تطبيق نظام إدارة قواعد البيانات الكيانية العلائقية على نظام الاندثارات باستخدام لغة أو راكل

آلاء فيصل سعيد
المديرية العامة لتوزيع كهرباء الشمال
وزارة الكهرباء

أسماء ياسين حمو
كلية علوم الحاسبات والرياضيات
جامعة الموصل

تاريخ استلام البحث:٢٠٠٣/١٢/٢٢ **تاريخ قبول البحث:٢٠٠٤/٨/٢٢**

ABSTRACT

This research implements concept of Object Relational DataBases Management System ORDBMS on depreciation system because it's considered as a complex model. ORDBMS abstracts the level of writing system by adding data types defined by user (user define type UDT) which represents business objects, these objects are stored in database as a column on table or object table which can be accessed by using methods defined by user (user define function UDF). Object reference REFs is used instead of multiple relational between tables. Multiple data are implemented using collection data type (Varray, Nested table) without the need for extra tables. The application is programmed using oracle8 software because it has the propriety of an object relational database.

الملخص

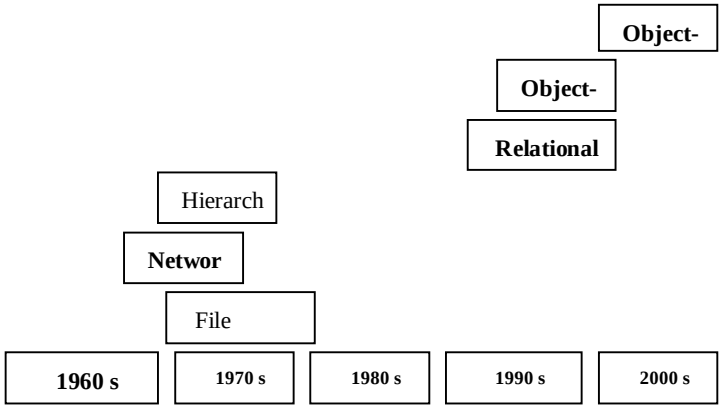
يعرض البحث كيفية تطبيق مفاهيم نظام إدارة قواعد البيانات الكيانية العلائقية Object Relational Data Base Management System (ORDBMS) على نظام الاندثارات بوصفه نموذجاً معقداً عن (ORDBMS) إذ يقوم برفع مستوى التجريد الذي يكتب فيه النظام من خلال إضافة أنواع بيانات عرفها المستخدم User Define Type(UDT) وهي تمثل كيانات إدارة الأعمال، وهذه الكيانات تخزن في قاعدة البيانات كعمود في الجدول أو كجدول كيانات، ويتم الوصول إليها من خلال استخدام الطرائق الخاصة بالكيانات التي عرفها المستخدم User Define Function (UDF). وكان لاستخدام مراجع الكيان References (REFs) الفائدة في إلغاء عمل الروابط المتعددة بين الجداول ، كما مكن استخدام أنواع بيانات المجاميع

(Nested table,Varray) من تضمين أنواع البيانات المتعددة دون الحاجة إلى فصلها في جداول جديدة.

١ - تاريخ نظم قواعد البيانات

ظهر أول نظام خاص بإدارة قواعد البيانات في أواخر عام ١٩٦٠ وتطور ليعرف بنظام الملفاتFile Systems الذي يقوم بخزن كميات كبيرة من البيانات ولفترات طويلة من الزمن ولكنه لا يوفر وسائل ذات كفاية للوصول إلى البيانات المخزونة في الملف ، كما انه لا يسمح بالوصول المتزامن لأكثر من مستخدم إلى الملف نفسه [١٥] . أعقب ذلك تطور نظام الملفات إلى النموذج الشبكي Network Model ويمثل هذا النموذج على شكل مجاميع من السجلات ، والعلاقات بينها تتم باستخدام الروابط التي تظهر على شكل مؤشرات . ثم ظهر النموذج الهرمي Hierarchical Model على غرار النموذج الشبكي ويختلف عنه في أن تمثيل القيود يتم بوصف مجاميع من البناء الشجري فضلاً عن التمثيل الرسومي [٣] [١٤] . يعّد النموذج الشبكي والنموذج الهيكلي من النماذج الأولية لتمثيل نظم قواعد البيانات التي استخدمت في أواخر الستينات وبداية السبعينات و حلت محلها النظم المعتمدة على النماذج العلائقية Relational Model بوصفها منافساً كبيراً [١٥] ويقوم هذا النموذج بتمثيل البيانات والعلاقات بينها من خلال الجداول التي تحتوي على عدد من الأعمدة التي لها أسماء أحادية مع ربط البيانات الموجودة في الجداول باستخدام العلاقات [١٤] [١٥] .وفي أواخر الثمانينيات وبداية التسعينيات ظهر نموذج الكيانات الموجهةObject Oriented Model لأسباب كثيرة منها فقدان الهيكلية في الجداول العلائقية وعدم دعمها للهيكلية المتداخلة مثل المجاميع فضلاً عن محدودية أنواع العلاقات اللازمة لتمثيل هذه الهيكلية مع زيادة المتطلبات للقيام بعملية البحث والفهرسة ، وعمل أمثلية لجمل الاستفسار على البيانات المعقدة . فضلاً عن الانتشار الواسع لاستخدام تقنية الكيانات الموجهة Object Oriented(OO) في عمليات التحليل والتصميم وتطوير التطبيقات. إن استخدام البرمجة الكيانية الموجهةObject Oriented programming(OOP) في قواعد البيانات وفرت خصائص الكيانات والأصناف والوراثة والكبسلة والتجريد وتعدد الأشكال وإعادة الاستخدام [٧][١٧] . ولكن ضعف نموذج قواعد البيانات الكيانية الموجهة من ناحية الأداء والتعقيد في جمل الاستفسار وعدم توفيرها للامثلية فضلاً عن عدم قابلية النموذج على تمثيل قواعد البيانات الكبيرة جداً Very Large Data Base VLDB مع عدم توفير سرية للبيانات المخزونة أدى إلى ظهور نموذج الكيانات العلائقية Object Relational Model الذي يقوم

بتوسيع النموذج العلائقي من خلال ضم صفات نموذج قواعد البيانات الكيانية الموجهة إليه [٩]. والشكل (١) يوضح التطور الزمني لنظم إدارة قواعد البيانات [١١].



الشكل (١) التطور الزمني لنظم إدارة قواعد البيانات

2- نوع الكيان **Object Type**

يعتبر نظام إدارة قواعد البيانات الكيانية العلائقية (ORDBMS) Object Relational Data Base Management System امتداداً لنظام إدارة قواعد البيانات العلائقية Relational Data Base Management System (RDBMS) مضافاً إليه تقنية الكيانات الموجهة Oriented Object (OO) من خلال السماح للمستخدمين بتعريف نوع الكيان Object_type وهو نوع مركب يتم تكوينه بناءً على تعريف المستخدم ولا يكون معرفاً مسبقاً ويقوم بتوفير الإجراءات والبيانات المجردة [٨] . يتم تعريف أنواع الكيانات وخزنها في الهيكل العام لقواعد البيانات التي يشترك فيها اكثر من برنامج واكثر من مستخدم واحد [١٣] .

إن الغرض من استخدام أنواع الكيانات هو تقليل التعقيد من خلال تقسيم النظام الكبير إلى أجزاء منطقية مما يسمح بتكوين أجزاء برمجية نموذجية تسمح بإعادة الاستخدام وتساعد المبرمج ليقوم بتطوير البرمجيات بصورة متزامنة .كما أن استخدام عمليات الكبسلة Encapsulation على البيانات وذلك بخزن الدوال والإجراءات مع البيانات تمكن من معالجة البيانات في التطبيقات على شكل كيانات ، وتقلل من التأثيرات الخارجية في البيانات من خلال

الوصول إليها باستخدام الدوال والإجراءات المرتبطة بها [٨] وبذلك يمكن استخدام أنواع الكيانات لضم نموذج الكيانات مباشرةً مع الهيكل العام لقاعدة البيانات بدلاً من تبسيط النموذج إلى جداول علائقية وأعمدة [١٣] .

أن المتغيرات التي تقوم بصياغة هيكلية البيانات تسمى الخصائص Attribute وتعرف في الجزء التخصيصي Specific part والذي يكون الزامياً والإجراءات والدوال التي تميز السلوك الخاص لنوع الكيان تسمى بالطرائق Methods وتعرف في جزء الجسم Body part الذي يكون اختيارياً .

إن النمذجة البيانية تعتمد على التحليل النصي للمسألة وفيما يأتي وصف نصي لنظام حساب الاندثارات الذي سيتم عمل نموذج له .

الاندثار (الاستهلاك) هو النقص التدريجي في قيمة الموجود الثابت لأسباب كثيرة منها الاستخدام الطبيعي والتقادم والتطورات التكنولوجيا.... الخ.فالكيان الرئيسي لنظام الاندثارات هو الموجودات الثابتة وهي ممتلكات منقولة ملموسة أو غير ملموسة منتجة أو مقتناة بمعرفة الوحدة لغير أغراض البيع أو التحوير بل لاستمرار استعمالها طول مدة وجودها بوصفها أدوات إنتاج أو مساعدة في الإنتاج . ولتعريف هذه الموجودات في النظام يكون من خلال صنف حركة الموجودات الذي يتضمن المعلومات كافة عن الموجود مثل رمز الحركة وتاريخها والقيمة الدفترية .. الخ . ويتم اقتناء هذه الموجودات بناءً على طلبات الشراء أو طلبات تحويل يوجهها الموظفون العاملون في أقسام الشركة المختلفة وهذا يتطلب أن يرتبط صنف حركة المواد مع صنف خاص بطلب الشراء والذي يضم رقم الطلب وتاريخه وقائمة المواد المطلوبة وأسعارها والمبلغ الإجمالي إضافة إلى ارتباط هذا الصنف مع صنف الموظفين لتحديد معلومات الموظف مثل الاسم والعنوان الوظيفي والقسم الذي يعمل فيه وتختلف الموجودات الثابتة من حيث طبيعتها واستخدامها وطريقة احتساب الاندثار السنوي عليها ، لذلك يتطلب أن يرتبط صنف حركة الموجودات مع صنف الموجودات الأساسية الذي يضم قيمة الاندثار لكل مادة مع جميع المعلومات عنها مثل الاسم والنوع والرمز وهو يضم الموجودات الثابتة الملموسة (المادية) وهي التي يتم استخدامها في الوحدة الاقتصادية لغرض الإنتاج أو في الوحدات الاقتصادية غير الإنتاجية ومنها المباني والآلات والمعدات والعدد والأدوات والسيارات الإنتاجية والخدمية والأثاث وأجهزة المكتب .ويحتسب عليها الاندثار بنسب مختلفة حددها النظام المحاسبي مما يتطلب ربط صنف الموجودات الأساسية مع صنف خاص بالنظام المحاسبي الموحد للحصول على أرقام

الحسابات ونسب الاندثار لكل مادة وتصنيف المواد حسب العمر التقديري المتوقع لكل من هذه الموجودات .

إن حساب الاندثارات هي طريقة محاسبية تهدف إلي توزيع تكلفة الموجودات الرأسمالية الملموسة أو قيمها الأساسية ناقصاً قيمة الخردة (في حال وجودها) على الحياة الإنتاجية المقدرة للوحدة بطريقة منتظمة ومعقولة. وهذا يتطلب وجود صنف خاص لحساب الاندثار يقوم بعمل الحساب بناءً على مبدأ التكلفة التاريخية أي توزيع التكلفة التاريخية للأصل الثابت على عدد سنوات العمر الإنتاجي المقدرة للموجود دون النظر إلى التغيرات التي تطرأ على الأسعار واحتساب الاندثار على هذه القيمة وعرضها بكشوفات ملحقة مع الحسابات الختامية لكي تتم مقارنتها والتعرف على التغييرات المستمرة في قيمة الموجودات .فضلاً عن احتساب عمر الآلة منها عدد ساعات التشغيل الفعلية التي يحتسب على أساسها معدل الاندثار لهذه الآلة . وهذا يتطلب وجود صنف خاص لساعات التشغيل الفعلية التي يحتسب على أساس معدل الاندثار لهذه الآلة وهذا يتطلب وجود صنف خاص لساعات الاشتغال وفي حالة انقضاء العمر الإنتاجي للآلة من خلال أتلافها أو فقدانها من الشركة لأي سبب مثل التحويل إلى جهات أخرى فان هذا يتطلب وجود صنف خاص بحركة الشطب والتحويل للموجود وتتم هذه الحركة بإشراف لجان خاصة تقوم بالعملية وهذا يتطلب الارتباط مع صنف اللجان وما يتضمنه من معلومات عن تاريخ تشكيل اللجنة وأعضائها فضلاً عن تعريف صنف الجهات المحول أليها أو الجهات التي تم الاستلام منها. ولاستلام الموجود من قبل الموظفين كالأدوات المكتبية المختلفة فهذا يتطلب أن يرتبط صنف حركة المادة مع صنف خاص بالذمم لمعرفة الموظف المستخدم لهذا الموجود والقسم العامل فيه وهذا يتطلب عمل ربط بين هذه الأصناف باستخدام علاقات ذات تعدديات مختلفة.

والشكل (٢) يوضح المخطط العام لنظام حساب الاندثارات والمكتوب بلغة النماذج الموحدة Unified Modeling Language (UML) وتم فيه تعريف أنواع البيانات الخاصة بنظام الاندثارات وهي (نوع كيان الصنف class_t و نوع الكيان الخاص بالاندثارات extint_t ونوع الكيان الخاص باسماء الموظفين name_t ونوع الكيان الخاص بالكتب الرسمية book_t ونوع الكيان الخاص بالمستندات doc_t ونوع الكيان الخاص بساعات الاشتغال work_t ونوع الكيان الخاص بالمادة item_t) [١] . فمثلاً نوع الكيان الخاص بالمادة item-t يتكون من خصائص المادة هي (الكمية qty و سعرالوحدة unit_price و تاريخ الاستخدام use_date) ، وسلوك المادة هي (الكمية valu و عمر المادة item_age) ، وبذلك تم عمل كبسلة للخصائص والطرائق في نوع الكيان [4] [8] [16]. والشكل(٣) يوضح المقطع البرمجي لتكوين نوع

الكيان الخاص بالمادة مع طرائقه وتمت الاستفادة من خاصية زيادة التحميل Over loading على دالة valu من خلال استخدام نفس اسم الدالة مرة بدون معلمات (Parameters)كـ(**value**) وأخرى باستخدام **valu**(qty IN number , unit_pric IN number) .

مقطع برمجي لتكوين نوع الكيان الخاص بالمادة مع طرائقه

```
CREATE OR REPLACE TYPE  "EXTIN". Item_t  AS  OBJECT
(  qty          NUMBER(12,3) ,
   unit_pric    NUMBER(12,3) ,
   use_date     DATE         ,

MEMBER  PROCEDURE  item_age( use_date  IN  date , age OUT number ) ,
MEMBER  FUNCTION   valu(qty  IN number , unit_pric IN number) RETURN NUMBER ,
MEMBER  FUNCTION   valu  RETURN  NUMBER

PRAGMA  RESTRICT_REFERENCES( item_age ,  RNDS  ,  WNDS )
PRAGMA  RESTRICT_REFERENCES( valu     ,  RNDS  ,  WNDS )


CREATE  OR  REPLACE  TYPE BODY  item_t   AS

MEMBER  PROCEDURE  item_age ( use_date IN date , age OUT  number)
IS
BEGIN
AGE := ( TRUNC( months_between ( sysdate,use_date ) / 12 ) ) ;
END item_age;


MEMBER  FUNCTION  valu( qty  IN  number , unit_pric IN  number ) RETURN NUMBER
IS
BEGIN
RETURN ( qty  *  unit_pric  ) ;
END valu;


MEMBER FUNCTION  valu  RETURN  number
IS
BEGIN
RETURN( qty  *  unit_pric ) ;
END valu;
END;
```

Attribute

Method SPEC

SPEC

Method BODY

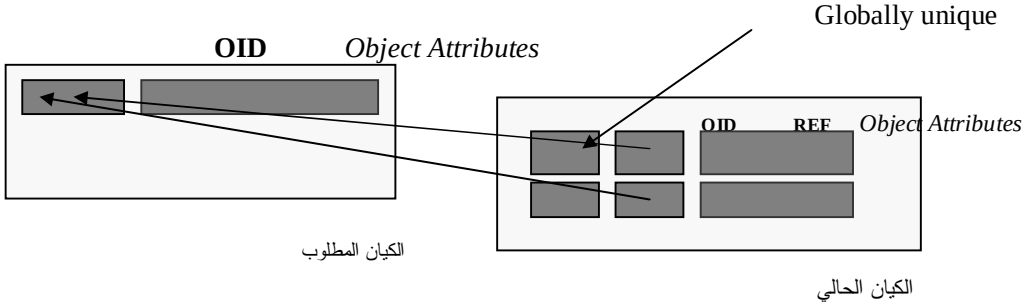
BODY

الشكل (3) نوع الكيان وطرائقه

٣ - المشاركة بين الكيانات Sharing Objects

تمتلك الكيانات معرفاً أحادياً يسمى بمعرف الكيان Object Identifier (ويختصر بـ OID) الذي يسيطر على المرجعية بين الكيانات ويتم عمل مشاركة بين الكيانات باستخدام الروابط المرجعية References (وتختصر REFs) التي تعّد من أنواع البيانات المبنية داخل قاعدة البيانات للغة المستخدمة (أو راكل) [٨].

ومن فوائد استخدام REF هو تجنب الفهرسة والروابط المكلفة وعدم تكرار البيانات غير الضرورية وعند القيام بتحديث الكيانات المشتركة فان التغيير يحدث في مكان واحد فقط ويقوم المرجعية REF باسترجاع القيم المحدثة انياً [٨] . وإذ يقوم REF بتوفير تعريف أحادي لسطر الكيان المخزون في جدول الكيان أو عرض الكيان من خلال قيمته التي تتكون من تعريف أحادي (OID) مرتبط بجدول الكيانات ومعرف لسطر الكيان (ROWID) المخزون في الجدول وهذا يوفر وصولاً سريعاً إلى الكيان المطلوب ويسهل عملية الربط بين الكيانات [١٢] .وتعطي نموذجاً لمجموعة الروابط تجاه الكيانات وبخاصة علاقات العديد-إلى-الواحد Many-To-One [٦] ، والشكل (4) يوضح عملية المشاركة بين الكيانات .



الشكل (4) المشاركة بين الكيانات

وفي نظام الاندثارات تم عمل مراجع بين نوع الكيان الخاص بالذمم Borou مع نوع الكيان الخاص بالموظفين employee_t باستخدام الخطوات البرمجية الموضحة في الشكل (٥) من خلال:-

١- تعريف متغير من نوع REF اسمه Brou_ref ليشير إلى نوع الكيان المطلوب employee_t.

٢- كتابة جملة Select لإرجاع قيمة OID الخاصة بالكيان المطلوب employee_t من عرض الكيان الخاص بالموظفين employee_t إلى المتغير المعرف Brou_ref.

مقطع برمجي للحصول على المراجع في جدول الذمم
Declare Brou_ref REF employee_t; Begin SELECT REF(E) INTO brou_ref FROM employee_t E WHERE employee_cod=123452; INSERT INTO borou_view_t VALUES(item_ref , brou_ref ,10 , 0 , NULL , dept_ref);

الشكل(٥) الحصول على المراجع

أما عملية استرجاع البيانات المشار إليها باستخدام REF فتتم من خلال ما يسمى بإعادة المرجعية DEREFRENCING REF (وتختصر بـ DEREF) [١٢] . تم استخدام عملية DEREF مع جملة Select لتقوم بإرجاع قيمة الكيان من الجدول المزدوج DUAL إلى متغير معرف مسبقاً ، [٨][١٦].والخطوات البرمجية لإعادة المرجعية في جدول الذمم موضحة في الشكل (٦) إذ تم :-

التعليق [1a]: يمثل جدول صغير يمتلكه معرف المستخدم SYSTEM يمكن لجميع مستخدمي اوراكل من الوصول إليه ويتكون DUAL من صف واحد وعمود واحد [٣].

١- تعريف متغير من نوع الكيان V_borou ليشير الى نوع الكيان المطلوب employee_t.

٢- كتابة جملة Select لاسترجاع البيانات من الجدول المزدوج DUAL إلى المتغير المعرف V_borou باستخدام عملية DEREF .

مقطع برمجي لإعادة استخدام المراجع مع جدول الذمم

DECLARE
V_borou employee_t;
 	
BEGIN
IF INSERTING THEN
SELECT DEREF (:NEW.brou_cod) INTO V_borou FROM DUAL ;
INSERT INTO borou (cod , conv_dat , conv_cod , brou_cod , qty , fback , note ,dept_cod)
VALUES (V_item.cod , :NEW.conv_dat , V_conv.employee_cod , V_borou.employee_cod :NEW.qty , :NEW.fback , :NEW.note , V_dept.dept_cod);

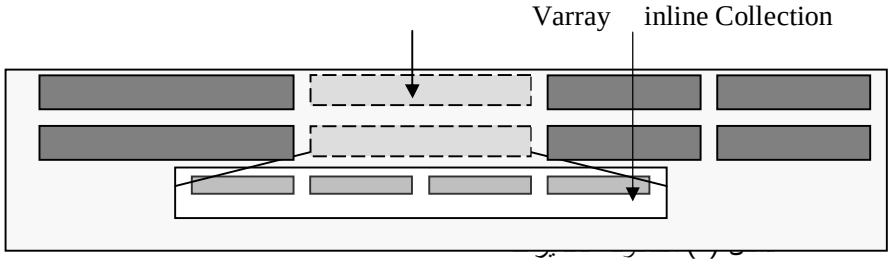
الشكل (٦) إعادة المرجعية

٤ - أنواع المجاميع Collection Types

هي مجموعة من الكيانات التي تتجمع في متغير واحد من نوع بيانات المجاميع (الجدول المتداخل Nested table ، مصفوفة المتغيرات Varray) التي تساعد المبرمج على معالجة مجموعة من القيم في نفس نوع البيانات . ويتم الوصول إليها ومعالجتها باستخدام لغات البرمجة مثل SQL و PL/SQL [٨] . يتم عمل كيان من نوع المجاميع من خلال استدعاء طرائق البناء الخاصة به إذ أن اسم الطريقة يمثل اسم نوع الكيان ، ومتغيرات الطريقة هي قائمة من العناصر الجديدة والمفصولة بالفارزة [١٣] .

٤ - ١ المصفوفة المتغيرة Varying array

وهي مصفوفة ذات بعد واحد تحتوي على عناصر بيانات مرتبة من النوع القياسي أو من نوع بيانات الكيان وكل عنصر في المصفوفة له تسلسل يعتمد على موقع العنصر في المصفوفة [١٧] [٢٨] . يحدد حجم المصفوفة بعدد العناصر الموجودة فيها ولغة (Oracle) تسمح بكون المصفوفة متغيرة الحجم ، لذلك سميت بالـVarying Array (وتختصر بـــVarray) ويجب تحديد أعلى حجم للمصفوفة عند تعريف المستخدم لها [١٢] [١٣] .تعرف الـــVarray بوصفها عموداً في جدول البيانات ، وجميع البيانات الموجودة فيها يتم استرجاعها ومعالجتها في خطوة واحدة عند عملية استخراج البيانات وعلى النحو الموضح في الشكل (٧) . لذلك تعّد Varray مثالية عندما تكون هناك حدود معروفة لعدد العناصر في المجموعة [٦] .وتستخدم الـVarray لتعريف نوع بيانات العمود في الجدول أو تعريف نوع البيانات لخصائص نوع الكيان أو كمتغيرات للغة PL/SQL [٢] .



وقد تم تعريف Varray خاصة بأرقام الهواتف والدرجات الوظيفية في نظـــام حـــساب الاندثارات والخطوات البرمجية اللازمة لتعريف مصفوفة المتغيرات Varray لارقام الهواتـــف موضحة في الشكل (٨) :

تكوين نوع بيانات الكيان الخاصة بارقام الهواتف من خلال إيعاز Create Type.

إذ تم تكوين نوع بيانات المجاميع (Varray) مـــن نـــوع الكيـــان المعـــرف **phone_objtyp** وباستخدام إيعاز Create Type.

تضمين نوع بيانات المجاميع ***Phone_list_t*** كعمود في نوع الكيان agent_t.

مقطع برمجي لتكوين نوع الكيان الخاص بأرقام الهواتف

```
CREATE TYPE "EXTIN". Phone_objtyp AS OBJECT
(       phone_no        VARCHAR2(14) ) ;
```

مقطع برمجي لتكوين نوع بيانات المجاميع Varry الخاصة بأرقام الهواتف

```
CREATE OR REPLACE TYPE "EXTIN". Phone_list_t AS VARRAY(10) OF
phone_objtyp;
```

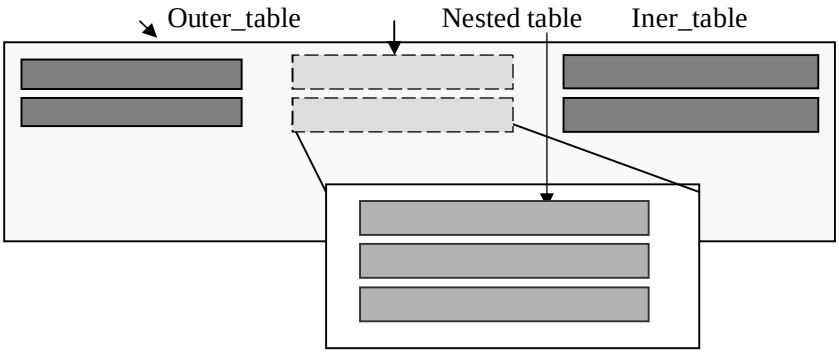
مقطع برمجي لتضمين نوع بيانات المجاميع (قائمة أرقام الهواتف) في نوع الكيان الخاص بالجهات

```
CREATE OR REPLACE TYPE agent_t AS OBJECT
( agent_cod   NUMBER(2)       ,
  agent_nam   VARCHAR2(20)    ,
  address     VARCHAR2(20)    ,
  phone_no    phone_list_t    ,
  ministry    VARCHAR2(50)  ) ;
```

الشكل (٨) تكوين مصفوفة المتغيرات

٤ - ٢ الجدول المتداخل Nested Table

هو جدول تتم إضافته إلى قاعدة البيانات بوصفه عموداً في الجدول الخارجي Outer table وعندها فان كل سطر في الجدول الخارجي يحتوي على جدول متداخل Nested table . والعناصر الخاصة بالجدول المتداخل تكون غير مرتبة وهي من أنواع البيانات القياسية أو من أنواع بيانات الكيان وتخزن هذه العناصر في جدول داخلي منفصل Inner table وبطريقة مشابهة لعلاقة الجدول الرئيس والجدول الفرعي في الأنظمة العلائقية [١٠][١٢][١٣] والشكل (٩) يوضح ذلك . ويستخدم الجدول المتداخل لتعريف نوع البيانات للأعمدة في الجدول العلائقي أو لتعريف نوع البيانات لخاصية من خصائص نوع الكيان أو كمتغيرات للغة PL/SQL . ويتم التعامل مع الجدول المتداخل بدلاً من مصفوفة المتغيرات عندما تكون البيانات غير محددة الحجم وترتيبها غير مهم أو عند عمل فهرسة أو استعلام لخاصية من خصائص الجدول المتداخل [٢] .



الشكل (٩) الجدول المتداخل

والخطوات البرمجية اللازمة لتعريف الجدول المتداخل Nested table التي تم تطبيقها في نظام الاندثارات على جدول المواد المطلوب شراؤها Items من خلال تعريفه كـ Nested table في جدول طلبات الشراء Sale_order وحسب الخطوات الموضحة في الشكل (١٠) وهي :-

١-تكوين نوع الكيان **item_typ** الخاص بالجدول الداخلي الذي يعطي هيكلية للجدول المتداخل .

مقطع برمجي لتكوين نوع الكيان الخاص بالمواد المطلوبة

```
CREATE OR REPLACE TYPE "EXTIN". item_typ AS OBJECT
(    order_id      NUMBER(6) ,
     qty           NUMBER(12,3),
     unit_pric     NUMBER(12,3),
     use_date      DATE ) ;
```

٢-تكوين نوع بيانات المجاميع ***item_table*** (الجدول الداخلي) كجدول لنوع الكيان item_typ المعرف في الخطوة ١، وهذه الخطوة مشابهة لتكوين جدول التفاصيل في النظام العلائقي غير أن العمود الخاص بالمفتاح الأجنبي غير موجود.

مقطع برمجي لتكوين جدول الكيان الخاص بالمواد المطلوبة

```
CREATE OR REPLACE TYPE "EXTIN".item_table AS TABLE OF
Item_typ;
```

٣- تكوين جدول قاعدة البيانات salenest_table(الجدول الخارجي) وهذه الخطوة مشابهة لتكوين الجدول الرئيس في النظام العلائقي ، وتم استخدام جملة Nested table لتكوين الجدول المتداخل من خلال دمج الجدول الداخلي بوصفه عموداً في الجدول الخارجي.

مقطع برمجي لتكوين جدول الكيان الخاص بالمواد المطلوبة

```
CREATE   Table   "EXTIN".salenest_table
(    order_id        NUMBER(4)   ,
     order_dat       DATE        ,
     employee_id     NUMBER(6)   ,
     ship_dat        DATE        ,
     total           NUMBER(8,2) ,
     items           item_table) NESTED TABLE items   STORE   AS
Snested_item_table ;
```

الشكل (١٠) تعريف الجدول المتداخل

في هذه الحالة يمثل items اسم العمود الخاص بالجدول الداخلي و Snested_item_table هو اسم الجدول الفيزيائي الذي يحمل الجدول المتداخل ويكون موضع خزنه في نفس مساحة الجداول Table Spaces الخاصة بالجدول الخارجي.

٤-٢-١ استخدام لغة معالجة البيانات في الجدول المتداخل

Using Data Manipulating Language in Nested Table

تمت معالجة البيانات الخاصة بالجدول المتداخل باستخدام لغة معالجة البيانات DML. والشكل (11) يوضح العمليات التي تم تنفيذها على الجدول المتداخل الخاص بطلبات الشراء والمواد المطلوبة والذي تم تكوينه في الخطوات البرمجية السابقة.

مقطع برمجي لعملية معالجة البيانات في الجدول المتداخل لطلبات الشراء

```
DECLARE
        Item_count  NUMBER;
BEGIN
                                        --عملية الإضافة إلى الجدول الخارجي--
 INSERT INTO   salenest_table    VALUES  ( 1 , sysdate , null , sysdate , 100,
                                                item_table (item_typ  ( 1 , 10 , 20 , sysdate ) ) );

 INSERT INTO    salenest_table    VALUES  ( 2 , sysdate , null , sysdate , 2000 ,s
                                                Item_table(item_typ( 2 , 100 , 20 ,sysdate ) ) );

                                        --عملية الإضافة إلى الجدول المتداخل--
INSERT  INTO  THE ( SELECT items  FROM Snested_item_table   WHERE   order_id=2)
      VALUES ( ITEM_TYP(1,543,10, sysdate));

                                        --عملية تحديث الجدول الخارجي--
UPDATE   salenest_table   SET  items = item_table(item_typ(9,30,99999,null))
                                                        WHERE order_id =1;

                                        --عملية تحديث الجدول المتداخل--
UPDATE THE ( SELECT  items  FROM  Snested_item_table  WHERE ORDER_id = 2)
                                SET use_date = sysdate   WHERE order_id = 2;

                                        --عملية الاختيار من الجدول المتداخل--
SELECT  COUNT(*)    INTO item_count
FROM  THE  ( SELECT  items  FROM Snested_item_table   WHERE  ORDER_id=1)
                                                        WHERE use_date <= sysdate;

                                        --عملية الحذف من الجدول الخارجي--
DELETE   FROM   salenest_table   WHERE   order_id=1;


                                        --عملية الحذف من الجدول المتداخل--
DELETE   THE ( SELECT  items  FROM   Snested_item_table   WHERE ORDER_id=2)
WHERE USE_DATE < sysdate;
END;
```

الشكل (١١) لغة معالجة البيانات في الجدول المتداخل

الاستنتاجات

تم الاعتماد على لغة النماذج الموحدة UMLوذلك لأنها توفر رموزاً سهلة الفهم والاستخدام في تمثيل نموذج قواعد البيانات الكيانية الذي مكن من الحصول على نموذج للتطبيق المستخدم (نظام حساب الاندثارات) فقد تم اتباع أسلوب تجزئته إلى مجموعة من المسائل الصغيرة مثل الجزء الخاص بنظام الأفراد والجزء الخاص بنظام المشتريات والجزء الخاص بنظام الحسابات العامة والجزء الخاص بالاندثارات . وباستخدام خاصية التجميع تم القيام بتجميع هذه الأجزاء بسهولة لتكوين النموذج العام للنظام .وتم التعامل مع كل جزء من النماذج السابقة بوصفه نموذجاً مستقلاً وهذا ما ساعد على إجراء التغيرات عليها دون الحاجة إلى تحوير

البرنامج بأكمله وان معالجة الأخطاء التي ظهرت في مرحلة البرمجة كانت محددة ضمن كل نموذج دون أن تؤثر في بقية النماذج بصورة كاملة .

إن استخدام نظام إدارة قواعد البيانات الكيانية العلائقية وفر إمكانيات جديدة في إدارة قواعد البيانات ومنها :-

مكن استخدام أنواع البيانات الكيانية من إغناء النظام من خلال تجهيزه بأنواع يعرفها المستخدم User Define Type (UDT) الخاصة بكيانات إدارة الأعمال التي تعامل كالأنواع القياسية المبنية في النظام مثل CHAR,DATE مع توفير حماية ضد سوء الاستخدام لهذه الأنواع من خلال منع الوصول إلى بياناتها إلا عن طريق الدوال المعرفة معها User (UDF Define Function) . وهذا من ثَمَّ أعطى إمكانية عمل الكبسلة التي توفر عَمل نموذج محكم لتطبيقات الكيانات.

إن استخدام مراجع الكيانات REFs مكن من الحصول على نموذج من مجاميع علاقات الواحد-إلى-العديد و العديد-إلى-الواحد دون الحاجة إلى عمل الروابط المتعددة بين الجداول وباعتبار REF من الأنواع المبنية داخل النظام فقد تم استرجاع البيانات من خلاله دون الحاجة إلى بناء جمل SQL بل تم ذلك من خلال عملية طلب واحدة.

مكن استخدام أنواع بيانات المجاميع (Varray,Nested table) من تضمين البيانات المتعددة مع الجدول الرئيس مباشرةً دون الحاجة إلى فصلها في جدول جديد مع روابطه لذلك تم استخدام Varray في تمثيل البيانات المحددة الحجم التي لا تحتاج إلى جملة SQL لاسترجاعها وهذا له فائدة في تسريع عملية استرجاع البيانات نفسها مقارنة بالجدول العلائقي والجداول المتداخلة .

المصادر

(١) الاء فيصل ، اسماء ياسين ، ٢٠٠٢. "**نمذجة قواعد البيانات الكيانية العلائقية باستخدام اوراكل**"، رسالة ماجستير /جامعة الموصل – كلية علوم الحاسبات والرياضيات .

(٢) دايتر كارول ، ٢٠٠٠ ." **اوراكل٨ باييل** "، دار الفاروق للنشر والتوزيع / مركز التعريب والترجمة / مصر.

[3] Beaverton O. R , ".2001 .**Object-Relational and Object-Oriented Database System** ", *http://www.software .ibm/is/sw-servers/database.*

[4] Johnny O ; .Allan R. L , ."**2000 , Experiences from** "Object-Relational Programming in Oracle8 Cot/4-06-V1.4 , *http://www.cit.dk/ COT/reports/reports/case4/.4-v1,./cot-4-06-v1.4.pdf.*

[5] Beaverton O. R , ."**2001 .Object-Relational and Object-Oriented Database System**", *http://www.software .ibm/is/sw-servers/database.*

[6] Lonsdale M, ."**1999 . Is Performance a Reason for Using Oracle8 Object ?**", *http://www.softlab.co.uk new/ uploads /ISPERFO8. pdf*

[7] Oracle-developer, "**2000 . Introduction to Object-Relational Database Development"**, *http://www.kingtraining.com/ downloads /o8diffs-paper.pdf.*

[8] Portfolio T. , 1999 . "**PL/SQL Users Guide and Reference**" http://www.technet.oracle.com

[9] Ramakanth S ;.Konda D , ".**2001 .ObjectRelational Database System-The Road ahead** *http://www.acm.org/crossroads/xrd S7-3/ordbms.html*

[10] Oracle Technical White Paper , "**2001** . **Simple Strategies for Complex Data : Oracle9i Object-Relational Technology**" *"http://www.ont . oracle. com /products/oracle9i/pdf/iot_twp.pdf. ,"*

[11] Rob P. ; Coronel C. , **2000 . Data base System Design** , *Implement and Management , 4th , course technology*.

[12] Russell J , ."1999 . **Application Developers Guide**" *Object-Relational Features , http://www.thinkspark .co.uk /ioug/PLSQL New Features in oracle8i.pdf.*

[13] Sikora Y ; .Peter J.L , .2001 .**Object to Object Communication** , *http//:www.odtug.com*.

[14] Ratio Group Ltd , 2001 . "**Persistence : Implementing Object Over a Relation Database version 1.0"** , *http://www.ratio.co.uk*.

[15] Ulman J. D ; .Widom J , ."1997 . **Afirst Course in Database System** ", *Prentic-Hall International , Inc.*

[16] Whithead A. N ."2000 . **Object-Relational Oracle8 and PL/SQL8** " ,*http://www.polito.itivrea /inofrmazioni /passaggio/ dispenseinseif/ basi/PLSQL8.pdf*"

[17] Woodger Computing Inc, 2001. "**Object Data Base**", *http://www.wci object DB capabilities.htm.*