

Rappels

set serveroutput on

▼ Créer des tables :

```
Drop table nomTable cascade constraints;    //On peut pas tout mettre en une ligne
Drop table nomTable2 cascade constraints;
Drop table nomTable3 cascade constraints;

create table nomtable
(
    id_nomtable number,
    blabla varchar2(50),                // ne surtout pas oublier le 2
    blabla2 char(5),
    constraints PK_nomtable_id_nomtable Primary key (id_nomtable);
    constraint UK_nomtable_blabla Unique (blabla)
)

alter table nomtable
constraints FK_nomtable_blabla2 Foreign key (blabla2) References autretable(blabla2);

alter table nomtable                //Obliger de remettre alter table
constraints CK_nomtable_blabla2 Check (blabla between 10 and 20)
```

▼ Créer séquence

```
Drop sequence nom_sequence

Create sequence nom_sequence increment by 1 start 1;

nom_sequence.nextval
```

▼ Update

```
UPDATE TABLE nom_table

set nom_colonne = ...

where condition
```

▼ Delete

```
DELETE nom_table

WHERE condition
```

▼ Les participants qui ont participé à toutes les éditions de tous les galas

```
select *

from table1
```

```

where NOT EXISTS (select 'X'
                    from table2)
where NOT EXISTS (select 'X'
                    from table3
                    where table1.id = table3.idtable1 and
                           table2.id = table3.idTable2 and condition

```

▼ Trigger

```

create or replace trigger alert
after insert or update on emp
begin
DBMS_OUTPUT.PUT_LINE('op done');
end;

create or replace trigger ouvrable
before insert or update or delete on emp
begin
  if To_char(sysdate, 'D')>5 then
    RAISE_APPLICATION_ERROR(-20001, 'WHOULA');
  END if;
end;

create or replace trigger change_job
before update of job on emp
for each row
begin
  :new.sal := :old.sal+500;
end;

```

▼ Table temporaire

```

CREATE GLOBAL TEMPORARY TABLE temporary_table3
(idgala integer, nomgala varchar2(50),
numedition integer, caparticipants FLOAT,
caaccompagnateurs FLOAT)
ON COMMIT PRESERVE ROWS;

```