مجلة تنمية الرافدين

**اسم المقال:** تحليل وتصميم نظام محاسبي وتصميمه بوساطة "دورة حياة بناء النظام"
**اسم الكاتب:** مضر زيد توفيق الدباغ
**رابط ثابت:** https://political-encyclopedia.org/index.php/library/3114
**تاريخ الاسترداد:** 2026/11/02 16:46 +03

الموسوعة السياسيّة هي مبادرة أكاديمية غير هادفة للربح، تساعد الباحثين والطلاب على الوصول واستخدام وبناء مجموعات أوسع من المحتوى العلمي العربي في مجال علم السياسة واستخدامها في الأرشيف الرقمي الموثوق به لإغناء المحتوى العربي على الإنترنت.
لمزيد من المعلومات حول الموسوعة السياسيّة – Encyclopedia Political، يرجى التواصل على info@political-encyclopedia.org

استخدامكم لأرشيف مكتبة الموسوعة السياسيّة – Encyclopedia Political يعني موافقتك على شروط وأحكام الاستخدام المتاحة على الموقع https://political-encyclopedia.org/terms-of-use

تم الحصول على هذا المقال من موقع مجلة تنمية الرافدين كلية الإدارة والاقتصاد / جامعة الموصل ورفده في مكتبة الموسوعة السياسيّة مستوفياً شروط حقوق الملكية الفكريّة ومتطلبات رخصة المشاع الإبداعي التي ينضوي المقال تحتها.

تنمية الرافدين ٨٥ (٢٩) ٢٠٠٧
ص ص [٢٠١-٢٢٤]

# تحليل وتصميم نظام محاسبي وتصميمه بوساطة "دورة حياة بناء النظام"

**مضر زيد توفيق الدباغ**
قسم علوم الحاسبات -كلية علوم الحاسبات والرياضيات
جامعة الموصل
mudhardz@yahoo.com

## المستخلص

**تعد عملية تحليل قاعدة البيانات وتصميمها من الأمور المهمة في مجال قواعد البيانات، اذ إنها تمثل العمود الفقري لأي قاعدة بيانات. وهذا البحث يمثل عملية تحليل نظام محاسبي للسيطرة على الديون والمخزن لمحل تجاري. لقد تم تطبيق هذا النظام في عدة محلات وشركات تجارية. يعتمد هذا النظام طريقة "دورة حياة بناء النظام" (System Development Life Cycle (SDLC)) المتبع مع الأنظمة غير المعقدة.**

**يتناول البحث تحليل وتصميم نظام في مجال الحسابات بالاعتماد على قاعدة البيانات. ومن ثم تنفيذ هذا التصميم باستخدام أحد نظم إدارة قاعدة البيانات (Database Management Systems (DBMS)) وفيه يمكن التعرف على كيفية توظيف أدوات تحليل النظم وقاعدة البيانات في تحليل وتصميم نظام متكامل، في هذا البحث تم استخدام نظام "FoxPro 2.6 for Windows" لإمكانياته الجيدة في تمثيل قاعدة البيانات في بيئة الويندوز.**

# Analysis and Design Accounting System by "System Development Life Cycle (SDLC)"

**Midhar Z. Tawfeq**
C.S. Dept. - College of C.S. and Math.
Mosul University

## Abstract

The operation database analysis and design is very important in the field of database. The current research represents the analysis and the design of accounting system in order to control the debts and stock of business store. This system applied in several stores and companies. The method "System Development Life Cycle" (SDLC) applied for analyzing and design the system used as simple systems.

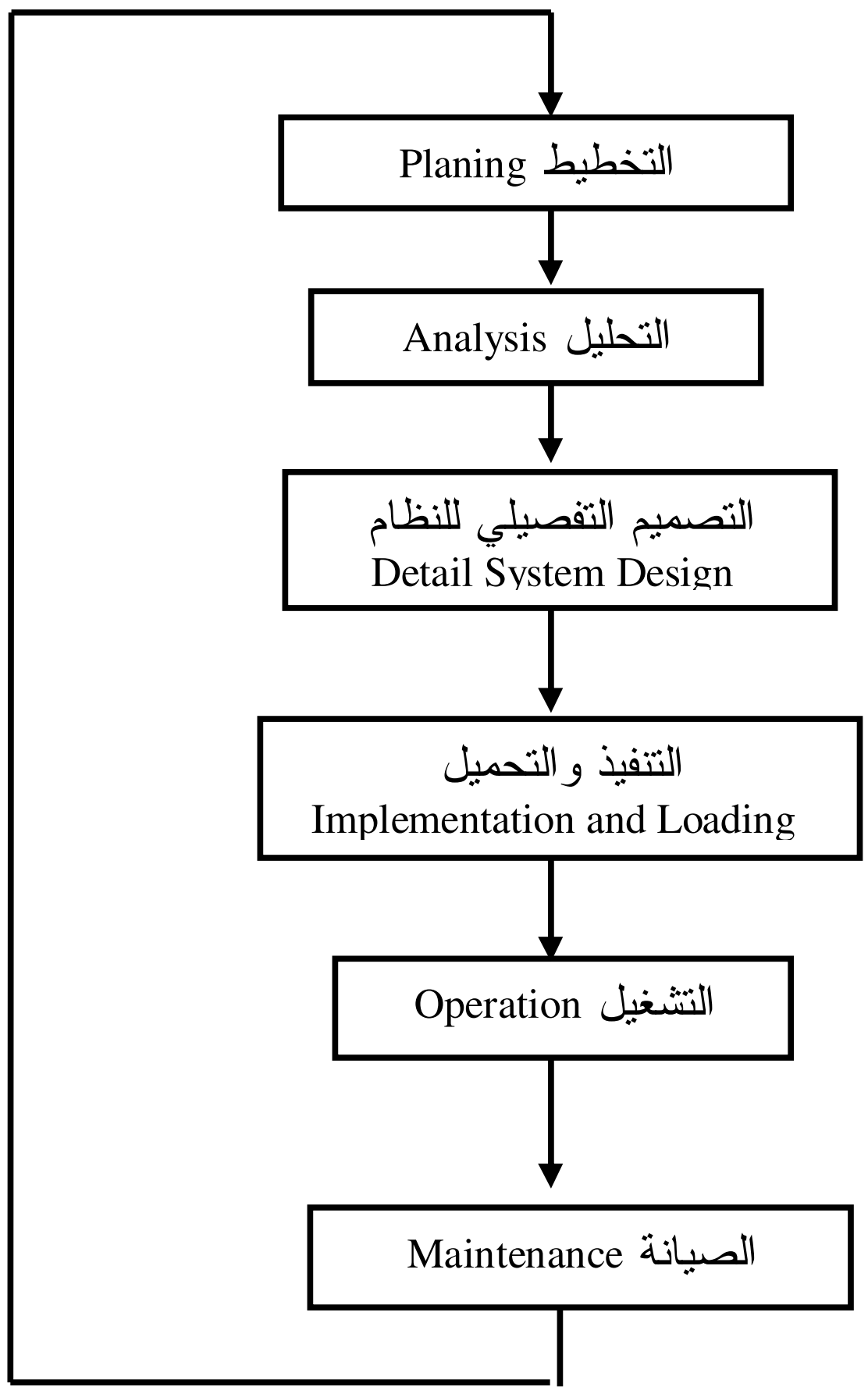
This is treated to the analysis and the design of database system. This research tries to give an idea about the method of designing the accounting system. As well as, the implementation of this design by using one of the Database Management Systems (DBMS). In this research, the system used was "FoxPro 2.6 for windows" because of its ability to represent database in Windows ecology.

تاريخ التسلم ٢٠٠٦/٥/٢٣ _____ تاريخ قبول النشر ٢٠٠٦/٧/١٢

**١. التمهيد**

للبدء بتصميم نظام قاعدة بيانات يجب أولاً تحديد الخطة العامة لهذه العملية متمثلة بالخطوات الرئيسة ضمن طريقة "دورة حياة تصميم النظام System Development Life Cycle (SDLC)" كما في الشكل ١ [١].

فمن المهم بعد دراسة كل الأدوات المستخدمة في مجال قواعد البيانات معرفة كيفية تشكيل هذه الأدوات ضمن سياق منظم، اذ يسهل استغلالها بشكل مفيد ومنتج للحصول على تصميم نظري كامل للنظام، ومن ثم تنفيذ هذا التصميم على الحاسبة باستخدام نظام ادارة قاعدة البيانات Database Management System (DBMS). وتم في هذا البحث استخدام نظام Foxpro 2.6 for windows.




**الشكل ١**
**دورة حياة بناء النظام**

### ٢. التخطيط Planning

قبل البدء بتصميم النظام الجديد يجب التخطيط لتصميم النظام المتمثل بالتقييم الأولي للنظام الحالي ودراسة الجدوى للنظام المقترح[١].

### ٢ -١. التقييم الأولي Initial Assessment

النظام الحالي هو نظام يدوي يتمثل بسجل الديون، اذ يكون لكل زبون صفحة تتضمن مبالغ القوائم والإيصالات وبعد ذلك يضاف كل مبلغ إلى المجموع أو يطرح من المجموع للحصول على المجموع الجديد.

يلاحظ صعوبة كتابة سجل الديون بالإضافة إلى الوقت اللازم لملء السجل. فضلاً عن أن عند حدوث أي خطأ في الحساب يجب إعادة العملية من جديد.

من نواقص النظام الحالي عدم وجود أي معالجة خاصة بالمخزن، اذ لا يمكن معرفة رصيد كل مادة والكمية المباعة والربح.

### ٢ -٢. دراسة الجدوى Feasibility Study

تكاليف النظام المقترح تتضمن ما يلي:

أ. جهاز حاسبة بمواصفات حدها الادنى: معالج بينتيوم٣ بسرعة ٥٠٠ ميكا هرتز، ذاكرة ١٢٨ميكا بايت، قرص صلب ١٠ كيكا بايت. سعر الجهاز كحد أدنى ٢٠٠ دولار أو أكثر بالاعتماد على المواصفات.

ب. جهاز طابعة نوع ليزر بسعر ١٣٠ دولار تقريباً.

ت. موظف يقوم بإدخال البيانات (القوائم والإيصالات) في حالة كون حجم البيانات كبير. هذا الموظف يتقاضى رتباً يتراوح بين ١٠٠٠٠٠ مائة الف دينار إلى ١٥٠٠٠٠ مائة وخمسين الف دينار.

أما فيما يتعلق بفوائد النظام المقترح فهي تتلخص بالنقاط الاتية:

أ. سهولة الحصول على الديون بعد إدخال القوائم والإيصالات.

ب. لكل زبون كشف حساب يتم الحصول عليه بسهولة وبشكل آني.

ت. كذلك يتم تحديد لكل مادة: الرصيد المتبقي منها، كلفة القطعة الواحدة، القيمة الكلية للمادة، كلفة المخزن بالكامل. فضلاً عن الكمية المباعة من كل مادة مع سعر البيع وسعر الشراء والفرق بينهما (الربح).

ث. لكل مادة هنالك كشف بالحركة التي أجريت على المادة من شراء وبيع ومقدار الربح المترتب على البيع ولأي مدة زمنية يمكن تحديدها.

ج. يتم إعطاء الحركة النقدية مع رصيد الصندوق بعد كل حركة.

ح. يتم إعطاء المبيعات والمشتريات مع الربح المترتب عليها خلال مدة معينة.

خ. عند حدوث أي خطأ في القوائم أو الإيصالات يمكن إصلاح الخطأ بسهولة وتلقائياً يتم تحديث الديون والمخزن والكشف الخاص بكل زبون وكل مادة فضلاً عن تحديث رصيد الصندوق.

نلاحظ من هذه الفوائد المترتبة من استخدام النظام المقترح أنها أكثر بكثير من التكاليف فبتجنب خطأ بسيط يحدث بسبب النظام الحالي يمكن توفير آلاف الدولارات.

**٣.التحليل Analysis**

بعد التخطيط لتصميم النظام والتأكد من قبول النظام والموافقة عليه من قبل إدارة الشركة أو صاحب المحل التجاري، يمكن البدء بتحليل النظام المقترح من خلال الخطوات الاتية [١]:

**٣ -١. متطلبات المستخدم User Requirements**

بعد إجراء مقابلة مع المستخدمين يتم تحديد كل ما يحتاج إليه المستخدم فيما يخص النظام. وهذه المتطلبات تمثل مخرجات النظام، وكما يأتي: الديون، المخزن، الصندوق، الزبون، والمادة

أ. الديون تتضمن ما يأتي: اسم المدين أو الدائن مع الرصيد بالعملتين الدينار والدولار.

ب. المخزن تتضمن ما يأتي: إسم المادة، المتبقي من المادة، تكلفة الوحدة، المبلغ الكلي، الكمية المباعة، سعر البيع، المبلغ الكلي، سعر الشراء، المبلغ الكلي، الربح الكلي، ربح القطعة الواحدة، والنسبة.

ت. الرصيد ويتضمن ما يأتي: المبلغ الوارد، المبلغ الصادر، نوع الحركة، رقم القائمة، تاريخ القائمة، والملاحظات.

ث. كشف الزبون ويتضمن ما يأتي: المبلغ، نوع الحركة، رقم القائمة، تاريخ القائمة، الملاحظات، وإسم الحساب المقابل في الحركة. كما يضاف في الكشف التفصيلي: اسم المادة، العدد، السعر.

ج. كشف المادة ويتضمن ما يأتي: اسم الزبون الذي قام بالشراء أو البيع، نوع الحركة شراء أم بيع، رقم القائمة، تاريخ القائمة، عدد القطع، سعر القطعة الواحدة، المبلغ، سعر الشراء، الربح، الملاحظات.

ح. المشتريات تتضمن ما يأتي: التاريخ، رقم القائمة، المادة، العدد، السعر، والمبلغ.

خ. المبيعات تتضمن ما يأتي: التاريخ، رقم القائمة، المادة، العدد، السعر، المبلغ، سعر الشراء، والربح.

**٣ -٢. تقييم النظام الحالي Existing System Evaluation**

يحتوي النظام الحالي على عدد من المشاكل فيما يأتي يتم ذكرها:

أ. طول الوقت وصعوبة حساب رصيد كل زبون. حيث يجب إضافة مبلغ قائمة البيع ومبلغ إيصال الدفع إلى رصيد الزبون، وطرح مبلغ قائمة الشراء ومبلغ إيصال الاستلام من رصيد الزبون.

ب. عند حدوث أي خطأ في الحساب بسبب خطأ في الجمع أو الطرح أو خطأ في ترحيل القوائم والإيصالات يجب حذف جميع العمليات بعد الخطأ وإعادة حسابها، وذلك لأن الخطأ سوف يؤثر على العمليات التي تليه في الكشف.

ت. لا يمكن حساب رصيد المادة اذ إن هذه العملية تكون صعبة ومجهدة، لذلك لا يتم إجراءها، ومن ثم لا يمكن اكتشاف أي نقص في أرصدة المواد عند إجراء الجرد.

ث. ليس هنالك أي سيطرة على حركة المواد ومعرفة المواد الأكثر مبيعاً لغرض التركيز عليها.

ج. ليس من الممكن حساب ربح كل مادة من خلال جرد مشتريات المادة ومبيعاتها وحساب الفرق بين المشتريات والمبيعات لهذه المادة، لأن هذه العملية أيضاً من الصعب إنجازها.

ح. حساب رصيد الصندوق يستغرق وقتاً طويلاً مع احتمالية وقوع أخطأ أثناء عملية التسجيل. فضلاً عن أن الإيصالات يجب أن ترحل إلى حساب الزبون وحساب الصندوق أيضاً في الوقت نفسه.

خ. عند إجراء أي تعديل على القوائم أو الإيصالات، من صعب بمكان تحديث الصفحة الخاصة بالزبون فيما يخص القائمة بالاجل والإيصال. وكذلك فيما يخص رصيد الصندوق. هذا فضلاً عن أن المواد لا يوجد أي كشف خاص بها.

كما لاحظنا فيما سبق المشاكل الكثيرة للنظام الحالي التي سوف يتم معالجتها من خلال النظام المقترح. وفيما يأتي التصميم المنطقي للنظام المقترح.

**٣-٣. تصميم النظام المنطقي Logical System Design**

بعد تحديد مشاكل النظام الحالي، يمكن البدء بتصميم النظام المقترح. اذ يجب تجنب جميع المشاكل المذكورة سابقاً، و إلا يعد النظام المقترح غير مفيد. بالاعتماد على متطلبات المستخدم (مخرجات النظام المقترح) [١].

قبل البدء بالتصميم المنطقي للنظام يجب تحديد نظام ادارة قاعدة البيانات المستخدم. وفي هذا البحث تم استخدام نظام "FoxPro 2.6 for Windows" لإمكانياته في تمثيل قاعدة البيانات ذات الحجم الكبير، والفصل بين البيانات والتطبيق ومن ثم يمكن تكوين نسخة احتياطية من البيانات.

وحسب مفهوم دورة حياة بناء النظام هنالك عدة مستويات او مراحل يجب المرور عليها وهي[١]:

**٣-٣-١. المستوى المفاهيمي Conceptual Level**

يتمثل هذا المستوى بمخطط الكينونة والعلاقة Entity-Relationship Diagram (ERD) من خلال المتطلبات المذكورة سابقاً نقوم بتحديد الكينونات الرئيسة التي يجب توافرها لغرض الحصول على هذه المتطلبات (المخرجات).

وهذا عرض لهذه الكينونات:

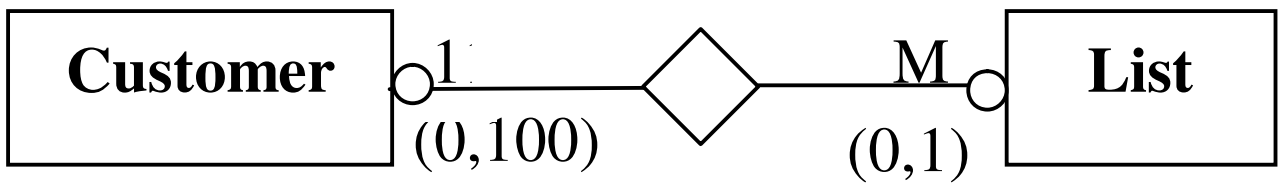
أ. List: تمثل القوائم والإيصالات. تستخدم في حساب الديون والصندوق المخزن والمبيعات والأرباح.

ب. Customer: تمثل الزبائن. تستخدم لغرض حساب الديون.

ت. Subject: تمثل المواد. تستخدم لغرض حساب المخزن والمبيعات والأرباح.

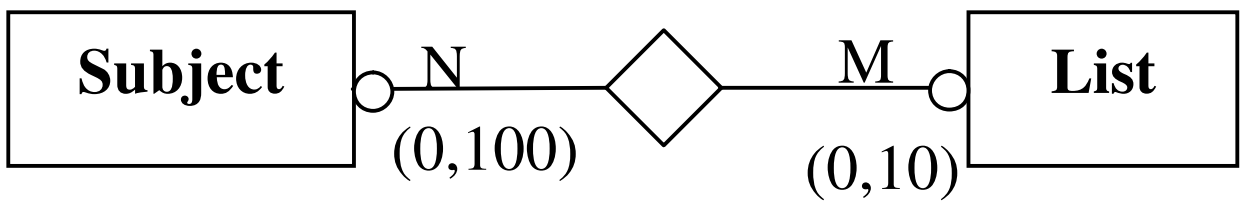
بعد تحديد الكينونات الرئيسة يجب وضع القواعد التي يتم من خلالها الربط بين هذه الكينونات ضمن ERD، الذي يتضمن: الترابط Connectivity وهي تكون (1:1, 1:M, M:1, M:N)، درجة العلاقة Cardinality التي تمثل الحد الأدنى والأعلى من عدد القيود التي ترتبط بالكينونة الآخر، ومشاركة العلاقة Relationship Participation التي تحدد نوع العلاقة هل هي اختيارية Optional (دائرة) أم إجبارية Mandatory (بدون دائرة) [٤]. وكما يأتي:

**القاعدة١:** لكل قيد من List قيد واحد من Customer، وللقيد الواحد من Customer أكثر من قيد من List كما في الشكل ٢.



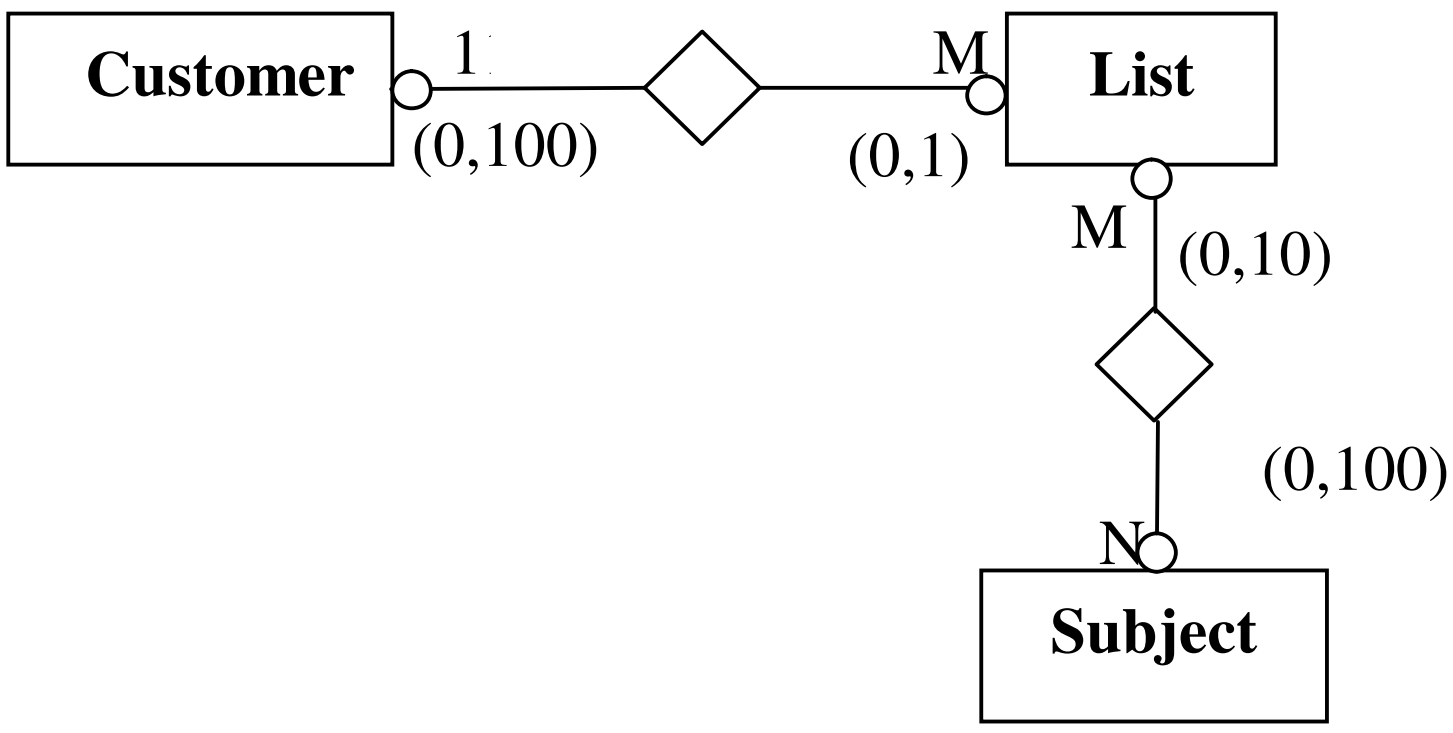

**الشكل ٢ ERD للقاعدة١**

**القاعدة٢:** لكل قيد من List أكثر من قيد من Subject، وللقيد الواحد من Subject أكثر من قيد من List كما في الشكل ٣.




**الشكل ٣ ERD للقاعدة٢**

الشكل ٤ يمثل ERD للمستوى المفاهيمي.



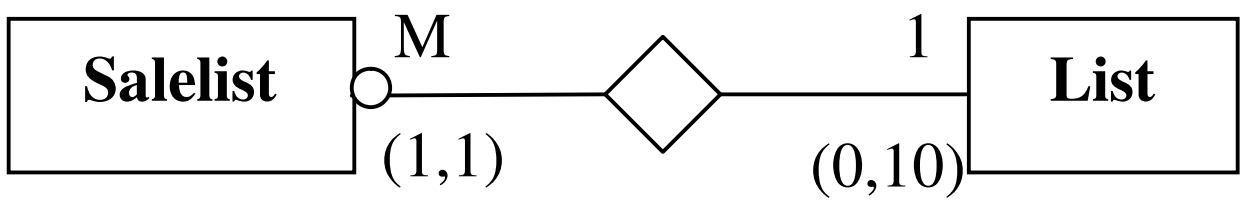

**الشكل ٤ ERD للمستوى المفاهيمي**

### ٣ -٣ -٢ المستوى الداخلي Internal Level

في هذا المستوى يتم تحويل الــ ERD الى مجموعة من الجداول. للقيام بذلك يجب تحويل أي علاقة من نوع M:N إلى نوع 1:M، لأن هذه العلاقة لا يمكن تمثيلها ضمن جدول.

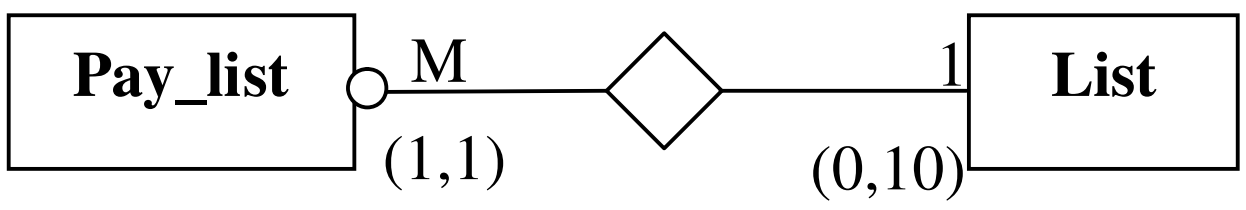
القاعدة ٢ تحتوي على علاقة M:N التي يجب تحويلها إلى علاقتين من نوع 1:M.[٤] فيما يأتي عملية التحويل:

**القاعدة٣:** لكل قيد من List أكثر من قيد من Salelist ولكل قيد من Salelist قيد واحد من List كما في الشكل ٥. Salelist تمثل فقرات قوائم البيع.



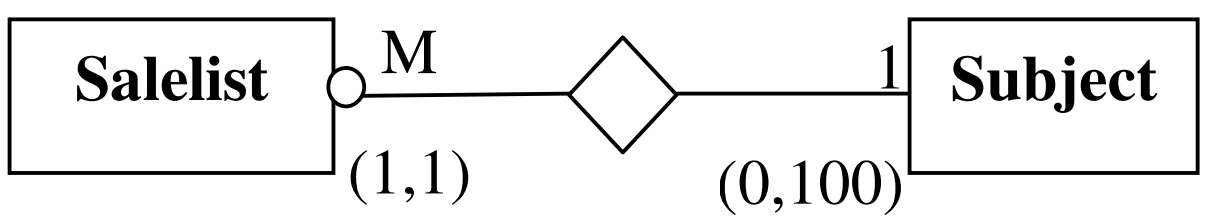

**الشكل ٥ ERD للقاعدة ٣**

**القاعدة٤:** لكل قيد من List أكثر من قيد من Pay_list ولكل قيد من Pay_list قيد واحد من List كما في الشكل ٦. Pay_list تمثل فقرات قوائم الشراء.



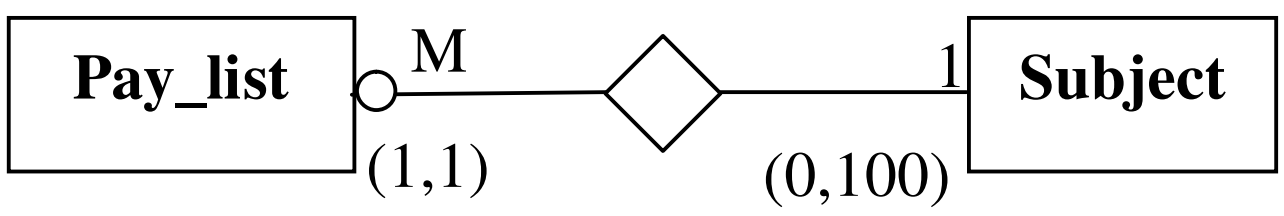

**الشكل ٦ ERD للقاعدة ٤**

**القاعدة ٥:** لكل قيد من Subject أكثر من قيد من Salelist ولكل قيد من Salelist قيد واحد من Subject كما في الشكل ٧.



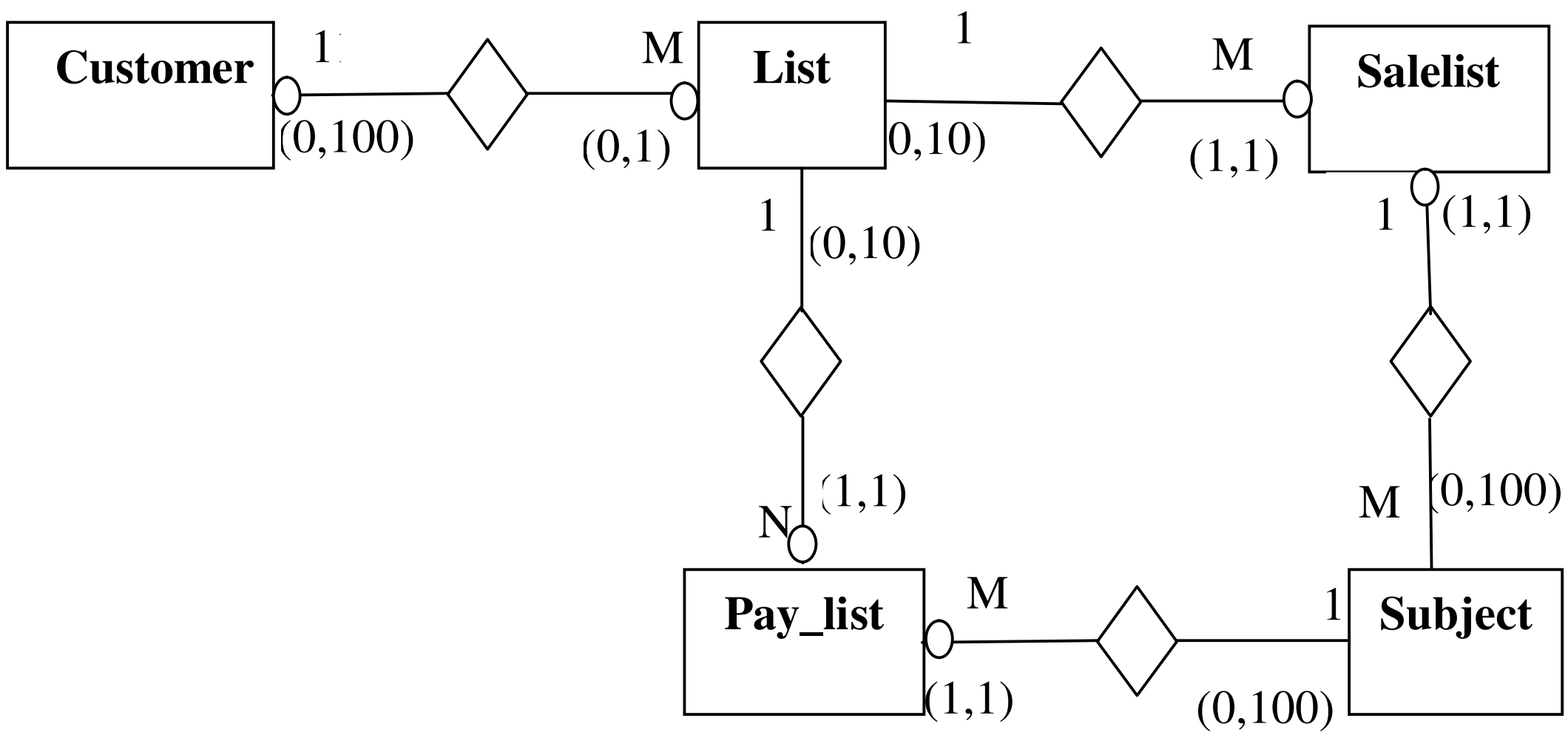

**الشكل ٧ ERD للقاعدة ٥**

**القاعدة ٦:** لكل قيد من Subject أكثر من قيد من Pay_list ولكل قيد من Pay_list قيد واحد من ubject كما في الشكل ٨.




**الشكل ٨ ERD للقاعدة ٦**

الشكل ٩ يمثل ERD النهائي للمستوى الداخلي.




**الشكل ٩ ERD المستوى الداخلي للقاعدة**

بعد الانتهاء من قواعد المستوى الداخلي يجب كتابة قاموس البيانات Data Dictionary وكما في الجدول ١.

**الجدول ١أ**

**قاموس البيانات للجدول List**

| اسم الجدول | اسم الحقل | النوع | الوصف | المفتاح | المرجع |
|---|---|---|---|---|---|
| List | Code | N(10) | رمز القائمة | أساسي | |
| | List_code | N(10) | رقم القائمة | | |
| | Customer | N(8) | رمز الزبون | أجنبي | Customer |
| | List_date | D(8) | تاريخ القائمة | | |
| | TIME | C(8) | وقت القائمة | | |
| | OUTTIME | C(9) | وقت القائمة للعرض | | |
| | LIST_TYPE | N(14) | نوع القائمة | | |
| | CURRENCY1 | L(1) | العملة للمبلغ الاول | | |
| | MONY1 | N(14.2) | المبلغ الاول | | |
| | PAYING | L(1) | نوع الدفع | | |
| | CURRENCY2 | L(1) | العملة للمبلغ الثاني | | |
| | MONY2 | N(14,2) | المبلغ الثاني | | |
| | RECEIVER | N(6) | الحساب المقابل | أجنبي | Customer |

يتبع ⇐

ماقبله ⟸

| | | | | | |
|---|---|---|---|---|---|
| | NOTE | C(60) | الملاحظات | | |
| | DOLARPRICE | N(4) | سعر الدولار | | |
| | **المجموع** | 156 | | | |

## الجدول ١ب
## ب قاموس البيانات للجدول SaleList

| اسم الجدول | اسم الحقل | النوع | الوصف | المفتاح | المرجع |
|---|---|---|---|---|---|
| SaleList | List_code | N(10) | رمز القائمة | أجنبي | List |
| | Subject | N(6) | رمز المادة | اجنبي | Subject |
| | Number | N(10,3) | الكمية | | |
| | Unitprice | N(14,3) | السعر | | |
| | **المجموع** | 40 | | | |

## الجدول ١ت
## قاموس البيانات للجدول Pay_List

| اسم الجدول | اسم الحقل | النوع | الوصف | المفتاح | المرجع |
|---|---|---|---|---|---|
| Pay_List | List_code | N(10) | رمز القائمة | أجنبي | List |
| | Subject | N(6) | رمز المادة | أجنبي | Subject |
| | Number | N(10,3) | الكمية | | |
| | Unitprice | N(14,3) | السعر | | |
| | **المجموع** | 40 | | | |

## الجدول ١ث
## قاموس البيانات للجدول Customer

| اسم الجدول | اسم الحقل | النوع | الوصف | المفتاح | المرجع |
|---|---|---|---|---|---|
| Customer | Code | N(8) | رمز الزبون أو الحساب | اساسي | |
| | Name | C(50) | اسم الزبون | | |
| | Orgin1 | N(10) | الرصيد القديم من الدينار | | |
| | Orgin2 | N(11,3) | الرصيد القديم من الدولار | | |
| | Constant | L(1) | قابل للمسح ام لا | | |
| | Important | N(1) | نوع الزبون | | |
| | **المجموع** | 81 | | | |

**الجدول ١جـ**
**قاموس البيانات للجدول Subject**

| اسم الجدول | اسم الحقل | النوع | الوصف | المفتاح | المرجع |
|---|---|---|---|---|---|
| Subject | Code | N(6) | رمز المادة | اساسي | |
| | Name | C(40) | اسم المادة | | |
| | Item_no | C(10) | PNللمادة | | |
| | Number | N(10,3) | الكمية القديمة من المادة | | |
| | Unitcost | N(14,3) | كلفة الكمية القديمة من المادة | | |
| | Type | L(1) | نوع العملة (دينار/دولار) | | |
| | Constant | L(1) | قابل للمسح ام لا | | |
| | **المجموع** | 82 | | | |

### ٣-٣-٣ المستوى الخارجي External Level

في هذا المستوى يتم تحويل متطلبات المستخدم (المذكورة في الفقرة ٣-١) الى مجموعة من عروض خارجية External Views، كل عرض يتكون من جزءين. الأول هو مخطط ERD يحتوي على الكينونات المستخدمة في العرض. أما الثاني فهو مجموعة استعلامات Queries، وكل استعلام يمثل ايعاز SQL. وقد تم استخدام الاستعلامات التالية [٤]:

أ. **الديون Payment**: للحصول على الديون نحتاج الى الكينونات List,Cusomter في الشكل ٤.

- **الاستعلام ١ Payment**: في هذا الاستعلام يتم حساب رصيد الحساب بالدينار باستعمال الدالة Balance1 وبالدولار باستعمال الدالة Balance2 مع اسم الزبون. يدخل في حساب الرصيد مبلغ البيع والشراء، وكذلك مبلغ وصل الاستلام ووصل الدفع، فضلاً عن ارباح قوائم البيع.

ب. **كشف الزبون Customer**: للحصول على كشف زبون نحتاج الى الكينونات List,Cusomter في الشكل ٤.

- **الاستعلام ٢ Amount1**: لغرض ربط الكينونة list مع الكينونة Customer وتكرار list اذا كان الحقل Customer والحقل Receiver يساوي M.customer (الحساب المطلوب كشفه). الجدول type1 لغرض تكرار List. كما في الشكل ١٠ [٣].

```
SELECT list_fields,;
   IIF(type1.i=1,customer,00000000) AS customer,;
   IIF(type1.i=2 OR receiver!=customer,;
                  receiver,00000000) AS receiver,;
   customer.name AS customername ;
 FROM list,customer1 customer,type1 ;
 WHERE (list.customer = m.customer OR ;
    list.receiver = m.customer) AND ;
    (type1.i=1 OR list.receiver = list.customer ) AND ;
    customer.code=receiver1(list.code) AND ;
    list.list_date<=m.edate ;
 INTO CURSOR amount1;
 ORDER BY list.list_date, list.time
```

**الشكل ١٠ الاستعلام2 Amount1**

- **الاستعلام٣ Amount21**: لاضافة الرصيد القديم للزبون بالدينار والدولار الى القوائم المتمثلة بالاستعلام ٢ amount1 عن طريق **UNION**.
- **الاستعلام ٤ Amount2**: يقوم بجمع القوائم التي تسبق تاريخ بدء الكشف m.bdate بوصفه الرصيد القديم وعرض جميع القيود التي تلي تاريخ بدء الكشف التي تمثل تفاصيل الكشف باستخدام **UNION**.
- **الاستعلام ٥ Amount3**: للتأكد من أن المبلغ الأول (المدين) يضم الأرقام الموجبة فقط، والمبلغ الثاني (الدائن) يضم الأرقام السالبة فقط.
- **الاستعلام ٦ Amount**: يمثل الصورة النهائية للكشف التي تضم عملة المدين (currency1)، مبلغ المدين (mony1)، عملة الدائن (currency2)، مبلغ الدائن (mony2)، تاريخ القائمة (list_date)، نوع القائمة (type)، رقم القائمة (list_code)، و في النهاية الرصيد بالدينار (total1) وبالدولار (total2) الذي يتم حسابه في هذا الاستعلام من خلال ربط الاستعلام السابق مع نفسه بالاعتماد على تاريخ القائمة ورقم القائمة كما في الشكل ١١.

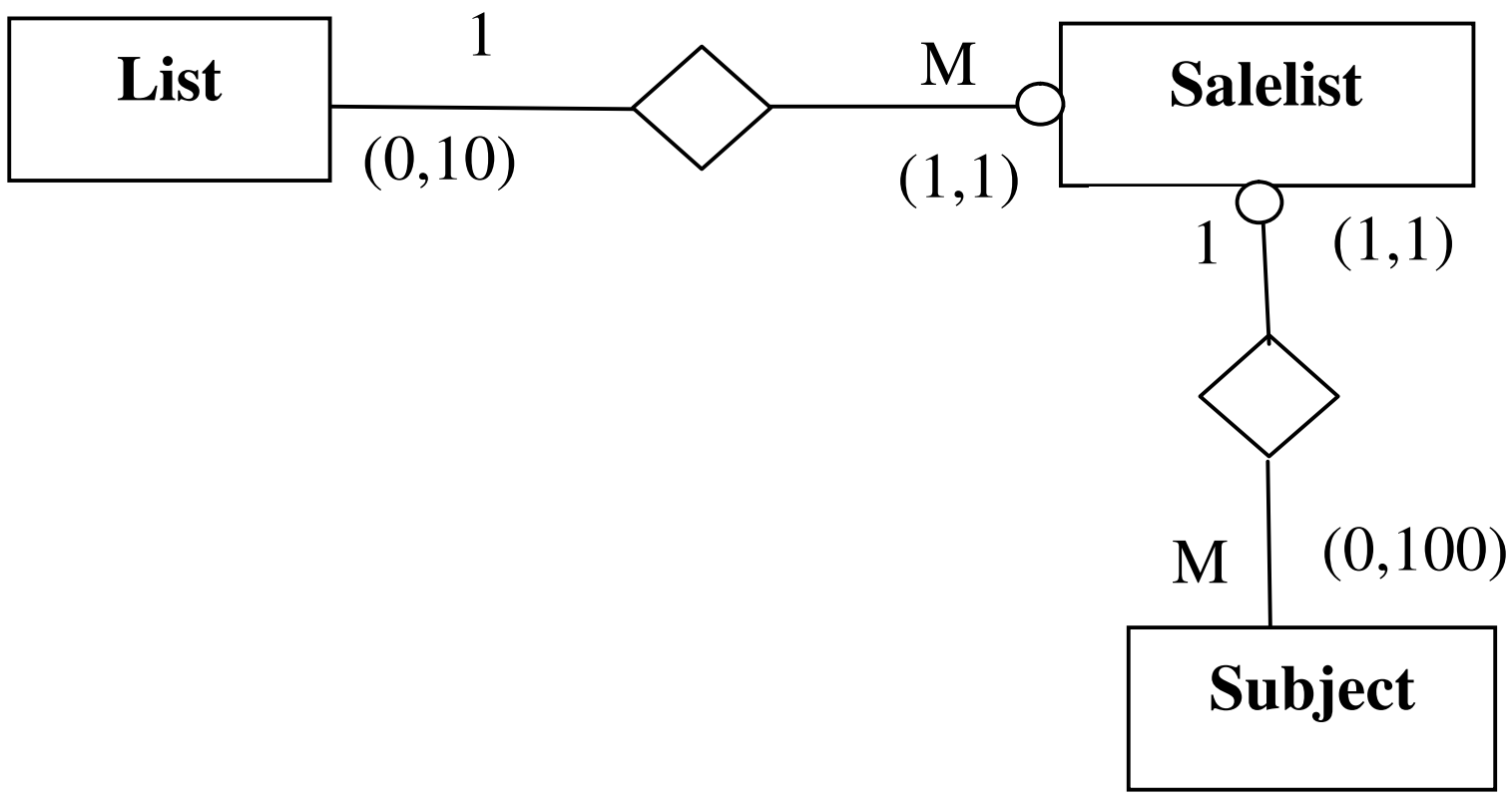
```
SELECT amount3_fields,;
   round(SUM(IIF(amount4.currency1="Id",;
         amount4.mony1,0000000000.00000)+;
             IIF(amount4.currency2="Id",;
         amount4.mony2,0000000000.00000)),0);
                                 AS total1,;
   round(SUM(IIF(amount4.currency1=" $",;
         amount4.mony1,0000000000.00000)+;
             IIF(amount4.currency2=" $",;
         amount4.mony2,0000000000.00000)),;
                        pricedec) AS total2;
 FROM amount3, amount3 amount4;
 WHERE (amount3.list_date>amount4.list_date;
     OR amount3.list_date=amount4.list_date;
    AND amount3.code>=amount4.code);
 GROUP BY amount3.code;
 ORDER BY list_date, code, type;
 INTO CURSOR amount
```

**الشكل ١١ الاستعلام ٦ Amount**

**ت. كشف تفصيلي للزبون CustDtl**: للحصول على كشف تفصيلي للزبون نحتاج الى جميع كينونات قاعدة البيانات كما في الشكل ٩.

- **الاستعلام ٧ Listratio**: لغرض حساب نسبة الخصم او الزيادة في المبلغ لكل قائمة. تم استخدام الكينونة List والكينونة Salelist في الجزء الاول من الاستعلام الخاص بالمبيعات. أما الجزء الثاني فهو مشابه للأول مع اختلاف استعمال الكينونة Pay_list بدلاً من Salelist. باستخدام **UNION** قمنا بالربط بين الجزءين بحيث نحصل على ناتج يمثل القوائم مع نسبة الخصم او الزيادة في كل قائمة.
- **الاستعلام ٨ List1**: يقوم باسترجاع تفاصيل القائمة (شراء او بيع) بالاضافة الى حساب السعر الحقيقي بعد تطبيق النسبة المأخوذة من الاستعلام ٧ Listratio.
- **الاستعلام ٩ Amount4**: يقوم بالربط بين الاستعلام ١١ amount3 والاستعلام ١٥list1 الذي يضم تفاصيل القائمة.
- **الاستعلام ١٠ Amount**: يمثل الصورة النهائية للكشف التفصيلي كما في الاستعلام ٦ مع اضافة: رقم المادة (subject)، الكمية (number)، سعر القطعة الواحدة (unitpirce)، العملة (currecny3)، اسم المادة (subjectname)، ورمز المادة (item_no). في هذا الاستعلام تم استخدام **SUM** و **GROUP BY**.

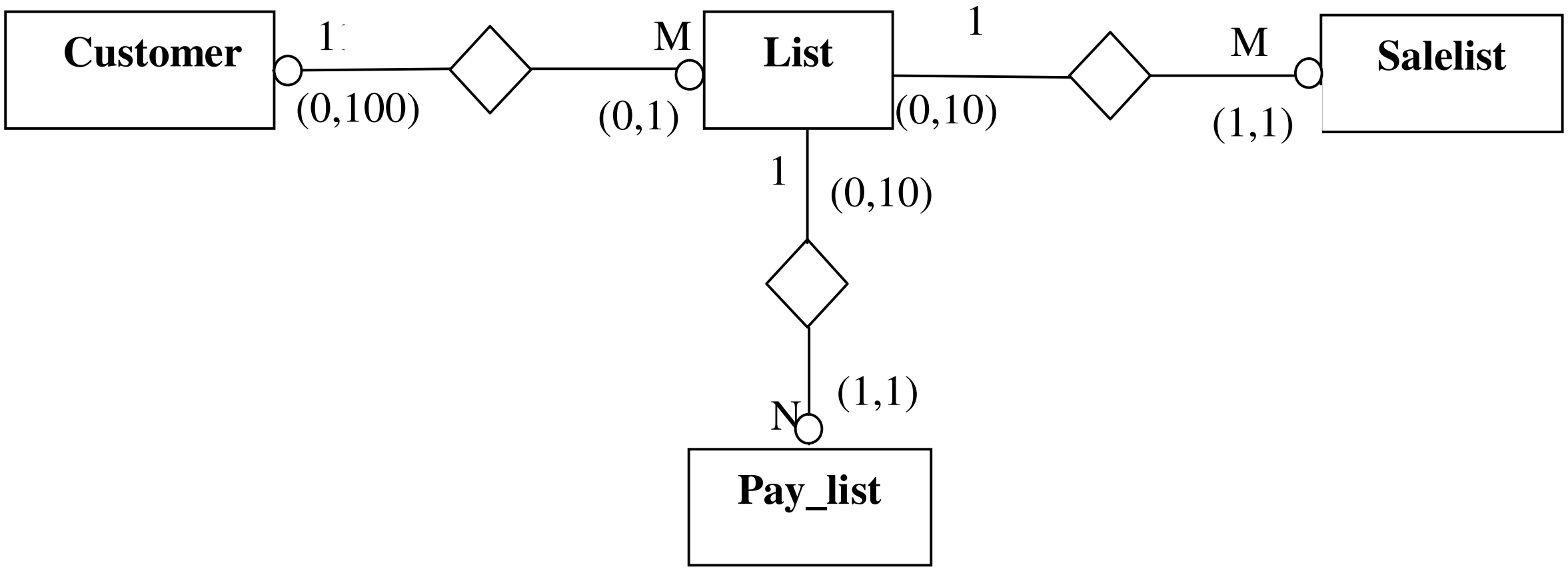
ث. **المخزن ViwStock**: يتضمن هذا العرض أسماء المواد مع الرصيد، المبيعات، والأرباح لكل مادة. في هذا العرض يتم استخدام الكينونات List، Salelist، وSubject كما في الشكل ١٢.




**الشكل ١٢ ERD العرض ViwStock**

- **الاستعلام ١١ Subjectsum**: لحساب مجموع المبيعات من كل مادة بالدينار والدولار. يستخدم في هذا الاستعلام **SUM** مع **GROUP BY**، فضلاً عن **UNION**.
- المتغيران m.bdate و m.edate تمثل المدة الزمنية للتقرير.
- **الاستعلام١٢ Stock1**: لحساب الرصيد (epiece) وكلفة القطعة الواحدة (eunitcost) واجمالي الكلفة بالدينار(etotalid) والدولار(etotalud) لكل مادة، فضلاً عن المبيعات التي تم حسابها في الاستعلام ١٨ Subjectsum. كذلك يتم حساب كلفة الكمية المباعة من المادة (tcostid,tcostud).
- الدالة piece لحساب الرصيد مع الكلفة. الدالة subcost لحساب كلفة المبيعات في هذا العرض يتم استخدام البرمجة الاجرائية (Procedural Language) لحساب الرصيد والكلفة، بخلاف العروض السابقة، اذ لايمكن حسابها باستخدام لغة استرجاع البيانات SQl التي تسمى اللغة غير الاجرائية (Non Procedural Language).
- **الاستعلام١٣ Stock2**: لحساب الربح بطرح الكلفة من المبيعات (tsaleid-tcostid)، (tsaleud-tcostud) بالاعتماد على الاستعلام١٢.
- **الاستعلام ١٤ Stocklist**: لحساب نسبة الربح لكل مادة من خلال قسمة الربح على الكلفة مضروباً في مائة .
- **الاستعلام١٥ Stocklist**: الاستعلام النهائي ويتم من خلاله حذف المواد التي ليس فيها رصيد وليس فيها مبيعات في نفس الوقت.

ج. **كشف المادة SubStock**: يقوم هذا العرض بإصدار كشف خاص بالمادة يتضمن عملية الشراء والبيع التي قد تمت على المادة المطلوبة. ويستخدم في هذا العرض الكينونات List, SaleList, PayList, and Customer كما في الشكل ١٣.




**الشكل ١٣ ERD العرض**
**SubStock**

---

**الاستعلام ١٦ Orgin**: من خلال هذا الاستعلام يتم تضمين الرصيد القديم من المادة عند بدء العمل بالنظام.

- **الاستعلام١٧ Stk1**: في هذا الاستعلام يتم ربط الرصيد القديم (الاستعلام ١٦ Orgin) مع المشتريات والمبيعات باستعمال **UNION**.
- **الرصيد Mony**: من خلال هذا العرض يتم الحصول على كشف يضم الحركة النقدية للشركة من استلام مقبوضات او مدفوعات. وتم استخدام الكينونتين Customer و list في هذا العرض كما في الشكل ٢.
- **الاستعلام١٨ Mony**: الاستعلام الوحيد المستخدم في هذا العرض والذي يضم كشف الحركة النقدية ضمن المدة الزمنية المحددة من خلال المتغيرين m.bdate وm.edate.

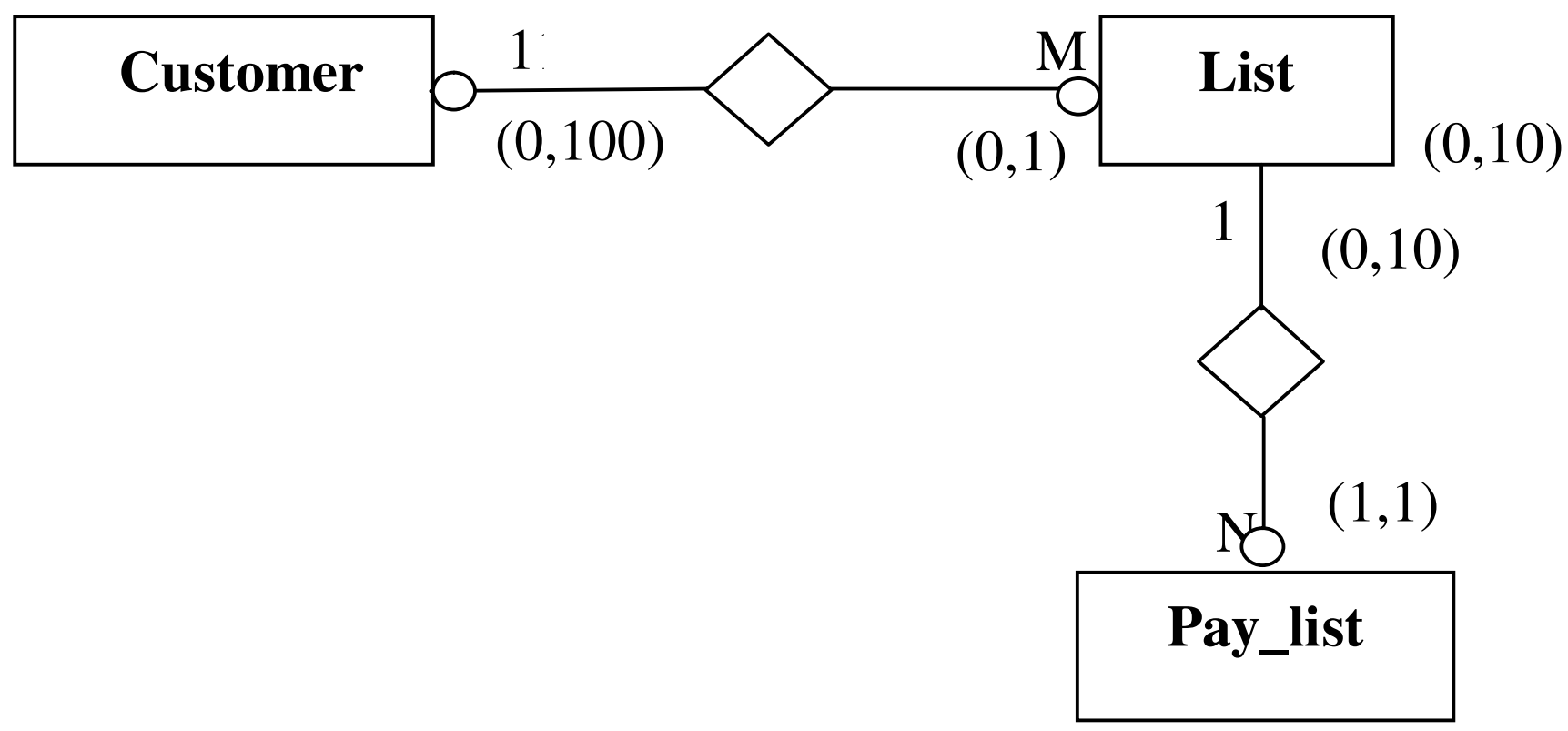
ح. **المبيعات Sales**: تم بناء هذا العرض لغرض الحصول على كشف المبيعات مرتباً حسب التأريخ، ويتم من خلاله تدقيق قوائم البيع الداخلة الى النظام. وفي هذا العرض تم استخدام جميع كينونات قاعدة البيانات كما في الشكل ٩.

- **الاستعلام١٩ List Num**: لغرض حساب مجموع كمية البيع لكل قائمة للمادة نفسها، وبالسعر نفسه. كما تم استخدام **SUM** مع **GROUP BY**.
- **الاستعلام٢٠ List1**: في هذا الاستعلام يتم حساب عدد المواد في كل قائمة من خلال **COUNT** و **GROUP BY**.
- **الاستعلام٢١ Sales1**: في هذا الاستعلام يتم تحديد اسم الزبون في القائمة، اسماء المواد، الكمية (مع تجنب تكرار المواد في القائمة من خلال حساب مجموع الكميات للمادة نفسها في القائمة نفسها)، مع باقي تفاصيل القائمة.
- **الاستعلام٢٢ Sales2**: يتم من خلال هذا الاستعلام حساب كلفة الكمية المباعة

بالدينار وبالدولار باستخدام دالة totalcost، وهي دالة لغرض حساب كلفة مادة في قائمة بيع.

- **الاستعلام٢٣ Sales3**: لحساب الربح من خلال طرح الكلفة من مبلغ البيع، ويتم حساب اجمالي الكلفة بالدينار او بالدولار اعتماداً على قائمة البيع مع احتساب سعر الدولار مقابل الدينار بتاريخ القائمة. هذا الاستعلام هو الاستعلام النهائي في هذا العرض.

خ. **كشف مشتريات مادة من مبيعات SubSales**: يتم من خلال هذا العرض تحديد مصدر مبيعات مادة في قائمة يتم اختيارها من العرض السابق. في هذا العرض تم استخدام (**SQL**)، فضلاً عن البرمجة الاجرائية. اذ يتم بناء الاستعلام، ومن ثم بالاعتماد على هذا الاستعلام يتم بناء جدول يحتوي على الكشف. بالتأكيد إن استخدام (**SQL**) اكثر كفاءة، ولكن لايمكن الحصول على كل شيء من خلال (**SQL**)، من هنا نضطر للرجوع الى البرمجة التقليدية. وفي هذا العرض تم استخدام الكينونات Customer, List, Pay_list كما في الشكل ١٤.




**الشكل ١٤ ERD العرض SubSales**

**الاستعلام ٢٤ Pays**: من خلال هذا الاستعلام يتم جلب المشتريات للمادة المطلوبة لغاية تاريخ البيع للقائمة المطلوب جردها. وتم الحصول على مجموع الحقل number لكل قائمة وبالسعر نفسه في حالة تكرار المادة في القائمة.

- **الجدول ١ Stk**: هذا الجدول مشابه في الهيكل للاستعلام السابق، وتم تكوينه من خلال الايعاز **COPY STUCTURE**. اولاً يجب حساب المتبقي من المادة لغاية تاريخ البيع باستعمال الدالة remain. بعد ذلك نقوم باستبعاد قوائم الشراء في ابتداء من النهاية لحين استنفاد المتبقي، وما يسبقه هو مصدر الكمية المباعة لحين استنفاد هذه الكمية. وتم استخدام الايعاز **APPEND BLANK** لاضافة قيد الى الجدول Stk. كما تم نقل القيد من الاستعلام ٢٤ من خلال الايعاز **GATHER MEMVAR**، وادخال القيد الى الجدول Stk باستخدام الايعاز **SCATTER**

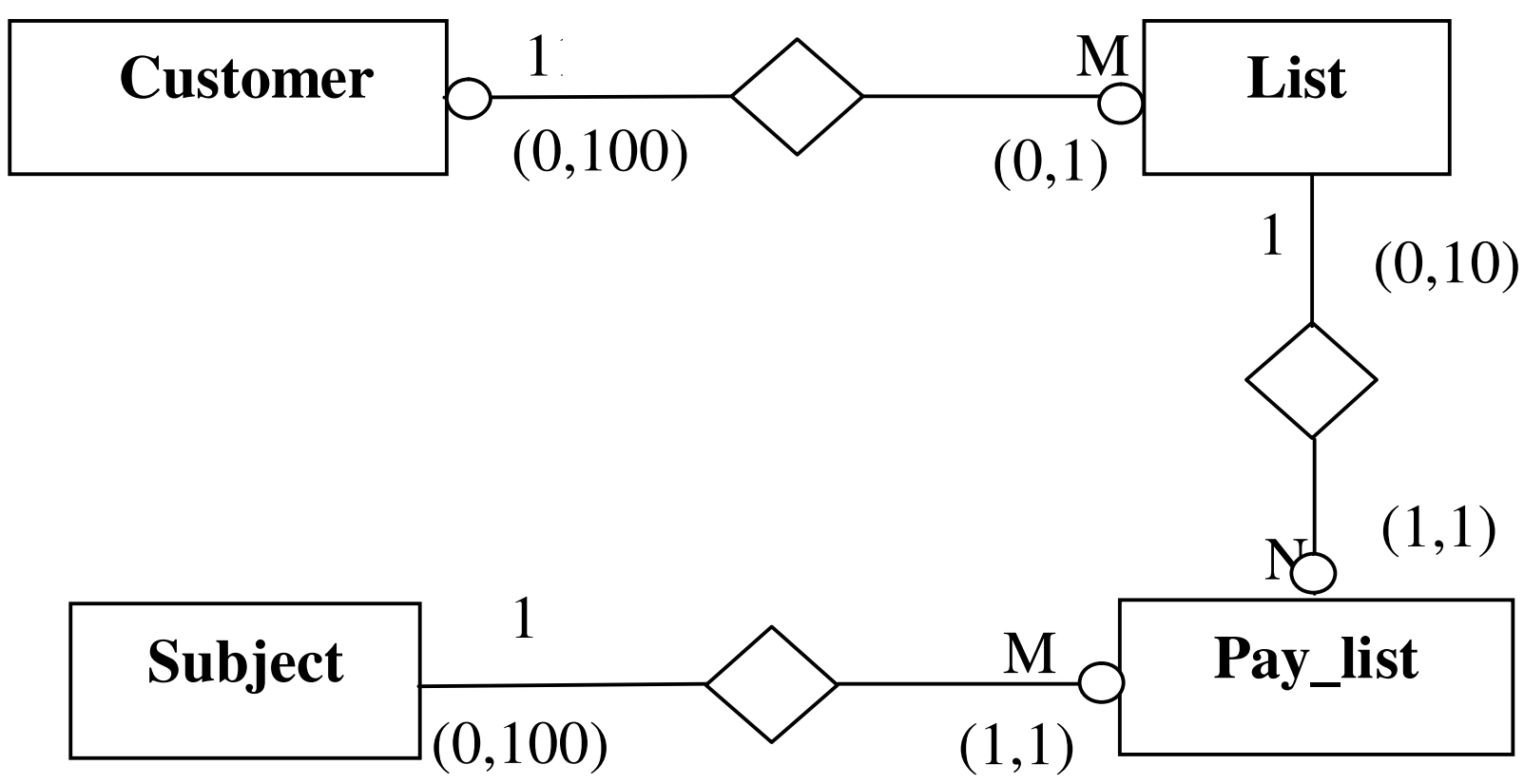
**MEMVAR**، كما في الشكل ١٥. المتغير m.listnumber هو الكمية المتبقية من المادة المباعة.

```
DO WHILE !BOF() AND m.listnumber>m.number
      m.listnumber=m.listnumber-m.number
      SELECT stk
      APPEND BLANK
      GATHER MEMVAR
      SELECT pays
      SKIP -1
      SCATTER MEMVAR
ENDDO
```

**الشكل ١٥ المقطع البرمجى الخاص بنقل البيانات الى Stk**

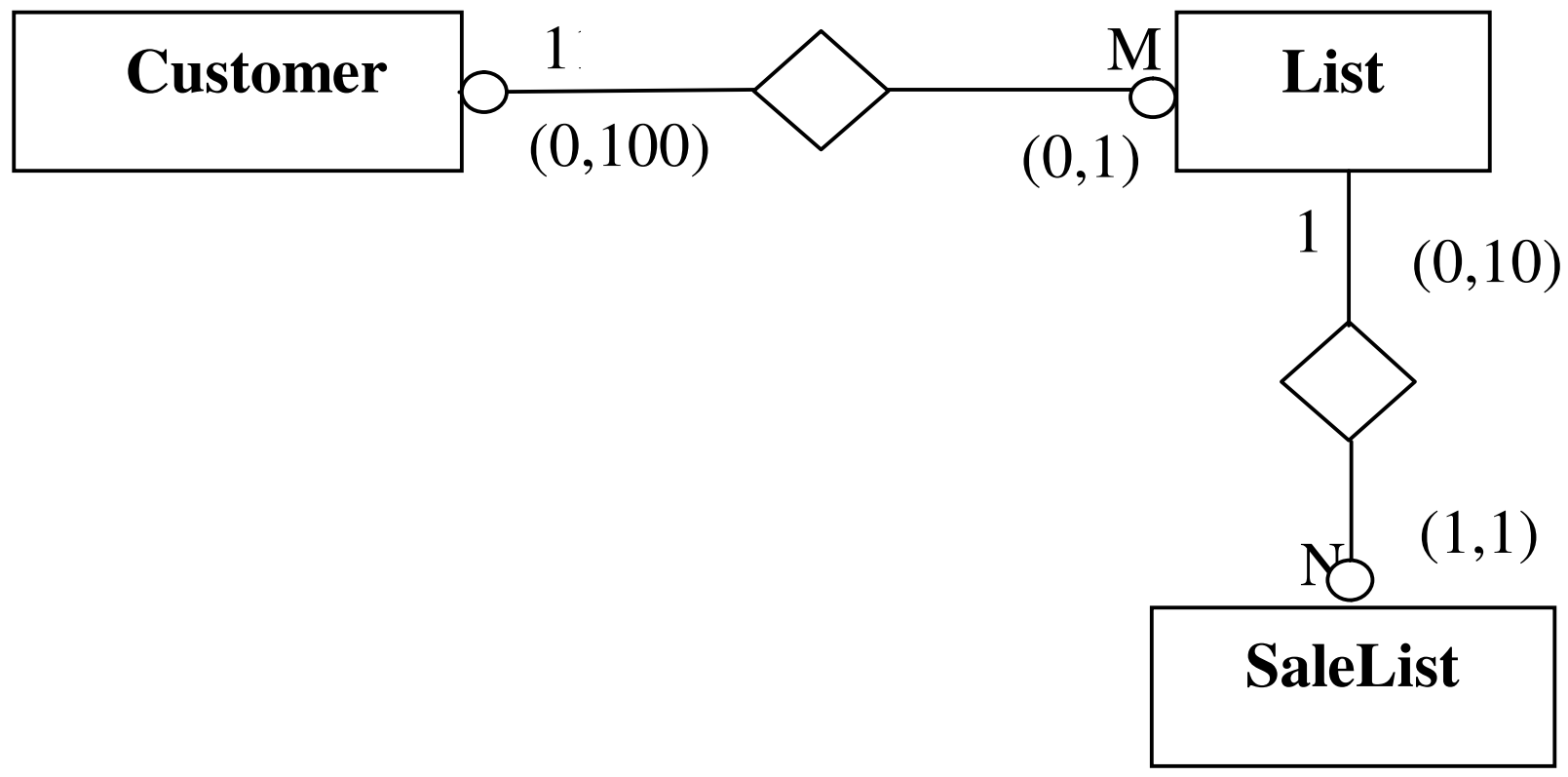
هـ . **المشتريات Pays**: تم بناء هذا العرض لغرض تدقيق قوائم الشراء. يضم هذا العرض المشتريات بكامل التفاصيل بضمنها اسم الزبون واسم المادة مرتبة حسب التاريخ. وفي هذا العرض تم استخدام الكينونات Customer, List, Pay_list, Subject كما في الشكل ١٦.

- **الاستعلام٢٥ Pays**: هذا الاستعلام يتضمن المشتريات بكامل التفاصيل، وهو الاستعلام الرئيس في هذا العرض. وتم استخدام **SUM** للحقل number للتخلص من المواد المتكررة في القائمة الواحدة، فضلاً عن **GROUP BY**.




**الشكل ١٦ مخطط ERD للعرض Pays**

د. **كشف مبيعات مادة من مشتريات Subject**: هذا العرض مشابه للعرض في الفقرة و. يتضمن استعلام وجدول يتم بناءه بالاعتماد على الاستعلام. ولتنفيذ هذا العرض تحتاج الى الكينونات Customer, List, Salelist كما موضح في الشكل ١٧.




**الشكل ١٧ مخطط ERD للعرض Subject**

- **الاستعلام٢٦ Sales**: ضم هذا الاستعلام المبيعات للمادة المطلوبة لغاية تاريخ قائمة الشراء المختارة من العرض السابق. كذلك تم استخراج مجموع الحقل number لكل قائمة وبالسعر نفسه في حالة تكرار المادة في القائمة.
- **الجدول٢ Stk**: هذا الجدول مشابه للجدول١ غير أنه يعتمد على الاستعلام٢٦.

**٤. تصميم النظام التفصيلي Detail System Design**

في هذه المرحلة ننتقل الى المستوى الفيزيائي Physical Level، يضم هذا المستوى تصميم الشاشات (screens) المستخدمة في إدخال البيانات وعرض المعلومات، تقارير الطابعة في كلتا الحالتين، والشاشة والقائمة (menu) الرئيسة في النظام. فضلاً عن تحديد جميع ملفات النظام. الجدول ٢، ٣ يحتوي على جميع شاشات النظام مع الملفات الخاصة بالشاشات(٢).

## الجدول ٢
## شاشات النظام مع الملفات

| ت | اسم الشاشة | الوظيفة | الايعازات المستخدمة | الملفات |
|---|---|---|---|---|
| ١ | Intro | الوجهة الرئيسة للنظام | Screen Builder | Intro.scx<br>Intro.sct<br>Intro.spr |
| ٢ | Login | إدخال كلمة المرور للدخول إلى النظام | Screen Builder | Login.scx<br>Login.sct<br>Login.spr |
| ٣ | List | إدخال البيانات الى الكينونات List، SaleList، Pay_List، و customer | Screen Builder<br>Wizard | List.scx<br>List.sct<br>List.prg |
| ٤ | Subjecte | اختيار مادة ضمن قائمة شراء او بيع كذلك يمكن اضافة مادة جديدة الى الكينونة Subject | Screen Builder | Subjecte.scx<br>Subjecte.sct<br>Subjecte.spr |
| ٥ | Fhras | قائمة بالمواد الموجودة في النظام مع سعر البيع لكل مادة | Modify Command<br>Define Window<br>Browse | Fhras.prg |
| ٦ | Custfhrs | قائمة الزبائن في النظام | Modify Command<br>Define Window<br>Browse | Custfhrs.prg |
| ٧ | Viwstock | جرد المخزن (الرصيد، المبيعات، والربح) لكل مادة | Modify Command<br>Define Window<br>Browse | Viwstock.prg |
| ٨ | Substock | كشف مادة (الشراء، والبيع) | Modify Command<br>Define Window<br>Browse | Substock.prg |
| ٩ | Payment | قائمة ديون الزبائن (الرصيد، الدائن/له، والمدين/عليه) لكل زبون | Modify Command<br>Define Window<br>Browse | Payment.prg |
| ١٠ | Customer | كشف زبون (الشراء، البيع، التسديد، السحب، والتحويل) | Modify Command<br>Define Window<br>Browse | Customer.prg |
| ١١ | Custdtl | كشف زبون فضلاً عن المادة، الكمية، والسعر | Modify Command<br>Define Window<br>Browse | Custdtl.prg |
| ١٢ | Mony | كشف الحركة النقدية (شراء نقدي، بيع نقدي، تسديد، وسحب) | Modify Command<br>Define Window<br>Browse | Mony.prg |
| ١٣ | Sales | المبيعات | Modify Command<br>Define Window<br>Browse | Sales.prg |
| ١٤ | Subsales | كشف مشتريات مادة ضمن المبيعات | Modify Command<br>Define Window<br>Browse | Subsales.prg |
| ١٥ | Pays | المشتريات | Modify Command<br>Define Window<br>Browse | Pays.prg |

**الجدول ٣**
**تابع شاشات النظام مع الملفات**

| ت | اسم الشاشة | الوظيفة | الايعازات المستخدمة | الملفات |
|---|---|---|---|---|
| ١٦ | Subject | كشف مبيعات مادة ضمن المشتريات | Modify Command<br>Define Window<br>Browse | Subejct.prg |
| ١٧ | Exchange | التحويلات | Modify Command<br>Define Window<br>Browse | Exhange.prg |
| ١٨ | Backup | تكوين نسخة احتياطية من البيانات | Screen Builder | Bakcup.scx<br>Backup.sct<br>Backup.spr |
| ١٩ | Restore | استرجاع البيانات من النسخة الاحتياطية | Screen Builder | Restore.scx<br>Restore.sct<br>Restore.spr |
| ٢٠ | Password | تغيير كلمة المرور للنظام | Screen Builder | Password.scx<br>Password.sct<br>Password.spr |
| ٢١ | Indexing | إعادة فهرسة البيانات | Delete File *.cdx<br>Index on | Indexing.prg |

Screen Builder برنامج لتكوين شاشة باستعمال الأسلوب المرئي وهو مرحلة أولية من البرمجة المرئية.
Modify Command لكتابة برنامج.
Define Window لتعريف نافذة.
Browse لعرض محتويات جدول او استعلام.

الجدول ٤ يحتوي على جميع التقارير المستخدمة في النظام.

القائمة الرئيسة باسم Salebuy تم بناؤها باستخدام البرنامج Menu Builder. ومن خلال هذه القائمة يتم تنفيذ جميع الشاشات في النظام. وتحتوي القائمة على شفرة الإعداد (setup code)، اذ في هذه الشفرة يتم تنفيذ الشاشة رقم ١ والشاشة رقم٢.

أيضاً تم بناء البرنامج Open لفتح الجداول، البرنامج Reset لتعريف المتغيرات العامة في النظام، والبرنامج Debts لحساب الديون.

بعد إكمال هذه المرحلة نكون قد حصلنا على نظام متكامل بجميع أجزائه.

### ٥. التنفيذ والتحميل Implementation and Loading

في هذه المرحلة نبدأ بالعمل على الحاسب الآلي. من خلال هذه الخطوة نقوم بتحويل التصميم النهائي من نظام مكتوب على الأوراق فقط إلى نظام يعمل في الحاسب الآلي. ويضم الخطوات الاتية:

**٥-١. كتابة الشفرة، الاختبار، والتصحيح Writing Code, Testing, and Debugging**

**الجدول ٤**
**تقارير النظام**

| ت | اسم التقرير | الوظيفة | الملفات |
|---|---|---|---|
| ١ | List | طباعة قوائم الشراء والبيع | List.frx<br>List.frt |
| ٢ | Mony | طباعة وصل القبض، ووصل التسديد | Mony.frx<br>Mony.frt |
| ٣ | Fhras | طباعة فهرس المواد | Fhras.frx<br>Fhras.frt |
| ٤ | Custfhrs | طباعة فهرس الزبائن | Custfhrs.frx<br>Custfhrs.frt |
| ٥ | Stocksud | طباعة المخزن بالدولار فقط | Stocksud.frx<br>Stocksud.frt |
| ٦ | Stocksid | طباعة المخزن بالدينار فقط | Stocksid.frx<br>Stocksid.frt |
| ٧ | Stocksui | طباعة المخزن بالدينار وبالدولار | Stocksui.frx<br>Stocksui.frt |
| ٨ | Substock | طباعة كشف المادة | Substock.frx<br>Substock.frt |
| ٩ | Payment | طباعة قائمة الديون (عامة، او وكلاء) او حسابات شخصية | Payment.frx<br>Payment.frt |
| ١٠ | Customer | طباعة كشف زبون | Customer.frx<br>Customer.frt |
| ١١ | Custdtl | طباعة كشف تفصيلي للزبون | Custdtl.frx<br>Custdtl.frt |
| ١٢ | Mony1 | طباعة الحركة النقدية في الصندوق | Mony1.frx<br>Mony1.frt |
| ١٣ | Salesid | طباعة المبيعات بالدينار فقط | Salesid.frx<br>Salesid.frt |
| ١٤ | Salesud | طباعة المبيعات بالدولار فقط | Salesud.frx<br>Salesud.frt |
| ١٥ | Salesui | طباعة المبيعات بالدينار وبالدولار | Salesui.frx<br>Salesui.frt |
| ١٦ | Subsales | طباعة كشف مبيعات مادة من المشتريات | Subsales.frx<br>Subsales.frt |
| ١٧ | Pays1 | طباعة المشتريات | Pays1.frx<br>Pays1.frt |
| ١٨ | Subject | طباعة كشف مشتريات مادة من المبيعات | Subject.frx<br>Subject.frt |
| ١٩ | Exchange | طباعة التحويلات | Exhange.frx<br>Exchange.frt |

في هذا المرحلة نقوم ببناء قاعدة البيانات، الشاشات، التقارير، القائمة الرئيسة،

وكتابة الشفرة الخاصة بكل جزء. أثناء ذلك يتم اختبار كل جزء وتصحيح أي خطأ يمكن أن يظهر.

مثال على ذلك:

عند تنفيذ الاستعلام١٣ (Listratio) واختباره لوحظ انه بسبب ربط الكينونة List مع Salelist في الجزء الاول وPay_list في الجزء الثاني، تم اهمال وصولات الاستلام والدفع وكذلك التحويلات، فحدث ان عند عرض كشف كل زبون نجد فقط قوائم الشراء والبيع.

لتصحيح هذه المشكلة تم اضافة جزء ثالث يحتوي على الكينونة List فقط، وتمت تصفية قوائم باستعمال التركيبة not exist مع استعلام فرعي على Salelist في حالة البيع، وعلى Pay_list في حالة الشراء لجلب تفاصيل القائمة. وبهذه الطريقة يتم اضافة القيود التي ليس فيها تفاصيل إلى الاستعلام عن طريق Union.

**٥ -٢.التنصيب Installation**

عند تنصيب النظام على حاسبة الشركة ظهرت مشكلة أن النظام لا يعمل على نظام التشغيل ويندوز أكس بي لأن FoxPro for Windows 2.6 (DBMS) مصمم على نظام التشغيل ويندوز بالإصدار ٣,١١، ويعمل على الإصدارات من ويندوز لغاية الإصدار ويندوز ميلينيوم Windows Me. وعليه يشترط لتنصيب النظام أن لا يتجاوز نظام التشغيل الإصدار ويندوز ميلينيوم.

**٦.التشغيل Operation**

تتضمن هذه المرحلة تشغيل النظام في الشركة وتدريب الكادر الوظيفي على كيفية استخدام النظام لكي يكون بالامكان استخدامه من قبل موظفي الشركة.

**٧.الصيانة Maintenance**

بعد البدء بالعمل على هذا النظام يجب متابعة المستخدمين لهذا النظام ومعالجة أي أخطأ قد تظهر في النظام، وهذا ما يسمى بالصيانة التي تستمر لغاية انتهاء العمل بهذا النظام والبدء بنظام جديد. تقسم الصيانة إلى ثلاثة أقسام وكما يأتي:

**٧ -١. تقييم Evaluation**

في هذه المرحلة يتم تقييم عمل النظام من قبل المستخدمين وقد تظهر نقاط ضعف في النظام وهنا يجب تجاوزها.

من نقاط الضعف التي تم اكتشافها في الشاشة List. عند ادخال قائمة بيع يجب اختيار الحساب الذي يمثل الارباح في الحقل الحساب المقابل. عند اختيار أي حساب يتم عرض رصيد هذا الحساب بوساطة الدالة balance1 للدينار والدالة balance2 للدولار. وفي حالة الربح تتأخر هذه العملية.

لتجاوز هذا الضعف تم الغاء عملية حساب الربح من الدالة balance1 والدالة balance2 في حالة الشاشة List. للقيام بهذا التحكم تم اضافة متغير بعنوان m.input يحمل هذا المتغير القيمة .T. عندما نكون في الشاشة List، والقيمة .F. عندما نكون في

الشاشة Payment، وبهذه الطريقة اصبح عمل الشاشة List افضل بالنسبة للمستخدمين.

**٧-٢. صيانة Maintenance**

اثناء عمل النظام ظهرت بعض المشاكل في العرض Customer والعرض CustDtl، وهو شاشة كشف الزبون، اذ لوحظ اختلاف بين رصيد الزبون في الشاشة Payment والشاشة Customer والشاشة Custdtl. بعد متابعة الكشف تم اكتشاف وجود قوائم بأسماء زبائن غير موجودين. بالتأكيد لايمكن ادخال قائمة برقم زبون خطأ! ولكن يمكن بعد ادخال القائمة حذف الزبون.

هذه الحالة أدت الى عدم ظهور القائمة في كشف الزبون، ولحل هذه المشكلة تم استخدام الاستعلام ٢٧ Customer1 بدلاً من الجدول Customer.

```
SELECT * FROM CUSTOMER;
UNION;
SELECT ... FROM list;
  WHERE NOT EXIST ;
    (SELECT * FROM customer ;
        WHERE customer.code=list.custoemr);
INTO CURSOR Customer1
```

**الشكل ١٨ الاستعلام١ Customer1**

- **الاستعلام ٢٧ Customer1:** يتم من خلال هذا الاستعلام الكشف عن زبائن لهم قوائم وغير موجودين في كينونة الزبائن. استخدم في هذا الاستعلام **NOT EXIST** للكشف عن الزبائن ودمجه مع باقي الزبائن باستخدام **UNION** كما في الشكل (٣,١٧). هذه طريقة بديلة عن **LEFT OUTTER JOIN** اذ إن **Foxpro for Window 2.6** لا يحتوي على هذه الخاصية.
- **الاستعلام ٢٨ Subject1:** كذلك يمكن أن تظهر المشكلة نفسها في المخزن، لذلك تم اضافة هذا الاستعلام بديلاً عن الجدول Subject ، لتحديد القوائم التي تحتوي على ارقام مواد غير موجودة ضمن كينونة المواد واكتشافها في كينونة Salelist أو pay_list.هذا الاستعلام مقارب للاستعلام ٢٧.

**٧-٣. تحسين Enhancement**

بعد عملية تقييم النظام وعملية إجراء الصيانة، تأتي عملية اضافة التحسينات أو التطوير على النظام. لغرض تحسين النظام في مجال حساب الديون، اذ لوحظ أن عملية حساب الديون تأخذ بعض الوقت. بعد تدقيق العرض Payment تم اكتشاف سبب التأخر، وهو استخدام البرمجة الاجرائية (التقليدية) في حساب الديون بدلاً من استخدام SQL بشكل كامل.

لحل هذه المشكلة تم اعادة صياغة العرض Payment بحيث عُوض عن تنفيذ الدالتين Balance1 وBalance2 ببعض الاستعلامات وكما يأتي:

- **الاستعلام ٢٩ Lastbalanc**: لغرض تصفية القوائم بحيث يعطي فقط القوائم المرتبطة بالحسابات المطلوبة اعتماداً على نوع الحساب m.important(٠ عامة ١ وكلاء ٢ حسابات شخصية). من خلال هذا الاستعلام يتم تقليل القوائم الداخلة في المعالجة.
- **الاستعلام ٣٠ Balance**: يحتوي على مجموع كل نوع من القوائم لكل حساب بالاعتماد على الحقل List_type وكما يأتي قائمة شراء، ١ قائمة بيع، ٢ وصل استلام، ٣ وصل دفع، ٤ تحويل. كما تم الربط بين المجاميع باستخدام **UNION**.
- **الاستعلام ٣١ Benfit1**: لتصفية قوائم البيع فقط التي فيها ارباح receiver!=0.
- **الاستعلام ٣٢ Benfit**: لحساب ارباح القوائم التي تم تصفيتها في الاستعلام السابق. باستخدام الدالة calbenfit. في حالة عدم وجود قوائم بيع هذا الاستعلام لا يأخذ أي وقت، لان تنفيذ الدالة Calbenfit تتطلب وقتاً ليس بالقليل.
- **الاستعلام٣٣ Balance**: لغرض دمج استعلام الارباح مع استعلام(٣) balance السابق عن طريق **UNION**.
- **الاستعلام ١ Payment**: هو الاستعلام النهائي في هذه الفقرة الذي يتضمن اسم الحساب (او الزبون) (cname)، الرصيد من الدينار (balance1)، والدولار (balance2)، مجموع ما له من الدينار (creditid) والدولار(creditud)، ومجموع ما عليه من الدينار(debitid) والدولار(debitud). في هذا الاستعلام تم حساب المجموع باستخدام SUM فضلاً عن GROUP BY.

وبهذه الطريقة تم حساب الديون بوقت قياسي، وبالاخص في حالة عدم وجود حساب ارباح، والنتيجة النهائية تم تحسين عمل النظام.

## ٨. الاستنتاجات

خرج البحث بالاستنتاجات الاتية:

١. الفرق الكبير بين البرمجة الاجرائية والبرمجة غير الاجرائية (باستخدام SQL).
٢. يتم استخدام البرمجة الاجرائية فقط في الحالات المعقدة جداً بحيث لايمكن تطبيقها على SQL.
٣. ضرورة متابعة النظام بشكل دقيق لاكتشاف الاخطأ التي من الصعب اكتشافها من دون التنفيذ التدريجي للشفرة (Trace).
٤. الجهد المبذول لصيانة النظام لا يقل اهمية عن الجهد المبذول في تحليل وتصميم وتنفيذ النظام. واهمال الصيانة قد يؤدي الى فشل النظام حتى في حالة كون التصميم جيد.

## ٨. المراجع

### أولاً - المراجع باللغة العربية

١. منيب احمد قطيشات، "قواعد البيانات"، الدار العربية، ١٩٩٩.

### ثانياً - المراجع باللغة الاجنبية

1. Alan Dennis and Barbara Haley Wixom, "SYSTEM ANALYSIS AND DESIGN", John Wiley and Sons, Inc, 2000.
2. C. J. Date, "An Introduction To DATABASE SYSTEMS", ADDISON-WESLEY, 2000.
3. Fred R. McFadden, Jeffrey A. Hoffer, and Mary B. Prescott, "Modern Database Management", ADDISON-WESLEY, 1999.
4. Peter Rob and Carlos Coronel, "DATABASE SYSTEMS Design, Implementation, And Management", Course Technology, 2000.