



PRÀCTICA 2

Guillermo Serraclara (guillermo.serraclara)

Òscar Pozuelo (oscar.pozuelo)

ÍNDEX

1. Introduction.....	3
2. Database Concepts.....	4
3. OLTP Structure.....	5
4. OLAP Structure.....	7
5. Program Explanation.....	7
6. Users Explanation.....	9
7. Stored Procedure Explanation.....	10
8. Event Explanation.....	11
9. OLAP vs OLTP.....	12
10. Time Dedicated.....	14
11. Conclusions.....	14
12. Bibliography.....	14

1. INTRODUCTION

Aquesta segona pràctica del curs tenia dos objectius:

- Aprendre a treballar amb JDBC
- Aplicar els coneixements sobre Procedures, Events, Triggers i DCL

Hem hagut de fer un programa que actués com a Java DataBase Connector per poder importar una base de dades situada en un server remot, hem creat un nou model d'aquesta base de dades i hem creat diferents Procedures i Events per poder treballar amb ella. Entre els reptes que teníem, entre fer el programa JDBC o crear els events o procedures necessaris, també hem hagut de fer la connexió amb server remot i hem hagut descobrir quina era l'estructura de la base de dades d'aquest servei, i amb la informació rebuda, crear nosaltres la propia estructura.

2. DATABASE CONCEPTS

OLTP

OLTP es un tipus de processament que facilita i administra aplicacions transaccionals, normalment per a entrada de dades i recuperació i processament de transaccions. Els paquets de software per a OLTP es basen en l'arquitectura client-servidor ja que normalment solen ser utilitzats per empreses amb una red informàtica distribuïda. El processament de transaccions en línia té dos beneficis: la simplicitat i l'eficiència. Els seus inconvenients són la seguretat i els costos econòmics o de temps.

OLAP

OLAP es una sol·lució utilitzada en el camp de Intel·ligència de negocis, el qual el seu objectiu és agilitzar la consulta de grans quantitats de dades. Per això utilitza estructures de dades diverses, normalment multidimensionals, que contenen dades resumides de grans bases de dades o sistemes OLTP. S'utilitza en informes de negocis de ventes, màrqueting, informes de direcció i àrees similars.

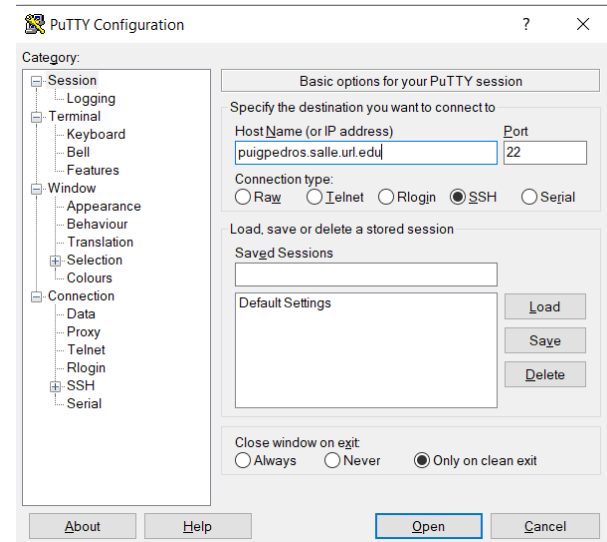
Data Warehouse

Data warehouse es com es denomina al magatzem electrònic on generalment una empresa o organització manté una gran quantitat de forma segura, fiable, fàcil de recuperar i fàcil d'administrar.

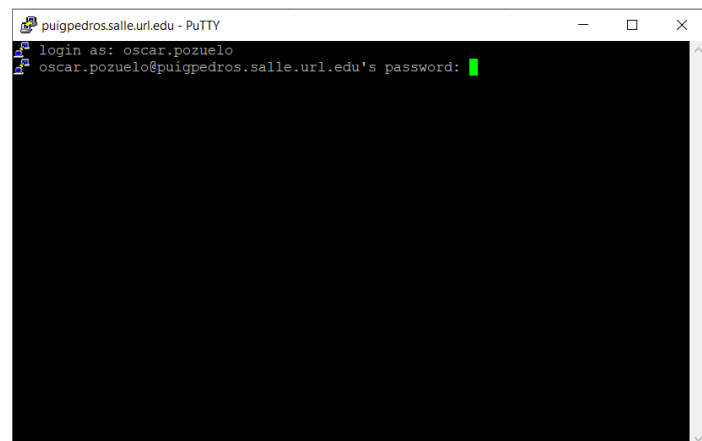
3. OLTP STRUCTURE

Per poder trobar l'estructura de la base de dades OLTP hem seguit els següents passos:

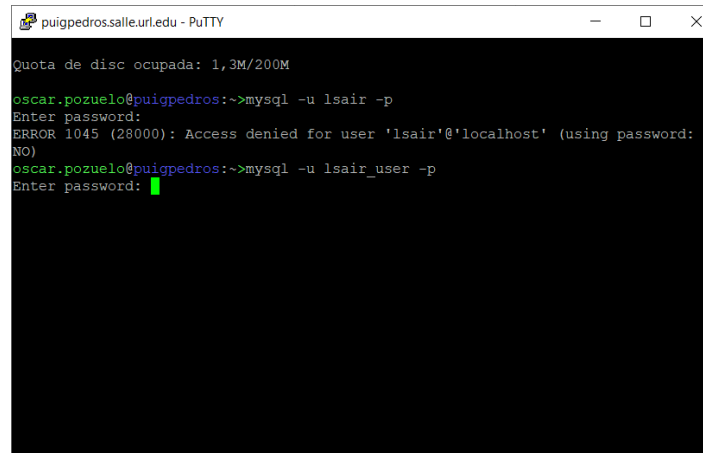
1. Obrir el programa Putty i escriure la direcció del servidor remot: `puigpedros.salle.url.edu`, i obrim la connexió.



2. Ens demanarà usuari i contrasenya, així que escriurem les nostres credencials de la universitat.

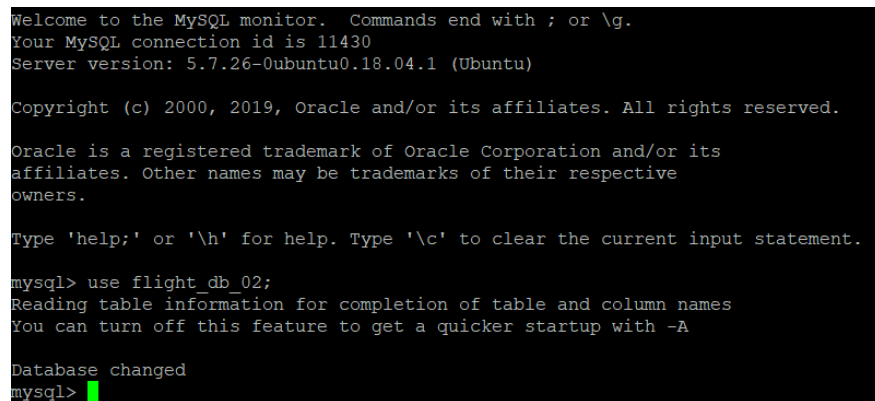


3. Un cop hem entrat, hem d'escriure la següent comanda: `mysql -u lsair_user -p` per poder accedir a les bases de dades del servidor. Un cop escrita la comanda ens demanarà la contrasenya, i escriurem `lsair_bbdd`.



```
Quota de disc ocupada: 1,3M/200M
oscar.pozuelo@puigpedros.salle.url.edu:~$ mysql -u lsair -p
Enter password:
ERROR 1045 (28000): Access denied for user 'lsair'@'localhost' (using password: NO)
oscar.pozuelo@puigpedros.salle.url.edu:~$ mysql -u lsair_user -p
Enter password: 
```

4. Com ja sabem el nom de les db guardades, fem servir una d'elles, i escribim la següent comanda: `use flight_db_02;`



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11430
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

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

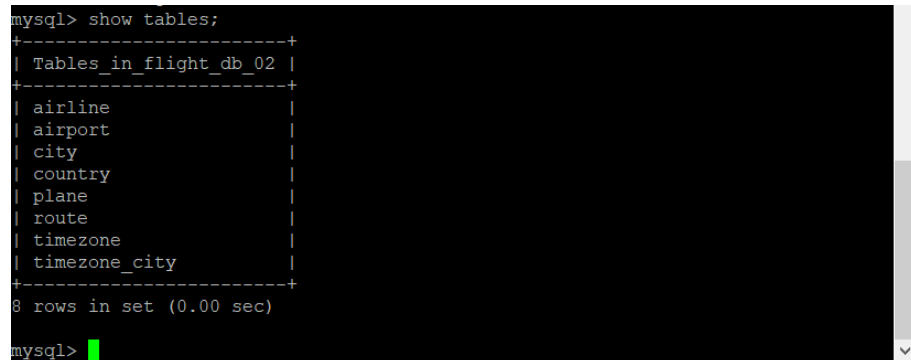
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use flight_db_02;
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> 
```

5. Ara ens cal escriure la comanda de `show tables;` per a que ens mostri el llistat amb totes les taules de la base de dades.



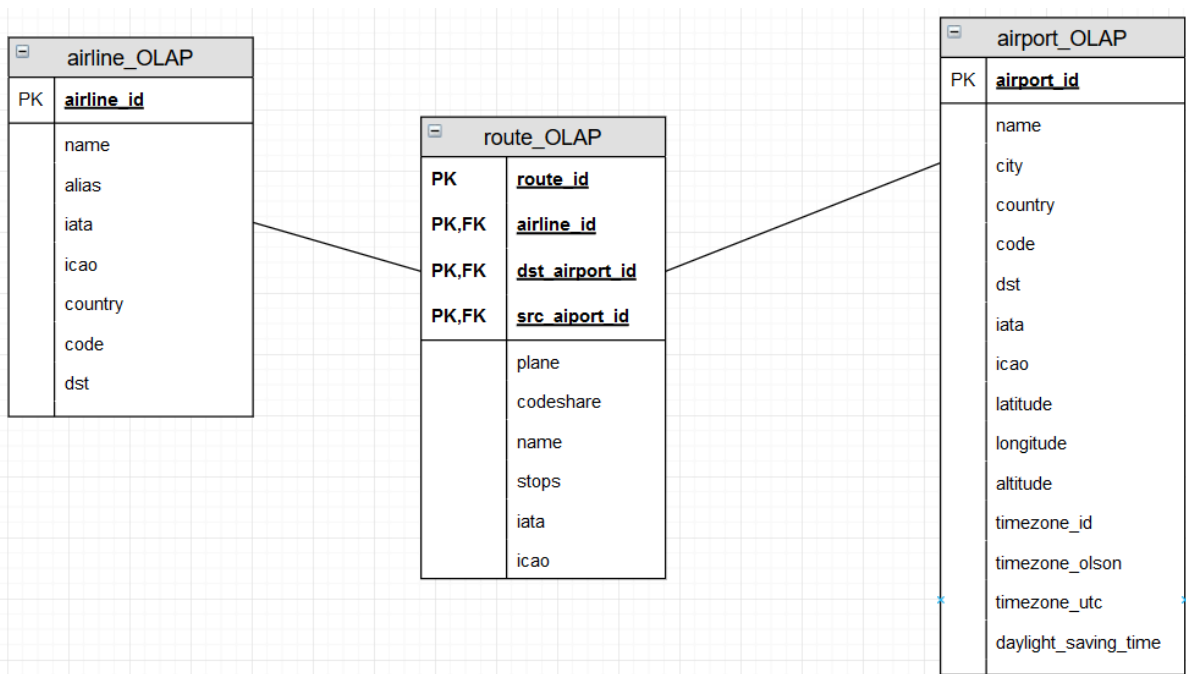
```
mysql> show tables;
+-----+
| Tables_in_flight_db_02 |
+-----+
| airline                 |
| airport                 |
| city                   |
| country                 |
| plane                  |
| route                  |
| timezone                |
| timezone_city           |
+-----+
8 rows in set (0.00 sec)

mysql> 
```

6. Per últim, per cada taula haurem de fer servir la comanda *show create table nom_taula;* per poder veure l'estructura de cada taula, els seus atributs i els seus tipus, les constraints, etc.

```
mysql> mysql> show create table plane;
+-----+-----+
| Table | Create Table |
+-----+-----+
| plane | CREATE TABLE `plane` (
  `plane_id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `iata_code` text,
  `icao_code` text,
  PRIMARY KEY (`plane_id`)
) ENGINE=InnoDB AUTO_INCREMENT=175 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

4. OLAP RELATIONAL MODEL



5. PROGRAM EXPLANATION

El programa s'inicia i ens mostra per pantalla una finestra on tenim per omplir dues caselles, una per el nom (sempre escriurem *root*) i una per la contrasenya. Aquesta primera finestra es per poder fer la connexió amb el *localhost*.

Un cop haguem posat les credencials correctament, s'ens mostrarà una nova finestra amb diversesn opcions:

- Import OLTP:

Si cliquem al botó d'aquesta opció el que farà serà iniciar la connexió amb el server remot de puigpedros i començar a importar tota la informació del model OLTP remot al model OLTP local. Es farà mitjançant dues simples queries per a cada taula: primer un select de la taula del servidor remot i després un insert a la taula del servidor local. D'aquesta manera, cada registre s'anirà copiant taula per taula. Un cop acabada aquesta importació, es tancarà la connexió amb el server remot.

- Import OLAP:

Si cliquem al botó d'aquesta opció el que farà serà cridar a un stored procedure que el que farà serà importar tota la informació del model OLTP local al model OLAP local. D'aquesta manera, s'ompliran les taules OLAP mitjançant un seguit de inserts i updates.

- Create Users:

Si cliquem al botó d'aquesta opció el que farà serà cridar a un stored procedure que s'encarrega de crear els usuaris que l'anunciat ens ha demanat. No només els crea, sinó que també es fan les gestions de permisos necessàries.

- Exit

Si cliquem al botó d'aquesta opció el que farà serà acabar la connexió amb el servidor local i tornar a la finestra d'inici.

6. USERS EXPLANATION

Necessitàvem crear 3 usuaris diferents:

- Analytic_user:

Aquest usuari pot fer servir tot tipus de SELECTS, creació i utilització de vistes en les taules del sistema OLAP.

Per tant li hem donat els privilegis de select, de create view i use view a les taules OLAP.

```
create user 'analytic_user';
grant select on lsair.airline_olap to 'analytic_user';
grant select on lsair.airport_olap to 'analytic_user';
grant select on lsair.route_olap to 'analytic_user';
grant create view on lsair.airline_olap to 'analytic_user';
grant create view on lsair.airport_olap to 'analytic_user';
grant create view on lsair.route_olap to 'analytic_user';
grant show view on lsair.airline_olap to 'analytic_user';
grant show view on lsair.airport_olap to 'analytic_user';
grant show view on lsair.route_olap to 'analytic_user';
```

- Manager_user:

Aquest usuari es responsable de mantenir les dues bases de dades actualitzades.

Per això, li hem donat els privilegis de select, update o insert sobre totes les taules del sistema.

```
create user 'manager_user';
grant select on lsair.* to 'manager_user';
grant update on lsair.* to 'manager_user';
grant insert on lsair.* to 'manager_user';
```

- Rrhh_user:

Aquest usuari es responsable de crear nous usuaris.

Per tant, li hem donat els privilegis de creació d'usuari sobre tota la base de dades.

```
create user 'rrhh_user';
grant create user on *.* to 'rrhh_user';
```

7. STORED PROCEDURES EXPLANATION

Per a aquesta pràctica, hem creat 5 stored procedures diferents:

- `creacioEstructures()`:
Aquest procedure bàsicament el que fa es crear les taules de la base de dades local per a poder ser utilitzades. El cridem a l'inici del programa. Aquí es creen tant les taules del sistema OLAP com les taules del sistema OLTP.
- `eliminaOLTP()`:
Aquest procedure el que fa es eliminar les taules del model OLTP de bases dades. L'hem creat per fer-lo servir durant la part de desenvolupament del programa.
- `eliminaOLAP()`:
Aquest procedure el que fa es eliminar les taules del model OLAL de bases dades. L'hem creat per fer-lo servir durant la part de desenvolupament del programa.
- `importacio()`:
Aquest procedure el que fa es fer la importació de totes les dades des de les taules OLTP fins les OLAP. Primer, borrem tota la informació de dins de la taula i després mitjançant inserts i updates, actualitzem el model OLAP.
- `creacioUsers()`:
Aquest procedure el que fa es la creació i la concessió de privilegis, com hem explicat en l'apartat anterior.

8. EVENT EXPLANATION

Per a aquesta pràctica hem hagut de crear un event, per a que cada dues setmanes el que fes fos actualitzat les taules del model OLAP. Per tant, l'únic que fa l'event es cridar al procedure *importacio* i les taules s'actualitzen.

```
DROP EVENT IF EXISTS updateOLAP $$  
CREATE EVENT updateOLAP  
ON SCHEDULE EVERY 2 WEEK  
DO BEGIN  
  
    call importacio;  
  
END $$  
DELIMITER ;
```

9. OLAP VS OLTP

Querys senzilles utilitzant una sola taula:

```
select name from airline_oltp where country like 'France ';
select name from airline_olap where country like 'France ';
```

I els temps d'execució:

2	16:48:23	select name from airline_olap where country like 'France ' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	16:48:36	select name from airline_oltp where country like 'France ' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Querys més complexes:

```
select r.* from route_oltp as r, airline_oltp as a1, airport_oltp as a1, airport_oltp as a2, plane_oltp as p
where r.airline_id = a1.airline_id and r.src_airport_id = a1.airport_id and r.dst_airport_id = a2.airport_id
and r.plane = p.plane_id and a1.country like 'United States ' and a1.name like '%Miami%' and a2.name like '%Detroit%'
and p.name like '%Airbus%';

select r.* from route_olap as r, airline_olap as a1, airport_olap as a1, airport_olap as a2
where r.airline_id = a1.airline_id and r.src_airport_id = a1.airport_id and r.dst_airport_id = a2.airport_id
and a1.country like 'United States ' and a1.name like '%Miami%' and a2.name like '%Detroit%' and r.name like '%Airbus%';
```

I els temps d'execució:

4	16:54:00	select r.* from route_oltp as r, airline_oltp as a1, airport_oltp as a1, airport_oltp as a2, plane_oltp as p where r.air...	1 row(s) returned	0.234 sec / 0.000 sec
5	16:54:33	select r.* from route_olap as r, airline_olap as a1, airport_olap as a1, airport_olap as a2 where r.airline_id = a1.airli...	1 row(s) returned	0.016 sec / 0.000 sec

Inserts a una taula:

```
insert into airline_oltp values(1, 'Aa', 'Bb', 'Cc', 'Ddd', 'Italy ');
insert into airline_olap values(1, 'Aa', 'Bb', 'Cc', 'Ddd', 'Italy ', 'IT', 'I');
```

I els temps d'execució:

9	16:58:38	insert into airline_oltp values(1, 'Aa', 'Bb', 'Cc', 'Ddd', 'Italy ')	1 row(s) affected	0.078 sec
10	16:58:46	insert into airline_olap values(1, 'Aa', 'Bb', 'Cc', 'Ddd', 'Italy ', 'IT', 'I')	1 row(s) affected	0.078 sec

Updates en una taula:

```
update airline_oltp set iata = 'Ab' where airline_id = 1;
update airline_olap set iata = 'Ab' where airline_id = 1;
```

I els temps d'execució:

13	17:05:53	update airline_oltp set iata = 'Ab' where airline_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.078 sec
14	17:05:58	update airline_olap set iata = 'Ab' where airline_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.094 sec

Deletes en una taula:

```
delete from airline_oltp where airline_id = 1;  
delete from airline_olap where airline_id = 1;
```

I el temps d'execució:

✓	15	17:08:22	delete from airline_oltp where airline_id = 1	1 row(s) affected	0.062 sec
✓	16	17:08:27	delete from airline_olap where airline_id = 1	1 row(s) affected	0.047 sec

Un cop fet aquests tests, ens donem compte de que en l'apartat de querys de tipus select, en les senzilles no hi ha cap diferència ja que el temps d'execució és pràcticament nul, pero quan fem un select on fem servir més de varies taules, la consulta en el model OLAP és molt més ràpida que en el model OLTP, en el cas d'aquesta prova, 14.625 vegades més ràpid, un nombre molt elevat. En quant a inserts, els dos models mostren el mateix temps d'execució. En quant a updates, podem apreciar que el model OLTP es una mica més ràpid que el model OLAP, pero sense tenir una ventatge que poguéssim considerar molt gran. En canvi al delete passa justament al contrari, el model OLAP es lleugerament més ràpid que el model OLTP.

Per tant, podem concloure que per querys del tipus SELECT, el model OLAP es molt més ràpid per a fer aquest tipus de consultes. Com més taules estiguin implicades en el select, més li costarà al model OLTP i per tant cada cop la diferència de temps d'execució serà més gran. Això es degut a que el model OLAP conté menys taules per tant no es perd tant temps en els unions o en desfer el producte cartesià entre les diferents taules. Després, en el les querys de tipus update, insert o delete la diferència no es tan gran en el temps d'execució, com a molt unes poques centèsimes.

10. TIME DEDICATED

Per a la realització d'aquesta pràctica, hem dedicat aproximadament un total de 17 hores repartides entre 2 setmanes aproximadament.

11. CONCLUSIONS

Com a conclusions podem dir que hem après a com desenvolupar una interfaç a Java que es connecti amb un servidor local i un remot de mysql, i com poder transferir informació d'una base de dades a una altre. Això ens ha fet aprofundir els nostres coneixements de Java ja que hem hagut d'investigar com poder fer les connexions, etc. També hem pogut adquirir coneixements sobre els diferents tipus de bases de dades que pot haver-hi i les seves utilitats, també quan és eficient utilitzar-ne una i quan no. Hem pogut posar en pràctica els coneixements sobre procedures, events, triggers i DCL adquirits a classe i durant les sessions de laboratori.

12. BIBLIOGRAPHY

Conectar con MySQL utilizando el driver JDBC en Java – MySQL JDBC. 3 de Febrer de 2016 [consulta: 5 d'Abril 2019]. Disponible en:
<http://codigoxules.org/conectar-mysql-utilizando-driver-jdbc-java-mysql-jdbc/>