



Tasca 6. Transaccions

L'objectiu d'aquesta activitat és treballar el concepte de transacció i els nivells d'aïllaments en cas de problemes amb les transaccions concurrents.

1. Un **banc** té comptes corrents. Cada compte corrent s'identifica amb un número. Als comptes corrents es poden fer moviments que s'identifiquen també per un número de moviment que dona la màquina (identity). Els moviments, a més, tenen data, concepte i import, aquest últim pot ser positiu o negatiu.
 - a. Mostra les sentències que utilitzarem per a **crear les taules** a la base de dades. Mostra també les sentències SQL per **insertar** 2 comptes bancaris amb 3 moviments cadascú.

1- CREATE TABLE ComptesCorrents (

NumeroCompte INT PRIMARY KEY,

Saldo DECIMAL(10, 2) DEFAULT 0);

2- CREATE TABLE Moviments (

NumeroMoviment INT PRIMARY KEY,

NumeroCompte INT NOT NULL,

DataMoviment DATE NOT NULL,

Concepte VARCHAR(255) NOT NULL,

Import DECIMAL(10, 2) NOT NULL,

FOREIGN	KEY	(NumeroCompte)	REFERENCES
ComptesCorrents(NumeroCompte));			

Per insertar:

Compte 1:

INSERT INTO ComptesCorrents (NumeroCompte, Saldo) VALUES (1, 0);



DAM. M02 BASES DE DADES. UF2 - Llenguatges SQL: DML i DDL

```
INSERT INTO Moviments (NumeroMoviment, NumeroCompte, DataMoviment, Concepte, Import) VALUES
```

```
(1, 1, '2024-02-06', 'Ingress 1', 500),
```

```
(2, 1, '2024-02-07', 'Sortida 1', -200),
```

```
(3, 1, '2024-02-08', 'Ingress 2', 300);
```

Compte 2:

```
INSERT INTO ComptesCorrents (NumeroCompte, Saldo) VALUES (2, 0);
```

```
INSERT INTO Moviments (NumeroMoviment, NumeroCompte, DataMoviment, Concepte, Import) VALUES
```

```
(4, 2, '2024-02-06', 'Ingress 3', 700),
```

```
(5, 2, '2024-02-07', 'Sortida 2', -400),
```

```
(6, 2, '2024-02-08', 'Ingress 4', 100);
```

- b. El banc vol un **resum del total** de tots els **ingressos** (imports positius), totes les **sortides** (imports negatius) i el **saldo**. I volen que cuadri. Per fer-ho calen tres consultes diferents. Explica si és necessari posar les tres consultes dins una transacció i quin ha de ser el nivell mínim d'isolació (ISOLATION) d'aquesta transacció. Exemple d'informe:

Total Ingressos: 1000

Total Sortides: 750

Saldo: 250

Total d'Ingressos

```
SELECT SUM(Import) AS TotalIngressos FROM Moviments WHERE Import > 0;
```

Total de Sortides

```
SELECT SUM(Import) AS TotalSortides FROM Moviments WHERE Import < 0;
```

Saldo

```
SELECT NumeroCompte, SUM(Import) AS Saldo FROM Moviments GROUP BY NumeroCompte;
```



El nivell mínim d'isolació (ISOLATION) per a aquesta transacció hauria de ser **SERIALIZABLE**, ja que assegura que cap altra transacció pugui accedir a les dades que estan sent utilitzades per la transacció actual fins que aquesta finalitzi.

- c. Fes una **transferència** de diners d'un compte a un altre. De manera que primer treus els diners d'un compte i després el poses a l'altre. Explica quin serà el nivell mínim d'isolació d'aquesta transacció per a que les altres transaccions no vegin el que està passant aquí dins.

START TRANSACTION ISOLATION LEVEL SERIALIZABLE;

Treu els diners del Compte 1

UPDATE ComptesCorrents

SET Saldo = Saldo - 100

WHERE NumeroCompte = 1;

Posa els diners al Compte 2

UPDATE ComptesCorrents

SET Saldo = Saldo + 100

WHERE NumeroCompte = 2;

COMMIT;

En aquest cas, el nivell mínim d'isolació que hauria de ser utilitzat és **SERIALIZABLE**. Aquest nivell d'isolació assegura que les transaccions s'executen de manera consecutiva, sense interrupcions ni accés concurrent a les dades.

2. Suposa les següents taules:



```
Clients ( Nif (PK), Nom )
Factures ( nFactura (PK), Data, Descripcio, Nif (FK1 a clients) )
FamiliesProductes ( nFamilia (PK), nomFamilia )
Productes ( nProducte (PK), Descripcio, PreuDeReferencia,
            nFailia (FK1 a FamiliesProductes) )
LiniesDeFactura( Quantitat, preuDeVendaUnitari,
                nProducte (PK) (FK1 a Productes),
                nFactura (PK) (FK2 a Factures) )
```

Suposem que en aquesta empresa els preus fluctuen constantment i volem assegurar-nos que les factures que es facin des del precís instant que modifiquem el preu d'un producte portaran el nou preu. Quan es fa una factura cal llegir el PreuDeReferencia de la taula 'Productes' per posar-lo a preuDeVendaUnitari de la taula 'LiniesDeFactura'.

Ho podem fer de la següent manera:

```
--inici transacció.
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL ???
--dades a insertar
SET @producte_a_comprar=12;
SET @nFactura=35;
SET @qtat=3;
|--busquem el preu
SELECT @PreuDeReferencia = PreuDeReferencia
FROM productes
WHERE nProducte = @producte_a_comprar;
--insertem la línia dins la factura
INSERT INTO LiniesDeFactura( Quantitat, preuDeVendaUnitari, nProducte, nFactura )
VALUES ( @qtat, @PreuDeReferencia, @producte_a_comprar, 3, @nFactura )
|--FI
COMMIT;
```

Llegint el codi, **argumenta quin és el primer nivell d'isolació que ens garantirà els requeriments de l'enunciat.**

Per garantir que les factures reflecteixin el nou preu de referencia immediatament després de modificar el preu d'un producte, és important utilitzar un nivell d'isolació que eviti la lectura de dades no confirmades. En aquest context, el primer nivell d'isolació que proporcionarà aquesta garantia és READ COMMITTED. Això implica que quan es llegeix el PreuDeReferencia de la taula 'Productes' per posar-lo com a



preuDeVendaUnitari a la taula 'LiniesDeFactura', es garantirà que aquesta lectura reflecteixi les dades confirmades fins a aquest moment, i no les canvis que estan pendents de confirmar.

3. Contesta les següents preguntes:

a. Quina diferència hi ha entre un motor MyISAM i un altre InnoDB ? Quin faries servir en el cas de l'exercici 2 ? Perquè?

- **MyISAM:** No suporta transaccions. No proporciona característiques de control de la concurrència.
- **InnoDB:** Ofereix suport complet per a transaccions i control de concurrència.

Utilitzaria InnoDB, ja que permet transaccions.

b. Què és un Autocommit ? Com es pot activar o desactivar?

- Autocommit és una característica en bases de dades que fa que cada sentència SQL sigui una transacció autònoma. Per activar o desactivar l'autocommit a MySQL s'utilitza:

SET autocommit = 0 | 1 o bé SET @@autocommit := 0 | 1

c. Què passa a una transacció si es fa un START TRANSACTION i encara no ha acabat l'anterior?

- Que comença una nova transacció, i acaba l'anterior.

d. Per que serveixen les instruccions SAVEPOINT i ROLLBACK TO SAVEPOINT ?

- SAVEPOINT fa una marca en la transacció de forma que al fer un ROLLBACK podem tornar enrera només fins a un cert punt.
- Un ROLLBACK desfà una transacció (torna enrera)

e. Per a que s'utilitza LOCK TABLES?

- LOCK TABLES s'utilitza per a bloquejar taules en MySQL, limitant l'accés d'altres connexions fins que es realitzi un UNLOCK TABLES.