



اسم المقال: تطبيق نموذج الكيانات على البيانات العلائقية في نظام الاندثارات باستخدام لغة اوركل
اسم الكاتب: أ.م.د. عصام محمود محمد، م. اسماء ياسين حمو، آلاء فضل سعيد
رابط ثابت: <https://political-encyclopedia.org/library/3055>
تاريخ الاسترداد: 2026/06/05 02:11 +03

الموسوعة السياسية 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 السياسية - Encyclopedia Political، يرجى التواصل على info@political-encyclopedia.org

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



تطبيق نموذج الكيانات على البيانات العلائقية في نظام الاندثارات باستخدام لغة اوركل

اسماء ياسين حمو
مدرس
كلية علوم الرياضيات والحاسبات
جامعة الموصل

الدكتور عصام محمود محمد
استاذ مساعد/ قسم انظمة حاسبات
المعهد التقني بالموصل
هيئة المعاهد الفنية

آلاء فيصل سعيد
رئيس ميرمجين/المديرية العامة
لتوزيع كهرباء الشمال
وزارة الكهرباء

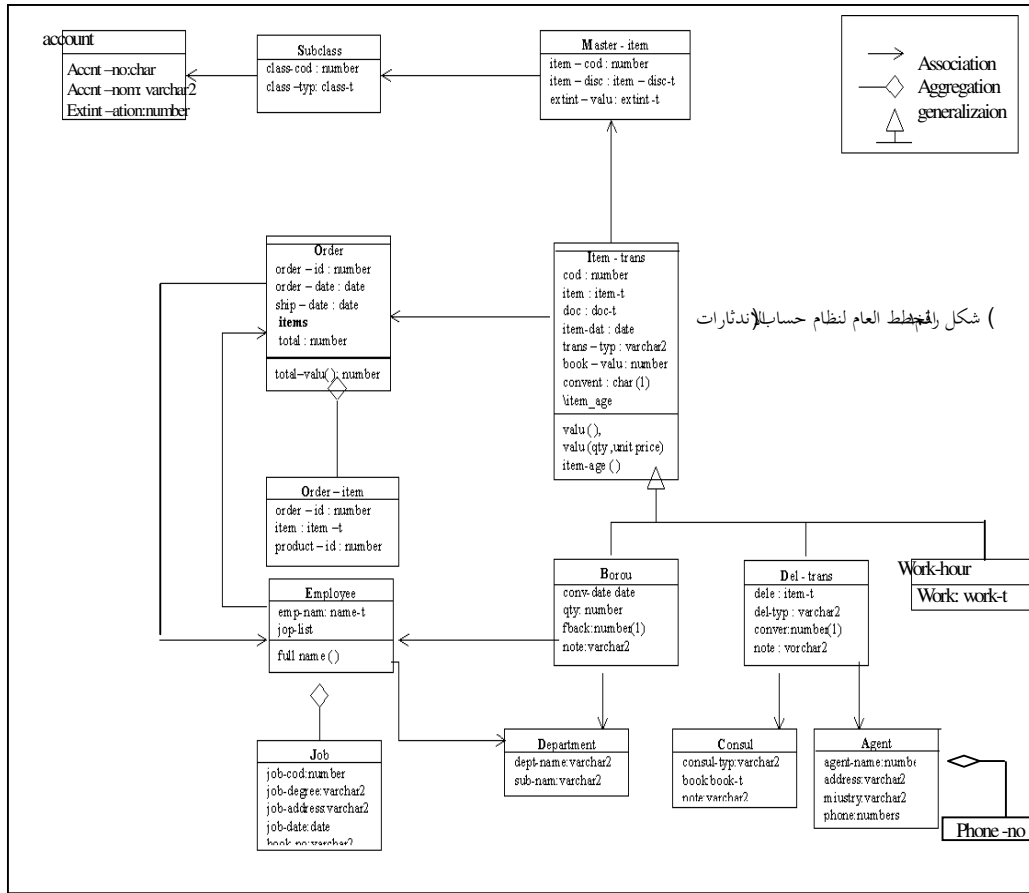
المستخلص

يعرض البحث كيفية تطبيق نظام إدارة قواعد البيانات الكيانية العلائقية ORDBMS Object Relational Data Base Management System على نظام الاندثارات بتحويله من تطبيق علائقي Relational إلى تطبيق كياني Object مع عدم إحداث تغييرات فيه أو الدخول في تفاصيله باستخدام عروض الكيان Object View التي تشتق من النظام العلائقي وترتبط معه بتكوين طبقة من البرامج الكيانية الموجهة فوق الجداول العلائقية، إذ وفرت طرق جديدة لعمل استفسار على البيانات العلائقية وتقليل جمل الاستفسار (SQL) Structure Query Language المستخدمة من خلال استخدام إيعاز MAKE- REF بدلاً من كتابة الروابط المعقدة مع الجداول المتعددة، وهذا يوفر عرض البيانات كوحدة واحدة. كما أضاف العرض الكياني إمكانية تضمين المتعددة بين المجاميع. وقد استخدمت لغة ORACL8 في برمجة التطبيق، لما توفره من دعم لمواصفات قواعد البيانات الكيانية العلائقية إضافة إلى تفرداها في توفير قادحات من نوع INSTEAD OF الخاصة بإدخال ومعالجة البيانات في الجداول العلائقية من خلال العروض الكيانية .

مقدمة

يعتبر نظام إدارة قواعد البيانات الكيانية العلائقية ORDBMS امتداداً لنظام إدارة قواعد البيانات العلائقية (RDBMS) Relational Data Base Management System مضافاً إليه تقنية الكيانات الموجهة (OO) Object Oriented من خلال السماح للمستخدمين بتعريف نوع الكيان object-type وهو نوع مركب يتم تكوينه بناءً على تعريف المستخدم ولا يكون معرف مسبقاً ويقوم بتوفير الإجراءات والبيانات المجردة (Beaverton,2001, 9) (Boss,2000, 15). يتم تعريف أنواع الكيانات وخصنها في الهيكل العام لقواعد البيانات التي يشترك فيها أكثر من برنامج وأكثر من مستخدم واحد (Giese,2001, 3). والشكل ١ يوضح المخطط العام لنظام حساب الاندثارات، والمكتوب بلغة Unified Modeling Language (UML) وفيه تم تعريف أنواع البيانات الخاصة بنظام الاندثارات وهي (نوع كيان الصنف

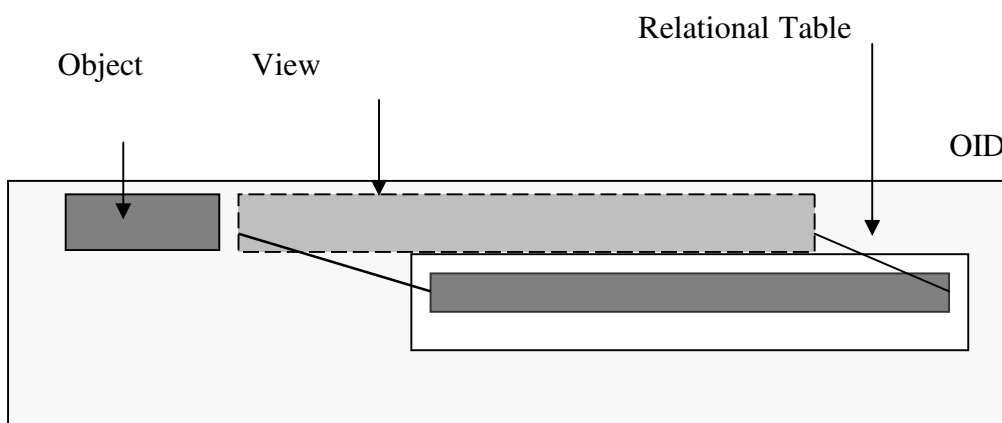
Class_t ونوع الكيان الخاص بالاندثارات Extint_t ونوع الكيان الخاص باسماء الموظفين Name_t ونوع الكيان الخاص بالكتب الرسمية Book_t ونوع الكيان الخاص بالمستندات Doc_t ونوع الكيان الخاص بساعات الاشتغال Work_t ونوع الكيان الخاص بالمادة (Item_t) (فصل، ٢٠٠٢، ٥).



١. عروض الكيان

إن نظام إدارة قواعد البيانات الكيانية العلائقية ORDBMS هو أسلوب مكمل لتكوين التطبيقات المعتمدة على طرق الوصول العلائقية فهو يسمح بمعاملة بيانات الكيانات بوصفها بيانات علائقية والبيانات العلائقية بوصفها بيانات كيانية من خلال استخدام الأدوات التي تكون طبقة من البرامج الكيانية الموجهة Object Oriented (OO) Program فوق الجداول العلائقية إذ تمكن هذه الطبقة من الوصول إلى البيانات العلائقية بالطريقة نفسها التي تخزن بها في قاعدة بيانات الكيانات الموجهة (Johnny and Allan, 2000, 13) Object Oriented Data Base (OODB)، باستخدام لغة معالجة البيانات (Data Manipulation Language (DML)) تم عمل استرجاع ومعالجة للبيانات الكيانية بالطريقة نفسها التي يتم الوصول بها إلى البيانات العلائقية

(oracle technical,2001, 4). وتعد عروض الكيان Object Views واحدة من هذه الأدوات، وهي عبارة عن جداول علائقية قياسية مع إضافة عمود مخفي يسمى بمعرف الكيان Object Identifier (OID) الشكل ٢. وهذا يعني ان تطبيقات الكيانات الموجهة OO يتم تصميمها واستخدامها للوصول إلى الجداول العلائقية بصورة شفافة، وهذا له أهمية كبيرة، إذ يسمح بإضافة تطبيقات جديدة من أنواع الكيانات الموجهة OO_type إلى التطبيقات من نوع العلاقات Relational_Type ، فضلاً عن الاستفادة من خصائص OOP مثل التجريد والكبسلة وإعادة الاستخدام . (Lonsdale,1999, 8).



الشكل ٢
عرض الكيان

٢- ١ الخطوات البرمجية لتعريف عروض الكيان

Defining Object Views

- تتمثل الإجراءات اللازمة لتعريف عروض الكيان في الآتي :
١. تكوين نوع الكيان، إذ إن كل خاصية في نوع الكيان تعرف طبقاً لعمود موجود في الجدول العلائقي.
 ٢. بناء عرض الكيان على أساس نوع الكيان المعرف في الخطوة ١ من خلال كتابة الاستفسار الذي يقوم باستخلاص البيانات من الجداول العلائقية وتحديد الأعمدة بالترتيب نفسه لتعريف خواص نوع الكيان.
 ٣. تعيين قيمة أحادية معتمدة على خصائص البيانات لتؤدي دور معرف الكيان OID الذي يسمح بتكوين المراجع (References) REFs إلى أسطر الكيانات في عرض الكيان، وعادة يتم استخدام المفتاح الأساسي الموجود في الجدول العلائقي لاشتقاق معرف الكيان من خلال جملة (OBJECT OID) WITH

(دايتير، ٢٠٠٠، ١٠) (بيتر ودروسي، ٢٠٠١، ٨) (Lonsdale,1999, 8) (Russell,1999,15). والشكل ٣ يبين تكوين الجدول العلائقي ونوع الكيان وعرض الكيان الخاص بجدول الجهات وكيفية اشتقاق المفتاح الأساسي *agent_cod* من الجدول العلائقي بوصفه معرفاً للكيان في عرض الكيان.

مقطع برمجي لتكوين الجدول العلائقي للجهات المستفيدة	مقطع برمجي لتكوين نوع الكيان للجهات
<pre>CREATE TABLE "EXTIN".agent (agent_cod NUMBER(2) NOT NULL, agent_nam VARCHAR2(20) NULL, address VARCHAR2(20) NULL, phone_no VARCHAR2(9) NULL, ministry VARCHAR2(50) NULL, CONSTRAINT agn1 PRIMARY KEY (agent_cod), CONSTRAINT agn2 UNIQUE (agent_cod));</pre>	<pre>CREATE OR REPLACE TYPE agent_t AS OBJECT (agent_cod NUMBER(2) , agent_nam VARCHAR2(20), address VARCHAR2(20), phone_no VARCHAR2(9) , ministry VARCHAR2(50));</pre>
مقطع برمجي لتكوين عرض الكيان للجهات	
<pre>CREATE OR REPLACE VIEW agent_view_t OF agent_t WITH OBJECT OID (agent_cod) AS SELECT A.agent_cod , A.agent_nam , A.address , A.phone_no , A.ministry , FROM AGENT A;</pre>	

الشكل ٣

تكوين الجدول العلائقي ونوع الكيان وعرض الكيان

٢-٢ الكيانات المتداخلة في عروض الكيان

Nesting Objects in Object Views

يسمى نوع الكيان بالكيان المتداخل عندما يتم تمثيل خصائص نوع الكيان بنوع كيان آخر معرف مسبقاً . ويتم تمثيل هذه الكيانات في عرض الكيان من خلال تركيبها من مجموعة من الأعمدة المستخلصة من الجدول العلائقي باستخدام نوع بنائي مائتم (Ratio Group,2001, 18) (Russell,1999,15). والشكل ٤ يمثل الخطوات البرمجية اللازمة لتكوين نوع الكيان المتداخل في عرض الكيان لجدول اللجان وهذه الخطوات هي :

١. تكوين الجدول العلائقي والقيام باستخلاص أنواع الكيانات المطابقة للأعمدة في الجدول العلائقي .
٢. تكوين نوع الكيان المتداخل من خلال تضمين أنواع الكيانات الخاصة به والتي تم تكوينها في الخطوة ١ .
٣. تكوين عرض الكيان الخاص بنوع الكيان المتداخل الذي تم تكوينه في الخطوة ٢ .

ولقد تم استخلاص نوع الكيان الخاص بالكتب الرسمية Book_t من الجدول العلائقي للجبان ، وتكوين نوع الكيان المتداخل *consul_t* من خلال تضمين نوع الكيان Book_t فيه ، وتكوين عرض الكيان *consul_view_t* الخاص بنوع الكيان المتداخل للجبان .

مقطع برمجي لتكوين الجدول العلائقي للجبان	مقطع لتكوين نوع الكيان للكتب الرسمية
<pre>CREATE TABLE "EXTIN".consul (consul_cod NUMBER(3) NOT NULL , consul_typ VARCHAR2(50) NULL , book_no VARCHAR 2(20) NULL , book_dat DATE , note VARCHAR 2(75) NULL , CONSTRAINT con1 PRIMARY KEY (consul_cod), CONSTRAINT con2 UNIQUE (consul_cod));</pre>	<pre>CREATE OR REPLACE TYPE "EXTIN".Book_t AS OBJECT (book_no VARCHAR2(20), Book_dat DATE);</pre>
مقطع برمجي لتكوين نوع الكيان المتداخل لجدول الجبان	
<pre>CREATE OR REPLACE TYPE "EXTIN". <i>consul_t</i> AS OBJECT (consul_cod NUMBER(3) , consul_typ VARCHAR 2(50) , book <i>Book_t</i> , note VARCHAR 2(75));</pre>	
مقطع برمجي لتمثيل الكيانات المتداخلة في عرض الكيان للجبان	
<pre>CREATE OR REPLACE VIEW <i>consul_view_t</i> OF <i>consul_t</i> WITH OBJECT OID (consul_cod) AS SELECT C.consul_cod , C.consul_typ , <i>Book_t</i>(C.book_no,C.book_dat) , FROM consul C;</pre>	

الشكل ٤

تكوين نوع الكيان المتداخل في عرض الكيان

٣. تكوين المراجع إلى عروض الكيان

Reating References To Object Views

تدار العلاقات بين عروض الكيان من خلال المراجع التي تربط بين عرض الكيان الرئيس وعرض الكيان المرتبط به باستخدام إيعاز MAKE REF الذي يقوم ببناء هذه المراجع بطريقة مشابهة لعلاقة المفتاح الأساسي PK والمفتاح الأجنبي FK في الجداول العلائقية (Boss,2000, 15) (Silbres (Rob and Coronel,2000,9) and Henry,1997,12) . والخطوات البرمجية الآتية توضح عملية تكوين المراجع بين عروض الكيان والتي تم تطبيقها على جدول الشطب والتحويل وعلى النحو الموضح في الشكل ٥ .

١. بناء نوع الكيان على أساس الأعمدة في الجدول العلائقي (الرئيس). اذ يقوم معرف الكيان ID بإعطاء قيمة أحادية لكل سطر في الكيان، وهذا يمثل المفتاح الأساسي في الجدول العلائقي.
 ٢. بناء عرض الكيان (الرئيس) على أساس نوع الكيان المعرف في الخطوة ١ من خلال استخلاص الأعمدة من الجدول العلائقي . وعلى النحو الموضح في الأشكال ٣ و ٤ الخاصة بتعريف أنواع الكيانات الخاصة باللجان والجهات.
 ٣. تكوين عرض الكيان (الفرعي) المرتبط مع عرض الكيان الرئيس من خلال الخطوات الآتية:
 أولاً: تكوين نوع الكيان الذي يحتوي على خصائص تعمل بوصفها مرجعاً REF إلى معرف الكيان OID الخاص بنوع الكيان (الرئيس) وهذه الخصائص تمثل المفتاح الأجنبي في الجدول العلائقي.
 ثانياً: بناء عرض الكيان (الفرعي) على أساس نوع الكيان المعرف في الخطوة ٣ - أولاً وباستخدام إيعاز MAKE_REF يتم عمل تأشير من كل سطر في عرض الكيان (الفرعي) إلى معرف الكيان الأحادي OID في عرض الكيان (الرئيس) .
- ولقد تم بناء نوع الكيان للشطب والتحويل على أساس أعمدة جدول الشطب والتحويل، إذ تمثل خاصيتا *agent_cod* , *consul_cod* مرجع REF إلى معرف الكيان OID الخاص بنوع الكيان الرئيس *agent_t* , *consul_t*، وتقابل هاتان الخاصيتان عمودي *agent_cod* *consul_cod* في جدول الشطب والتحويل. وبني عرض الكيان الخاص بالشطب والتحويل على أساس نوع الكيان، ويقوم إيعاز MAKE_REF بتركيب المفتاح الأجنبي للجدول العلائقي والممثل الـ *D.agent_cod* , *D.consul_cod* مع الكيان الحقيقي في العرض الرئيس *agent_view_t* , *consul_view_t* لعمل مرجع إلى الكيان الرئيس باسم *Agent_cod* , *Consul_cod*

مقطع برمجي لتكوين الجدول العلائقي للشطب والتحويل

```

CREATE TABLE "EXTIN".del_trans
(
cod          NUMBER(6)  NOT NULL ,
qty          NUMBER(12,3)  NULL ,
unit_pric   NUMBER(12,3)  NULL ,
use_date    DATE          NULL ,
del_valu    NUMBER(12,3)  NULL ,
del_typ     VARCHAR2(15)  NULL ,
convert     NUMBER(1,0)   NULL ,
note        VARCHAR2(50)  NULL ,
agent_cod   NUMBER(2)     ,
consul_cod  NUMBER(3)     ,
CONSTRAINT trn1 FOREIGN KEY (cod) REFERENCES item(cod) ,
CONSTRAINT trn2 FOREIGN KEY (agent_cod) REFERENCES agent
(agent_cod) ,
CONSTRAINT trn3 FOREIGN KEY (consul_cod)REFERENCES consul
(consul_cod),
CONSTRAINT trn4 check (del_typ IN ('محول','شطب','هداء')),
CONSTRAINT trn5 check (convert BETWEEN 0 AND 1)
);
    
```

مقطع برمجي لتكوين نوع الكيان لجدول الشطب والتحويل

```

CREATE OR REPLACE TYPE "EXTIN".del_trans_t AS OBJECT
(
cod          REF          item_trans ,
dele        ITEM_T       ,
del_typ     VARCHAR2(15) ,
convert     NUMBER(1,0)  ,
note        VARCHAR2(50) ,
agent_cod   REF          agent_t   ,
consul_cod  REF          consul_t  );
    
```

مقطع برمجي لتكوين عرض الكيان لجدول الشطب والتحويل

```

CREATE OR REPLACE VIEW deltrans_view_t AS
SELECT
MAKE_REF ( item_view_t , D.cod )    COD,
ITEM_T (D.qty,D.unit_pric,D.USE_datE)  ITEM,
D.del_typ                             ,
D.convert                              ,
D.note                                 ,
MAKE_REF (agent_view_t , D.agent_cod ) Agent_cod ,
MAKE_REF ( consul_view_t , D.consul_cod ) Consul_cod
FROM del_trans D;
    
```

الشكل ٥

تكوين المراجع إلى عروض الكيان

٤. استخدام أنواع المجاميع في عروض الكيان

Using Collection Type in Object Views

يمكن أن تكون أنواع المجاميع بضمنها مصفوفة المتغيرات والجدول المتداخل أعمدة في العروض الكيانية، وبذلك تنتقي الحاجة إلى الروابط بين الجدول الرئيس والجدول الفرعي، ويتم تركيب هذه المجاميع باستخدام الاستفسار الفرعي SELECT الذي يكتب داخل المقطع البرمجي الخاص بجملته - CAST MULTISet، ويقوم هذا الاستفسار بإرجاع قائمة من العناصر الخاصة بالجدول الفرعي التي تنتمي للجدول الرئيس، إذ يقوم إيعاز CAST بتسقيط قائمة العناصر الناتجة عن الاستفسار في نوع بيانات المجاميع والذي يسمى عادةً Name_List_t (Name) هو أي اسم يتم اختياره) وكلمة Multiset تعني مجموعة متعددة من القيم (Oracle Technical,2001, 4) (Russell,1999,15) (Whithed,2000,15) .

Varying Array

٤ - ١ مصفوفة المتغيرات

تسمح مصفوفة المتغيرات بتعريف مجموعة من العناصر المرتبة والمحددة الحجم في الجدول العلائقي ويقوم الاستفسار الفرعي داخل إيعاز - CAST MULTISet بإرجاع مجموعة العناصر التي تنتمي للجدول الرئيس (Russell,1999,15) (Whithed,2000,15) . وبتطبيق الخطوات البرمجية الآتية على جدول الموظفين وجدول الدرجات الوظيفية وباستخدام الاستفسار الفرعي في العرض الكياني يتم إرجاع قائمة من الموظفين ومع كل موظف قائمة من العناوين الوظيفية العائدة له وعلى النحو المبين في الشكل ٦ .

١. بناء نوع الكيان (الفرعي) على أساس الأعمدة في الجدول العلائقي (الفرعي) .
٢. بناء نوع بيانات المجاميع (Varray) على أساس نوع الكيان (الفرعي) المعروف في الخطوة ١ .

٣. بناء نوع الكيان (الرئيس) على أساس الأعمدة في الجدول العلائقي (الرئيس) .
٤. تكوين عرض الكيان الخاص بنوع الكيان (الرئيس) مع تضمين نوع الكيان الفرعي بداخله باستخدام إيعاز CAST_MULTISet وتخزنه باسم نوع بيانات المجاميع (Varray) من خلال كلمة AS. إذ تم تكوين الجدول العلائقي (الفرعي) الخاص بالدرجات الوظيفية مع تكوين نوع كيان الدرجات الوظيفية job_t وتكوين نوع بيانات المجاميع JOB_LIST_T كـ Varray لكيان الدرجات الوظيفية job_t . ثم تكوين الجدول العلائقي (الرئيس) الخاص بالموظفين وتكوين نوع الكيان المستخلص منه employee_t مع تكوين عرض الكيان الخاص به employee_view_t وتضمين نوع الكيان الخاص بالدرجات الوظيفية بداخله باستخدام إيعاز CAST_MULTISet وتخزنه باسم JOB_LIST_T من خلال كلمة AS.

مقطع برمجي لتكوين الجدول العلائقي للدرجات الوظيفية
<pre>CREATE TABLE "EXTIN".job (job_cod NUMBER(2) NOT NULL , job_degree VARCHAR2(30) NULL, job_address VARCHAR2(30) NULL , job_date DATE NULL , book_no VARCHAR2(20) NULL , CONSTRAINT job1 PRIMARY KEY (job_cod), CONSTRAINT job2 UNIQUE (job_cod));</pre>
مقطع برمجي لتكوين نوع الكيان للدرجات الوظيفية
<pre>CREATE TYPE "EXTIN".job_t AS OBJECT (job_cod NUMBER(2) , job_degree VARCHAR2 (30) , job_address VARCHAR2 (30) , job_date DATE , book_no VARCHAR2 (20));</pre>
مقطع برمجي لتكوين نوع بيانات المجاميع للدرجات الوظيفية
<pre>CREATE OR REPLACE TYPE "EXTIN".JOB_LIST_T AS VARRAY(20) OF job_t;</pre>
مقطع برمجي لتكوين الجدول العلائقي للموظفين
<pre>CREATE TABLE "EXTIN".employee (employee_cod NUMBER(6) NOT NULL, first_nam VARCHAR2(30) NULL, last_nam VARCHAR 2(30) NULL , custom_nam VARCHAR 2(30) NULL, dept_cod NUMBER(2) NULL , job_cod NUMBER(3) NULL , CONSTRAINT emp1 PRIMARY KEY (employee_cod), CONSTRAINT emp2 UNIQUE (employee_cod), CONSTRAINT emp3FOREIGNKEY(dept_cod)REFERENCES EXTIN".department(dept_cod), CONSTRAINT emp4 FOREIGN KEY (job_cod) REFERENCES EXTIN".JOB(job_cod)</pre>
مقطع برمجي لتكوين نوع الكيان لجدول الموظفين
<pre>CREATE OR REPLACE TYPE "EXTIN".employee_t AS OBJECT (employee_cod NUMBER(6), emp_nam NAME_T , dept_COD REF DEPARTMENT_T, job_cod NUMBER(3) , emp_job JOB_LIST_T);</pre>
مقطع برمجي لتكوين عرض الكيان الخاص بالموظفين مع تضمين بيانات المجاميع للدرجات الوظيفية
<pre>CREATE OR REPLACE VIEW employee_view_t OF employee_t WITH OBJECT OID (employee_cod) AS SELECT E.EMPLOYEE_cod, NAME_T(E.first_nam,E.last_nam,E.custom_nam), MAKE_REF(DEPARTMENT_VIEW_T,E.dept_cod) DEPT_COD, CAST(MULTISSET(SELECT JOB_T(J.job_cod,J.job_degree,J.job_address,J.job_date,J.book_no) FROM job J WHERE E.job_cod=J.job_cod) AS JOB_LIST_T) FROM employee E;</pre>

الشكل ٦

تكوين نوع بيانات المجاميع (مصفوفة المتغيرات) في عرض الكيان

Nested Table

٤- ٢- الجدول المتداخل

يقوم الجدول المتداخل بتعريف مجموعة من العناصر غير المرتبة وبأحجام غير محددة في الجدول العلائقي . ويقوم الاستفسار الفرعي داخل إيعاز CAST MULTISSET بإرجاع مجموعة العناصر التي تنتمي للجدول الرئيس (Russell,1999,15) (Whithed,2000,15). والخطوات البرمجية الآتية توضح كيف يتم تكوين الجدول المتداخل الخاص بالمواد المطلوب شراؤها في العرض الكياني الخاص بطلبات الشراء وعلى النحو الموضح في الشكل ٧.

١. بناء نوع الكيان (الفرعي) على أساس الأعمدة في الجدول العلائقي (الفرعي) .
٢. بناء نوع بيانات المجاميع (Nested table) على أساس نوع الكيان (الفرعي) المعرف في خطوة ١ .
٣. بناء نوع الكيان (الرئيس) على أساس الأعمدة في الجدول العلائقي (الرئيس).
٤. تكوين عرض الكيان الخاص بنوع الكيان (الرئيس) مع تضمين نوع الكيان الفرعي بداخله وباستخدام إيعاز CAST_MULTISSET .

إذ تم تكوين الجدول العلائقي (الفرعي) الخاص بالمواد المطلوب شراؤها مع تكوين نوع الكيان الخاص به *orderitem_t* وتكوين نوع بيانات المجاميع **ORDERITEM_TABLE** بوصفه جدولاً كيانياً للمواد المطلوب شراؤها *orderitem_t* . ثم تكوين الجدول العلائقي (الرئيس) الخاص بطلبات الشراء وتكوين نوع الكيان المستخلص منه *saleorder_t* مع تكوين عرض الكيان الخاص به *saleorder_view_t* وتضمين نوع الكيان الخاص بالمواد المطلوب شراؤها بداخله باستخدام إيعاز CAST_MULTISSET وخرزته باسم **ORDERITEM_TABLE** .

مقطع برمجي لتكوين الجدول العلائقي للمواد المطلوب شراؤها
<pre>CREATE Table "EXTIN".order_item (order_id NUMBER(4) , qty NUMBER(12,3) , unit_pric NUMBER(12,3) , use_date DATE , product_id NUMBER(6) , CONSTRAINT OR1 FOREIGN KEY (ORDER_ID) REFERENCES sale_order(ORDER_ID)) ;</pre>
مقطع برمجي لتكوين نوع الكيان للمواد المطلوب شراؤها
<pre>CREATE OR REPLACE TYPE "EXTIN".orderitem_t AS OBJECT (order_id NUMBER(6) , item ITEM_T , product_id NUMBER(6));</pre>
مقطع برمجي لتكوين جدول الكيان للمواد المطلوب شراؤها
<pre>CREATE OR REPLACE TYPE "EXTIN".ORDERITEM_TABLE AS TABLE OF orderitem_t</pre>
مقطع برمجي لتكوين الجدول العلائقي لطلبات الشراء
<pre>CREATE TABLE sale_order (order_id NUMBER(6) NOT NULL , order_dat DATE NULL , employee_cod NUMBER(6) NULL , ship_dat DATE NULL , total NUMBER(8,3) NULL , CONSTRAINT s11 PRIMARY KEY (order_id) , CONSTRAINT s12 UNIQUE (order_id) , CONSTRAINT s13 FOREIGN KEY (employee_cod) REFERENCES employee(employee_cod)</pre>
مقطع برمجي لتكوين نوع الكيان لجدول طلبات الشراء
<pre>CREATE OR REPLACE TYPE "EXTIN".saleorder_t AS OBJECT (order_id NUMBER(4) , order_dat DATE , employee_cod REF employee_t , ship_dat DATE , total NUMBER(8,3) , items ORDERITEM_TABLE , MEMBER FUNCTION total_value RETURN NUMBER , PRAGMA RESTRICT_REFERENCES (total_value, WNDS, WNPS));</pre>
مقطع برمجي لتكوين عرض الكيان الخاص بطلبات الشراء مع تضمين نوع بيانات المجاميع للمواد المشتراة
<pre>CREATE OR REPLACE VIEW "EXTIN".saleorder_view_t OF saleorder_t WITH OBJECT OID (order_id) AS SELECT SO.order_id, SO.order_dat, MAKE_REF(EMPLOYEE_VIEW_T,SO.EMPLOYEE_cod) EMPLOYEE_COD, SO.ship_dat, SO.total, CAST(MULTISET(SELECT Orderitem_t(I.order_id, ITEM_T(I.qty , I.unit_pric , I.use_date), I.product_id) FROM order_item I WHERE I.order_id=SO.order_id) AS ORDERITEM_TABLE) FROM sale_order SO;</pre>

الشكل ٧

تكوين نوع بيانات المجاميع (الجدول المتداخل) في عرض الكيان

```

مقطع برمجي يوضح القادح المستخدم لتحديث عرض الكيان الخاص بالموظفين
CREATE OR REPLACE TRIGGER employee_trigger_t
INSTEAD OF INSERT OR UPDATE OR DELETE ON employee_view_t
FOR EACH ROW
DECLARE
    v_message VARCHAR2(100);
    v_dept DEPARTMENT_t;
    THIS_JOB JOB_t;
BEGIN
    IF INSERTING THEN v_message:="إضافة قيد جديد إلى جدول الموظفين والدرجات الوظيفية باستخدام عرض الكيان"
        SELECT Deref (:NEW.dept_cod) INTO v_dept from DUAL;
        INSERT INTO employee VALUES
            ( :NEW.employee_cod ,
              :NEW.emp_nam.first_nam ,
              :NEW.emp_nam.last_nam ,
              :NEW.emp_nam.custom_nam ,
              v_dept.dept_cod ,
              :NEW.job_cod );
            FOR I IN 1..20
            LOOP
                THIS_JOB := :NEW.emp_job(I);
                INSERT INTO job VALUES
                    ( THIS_JOB.job_cod,
                      THIS_JOB.job_degree,
                      THIS_JOB.job_address,
                      THIS_JOB.job_date,
                      THIS_JOB.book_no );
            END LOOP;
    ELSIF UPDATING THEN v_message:="تحديث قيد في جدول الموظفين والدرجات الوظيفية باستخدام عرض الكيان"
        SELECT Deref (:NEW.dept_cod) INTO v_dept FROM DUAL;
        UPDATE employee SET
            Employee_cod = :NEW.employee_cod ,
            first_nam = :NEW.emp_nam.first_nam ,
            last_nam = :NEW.emp_nam.last_nam ,
            custom_nam = :NEW.emp_nam.custom_nam ,
            dept_cod = v_dept.dept_cod ,
            job_cod = :NEW.job_cod WHERE employee_cod = :OLD.employee_cod;
        FOR I IN 1..20
        LOOP
            THIS_JOB := :NEW.EMP_JOB(I);
            UPDATE job SET
                Job_cod = THIS_JOB.job_cod,
                Job_degree = THIS_JOB.job_degree,
                Job_address = THIS_JOB.job_address,
                Job_date = THIS_JOB.job_date,
                Book_no = THIS_JOB.book_no
            WHERE job_cod = :OLD.job_cod;
        END LOOP;
    ELSE
        v_message:="حذف قيد من جدول الموظفين والدرجات الوظيفية باستخدام عرض الكيان"
        DELETE FROM job WHERE job_cod = :OLD.job_cod;
        DELETE FROM employee WHERE job_cod = :OLD.job_cod;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20007,'ERROR:'||v_message||'-'||SQLERRM);
END;

```

الشكل ٨

القادح المستخدم لتحديث عرض الكيان لمصفوفة المتغيرات

```

مقطع برمجي يوضح القادح المستخدم لتحديث عرض الكيان الخاص بطلبات الشراء
CREATE OR REPLACE TRIGGER "EXTIN". Saleorder
INSTEAD OF INSERT OR UPDATE OR DELETE ON saleorder_view_t
FOR EACH ROW
DECLARE
    v_message    VARCHAR2(100);
    v_employee   EMPLOYEE_t;
    v_count      INTEGER;
    THIS_ITEM    ORDERITEM_T;
    i            INTEGER;
BEGIN
IF INSERTING THEN
    SELECT Deref (:NEW.EMPLOYEE_cod) INTO v_employee FROM DUAL;
    v_message:= "اضافة قيد جديد الى ملف طلبات الشراء والمواد المطلوبة باستخدام عرض الكيان"
    INSERT INTO sale_order VALUES
    ( :NEW.order_id ,
      :NEW.order_dat ,
      v_employee.employee_cod ,
      :NEW.ship_dat ,
      :NEW.total
      );
    SELECT COUNT(*) INTO v_count FROM order_item
    WHERE order_item.order_id =:new.order_id;
    FOR I IN 1..v_COUNT
    LOOP
        THIS_ITEM := :NEW.items(i);
        INSERT INTO order_item
        VALUES(:NEW.order_id, THIS_ITEM.item.qty, THIS_ITEM.item.unit_pric
        THIS_ITEM.item.use_date,i);
    END LOOP;
ELSIF UPDATING THEN
    v_message:= "تحديث قيد في ملف طلبات الشراء والمواد المطلوبة باستخدام عرض الكيان"
    SELECT Deref (:NEW.employee_cod) INTO v_employee FROM DUAL;
    UPDATE sale_order SET order_id =:NEW.order_id , order_dat=:NEW.order_dat ,
    Employee_cod = v_employee.employee_cod , ship_dat =:NEW.ship_dat ,
    Total =:NEW.total WHERE order_id =:OLD.order_id;
    SELECT COUNT(*) INTO v_count FROM order_item
    WHERE order_item.order_id =:OLD.order_id;
    FOR I IN 1..v_COUNT
    LOOP
        THIS_ITEM := :NEW.items(i);
        UPDATE order_item SET
        Order_id= :NEW.order_id,
        qty = THIS_ITEM.item.qty,
        unit_pric = THIS_ITEM.item.unit_pric,
        use_date = This_Item.item.use_date,
        product_id = i ;
    END LOOP;
ELSE
    v_message:= "حذف قيد من ملف طلبات الشراء والمواد المطلوبة باستخدام عرض الكيان"
    DELETE FROM order_item WHERE order_id = :OLD.order_id;
    DELETE FROM sale_order WHERE order_id = :OLD.order_id;
END IF;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20009,'ERROR:"||'-'||SQLERRM);
END;
```

الشكل ٩

القادح المستخدم لتحديث عرض الكيان للجدول المتداخل

٥. تحديث عروض الكيان Updating Object Views

تستخدم عروض الكيان اللغة الخاصة بمعالجة البيانات DML لتحديث الجداول العلائقية الأساسية التابعة للعروض الكيانية. وقد وفرت اللغة البرمجية المستخدمة Oracle طريقة شفافة لتحديث عروض الكيان أو العروض العلائقية بدلاً من الجمل المباشرة الخاصة بلغة معالجة البيانات من خلال استخدام قادحات INSTEAD OF Trigger، وتسمى بهذا الاسم لتنفيذ هيكل القادح مع العملية المطلوبة، مثل الإضافة والحذف والتحديث بدلاً من استخدام الجمل الخاصة بلغة SQL، وتوفر هذه القادحات طريقة لتحديث العروض المعقدة ومنها عروض الكيان التي تحتوي على أنواع المجاميع مثل مصفوفة المتغيرات، والجدول المتداخل وعند إجراء عمليات معالجة البيانات مثل إضافة عناصر جديدة أو تحديث وحذف العناصر الموجودة على أعمدة Varray، Nested table، يتم إطلاق قادحات INSTEAD OF (دايتز، ٢٠٠٠، ١٠) (بيتر و دروسي، ٢٠٠١) (Portfolio, 1999, 13). والشكل ٨ يوضح القادح المستخدم لتحديث مصفوفة المتغيرات في عرض الكيان الخاص بالموظفين والمعرف في الشكل ٥، والشكل ٩ يوضح القادح المستخدم لتحديث الجدول المتداخل في عرض الكيان الخاص بطلبات الشراء والمعرف في الشكل ٧.

٦. نمذجة العلاقات العكسية باستخدام عروض الكيان

Modeling Inverse Relation Ships With Object Views

تستخدم العروض مع الكيانات في عمل نموذج من العلاقات العكسية والتمثلة في علاقات الواحد -إلى- الواحد One-To-One وعلاقات الواحد -إلى- العديد One-To-Many. ويمثل نموذج علاقات الواحد -إلى- الواحد في النموذج العلائقي باستخدام المفتاح الأجنبي بين الجداول، أما الكيانات فتمثل هذا النموذج من العلاقات من خلال المراجع REFs بين الكيانات، وتقوم عروض الكيان بتمثيل نموذج العلاقات هذا بواسطة المراجع REF بين الكيانات وباستخدام إيعاز .oracle (developer, 2000, 17) MAKE_REF.

ويمثل نموذج علاقات الواحد -إلى- العديد أو علاقات العديد -إلى- الواحد في النموذج العلائقي باستخدام المفتاح الأجنبي بين الجداول، أما الكيانات أو عروض الكيان فتمثل هذا النموذج من العلاقات باستخدام مراجع الكيان أو باستخدام الكيانات المتضمنة (Russell, 1999, 15) embedding object (Skora, 2001, 6). مثلاً جدول طلبات الشراء وجدول المواد المطلوب شراؤها يتم ربطهما في النموذج العلائقي باستخدام المفتاح الأجنبي. أما عروض الكيان فتمثل هذه العلاقة باستخدام مجاميع الكيانات إذ إن العرض الكياني الخاص بالمواد المطلوب شراؤها يمتلك مراجع REFs إلى العرض الكياني لطلبات الشراء وهذه تمثل حالة العلاقات الأمامية Forward Relation وعرض الكيان الخاص بطلبات الشراء يمتلك مجاميع من المواد المطلوب شراؤها المتضمنة فيه من خلال إيعاز CAST_MULTISSET. وهذه تمثل حالة العلاقة العكسية Inverse Relation.

الاستنتاجات

- إن استخدام عروض الكيانات يوفر تطويراً للأنظمة العلائقية الحالية من خلال:
 - عدم اللجوء إلى تفاصيل النظام العلائقي إذ إنها تقوم بعمل عرض كياني مشتق منه ومرتبطة معه، وينتج عن ذلك عدم ضياع الجهود السابقة في إعداد وتصميم الأنظمة العلائقية المتوفرة حالياً، ومن ثم فإنها تعطي خطوة جديدة نحو تطبيقات OO وتوحيد التطبيقات.
 - المرونة العالية في تكوين نموذج كياني جديد أو تطوير نموذج كياني موجود من خلال استبدال العروض الكيانية السابقة بتعريفات لعروض كيانية جديدة وللبيانات نفسها .
 - وفرت طرائق جديدة لعمل استفسار على البيانات العلائقية إذ تم عرض البيانات من الجداول العلائقية المختلفة من خلال تركيبها في نوع كياني جديد باستخدام إيعاز MAKE_REF بدلاً من كتابة الروابط المعقدة مع الجداول المتعددة.
 - كفاءة أفضل في تقليل جمل SQL المستخدمة للاستفسار، ينتج عن ذلك تقليلاً لحركة المرور والعثرات التي تحدث على الشبكة في حال نقل البيانات عليها، إذ إن البيانات العلائقية تقوم ببناء الأسطر الخاصة بعرض الكيان بوصفها وحدة واحدة .
 - تم الحصول على عرض كياني مكون من البيانات الرئيسية مع تضمين البيانات المترتبة بها باستخدام إيعاز CAST _ MULTISSET إذ وفر تمثيلاً لأنواع بيانات المجاميع وعدم الحاجة إلى الروابط المتعددة.
 - تم استخدام لغة Oracle8 بوصفها لغة برمجية توفر دعماً لمواصفات قواعد البيانات الكيانية العلائقية، فضلاً عن تفردتها في توفير قادات INSTEAD OF الخاصة بإدخال البيانات ومعالجتها في الجداول العلائقية باستخدام العروض الكيانية أو العروض العلائقية .

المراجع

أولاً - المراجع باللغة العربية

١. الاء فيصل، نمذجة قواعد البيانات الكيانية العلائقية باستخدام اوراكل، رسالة ماجستير، جامعة الموصل، كلية علوم الحاسبات والرياضيات، ٢٠٠٢.
٢. دايتز كارول، اوراكل بايبل، دار الفاروق للنشر والتوزيع، مركز التعريب والترجمة، مصر، ٢٠٠٠.
٣. كوليتريك بيتر، بول دروسي، أوراكل ديفلوير النماذج والتقارير المتقدمة، دار الفاروق للنشر والتوزيع، مركز التعريب والترجمة، مصر، ٢٠٠١.

ثانياً - المراجع باللغة الاجنبية

1. Beaverton O. R .Object-Relational and Object-Oriented Database System, 2001.
<http://www.software.ibm/is/sw-servers/database>.

2. Boss E, Working With Oracle8 DataType , 2000.
http://www.boss-consulting-inc.com/data_typesppt.pdf.
3. Giese H, Design Pattern and Software Architecture, 2001.
http://www.odtug.com/2001_papers/sikora.pdf.
4. Johnny O & Allan R. L & .Experiences from, 2000.
5. 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>.
6. Lonsdale M .Is Performance a Reason for Using Oracle8 Object?, 1999.
<http://www.softlab.co.uk/news/uploads/ISPERFO8.pdf>
7. Oracle-developer Introduction to Object-Relational Database Development, 2000.
<http://www.kingtraining.com/downloads/o8diffs-paper.pdf>.
8. Oracle Technical White Paper, Simple Strategies for Complex Data : Oracle9i Object-Relational Technology, 2001.
http://www.ont.oracle.com/products/oracle9i/pdf/iot_twp.pdf.
9. Portfolio T. , PL/SQL Users Guide and Reference, 1999 .
<http://www.technet.oracle.com>
10. Ratio Group Ltd, Persistence : Implementing Object Over a Relation Database version 1.0, 2001.
<http://www.ratio.co.uk>.
11. Rob P.; Coronel C., Data base System Design , Implement and Management , 4th , course technology , 2000 .
12. Russell J. , Application Developers Guide, 1999 .
13. Object-Relational Features & http://www.thinkspark.co.uk/ioug/PLSQL_New_Features_in_oracle8i.pdf.
14. Sikora Y ., Peter J.L ..Object to Object Communication , 2001.
<http://www.odtug.com> .
15. Silbers A. Henry F. K & .Sudarshan S ..Database System Concepts , 3rd , Mc Graw-Hill Companies,Inc, 1997.
16. Whithead A. N ..Object-Relational Oracle8 and PL/SQL8, 2000.
<http://www.polito.it/ivrea/inofrmazioni/passaggio/dispenseinseif/basi/PLSQL8.pdf>

The Application of Management System ORDBMS Object Relational Data Base in System by Using ORACCL

ABSTRACT

This research implements the concept of Object Relational Data Base Management System (ORDBMS) on depreciation by converting it from relational application to object application without changing current relational application and dealing with its details by using Object Views. Object Views can be derived from the relational system and form a layer of object oriented program above the relational table. Thus ,it gives us new method for making queries on relational database and decreasing SQL sentences by using MAKE_REF instead of making complex relations between multiple tables so data is viewed as one unit. by using CAST_MULTISSET multiple data can be implemented into object view without making multiple relational between collections.

The application is programmed using Oracle8 because it has the propriety of object relational data base and also it provides INSTEAD OF trigger for inserting and manipulating data in the relational table by using object views.