Pràctica 2

Bases de Dades

LSAir

Albert Luna i Pérez

albert.luna

10/07/2019

ÍNDEX

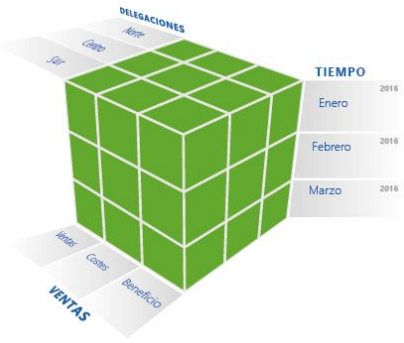
INTRODUCCIÓ

El programa consisteix en connectar-se a través d'una aplicació feta amb java a una de les tres bases de dades guardades a puigpedrós. L'usuari podrà seleccionar una d'elles i la guardarà en local. Automàticament, les dades, que estaran guardades amb un model OLTP, es guardaran també en un de model OLAP.

A més, hi haurà tres tipus d'usuari segons les seves funcions dins del control de les bases de dades locals.

OLAP, OLTP and Data Warehouse

OLAP (On-Line Analitical Procesing): És un tipus de procediment que té com a objectiu agilitzar la consulta de grans quantitats de dades. S'utilitza per almacenar dades històriques. Són bases de dades grans amb dades estàtiques. No s'hauria d'actualitzar molt sovint. Té diferents nivells. És el més ràpid per executar les sentencies tipus SELECT.



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

Data Warehouse: És una base de dades amb un disseny que facilita l'analisis de dades. Emmagatzema dades històriques i actuals. OLAP es pot utilitzar a un Data Warehouse.

OLTP PUIGPEDRÓS

1. Ens conectem al VPN de PaloAlto.
2. Obrim Simbolo del Sistema a Windows o Terminal a Mac
3. Accedim a Puigpedrós

```
Microsoft Windows [Versión 10.0.17134.765]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Uni>ssh laia.abad@puigpedros.salle.url.edu
laia.abad@puigpedros.salle.url.edu's password:
```

```
Quota de disc ocupada: 1,7M/200M

laia.abad@puigpedros:~>
```

4. Amb la comanda “mysql –u lsair_user –p” i la contrasenya lsair_bbdd accedim al servidor.

```
laia.abad@puigpedros:~>mysql -u lsair_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6931
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

5. Fem “show databases;” i “use <nom_database>” per utilitzar una de les bases de dades.

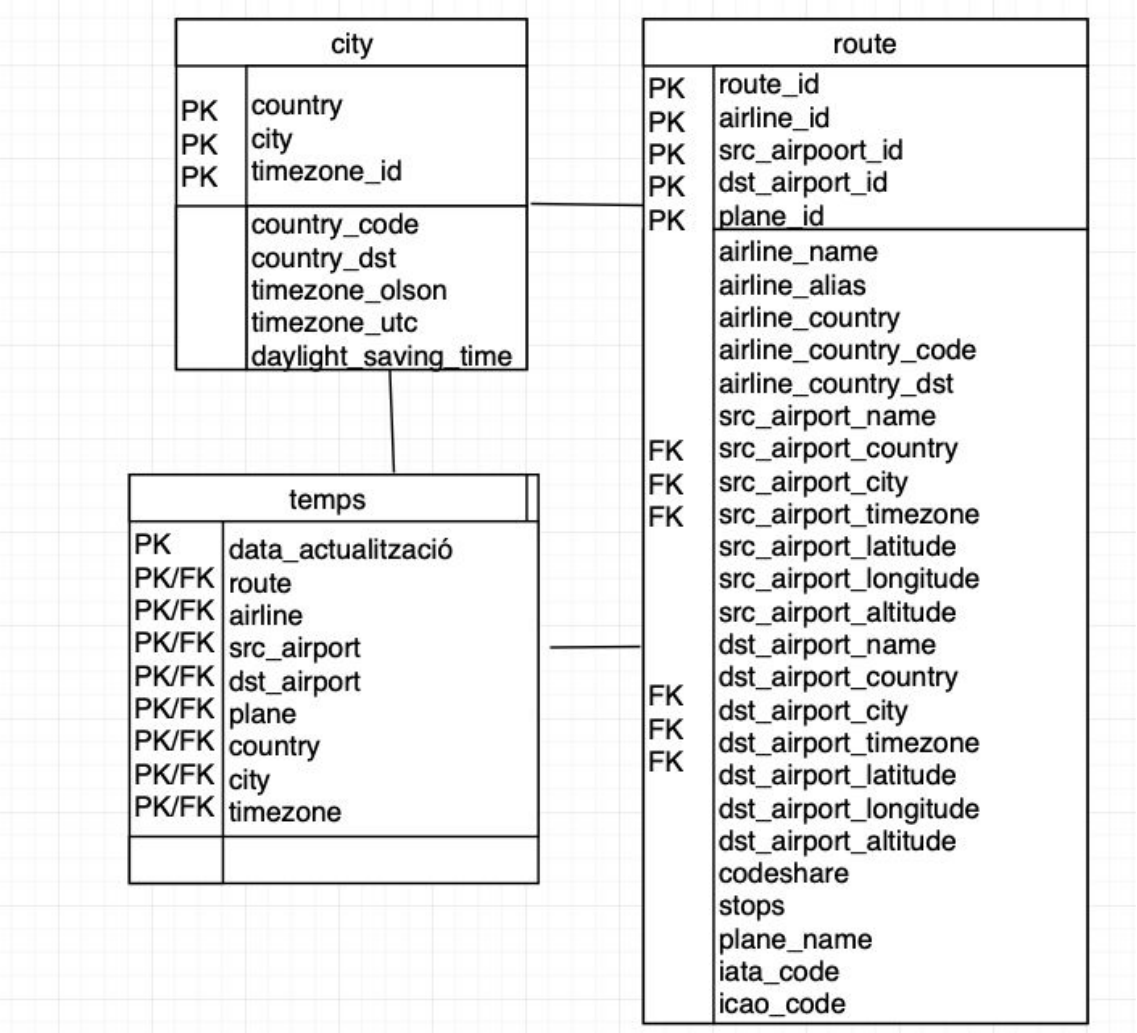
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| flight_db_00       |
| flight_db_01       |
| flight_db_02       |
+--------------------+
4 rows in set (0.00 sec)

mysql> use flight_db_00
Database changed
```

6. Fem “show tables” per veure quines taules hi ha.

```
mysql> show tables;
+------------------------+
| Tables_in_flight_db_00 |
+------------------------+
| airline                |
| airport                |
| city                   |
| country                |
| plane                  |
| route                  |
| timezone               |
| timezone_city          |
+------------------------+
8 rows in set (0.00 sec)
```

7. Fem “SELECT * FROM <nom_taula>\G” per veure cadascuna de les columnes a

Model relacional OLAP



La taula city emmagatzema tota la informació referent a la localització dels diferents elements de la base com els aeroports o les aerolínies, amb tot allò referent a la zona horària. S'han ajuntat les taules country, city, timezone_city i city de la OLTP.

La taula ruta emmagatzema la resta de taules: gaurda tot allà que s'encarrega de les rutes que fan els avions i tota la seva informació, aeroports, avió, parades, etc.

La taula temps és la que té un registre de cada transacció de la base de dades i per saber quan s'han actualitzat les dades de la base.

Programa de Java

El nostre programa java, primer mostra per pantalla totes les opcions de bases de dades que pot accedir a puigpedrós i fa escriure a l'usuari quina d'elles vol escollir.

Després, es connecta a la base de dades local i la de puigpedrós paral·lelament mitjanaçant JDBC.

Per obtenir les dades de puigpedrós, el que fa mitjançant les funcions de la classe ConnexioPuigedros és transformar totes les dades obtingues del servidor a un String (per cada taula del servidor). Llavors per inserir totes les dades en local, es fa servir la funció insert de la classe ConnexioLocal on es diu quina taula es vol actualitzar i la cadena de caràcters on estan totes les dades del servidor. Així es fa servir la funció 'INSERT INTO (nom de taula) VALUES [...]' on despres de values es posa l'String amb les dades de puigpedrós.

Abans de l'actualització de les dades, es borren totes les que hi ha en local per a poder ser introduïdes les noves dades.

Usuaris

L'enunciat demanava que es creesin tres usuaris diferents.

El primer és l'analític que s'encarrega de fer selects, crear views i veure-les. És per això que se l'ha otorgat els permisos de: 'SELECT, SHOW DATABASES, CREATE VIEW, SHOW VIEW'

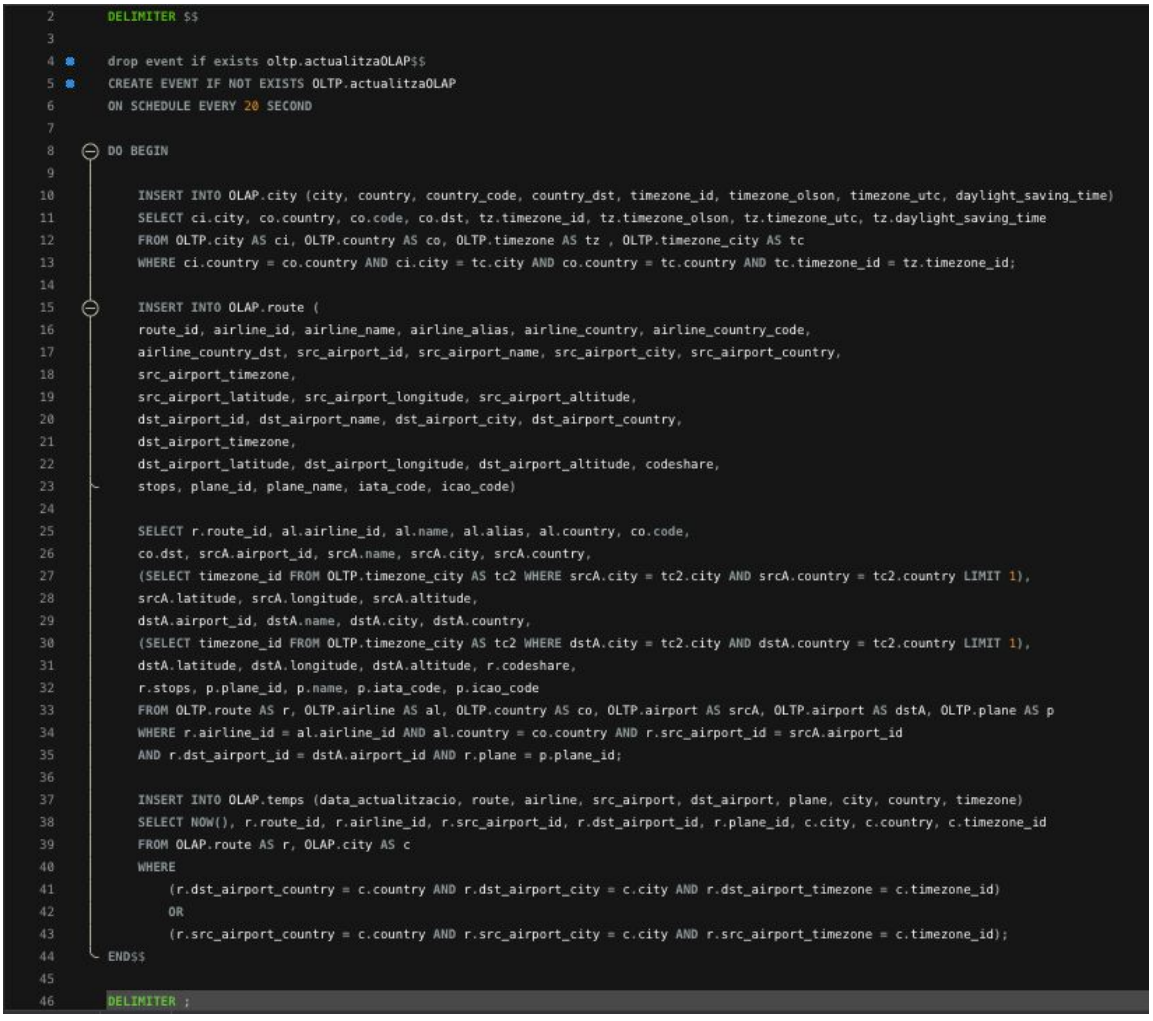
El segon, el manager s'encarrega de mantenir al dia les dues bases de dades. Se li ha otorgat els permisos necessaris per gestiona una base de dades que són: 'SELECT, UPDATE, INSERT, DELETE' i això en les dues bases OLAP i OLTP.

L'últim és el rr_hh que es limita a crear usuaris. El que pot fer és 'CREATE USER'.

EVENT

El meu event s'encarrega d'anar actualitzant la bases de dades OLAP cada 20 segons. D'aquesta forma quan s'actualitza la base de dades OLTP, es transformen totes les noves dades en el model OLAP.

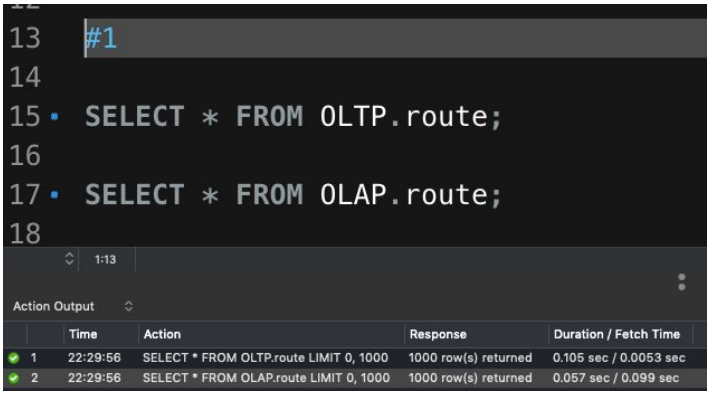
Com a la base de dades OLAP només hi ha tres taules, l'event disposa de tres INSERTS, un per a cada taula i utilitza les taules de la base de dades OLTP per fer aquest trespàs d'informació.



```
DELIMITER $$

drop event if exists oltp.actualitzaOLAP$$
CREATE EVENT IF NOT EXISTS OLTP.actualitzaOLAP
ON SCHEDULE EVERY 20 SECOND

DO BEGIN

    INSERT INTO OLAP.city (city, country, country_code, country_dst, timezone_id, timezone_olson, timezone_utc, daylight_saving_time)
    SELECT ci.city, co.country, co.code, co.dst, tz.timezone_id, tz.timezone_olson, tz.timezone_utc, tz.daylight_saving_time
    FROM OLTP.city AS ci, OLTP.country AS co, OLTP.timezone AS tz , OLTP.timezone_city AS tc
    WHERE ci.country = co.country AND ci.city = tc.city AND co.country = tc.country AND tc.timezone_id = tz.timezone_id;

    INSERT INTO OLAP.route (
    route_id, airline_id, airline_name, airline_alias, airline_country, airline_country_code,
    airline_country_dst, src_airport_id, src_airport_name, src_airport_city, src_airport_country,
    src_airport_timezone,
    src_airport_latitude, src_airport_longitude, src_airport_altitude,
    dst_airport_id, dst_airport_name, dst_airport_city, dst_airport_country,
    dst_airport_timezone,
    dst_airport_latitude, dst_airport_longitude, dst_airport_altitude, codeshare,
    stops, plane_id, plane_name, iata_code, icao_code)

    SELECT r.route_id, al.airline_id, al.name, al.alias, al.country, co.code,
    co.dst, srcA.airport_id, srcA.name, srcA.city, srcA.country,
    (SELECT timezone_id FROM OLTP.timezone_city AS tc2 WHERE srcA.city = tc2.city AND srcA.country = tc2.country LIMIT 1),
    srcA.latitude, srcA.longitude, srcA.altitude,
    dstA.airport_id, dstA.name, dstA.city, dstA.country,
    (SELECT timezone_id FROM OLTP.timezone_city AS tc2 WHERE dstA.city = tc2.city AND dstA.country = tc2.country LIMIT 1),
    dstA.latitude, dstA.longitude, dstA.altitude, r.codeshare,
    r.stops, p.plane_id, p.name, p.iata_code, p.icao_code
    FROM OLTP.route AS r, OLTP.airline AS al, OLTP.country AS co, OLTP.airport AS srcA, OLTP.airport AS dstA, OLTP.plane AS p
    WHERE r.airline_id = al.airline_id AND al.country = co.country AND r.src_airport_id = srcA.airport_id
    AND r.dst_airport_id = dstA.airport_id AND r.plane = p.plane_id;

    INSERT INTO OLAP.temps (data_actualitzacio, route, airline, src_airport, dst_airport, plane, city, country, timezone)
    SELECT NOW(), r.route_id, r.airline_id, r.src_airport_id, r.dst_airport_id, r.plane_id, c.city, c.country, c.timezone_id
    FROM OLAP.route AS r, OLAP.city AS c
    WHERE
        (r.dst_airport_country = c.country AND r.dst_airport_city = c.city AND r.dst_airport_timezone = c.timezone_id)
        OR
        (r.src_airport_country = c.country AND r.src_airport_city = c.city AND r.src_airport_timezone = c.timezone_id);
END$$

DELIMITER ;
```

OLTP vs OLAP

1.A simple query which involves only 1 table.

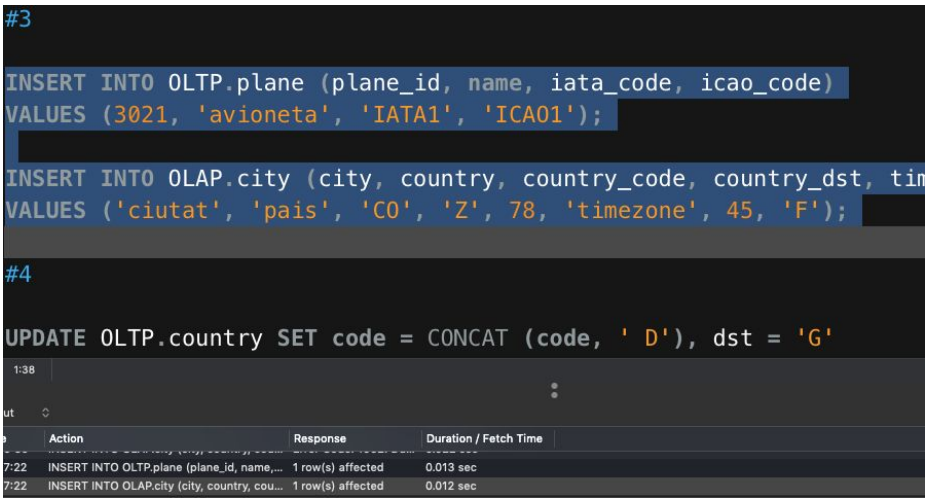


Com podem veure a la foto, fent exactament la mateixa query en les dues bases de dades, la OLAP tarda el doble que la OLTP. Això és degut a que la taula OLAP conté més camps i per això necessita més temps per carregar tota la informació.

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

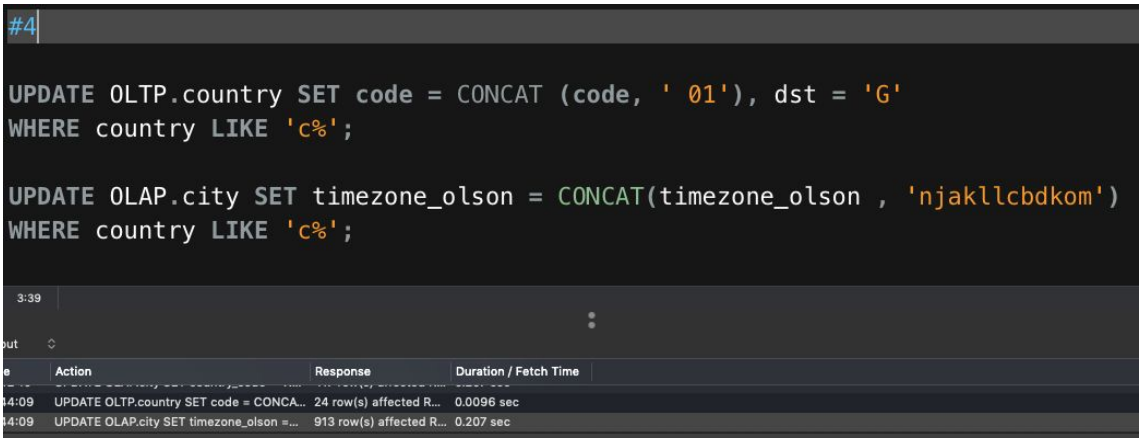
Les dues taules ensenyen exactament la mateixa informació, l'únic que la OLTP ha hagut de fer el producte cartesià de 5 taules, en canvi la OLAP ja té tota aquesta informació guardada en una taula. És per això que la OLTP ha tardat 5 cops més que la OLAP.

3.An insert into 1 table.



Fent servir la funció per inserir de VALUES, veiem que les dues bases de dades tarden el mateix. Nogensmenys, si insereixes informació des d'una base de dades OLTP fins a una OLAP es tarda molt més que no pas a l'inrevés, ja que per aconseguir tota la informació de la base de dades OLAP es necessiten fer productes cartesians de diferents taules de la OLTP; en l'inrevés és només agafar unes columnes.

4.An update into 1 field.



Per fer inserts, veiem que el format OLAP tarda molt més temps que no pas l'OLTP. Això és degut a que el model OLAP gestiona molta més informació en cada taula que l'OLTP. Per això el sistema necessita més temps.

5. A delete into 1 table.

La diferència de temps entre borrar un camp en una taula d'un sistema OLAP d'un OLTP és abismal ja que per defecte el les taules OLAP contenen molts més camps que no pas l'OLTP i és per això que s'ha d'esborrar molta més informació. Per contra, si volguessis esborrar la mateixa informació que la d'un OLAP mitjançant l'OLTP, es necessitarien fer servir productes cartesians que també impliquen molt més temps.

Temps

Temps dedicat en cada part del treball:

1. Llegir, entendre l'enunciat i documentació: 4h
2. Disseny OLAP: 3h
3. Fer funcionar el programa java: 17 hores.
4. Creació event: 8h
5. Queries: 3h
6. Usuaris: 3h
7. Memòria: 5h

Total aproximat - 43 hores

Conclusions

Per concloure, puc dir que he aprés a saber utilitzar el sistema JDBC per connectar-me a les bases de dades mitjançant java i saber gestionar tota aquesta informació. A més, he aprés a connectar-me a una base de dades que no fos local, ja que normalment sempre he m'he connectat a bases de dades del propi ordinador i mai les d'un servidor com puigpedrós.

Per altra banda, he aprés les diferents maneres de poder emmagatzemar una mateix informació mitjançant dos formats diferents com és l'OLAP i l'OLTP i veure les seves diferències: els punts dèbils de cadascuna i els punts forts.