المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 1

مقدمة عامة

أكواد Pl/Sql:

1. تعليمات (DCL):

هي التعليمات التحكم بالنظام و الحسابات

DCL: Data Control Language

2. البرمجة بلغة pl/sql:

PL: Procedure Language

هي كتابة الاكواد ضمن قاعده البيانات و تتضمن كتابة:

1. Functions
2. Packages
3. Triggers يتم تفعيله عند (حدث ما)

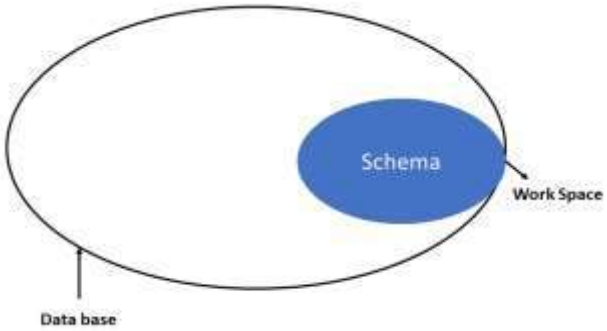
كتابة الاكواد ضمن قاعده البيانات (DB) أفضل من كتابتها ضمن Forms و هذا ما توفره pl/sql

Forms

----
views

عند بناء ال user:

تلقائيا يبنى له بيئة عمل (Work space) تسمى Schema و لها نفس اسم user



Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

خصائص يحددها Admin عند إنشاء user:

1. **المساحة Gouta:** يحجز مكان عند انشاء DB تكون فارغة و تأخذ مساحة 1G بشكل افتراضي

   ثم يمكننا تحديد مساحة لكل user بإحدث طرق التالية:

   1. قيم محددة، مثال: M2
   2. نسبة مئوية، مثال: 3٪
   3. قيمه مطلقة، أي تكبر بشكل أوتوماتيكي

2. **خصائص الحساب Profile:** القيود التي يمكننا فرضها على حساب المستخدم:

   مثال:

   1. درجه تعقيد كلمه السر (ثمانية أحرف على الأقل).
   2. صلاحية كلمه السر (كل شهر يجب تغييرها).
   3. صلاحيه الحساب (حساب مفعل لمده شهرين).
   4. عدد Sessions المسموحة (يمكن المستخدم فتح الجلسة من حاسب واحد فقط).

3. **مكان التخزين Table Space:** Admin (فقط) يستطيع تحديد أمكنة التخزين (يوزع ال Table Space)

   مثال:

   عند استخدام التعليمة Create table تعلمنا فرض قيود على الجدول لكن لا نستطيع تحديد مكان تخزين الجدول المنشأ فهي مهمة Admin فقط.

   يوجد ما يدعى بمساحات تخزين المؤقت (Temporary Table Space)

   مثال عليها: recycle bin

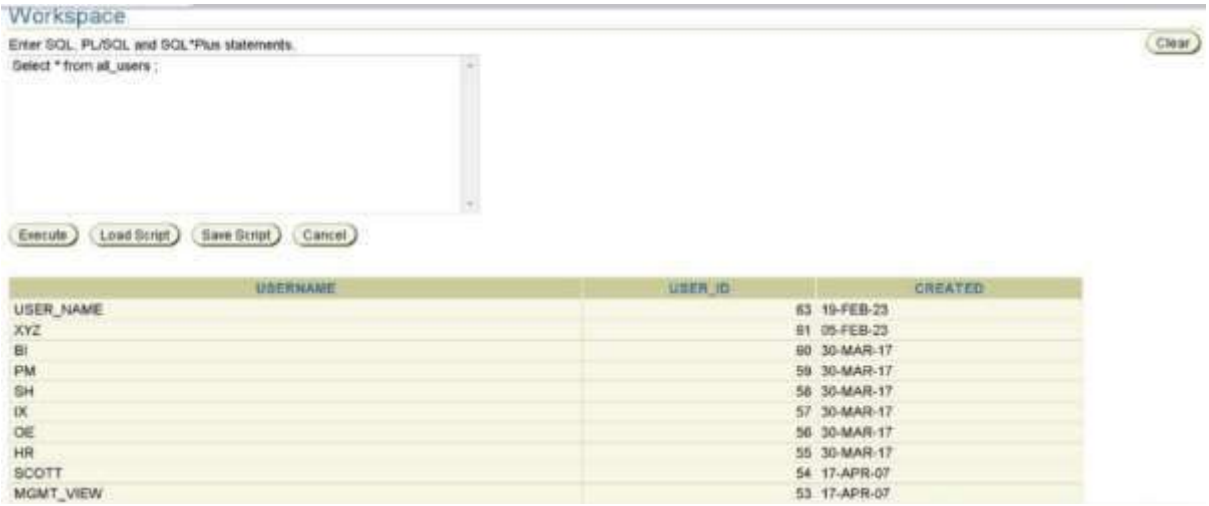
4. **فعالية الحساب lock/unlock:** عند إنشاء حساب user فانه يكون غير فعال بشكل افتراضي (lock) ثم يمكن لل Admin تغييره إلى حساب فعال unlock

> عند إنشاء الحساب فانه ممنوع من كل شيء ثم يمكن لل Admin أن يعطيه اذن اي شيء

Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

عرض جميع المستخدمين:

```
Select * from all_users;
```



USERNAME	USER_ID	CREATED
USER_NAME	63	19-FEB-23
XYZ	61	05-FEB-23
BI	60	30-MAR-17
PM	59	30-MAR-17
SH	58	30-MAR-17
IX	57	30-MAR-17
OE	56	30-MAR-17
HR	55	30-MAR-17
SCOTT	54	17-APR-07
MGMT_VIEW	53	17-APR-07

عرض جميع خصائص المستخدم المفتوح في الجلسة حالاً:

```
Select * from user_users;
```

USERNAME	USER_ID	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE	CREATED	INITIAL_RSRC_CONSUMER_GRO
SCOTT	54	OPEN			USERS	TEMP	17-APR-07	DEFAULT_CONSUMER_GROUP

ملاحظة:

1. كلمة السر بعد تعيينها لا يمكن ان تظهر عند user و لا عند Admin ابداً.
2. يمكن للمشرف تعيين كلمه سر جديدة لحساب user لكنه لا يمكن رؤية كلمة السر القديمة، المالك و Admin لهما الاسم Sys و dbs و لهما جميع السماحيات الحساب system له سماحيات أقل من الحسابين السابقين.

إنشاء user:

```
Create user user_name identified by password;
```

user_name: يجب أن يون فريداً unique

password: غير حساسة لحركه الاحرف الا في حال وضعها ضمن "_"

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

ملاحظة: أي من التعليمات السابقة التي تعلمناها فان التسميات فيها غير حساسة لحالة الأحرف، إلا في حال وضعها ضمن "_"

مثال:

Create Table Emp; ⟶ Case Insensitive

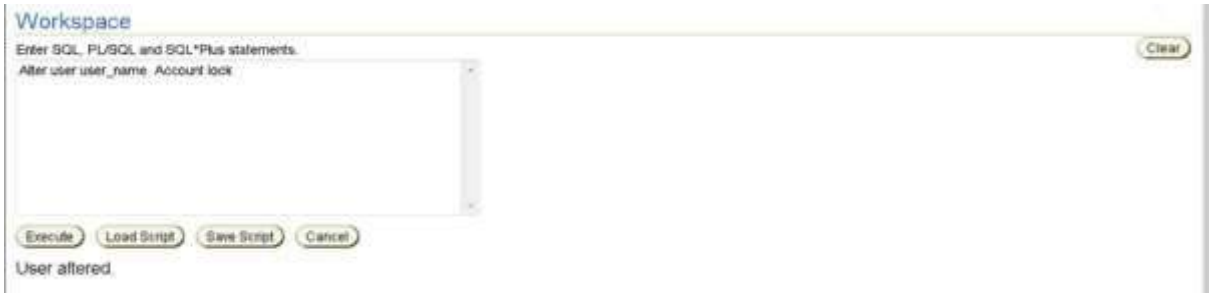
Create Table "Emp"; ⟶ Case Sensitive

تغيير كلمة السر:

Alter user user_name identified by newpassword;

تغيير فعالية الحساب:

Alter user user_name Account lock;



Account lock; ⟶ حساب مغلق (غير فعال)

Account unlock; ⟶ حساب مفتوح (فعال)

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

حذف الحساب:

```
Drop user user_name ;
```

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 2

Privileges

السماحيات

للسماحيات نوعين:

1. سماحيات على System:

هي السماحية المطلوبة لإنشاء كائن (Sequence,table....)

مثال: Create table , Create sequence..

لتنفيذ التعليمتين السابقتين نحن بحاجة لسماحية من نوع System

2. سماحيات على Object:

هي السماحية المطلوبة لإجراء أي عملية على كائن (Select,Alter,Delete....)

مثال: Select * from emp , Alter sequence...

لتنفيذ التعليمتين السابقتين نحن بحاجة لسماحية من نوع Object

> السماحية بشكل عام يمنحها مدير النظام للمستخدمين

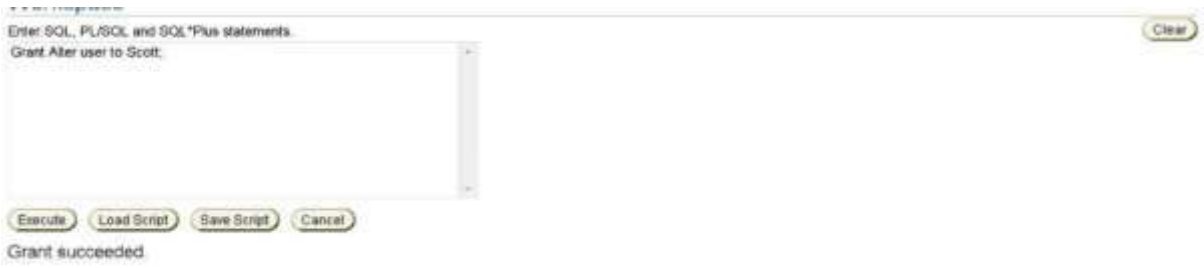
1. تعليمة إعطاء سماحية على مستوى System:

Grant sys, priv / Role ____ to user / Public / Role ____ with Admin option;

with Admin option: عند كتابتها بعد تعليمة منح السماحية فإن المستخدم الممنوح يصبح Admin على هذه السماحية

منح المستخدم Scott إمكانية تعديل المستخدمين:

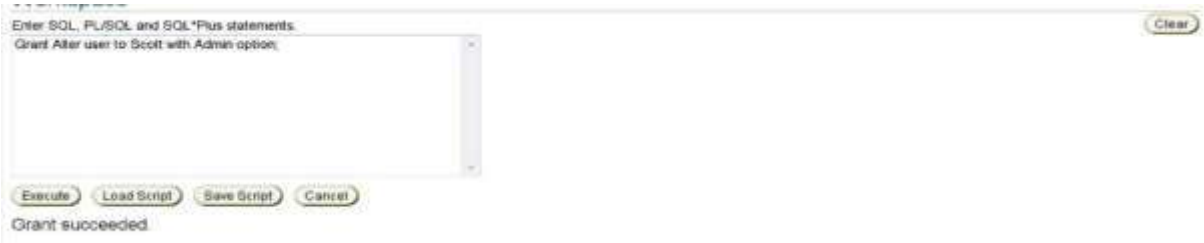
```
Grant Alter user to Scott;
```



Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

منح المستخدم Scott إمكانية تعديل المستخدمين مع إمكانية هذا المستخدم منح أو سحب هذه السماحية لمستخدمين آخرين:

```
Grant Alter user to Scott with Admin option;
```



إعطاء سماحية معينة لمستخدم معين:

```
Grant Sys_Priv user to user;
```

مثال:

إعطاء صلاحية تعديل المستخدمين للمستخدم user12:

```
Grant Alter user to user12;
```

إعطاء سماحية معينة Public:

```
Grant Sys_Priv user to Public;
```

مثال:

إعطاء صلاحية إنشاء المستخدمين لل Public:

```
Grant Create user to Public;
```

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

Public: تعني جميع المستخدمين من دون استثناء

أي في مثالنا هذا يمكن للمستخدم إنشاء أي حساب جديد

ملاحظة: عند إنشاء user جديد فإنه تلقائيا يأخذ جميع الصلاحيات المعطاة لل Public

عند إنشاء الحساب وبعد أن يأخذ السماحيات الموجودة في Public

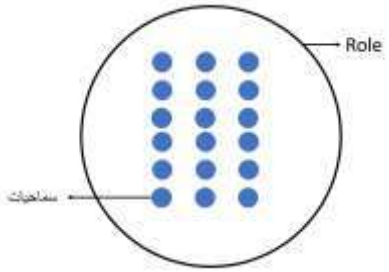
يحتاج لسماحية إنشاء جلسة: create session

يحتاج لسماحية إنشاء جدول: create emp

أما عن المكان الذي يخزن فيه البيانات (المساحة المسموح له التخزين عليها) فهي خاصية من خاصيات user (Property) إن لم أعطه هذه الخاصية لن يستطيع تخزين أي بيانات فإما أن أعطيه سماحية تسمى unlimited table space فهي تسمح للمستخدم في مساحة مطلقة غير محددة

Role:

هي أداة لتنظيم السماحيات (يمكن تشبيه بمجلد يحتوي عدة سماحيات)، كما أن إنشاءه يحتاج إلى سماحية.



ملاحظة: لتسمية ال Role نتبع نفس قواعد تسمية view أو table

إنشاء Role:

```
Create Role role_name;
```

إعطاء هذا ال Role مجموعة سماحيات:

```
Grant Sys_Priv to Role;
```

منح هذه Role إلى مستخدم معين أو إلى ال Public :

```
Grant Role to user/Public;
```

مثال:

Enass Al-kadree +963 944 078 773

Mohammad Abou Keef +963 991 893 691

إنشاء Role عطاء سماحية معينة لمستخدم معين:

```
Create Role axyz;
```

إعطاء هذا ال Role مجموعة سماحيات:

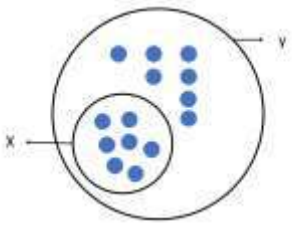
```
Grant create user,alter user to axyz;
```

منح هذه Role إلى مستخدم معين أو إلى ال Public :

```
Grant xyz to user1,user2;
```

وضع Role داخل Role:

```
Grant Role to Role;
```



مثال:

إنشاء Role باسم xml:

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

```
Create Role xml;
```

إنشاء Role باسم x2:

```
Create Role zx2;
```

وضع x2 ضمن xml (إعطاء سماحيات x2 إلى xml):

```
Grant zx2 to xml;
```

إعطاء سماحيات x2 إلى user1:

```
Grant x2 to user12;
```

إعطاء سماحيات xml إلى scott وجعلها admin لهذه ال Role:

Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

Grant xml to Scott with Admin option;

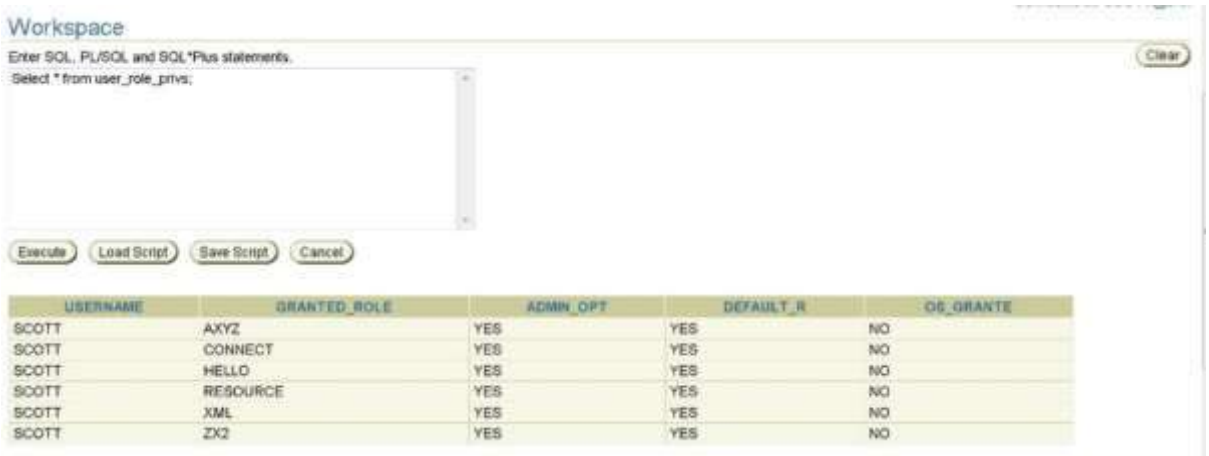
في مثالنا هذا scott يستطيع تعديل محتويات xml و لا يستطيع تعديل محتويات x2

أسماء views للاستعراض :

1. user_sys_privs:

عرض جميع السماحيات المعطاة للمستخدم الحالي إما بشكل خاص أو عن طريق Public:

Select * from user_sys_privs;



username: يعرض بها اسم المستخدم أو Public

privileges: يعرض بها اسم السماحية

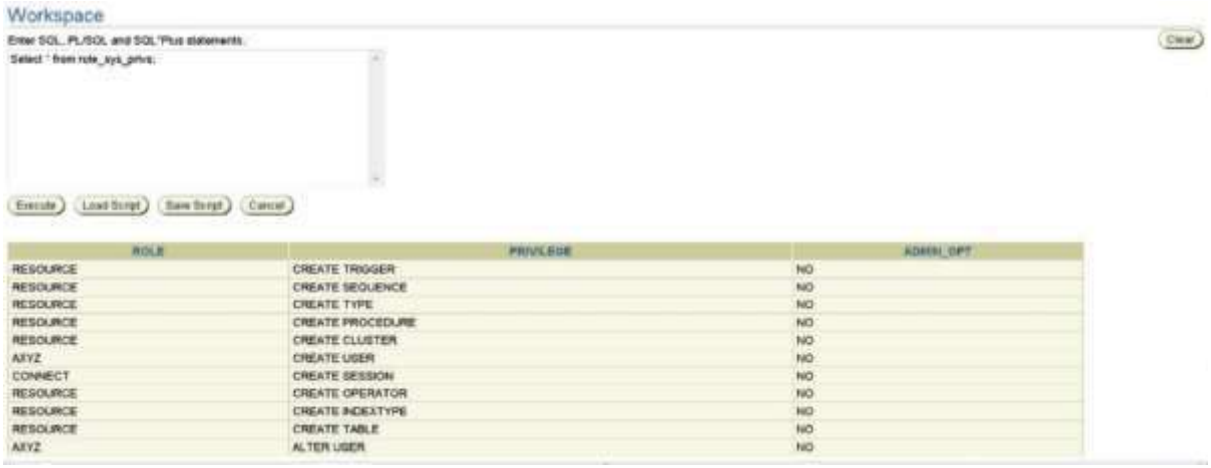
Admin option: يعرض بها أما yes أو no

2. user_role_privs:

عرض جميع ال Roles المعطاة للمستخدم الحالي:

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

Select * from user_role_privs;



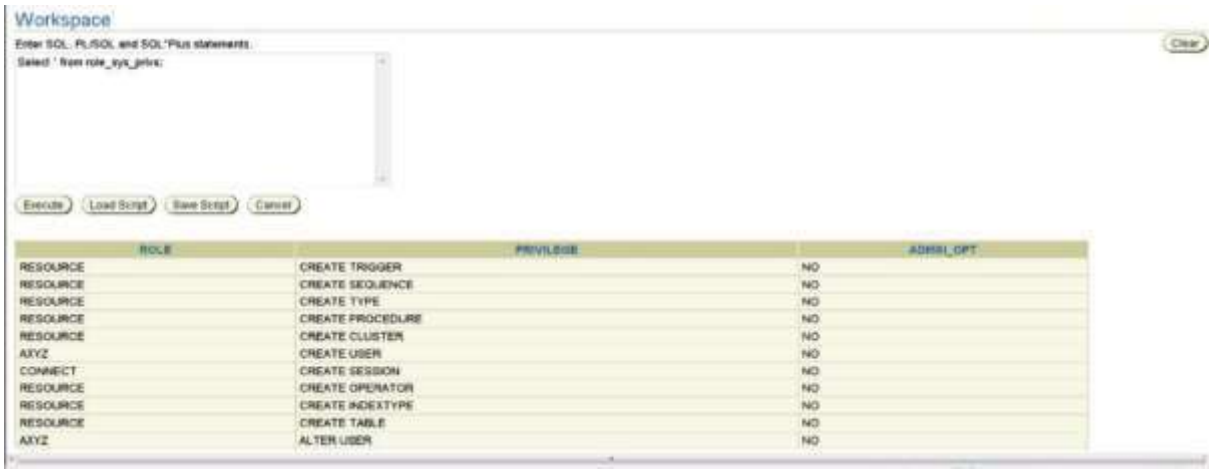
Admin option: يعرض بها أما yes أو no

yes: تعني بإمكان المستخدم التصرف في هذا ال role و تعديل محتوياته

3. role_sys_privs:

عرض سماحيات كل role (محتوياته):

Select * from role_sys_privs;



4. role_role_privs:

عرض ال Role الموجودة داخل Role أخرى:

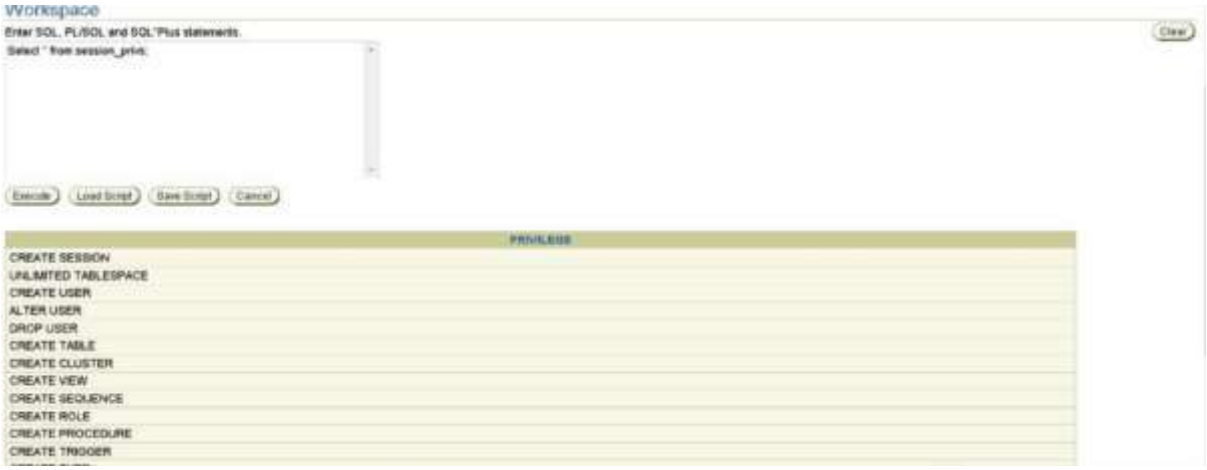
Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

Select * from role_role_privs;

5. session_privs:

عرض جميع السماحيات و ال Role و ال Role الموجودة داخل Role أخرى:

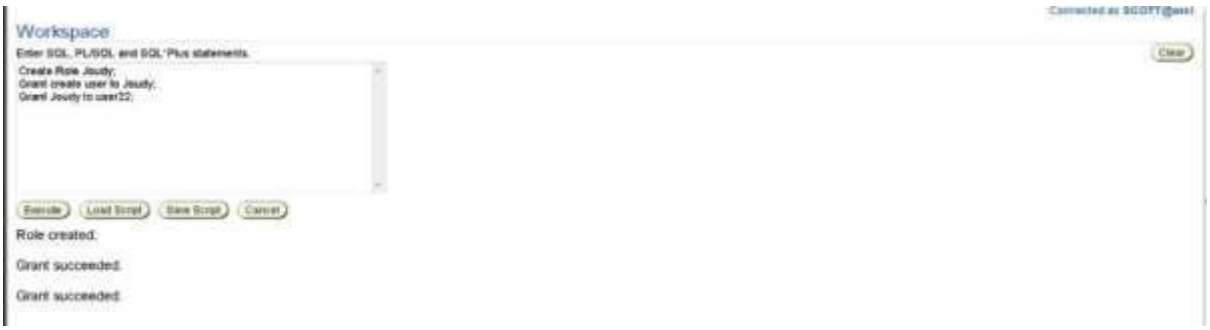
Select * from session_privs;



مثال:

إنشاء Role باسم Joudy:

```
Create Role Joudy;
grant unlimited tablespace, create user to Joudy;
grant Joudy to user22;
```



Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 3

Privileges

السماحيات

للسماحيات نوعين:

1. سماحيات على System:

هي السماحية المطلوبة لإنشاء كائن (Sequence,table....)

مثال: Create table , Create sequence..

لتنفيذ التعليمتين السابقتين نحن بحاجة لسماحية من نوع System

2. سماحيات على Object:

هي السماحية المطلوبة لإجراء أي عملية على كائن (Select,Alter,Delete....)

مثال: Select * from emp , Alter sequence...

لتنفيذ التعليمتين السابقتين نحن بحاجة لسماحية من نوع Object

> في حال أعطى الأدمن صلاحية من نوع system و هي مثلاً create table للمستخدم user5 , فقام user5 بإنشاء جدولاً باستخدام الصلاحية السابقة create table emp , لو قام الأدمن بسحب سماحية create table من user5 فإن الجدول emp يبقى موجوداً لكن لن يستطيع user5 إنشاء غيره

ملاحظة: عند إنشاء جدول (مثلا) يمكننا تنفيذ جميع العمليات عليه (نكون الأدمن)

1. تعليمة إعطاء سماحية على مستوى Object:

Grant obj, priv / all ___on schema.objName to user / Public / Role ___ with grant option;

with grant option: عند كتابتها بعد تعليمة منح السماحية فإن المستخدم الممنوح يصبح Admin على هذه السماحية

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

مثال:

إعطاء صلاحية select,update على الجدول emp (الخاص بـ scott) للمستخدم user5:

```
scott:Grant select, update on emp to user5;
```

عرض محتويات جدول emp (الخاص بـ user1):

```
user1:Select * from emp;
```

عرض محتويات جدول emp (الخاص بـ scott):

```
user1:Select * from scott.emp;
```

إعطاء صلاحية select,insert into على الجدول emp للمستخدم user5:

```
scott:Grant select,insert into on emp to user5 with grant option;
```

إضافة سجل على الجدول باستخدام الخاصية الممنوحة من scott:

```
User2:insert into scott.emp (empno) values (7020);
```

إعطاء صلاحية select على الجدول emp للمستخدم user17:

```
User2: Grant select on scott.emp to user17;
```

owner: مالك الكائن

Granter: معطي السماحية

Grantee: أخذ السماحية

من منح الصلاحية هو من يسحبها

في حال كان scott مالكاً للجدول emp و أعطى صلاحية select (مثلاً) إلى user5 و جلعه Admin عليها , فقام user5 بمنح هذه السماحية إلى user6, بعد ذلك قام scott بسحب السماحية من user5, فإنها تسحب تلقائياً من user6

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

عند سحب السماحية من user5 تسحب من user6 تلقائياً

عند يسحب scott السماحية من user5 فإن السماحية تبقى لديه بسبب أنها ممنوحة له عن طريق أخر وهو Tiger

رؤية خصائص شيء ما نستخدم (الناتج هو أعمدة هذا الجدول و الخصائص):

```
desc emp;
```

مثال:

إعطاء صلاحية select,insert into على الجدول emp للمستخدم 5و7و2:

```
Scott:Grant select,update on emp to user2,user5,user7;
```

تذكرة: منفذ التعليمة هنا هو مالك الجدول emp (owner) لذلك لم يكتب قبل اسم المالك ,أما لو لم يكن مالك الجدول فعليه أن يسبق اسم الجدول باسم المالك مثال:

scott.emp

إعطاء صلاحية select,updateعلى الجدول emp للمستخدم user8:

```
Grant select, update on emp to user8 with grant option;
```

أسماء views للاستعراض:

1. user_tab_privs:

عرض جميع السماحيات المعطاة للمستخدم الحالي إما بشكل خاص أو عن طريق Public:

```
Select * from user_sys_privs;
```

grantee: اسم المستخدم الذي آخذ السماحية

owner: اسم المستخدم المالك للجدول

Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

table-name: اسم الجدول الذي منحت السماحية منه

granter: اسم المستخدم الذي أعطى السماحية

privilege: اسم السماحية الممنوحة

grant option: يعرض بها أما yes إذا كان أدمن على السماحية أو no إذا لم يكن أدمن على السماحية

لاستعراض السماحيات المعطاة مرتبة لكل user:

```
Select * from user_sys_privs order by 1;
```

مثال:

تم إعطاء سماحيات عدة ل user5 على الجدول dept مع جعله ادمن عليها (with grant option) من قبل user12 (المالك) و الآن:

استفاد من السماحية المعطاة له و قام بالإضافة على الجدول:

```
Update user12.dept set dname=lower(dname);
```

استفاد من السماحية المعطاة له و قام بالإضافة حقل على الجدول و سماه ddate و نوعه date:

```
Alter table user12.dept add ddate date;
```

استفاد من السماحية المعطاة له و قام بحذف العامود ddate:

```
Alter table user12.dept add ddate date;
```

استفاد من السماحية المعطاة له و قام باعطاءه لغيره:

```
Grant update on user12.dept to scott;
```

2. user_tab_privs_made:

عرض السماحيات التي قمت ب grant لها:

```
Select * from user_tab_privs_made;
```

الناتج لا يحتوي حقل owner و السبب أن منفذ التعليمة هو ال owner

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

3. user_tab_privs_recd:

عرض السماحيات التي الممنوحة للمستخدم:

```
Select * from user_tab_privs_recd;
```

الناتج لا يحتوي حقل grantee و السبب أن منفذ التعليمة هو ال grantee

4. role_tab_privs:

استعرض سماحيات ال object المعطاة لها role:

```
Select * from role_tab_privs;
```

Grantee: اسم المستخدم الذي آخذ السماحية

role: اسم ال role

owner: اسم مالك ال role

table-name: اسم table صاحب السماحية

column: اسم الحقل المسموحة إليه السماحية

منح هذه Role إلى مستخدم معين أو إلى ال Public:

```
Create role myrole ;
grant select on emp to myrole;
```

تذكرة: لاستعراض السماحيات المعطاة لل role الحالي: user_role_privs

منح جميع السماحيات على جدول معين ل role:

```
grant all on dept to myrole;
grant myrole to public;
```

يمكن منح الرول و من ثم تعبئته

ملاحظة:

1) يوجد سماحيات لا يمكن منها عن طريق role و هي:

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

1. References

2. Index

3. unlimited table space

2) يوجد سماحيات يمكن تجزئتها:

1. update

2. Insert into

مثال:

إعطاء صلاحية update على الجدول emp على الحقلين sal,comm للمستخدم scott:

User17:Grant update(sal,comm) on emp to scott;

يستطيع scott الآن تنفيذ update على هذين الحقلين فقط.

أما بقية السماحيات لا يمكن تجزئتها فإن أردت تجزئة سماحية ال select فلا يمكنني إنشاء view للحقول المرادة و إعطاء ال user سماحية select على ال view

Enass Al-kadree +963 944 078 773

Mohammad Abou Keef +963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 4

Cursor

لل cursor نوعين:

1. implicit cursor:

شرطها: إعادة سجل واحد حصراً (في حال لم تعيد أي سجل أو أعادت أكثر من سجل واحد ← تعطي error)

ميزتها: أنها لا تستهلك شيئاً من موارد النظام أو الذاكرة

2. Exceplicit cursor:

شرطها:

1. أي حقل محسوب يأخذ alias
2. لا أستطيع إعادة حقلين بنفس الاسم

حلقة For:

- تدور بعدد سجلات ال cursor
- متحول العداد فيها له صفات ال cursor التي سيدور عليها.
- متحول العداد الخاص بالحلقة:
  - لا يمكن التعديل على قيمته (read only)
  - لا يتم التعرف عليه بعد انتهاء الحلقة

مثال:

```
Declare
        Cursor X is select * from emp;
Begin
        For each-Rec in X loop
                …
        End loop;
End;
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

مثال:

```
Declare
        X number;
Begin
        Select sal into X from emp where empno =7844;
End;
/
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

Procedure

طريقة تعريف ال procedure:

```
Create or replace procedure Procedure-Name (parameter list) is
{   تصريح عن متحولات...
Begin
{   تعليمات برمجية...
End procedure-Name;
/
```

or replace: (خيارية) نكتب عندما أكون قد يكون الاسم مستخدم سابقاً في اللغة و نريد استبداله

procedureName: لا يمكنني تسمية الاجرائية باسم اجرائيات اخرى موجودة سابقاً في اللغة.

Parameter list: الاجرائية قد تأخذ عدة بارمترات و قد لا تأخذ أي بارمتر

أمثلة عن ال function:

Sysdate: تابع لا يأخذ بارمترات

Nvl(,): تابع يأخذ بارمترين

Instr(, , ,): تابع يأخذ 4 بارمترات:

أمثلة عن ال procedure:

Put_line(): اجرائية تأخذ بارمتر واحد

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

اجرائية تقوم بطباعة أسماء جميع الموظفين:

```
Create procedure print_emp is
        Cursor C is select * from emp ;
Begin
        For S in C loop
                Dbms_output.put_line(S.ename);
        End loop;
End print_emp;
/
```

لاستدعاء الاجرائية السابقة:

```
Declare
Begin
        Print_emp;
End;
/
```

جميع هذه ال function تعيد قيم بأحرف كبيرة

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

Function

طريقة تعريف ال function:

```
Create or replace function Function-Name (parameter list) return Data_Type
is
{   تصريح عن متحولات...
Begin
{   تعليمات برمجية...
End Function-Name;
/
```

→ يجب أن يعيد قيمة وتكون القيمة المعادة من نفس النوع المعرف في الترويسة

يجب أن يحوي آخر سطر يخرج منه البرنامج على return لقيمة من نوع ارجاع ال function

مثال:

```
Create or replace function Function_Name (parameter list) return Data_Type is
Begin
        If ____
        return _____;
        End if;
        For  ____
        _____ ;
        End loop;
return _____;
 End Function_Name;
```


Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

💡 ليس شرطاً وجود return في نهاية التابع فقط ، بل يجب وجودها في كل مكان يُحتمل فيه انهاء التنفيذ

تمرين:

تابع يقوم بحساب عدد الموظفين الذين تبدأ أسماؤهم بالحرف S

```
Create or replace function get_cnt return number is
        Xcnt number ;
Begin
        Select count(*) into Xcnt from emp where ename like 'S%';
        Return Xcnt;
 End get_cnt;
```

استدعائه:

```
Declare
        X number;
Begin
        X :=get_cnt ;
End;
```

ملاحظة:

في لغة Oracle اذا لم يكن لل procedure او ال function بارمترات لا نضع اقواسا عند الاستدعاء و لا عند الإنشاء

مثال:

❌ Sysdate()

✔ Sysdate

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

أسماء Views للاستعراض:

1. user_source:

لعرض جميع الاجرائيات و التوابع التي قمت بإنشائها:

```
Select * from user_source;
```

لعرض جميع الكائنات (Objects) الخاصة بي سواء جدول أو تابع أو. ...:

```
Select * from user_objects;
```

طباعة مجموع الرواتب مع العمولة للموظف صاحب الرقم 7844:

```
Set serveroutput on;
Declare
        X number;
Begin
        Select sal+ nvl(comm,0) into X from emp where empno = 7844;
        Dbms_output.put_line(X);
End;
```

و في حال كان رقم الموظف السابق غير موجود سيعطي error

سنعدل على الكود السابق باضافة شروط و معالجة exception

```
Set serveroutput on;
Declare
        X number;
Begin
        Select sal+ nvl(comm,0) into X from emp where empno = 7844
        and sal < 2000;
        Dbms_output.put_line(X);
```


Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

```
Exception
When no_data_found then
        Dbms_output.put_line('there is no data');
When too_many_rows then
        Dbms_output.put_line('there is too many rows');
When others then
        Dbms_output.put_line('other error ');
End;
```

- في حال لم يجد موظف يحقق الشروط يطبع there is no data
- في حال أعاد أكثر من موظف محقق للشروط there is too many rows
- في حال وجود خطأ آخر يطبع other error

تمرين هام:

```
Declare
        X number ;
Begin
        Selecte count(*) into X from emp where deptno = 30;
        Dbms_output.put_line('The count employees of dept 30 is ' || X);
Exception
When no_data_found then
        Dbms_output.put_line('there is no data');
When too_many_rows then
        Dbms_output.put_line(' there is too many rows ');
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

```
When others then
        Dbms_output.put_line('other error ');
End;
```

نذكر أن التوابع الاجمالية حتما تعيد قيمة

Count() يعيد قيمة أو 0

باقي التوابع تعيد قيمة أو null

هذا يعني في حال عدم وجود القسم 30 لن يعطي error

لكن في حال جربت اضافة group by لتعليمة ال select السابقة عندها احتمال ان تعطي التوابع الاجمالية error

- توابع اجمالية لا تعيد error
- توابع اجمالية + group by يمكن أن يعطي error

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 5

Procedure & Function

تمرين: قم ببناء function يمرر له رقم القسم و يعيد أحد القيم التالية:

1. إذا كان القسم غير موجود ← null
2. إذا كان القسم فارغ (لا يحوي موظفين) ← 0
3. إذا كان القسم موجود و يحوي موظفين ← عدد الموظفين في القسم

```
Create or replace function print_cnt (dno in number) Return number is
        Cnt number:=0;
Begin
        Select count(*) into Cnt from emp where dno=deptno
        group by deptno;
        Return Cnt;
        Begin
                Select count(*) into Cnt from dept where dno=deptno
                group by deptno;
                Exception
                When no_data_found then
                        return null;
        End;
Exception
When no_data_found then
        return 0;
End print_cnt;
/
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

عرض ال error الموجودة في أي procedure أو function:

```
Show error;
```

ملاحظة:

دائما أنماط المعطيات في البارمترات هي أنماط عامة

مثال: ..... char, varchar2 ,number

أي لا نحدد له size محدد

وظيفة:

تحويل ال function الى procedure في المثال السابق يستخدم بارمتر من نوع out او inout لإعادة القيمة.

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 6

Trigger

لا يمكن بناء زنادين على نفس الحدث

أسماء views للاستعراض:

1. user_Triggers:

استعراض جميع ال Trigger:

```
Select * from user_Triggers;
```

حذف Trigger ما:

```
Drop trigger trigger_name;
```

تمرين: قم ببناء Trigger يعطى قيم تسلسلية لرقم تسلسلية لرقم الموظف (empno) عند الإدخال, بدءاً من الأعلى قيمة:

```
Create or replace Trigger tr1 before insert on emp for each row
Declare
        x number;
Begin
        Select :max(empno) into x from emp;
        :new.empno := x+1;
End tr1;
/
```

لتجربة ال trigger السابق:

```
Insert into emp (ename,sal) values ('Joudy',1500);
```

ثم للتأكد من أن ال trigger أضاف empno للسجل السابق:

```
Select * from emp;
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

مثال: على فرض Vcnt حقل ضمن جدول emp يحوي على أيام الإجازات المسموحة لكل موظف ,نريد كتابة code ل trigger يحرض عند إضافة إجازة جديدة للجدول vac هدفه أن يقوم بإنقاص ال Vcnt وفقاً للإجازات المضافة للموظف المطلوب و في حال كان عد الأيام المتبقية المسموحة غير كافي للإجازة المضافة يقوم بإلغاء العملية:1.

```
Alter table emp add Vcnt add number(3);
```

2.يعطي هذه الحقل قيمة 20 لكل موظف:

```
Update emp set Vcnt =20;
```

3.نكتب كود ال trigger المطلوب:

```
Create or replace Trigger tr2 after insert on vac for each row
Declare
        x number;
Begin
        Select Vcnt into x from emp where empno:=new.empno;
        if( x - (:new. Edate - :new.Sdate)) <0 then
              Raise_appliction_error(-20100,'not allowed');
        Else
             Update emp set
             Vnct= Vnct  - ( :new.Edate - :new. Sdate)
             Whare : empno = : new.empno;
         end if;
End tr2;
/
```


Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

لتجريب ال trigger:

```
Insert into vac (empno,sdate,edate) values (7935,sysdate,sysdate+3);
```

للتأكد من صحة النتيجة:

```
Select * from emp where empno =7935;
```

تحريض ال trigger عند أكثر من event:

يمكن بناء زنادين على نفس الحدث ثم تخصيص التعليمات لكل حدث يمكن استخدام flags التالية:

1. **Inserting:**

تصبح قيمة true فقط عندما يتحرض ل trigger بسبب Insert

2. **Updating:**

تصبح قيمة true فقط عندما يتحرض ل trigger بسبب Update

3. **Deleting:**

تصبح قيمة true فقط عندما يتحرض ل trigger بسبب Delete

> واحد فقط من هذه ال flags الثلاثة يأخذ true إذ لا يمكن تنفيذ أكثر من عملية في عملية في وقت واحد

مثال: كتابة كود ال trigger يمنع حذف أي سجلات و يمنع تعديل المعاشات نقصاناً:

```
Alter table emp add Vcnt add number(3);
```

2.يعطي هذه الحقل قيمة 20 لكل موظف:

```
Update emp set Vcnt =20;
```

3.نكتب كود ال trigger المطلوب:

```
Create or replace Trigger tr3 after update for each row
Declare
```


Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

```
Begin
   if( x - (:new. Edate - :new.Sdate)) <0 then
      Raise_appliction_error(____,'_____');
   Elseif Updating then
       if :new. sal < :old.sal then
          Raise_appliction_error(____,'_____');
       end if;
   end if;
End tr3;
/
```

تخصيص flags للتعليمة update:

يمكن تجزيء update ل flags أصغر و محددة لكل column على حدة.

للتوضيح:

```
Update emp set ename = upper (ename);
```

حسب التعليمة السابقة:

False ← updating('sal')

True ← updating('ename)

تعريف trigger يتحرض عند تعديل sal و comm فقط:

```
Create or replace trigger tr3 before update for sal, comm …
```


Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691

المعهد التقاني للحاسوب

هندسة البرمجيات

السنة الثانية

Databases 2

قواعد معطيات 2

2023 - 2022

م. راغب شقير

عملي 7

وظيفة:

على فرض الجدول dept يحوي على الحقل count يمثل هذا الحق العدد الأعظمي لعدد الموظفين الممكن اضافتهم للجدول

و نريد تنفيذ Trigger عند الأحداث التالية:

1. Insert عند اضافة موظفين للجدول emp يتم إنقاص count من الجدول dept حسب قسم الموظف المضاف.
2. Update في حال قمت بتعديل قسم موظف معين من جدول emp يتم التعديل على count من الجدول dept حسب التغيير .
3. Delete عند حذف سجل موظف معين من الجدول emp تتم زيادة count الخاص بقسم الموظف المحذوف

أنواع ال Trigger:

1. **Data trigger:** (insert, update, delete)
2. Object: (Create, alter, drop)

1. كتابة trigger على مستوى Object:

```
Create or replace Trigger TriggerName before|after create|alter|drop on schema|database ;
```

Scema: للمتسخدم الحالي

database: لجميع المستخدمين و بالتالي هي بحاجة لصلاحية على جميع المستخدمين تسمى هذه الصلاحية (Administe)

أسماء function مساعدة:

Ora_sysevent: تعيد اسم الحدث الذي تحرض سببه ال trigger

Ora_dict_obj_name: تعيد اسم الكائن الذي جرى عليه الحدث فتحرض بسببه ال trigger

Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

Ora_dict_obj_type: تعيد نوع الكائن الذي جرى عليه الحدث (table, view, function. …)

Ora_dict_obj_owner: تعيد اسم مالك الكائن الذي جرى عليه الحدث

جميع هذه ال function تعيد قيم بأحرف كبيرة

تمرين:

اكتب كودا ل trigger يقوم بالتالي:

1) منع حذف أي جدول أو view
2) منع تعديل الجدول emp
3) يسمح بإنشاء جداول بعد الساعة الثانية ظهرا
4) يمنع انشاء أي procedure بين الساعة 8 و 12 صباحاً

الحل:

```
Create or replace Trigger Tr1 Before create or alter or drop on schema

Declare

Begin

        If Ora_sys_event = 'DROP' then

                If Ora_dict_obj_type = 'VIEW' or Ora_dict_obj_type='TABLE' then

                        Raised_Application_Error(-20100,'can't delete');

                End if;

        Elsif Ora_sys_event = 'ALTER' then

                If Ora_dict_obj_name= 'EMP' then

                        Raised_Application_Error(-20100,'can't edit);
```


Enass Al-kadree +963 944 078 773
Mohammad Abou Keef +963 991 893 691

```
            End if;
        Elsif Ora_sys_event = 'CREATE' then
            If Ora_dict_obj_type='TABLE' and to_char(sysdate,hh24) < 14 then
                Raised_Application_Error(-20100,'can't create now);
            Elsif Ora_dict_obj_type='PROCEDURE' and to_char(sysdate,hh24)
            not between 8 and 12 then
            Raised_Application_Error(-20100,'can't create now);
            End if ;
        End if;
End Tr1;
```

تمرين :

اكتب كودا ل function لحساب ضريبة (ضريبة شرائح)

بحيث يأخذ رقم الموظف كبارمتر يعيد قيمة الضريبة المستحقة على أن الضرائب كالتالي

0	1000	1٪
1001	4500	3٪
4501	10000	7٪
10001	فما فوق	9٪

Enass Al-kadree
Mohammad Abou Keef
+963 944 078 773
+963 991 893 691

```
Create or replace function Tax-Clac (eno number) return numbrr
        S number;
        Tax number;
Begin
        Select sal into S from emp where empno =eno;
        If  S <= 1000 then
                Tax:=S*1/100;
        Elsif S<= 4500 then
                Tax:=1000*1/100 + (S-1000)*3/100;
        Elsif S<= 10000 then
                Tax := 1000*1/100+3500*3/100+(S-4500)*7/100;
        Elsif S>=10000 then
                Tax:=1000*1/100+3500*3/100+5500*7/100+(S-
        10000)*9/100;
        End if;
Return Tax;
End Tax-Clac;
```

وظيفة:

اكتب كودا ل trigger يقوم بتحديد empno تلقائيا عند كل عملية insert لموظف جديد ، لكن في حال كان الجدول يحوي موظفين بالأرقام 1،5،12،27

فإن ال trigger يعطي أرقاما تقوم بتعبئة الأرقام الفارغة بين القيم السابقة.

Enass Al-kadree
+963 944 078 773
Mohammad Abou Keef
+963 991 893 691