المحاضرة الرابعة لغة SQL الجزء الثاني

مخبر نظم قواعد البيانات – BDBL

SLASH TEAM

برنامج الهندسة المعلوماتية – الجـــــامعة الافتراضية السورية

تقديم	يوسف الزنيفير
وصف	المحاضرة الأولى: لغة SQL الجزء الثاني
عدد الصفحات	15

لغة SQL الجزء الثاني

توصيف المحاضرة

التوصيف:

تعريف وشرح المناظير مع تكملة توابع من نمط تاريخ وإعطاء بعض التوابع الأخرى الشائعة.

طريقة الدراسة:

نعيد ونكرر أن اهم نقطة لدراسة اللغة هي التأكيد على التجربة المباشرة على اوراكل مع تجربة جميع الاكواد.

توابع النمط تاريخ: (تكملة)

2-4 الدوال (التوابع):

تمتلك SQL مجموعة كبيرة من التوابع التي يمكنك استخدامها في حقول الخرج او ضمن الشرط او في الترتيب وتقسم بحسب نمط الخرج. (أي توابع جاهزة تنفذ مجموعة من التعليمات)

توابع رقمية:

الدالة	عمل الدالة
ABS(n)	القيمة المطلقة للعدد n
CEIL(n)	أصغر عدد صحيح أكبر او يساوي n
FLOOR(n)	أكبر عدد صحيح أصغر او يساوي n
MOD(m,n)	الباقي الصحيح من قسمة العددmعلى العدد n
Round(m,[n])	تقريب العدد m الى عدد مع عدد خانات عشرية هو n
SIGN(n)	1 في حال n موجب -1 في حال سالب 0غير ذلك.

توابع للسلاسل المحرفية:

عمل الدالة	الدالة
لدمج سلسلتي محارف	CONCAT (char1,char2)
لحذف الفراغات من الجانب الأيسر	LTRIM(char[,set])
لاستبدال سلسلة محارف فرعية بسلسة أخرى ضمن سلسلة	REPLACE(char[,searchstring[, replacestring]])
لحذف الفراغات من الجانب الأيمن	RTRIM(char[,set])
لاقتطاع جزء من سلسلة ابتداءً من الموقع n في السلسلة إلى نهايتها أو بطول m في حال تم إدخاله	SUBSTR(string.n[.m])
تعيد موقع substr ضمن str ابتداءً من الموقع n وبالظهور occur	INSTR(str,substr[,pos[,occur]])

بعض توابع التحويل التي تستخدم في تحويل الى أنماط بديلة:

عمل الدالة	الدالة
تحويل القيمة d من نمط تاريخ إلى سلسلة محارف ويمكن استخدام تنسيق معين في الدخل الثاني الاختياري	TO_CHAR (d [fmt])
تحويل القيمة num من نمط رقمي إلى سلسلة محارف ويمكن استخدام تنسيق معين في الدخل الثاني الاختياري	TO_CHAR (num [,fmt])
تحويل القيمة str من نمط سلسلة محارف إلى تاريخ ويمكن استخدام تنسيق معين في الدخل الثاني الاختياري	TO DATE (str [,fmt])
تحويل القيمة str من نمط سلسلة محارف إلى رقم ويمكن استخدام تنسيق معين في الدخل الثاني الاختياري	TO_NUMBER(str [,fmt])

بعض دوال نمط تاريخ:

الدالة	عمل الدالة
LAST_DAY(dt)	تعيد تاريخ اليوم الأخير من الشهر لتاريخ الدخل dt
SYSDATE	قيمة التاريخ الحالي من نظام التشغيل
TRUNC(dt)	لتصفير الوقت من التاريخ dt
EXTRACT (interval FROM dt)	لاستخلاص السنة أو الشهر أو اليوم أو الساعة من dt، ننوه هنا أن قيمة interval هي إحدى الكلمات year, month, day, hour, minute, second

أنواع أخرى من الدوال الشائعة للاستخدام:

الدالة	عمل الدالة
COALESCE(expr[.expr....])	الخرج هو أول قيمة غير معدومة من بين قيم الدخل
DECODE (expr,search result [,search,result...][,default])	مقارنة أول قيمة من الدالة مع أول قيمة من ثنائيات الدخل في حال التطابق الخرج هو القيمة الثانية من الثنائية
GREATEST(expr[,expr....])	الخرج هو القيمة الأكبر من بين قيم الدخل
LEAST(expr[,expr,...])	الخرج هو القيمة الأصغر من بين قيم الدخل
NULLIF(expr1,expr2)	الخرج معدوم أي null في حال تطابق القيمتين وإلا يكون الخرج expr1
NVL(expr1,expr2)	الخرج هو expr2 إذا كان expr1 معدوماً وإلا يكون الخرج expr1
NVL2(expr1,expr2,expr3)	الخرج هو expr2 إذا كان expr1 معدوماً وإلا يكون الخرج expr3
USER	الخرج هو اسم حساب المستخدم في الجلسة

اختبار:

يمكن اختبار التوابع التالية على جداول الحساب كما يمكن استخدام جدول مزوّد من النظام هو dualيحوي سجل واحد وعمود واحد فقط بغاية استخدام الدوال من دون أن يكون لدينا الحاجة لتطبيقها على أحد الجداول.

مثال إذا احتجنا معرفة التاريخ الحالي يمكننا كتابة:

```
Select sysdate from dual;
```

أو اسم الحساب الحالي:

```
Select user from dual;
```

أو لاختبار الدالة replace في استبدال كلمة " world "بكلمة 'sql' في السلسلة "hello world":

```
Select replace ('hello world', 'world', 'sql') from dual
```

في الخانة الأولى الجملة كاملة في الخانة الثانية الكلمة المراد استبدالها في الخانة الثالثة الكلمة التي سيتم وضعها.

مثال واقعي لاستخدام الدوال:

قامت أحد المدارس بتنصيب نظام معلومات شامل يستخدم نظام أوراكل لقاعدة البيانات. وكانت جميع المعلومات سابقاً موجودة على ملفات اكسل بما فيها ملف يحوي أسماء الطلاب.

تبين أن المدرسة تقوم بتسجيل الاسم كاملاً في عمود واحد في جدول الإكسيل بينما بنية جدول الطلاب في النظام على الشكل:

```
Students (id number, firstname varchar2(25), lastname varchar2(25),....)
```

تم تهجير ملف الإكسيل على جدول مؤقت student excel له البنية التالية:

```
Studentexcel (id number, fullname varchar2(50),...)
```

نريد الآن تجزيئ الحقل full name في الجدول السابق إلى حقلين الاسم والكنية ثم نسخه إلى جدول الطلاب الأساسي، ولتبسيط الحالة سنستبعد في هذا المثال حالة الأسماء المركبة في اسم الطالب أو الكنية.

بمعنى نريد تقسيم الاسم الكامل الى اسم وكنية مثلا: يوسف في جدول الاسم الأول ، والكنية الزنيفير في جدول الاسم الأخير.

بتحليل المسألة يجب اقتطاع السلسلة المحرفية التي تحوي اسم الطالب من بدايتها إلى مكان الفراغ لنحصل على الاسم الأول أي:

substr(full name, 1,[عدد المحارف إلى الفراغ)

أما عدد المحارف إلى الفراغ فهو يقابل موقع الفراغ في السلسلة أي:

Instr (full name,' ')

فيصبح التعبير الذي يعطينا الاسم الأول:

substr(fullname, 1, Instr(fullname,' '))

ويكون الاسم الأخير على الشكل:

Substr(fullname, Instr(fullname,' ')+1)

وأخيراً يصبح الاستعلام المطلوب:

```
SELECT
    SUBSTR(fullname, 1, INSTR(fullname, ' ')) AS "first name",
    SUBSTR(fullname, INSTR(fullname, ' ') + 1) AS "last name"
FROM studentexcel;
```

يمكنك اختبار الحل من دون أن يكون لديك الجدول باختيار اسم ما وليكن naji alali:

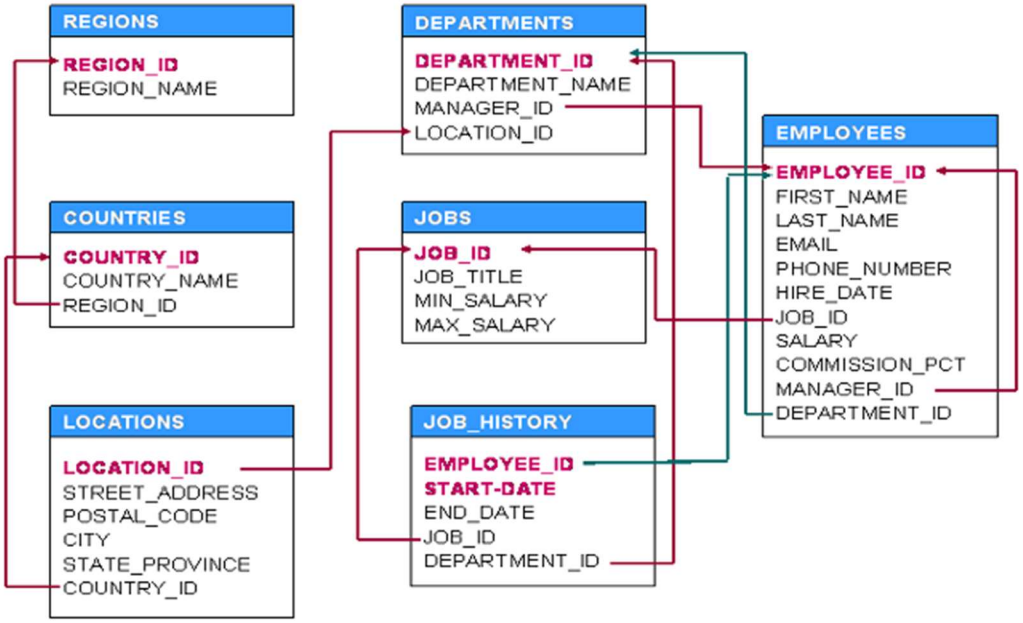
```
SELECT
    SUBSTR('naji alali', 1, INSTR('naji alali', ' ')) AS "first name",
    SUBSTR('naji alali', INSTR('naji alali', ' ') + 1) AS "last name"
FROM dual;
```

توابع أخرى شائعة الاستخدام:

نستخدم Group by لاستخلاص مجموعة جديدة من سجلات جدول (أو أكثر كما سنرى لاحقاً) بعد تجزئ هذا الجدول إلى مجموعات أخرى وتطبيق توابع تجميع أخرى.

اختبار:

سنستخدم الحساب HR في اختبار هذه التعليمة بدلاً من حساب SCOTT البسيط، فيما يلي مخطط الجداول والعلاقات فيما بينها لهذا الحساب.



يوجد في الحساب سبعة جداول هي التالية:

- جدول الموظفين: الرقم الذاتي – الاسم الكنية البريد الهاتف – تاريخ التعيين – رمز الوظيفة – لراتب – نسبة العمولة – الرقم الذاتي للرئيس المباشر – رمز القسم.
- جدول الأقسام: رمز القسم – اسم القسم – الرقم الذاتي لرئيس القسم – رمز الموقع.
- جدول المناطق: رمز المنطقة - اسم المنطقة.
- جدول البلدان: رمز البلد – اسم البلد – رمز المنطقة.
- جدول المواقع: رمز الموقع، عنوان الشارع البريد المدينة، اسم المحافظة أو الولاية، رمز البلد.
- جدول الوظائف: رمز الوظيفة، اسم الوظيفة، الراتب القاعدي، الراتب الأعظمي.
- جدول أرشيف الوظائف: الرقم الذاتي للموظف، تاريخ المباشرة في الوظيفة، تاريخ النهاية، رمز الوظيفة، رمز القسم.

فيما يلي بعض الأمثلة لاختبار التجمع:

1- لعرض عدد العاملين المنتسبين في كل سنة للقسم رقم 30 مع ترتيب السنوات:

```
SELECT
    TO_CHAR(hire_date, 'yyyy') AS year,
    COUNT(*) AS total
FROM employees
WHERE department_id = 30
GROUP BY TO_CHAR(hire_date, 'yyyy')
ORDER BY TO_CHAR(hire_date, 'yyyy');
```

في Oracle يمكن عدم كتابة AS كنوع من الاختصار

احظ أننا استخدمنا الدالة to_char ضمن التجميع والترتيب لاستيعاب الاستعلام السابق يمكن تجزئة الحل إلى مرحلتين.

الأولى هي استخلاص السنة من التاريخ:

```
SELECT
    TO_CHAR(hire_date, 'yyyy') year,
    employees.*
FROM employees
WHERE department_id = 30
ORDER BY TO_CHAR(hire_date, 'yyyy');
```

اختبر الاستعلام السابق سنكون بذلك كما لو أضفنا حقل جديد year على الجدول employee. المرحلة التالية هي تجميع السجلات على أساس السنة أي كما لو أننا نجزئ نتائج السجلات بحسب السنة. لذا لم يعد بالإمكان إظهار باقي الحقول ولكن يمكن تطبيق توابع التجميع على باقي الأسطر، وهذا ما تعنيه الدالة count(*) أي العدد الكلي لمجموعة السجلات:

2- نريد عرض رموز الأقسام التي يعمل بها موظفون بالإضافة إلى الراتب الأعلى والأدنى والوسطي في كل قسم:

```
SELECT
    department_id,
    MAX(salary),
    MIN(salary),
    AVG(salary)
FROM employees
GROUP BY department_id;
```

ملاحظة هامة: لا يمكنك إضافة شرط على توابع التجميع ضمن where ولكن يمكن إضافته باستخدام:

Having بدلاً من where وتأتى بعد استخدام group by

ربط الجداول:

إمكانية الاختيار من مجموعة جداول أمر هام لأن قاعدة المعطيات التي نعمل بها علائقية أي أن المعلومات مترابطة مقسمة إلى مجموعة جداول ترتبط فيما بينها. لذا قد تكون المعلومة التي نبحث عنها في قاعدة البيانات تنتج فقط عند تزويد محرك قاعدة البيانات بهذا الارتباط المنطقي.

1-6 تذكرة عن الجداء الديكارتي:

ليكن لدينا المجموعتين: T1,T2 فيكون الجداء الديكارتي للمجموعتين هو:

T1= {a,b,c), T2= {c,f}

T1 T2={(a,c).(a,f), (b,c), (b,f), (c,c), (c,f)}

يمكن القيام بهذا الجداء إلى أكثر من مجموعتين ((nمجموعة). فإذا أضفنا المجموعة {azz,ttt3 يكون جداء المجموعات الثلاثة هو:

T1*T2*T3=(T1*T2) *T3=={(a,c,azz), (a,f,azz), (b,c,azz), (b,f,azz),(c,c,azz),(c,f,azz),

(a,c,ttt).(a,f,ttt), (b,c,ttt),(b,f,ttt), (c,c,ttt),(c,f,ttt)}}

كما هو ملاحظ نقوم بضرب المجموعتين T1 وT2 ثم اعتبار الناتج مجموعة جديدة ثم نقوم بضرب المجموعة الناتجة مع المجموعة التي تليها.

انتبه إلى أن عدد عناصر المجموعة الناتجة عن الجداء الديكارتي يحسب بالعلاقة التالية:

Card (T1*72* .... Tn) Card(T1) * Card (2) * .... Card(tn) أي حاصل جداء عدد عناصر كل مجموعة.

2-6 ربط الجداول:

يتم ربط الجداول باستخدام مفهوم الجداء الديكارتي حيث يمكن تعريف الجدول بأنه مجموعة من الأسطر ويقابل كل سطر من الجدول عنصر من المجموعة.

إن التعبير select * from T1 يعني اختر لي عناصر المجموعة T1 بأكملها كذلك إذا قلنا select * from T2 اختر جميع السطور من الجدول T2 إذا أردنا تطبيق الجداء الديكارتي عن طريق ال SQL ، يتم ذلك بالعبارة التالية:

Select T1. *, T2. * From T1, T2

وبالتالي ستكون النتيجة هي جدول عدد الأسطر فيه يساوي عدد أسطر الجدول الأول مضروباً بعدد أسطر الجدول الثاني.

مثال: نوضح العملية بالمثال التالي ليكن لدينا الجدولين التاليين:

T1: select * from T1

Pname	Mark
Samer	100
Fuad	50

T2: select * from T2

Empname	Tot	Birthdate
Samer	500	1980/1/1
Basem	50	1972/12/5
Kinda	2455	1971/1/1

T1×T2: select T1. *, T2 *. from T1, T2

pname	Mark	empname	Tot	birthdate
Samer	100	Samer	500	1980/1/1
Samer	100	Bassam	50	1972/12/5
Samer	100	Kinda	2455	1971/1/1
Fuad	50	Samer	500	1980/1/1
Fuad	50	Bassam	50	1972/12/5
Fuad	50	Kinda	2455	1971/1/1

ربط الجداول هو تعبير عن علاقة بين الجداول في الجداء الديكارتي للجدولين.

ربط الجداول مع تعليمة WHERE:

يمكن ربط جدولين بعدة طرق مع وضع شروط حسب الحاجة لأداء هذا الغرض يمكن استخدام تعليمة where.

مثال:

```
select T1. *, T2. * from T1, T2
where T1. Pname-T2. Empname
```

النتيجة:

pname	Mark	empname	Tot	birthdate
Samer	100	Samer	500	1980/1/1

```
select T1. *, T2. * from T1, T2
where T1. Mark =T2.Tot
```

النتيجة:

pname	Mark	empname	Tot	birthdate
Fuad	50	Bassam	50	1972/12/5

```
select T1. *, T2. * from T1, T2
where T1. Mark >=T2.Tot
```

النتيجة:

pname	Mark	empname	Tot	birthdate
Fuad	50	Samer	500	1980/1/1
Fuad	50	Bassam	50	1972/12/5

ربط الجداول مع تعليمة FROM:

لا تكفي تعليمة where لتلبية جميع الاستفسارات التي قد نحتاجها، بسبب الحاجة إلى تضمين كامل أحد الجدولين أو كلاهما والقيم التي تقابلها إن وجدت وإلا وضع null الطريقة المتبعة في ربط الجداول هي تضمين الشرط ضمن From وليس where ويتم ذلك مع العبارات:

Inner join, left join, right join, full outer join

مثال:

بالعودة للمثال:

```
select T1. *, T2. * from T1, T2
where T1. Pname=T2.empname
```

يمكن الحصول على نفس النتيجة:

```
Select T1.*,T2.* from T1 inner join T2 on T1.Pname=T2.empname
```

الآن نريد جميع السجلات من T1 وفقط السجلات التي تحقق الشرط من، T2 نستخدم لتحقيق ذلك left join أو night join عوضاً عن inner join على الشكل التالي:

```
Select T1. *, T2. * from T1 left join T2 on T1. Pname=T2.empname
```

تم كتابة left join لأننا وضعنا T1 على الجانب الأيسر، يمكن الحصول على نفس النتيجة:

```
Select T1. *, T2. * from T2 right join T1 on T1. Pname=T2.empname
```

تم كتابة right join لأننا وضعنا T1 على الجانب الأيمن. ونحن في كلا الطريقتين نريد جلب جميع السجلات من T1.

نتيجة أحد الاستعلامين السابقين هي:

pname	Mark	empname	Tot	birthdate
Samer	100	Samer	500	1980/1/1
Fuad	50			

وأخيراً إذا أردنا جلب جميع السجلات من الجدولين معاً مع مقابلة السطور التي تحقق الشرط نستخدم العبارة الأخيرة full outer join على الشكل التالي:

```
Select T1. *, T2. * from T2 full outer join T1 on T1. Pname=T2.empname
```

يمكن الحصول على نفس نتيجة الاستعلام السابق من دون استخدام full outer join باستخدام union:

```
Select T1. *, T2. * from T1 left join T2 on T1. Pname=T2.empname
Union
Select T1. *, T2. * from T1 right join T2 on T1. Pname=T2.empname;
```

المناظير:

يمكن تعريف المنظار على أنه جدول افتراضي ناتج عن استعلام، أي أنه تعليمة select مخزنة في قاعدة البيانات. تعليمة إنشاء المنظار هي على الشكل:

```
Create view اسم المنظار as
Select.....
```

يمكن تضمين المنظار في تعليمة الاستعراض Select كأي جدول آخر كما يمكن استخدامه ضمن شروط مع باقي تعليمات DML أي update, delete, insert ولكن ضمن شروط معينة.

مثال على ذلك في حساب scott لنقم بإنشاء المنظار vemp كالتالي:

```
Create or replace view vemp as select empno as id, ename, mgr, sal from emp;
```

اختبر التعليمات التالية:

من هذا المثال نستنتج ما يلي:

```
-- إدراج سجل جديد في جدول vemp
INSERT INTO vemp VALUES (1020, 'IYAD', 7902, 2500);
-- تحديث الراتب للموظف الذي يحمل المعرف 1020
UPDATE vemp
SET sal = 1500 WHERE id = 1020;
SELECT *
FROM vemp WHERE id = 1020;
-- حذف السجل للموظف الذي يحمل المعرف 1020
DELETE FROM vemp WHERE id = 1020;
```

في نظم أخرى نحتاج لوضع أسماء العواميد عند الإضافة: (id,name, ...)

A. تم استبعاد بعض الحقول من جدول emp أي تم تبسيط استخدام الجدول بالمنظار vemp.
B. تم تغيير اسم الحقل empno إلى id.
C. تمكنا من استخدام كافة تعليمات معالجة المعطيات بالحالة العامة لا يمكن بهذه البساطة).

يمكن الاستفادة من المنظار بتوزيع الصلاحيات والسماحيات، سنشرح هذا عن طريق المثال التالي: قم بتنفيذ التعليمات التالية:

```
-- إنشاء العرض vemp30
CREATE VIEW vemp30 AS
SELECT
    empno id,
    ename fullname,
    sal salary,
    hiredate,
    deptno department
FROM emp WHERE deptno = 30;

-- منح الصلاحيات للمستخدم devuser
GRANT SELECT, UPDATE, INSERT, DELETE ON vemp30 TO devuser;
```

تعليمة grant تعطي الصلاحيات الأربعة على المنظار vemp30 إلى الحساب devuser. ثم نقوم باستخدام الحساب devuser لاختبار التعليمات:

```
Conn devuser;
-- تحديد جميع الأعمدة من جدول scott.emp
SELECT * FROM scott.emp;

-- إدراج سجل جديد في جدول SCOTT.VEMP30
INSERT INTO SCOTT.VEMP30 VALUES (1030, 'mazen', 1400, TO_DATE('20-01-1985',
'dd-MM-yyyy'), 30);

-- تحديد جميع الأعمدة من جدول SCOTT.VEMP30
SELECT * FROM SCOTT.VEMP30;

-- تحديث القسم للموظف الذي يحمل المعرف 1030
UPDATE SCOTT.VEMP30
SET department = 20
WHERE id = 1030;

-- تحديد جميع الأعمدة للموظف الذي يحمل المعرف 1030
SELECT * FROM SCOTT.VEMP30 WHERE id = 1030;

-- إدراج سجل جديد في جدول SCOTT.VEMP30
INSERT INTO SCOTT.VEMP30 VALUES (1040, 'kinda', 1400, TO_DATE('01-01-1972',
'dd-MM-yyyy'), 20);
```

لاحظ أنه ضمن الحساب devuser يجب علينا كتابة اسم المالك ثم النقطة ثم اسم الجدول scott. vemp30 وذلك لأن مالك المنظار (أي الحساب الذي قام بإنشائه هو الحساب scott وليس devuser).

مما سبق نجد أن الحساب scott أعطى كافة الصلاحيات للحساب devuser لاستخدام المنظار vemp30 لإدارة موظفي القسم 30 والنتيجة هي:

- من التعليمة 1 لا يمكن للمستثمر devuser قراءة الجدول emp
- من التعليمة 2 يمكنه إدخال موظف جديد على المنظار vemp30 وبالتالي على الجدول emp
- من التعليمة 3 يمكنه استعراض بعض من الحقول لموظفي القسم 30
- من التعليمة 4 يمكنه نقل أحد موظفي القسم 30 إلى القسم 20
- من التعليمة 5 لم يعد بإمكانه بعد نقل الموظف 1030 التغيير أو التعديل عليه بسبب أن السجل أصبح خارج نطاق القسم 30
- من التعليمة 6 يمكنه إدخال موظف جديد على قسم غير القسم 30

مما سبق يبدو الأمر أكثر وضوحاً للاستفادة من المناظير من حيث إدارة الصلاحيات باستثناء الحالة في التعليمة رقم 6 ولكن يمكن تصحيح ذلك بالدخول مجدداً إلى حساب scott وتنفيذ التعليمة التالية:

```
CREATE OR REPLACE VIEW vemp30 AS
SELECT
    empno id,
    ename fullname,
    sal salary,
    hiredate,
    deptno department
FROM emp WHERE deptno = 30
WITH CHECK OPTION;
```

لاحظ أنه بدلاً من حذف المنظار قمنا بإضافة or replace أي استبدال بنية المنظار وبذلك لم نعد بحاجة إعادة كتابة سطر الصلاحيات، أما لو استخدمنا drop view vemp30 فذلك يلزمنا إعادة كتابة التعليمة التي تعطي الصلاحيات اللازمة. نقوم بالدخول مجدداً إلى حساب devuser واختبار التعليمة رقم 6 نجد أنها باءت بالفشل، لاحظ أيضاً باختبار التعليمة رقم 4 (أي تعديل رقم القسم نجد أنها ستفشل أيضاً وذلك لأننا أضفنا with check option التي تمنع الإدخال والتعديل الذي يؤدي إلى خروج السجل عن نطاق المنظار.

سيتم عمل ملف كامل عن الأمثلة غير المحلولة مع شرحها وامثلة المحاضرات.