

Bases de Dades

# Practica 2

---



Name: Laia Abad

Login: laia.abad

# Index

Introduction with a brief summary.....	2
OLAP and OLTP concepts explained.....	3
Explain how you found out the remote OLTP database structure.....	4
Relational OLAP model diagram.....	5
Explain how your Java/C program works.....	7
Explain how you created the users.....	8
Explain how your stored procedures work.....	9
Explain how your events and triggers works.....	10
Report about OLTP vs OLAP comparison.....	11
Time dedicated per part.....	16
Conclusions.....	17
References as ISO 690:2010.....	18

## Introduction with a brief summary.

En aquesta practica, hem d'accedir a una base de dades remota i importar-la utilitzant un programa Java. A partir d'aquesta base de dades, hem creat una OLTP, que es igual a la base de dades remota, i una OLAP, que té menys taules que la OLTP per tal de que fer queries sigui més ràpid i senzill.

Al importar-la amb Java a la OLTP, tenim triggers que importen automàticament la informació a la OLAP. A més, tenim un event que s'executa cada 10 minuts per a comprovar que les dades de la OLAP son correctes a partir de la OLTP.

També hem creat usuaris amb diferents privilegis per que puguin veure o canviar la informació.

Per a comparar extreure o introduir informació a la base, hem fet unes sentencies amb la OLTP i amb la OLAP.

## OLAP and OLTP concepts explained.

OLAP (On-Line Analytical Processing): Tipus de procediment que té com a objectiu agilitzar la consulta de grans quantitats de dades. S'utilitza per almacenar dades històriques. És el més ràpid per executar les sentències tipus SELECT. Son bases de dades grans que no s'haurien d'actualitzar molt sovint.

OLTP (OnLine Transaction Processing): Tipus de processament que facilita i administra aplicacions transaccionals. S'utilitza per almacenar dades actuals. És el més ràpid per executar les sentències tipus INSERT, UPDATE i DELETE. Són bases de dades mitjanes que es poden actualitzar diàriament.

## Explain how you found out the remote OLTP database structure.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| F1 |
+-----+
2 rows in set (0.00 sec)

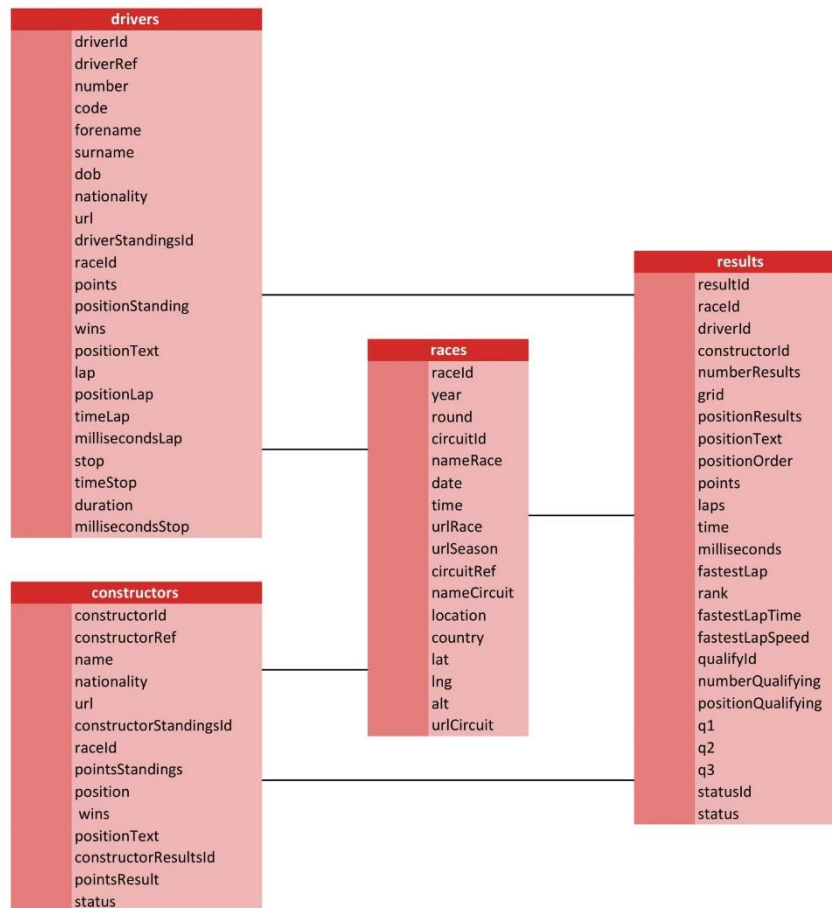
mysql> USE F1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_F1 |
+-----+
| circuits |
| constructorResults |
| constructorStandings |
| constructors |
| driverStandings |
| drivers |
| lapTimes |
| pitStops |
| qualifying |
| races |
| results |
| seasons |
| status |
+-----+
13 rows in set (0.00 sec)

mysql> DESC circuits;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| circuitId | int(11) | NO | PRI | NULL | auto_increment |
| circuitRef | varchar(255) | NO | | | |
| name | varchar(255) | NO | | | |
| location | varchar(255) | YES | | NULL | |
| country | varchar(255) | YES | | NULL | |
| lat | float | YES | | NULL | |
| lng | float | YES | | NULL | |
| alt | int(11) | YES | | NULL | |
| url | varchar(255) | NO | UNI | | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.01 sec)
```

Una vegada hem iniciat sessió a la terminal amb el nom d'usuari i contrasenya que hem trobat a l'enunciat, utilitzem la comanda `SHOW DATABASES;` per a veure el nom de la base de dades. Quan hem aconseguit el nom, per a utilitzar-la utilitzem la comanda `USE F1;` i la comanda `SHOW TABLES;` per a veure les taules. Finalment, amb la comanda `DESC nom_taula;` podem veure la estructura de la taula seleccionada.

# Relational OLAP model diagram.



He dividit la OLAP en 4 taules:

1. Results: conté les taules Results, Qualifying i Status. He decidit juntar Results i Qualifying per que les dues depenien de raceId, driverId i constructorId. La única taula que depenia de Status es Results, així que també l'he ficat aquí.
2. Races: conté les taules Races, Circuits i Seasons. Circuits i Seasons les he juntat amb Races ja que només depenen de Races.
4. Constructor: conté les taules Constructors, ConstructorResults i ConstructorStandings. Les he juntat per que totes contenen informació de Constructors.

5. Driver: conté les taules Driver, DriverStandings, LapTimes i PitStops. DriverStandings està aquí per que conté informació de Driver. LapTimes i PitStops depenen ambés de Driver i Races, igual que DriverStandings, a més he pensat que té més sentit ficar-ho a Driver, ja que els LapTimes i PitStops els fa un Driver.

## Explain how your Java/C program works.

El primer que fa es connectar-se a la base de dades remota del Puigpedros i la base de dades local del MySQL. Llavors posem la base de dades que volem utilitzar. A continuació, eliminem el que hi ha a les taules, per si hi ha hagut problemes en una importació anterior i importem les taules fila per fila a la base de dades, seleccionant totes les dades d'una taula, guardant-les del Puigpedros a una variable i insertant les dades de la variable a la taula OLTP local.

Per a poder guardar les dades de les taules, he creat un objecte per cada taula.

Per ajudar a supervisar quant s'ha importat, el nombre de fila pel que va el programa es printa per pantalla.



## Explain how you created the users.

Per l'analytic\_user, l'he concedit el permís de poder fer selects i crear i veure views.

Per el manage\_user, he concedit el permís de poder fer selects per veure les dades que van a modificar, i el permís de modificar les dades, amb el insert, update i delete, per a les dues bases de dades.

Per al rrhh\_user, he concedit el permís de creació d'usuaris que permet fer tot el que es pot fer amb usuaris (CREATE USER, DROP USER, RENAME USER, REVOKE ALL PRIVILEGES).

## Explain how your stored procedures work.

He creat un stored procedure per cada sentència que utilitzem per a comparar la taula OLTP i la OLAP, és a dir que cada comparació té 2 stored procedures.

Les dues primeres stored procedures fan un select de una query simple, una a la OLTP i una a la OLAP; les segones una query complexa, les terceres un insert... I així amb totes les sentences que hem d'utilitzar per a comparar-les.

## Explain how your events and triggers works.

Hi ha un trigger per a cada taula de la OLTP per a que quan s'ompli la taula OLTP, s'ompli també la OLAP.

Per fer això, primer he decidit quina serà la primera taula que importarem de el grup de taules OLTP que hi ha en una sola OLAP, i a partir d'aquesta, he creat els altres triggers, de tal manera que faci un update en el cas de que ja existeixi una fila amb el mateix ID, o un insert si no existeix.

Per els events, he creat un per cada taula OLAP. El que fan aquests events es cada 10 minuts, miren si el nombre de files a una taula OLAP es el correcte segons el que hi ha a la OLTP. En el cas de que sigui correcte, s'inserta una nova fila en una taula prèviament creada que conte el nom de la taula, true si es correcte, fals si es erroni i la data i hora en la que s'ha executat l'event.

# Report about OLTP vs OLAP comparison.

## 1. A simple query which involves only 1 table:

Sentencia:

OLAP:

```
SELECT DISTINCT nameCircuit  
FROM OLAP.races  
WHERE country LIKE 'Spain';
```

OLTP:

```
SELECT name  
FROM OLTP.circuits  
WHERE country LIKE 'Spain';
```

Temps de resposta:

OLAP: 0.078 sec / 0.000 sec

OLTP: 0.015 sec / 0.000 sec

Explicació:

Com podem veure, a la OLAP es triga més temps en extreure la informació, això probablement es degut a que l'OLAP té més files que contenen la mateixa informació que la OLTP, i que per tant s'ha de fer un distinct, que fa que la sentència trigui més.

## 2. A complex query which involves at least 5 tables.

Sentencia:

OLAP:

```
SELECT resultId, nameRace, nameCircuit, qualifyId,
status
FROM OLAP.Results AS re
JOIN OLAP.Races AS ra ON re.raceId = ra.raceId
ORDER BY resultId
LIMIT 1;
```

OLTP:

```
SELECT resultId, ra.name, ci.name, qualifyId, status
FROM OLTP.Results AS re
JOIN OLTP.Races AS ra ON re.raceId = ra.raceId
JOIN OLTP.Circuits AS ci ON ra.circuitId =
ci.circuitId
JOIN OLTP.Qualifying AS q ON re.raceId = q.raceId
AND q.constructorId = re.constructorId AND
q.driverId = re.driverId
JOIN OLTP.Status AS s ON s.statusId = re.statusId
ORDER BY resultId
LIMIT 1;
```

Temps de resposta:

OLAP: 3.187 sec / 0.000 sec

OLTP: 52.390 sec / 0.000 sec

Explicació:

La query a la OLTP triga molt degut a que es fa producte cartesià de 5 taules comparat amb la OLAP que fa producte cartesià de només 2.

### 3. An insert into 1 table.

Sentencia:

OLAP:

```
INSERT INTO OLAP.races (circuitId, circuitRef,  
nameCircuit, location, country, lat, lng, alt,  
urlCircuit)  
  
VALUE (197643, 'hola', 'hola', 'hola', 'hola', 2.3,  
2.3, 1, 'hola');
```

OLTP:

```
INSERT INTO OLTP.circuits  
VALUE (197643, 'hola', 'hola', 'hola', 'hola', 2.3,  
2.3, 1, 'hola');
```

Temps de resposta:

OLAP: 0.000 sec

OLTP: 0.016 sec

Explicació:

El insertar a la taula OLTP triga més ja que la OLTP te PKs i la OLAP no.

#### 4. An update into 1 field.

Sentencia:

OLAP:

```
UPDATE OLAP.races  
  
SET circuitRef = 'circuitRef', nameCircuit = 'name',  
location = 'location'  
  
WHERE urlCircuit LIKE 'hola';
```

OLTP:

```
UPDATE OLTP.circuits  
  
SET circuitRef = 'circuitRef', name = 'name',  
location = 'location'  
  
WHERE url LIKE 'hola';
```

Temps de resposta:

OLAP: 0.031 sec

OLTP: 0.015 sec

Explicació:

A la taula OLAP triga més per que té més files, per tant la condició del where triga més en trobar-se.

## 5. A delete into 1 table.

Sentencia:

OLAP:

```
DELETE
FROM OLAP.races
WHERE urlCircuit LIKE 'hola';
```

OLTP:

```
DELETE
FROM OLTP.circuits
WHERE url LIKE 'hola';
```

Temps de resposta:

OLAP: 0.047 sec

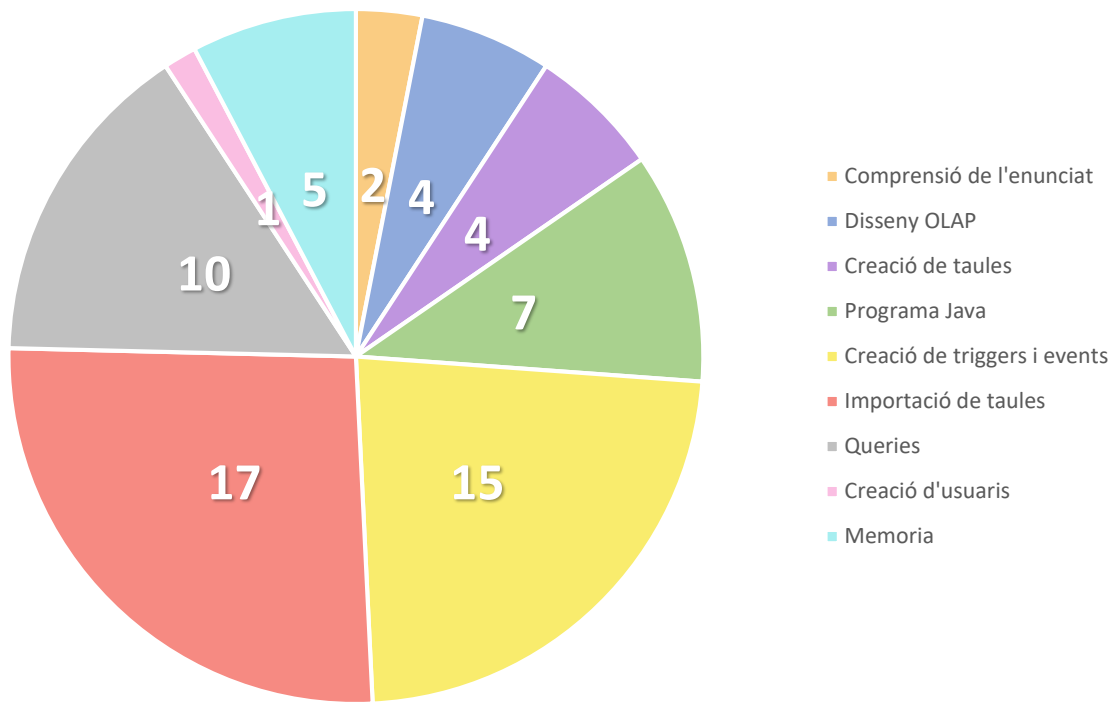
OLTP: 0.016 sec

Explicació:

A la taula OLAP triga més per que té més files, per tant la condició del where triga més en trobar-se.



## Time dedicated per part



## Conclusions.

Amb aquesta practica he après sobre les OLAPs i les OLTPs: que son, per a que serveixen, quins son els punts forts i febles de cadascuna, com es el seu disseny... Encara que al principi m'ha costat entendre que una OLAP no ha de tenir taules com las que vam crear a la primera practica, si no com les que teníem abans de repartir més la informació normalitzant les taules.

També he entès millor com es fan els triggers, els events i els stored procedures, ja que als labs només ho vam fer un parell de vegades i sabia més be conceptualitzar-les que aplicar-les. El mateix amb la creació d'usuaris, que ara he après a aplicar-ho millor.

Finalment, he après a com connectar una base de dades remota i importar-la amb un programa Java, utilitzant ODBC. He après el funcionament d'aquesta per a poder executar queries a la base de dades des de l'IntelliJ.

## References.

SORO, JORGE, 2013, ¿Cuál es la diferencia entre una base de datos OLTP y OLAP? *Jorge Soro* [online]. 2013. Available from:  
<https://jorsodom.wordpress.com/2013/09/19/cual-es-la-diferencia-entre-una-base-de-datos-oltp-y-olap/>