

# Concurrencia en les transaccions

En aquesta pràctica de MySQL comprovarem com es mostren les dades des d'un usuari mentre des d'un altre usuari realitzem transaccions.

Per dur a terme la pràctica, he preparat una transacció on introduïm valors nous.

```
START TRANSACTION;
INSERT into clients (nom, cognoms, idPais, idProvincia, ciutat, email, telefon)
VALUES ("Javier", "Pedragosa Lozano", 3, 5, "Barcelona", "javipedragosa@gmail.com", "52387234"),
       ("Rafaela", "Lozano Martí", 3, 5, "Barcelona", "rafiloz@gmail.com", "592438409"),
       ("Manuel", "Pedragosa Cabrera", 3, 5, "Barcelona", "mpedragosa@gmail.com", "4358345232");
```

Durant aquesta transacció, comprovarem si aquests canvis es reflexen a la base de dades des d'un altre usuari i després desfarem els canvis amb la comanda *rollback* per poder fer la mateixa prova amb altres nivells d'aïllament. La consulta busca precisament aquells clients la ciutat dels quals és «Barcelona» i la idProvincia és 5, ja que cap dels clients preexistents a la taula Clients compleix amb aquests requisits, i els clients que inseriré s'haurien de mostrar en fer la següent consulta:

```
SELECT * FROM hotels_blanxart.clients
WHERE ciutat = "Barcelona" AND idProvincia = 5;
```

## 1. Nivell d'aïllament: *Read Uncommitted*

| # | idClient | nom     | cognoms           | idPais | idProvincia | ciutat    | email                   | telefon    |
|---|----------|---------|-------------------|--------|-------------|-----------|-------------------------|------------|
| 1 | 72       | Javier  | Pedragosa Lozano  | 3      | 5           | Barcelona | javipedragosa@gmail.com | 52387234   |
| 2 | 73       | Rafaela | Lozano Martí      | 3      | 5           | Barcelona | rafiloz@gmail.com       | 592438409  |
| 3 | 74       | Manuel  | Pedragosa Cabr... | 3      | 5           | Barcelona | mpedragosa@gmail.com    | 4358345... |
| * | NULL     | NULL    | NULL              | NULL   | NULL        | NULL      | NULL                    | NULL       |

Com podem veure i com bé descriu el seu nom, amb aquest nivell d'aïllament podem veure els canvis tal com es fan, sense fer el commit.

## 2. Nivell d'aïllament: *Read Committed*

| # | idClient | nom  | cognoms | idPais | idProvíncia | ciutat | email | telefon |
|---|----------|------|---------|--------|-------------|--------|-------|---------|
| * | NULL     | NULL | NULL    | NULL   | NULL        | NULL   | NULL  | NULL    |

Com podem veure i com bé descriu el seu nom, amb aquest nivell d'aïllament no es veuen reflexats els canvis tal com es fan, sense fer el commit. Si fem commit llavors si que surten reflexats.

| # | idClient | nom     | cognoms           | idPais | idProvíncia | ciutat    | email                   | telefon    |
|---|----------|---------|-------------------|--------|-------------|-----------|-------------------------|------------|
| 1 | 75       | Javier  | Pedragosa Lozano  | 3      | 5           | Barcelona | javipedragosa@gmail.com | 52387234   |
| 2 | 76       | Rafaela | Lozano Marti      | 3      | 5           | Barcelona | rafilo2@gmail.com       | 592438409  |
| 3 | 77       | Manuel  | Pedragosa Cabr... | 3      | 5           | Barcelona | mpedragosa@gmail.com    | 4358345... |
| * | NULL     | NULL    | NULL              | NULL   | NULL        | NULL      | NULL                    | NULL       |

## 3. Nivell d'aïllament: *Repeatable Read*

| # | idClient | nom  | cognoms | idPais | idProvíncia | ciutat | email | telefon |
|---|----------|------|---------|--------|-------------|--------|-------|---------|
| * | NULL     | NULL | NULL    | NULL   | NULL        | NULL   | NULL  | NULL    |

Amb aquest nivell d'aïllament, no es veuen els canvis reflexats abans del commit, però si que es veuen reflexats després. El seu funcionament pràctic és similar al nivell de *Read Committed*, però la diferència principal és que *Repeatable Read* només es fixa en una «versió» concreta de la base de dades que es fixa automàticament.

## 4. Nivell d'aïllament: *Serializable Read*

| # | idClient | nom  | cognoms | idPais | idProvíncia | ciutat | email | telefon |
|---|----------|------|---------|--------|-------------|--------|-------|---------|
| 1 | NULL     | NULL | NULL    | NULL   | NULL        | NULL   | NULL  | NULL    |

Amb aquest nivell d'aïllament, no es veuen els canvis reflexats abans del commit, però si que es veuen reflexats després. El seu funcionament és exactament igual que el del nivell de *Repeatable Read*, però la diferència principal és que *Serializable Read* crea un bloqueig compartit per totes les comandes *Select* depenent si s'utilitza *for share* o no.