Employment the Object Oriented Idea with Relational Database using by Ado.Net

Ban Ghanm M.
ban_ghanm@yahoo.com
Collage of Computer Science and Mathematics
University Of Mosul

Received on: 9/1/2011 **Acceptance on : 16/3/2012**

ABSTRACT

With the development of technology and appearing of computers that simplified many of computer and logical applications, so the developed with all capabilities to take and give the benefit in the same time, it is necessary because the updating and extending in bank operation space by using the computer which became the spinal cord in managing these operation. This research present the system that simplicity and quicklicity of the bank job by inserting, deleting, data viewing and information and special reports extracting to any customer instead of the manual usage which is usually doing probability of mistakes is large and data corruption. We depend on ADO.NET technology to data base access which more advance technology that is presented from micro soft to support language that not special in database like visual basic, visual java, visual C++, etc. ADO.NET technology was presented to programmers who specialist not in data base language. it enables to database access and manipulating with it in higher degree of simple and flexible way by mixing the command s of programming language with SQL commands to use data base in all applications.

Keywords: Object Oriented, Relational Database, Ado.Net.

توظيف فكرة البرمجة الكيانية مع قواعد البيانات العلائقية باستخدام تقنية do.net

بان غانم معيوف

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ الاستلام : ٢٠١١/١/٩ **تاريخ القبول: ٢٠١٢/٣/١٦**

الملخص

إن التطور التكنولوجي للعصر وظهور اللغات البرمجية التي من خلالها يتم التحديث والتوسيع في شتى المجالات ومنها المجال المصرفي فقد أصبح من الضروري التحديث والتوسيع في هذا المجال لذا فقد تم في هذا البحث بناء نظام لتسهيل وتسريع العمل المصرفي من إضافة وحذف وعرض البيانات واستخراج المعلومات والتقارير الخاصة لأي زبون بدل الاستخدام اليدوي الذي يكون عادة بطيئا ويحتمل الكثير من الأخطاء وتكون البيانات عرضة للتلف والعبث. وتم بناء ثمانية جداول وبناء أربعة عروض تقوم بعرض البيانات بطريقة منظمة وسهلة القراءة ومن خلاله تتم طباعة التقارير باستخدام لغة SQLserver2000. وقد تم إجراء جميع العمليات المصرفية عليها عن طريق تقنية ADO.NET. إن الهدف الرئيسي ADO.NET هو توفير وصول بسيط إلى قاعدة البيانات من خلال أصناف سهلة الاستخدام تمثل الجداول والأعمدة والحقول في قاعدة البيانات بالإضافة إلى ذلك فان ADO.NET تقدم الصنف DATASET الذي يمثل مجموعة من البيانات المعادة من جدول أو أكثر ومغلفة ضمن وحدة واحدة وهذا المفهوم الجديد بحد ذاته يمثل توسعة لإمكانيات الوصول إلى البيانات واستخدمت لغة Visual C++ 2005 في بناء واجهات (Forms) هذا النظام.

الكلمات المفتاحية: البرمجة الكيانية، البيانات العلائقية، تقنية Ado.net

1. مدخل إلى قواعد البيانات و ADO.Net (2, 3, 5)

في السنوات الأولى من عملية بناء البرامج كانت معظم برامج الكومبيوتر تستخدم طريقة خاصة بها لتوصيف بياناتها، حيث كانت تحفظ تلك البيانات على شكل ملفات في جهاز الحاسوب، وكانت كل شركة من شركات صناعة البرمجيات لها طرقها الخاصة في بناء ملفات برامجها، بل أكثر من ذلك حيث كان كل برنامج من البرامج له طريقته الخاصة في حفظ بياناته بجهاز الحاسوب، قد لا يرى البعض مشكلة كبيرة في بادئ الأمر بأن يحفظ البرنامج بياناته في ملفات خاصة به، إلا انه مع بناء أول أجزاء البرامج تبدأ المشاكل بالظهور، وتزداد كلما ازداد حجم البرنامج وكمية البيانات المخزنة بواسطتها حيث من المستحيل عمل برنامج قادر على تغطية كافة الاحتمالات والأسئلة. التي قد نطلبها منه ولو حاولنا بناء البرنامج ليتمكن من ذلك لكلفنا الكثير من الوقت والجهد. لذلك اتجهت أنظار العلماء إلى بناء نظام تخزين موحد للبيانات قادر على استرجاع البيانات بطريقة موحدة وأكثر سلاسة وسرعة في نفس الوقت للتقليل من المشاكل المذكورة أنفا.

قواعد البيانات هي طريقة تخزين موحدة للبيانات، بحيث يمكن للبرامج العادية استخدامها لتخزين بياناتها، كما يمكن في نفس الوقت الوصول إلى البيانات مباشرة بدون الحاجة إلى البرنامج نفسه ولا يجب علينا أن نهتم بتفاصيل طريقة الحفظ الفيزيائية لتلك البيانات.

إن نظم إدارة قواعد البيانات تتكون من تجمع ذي علاقة من البيانات ومجموعة من البرمجيات للتخاطب والتعامل مع هذه البيانات. ومجموعة البيانات عادة يطلق عليها اسم قواعد البيانات، والهدف الرئيسي من إدارة نظم قواعد البيانات هو توفير البيئة المناسبة والفعالة لتجميع هذه المعلومات وفرزها وتبويبها وحذفها واسترجاعها بطريقه سهلة ونموذجية. وتوفر لنا الفصل بين الهيكلية المنطقية والفيزيائية للملف وطريقة موحدة لحفظ واسترجاع البيانات وحماية البيانات من التلف وحمايتها من سوء الاستخدام والمحافظة على تكامل البيانات منطقيا ككتلة واحدة والوصول المباشر لها عند الحاجة، أما الأجزاء الرئيسية لقواعد البيانات فهي الجدول عبارة عن الملف الذي يستخدمه البرنامج لتخزين بياناته، الجداول تخزن بياناتها على شكل صفوف و أعمدة، كما يمكننا إضافة أعمدة جديدة للجداول في أي وقت، لكل عمود نوع معين من البيانات التي يمكن أن نخزنها فيه، كما يجب أن نحدد عدد الأعمدة و أسمائها ونوع البيانات التي ستخزن في كل منها وثم بناء علاقات بين الجداول والحفاظ على تكامل البيانات وهذه العلاقات تضمن لنا بتكامل البيانات، حيث لا يمكن أن نضيف حركة مالية في جدول الحركات المالية للمشترك، بدون أن نقوم بإضافة المشترك أساسا إلى قاعدة البيانات، كما لا يمكننا حذف مشترك من المشتركين و ذلك بدون حذف حركاته، وإلا اختلت البيانات في قاعدة البيانات والاستعلامات عبارة عن أوامر بلغة SQL مخصصة للقيام بعملية ما على البيانات وبالمناسبة الاستعلامات لا تقوم بعملية الاستعلام فقط كما قد يوحي اسمها بذلك ولكنها أيضا قادرة على إجراء التعديلات على البيانات في قاعدة البيانات نفسها، بل إنها قادرة أيضا على بناء جداول أخرى في قاعدة البيانات. جزء مهم آخر من قواعد البيانات هو الصلاحيات والمستخدمين، فلو تخيلنا لو كان الجميع يستطيع الوصول إلى البيانات وتعديلها، لكانت كارثة حقيقية، حيث أن خادم قواعد البيانات يعطينا إمكانية إغلاق البيانات بواسطة أسماء مرور وكلمات سر. وبهذا قد نعطي الإمكانية لقسم إدارة المشتركين مثلا من الاطلاع على بيانات المشتركين بدون الاطلاع على بعض المعلومات المالية الحساسة الخاصة بهم، ونعطي صلاحية للمحاسبين في دائرة الحسابات تلك الصلاحية فقط ونمنع قسم الصيانة من الوصول إلى جدول المشتركين تماماً ،حيث انه لا علاقة لهم بذلك الجدول على الإطلاق. وكذلك على محركات قواعد البيانات أن تتبع طريقة تمنع إمكانية إحداث تلف في تلك الملفات مهما كانت الظروف،

فالبيانات قد تكلف الكثير من المال، فضياع حركات آخر نصف ساعة من حركات البيانات في بنك قد تكون كارثة، وضياع حركة واحدة قد تكون مشكلة كبيرة للبنك نفسه.

2 . نبذة عن ADO.NET (1, 9, 10)

هي مجموعة من الأصناف (Classes) المدمجة مع إطار عمل NET. والتي تمكننا من الوصول إلى البيانات من خلال اللغات التي تدعمها NET. يمكننا من خلال ADO.NET الوصول إلى البيانات العلائقية مثل قواعد البيانات Microsoft Access أو قواعد بيانات SQL SERVER بالإضافة إلى قواعد بيانات أخرى .ويمكننا استخدام ADO.NET للوصول إلى مصادر البيانات الغير علائقية أيضاَ.

1.2 - الأهداف التصميمية لـ ADO.NET :- **إن** الهدف الرئيسي لـ ADO.NET هو توفير وصول بسيط للبيانات العلائقية ويتم ذلك من خلال أصناف سهلة الاستخدام تمثل الجداول والأعمدة والحقول في قواعد البيانات العلائقية وبالإضافة إلى ذلك فأن ADO.NET تقدم الصنف Dataset والذي يمثل مجموعة من البيانات المعادة من جدول أو أكثر ومغلفة ضمن وحدة واحدة وهذا المفهوم الجديد بحد ذاته يمثل توسع لإمكانيات الوصول إلى البيانات.

وإن تقنية ADO.NET قابلة للتوسع و التطور فهي توفر إطار عمل لمزودي بيانات NET. بحيث يمكننا إنشاء مزودات جديدة للبيانات (Data source) تمكننا من الوصول إلى أي مصدر من مصادر البيانات ولقد تم دمج مزودين للبيانات في ADO.NET حيث تمكننا للوصول إلى أربعة أنواع من الـ DataSource وهي

1. مصدر بيانات من نوع SQL Server (والذي اعتمد في هذا البحث).
2. مصدر بيانات من نوع Oracle.
3. مصدر بيانات من نوع OLE.
4. مصدر بيانات من نوع ODBC الذي يحتفظ بالعديد من مصادر البيانات في داخله.

وبإمكاننا الجمع بين البرمجة الشيئية وبرمجة قواعد البيانات العلائقية:- حيث يمكن المبرمجين من تكوين برامجهم والتعامل معها كـ OOP واستخدام خصائصها مثل التوارث (Inheritance) وخاصية إخفاء البيانات (Encapsulation) والتعامل مع البيانات ككيانات (Objects) بدل التعامل مع الصفوف والأعمدة بالشكل المبسط.

ومن خصائصها (ADO.NET) أنها مصممة من الأساس للعمل في بيئة غير متصلة بقواعد البيانات (ويمكنها العمل باتصال دائم مع قاعدة البيانات)، ويستخدم الكائن Data Set للاحتفاظ بعدة مجموعات من البيانات، وتعمل كنظام بيانات منفصل عن قاعدة البيانات فهي لا تتصل بقاعدة البيانات إلا عند الضرورة وبالتالي لن نستهلك مورد النظام إلا عند الضرورة.

2.2 معمارية الـ ADO.NET (1, 9, 10)

والمقصود هنا الخصائص المزودة والمساعدة في عملية الاتصال لقراءة وتكييف البيانات وهي:-

1. مجموعة البيانات (DataSet):- وهي أن الكائن يستطيع تخزين أكثر من جدول أو نتيجة استعلام في نفس الوقت حيث يمثل كل واحد من هذه الجداول كائنا منفصلا عن الأخر. كما موضح أدناه

```
// Construct the dataset and fill it
        DataSet myDataSet = new DataSet("table name");
        myAdapter.Fill(myDataSet, "table name");
```

2. مجموعة البيانات (SqlDataAdapter) :- يمثل الجسر الذي يربط بين DataSet وقاعدة بيانات ويدعم أوامر (Select – Update – Delete - Insert into) بالتالي بإمكانه القيام بعمليات مختلفة على البيانات كما أنه المسؤول عن تحميل DataSet بالبيانات. كما موضح أدناه

Constructing the Data Adapter

```
// Creating a SQL string and data adapter object
        string sql = "select * from tablename";
        SqlDataAdapter myAdapter = new SqlDataAdapter(sql, myConnection);
```

3. مجموعة البيانات (SqlDataReader):- يستخدم هذا الكائن لقراءة البيانات فقط ويمكنها قراءة كميات ضخمة من البيانات، منها تلك التي لا يمكن تخزينها في الذاكرة مؤقتا .

```
//create a SqlDataReader object and call the ExecuteReader()
// method of the SqlCommand object to run the SELECT statement
SqlDataReader mySqlDataReader = mySqlCommand.ExecuteReader();
// step 6: read the row from the SqlDataReader object using
// the Read() method
          mySqlDataReader.Read();
```

4. مجموعة البيانات (SqlConnection):- يمكننا هذا الكائن من إنشاء اتصال مع قاعدة البيانات.

Constructing the Sql Server connection object

```
//Create a Connection object
    string ConnectionString = "initial catalog=database1;
    data  source=tablename\\SQLSERVER2000;

    SqlConnection myConnection = new SqlConnection();
    myConnection.ConnectionString = ConnectionString;
```

5. مجموعة البيانات (SqlCommand):- يسمح هذا الكائن لكائن SqlDataAdapter تنفيذ الأوامر على قاعدة البيانات

3.2 المميزات الجديدة في ADO.NET (12)

إنّ الميزّاتَ التاليةَ جديدة في ADO.NET

ADO.NET Entity Framework يقلل من حجم البرنامج والصيانة المطلوبة للتطبيقات المحركة للبيانات وبإمكانك البرمجة باتجاه موديلات البيانات المعرفة بمصطلحات من الكيانات والعلاقات.

المفاتيح الأجنبية في النموذجِ التصوريِ ممْكِنُ تكوين نماذجَ تصوريةَ (conceptual models) التي فيها تُقابلُ الأعمدةَ الرئيسيةَ الأجنبيةَ في قاعدةِ بيانات ملكياتَ فردية على أنواعِ الكيانِ.

كيانات تتبع الذاتيةِ لتطوير تطبيقات N-TIER ممْكِنُ أَنْ تَستعملَ الكياناتَ تتبع الذاتيةِ عندما تعَمَل مَع تطبيقات N-TIER. الكيانات تتبع الذاتيةِ يُمْكِنُ أَنْ تُسجّلَ التغييراتَ إلى مفرد، مركّب، وملكيات ملاحةِ. معلومات التتبع في أجسامِ تتبع الذاتيةِ يُمْكِنُ أَنْ تُقدّمَ إلى سياقِ الجسمَ على جانبِ الخدمةَ.

طرق جديدة لتطوير تطبيقات N-TIER الطرق الجديدة على الأصنافِ في نظامِ البيانات تَجْعلُ كياناتُ namespace أكثر سهولة لتَطوير تطبيقات N-TIER.

دعم EntityDataSource لسيطرةِ QueryExtender تَدْعمُ سيطرةُ EntityDataSource سيطرة QueryExtender الآن، التي تُستَعملُ لتكوين المرشحاتِ للبياناتِ وتحديد الإسترجاعَ مِنْ مصدر البيانات.

امكانية الأختبار مع object set (T entity) طَبعَ صنفُ ObjectContext الآن يرجع (ObjectSet (Tentity)) التي ممْكِنُ أَنْ تَستعملَ لأنجاز عمليات التكوين، القَرأةَ، التحديث، والحَذفُ على أجسامِ الكيانِ.الصنف(ObjectSet (Tentity) يُطبّقُ وصلة Iobjectset (T entity). وصلة(Iobjectset (T entity) قَدْ يَكُون مفيدةَ في اختبار السيناريوهات.

التنفيذ المباشر لأوامرِ الخزنِ ممكنْ تُنفيّذَ أوامرَ الخزنِ مباشرة مِنْ ObjectContext.

أجسام إصرارِ الجاهلةِ ممْكِنُ أَنْ تَستعملَ أصنافَ البياناتِ الخاصةِ مع موديل البيانات بدون تغيير أصنافِ البياناتَ هذا يَعْني بأنّك يُمْكِنُ أَنْ تَستعملَ "بسيط" clr object مثل أجسامِ المجالِ الحاليةِ، بتطبيقِ entity framework.

التحميل البطيء للأجسامِ ذات العلاقةِ بالتحميل البطيءِ، مَعروف كذلك بِالتحميل المُرْجَأ، تَعلّقَ بالأجسامِ تُحمّلُ آلياً مِنْ مصدرِ البياناتَ عندما تَدْخلُ ملكية ملاحةِ.

وظائف في LINQ إلى إستفساراتِ الكياناتِ يُزوّدُ EntityFunctions وأصناف SqlFunctions وصولاً إلى قانونيةِ وقاعدةِ بيانات يَشتغلُ مِنْ LINQ إلى إستفساراتِ الكياناتِ. يَسْمحُ EdmFunctionAttribute طريقة CLR للعَمَل وكيل وظيفة عرّفَ في النموذجِ التصوريِ أَو نموذجِ الخزنِ.

تحسينات OrderBy في LINQ للكياناتِ linq للكياناتِ يَسْألُ الذي يَحتوي مشغلَ OrderBy يُرجعُ النَتائِجَ الآن طَلَبَ بالحجّةِ حدّدَ في المشغلِ، بغض النظر عن موقعِ المشغلِ في الإستفسارِ. على أية حال، مشغل OrderBy في SQL الكيانِ وObjectQuery إستفسارات مهملةُ إذا هي لَيستْ المشغلَ الأخيرَ في الإستفسارِ. جيل رمزِ طبقةِ الجسمِ حسب الطلبِ ممْكِنُ أَنْ تُشكّلَ ADO.NET مصمم موديل بياناتُ الكيانِ لإسْتِخدام طبعاتِ النَصّ لتَوليد رمزِ طبقةِ الجسمِ حسب الطلبِ.

دعم الموديل الأولُ ساحر توليد قاعدةِ البيانات يُمكّنُك لتَعمَلُ عرض تصوريُ أولاً، ثم تكوين قاعدة بيانات التي تَدْعمُ النموذجَ. دعم النوعِ المعقّد ADO.NET مصمم موديل بيانات كيان يَدْعمُ أنواعَ معقّدةَ.

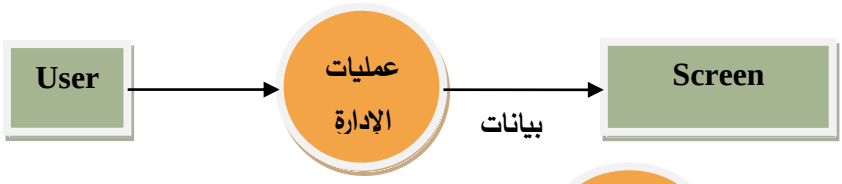
تَسْمِية الخدمةِ ساحر موديل بياناتُ الكيانَ وساحر موديلَ التحديثَ يُزوّدُ الخيارَ لاستعمال الأشكالِ المفردةِ أَو الجمعيةِ مِنْ الكيانِ، Entity Set، ويَسمّي Navigation Property لتَقديم رمزِ الطلبِ الأكثرِ قراءة.

3 . البرامج المستخدمة في عملية بناء هذا النظام:-

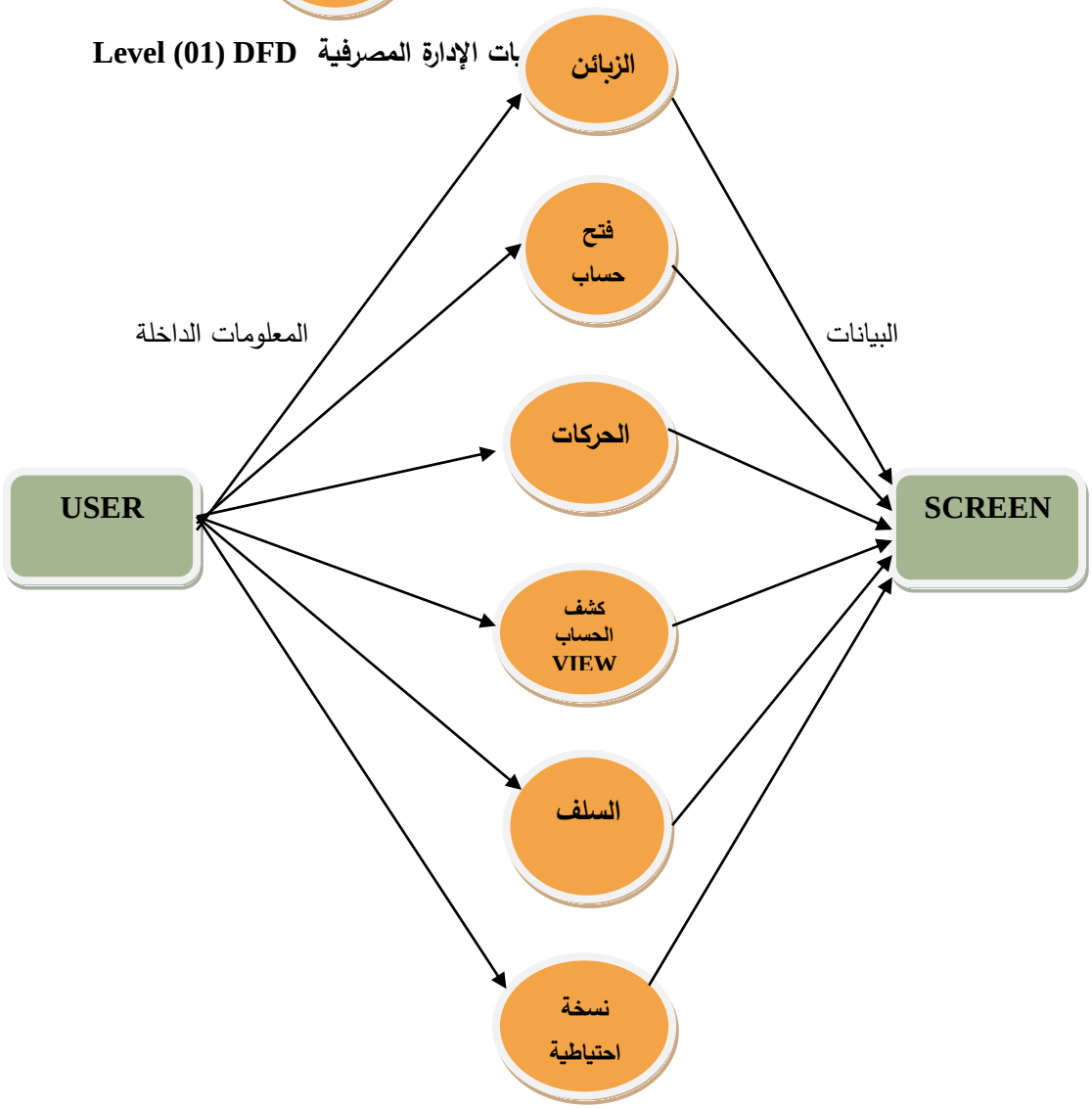
1- (Visual C++ 2005):- لقد استخدمت لغة Visual C++ 2005 ألتي هي إحدى لغات (Visual Studio2005) كلغة أساسية في بناء واجهات (Forms) هذا النظام حيث توفر هذه اللغة العديد من الأدوات التي تساعد في بناء هذا النظام.

2- (SQL Server 2000):- استخدمت هذه اللغة لبناء قاعدة البيانات التي تم إجراء جميع العمليات المصرفية عليها عن طريق الـ ADO.NET.

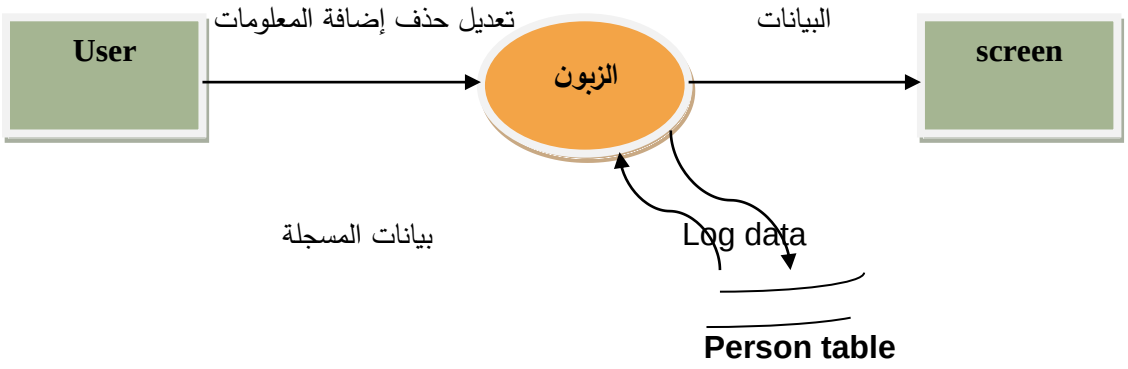
4 . تحليل النظام:- قبل الدخول إلى أي نظام يجب أن تكون هناك مخططات توضيحية تحلل هذا النظام وتم استخدام Data Flow Diagram وهي من طرق التحليل بالاعتماد على انسيابية (سريان) البيانات داخل النظام وذلك لدراسة المتطلبات والعمليات التي تحدث بهذا النظام في المرحلة الأولية ومن ثم الدخول في تفصيلات النظام



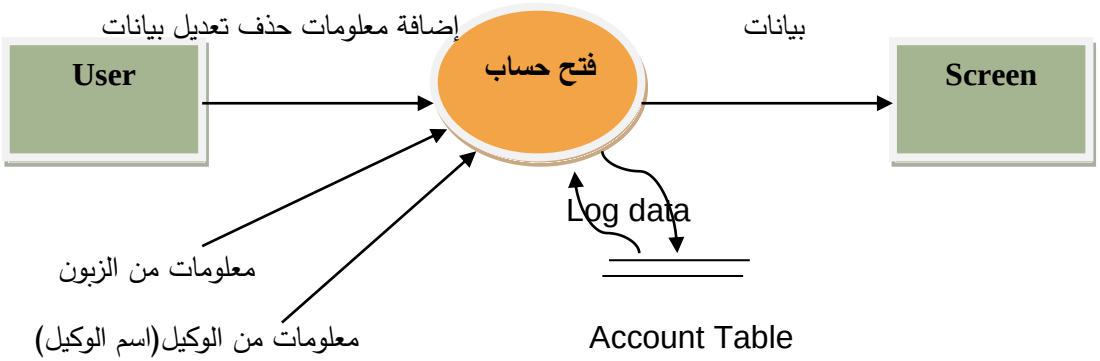
Level (01) DFD بات الإدارة المصرفية



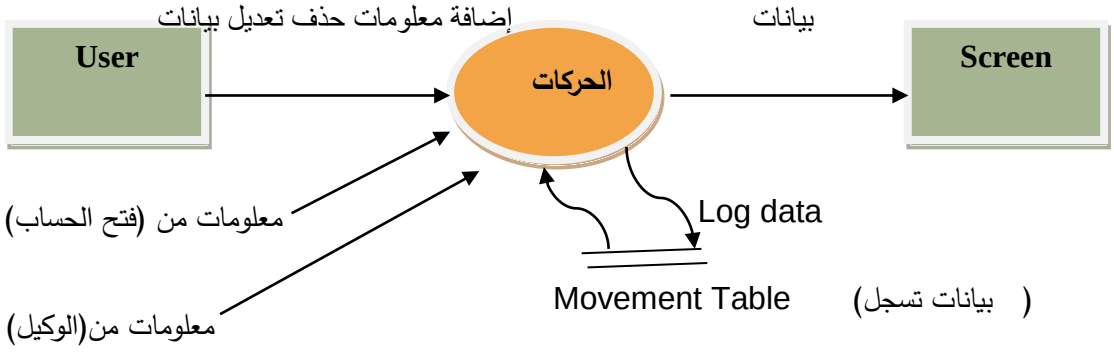
Level (02)DFD العمليات الإدارة المصرفية



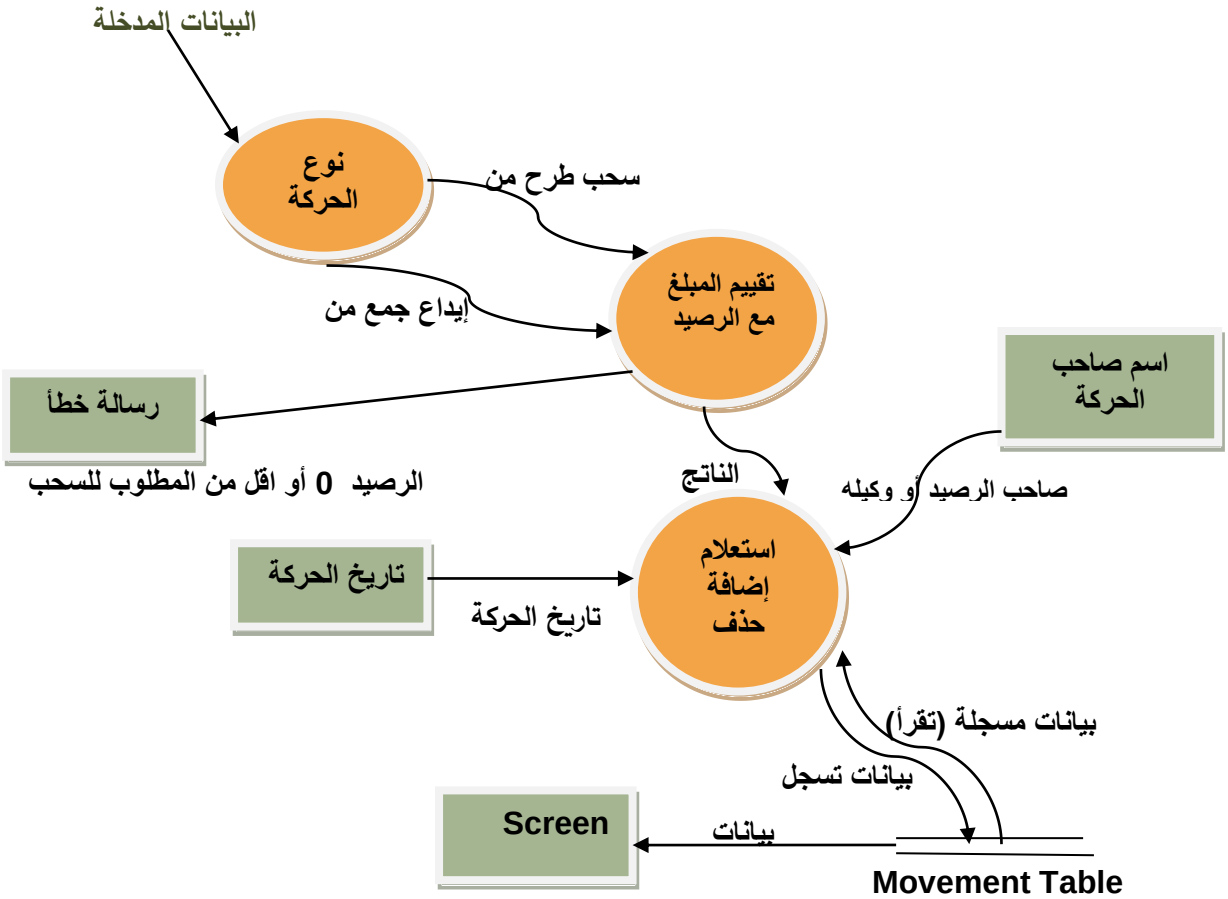
الجدول للزبونLevel (01) DFD



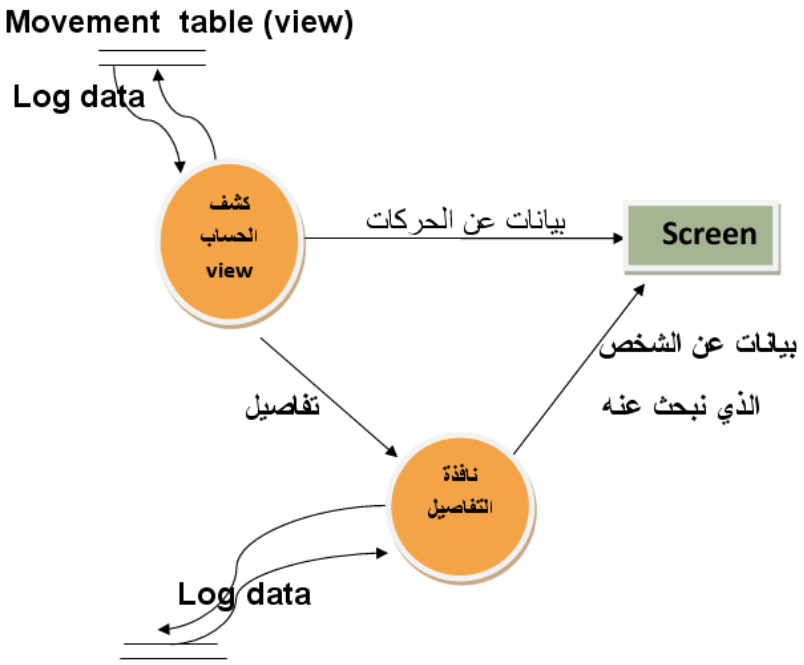
فتح حساب Level (01) DFD



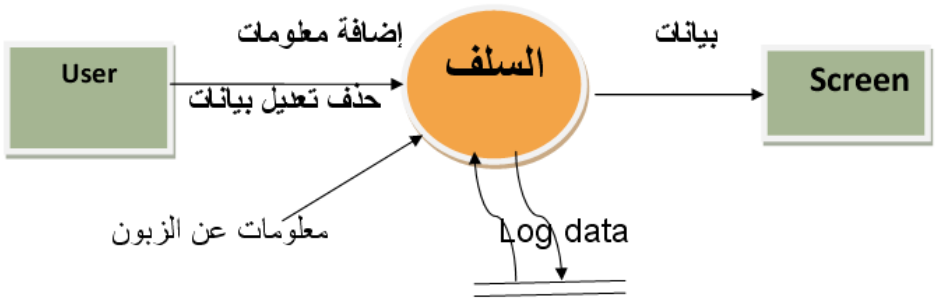
للقيام بأي حركة Level (02) DFD



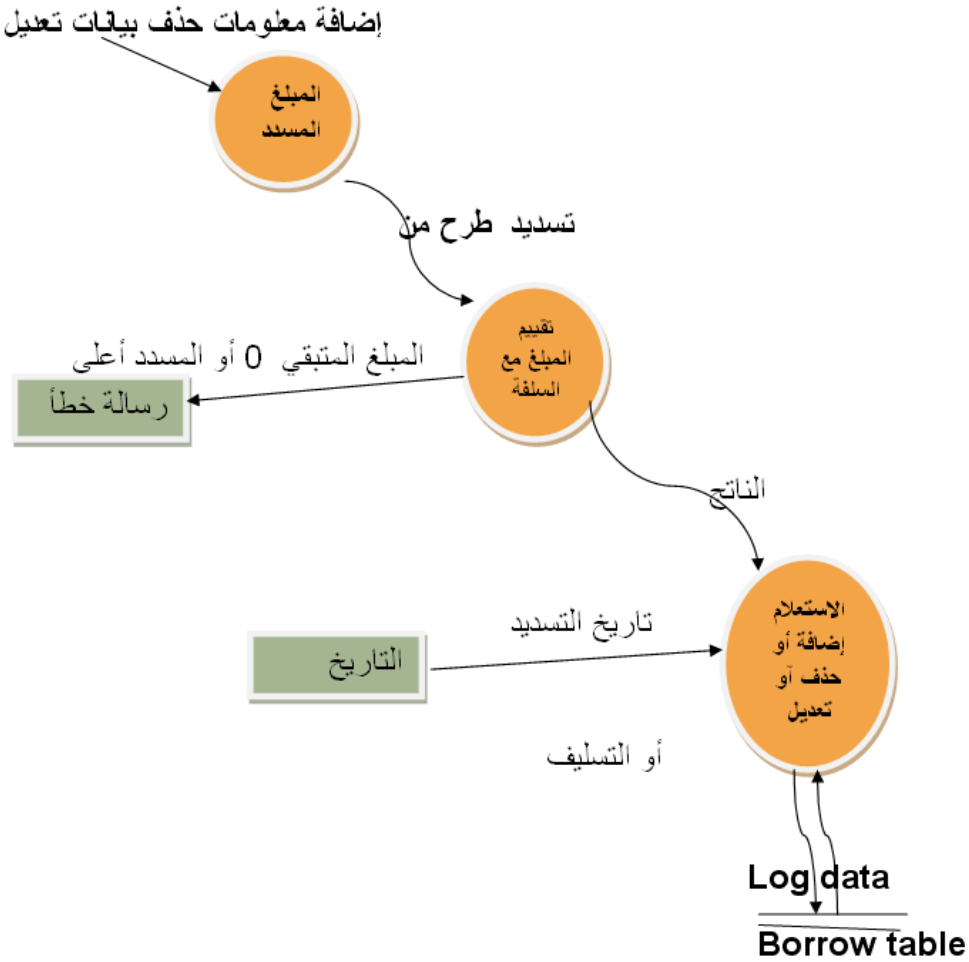
Level (03) DFD للحركات التي تمت



Level (04)DFD for لكشف الحسابات والتفاصيل للعمليات



Level (01) DFD السلف



Level (02) DFD العمليات التي تحدث على جدول السلفة

5 . خطوات بناء النظام

1- **بناء الجداول** :- قمنا بعملية بناء الجداول بلغة SQL Server ولقد احتجنا إلى بناء ثمانية جداول، في هذه اللغة يتكون الجدول من ثلاث أعمدة العمود الأول هو اسم الحقل و الثاني نوع الحقل والثالث يتعلق بإمكانية إبقاء الخلية فارغة.

1- **جدول تسجيل الدخول** (Login):- يقوم هذا الجدول بالاحتفاظ باسم المستخدم وكلمة المرور. فائدته أمكانية تغير أيا من الكلمتين في أي وقت بعد إدخال اسم المستخدم وكلمة المرور القديمتين وسيتم شرح ذلك في الواجهات (Forms).

Column Name	Data Type	Allow Nulls
Name	varchar(50)	☐
▶ Password	varchar(50)	☐
		☐

2- **جدول الزبائن** (Person) :- يقوم هذا الجدول بالاحتفاظ بالمعلومات الشخصية للزبون.

Column Name	Data Type	Allow Nulls
▶ FullName	varchar(60)	☐
Address	varchar(50)	☐
Home	decimal(30, 0)	☑
Mobile	decimal(36, 0)	☑
🔑 NoCard	varchar(50)	☐
Birth	datetime	☐
Gender	varchar(50)	☐
Note	varchar(200)	☑
Picture	varchar(150)	☐
RedCard	varchar(50)	☐
CardID	varchar(50)	☐

NoCard يمثل رقم الهوية وهو المفتاح الرئيسي (Primary Key) لهذا الجدول كما هو موضح في الجدول أعلاه وكما نلاحظ إن جميع الحقول Not Null عدا رقم الجوال ورقم الهاتف والملاحظات حيث إن حقل الملاحظات يمكن أن يبقى فارغا في جميع الجداول لعدم تقيد الزبون به.

3- **جدول فتح الحساب** (Account) :- هذا الجدول لحفظ الزبائن الذين يقومون بفتح حساب جديد حيث إن لكل زبون رقم حساب خاص به ولذلك فإن رقم الحساب (AccountNo) هو مفتاح رئيسي (Primary Key) في هذا الجدول و الاسم الكامل ورقم الهوية هما مفتاح أجنبي (Forgin Key).

Column Name	Data Type	Allow Nulls
▶🔑 AccountNo	decimal(36, 0)	☐
AccountDate	datetime	☐
AccountBalance	decimal(36, 0)	☐
CutterType	varchar(50)	☐
FullName	varchar(60)	☐
NoCard	varchar(50)	☐
Note	varchar(200)	☑

4- **جدول الوكلاء** (Agent):- يحتفظ هذا الجدول بوكلاء الزبون حيث يتم إدخال الوكيل كزبون عادي من جدول الزبائن (Person) .

Column Name	Data Type	Allow Nulls
AgentName	varchar(60)	☐
FullName	varchar(60)	☐
AccountNo	decimal(36, 0)	☐
NoCard	varchar(50)	☐

5- **الحركات (Movement)**:- يحتفظ هذا الجدول بأخر حركة يقوم فيها الزبون على الرصيد من سحب أو إيداع و رقم الصك هو المفتاح الرئيسي في هذا الجدول.

Column Name	Data Type	Allow Nulls
Type	varchar(50)	☐
FullName	varchar(60)	☐
MoverName	varchar(60)	☐
MoveDate	datetime	☐
AccountBalance	decimal(36, 0)	☐
AccountNo	decimal(36, 0)	☐
MoveAmount	decimal(36, 0)	☐
NoCheque	varchar(50)	☐
Note	varchar(200)	☑
NoCard	varchar(50)	☐

6- **جدول تفاصيل الحركات** (Detail):- يحتفظ هذا الجدول بأي حركة يقوم بها الزبون على الرصيد خلاف جدول الحركات ألذي يحتفظ بأخر حركة .

Column Name	Data Type	Allow Nulls
MoverName	varchar(60)	☐
AccountNo	decimal(36, 0)	☐
Type	varchar(50)	☐
MoveAmount	decimal(36, 0)	☐
AccountBalance	decimal(36, 0)	☐
MoveDate	datetime	☐
NoCard	varchar(50)	☐

7- **جدول السلف** (Borrow):- يحتفظ بأسماء المستلفين وكفيل كل مستلف ومبلغ السلفة والاستقطاعات ويتم إدخال المستلف وكفيله عن طريق جدول الزبائن ولقد تم استخدام رقم السلفة (BorrowNo) كمفتاح رئيسي.

Column Name	Data Type	Allow Nulls
BorrowNo	decimal(36, 0)	☐
FullName	varchar(60)	☐
workplace	varchar(40)	☐
KName	varchar(60)	☐
KWorkPlace	varchar(40)	☐
KCardNo	varchar(50)	☐
DateBorrow	datetime	☐
BorrowAmount	decimal(36, 0)	☐
DatePay	datetime	☐
Pay	decimal(36, 0)	☐
Total	decimal(36, 0)	☐
CardNo	varchar(50)	☐
Mony	decimal(36, 0)	☐
DateWork	varchar(10)	☐
Note	varchar(200)	☑

8- **جدول تفاصيل المستلفين** (DetailB):- يحتفظ بكل الاستقطاعات الجارية على المبلغ المتبقي من السلفة.

Column Name	Data Type	Allow Nulls
BorrowNo	decimal(36, 0)	☐
FullName	varchar(60)	☐
BorrowAmount	decimal(36, 0)	☐
Pay	decimal(36, 0)	☐
Total	decimal(36, 0)	☐
DatePay	datetime	☐

2- **العرض (View)**:- وهو من الأجزاء المهمة من أجزاء النظام حيث يقوم بعرض البيانات بطريقة منظمة وسهلة القراءة ومن خلاله تتم طباعة التقارير. تم تكوين الـ View بلغة SQL Server، ولقد استلزم النظام بناء أربعة عروض وهي:

1- عرض آخر حركة (V1):- Create view V1

([الرصيد] ، [رقم الحساب] ، [اسم المحرك] ، [نوع الحركة] ، [الاسم الكامل])

([رقم الصك] ، [تاريخ الحركة] ، [كمية الحركة،]

```
select FullName, Type, MoverName, AccountNo, AccountBalance, MoveAmount,
MoveDate, NoCheque  from  Movement1
```

2- عرض تفاصيل الحركات (V3) Create view V3

([كمية الحركة]، [نوع الحركة]، [الرصيد]، [اسم صاحب الحركة] ،

[رقم الحساب]، [تاريخ الحركة])

```
as select MoverName, AccountBalance, Type, MoveAmount, MoveDate,
AccountNo  from  Detail1
```

3- عرض السلف (VS):- يقوم Create view VS

([مبلغ السلفة]، [تاريخ المنح] ، [اسم الكفيل]، [رقم السلفة]، [اسم المستلف]،

[تاريخ الاستقطاع]، [الراتب]، [المتبقي]، [الاستقطاع])

```
as select
FullName,BorrowNo,Kname,DateBorrow,BorrowAmount,Pay,Total,Mony,Date
Pay  from  Borrow
```

4- عرض تفاصيل السلف (VB):- Create view VB

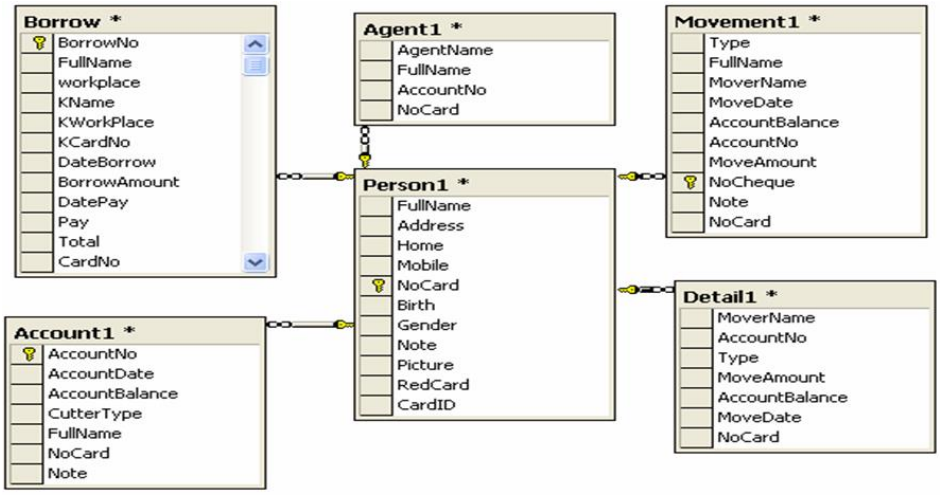
([المبلغ المتبقي]، [الاستقطاع]، [مبلغ السلفة]، [اسم المستلف]،

[رقم السلفة]، [تاريخ الاستقطاع])

```
as selec   FullName, BorrowAmount, Pay, Total, DatePay, BorrowNo from
DetailB
```

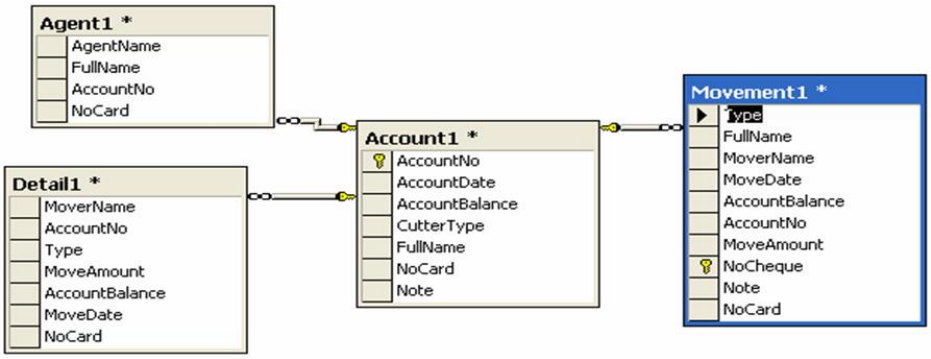
الرسم البياني للارتباط بين الجداول (Diagram):- عبارة عن رسم توضيحي يوضح العلاقة بين الجداول.

العلاقة بين جدول الزبائن و بقية الجداول :- تم ربط جدول الزبائن Person مع بقية الجداول بواسطة NoCard رقم الهوية الذي هو المفتاح الرئيسي كما هو موضح في الشكل (1)



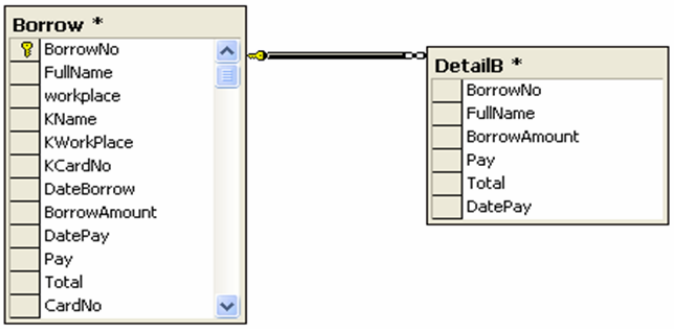
الشكل (1)

جدول فتح الحساب (Account) :- جدول فتح الحساب مربوط مع الجداول المبينة أدناه بواسطة رقم الحساب (AccountNo) كما موضح في الشكل (2)



الشكل (2)

جدول السلف (Borrow):- جدول السلف يرتبط مع جدول تفاصيل السلف بواسطة رقم السلفة كما هو موضح في الشكل (3)



الشكل (3)

بناء الواجهات (Forms):- الـ Form عبارة عن واجهة رسومية (GUI) تسهل عمل المستخدم وتحتوي على كافة الأدوات التي يحتاجها عند إجراء العمليات. تم بناءها باستخدام ADO.NET

لبناء واجهة تحتاج أن تملأ الـ DataGrid باستخدام الـ SQL data provider للاتصال مع SQL server database، تحتاج لتأكد من استيراد جميع الـ namespaces الضرورية لاستخدام ADO.NET و data provider. هذا المقطع البرمجي ضروري في بداية البرمجة للـ form لجلب ADO.NET و SQL server data provider

```
using System.Data;   // dataset and related classes
using System.Data.SqlClient;   // sql server data provider
classes
```

واجهة تسجيل الدخول (Login):- لضمان أمنية النظام يجب أن يكون هنا اسم مستخدم وكلمة مرور لمنع من ليس لهم صلاحية بالدخول على البرنامج وهذه هي واجهة تسجيل الدخول في هذا البرنامج.

الشكل (4)

بعد إن يدخل المستخدم اسم المستخدم وكلمة المرور الصحيحتين يدخل إلى النظام مباشرة وإلا ستظهر له رسالة إدخال بشكل خاطئ. أما زر خيارات يتيح لنا إمكانية تغيير كلمة الولوج كما في الشكل (5)

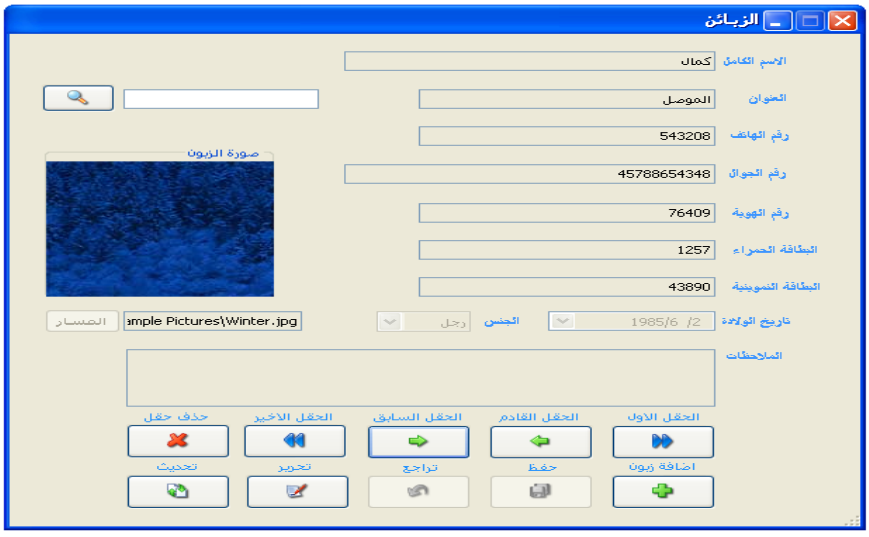
الشكل (5)

يجب إدخال اسم المستخدم وكلمة المرور القديمة ثم الجديدة ثم الضغط على زر موافق كما هو واضح بالشكل أعلاه ، وعند إدخال كلمة الولوج القديمة بشكل خاطئ ستظهر رسالة خطأ

واجهة إدارة المصرف:- تعتبر هذه الواجهة الواجهة الرئيسية للنظام حيث تتم فيها جميع العمليات المتعلقة بهذا النظام كما موضح في الشكل (6)

الشكل (6)

واجهة المعلومات الشخصية للزبائن :-كما موضح في الشكل (7)



الشكل (7)

في هذه الواجهة يجب تجنب إدخال زبونين بنفس رقم الهوية لان رقم الهوية يجب أن يكون غير مكرر لذلك إن أُدخل زبونين بنفس رقم الهوية تظهر رسالة خطأ الموضحة أدناه.

واجهة فتح الحساب :- كما موضح في الشكل (8)

الشكل (8)

هذه الواجهة تحتوي على قائمة الخيارات (Combo Box) حيث تحتوي هذه القائمة على كل الزبائن الذين قد تم إدخالهم من واجهة الزبائن وعند اختيار الزبون (المراد فتح الحساب له) فإن اسم الزبون ورقم هوية الزبون يكون قد ظهر على الواجهة ذاتياً، بعد ذلك يجب إدخال رقم الحساب وتكملة بقية المعلومات. ويجب إدخال رقم حساب مختلف عن أرقام الحسابات المخزونة مسبقاً (لا يمكن تكرار رقم الحساب في الجدول لأنه مفتاح رئيسي) وإلا فستظهر رسالة خطأ.

إذا كان للزبون وكلاء يجب الضغط على زر إضافة الوكلاء وستظهر واجهة أُخرى كما في الشكل (9)

الشكل (9)

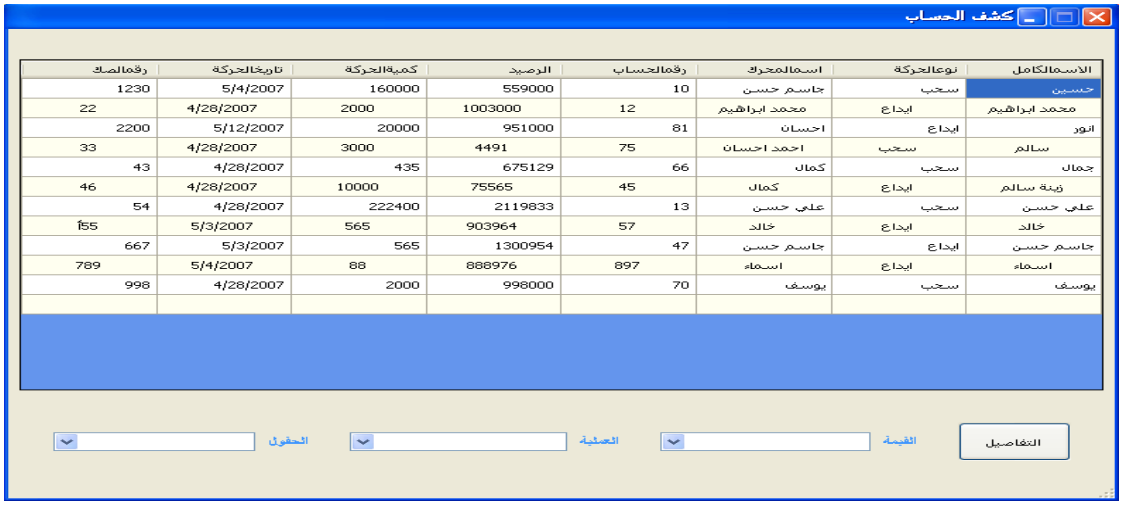
حيث تحتوي قائمة الخيارات (Combo Box) الأولى على أسماء الزبائن الذين تم فتح الحساب لهم (يجب فتح حساب للزبون قبل الدخول إلى هذه الواجهة لكي يظهر اسمه في قائمة الخيارات) أما الثانية فتحتوي على كل الزبائن الذين تم إدخالهم من واجهة الزبائن، يتم اختيار الوكيل (أو الوكلاء) من قائمة الخيارات الثانية والضغط على زر (إضافة الوكيل) سيظهر الوكيل في الـ (List Box) الظاهر في الشكل أعلاه وتكرار العملية ذاتها إذا كان هناك أكثر من وكيل ويمكن استخدام (حذف) لحذف الاسم من الـ (List Box) لتجنب الأخطاء.

واجهة الحركات:- كما موضح في الشكل (10)

الشكل (10)

وواجهة تمنع المستخدم من سحب مبلغ أكثر من المبلغ المتبقي من الرصيد بواسطة هذه الرسالة. "رصيدك لا يكفي لسحب هذا المبلغ".

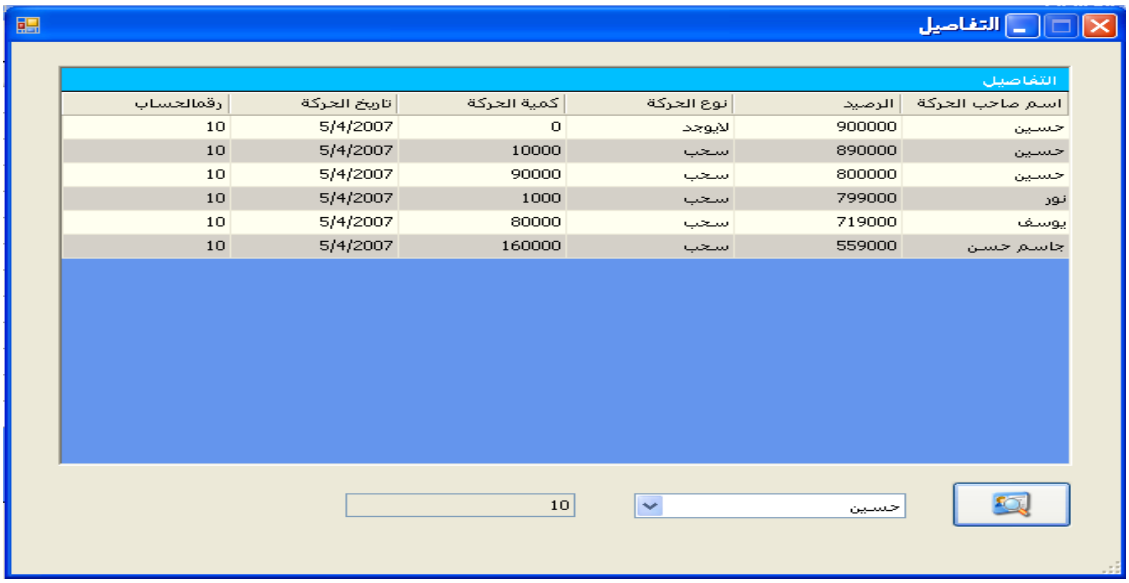
واجهة كشف الحساب:- كما موضح في الشكل (11)



الشكل (11)

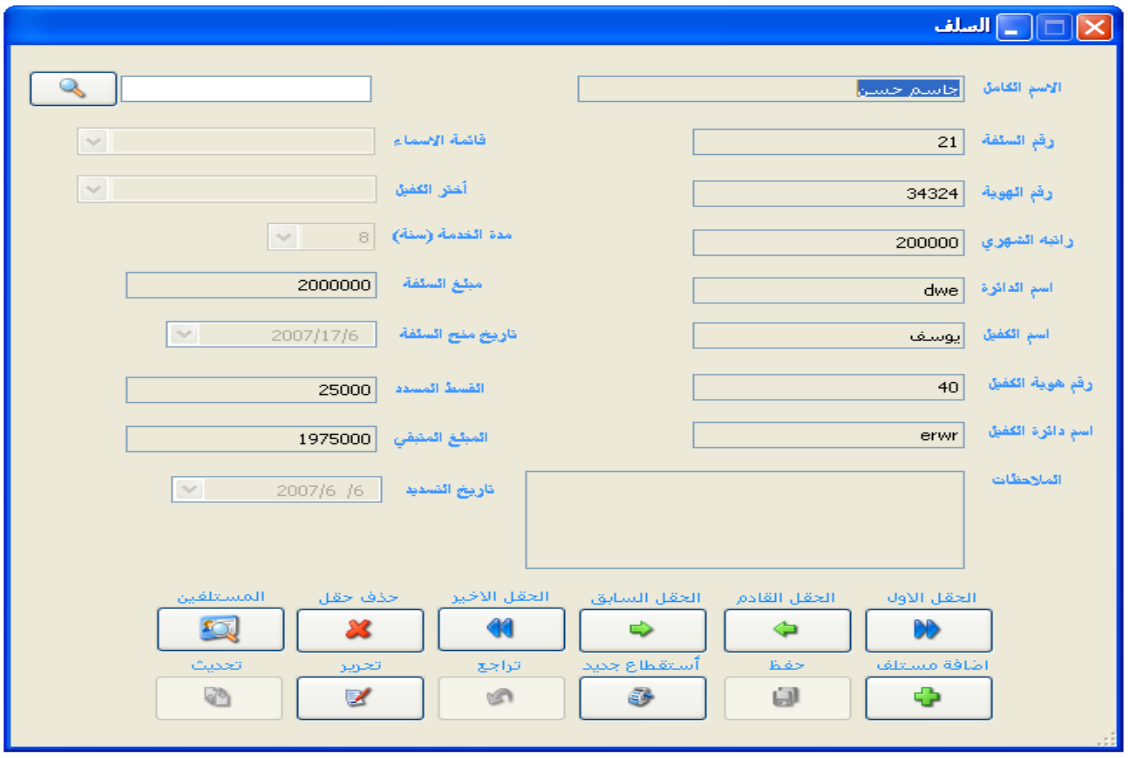
تحتوي هذه الواجهة على ثلاث قوائم اختيار (Combo Box) وضيفتها تقوم بترشيح (Filter) الـ View حيث يمكن عرض المعلومات المتوفرة في هذا الـ View حسب الحاجة حيث يمكن اختيار الحقل المراد من قائمة الحقول والعملية (Operator) المراد إجرائها على الحقل الذي تم اختياره ثم اختيار القيمة من القائمة الثالثة لذلك الحقل.

1- **واجهة التفاصيل:**- تعرض هذه الواجهة تفاصيل كل زبون على حدا مع عرض جميع الحركات للزبون و تاريخ كل حركة عند اختياره من قائمة الخيارات (Combo Box) ثم النقر على زر العرض كما هو موضح في الشكل (12)



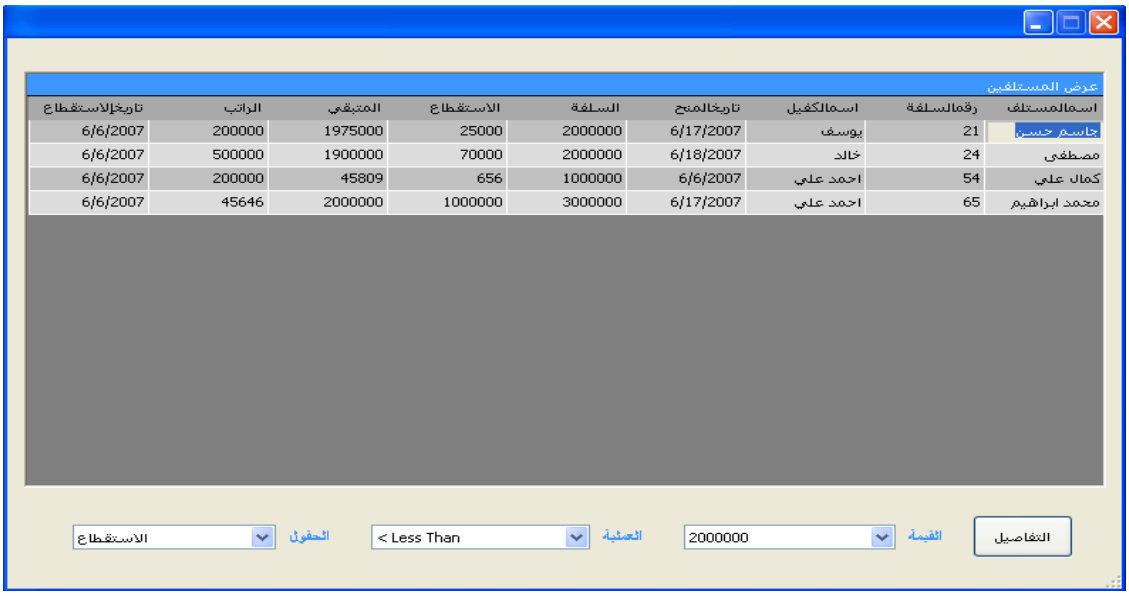
الشكل (12)

1- واجهة السلف :-موضح في الشكل (13)



الشكل (13)

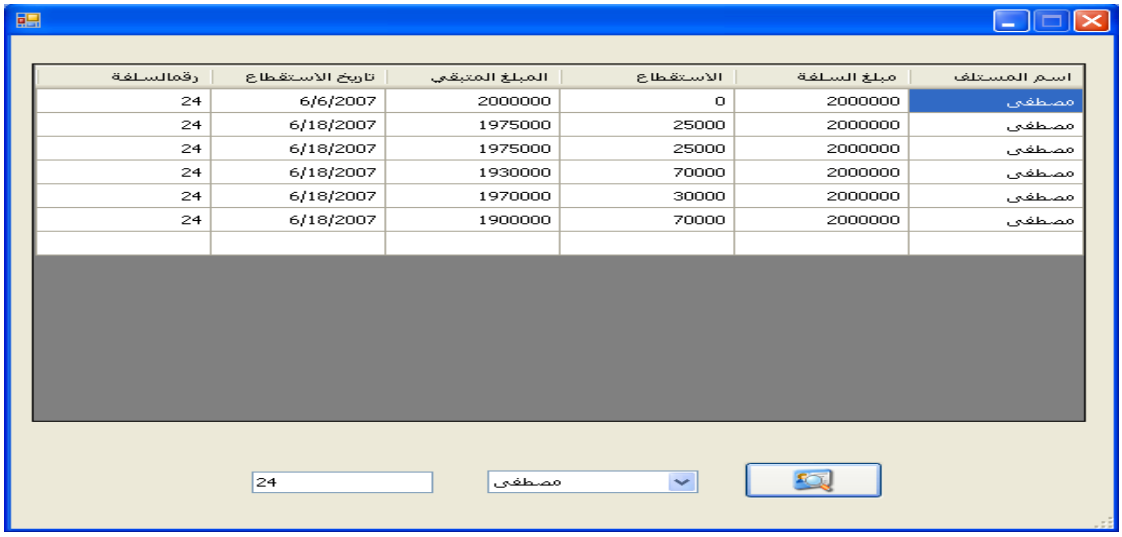
3- واجهة المستلفين:- تعرض هذه الواجهة آخر استقطاع جرى على المبلغ المتبقي لكل الزبائن كما هو موضح في الشكل (14)



الشكل (14)

في هذه الواجهة استخدم المرشح (Filter) كما في واجهة عرض الحركات حيث إن الترشيح في هذا الشكل هو لحقل الاستقطاعات والعملية اقل من (Less Than) والقيمة هي 2000000 حيث يتم عرض الاستقطاعات الأقل من 2000000 فقط.

4- واجهة تفاصيل المستلفين:- تعرض هذه الواجهة تفاصيل كل زبون على حدا مع جميع الاستقطاعات وتاريخ كل استقطاع عند اختياره من قائمة الخيارات (Combo Box) ثم النقر على زر العرض كما موضح في الشكل (15)



الشكل (15)

5- واجهة النسخة الاحتياطية:-كما موضح في الشكل (16)

الشكل (16)

6. الاستنتاجات

من خلال العمل على ADO.Net أصبح من الممكن تحقيق جميع الأهداف التي يطمح إليها لأجل بناء نظام يوفر الأمنية والسرعة اللازمة في خزن المعلومات وأيضا يوفر الدقة من خلال تقليل الأخطاء الناتجة من التدوين اليدوي لهذه المعلومات في السجلات التي تكون عرضة للتلف والتلاعب من قبل أشخاص غير مرغوب فيهم. وان الـ ado.net مصممة من الأساس للعمل في بيئة غير متصلة مثل الانترنت (ويمكنها العمل باتصال دائم مع قاعدة البيانات) وحاوي البيانات هو الكائن Dataset يمكن أن يحتوي على عدة مجموعات بيانات. وهي لا تستخدم المؤشرات فهي تعمل في بيئة غير متصلة بقاعدة البيانات والتي تلغي بطبيعتها الحاجة للمؤشرات وتخزن مجموعات بيانات ADO.net داخليا بهيئة XML العالمية تلك الهيئة المصممة لكي ترسل عبر جدران الحماية وعبر الشبكات دون مشاكل ويمكن لأي تطبيق قراءة البيانات بهيئة XMLبسهولة. ولقد صممت ADO.net لتعمل كنظام بيانات منفصل عن قاعدة البيانات فهي لا تتصل مع قاعدة البيانات إلا عند الضرورة وبالتالي لن تستهلك موارد النظام إلا عند الضرورة فقط.

المصادر

[1] Anders Hejlsberg, (2005), "introduction to C#", third addition copyright by Eric Gunnerson,nick wienholt, and
http//www.apress.com

[2] books.us/databasesystems-0002.php

[3] Dr.jiawei Han, Dr. Veronica Dahl, and Dr.Li-Yan Yuan"database systems and structures" OsmarR.Zaiane

[4] gale Coffman "sql server7 complete reference"

[5] Jeff ullman and Jennifer Widom "a first course in database systems" prentice-hall 2007 third addition

[6] msdn.microsoft.com/sql

[7] peter kitson "teach yourself sql in 21 day"

[8] Sams teach yourself ADO_NET in 21 days the relational guide to sql server reporting services

[9] Online: http://www.csharp-statin.com

[10] Online: http://www.functionx.com

[11] Online: http://www.arabmoheet.com

[12] Online: http://www. Teach Yourself Microsoft ADO_NET Step By Step.html

[13] Online: http://www.Whats new in ADO.NET.HTMl