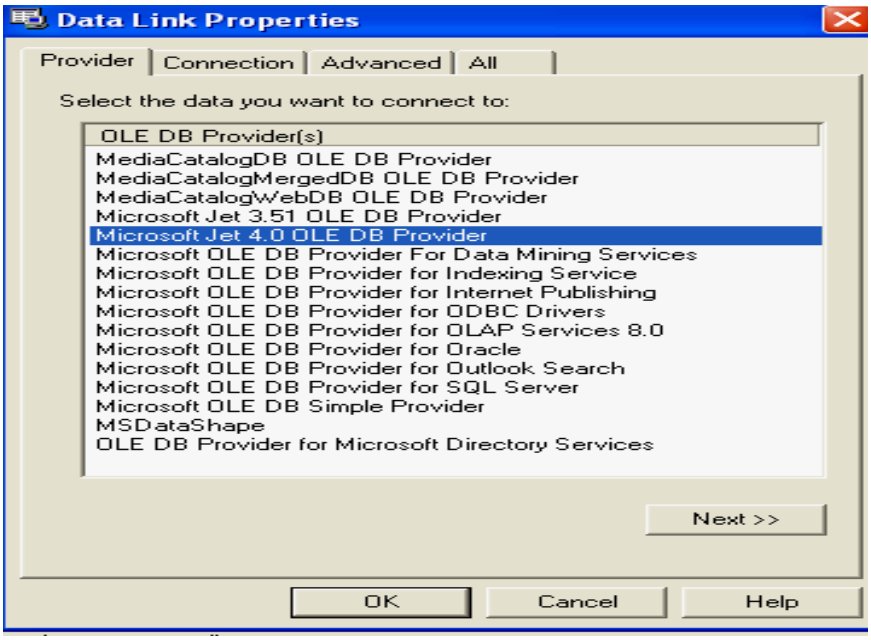


شكل رقم (3 – 22) : اضافة تقرير لبيانات القاعدة الى المشروع

3. من الأمر Connection والذي يظهر في نافذة Data Environment ننقر زر الفأرة الأيمن ونختار الأمر Properties ، ثم نحدد نوع خاصية الربط ونضغط Next وكما موضح بالشكل رقم (3 – 23) .

Connection1 ⟶ Properties ⟶ Microsoft jet 4.0 OLE DB Provider



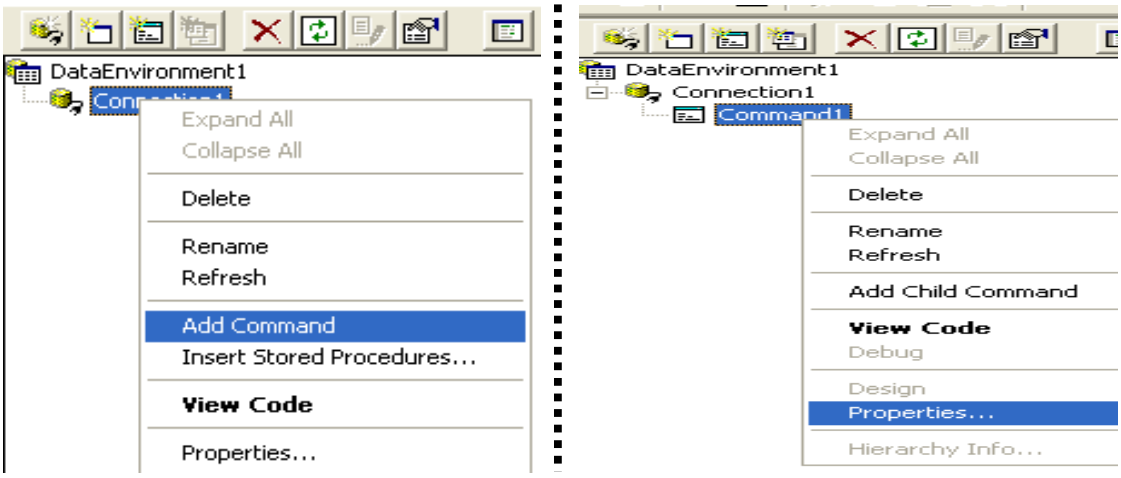


شكل رقم (3 – 23) : تحديد نوع خاصية ربط التقرير

4. تحديد قاعدة بيانات أكسس المصممة مسبقاً والتي سنربطها مع الفيجوال بيسك كما في الشكل رقم (3 – 24) مع أمكانية أختبار عملية الربط مع القاعدة .

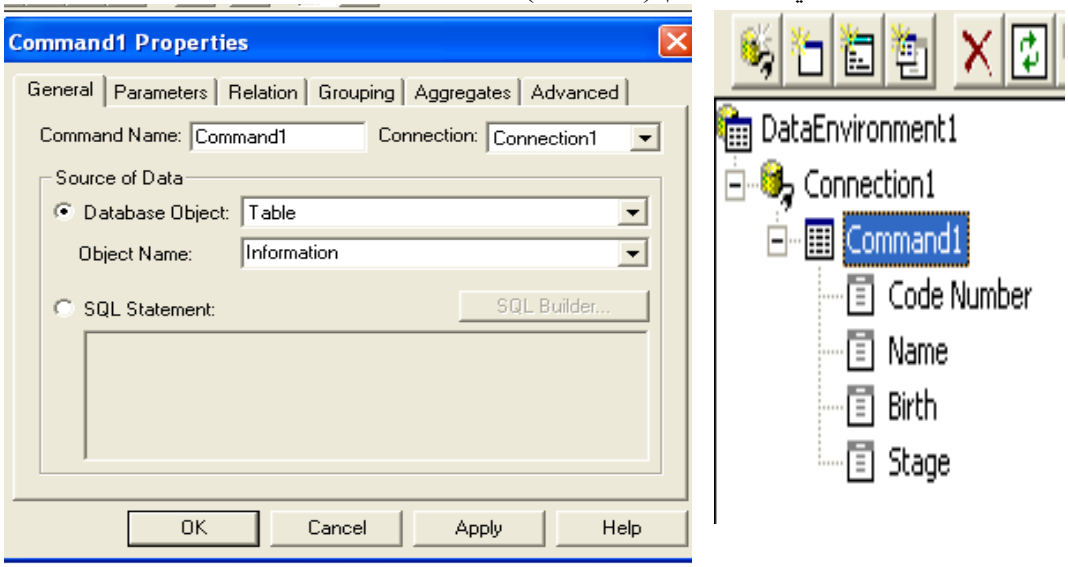
شكل رقم (3 – 24) : تحديد أسم قاعدة البيانات المصممة مسبقاً

5. نقوم بعمل زر أمر Command Button والذي من خلاله سنقوم بعرض التقرير ، لذلك لابد من ربطه مع بيانات القاعدة وذلك من خلال نقر زر الفأرة الأيمن على الأمر Connection1 ونختار الأمر Add Command ، ثم ننقر زر الفأرة الأيمن على الأداة Command1 ونختار الأمر Properties ، وكما موضح ذلك بالشكل رقم (3 – 25) .



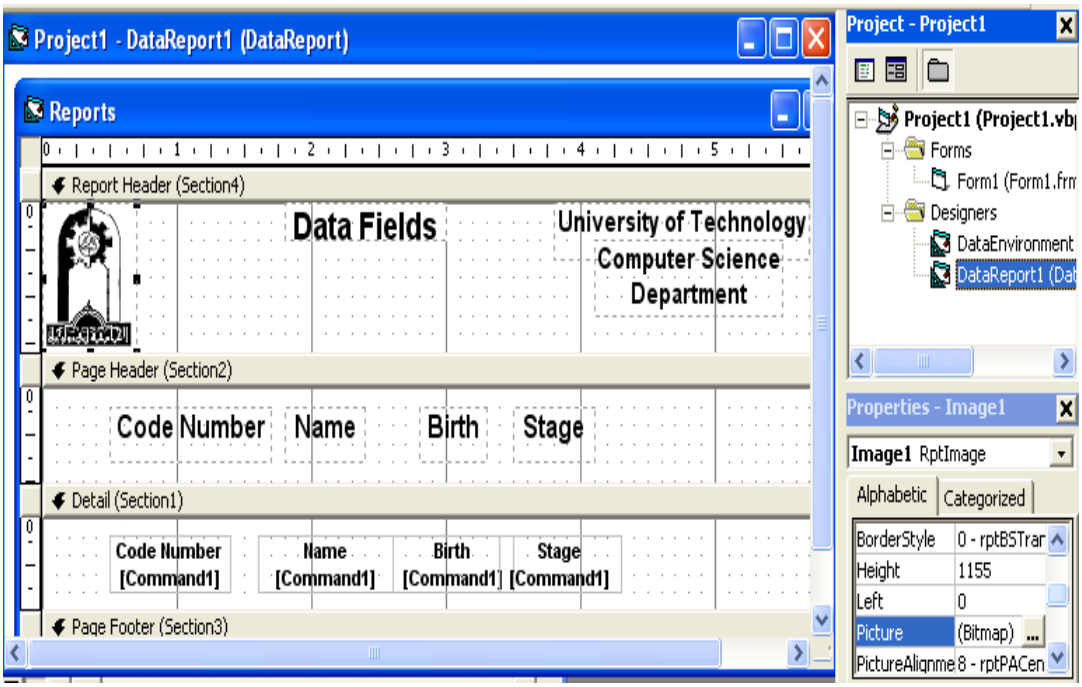
شكل رقم (3 – 25) : عرض التقرير من خلال كائن زر الأمر

6. نقوم بتحديد الجدول الذي سنعرض بياناته من خلال الأمر Database Object والأمر Object Name وأخيراً نضغط موافق . سنلاحظ أن الأداة Command1 مرتبطة مع حقول الجدول كما في الشكل رقم (3 – 26) .



شكل رقم (3 – 26) : ربط كائن زر الأمر مع حقول الجدول

7. من نافذة الـ Data Report1 نقوم بتصميم واجهة عرض التقرير كما في الشكل رقم (3 – 27) من خلال أضافة عناوين Labels (عناوين أو اسم الحقول المستخدمة في جدول البيانات) وذلك في صفحة Page Header مع أمكانية أضافة صور ورسوم واجراء التنسيقات المطلوبة عليها من خلال نافذة Properties . وفي صفحة Detail نقوم بأضافة أربعة صناديق نص Text Box والتي سيتم فيها عرض بيانات الحقول .



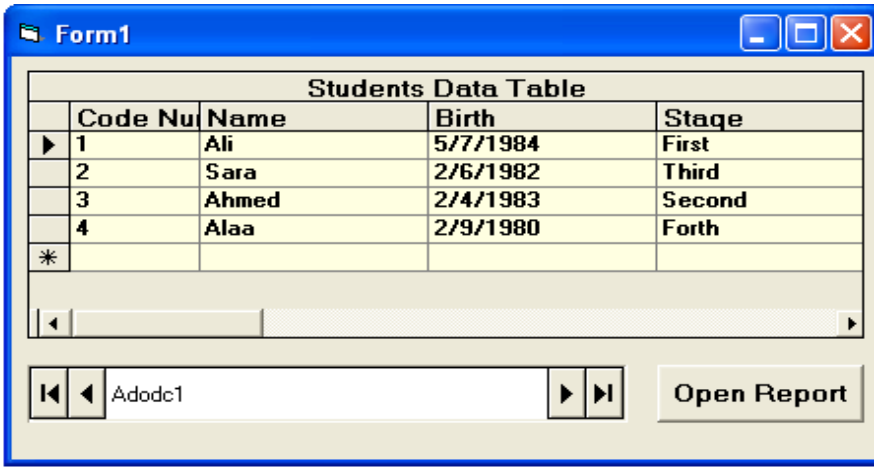
شكل رقم (3 – 27) : تصميم واجهة عرض التقرير

8. نقوم بربط الـ Data Report مع الـ Data Environment من خلال نافذة خصائص الـ Data Report وأختيار الخاصية Data Source .
9. نقوم بربط الـ Data Report مع الـ Command من خلال نافذة خصائص الـ Data Report وأختيار الخاصية Data Member .
10. نقوم بربط كائنات صندوق النص Text Box مع الـ Command من خلال نافذة خصائص الـ Data Report وأختيار الخاصية Data Member .
11. نقوم بربط كل كائن على حدة مع حقول القاعدة من خلال خاصية الـ Data Field والتي تظهر في نافذة خصائص الـ Data Report .

12.أخيراً نقوم باضافة زر أمر Command Button الى النموذج ونكتب بداخله الشفرة التالية (من نافذة الشفرة الخاصة بالكائن Code Window) :

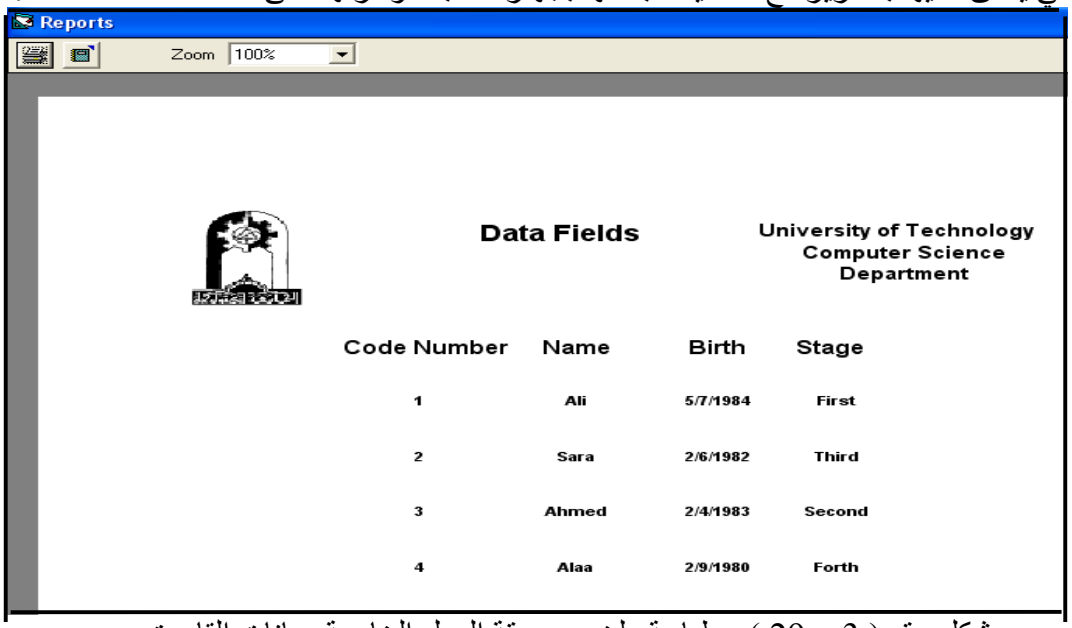
```
Private Sub Command1_Click()
DataReport1.Show
End Sub
```

عند التنفيذ سيظهر لنا البرنامج بالشكل رقم (3 – 28) .



شكل رقم (3 – 28) : تنفيذ برنامج قاعدة البيانات مع عرض تلخيص لورقة العمل

وعند النقر على الأمر Open Report ستظهر لنا ورقة تحتوي على ملخص لكافة الحقول وبياناتها والتي يُطلق عليها بالتقرير مع أمكانية طباعتها بجهاز الطابعة وخزنها على شكل ملف HTML .



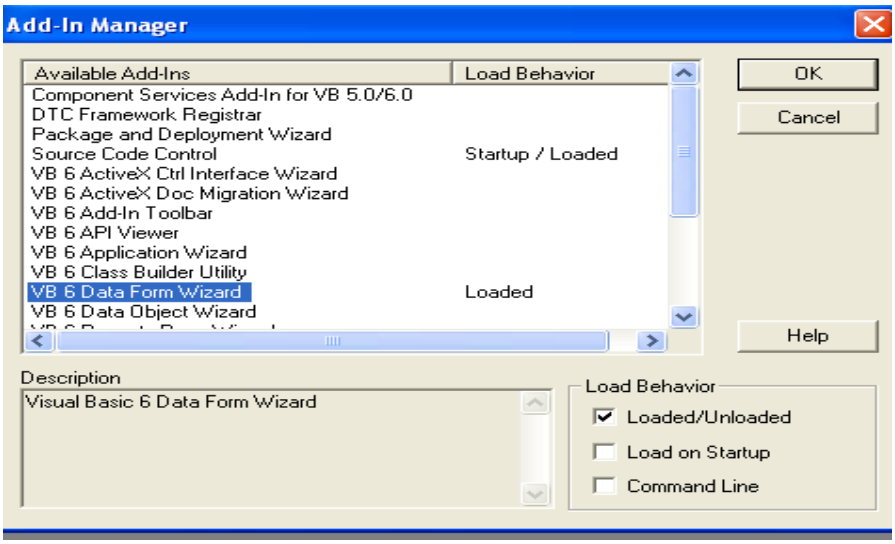
شكل رقم (3 – 29) : طباعة ملخص ورقة العمل الخاصة ببيانات القاعدة

3 – 15 قاعدة البيانات بأستخدام المعالج : Database Using Wizard

يستخدم المعالج Wizard في تصميم برامج قواعد البيانات بدون كتابة أي شفرة برمجية (Code) ، ولعمل ذلك نتبع الخطوات التالية :

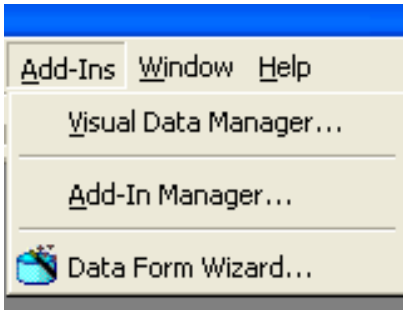
ملاحظة : قاعدة البيانات بأستخدام المعالج يجب أن تكون من نوع MS-Access 97 .

1. نقوم بفتح مشروع جديد في الفيجوال بيسك ومن القائمة Add-Ins نختار الأمر Add-In Manager .
2. تظهر لنا قائمة نختار منها VB6 Data Form Wizard نضغط عليها مرتين حتى تظهر بجوارها كلمة Loaded كما في الشكل رقم (3 – 30) ، ثم نضغط موافق OK .



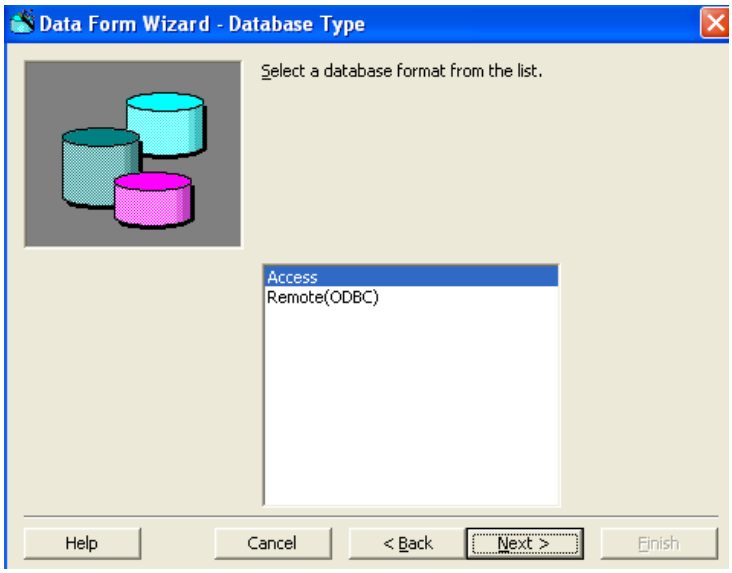
شكل رقم (3 – 30) : تكوين قاعدة البيانات بأستخدام المعالج

3. نفتح القائمة Add-Ins مرة ثانية سنلاحظ أنه قد تم اضافة Data Form Wizard الى القائمة كما في الشكل رقم (3 – 31)، نقوم بأختيارها عندها ستظهر لنا نافذة نضغط على الزر Next .



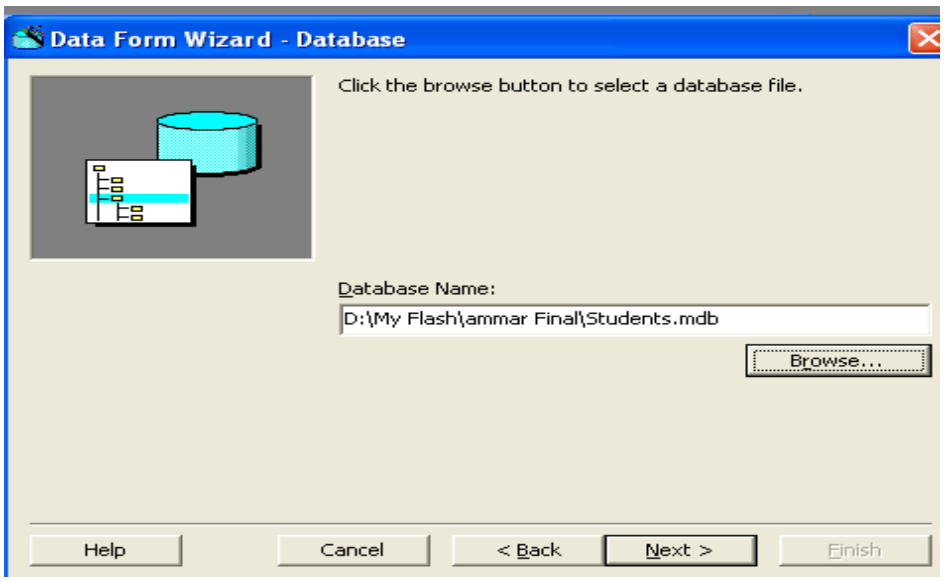
شكل رقم (3 – 31) : معالج نموذج البيانات

4. ستظهر لنا نافذة جديدة تطلب منا تحديد نوع الربط والموضحة بالشكل رقم (3 – 32) نختار Access .



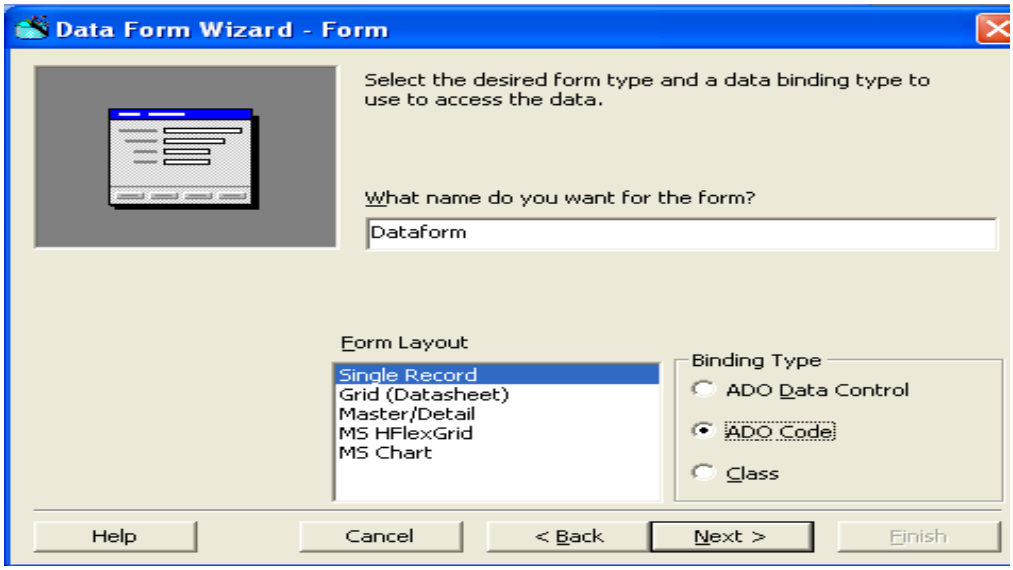
شكل رقم (3 – 32) : تحديد نوع الربط مع قاعدة البيانات

5. بعد الضغط على الأمر Next ستظهر لنا نافذة تطلب منا تحديد أسم القاعدة التي نريدها ومكانها ، حيث سنقوم بتحديد مكان القاعدة ونضغط مرة أخرى Next كما في الشكل رقم (3 – 33) .



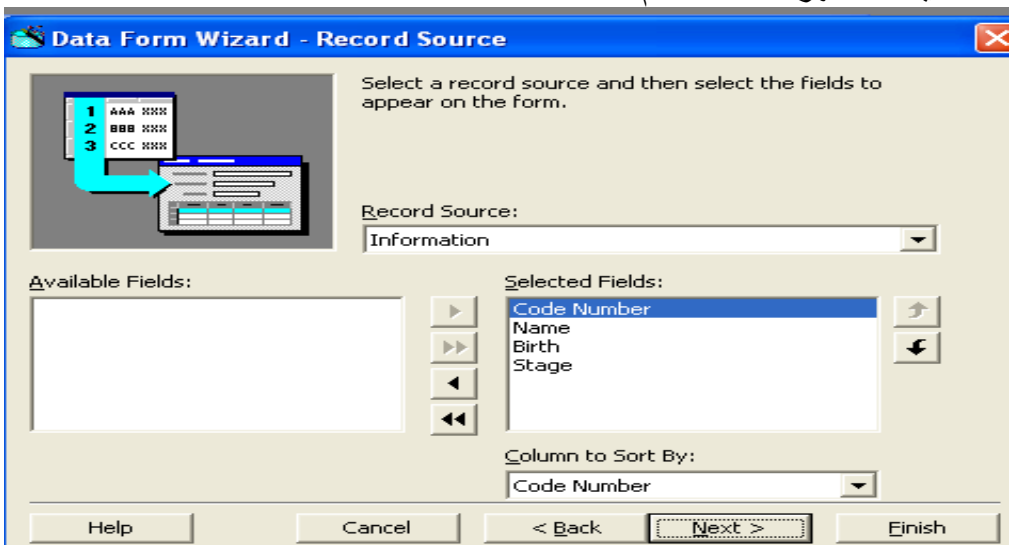
شكل رقم (3 – 33) : تحديد أسم قاعدة بيانات أكسس ومسارها

6. تظهر لنا نافذة جديدة والظاهرة بالشكل رقم (3 – 34) تطلب منا تحديد أسم النموذج Form الذي سنجعل فيه البرنامج ، حيث سنقوم بكتابة الأسم الذي نريده ثم نحدد النوع بـ ADO Code ، بعدها نضغط على الأمر Next .



شكل رقم (3 – 34) : تحديد أسم نموذج قاعدة البيانات بأستخدام المعالج

7. تظهر نافذة تطلب منا تحديد أسم الجدول الذي سنستخدمه (في حالة كون القاعدة تحتوي على أكثر من جدول) كما في الشكل رقم (3 – 35). بعد أختيار الجدول ستظهر على اليسار قائمة بالحقول الموجودة في الجدول حيث سنقوم بأختيار الحقول التي نريدها مع أمكانية عمل فهرسة للحقل ، ثم نضغط Next .



شكل رقم (3 – 35) : أختيار أسم جدول قاعدة البيانات والحقول المطلوبة

8. نقوم بأختيار جميع الأزار التي ستظهر في النموذج Select All كما في الشكل (3 – 36) ثم ضغط Next وأخيراً Finish وبذلك فقد قمنا بعمل برنامج بدون أن نقوم بكتابة أي شفرة.

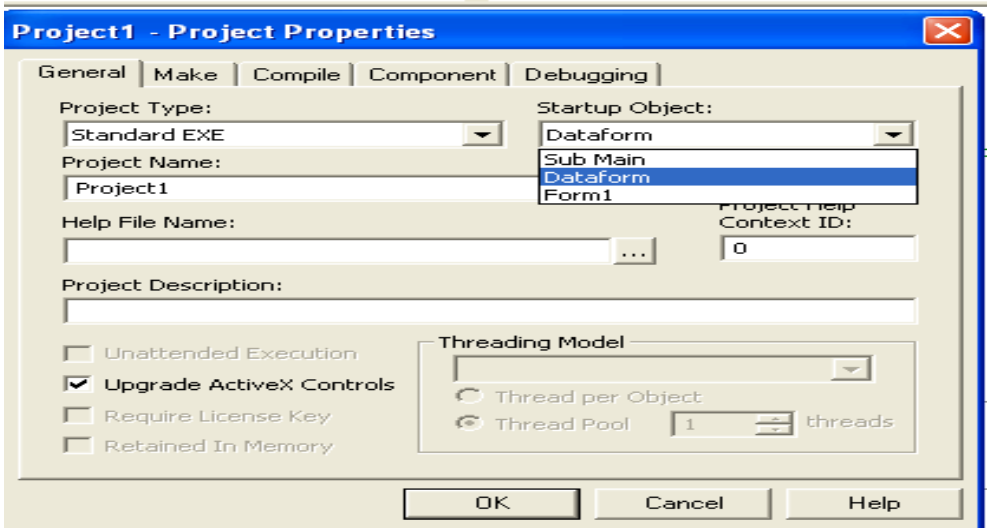
شكل رقم (3 – 36) : تحميل أزرار التحكم الخاصة بقاعدة البيانات بأستخدام المعالج

9. أخيراً سيظهر لنا البرنامج بعد التنفيذ بالشكل رقم (3 – 37) حيث يمكن أجراء عمليات الأضافة وحذف قيد وتحديث قيد وأنهاء البرنامج من خلال أزرار الأوامر والتي تظهر بالصورة أدناه مع ملاحظة أن كل زر من هذه الأزرار يحتوي على شفرة خاصة به تم أنشائها من قبل البرنامج نفسه والتي يمكن الدخول اليها من خلال نافذة الشفرة .

شكل رقم (3 – 37) : تنفيذ برنامج قاعدة البيانات بأستخدام المعالج

ملاحظة : في حالة كون المشروع يحتوي على أكثر من نموذج كما في الشكل رقم (3 – 38) نستطيع اظهار النموذج الذي نريده أن يظهر في المقدمة أثناء تنفيذ المشروع وذلك من خلال أختيار الأمر التالي :

Project ⟶ Project1 Properties ⟶ General ⟶ Startup Object



شكل رقم (3 – 38) : تحديد النموذج الأول أثناء تنفيذ المشروع

3 – 16 لغة الأستفسارات المهيكلة: Structured Query Language (SQL)

تستخدم هذه اللغة ضمن اية لغة برمجية للأستعلام عن بيانات معينة ضمن قاعدة البيانات وذلك وفق شروط محددة مُسبقاً من قبل مصمم القاعدة ، وتسمى أيضاً بلغة الأستفسارات البنيوية . بواسطة هذه اللغة يمكن استرجاع البيانات التي نحتاجها من القاعدة ، أو تصفيتها حسب شرط معين أو ترتيبها تصاعدياً أو تنازلياً ، وكذلك أدارة قواعد البيانات العلائقية Relational DB .

3 – 16 – 1 أنماط لغة الأستفسارات المهيكلة (البنيوية) : Types of SQL

يمكن أستخدام عبارة الأختيار SQL Select في أي مكان من خلال أسم الجدول أو أسم الأستعلام أو أسم الحقل ، فعلى سبيل المثال يمكن أستخدام عبارة الـ SQL لأستدعاء أسم جدول قاعدة البيانات بطريقة كائن مجموعة السجلات Open Recordset . بشكل عام هناك أربعة أشكال رئيسية لتنفيذ لغة SQL ، وهي كالأتي :

1. عرض جميع حقول وسجلات قاعدة البيانات ، والصيغة العامة لها :

Select * from < *table name* > .

2. عرض بعض حقول القاعدة مع جميع سجلاتها ، والصيغة العامة لها :

Select field , field2 from < *table* > .

3. عرض جميع حقول القاعدة وبعض سجلاتها بالأعتماد على شرط ما ، والصيغة العامة لها :

Select * from table where < *condition* > .

Ex : Select * from info where [Number] > 2

4. عرض بعض حقول وسجلات القاعدة بالأعتماد على شرط ما ، الصيغة العامة لها :

Select field1 , field2 from table where < *condition* >

Ex : Select Number , Name from Info where [Number] > 2

3 – 16 – 2 برمجة أنماط لغة الأستفسارات المهيكلة: SQL types programming

البرنامج التالي يوضح كيفية عمل لغة SQL :

1. نقوم بتصميم قاعدة بيانات مايكروسوفت أكسس MS-Access 97 تحتوي على جدول واحد وليكن أسمه " Information " يضم ثلاثة حقول ولتكن أسمائها على التوالي " Number , Name , Marage " كما في الشكل رقم (3 – 39) .

Information : Table

	Number	Name	Marage
▶	1	Sameer	☑
	2	Ahmed	☑
	3	Fadi	☐
	4	Ali	☐
	5	Sara	☑
	6	Taha	☐
	7	Mays	☑
	8	Noor	☐
	9	Mohamed	☑
	10	Zaki	☑
*	0		☐

شكل رقم (3 – 39) : جدول قاعدة بيانات أكسس

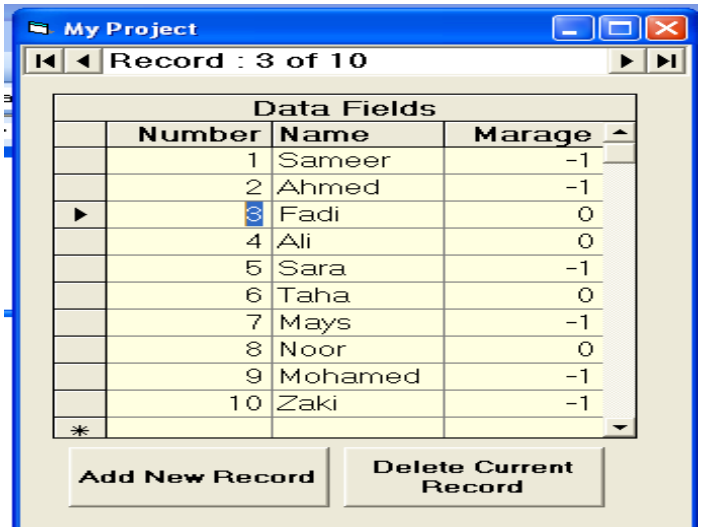
2. نقوم بفتح مشروع جديد بلغة الفيجوال بيسك ونقوم بأضافة كائن الوصول الى البيانات Data Control الى النموذج مع أضافة كائن عرض البيانات الشبكي Data Bound Grid ، وذلك من خلال أختيار الأمر التالي :

Project ⟶ Component ⟶ Data Bound Grid Control 5.0 (SP3)

3. نقوم بربط كائن Data مع القاعدة من خلال خاصية Database Name ، ومن ثم ربطه مع الجدول من خلال خاصية Record Source .

4. نقوم بربط كائن Data Bound Grid مع الكائن Data من خلال خاصية Data Source ، ثم نقوم بربطه مع حقول القاعدة من خلال نقرة يمين بزر الفأرة على كائن الشبكة وأختيار الأمر Retrieve Fields ، مع أمكانية جعل كائن الشبكة قابل لأجراء عمليات الأضافة والحذف والتحديث عليه من خلال الأمر Properties .

5. نقوم بأضافة زر أمر Command Button الى النموذج حيث سيتم عند النقر عليه أضافة قيد جديد الى الشبكة ، كما نقوم بأضافة زر أمر ثاني يقوم بحذف قيد من الشبكة أثناء النقر عليه وكما موضح ذلك بالشكل رقم (3 – 40) وذلك من خلال كتابة الشفرات التالية :



شكل رقم (3 – 40) : قاعدة البيانات بأستخدام عبارات الـ SQL

```
Private Sub cmdaddnew_Click()
Data1.Recordset.AddNew
Data1.Caption = Data1.Recordset.RecordCount + 1 _
& " of " & _
Data1.Recordset.RecordCount + 1
End Sub
```

```
Private Sub cmddelete_Click()
On Error Resume Next
Data1.Recordset.Delete
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
  Data1.Recordset.MoveFirst
End If
End Sub
Private Sub Data1_Reposition()
Data1.Caption = "Record : " & Data1.Recordset.AbsolutePosition + 1 _
& " of " & _
Data1.Recordset.RecordCount
End Sub
```

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
End Sub
```

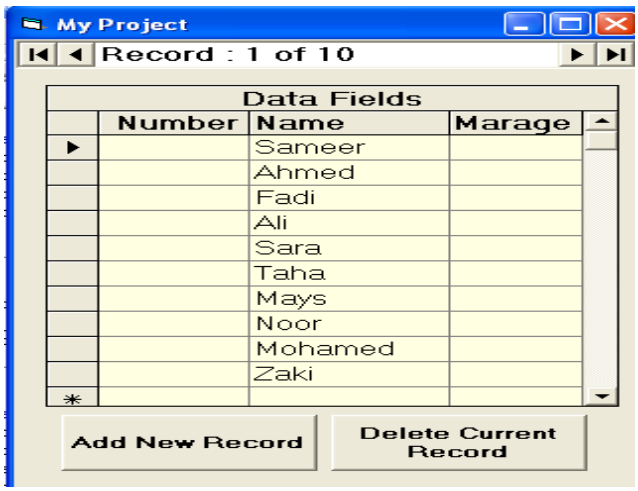
6. نقوم الأن بأستخدام أيعازات لغة SQL وملاحظة تأثيرها في عرض بيانات الجدول ، وذلك من خلال كتابة الشفرات التالية داخل النموذج عندما يكون حدث النموذج في حالة التحميل Load وكالأتي :

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
```

```
Data1.Recordset.MoveFirst
Data1.DatabaseName = App. Path + "\students.mdb"
Data1.RecordSource = "select Name from information "
End Sub
```

أسم قاعدة البيانات

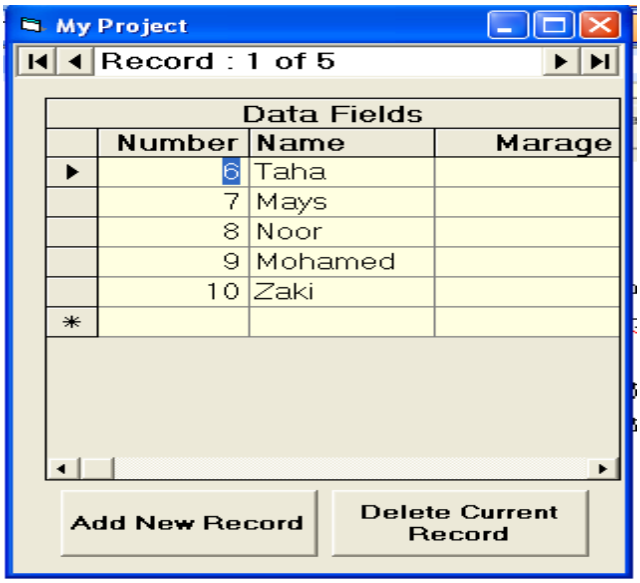
- بعد تنفيذ البرنامج سيتم عرض بيانات حقل الأسم فقط ، وكما موضح ذلك بالشكل رقم (3 – 41).



شكل رقم (3 – 41) : عرض بيانات حقل الأسم فقط بأستخدام عبارة Select Name

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\students.mdb"
Data1.RecordSource = "select number,name from information" _
 & " where [number] > 5 "
End Sub
```

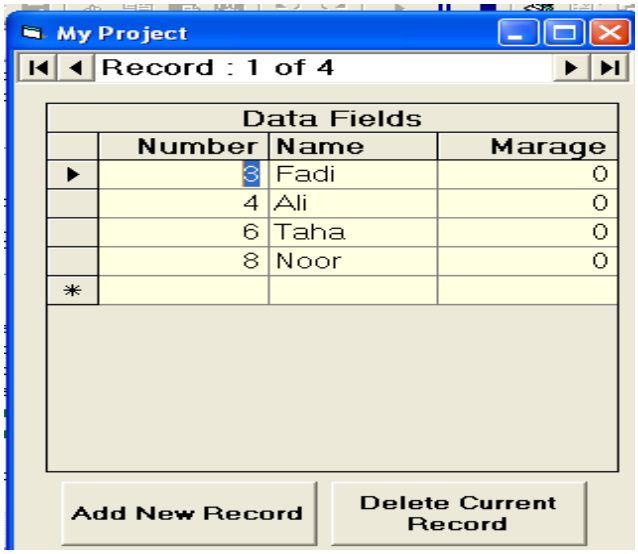
- بعد تنفيذ البرنامج سيتم عرض بيانات حقل الرقم والأسم فقط مع عرض سجلات البيانات التي حقل الرقم فيها أكبر من العدد 5 ، وكما موضح ذلك بالشكل رقم (3 – 42) .



شكل رقم (3 – 42) : عرض بيانات حقلي الرقم والأسم فقط مع بعض السجلات

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\students.mdb"
  Data1.RecordSource = "select * from information" _
  & " where [marage]=0 "
End Sub
```

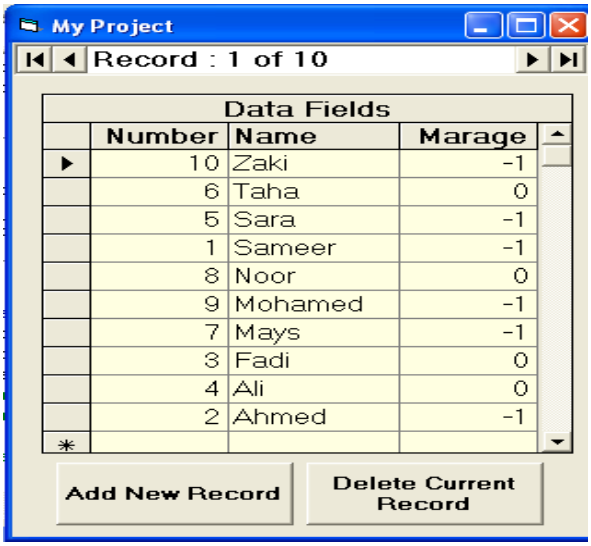
- بعد تنفيذ البرنامج سيتم عرض جميع الحقول مع بيانات أربعة سجلات فقط ينطبق عليها الشرط ، وكما موضح ذلك بالشكل رقم (3 – 43) .



شكل رقم (3 – 43) : عرض جميع حقول القاعدة مع بعض سجلاتها

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\students.mdb"
Data1.RecordSource = "select  * from information " _
    & " order by name desc"
End Sub
```

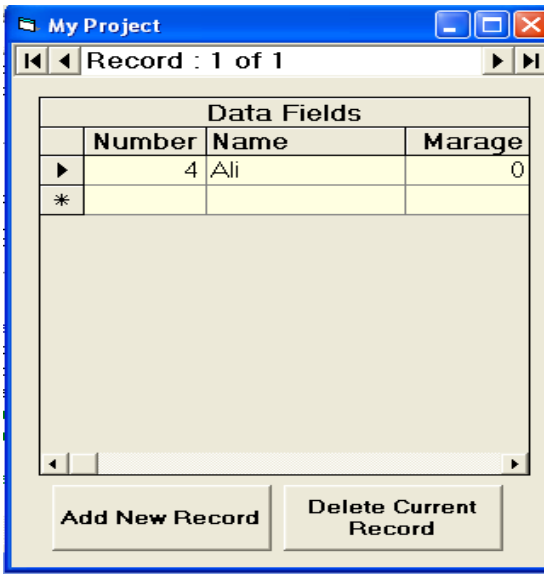
- بعد تنفيذ البرنامج سيتم عرض جميع الحقول والسجلات مع ترتيبها تنازلياً ، وكما موضح ذلك بالشكل رقم (3 – 44) .



شكل رقم (3 – 44) : ترتيب بيانات القاعدة تنازلياً بأستخدام عبارة SQL الشرطية

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\students.mdb"
Data1.RecordSource = "select * from information" _
 & " where [name] ='ali'"
End Sub
```

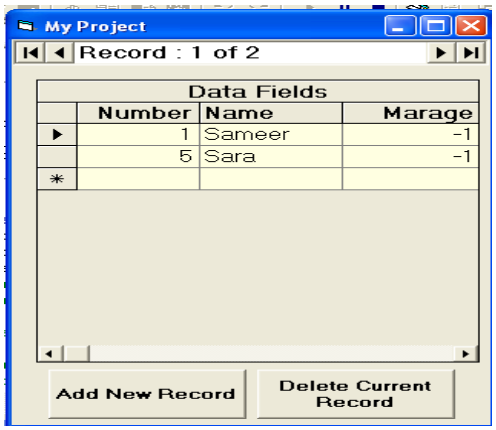
- بعد تنفيذ البرنامج سيتم عرض جميع الحقول ولكن لسجل واحد فقط وذلك من خلال عبارة SQL الشرطية ، وكما موضح ذلك بالشكل رقم (3 – 45) .



شكل رقم (3 – 45) : عرض سجل واحد مع جميع حقول القاعدة

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\students.mdb"
Data1.RecordSource = "select * from information " _
    & " where [name] like 's*' and " _
    & " [number]<=5"
End Sub
```

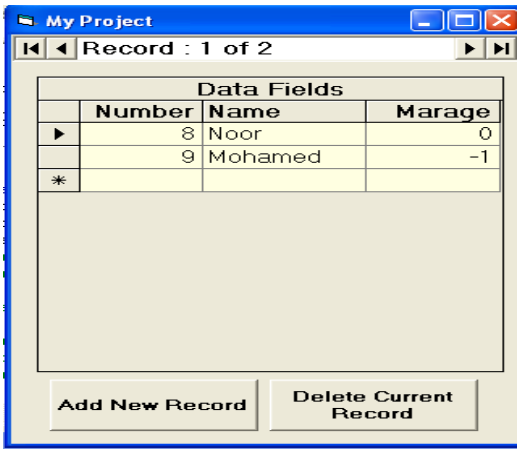
- بعد تنفيذ البرنامج سيتم عرض جميع الحقول ولسجلين فقط وذلك من خلال عبارة SQL الشرطية ، وكما موضح ذلك بالشكل رقم (3 – 46) .



شكل رقم (3 – 46) : عرض جميع حقول القاعدة مع سجلين فقط

```
Private Sub Form_Load()
On Error Resume Next
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Data1.DatabaseName = App.Path + "\studentsmdb"
Data1.RecordSource = "select * from information " _
    & " where mid$(name,2,1)='o'"
End Sub
```

- بعد تنفيذ البرنامج سيتم عرض جميع الحقول ولسجلين فقط وذلك من خلال عبارة SQL الشرطية ، وكما موضح ذلك بالشكل رقم (3 – 47) .



شكل رقم (3 – 47) : عرض جميع حقول القاعدة مع سجلين فقط أعتماداً على حقل الأسم

Ex \ Implement the following code :

```
1. Data1.RecordSource = "select  * from information " _
    & " order by name asc"
2. Data1.RecordSource = "select * from information " _
     & " where [name] like '*i'"
3. Data1.RecordSource = "select * from information " _
     & " where [name] like '*e?'"
4.
Private Sub cmddisplay_Click()
Dim DB As Database
Dim RS As Recordset
Set DB = DBEngine.OpenDatabase(App.Path + "\students.mdb")
Set  RS = DB.OpenRecordset("select mid$(name,1,2),name
from information", dbOpenDynaset)
RS.MoveFirst
Do Until RS.EOF = True
  MsgBox " Hello " & RS.Fields(1) & " - " & RS.Fields(0), , " Welcome "
  RS.MoveNext
Loop
RS.Close
End Sub
```

ملاحظة : في البرنامج الأخير سنحتاج الى تحميل مكتبة البرمجة التالية وذلك للحصول على كل الأمكانيات المطلوبة والتي تُساعدنا في التعامل مع قواعد البيانات :

Project ⟶ References ⟶ Microsoft DAO 3.51 Object Library

1 . Visual Basic - university of technology – academic board – by Prof. Dr. Abdul Mutalib I. Ahmed 2004 .

2 . تعلم MS : Visual Basic (عن طريق الأمثلة) .
أعداد : د. باسل الخطيب 2001 .

3 . Visual Basic 6 – How to .
أعداد : ايريك بربرلي ، أنطوني برينس ، دايفيد رينالدي – 1999 .

4. MSDN library (CD) – Microsoft Developer Network
http ://www.microsoft.com .

5 . Visual Basic 6 للجميع نحو برمجة كائنية التوجه OOP .
أعداد : تركي العسيري 2002 . (PDF from internet web page) .

6 . تعلم لغة فيجوال بيسك للمبتدئين والمتوسطين .
أعداد : جوزيف شاكر يوسف (PDF from internet web page) .

7 . موسوعة الفيجوال بيسك .
أعداد : محمد علي حسن قاسم (PDF from internet web page) .

8 . Working in Microsoft Office – by
Ron Mansfield 1996 .

9 . Microsoft Access 2002 – by
Sara E. Hutchinson , Glen J. coulthard 2002

10 . دورة خاصة Access 2003 .
أعداد : فراس خيروكجان 2004 .