

## EXERCICIS DDL SOBRE TRANSACCIONS

1.- Crea BD proves\_trans

```
CREATE DATABASE proves_trans;
```

2.- Crea les taules següents:

ALUMNES = exp + nom + d\_naix

ASSIGS = codi + nom + hores

NOTES = alumne + assig + n1 + n2 + n3 + nf

R.I: si la nota final no és nula, ha de ser major o igual que la mitja de les 3 notes.

```
CREATE TABLE alumnes(  
  exp    INTEGER PRIMARY KEY,  
  nom    VARCHAR(30),  
  d_naix DATE  
);
```

```
CREATE TABLE assigs(  
  codi   CHAR(4) PRIMARY KEY,  
  nom    VARCHAR(70),  
  hores  DECIMAL(4,2)  
);
```

```
CREATE TABLE notes(  
  alumne INTEGER  
    CONSTRAINT fk_alumne  
      REFERENCES alumnes(exp)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
  
  assig  CHAR(4)  
    CONSTRAINT fk_assig  
      REFERENCES assigs(codi)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
  
  n1     NUMERIC(4,2),  
  n2     NUMERIC(4,2),  
  n3     NUMERIC(4,2),  
  
  nf     NUMERIC(4,2)  
    CONSTRAINT val_nf  
      CHECK((nf is NULL) OR (nf>=((n1+n2+n3)/3))),  
  
  CONSTRAINT pk_notes  
    PRIMARY KEY (alumne,assig)  
);
```

3.- Insereix dades:

```
INSERT INTO alumnes VALUES(1,'Pep','04/03/75');
```

```
INSERT INTO alumnes VALUES(2,'Josep','05/06/74');
```

```
INSERT INTO assigs  
VALUES('ADDA','Anàlisi i Disseny d'Aplicacions Informàtiques',10);
```

```
INSERT INTO assigs  
VALUES('PLE','Programació en Llenguatges Estructurats',8);
```

```
INSERT INTO notes(alumne,assign,n1,n2,n3) VALUES(1,'ADDA',1,2,3);
```

```
INSERT INTO notes(alumne,assign,n1,n2,n3) VALUES(1,'PLE',4,5,6);
```

```
INSERT INTO notes(alumne,assign,n1,n2,n3) VALUES(2,'ADDA',7,8,9);
```

4.- Transacció anul·lada

4.1. Inicia una transacció.

```
BEGIN
```

4.2 Insereix la nota final de cada alumne (mitja de les 3 notes) sumant-li 1 punt

```
UPDATE notes  
SET nf=((n1+n2+n3)/3)+1;
```

4.3. Elimina els alumnes que no estiguen matriculats en totes les assignatures.

```
DELETE FROM alumnes  
WHERE EXISTS ( SELECT *  
                FROM assigs  
                WHERE codi NOT IN (SELECT assign  
                                   FROM notes  
                                   WHERE alumne=exp  
                                   )  
                );
```

4.4. Comprova els canvis fets en la sessió de la transacció i en altra sessió.

Fent açò en 2 sessions:

```
SELECT * FROM alumnes;
```

```
SELECT * FROM notes;
```

Ens adonem que en la sessió “de fora” no hi ha canvis fets.

4.5. Desfes els canvis fets

```
ROLLBACK
```

4.6. Comprova els canvis en les 2 sessions

Repetint l'apartat 6.3, veiem que no estan els canvis que s'havien fet.

5. Transacció amb final feliç: torna a fer l'exercici 6 però acceptant els canvis.

Cal fer tots els passos però en compte del ROLLBACK, cal fer COMMIT.

Veiem que, després del commit, els canvis es veuen en les 2 sessions.

6. Comprova el resultat de les següents sentències, executant-les de 2 formes: amb SQL interactiu i amb SQL programàtic.

Begin

Insert into alumnes (exp) values (100);

Insert into alumnes (exp) values (100); -- ací dóna error (per la clau principal)

Insert into alumnes (exp) values (102);

Commit

#### SOLUCIONS:

6) Tant en SQL interactiu com en programàtic no es farà cap de les sentències, ja que n'hi ha una d'elles que falla.

a) SQL interactiu (anar posant línia a línia polsant enter després de cada sentència).  
El segon insert dóna error per la clau principal:

ERROR: duplicate key violates unique constraint "alumnes\_pkey"

El tercer insert dóna error perquè la transacció està avortada. Diu que qualsevol sentència serà ignorada fins el final de la transacció:

ERROR: current transaction is aborted, commands ignored until end of transaction block

Encara que ara es tanque la transacció amb el COMMIT, el resultat és que no ha fet cap dels inserts.

b) SQL programàtic (posant les 5 línies i executar-les totes juntes).  
Dóna error en el segon insert. La transacció ha finalitzat (sense fer res). Si ara intentem tancar la transacció ens donarà un avís dient que no hi ha cap transacció en progrés.

WARNING: there is no transaction in progress