

Informe Projecte GABD

1.INFORME MONGODB:

Repositori amb el codi:

<https://www.github.com/nilde/GABD>

1.1 Informació important:

Per a poder executar l'script s'hauria d'inserir els fitxers proporcionats per els professors a la carpeta IMAGES, i ja es podria executar tot correctament. Abans d'executar el contingut dels script en python, s'ha de realitzar l'execució de l'script en bash tal com s'explica a l'apartat 5 d'aquesta memòria.

1.2 Informació general sobre el script:

L'script està estructurat en 3 parts molt diferenciades:

1.2.1. Nexa

És el punt central del nostre script es fa la crida a aquest fitxer i ja s'encarregarà de fer totes les gestions necessaries.

El fitxer es el main.py.

Exemple de trucada al script:

```
python main.py --drop --db VECT
```

--drop assegura que la base de dades estigui neta abans d'inserir les noves dades.

--db VECT crea una nova base de dades amb el nom VECT

1.2.2. Fitxers d'extracció:

Son un conjunt de mòduls diferents, on cadascun s'utilitza per a poder extreure les dades automàticament de tots els fitxers que contenen les dades que s'han d'inserir. Cadascun dels fitxers al tenir formats diferents, necessiten diverses tècniques per a poder realitzar les insercions en el format correcte

Els fitxers són:

```
ExtractDataData  
ExtractDataImages  
ExtractDataNames  
ExtractDataResults
```

1.2.3. Fitxers d'inserció:

Son un altre conjunt de petits mòduls, que s'encarreguen de configurar les dades tal que es puguin insertar correctament en els documents de la base de dades. És aquí on hi ha la configuració implícita del format dels documents.

Els fitxers són:

MakeInsertionsToDatabaseData.py
MakeInsertionsToDatabaseImages.py
MakeInsertionsToDatabaseMethods.py
MakeInsertionsToDatabaseNames.py
MakeInsertionsToDatabaseResults.py

1.2.4. Fitxer de creació dels usuaris:

Fitxer que conté una assignació de diferents rols.

El fitxer és el MakeInsertionsUsers.py

1.2.5. Fitxer de configuració:

També hi ha un fitxer de configuració que permet modificar certes parts de la generació de la base de dades.

El fitxer és el configFile.py

Estructura de la base de dades:

La base de dades té un nom variable, que contindrà tots els documents generats.

Els documents generats són:

A. DATABASE_INFO

Conté tota la informació sobre les bases de dades,

B. RESULTS

Contindrà tota l'informació relacionada amb la generació dels resultats.

C. SET_UP_EXPERIMENTS

Contrindrà tota l'informació associada a les configuracions dels diferents experiments.

D.METHODS

Contindrà tota l'informació associada als diferents mètodes d'evaluació que s'utilitzen en la detecció d'outliers

E. EXPERIMENTS

Contindrà tota l'informació relacionada amb la generació dels experiments.

F. VECTORS

Contindrà tota l'informació associada amb els vectors de les diferents bases de dades importades de l'UCI.

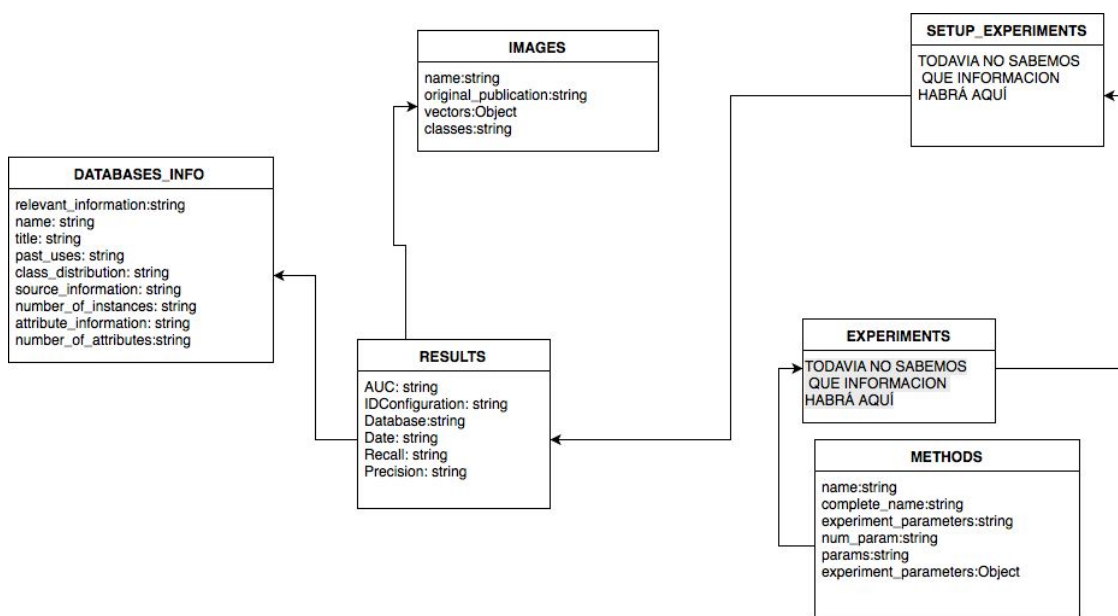
G. IMAGES

Contindrà tota l'informació de les dades associades a les diferents imatges que s'han de classificar, tant les seves imatges com els vectors de característiques que estàn associats.

1.2.6 Observacions en la generació:

En la creació de la base de dades, ens hem adonat de que l'estructura que generem té un impacte molt significatiu en el rendiment que obtenim després. Si obtem per des

1.3 Diagrama de classes de la base de dades:



1.4 Tutorial execució dels scripts de bash:

1. Obrir el terminal
2. Situarse al directori (directori actual + /GABD/mongoDBScripts/consoleScripts)

3. Executar la comanda `chmod 700 scriptMongo.js`

4. Executar la comanda `mongo < scriptMongo.js`

El petit script aquest, permet utilitzar tamany del chunk personalitzats. Està posat a 1024MB per a poder permetre que càpiguen totes les dades necessaries.

1.5 Exercicis de Seguretat de BD: creació d'usuaris i drets d'accés (associats a mongo)

1. Feu el disseny dels perfils, rols i privilegis necessaris per a implementar els usuaris descrits en la secció anterior. Per a cada usuari definiu un primer conjunt de privilegis segons les necessitats que preveieu que pugueu tenir. Com a mínim el conjunt de privilegis que haurà de tenir cada usuari són:

Detector d'outliers: consultes a les taules o col·leccions.

Gestor de Dades: operacions de lectura, actualització i esborrat de dades.

Desenvolupador: creació, modificació i esborrat de taules o col·leccions.

Auditors: veure els objectes associats al projecte i connexions actives.

Més endavant haureu d'ampliar els privilegis per algun dels usuaris que heu definit. Ho haureu de fer i justificar arribat el moment.

Aquesta creació dels usuaris s'ha realitzat al fitxer **makeInsertionUsers**, on hem utilitzat simplement rols per defecte que té mongoDB. L'implementació d'aquestes creacions es troben de la línia 9 a la 13 i necessitem utilitzar el client que s'utilitza en la creació i inserció de les dades.

2. Feu un script que mostri, per a cada usuari i rol que hageu creat, els privilegis assignats a cadascun d'ells. En cas que creeu algun perfil, el script que creeu també ho haurà de mostrar així com els paràmetres que hagueu configurat. Consulteu la documentació associada a aquesta sessió per obtenir un llistat d'instruccions, taules i vistes on buscar la informació que es demana. Aquest script s'ha d'executar des de l'usuari que faci les funcions de DBA en cada SGBD.

Per a comprovar quins son els usuaris associats a la BD de dades, ho hem realitzat de dues maneres diferents, la primera es mitjançant l'execució del script que inserta les dades i un cop surt la consola interactiva seleccionar l'opció `usersinfo`, i retornarà una llista completa dels usuaris que hi han a la bd. La segona es mitjançant un microscript que es diu `showUsers` i on l'execució es fa mitjançant la comanda `mongo < showUsers.js`, (pot ser necessari donar permisos d'execució `chmod +x`, o bé `chmod 700`).

3. Feu un script de validació que creï un joc de proves que posi en evidència els rols/privilegis assignats a cada usuari.). Aquest script haurà de crear una taula Test amb dos camps: camp1 i camp2 amb tipus varchar2 i number, respectivament. A continuació haurà d'inserir almenys 2 registres, modificar-ne un i esborrar un altre. També haurà de fer una consulta. Aquest script s'ha d'executar des de l'usuari que faci les funcions de DBA. Dins el script, s'haurà de canviar d'usuari per comprovar que pot fer, i que no pot fer, cadascun un d'ells.

Per a poder permetre la connexió amb autenticació, el que s'ha de fer primer es generar tots els rols que es vulguin i després engegar el dimoni de mongo amb els següents paràmetres `mongod --auth --port 27017 --dbpath /data/db`. Això permetrà la connexió mitjançant autenticació.

Per a poder realitzar les comprovacions necessaries, s'ha de fer la crida al script testUsers, amb la comanda **mongo < testUsers.js** al directori /GABD/mongoDBScripts/consoleScripts

1.6 Exercici Nivell intern dels SGBD

4. [mongoDB] Indiqueu a quin directori es troben els fitxers de col·leccions que conté mongoDB. Indiqueu també el nom dels fitxers que conté i comproveu si hi ha alguna modificació si intentem crear una nova col·lecció. Investigueu si podem trobar la mida de les col·leccions des de la base de dades. Si és així genereu el script que ho evidenciï.

Per saber quin path estem utilitzant, quan correm el dimoni de mongo, el que hem de fer es utilitzar una opció concreta que permet especificar un path concret on es trobaran allotjats tots els fitxers relacionats amb la base de dades que estarà en execució.

El nom dels fitxers que contenen las col·leccions són tots aquells que contenen el prefix collections dintre de la carpeta. També hi ha un fitxer _mdb_catalog.wt que conté informació genèrica sobre el contingut total de la base de dades.

A més a més, conté una sèrie de fitxers d'index que indiquen tots els índex associats a les col·leccions. També hi han més fitxers tots relacionats amb la supervisió de la base de dades.

La mida de les col·leccions de la base de dades, la podem trobar mitjançant la connexió amb un client gràfic directament o mitjançant la comanda db.stats.

L'script que conté aquesta prova es el showDatabaseInfo.

*Per a poder comprovar el correcte funcionament millor primer executar el script de test amb la comanda, **mongo < testUsers.js** i després **mongo < showDatabaseInfo.js***

5. [Oracle i mongoDB] Feu la inserció de les dades utilitzant els scripts en python que us proporcionem². Dels scripts només heu de modificar la part on es fan els inserts (ja que els heu d'adaptar a l'estructura de taules que heu dissenyat) així com els paràmetres de connexió. Feu la inserció de les dades amb el usuari de tipus Desenvolupador. Les funcions, o procediments, que haureu d'editar són: insertVectorDataset, insertImageDataset i insertDescriptors i que trobareu als fitxers: oracleConnection i mongoConnection. Important!! Aspectes a tenir en compte:

insertVectorDataset de les dades de UCI està insertat a la base de dades, el insertImageDataset encara no.

ii) [Oracle i mongoDB] No confirmeu cap transacció fins que no estiguin fetes totes les insercions de cada una de les funcions python que heu modificat.

*Això, es pot fer mitjançant el script main.py contingut en la carpeta GABD/mongoDBScripts amb la comanda **python main.py —drop —db VECT***

6. [Oracle i mongoDB] Feu les consultes que calguin per demostrar que heu inserit correctament les dades del exercici 5.

*Això es fa mitjançant l'script checkInsertions amb la comanda **mongo < checkInsertions.js**, s'ha de tenir en compte que primer s'ha de realitzar la inserció de les dades anteriors.*

De moment únicament es retornan tots els elements ja que ja serveix per a mirar si les dades s'han inserit correctament.

7. Altres consideracions

Per falta de temps, l'script únicament treballa en local això se soluciona copiant directament el client que ens proporciona el professorat en el seu script i fent les crides mitjançant aquest.

2. INFORME ORACLE:

2.1 Adaptació del script nostre per oracle

Amb l' script que hem explicat al principi de l'informe l'hem adaptat amb el format de SQL per oracle per així treballar amb el mateix script.

El que hem fet es crear una classe que s'encarregui d'oracle. Una funció per connectarse a la base de dades, una generica d'inserció de dades quan el format era similar al del mongoDB, y dues funcions específiques quan l'estructura era massa diferent que la de mongoDB.

2.2 Exercicis de Seguretat

1. Feu el disseny dels perfils, rols i privilegis necessaris per a implementar els usuaris descrits en la secció anterior. Per a cada usuari definiu un primer conjunt de privilegis segons les necessitats que preveieu que pugueu tenir. Com a mínim el conjunt de privilegis que haurà de tenir cada usuari són:

Detector d'outliers: consultes a les taules o col·leccions.

Gestor de Dades: operacions de lectura, actualització i esborrat de dades.

Desenvolupador: creació, modificació i esborrat de taules o col·leccions.

Auditors: veure els objectes associats al projecte i connexions actives.

Més endavant haureu d'ampliar els privilegis per algun dels usuaris que heu definit. Ho haureu de fer i justificar arribat el moment.

Creem 4 usuaris en la nostra base de dades, així que ho hem organitzat en:

- ROLS: amb els que agrupem diversos privilegis.
- PERFILS: amb ells configurem sessions per cada usuari, contrasenya i gestionem paràmetres pel temps de connexió a la base de dades.
- USUARIS: agreguen els rols i perfils creats anteriorment i en algun cas s'afegeix algun privilegi més.

Per implementar-lo, hem començat a crear diferents rols per a cada usuari amb els seus privilegis específics. Com en els tres primers són usuaris que adquireixin més privilegis en herència entre aquests rols per així no repetir "Privilegis". Després hem creat perfils per configurar cada connexió

i opcions de contrasenyes per a cadascun d'ells i, finalment, la creació d'usuaris i assignacions de cada rol i perfil a cada un.

A continuació, explicarem quin privilegi conté cada rol. Sempre treballarem amb els privilegis que treballen amb "TABLES" i "INDEXS" per accedir a les dades i taules de la nostre base de dades.

ROLS:

Hem aplicat els privilegis amb els rols amb la comanda "GRANT" la qual assigna el privilegi.

- Rol_Low: Aquest rol contindrà el "SELECT" lectures i "CREATE SESSION" per tal que l'usuari pugui connectar.
- Rol_Medium: Contindrà "Rol_Low" i també li afegim els privilegis de modificació i la supressió de les dades: "UPDATE, BACKUP, INSERT, COMMENT, ALTER,".
- Rol_High: Contindrà "Rol_Medium" i la modificació, la creació i la supressió de les taules: "CREATE, DELETE, DROP".
- Rol_Super_Connections_Objects: que contindrà la "CREATE SESSION" i la "ADMINISTER SQL MANAGEMENT OBJECT".

Els perfils els hem anomenat com els propis usuaris que s'especifiquen a l'enunciat ja que cadascú té uns paràmetres concrets d'expiracions de contrasenya, connexions actives, etc.

Finalment, la creació i l'assignació dels rols anteriors amb els usuaris és la següent:

AA_DETECTOR_OUTLIERS amb contrasenya "12" → Rol_Low

AA_GESTOR_DADES amb contrasenya "1234" → Rol_Medium

AA_DESENVOLUPADOR amb contrasenya "123456" → Rol_High

AA_AUDITOR amb contrasenya "12345678" → Rol_Super_Connections_Objects

Adjuntem l' script "SQL_SECURITY.sql" el qual conté tota la creació de l'estructura dels usuaris.

2. Feu un script que mostri, per a cada usuari i rol que hageu creat, els privilegis assignats a cadascun d'ells. En cas que creeu algun perfil, el script que creeu també ho haurà de mostrar així com els paràmetres que hagueu configurat. Consulteu la documentació associada a aquesta sessió per obtenir un llistat d'instruccions, taules i vistes on buscar la informació que es demana. Aquest script s'ha d'executar des de l'usuari que faci les funcions de DBA en cada SGBD.

Fent la connexió amb l'usuari desitjat i consultant les taules cada usuari pot mostrar els seus privilegis, rols i paràmetres de perfil que té cadascun. Aquestes taules són: USER_ROLE_PRIVS; ROLE_SYS_PRIVS i USER_USERS.

A continuació podem veure la sortida on mostra tota la informació demanada de l'usuari connectat.


```
Salida de Script x
Tarea terminada en 0,712 segundos

-----
USERNAME                                GRANTED_ROLE                                ADM DEF OS_
-----
AA_DETECTOR_OUTLIERS                    ROL_LOW                                     NO  YES NO

ROLE                                    PRIVILEGE                                    ADM
-----
ROL_LOW                                CREATE SESSION                                NO
CONNECT                                CREATE SESSION                                NO
ROL_LOW                                SELECT ANY TABLE                            NO

La conexi3n creada por el comando de script CONNECT se ha desconectado
```

```
Salida de Script x
Tarea terminada en 0,555 segundos

Conectado.

-----
USERNAME                                USER_ID ACCOUNT_STATUS                                LOCK_DATE                                EXPIRY_DATE
-----
EXTERNAL_NAME
-----
AA_DESENVOLUPADOR                        102 OPEN                                     30/03/19

La conexi3n creada por el comando de script CONNECT se ha desconectado
```

```
Salida de Script x
Tarea terminada en 0,555 segundos

-----
DEFAULT_TABLESPACE                    TEMPORARY_TABLESPACE                    CREATED                                INITIAL_RSRC_CONSUMER_GROUP
-----
USERS                                TEMP                                01/10/18                                DEFAULT_CONSUMER_GROUP
```

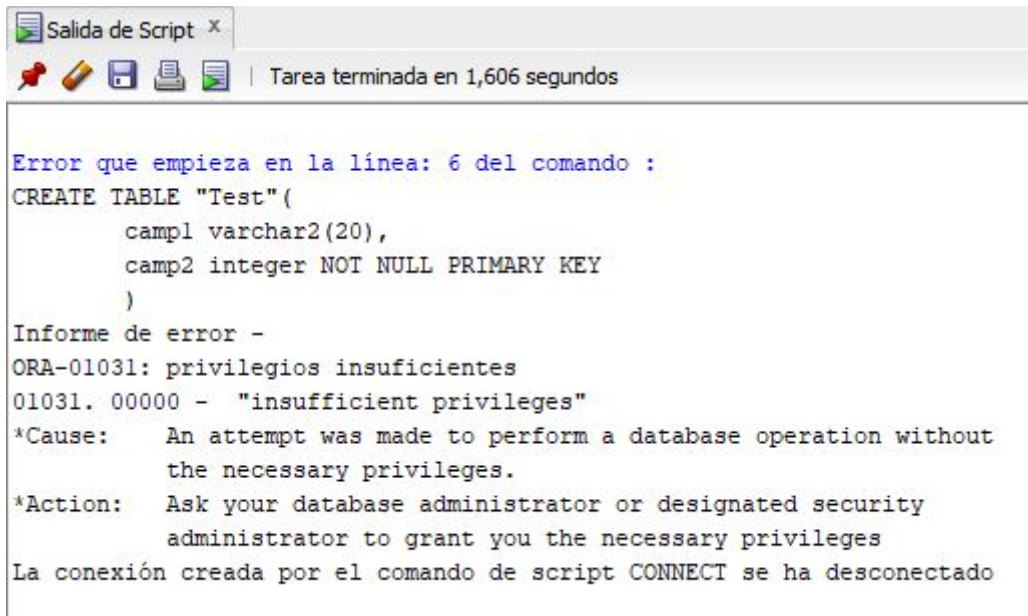
Adjuntem l'script corresponent aquest exercici amb el nom de "SHOW_PRIVILEGES_ROLES_PROFILES.sql".

3. Feu un script de validaci3n que crei un joc de proves que posi en evid3ncia els rols/privilegis assignats a cada usuari.). Aquest script haur3 de crear una taula Test amb dos camps: camp1 i camp2 amb tipus varchar2 i number, respectivament. A continuaci3n haur3 d'inserir almenys 2 registres, modificar-ne un i esborrar un altre. Tamb3 haur3 de fer una consulta. Aquest script s'ha d'executar des de l'usuari que faci les funcions de DBA. Dins el script, s'haur3 de canviar d'usuari per comprovar que pot fer, i que no pot fer, cadascun un d'ells.

Per comprovar que tota l'estructura dels nostres usuaris funciona adjuntem l'script "SQL_VALIDATION.sql" el qual executa un joc de proves de creació de la taula especificada i a continuació varies operacions amb les que es tracta la taula i les seves dades i finalment l'eliminació d'aquesta.

Per canviar d'usuari està especificat a l'script amb 4 connectes comentats i per escollir l'usuari desitjat s'ha de descomentar el que es vol.

En cas de voler executar alguna comanda SQL on l'usuari no té el permís per fer-ho la sortida del SQL Developer mostra una excepció de privilegis com aquesta.



The screenshot shows a window titled "Salida de Script" with a toolbar containing icons for redo, undo, save, print, and a magnifying glass. The status bar indicates "Tarea terminada en 1,606 segundos". The main text area displays the following error message:

```
Error que empieza en la línea: 6 del comando :
CREATE TABLE "Test"(
    camp1 varchar2(20),
    camp2 integer NOT NULL PRIMARY KEY
)
Informe de error -
ORA-01031: privilegios insuficientes
01031. 00000 - "insufficient privileges"
*Cause:      An attempt was made to perform a database operation without
              the necessary privileges.
*Action:     Ask your database administrator or designated security
              administrator to grant you the necessary privileges
La conexión creada por el comando de script CONNECT se ha desconectado
```

2.3 Exercicis de Disseny Físic

1.[Oracle] Busqueu l'espai que ocupa en disc les estructures lògiques de l'usuari SYS. Per a això busqueu dins el directori d'instal·lació d'Oracle EE. Un cop localitzats, apunteu l'espai que ocupen. Per comprovar-ho feu una consulta SQL que retorni la mida i el percentatge utilitzat en disc (PISTA! Busqueu informació sobre les taules dba_free_space, dba_data_files, dba_tablespace, dba_segments).

El primer select obté la mida total i li resta l'espai lliure, donant com a resultat l'espai utilitzat. Pel que fa al segon select, es calcula el tant per cent de l'espai utilitzat fent una resta del total amb l'espai lliure, dividint-lo entre el total i multiplicant-ho per 100.

```
SELECT
(SELECT ROUND(sum(bytes)/1024/1024,0)
FROM dba_data_files
WHERE tablespace_name='SYSTEM') - (
SELECT ROUND(sum(bytes)/1024/1024,0)
FROM dba_free_space
WHERE tablespace_name='SYSTEM') as EspaiUtilitzatEnMB,
((SELECT ROUND(sum(bytes)/1024/1024,0)
FROM dba_data_files
WHERE tablespace_name='SYSTEM') - (
SELECT ROUND(sum(bytes)/1024/1024,0)
```

```

FROM dba_free_space
WHERE tablespace_name='SYSTEM')) / (
SELECT ROUND(sum(bytes)/1024/1024,0)
FROM dba_data_files
WHERE tablespace_name='SYSTEM')*100 as PercentatgeEspaiUtilitzat
FROM dual;

```

ESPAIUTILITZATENMB	PERCENTATGEESPAIUTILITZAT
685	99,2753623

El resultat es mostra en MB i en tant per cent.

2. [Oracle] Amb l'usuari SYS, feu un script per la creació d'un tablespace de prova que heu d'anomenar: PROVA. Aquest script haurà de crear un tablespace que ocupi com a màxim 5MB amb nom de datafile "prova.dbf". Creeu una taula Test, dins el tablespace PROVA, amb dos camps: camp1 i camp2 amb tipus varchar2 i number, respectivament. Feu una consulta per comprovar el espai lliure del tablespace. A continuació haureu d'inserir 20 registres. Torneu a fer una consulta per a comprovar l'espai.

```

--canviem a l'usuari sys
ALTER SESSION SET CURRENT_SCHEMA = SYS;

```

```

--fem un drop del tablespace existent
drop tablespace prova INCLUDING CONTENTS and datafiles;

```

```

--creem el nou tablespace
CREATE TABLESPACE PROVA
DATAFILE 'prova.dbf'
SIZE 5M;

```

```

--creem la taula test al tablespace prova
CREATE TABLE Test (
  camp1 varchar2(255),
  camp2 number
)
TABLESPACE PROVA;

```

```

--calculem l'espai lliure del tablespace
SELECT ROUND(sum(bytes)/1024/1024,0) as EspaiLliureEnMB
FROM dba_free_space
WHERE tablespace_name='PROVA';

```

Tablespace PROVA borrado.

TABLESPACE PROVA creado.

Table TEST creado.

```
ESPAILLIUREENMB
-----
4
```

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

L'espai lliure del tablespace en aquest punt és el mostrat en MB

```
--inserir 20 registres
INSERT INTO Test
(camp1, camp2)
VALUES ('a', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('b', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('c', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('d', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('e', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('f', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('g', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('h', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('i', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('j', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('k', 1);
```

```

INSERT INTO Test
(camp1, camp2)
VALUES ('l', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('m', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('n', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('o', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('p', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('q', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('r', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('s', 1);
INSERT INTO Test
(camp1, camp2)
VALUES ('t', 1);

```

```

--tornem a calcular l'espai lliure del tablespace
SELECT ROUND(sum(bytes)/1024/1024,0) as EspaiLliureEnMB
FROM dba_free_space
WHERE tablespace_name='PROVA';

```

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

ESPAILLIUREENMB

4

Després d'inserir els 20 registres l'espai lliure del tablespace és el mostrat en MB

3. [Oracle] Amb l'usuari SYS, creeu un tablespace anomenat outliersDB per al que reservarem 10MB. Amb l'usuari de tipus Desenvolupador, creeu les taules que porteu

dissenyades del treball previ. Modifiqueu, si cal, els privilegis de l'usuari Desenvolupador per a que les creï en el tablespace que acabeu de crear.

```
ALTER SESSION SET CURRENT_SCHEMA = SYS;
```

```
drop tablespace outliersDB INCLUDING CONTENTS and datafiles;
```

```
CREATE TABLESPACE outliersDB  
DATAFILE 'outliersdb.dbf'  
SIZE 10M;
```

```
ALTER SESSION SET CURRENT_SCHEMA = AA_DESENVOLUPADOR;
```

```
CREATE TABLE NeuralNets (  
    id integer Not Null Primary Key,  
    nombre varchar2(255),  
    descripcion varchar2(255),  
    Configuration varchar2(255)  
)  
TABLESPACE outliersDB;
```

```
CREATE TABLE Images (  
    ID integer Not Null Primary Key,  
    ImagesValues varchar2(255),  
    labels varchar2(255),  
    NeuralNet_ID integer not null references NeuralNets,  
    isOutlier integer  
)  
TABLESPACE outliersDB;
```

```
CREATE TABLE Results_Images (  
    NeuralNet_ID integer not null references NeuralNets,  
    SetUpExperiment_ID integer not null,  
    Experiments_ID integer not null,  
    AUC varchar2(255),  
    precisio varchar2(255),  
    Recall varchar2(255),  
    primary key (SetUpExperiment_ID, Experiments_ID)  
)  
TABLESPACE outliersDB;
```

```
CREATE TABLE Methods (  
    ID integer Not Null Primary Key,  
    descripcion varchar2(255),  
    nombre varchar2(255),  
    parametros varchar2(255)  
)  
TABLESPACE outliersDB;
```

```
CREATE TABLE DatabaseInfo (  
    ID integer Not Null Primary Key,  
    Class varchar2(255),  
    Title varchar2(255),  
    SourceInformation varchar2(255),
```

```
PastUsage varchar2(255),
RelevantInformation varchar2(255),
NumberofInstances varchar2(255),
NumberOfAttributes varchar2(255),
AttributeInformation varchar2(255),
MissingAttributeValues varchar2(255),
ClassDistribution varchar2(255)
```

```
)
TABLESPACE outliersDB;
```

```
CREATE TABLE Vector (
  ID integer Not Null Primary Key,
  VectorValues varchar2(255),
  Clase varchar2(255),
  DatabaseInfo_ID integer not null references DatabaseInfo,
  isOutlier integer
```

```
)
TABLESPACE outliersDB;
```

```
CREATE TABLE SetupExperiment (
  id integer not null primary key,
  Images_ID integer not null references Images,
  Vector_ID integer not null references Vector
```

```
)
TABLESPACE outliersDB;
```

```
CREATE TABLE Experiments (
  SetUpExperiment_ID integer not null references SetUpExperiment,
  Methods_ID integer not null references Methods,
  ID integer Not Null Primary Key,
  Parameters_Values varchar2(255)
```

```
)
TABLESPACE outliersDB;
```

```
CREATE TABLE SetupExperimentHASExperiment (
  SetupExperiment_ID int not null references SetupExperiment,
  Experiments_ID int not null references Experiments,
  fecha DATE
```

```
)
TABLESPACE outliersDB;
```

```
CREATE TABLE Results_Vector (
  DatabaseInfo_ID integer not null references DatabaseInfo,
  SetUpExperiment_ID integer not null,
  Experiments_ID integer not null,
  AUC varchar2(255),
  precisio varchar2(255),
  Recall varchar2(255),
  IDConfiguration integer,
  primary key (SetUpExperiment_ID, Experiments_ID)
```

```
)
TABLESPACE outliersDB;
```

5. [Oracle i mongoDB] Feu la inserció de les dades utilitzant els scripts en python que us proporcionem2. Dels scripts només heu de modificar la part on es fan els inserts (ja que els heu d'adaptar a l'estructura de taules que heu dissenyat) així com els paràmetres de connexió. Feu la inserció de les dades amb el usuari de tipus Desenvolupador. Les funcions, o procediments, que haureu d'editar són: insertVectorDataset, insertImageDataset i insertDescriptors i que trobareu als fitxers: oracleConnection i mongoConnection. Important!! Aspectes a tenir en compte:

i) [Oracle] Si la mida del tablespace no suporta la inserció de totes les dades genereu el script SQL per a modificar la mida del tablespace creat i torneu a executar el script python per inserir les dades.

ii) [Oracle i mongoDB] No confirmeu cap transacció fins que no estiguin fetes totes les insercions de cada una de les funcions python que heu modificat.

6. [Oracle i mongoDB] Feu les consultes que calguin per demostrar que heu inserit correctament les dades del exercici 5.

```
SELECT ID, Class, Title, Source_Information, PAST_USES, Relevant_Information,  
Number_of_Instances, Number_of_Attributes, Attribute_Information, Missing_Attribute_Values,  
Class_Distribution  
FROM DATABASEINFO;
```

```
SELECT DATABASEINFO_ID, CLASE, VECTORVALUES  
FROM VECTOR ;
```

```
SELECT LABELS, IMAGEVALUES  
FROM IMAGES;
```

```
SELECT NAME, DESCRIPTION, CONFIGURATION  
FROM NEURALNETS;  
SELECT NOMBRE, DESCRIPCION PARAMETROS  
FROM METHODS;
```


ENTREGA 2

Sessió 10 Oracle

Exercici 0. En el SQLDeveloper creeu una connexió al servidor oracle-2 amb l'usuari sys. Recordeu que haureu de fer un nou forward local a la màquina oracle-2.<grup>.gabd. Habilitau una connexió de DBA a la pestanya de DBA del SQLDeveloper.

Hem creat la nova connexió i ens hem connectat a oracle-2 per poder fer les proves de backup i recuperació.



Exercici 1a. Connecteu-vos mitjançant ssh al servidor Oracle-2. Mou tots els datafiles (/u01/app/oracle/oradata/EE/*.dbf) al directori /u03/tbs. Des del sqlplus apagueu la base de dades i inicieu-la (vegeu Exemple 1). Què ha passat? Configura la base de dades per a que els trobi. Com ho heu fet? Heu pogut recuperar tota la informació?

Una vegada connectats al oracle-2 anirem a la carpeta /u01/app/oracle/oradata/EE/*.dbf per mostrar tots els datafiles i els mourem al directori /u03/tbs

Una vegada moguts executarem les comandes del sqlplus shutdown abort i startup per tirar la base de dades i inicialitzar-la un altre cop on podrem veure com no pot recuperar els datafiles moguts.

```
[oracle@oracle-2 ~]$ ls /u01/app/oracle/oradata/EE/*
/u01/app/oracle/oradata/EE/control01.ctl /u01/app/oracle/oradata/EE/redo01.log /u01/app/oracle/oradata/EE/redo02.log /u01/app/oracle/oradata/EE/redo03.log
[oracle@oracle-2 ~]$ ls /u03/tbs/
sysaux01.dbf system01.dbf temp01.dbf undotbs01.dbf users01.dbf
[oracle@oracle-2 ~]$ ls /u03/tbs/
sysaux01.dbf system01.dbf temp01.dbf undotbs01.dbf users01.dbf
[oracle@oracle-2 ~]$ sqlplus sys/oracle as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Dec 14 11:43:41 2018

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> shutdown immediate;
ORA-00604: error occurred at recursive SQL level 1
ORA-01116: error in opening database file 1
ORA-01110: data file 1: '/u01/app/oracle/oradata/EE/system01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> shutdown abort;
ORACLE instance shut down.
SQL> startup;
ORACLE instance started.

Total System Global Area 1603411968 bytes
Fixed Size 2213776 bytes
Variable Size 973080688 bytes
Database Buffers 603979776 bytes
Redo Buffers 24137728 bytes
Database mounted.
ORA-01157: cannot identify/lock data file 1 - see DBWR trace file
ORA-01110: data file 1: '/u01/app/oracle/oradata/EE/system01.dbf'
```

El que ha passat és que no troba els fitxers de la base de dades en el path que té assignat, i per tant s'ha de canviar el path per a que vagi al directori correcte i pugui obrir els database files. Si s'ha pogut recuperar tota la informació ja que no s'han modificat els arxius, només el lloc on es van a buscar.

Per configurar la base de dades perquè pugui recuperar els arxius que s'han mogut hem canviat l'estat de la base de dades de la següent forma.

```
ALTER DATABASE RENAME FILE '/u01/app/oracle/oradata/EE/undotbs01.dbf'  
TO '/u03/tbs/undotbs01.dbf';  
ALTER DATABASE RENAME FILE '/u01/app/oracle/oradata/EE/system01.dbf'  
TO '/u03/tbs/system01.dbf';  
ALTER DATABASE RENAME FILE '/u01/app/oracle/oradata/EE/sysaux01.dbf'  
TO '/u03/tbs/sysaux01.dbf';  
ALTER DATABASE RENAME FILE '/u01/app/oracle/oradata/EE/users01.dbf'  
TO '/u03/tbs/users01.dbf';
```

Exercici 1b [opcional]. Fes els moviments de fitxers necessaris i aplica les configuracions necessàries dels SGBD per a que oracle-2 tingui una replica de tot el que teniu implementat a l'oracle-1. Encapsula totes aquestes operacions en un script.

Exercici 2. Programeu un script RMAN per fer una copia de seguretat. Feu la copia de seguretat de l'oracle de la màquina Oracle-2.

Per fer el backup de la màquina Oracle-2, primer ens connectem per ssh a la màquina, i llencem l'script amb la comanda: `rman @backup.rman`. El fitxer `backup.rman` conté la connexió a la base de dades i la comanda `rman` per fer el backup.

Fitxer `backup.rman`:

```
connect target sys/oracle;  
backup database;
```

Exercici 3. Ajudeu-vos de l'eina cron/crontab de linux per a programar còpies de seguretat periòdiques.

Feu el script que programa aquestes còpies de seguretat utilitzant el script RMAN de l'exercici anterior.

```
#!/bin/bash  
  
echo "10 0 0 0 0 oracle 'rman @PATH/scripts/backup.rman'" >> /etc/crontab
```

Exercici 4. Des de l'Oracle-2 anem a simular una pèrdua d'un datafile. Creeu un nou tablespace. Creeu una taula test dins el tablespace. Inseriu 20 registres dins la taula test. Elimineu el datafile corresponent al tablespace creat (prèviament feu una còpia) en el que es troben les dades de la taula test. Apagueu i inicieu oracle (com al exercici 1a). Que ha passat? Restaura la base de dades. Com ho heu fet? Heu pogut recuperar tota la informació?

Fes el mateix experiment però inserint 200.000 registres. Heu pogut recuperar tota la informació?

Perquè? On estaven guardades les dades que s'han pogut recuperar tenint en compte que la còpia de seguretat era prèvia?

Nota: Per inserir les dades a la taula test ajudeu-vos d'un procediment PL/SQL que automàticament faci els inserts corresponents.

A l'apagar l'oracle el que passa es que l'última copia de seguretat és la que estava feta offline de tota la base de dades. Per tant, totes les dades inserides es perdran ja que no s'ha instaurat cap punt de control sobre la base de dades.

Hem generat els scripts però al voler executar-los no ens permetia

Per fer-ho hem creat un tablespace amb una copia del datafile i hem creat una taula. Una vegada creada hem fet la copia de seguretat amb el script RMAN que la efectua. Finalment inserim els 20 registres amb valors i seguidament eliminem el datafile que estava assignat al tablespace, finalment amb l'altre script amb comandes SQLPLUS tirem el servidor d'oracle-2 amb SHUTDOWN ABORT i STARTUP. No s'han pogut recuperar les dades ja que el datafile del tablespace s'ha borrrat i no teniem cap punt de control per poder recuperar les dades d'aquell datafile.

```
[oracle@oracle-2 scripts]$ rman @backup.rman

Recovery Manager: Release 11.2.0.1.0 - Production on Thu Dec 20 21:43:42 2018

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

RMAN> connect target *
2> backup database;
3>
connected to target database: EE (DBID=1900685629, not open)

Starting backup at 20-DEC-18
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=193 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=/u03/tbs/sysaux01.dbf
input datafile file number=00001 name=/u03/tbs/system01.dbf
input datafile file number=00003 name=/u03/tbs/undotbs01.dbf
input datafile file number=00004 name=/u03/tbs/users01.dbf
channel ORA_DISK_1: starting piece 1 at 20-DEC-18
channel ORA_DISK_1: finished piece 1 at 20-DEC-18
piece handle=/u01/app/oracle/flash_recovery_area/EE/backupset/2018_12_20/o1_mf_nnndf_TAG20181220T214344_g1r38jvv_.bkp tag=TAG20181220T214344 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:45
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 20-DEC-18
channel ORA_DISK_1: finished piece 1 at 20-DEC-18
piece handle=/u01/app/oracle/flash_recovery_area/EE/backupset/2018_12_20/o1_mf_ncsnf_TAG20181220T214344_g1r39z34_.bkp tag=TAG20181220T214344 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 20-DEC-18

Recovery Manager complete.
```

Aquí en aquesta captura mostrem la connexió i l'execució del script RMAN on fem el backup de la base de dades.

En els dos casos de 20 registres i 200.000 registres es perden les mateixes dades que modifiquen la base de dades ja que al caure el servidor recupera l'últim estat offline, osigui d'un datafile més antic.

Exercici 5. Igual que a l'exercici 4 anem a simular una pèrdua d'un datafile però en aquest cas amb el ARCHIVELOG activat. Explica que passa si repetim el experiment de l'exercici 4 (feu directament el cas de 200.000 registres). Ha pogut recuperar totes les dades? On estaven guardades les dades que s'han pogut recuperar tenint en compte que la còpia de seguretat era prèvia?

```
--activar
SQL> alter database archivelog
SQL> alter database open

--Revisar archivelog activado
SQL> archive log list
```

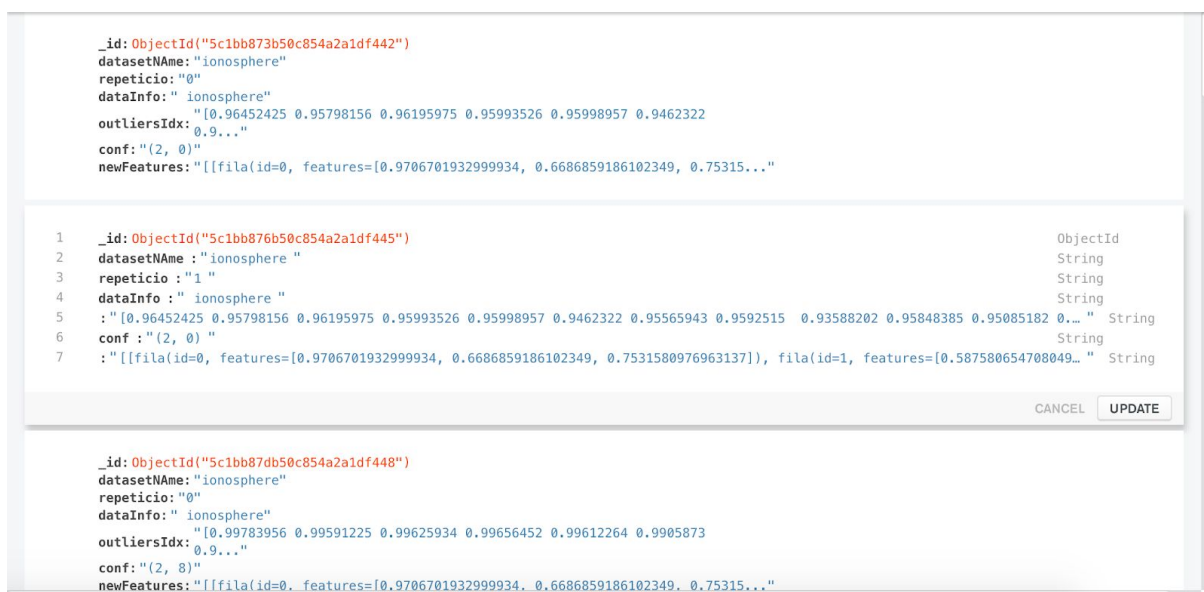
Una vegada activat el archivelog al nostre servidor el que ens produirà seràn punts de control quan modifiquem la base de dades aleshores es podràn recuperar les dades que insertim perquè tindrem un punt de control establert. El procediment és el mateix que hem seguit abans i ara les dades estaràn al nou arxiu que generem de archivelog per poder recuperar-les.

MongoDB

Per a aquesta segona entrega, hem seguit utilitzant el nostre codi pròpi per a la versió d'inserció realitzada a mongo. L'script base segueix essent el mateix, però en aquesta segona part hem realitzat diferents millores per a poder inserir el contingut que se'ns demanava en aquestes noves sessions. Alguns dels canvis que s'han fet permeten ara consultar els outliers associats a alguna de les bases de dades proveïdes o insertar-ne de nous. També s'han realitzat millores per a poder contenir també totes les dades necessàries per a poder representar els experiments realitzats.

Exemples:

Captura de pantalla associada als outliers generats:



Captura de pantalla associada als experiments:

```
_id: ObjectId("5c1bb873b50c854a2a1df444")
repeticio: "1"
method: "DMOD"
conf: "(2, 0)"
paramsMethod: "{ 'k': 1}"

> _id: ObjectId("5c1bb876b50c854a2a1df447")
repeticio: "0"
method: "DMOD"
conf: "(2, 8)"
paramsMethod: "{ 'k': 1}"

_id: ObjectId("5c1bb87db50c854a2a1df44a")
repeticio: "1"
method: "DMOD"
conf: "(2, 8)"
paramsMethod: "{ 'k': 1}"

_id: ObjectId("5c1bb885b50c854a2a1df44d")
repeticio: "0"
method: "DMOD"
conf: "(5, 5)"
paramsMethod: "{ 'k': 1}"

_id: ObjectId("5c1bb88ab50c854a2a1df44e")
```

Captura de pantalla associada als resultats obtinguts:

```
_id: ObjectId("5c1bb87db50c854a2a1df449")
AUC: "0.29281111111111113"
IDConfiguration: " ionosphere"
Database: "ionosphere"
idExperiment: "5"
tpr: "[0. 0. 0. 0.01 0.01 0.03 0.03 0.04 0.04 0.05 0.05 0.06 0.06 0.07..."
fpr: "[0. 0.00111111 0.00777778 0.00777778 0.01555556 0.01555556 0...."

> _id: ObjectId("5c1bb885b50c854a2a1df44c")
AUC: "0.29802222222222224"
IDConfiguration: " ionosphere"
Database: "ionosphere"
idExperiment: "6"
tpr: "[0. 0.01 0.02 0.02 0.03 0.03 0.04 0.04 0.05 0.05 0.06 0.06 0.07 0.07..."
fpr: "[0. 0. 0. 0.00555556 0.00555556 0.00777778 0...."

_id: ObjectId("5c1bb88ab50c854a2a1df44f")
AUC: "0.37691111111111111"
IDConfiguration: " ionosphere"
Database: "ionosphere"
idExperiment: "7"
tpr: "[0. 0. 0. 0.01 0.01 0.02 0.02 0.03 0.03 0.04 0.04 0.05 0.05 0.06..."
fpr: "[0. 0.00111111 0.00666667 0.00666667 0.02111111 0.02111111 0...."
```

Execució:

Per a poder executar tot el codi, s'ha de utilitzar la mateixa comanda que la versió anterior: `python main.py --drop --db VECT`. La seva execució tarda vora d'un minut degut a que s'ha d'inserir la informació cada vegada. El codi incorpora el fitxer `testOutliers` proporcionat pel professorat, però modificat per adaptar-se al disseny que ja teníem del nostre pròpi disseny de la base de dades que contava amb algunes diferències importants.

La diferència més important amb el disseny que es proporcionava per part del professorat, es la idea de que s'agrupen tots els vectors en la col·lecció que conté la informació dels diferents conjunts de dades dels que disposem. En més detall, en aquesta col·lecció els documents contenen un camp que s'anomena `vectors` que té una estructura interna del array, i necessiten d'un accés extra a memòria. En la pràctica, aquest nivell extra d'indexació no suposa un gran

impacte en el rendiment global del script ja que no provoca cap increment notable en el temps total que es necessita per a fer la consulta. Aquest disseny, per exemple implica canviar la forma en que les dades són llegides per l'algoritme que necessita fer el càlcul dels outliers.

Un altre canvi, es que utilitzem un segon client per a fer totes les operacions relacionades amb els outliers, ja que ja hem aixecat un per a poder tenir la consola interactiva. Aquesta consola interactiva, permet consultar informació molt genèrica de la base de dades, com per exemple els usuaris que estan inserits en aquesta. Segueix essent necessari l'execució dels diferents scripts per a la gestió d'usuaris i per testejar el seu correcte funcionament.

Sessió 10:

En la sessió 10 no s'havia de realitzar cap implementació en MongoDB.

Sessió 15:

En aquesta sessió, es demana que es realitzi l'inserció de les dades a mongoDB. Nosaltres ja vem realitzar aquesta inserció en la primera entrega, no obstant comentem breument les modificacions que s'han realitzat per a poder acollir les noves necessitats que han sorgit. Un comentari important, es que tot es guarda en format de cadena de text per a poder simplificar en general, i després si hem de convertir a arrays com en el càlcul dels outliers utilitzem un parser per a transformar-los en el format llista que es requereix.

Outliers:

DatasetName = Nom del dataset amb el que estem treballant.

Repetició = Indica quina es la repetició amb la qual s'han generat aquests outliers.

OutliersIdx = Conté informació sobre els vectors associats.

Conf = Conté informació sobre la configuració del experiment.

newFeatures= Conté informació associada a les característiques.

Results:

AUC = paràmetre d'evaluació de rendiment.

TPR = paràmetre d'evaluació de rendiment.

FPR = paràmetre d'evaluació de rendiment.

idExperiment = ID per a una relació posterior.

Database = Nom del dataset amb el que estem treballant.

Experiments:

Repetició = Indica quina és la repetició amb la qual s'ha generat aquest experiment.

Mètode = Mètode usat per a la generació del experiment.

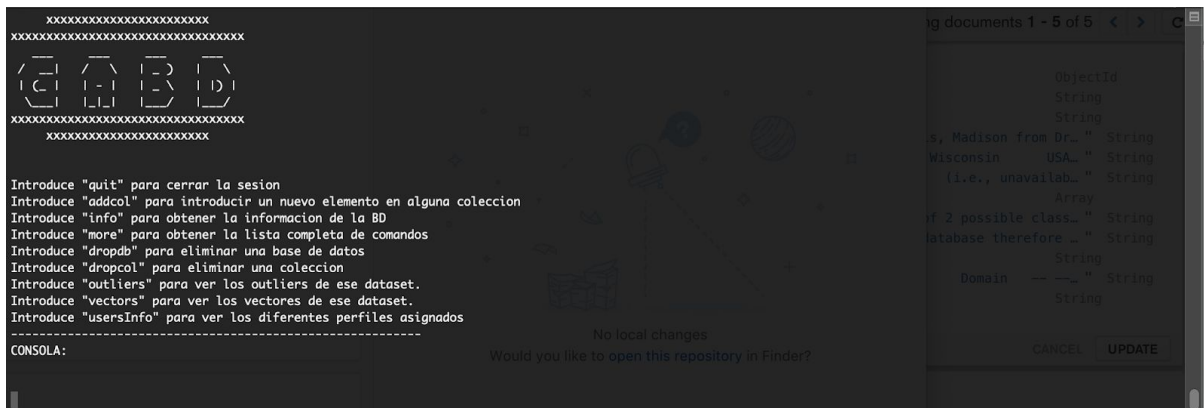
Conf = Configuració utilitzada en aquest experiment en particular.

ParamsMethod = Valor que prenen els paràmetre per aquest experiment en particular.

Sessió 17:

Pera aquesta sessió, hem realitzat unes petites modificacions a la consola interactiva que hi ha en el script main, que permet fer consultes dels vectors i outliers associats a un dataset determinat. Per a fer això hem de carregar l'script com fem normalment, i quan arribem a la pantalla de selecció de les diferents opcions, només hem de seleccionar l'opció vectors o bé els outliers. Un cop seleccionem l'opció introduïm el nom de la base de dades a consultar, i se'ns retornaran tots els vectors associats.

Captura del menú:



Sessió 17 Oracle

Exercici 0. Per fer aquest exercici no necessiteu res del projecte. Creeu una taula A amb la següent configuració: Create table A (id_seq number, conf varchar, nom varchar) amb una restricció "unique" a (conf , nom), i id_seq com a clau primaria. Feu un trigger (before insert) que afegeixi el camp id_seq amb el valor extret a partir d'una seqüència que generi enters consecutius. El trigger s'haurà d'executar al fer les següents insercions: Insert into A (conf,nom) values ('2-0','hola'); Insert into A (conf,nom) values ('2- 0','adeu'); Insert into A (conf,nom) values ('2-0','bonaTarda'). Feu el script corresponent per generar la taula, el trigger, la seqüència i els inserts.


```

1 0.
2 GRANT CREATE ANY TRIGGER TO AA_DESENVOLUPADOR;
3 GRANT CREATE ANY SEQUENCE TO AA_DESENVOLUPADOR;
4
5 CONNECT AA_DESENVOLUPADOR/123456;
6
7 DROP SEQUENCE id_seq_sequence;
8
9 CREATE SEQUENCE id_seq_sequence
10 INCREMENT BY 1
11 START WITH 1 ORDER
12 NOMAXVALUE;
13
14 DROP TABLE A;
15
16 CREATE TABLE A (
17     id_seq number Not Null Primary Key,
18     conf varchar2(255),
19     nom varchar2(255)
20 );
21
22 CREATE UNIQUE INDEX unique_A
23 ON A(conf, nom);
24
25 CREATE OR REPLACE
26 TRIGGER A_trigger
27 BEFORE INSERT ON A
28 FOR EACH ROW
29 BEGIN
30     :NEW.id_seq := id_seq_sequence.NEXTVAL;
31 END;
32 /
33
34 CONNECT sys/oracle as SYSDBA;
35
36 GRANT SELECT, INSERT ON AA_DESENVOLUPADOR.A TO AA_DETECTOR_OUTLIERS;
37
38 CONNECT AA_DETECTOR_OUTLIERS/12;
39
40 INSERT INTO AA_DESENVOLUPADOR.A (conf,nom) VALUES ('2-0','hola');
41 INSERT INTO AA_DESENVOLUPADOR.A (conf,nom) VALUES ('2-0','adeu');
42 INSERT INTO AA_DESENVOLUPADOR.A (conf,nom) VALUES ('2-0','bonaTarda');
43 select * from AA_DESENVOLUPADOR.A;

```

Exercici 1. Feu la creació de les noves taules amb el usuari developer. Utilitzeu el disseny E-R que veu implementar a l'exercici 3 de la sessió 15. Actualitzeu els privilegis de l'usuari "Detector outliers" per a que pugui inserir/actualitzar les dades sobre aquestes noves taules. Genereu un script per aquest exercici.

```

1.
--tables already created

GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.DATABASEINFO TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.RESULTS_IMAGES TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.IMAGES TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.SETUPEXPERIMENT TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.NEURALNETS TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.RESULTS_VECTOR TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.VECTOR TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.SETUPEXPERIMENTHASEXPERIMENT TO AA_DETECTOR_OUTLIERS;
GRANT INSERT, UPDATE ON AA_DESENVOLUPADOR.EXPERIMENTS TO AA_DETECTOR_OUTLIERS;

```

Exercici 2. Feu la funció PL/SQL loadVectorData. Aquesta funció haurà de retornar els vectors de característiques del data set que es passa per paràmetre. El retorn dels vectors de característiques es fa a partir d'un cursor.

Exercici 2

```
CREATE FUNCTION loadVectorData(my_dataSet IN VARCHAR2)
RETURN VARCHAR2 AS
IS
    v_test VARCHAR2;
    CURSOR c1
    IS
        SELECT ATTRIBUTEINFORMATION FROM DATABASEINFO WHERE TITLE = my_dataSet;
BEGIN
    OPEN c1;
    FETCH c1 INTO v_test;
    CLOSE c1;
    RETURN v_test;
END;
```

Exercici 3. Feu la funció PL/SQL loadVectorOutliers. Aquesta funció haurà de retornar els vectors de característiques del outliers definits per una configuració d'experiment, repetició i data set. El retorn dels vectors de característiques dels outliers es fa a partir d'un cursor.

Exercici 3

```
CREATE FUNCTION loadVectorOutliers(my_dataSet IN VARCHAR2, my_configuracio IN VARCHAR2, my_repetition IN VARCHAR2)
RETURN VARCHAR2 AS
IS
    v_test VARCHAR2;
    CURSOR c1
    IS
        SELECT ATTRIBUTEINFORMATION FROM DATABASEINFO
        WHERE TITLE = my_dataSet
        AND Configuracio = my_configuracio
        AND repetition = my_repetition;
BEGIN
    OPEN c1;
    FETCH c1 INTO v_test;
    CLOSE c1;
    RETURN v_test;
END;
```

Exercici 4. Feu la funció PL/SQL insertExperiment. Aquesta funció insereix la repetició dels experiments amb una configuració i retorna l'identificador de l'experiment inserit.

Exercici 4

```
CREATE FUNCTION insertExperiment(my_dataSet IN VARCHAR2)
RETURN int AS
IS
    v_test VARCHAR2;
    CURSOR c1
    IS
        SELECT ID FROM DATABASEINFO WHERE TITLE = my_dataSet;
BEGIN
    OPEN c1;
    FETCH c1 INTO v_test;
    CLOSE c1;
    RETURN v_test;
END;
```

Exercici 5. Feu el procediment PL/SQL insertVectorOutliers. Aquest procediment insereix un vector de característiques que ha sigut canviat a outlier.

Exercici 6. Feu el procediment PL/SQL insertResultats. Aquest procediment insereix els resultats generats per l'aplicació d'un mètode de detecció d'outliers.

ENTREGA 3

Sessió 22

Exercici 1. [Oracle] Feu 3 vistes Materialitzades en Oracle equivalents a 3 col·leccions de mongo (mostres, experiments, outliers). Utilitzarem aquestes vistes per analitzar les consultes del següent exercici. Feu tres vistes materialitzades que continguin la informació següent:

V1 Mostres: per cada data set veure totes les mostres associades. Els camps a mostrar són: nom data set, id de la mostra, vector de característiques de la mostra, label.

V2 Experiments: per cada experiment veure tots els mètodes que s'han aplicat i amb quins paràmetres. Els camps a mostrar són: configuració de l'experiment, repetició del experiment, nom del mètode, paràmetres del mètode i els seus valors.

V3 Outliers: per cada experiment veure tots els outliers associats. Els camps a mostrar són: nom del data set, configuració del experiment, repetició del experiment, id de la mostra a canviar, vector de característiques del outlier.

Creem les vistes materialitzades ja que així podrem tenir la informació sincronitzada de les diferents taules agregades. Script de la creació d'aquestes vistes: S22_Ejercicio1.sql

Exercici 2. [Oracle i mongo] Feu el codi SQL per a solucionar les consultes que us proposem. En el cas d'oracle utilitzeu les vistes materialitzades i en el cas de mongo les pròpies col·leccions. Per cada una de les consultes indiqueu quins índex (recordeu que els índex poden ser compostos) utilitzaríeu i per què i expliqueu les diferències en els plans d'execució entre fer la consulta amb índex (el seleccionat) o sense. Haureu de crear els índex que trieu i executar les consultes. Finalment feu una comparació entre oracle i mongo basant-vos en el temps i els plans execució en el cas d'utilitzar els índex seleccionats. Feu una taula per a cada consulta seguint el següent exemple:

Consultes Oracle:

Hem realitzat les 4 consultes sense índex i forçant índexs. Com no tenim les dades ben inserides no podem extreure valors exactes de temps, cost i ensenyar el pla d'execució. Però el resultat esperat amb les consultes que hem realitzat seria que totes les consultes tindran un temps d'execució molt similar entre elles. La diferència estarà més reflexada en el cost de la consulta, ja que les que utilitzen l'índex tindran un cost més baix, ja que es busca directament mitjançant aquests índexs.

Les 4 consultes están en el script S22_Ejercicio.sql

Consultes Mongo:

Consulta 1: Nom dels datasets en els que s'ha aplicat la configuració "2-0" per obtenir outliers.

```
db.OUTLIERS.find({"conf":"(2, 0)"},{datasetName:1, _id:0})
```

Consulta 2: Pel dataset letter-recognition quins labels diferents hi ha en les seves mostres.

```
db.DATABASES_INFO.find({"name":"letter-recognition",{differentLabels:1, _id:0})
```

Consulta 3: Per a cada dataset, nom de les classes, numero total de mostres.

```
db.DATABASES_INFO.find({classesName:2,total:1, _id:0})
```

Consulta 4: Configuració d'experiment que s'ha aplicat més cops pel mètode HOAD.

```
db.EXPERIMENTS.aggregate([{$match:{method:"HOAD"}},{ $group : { _id :{$metodo:"$method",cofiguracion:"$paramsMethod"},total : {$sum : 1}}},{ $sort:{total:-1}}]);
```

Creació index:

```
db.OUTLIERS.createIndex({conf:1})
```

```
db.DATABASES_INFO.createIndex({name:1,differentLabels:2})
```

```
db.DATABASES_INFO.createIndex({name:1,classesName:2})
```

```
db.EXPERIMENTS.createIndex({method:1,paramsMethod:2})
```

Consulta 1 Índex triats:conf Per què? Atribut sobre el qual fem la consulta		
Temps noíndex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
1ms	16ms	65536
Imatge pla execució no-índex mongo <pre>{ "queryPlanner" : { "plannerVersion" : 1, "namespace" : "VECT.OUTLIERS", "indexFilterSet" : false,</pre>		

```

    "parsedQuery" : {
      "conf" : {
        "$eq" : "(2, 0)"
      }
    },
    "winningPlan" : {
      "stage" : "PROJECTION",
      "transformBy" : {
        "datasetName" : 1,
        "_id" : 0
      },
      "inputStage" : {
        "stage" : "COLLSCAN",
        "filter" : {
          "conf" : {
            "$eq" : "(2, 0)"
          }
        },
        "direction" : "forward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "serverInfo" : {
    "host" : "MBP-de-Nil.home",
    "port" : 27017,
    "version" : "4.0.3",
    "gitVersion" : "7ea530946fa7880364d88c8d8b6026bbc9ffa48c"
  },
  "ok" : 1
}

```

Imatge pla execució index mongo

```

{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "VECT.OUTLIERS",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "conf" : {
        "$eq" : "(2, 0)"
      }
    },
    "winningPlan" : {
      "stage" : "PROJECTION",
      "transformBy" : {

```

```

        "datasetName" : 1,
        "_id" : 0
    },
    "inputStage" : {
        "stage" : "FETCH",
        "inputStage" : {
            "stage" : "IXSCAN",
            "keyPattern" : {
                "conf" : 1
            },
            "indexName" : "conf_1",
            "isMultiKey" : false,
            "multiKeyPaths" : {
                "conf" : [ ]
            },
            "isUnique" : false,
            "isSparse" : false,
            "isPartial" : false,
            "indexVersion" : 2,
            "direction" : "forward",
            "indexBounds" : {
                "conf" : [
                    ["(2, 0)", "(2, 0)"]
                ]
            }
        }
    },
    "rejectedPlans" : [ ]
},
"serverInfo" : {
    "host" : "MBP-de-Nil.home",
    "port" : 27017,
    "version" : "4.0.3",
    "gitVersion" : "7ea530946fa7880364d88c8d8b6026bbc9ffa48c"
},
"ok" : 1
}

```

Explicació de les diferències mongo.

Consulta 2

Índex triats: name, differentLabels

Per què? Atribut sobre el qual fem la consulta

Temps noíndex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
5ms	5ms	548148
Imatge pla execució no-índex mongo		
Imatge pla execució índex mongo		
Explicació de les diferències mongo.		

Consulta 3 Índex triats:name,classesNames Per què?Atributs sobre el quals fem les consultes		
Temps noíndex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
4ms	7ms	520288
Imatge pla execució no-índex mongo		
Imatge pla execució índex mongo		
Explicació de les diferències mongo.		

Consulta 4 Índex triats:method,paramsMethod Per què?Atributs sobre el quals fem les consultes		
Temps noíndex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
3ms	12ms	492152
Imatge pla execució no-índex mongo No es pot fer amb la funció aggregate		
Imatge pla execució índex mongo No es pot fer amb la funció aggregate		
Explicació de les diferències mongo.		

Exercici 3. [Oracle] Proposeu dues consultes i ompliu la taula del exercici 2 aplicant els diferents índex que us proposem sobre una vista materialitzada que també us proposem a continuació. Vista proposada: Per cada configuració d'experiment i mètode aplicat veure les mostres que s'han modificat (outliers). Els camps a mostrar són: configuració del experiment, repetició del experiment, nom mètode, paràmetres del mètode, id de la mostra modificada, nom del data set, vector modificat del outlier.

Important!: La creació de la vista i els índex pot durar uns minuts.

Primer creem una vista materialitzada amb tot el que es proposa a l'enunciat. Aquesta vista es pot trobar a l'inici de l'script següent: S22_Ejercicio3.sql. En aquest script també es poden trobar les consultes d'aquest exercici i la creació dels índex amb les consultes usant els índexs. De igual manera que a l'anterior exercici, al no tenir les dades ben inserides no podem extreure valors exactes de temps i cost ni els plans d'execució.

Índex proposats: 1) id_mostra, configuració, nom_mètode, repetició; 2) nom_mètode, id_mostra, configuració, repeticio; 3) configuració, repetició, nom_mètode, id_mostra

Consulta 1 Índex triats: Per què?		
Temps noíndex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
Imatge pla execució no-índex mongo		
Imatge pla execució índex mongo		
Explicació de les diferències mongo.		

Consulta 2 Índex triats: Per què?		
Temps no índex (mongo)	Temps Índex (mongo)	Espai índex (mongo)
Imatge pla execució no-índex Oracle		
Imatge pla execució índex Oracle		
Explicació de les diferències Oracle.		
Imatge pla execució no-índex mongo		

Imatge pla execució índex mongo
Explicació de les diferències mongo.

Sessió 23

Exercici 1. Els exemples que trobareu funcionen sense cap problema en sistemes gestors de MongoDB sense la autenticació activada. Com vosaltres l'hauríeu de tenir activada des del primer lliurament abans haureu de generar un fitxer de claus a la màquina `main.grupXX.gabd` i copiar-ho a les màquines `mongo-1.grupXX.gabd` i `mongo-2.grupXX.gabd`, on `XX` és el número del vostre grup.

Comandes:

```
[student@main ~]$ mkdir ssl
[student@main ~]$ openssl rand -base64 756 > ssl/mongodb.pem
[student@main ~]$ chmod 400 ssl/mongodb.pem
[student@main ~]$ ssh student@mongo-1.grup18.gabd 'mkdir ssl'
[student@main ~]$ scp ssl/mongodb.pem student@mongo-1.grup-18.gabd:ssl/mongodb.pem
```

Encenem ara mongo amb els flags activats de :

```
--keyFile ssl/mongodb.pem
--auth
```

Exercici 2. Fitxer de configuració2. Tots els flags que utilitzeu per executar el mongo es poden posar en un fitxer de configuració, `mongod.conf` que trobareu a la carpeta `/etc/`. Aquest fitxer és del root i com a usuari `student` no el podreu modificar. Feu una còpia d'aquest fitxer en el vostre `$home` del main i poseu-li el nom `standAlone.conf`. Editeu-lo i afegiu els flags que feu servir per executar el main. Atureu-lo de nou i inicieu-lo amb el flag: `--config standAlone.conf`. Feu el mateix per a `mongo-1` i `mongo-2`.

Comandes:

En la màquina main:

```
cp -f /etc/mongod.conf /home/student/standAlone.conf
```

En la màquina mongo-1:

```
cp -f /etc/mongod.conf /home/student/standAlone.conf
```

En la màquina mongo-2:

```
cp -f /etc/mongod.conf /home/student/standAlone.conf
```

Exercici 3. Replica set. Feu una còpia del fitxer `standAlone.conf` i poseu-li el nom `Master-Slave.conf`.

Modifiqueu-lo per a crear un replica set format pels nodes: `main.grupXX.gabd`, `mongo-1.grupXX.gabd` i `mongo-2.grupXX.gabd`. Feu les instruccions de `mongodb`

necessàries en tots 3 nodes per a configurar el replica set i crear una replica idèntica del que teniu al main.grupXX.gabd a mongo-1 i a mongo-2. Guardeu aquestes instruccions en un script de mongo. En cas que s'hagi d'executar instruccions a diferents servidors, feu un script per a cada servidor.

Per a poder crear correctament el replica set, el que hem de fer es generar cadascun dels diferents fitxers de configuració un per a cada node del que volem que disposi la nostre base de dades replicada. Un cop els hem generat obrim el fitxer mitjançant la comanda:

```
vim Master-slave.conf
```

i a sota de la línia on apareix el comentari "replication", afegim la següent línia:

```
replSet = GABD
```

un cop hem fet això per a cada una dels diferents fitxers de configuració.

El que hem de fer es reiniciar el nostre servei de mongo per a que els canvis afectin:

```
top  
kill 39 (o qualsevol altre pid associat al mongod)
```

Ara el que hem de fer es connectarnos a la base de dades de mongo i escriure:

```
use local  
rs.initiate()
```

I amb això quedarà encesa la nostre versió amb el mode rèplica.

Si volem inserir els nostres dos nodes extres al main (mongo-1, mongo-2), el que hem de fer es fer el mateix que hem fet abans. Un cop hem fet això, hem d'instanciar al node principal l'adició d'aquests dos nodes que hem afegit amb la següent comanda.

```
rs.add("mongo-1.grup18.gabd:27017")  
rs.add("mongo-2.grup18.gabd:27017")
```

I amb això ja quedaria configurat el nostre sistema de rèpliques, una bona forma de comprovar el bon funcionament d'aquest conjunt de rèplica.

Sessió 24

Exercici 1. Canvieu la configuració del replica set de la sessió anterior per a que el servidor del main, en comptes d'escoltar pel port 27017 escolti pel port 27018.

Per a poder canviar el port, simplement hem d'anar al fitxer de configuració i canviar l'ip per la que està escoltant. El fitxer a modificar es Master-Slave.conf
A l'apartat de network interfaces, hem de canviar el port de 27017 a 27018

Exercici 2. Traieu mongo-2 del replica set que heu creat en l'anterior sessió (Replicació) i torneu-ho a configurar com una màquina que funciona en mode Standalone pel port 27018. Per fer això últim a més dels canvis en el fitxers de configuració també haureu d'esborrar la base de dades "local". En previsió les tasques que haureu de fer més tard esborreu les bases de dades outliers i test. Feu un fitxer de configuració: mongo-2.conf amb les dades necessàries i executeu mongo amb aquest fitxer.

Per a canviar el mode standAlone de les màquines mongo-1 mongo-2, hem d'anar als fitxers standAlone de la mateixa manera que l'apartat anterior.

```
#network interfaces
net:
    port: 27018
```

Exercici 3. Reinicieu els servidors de main i mongo-1 com a shard. Feu els canvis necessaris a les configuracions, poseu-les en els fitxers main.conf i mongo-1.conf. Seguiu les indicacions de la secció **Restart the Replica Set as a Shard** del enllaç del tutorial que us hem indicat.

Per a poder resetejar el replica set com a sharding, el que hem de fer es reiniciar cadascun dels diferents nodes amb la versió —shardsvr, ja que això activa el mode sharding per a cadascun dels modes que hem d'afegir (mongo-1 i mongo-2)

Després hem d'obrir una nov terminal i executar la comanda rs.stepDown, el que assegura que no hi hagi cap client connectat a aquesta base de dades.

Després el que hem de fer es engegar de nou el sistema amb la versió —shardsvr activada en el nostre node que fa de principal (main).

Exercici 4. Creació del replica set pels config server. Poseu-li el nom configReplSet. Feu un nou fitxer de configuració pels servidors de configuració. Farem servir el mateix pels 3 servidors d'aquest replica set.

Definiu el dbpath: /home/Student/data/configdb (l'haureu de crear prèviament) i el port 27019.

Aixequem els servidors, creeu l'usuari sys amb contrasenya la que vulgueu a la base de dades

d'admin i doneu-li privilegis de root. Habiliteu l'autenticació fent servir els mateixos fitxers de claus que heu fet servir a la sessió anterior.

Feu els canvis en el fitxer de configuració per definir aquests servidors com a servidors de configuració. Haureu de definir el role del clúster com configsvr en la secció corresponent del fitxer de configuració. Reinicieu els servidors en els 3 nodes.

Anem al nostre node principal i creen el path /data/configdb que serà on estarà allotjada la nostre configuració per al nostre config server. Un cop hem fet això aixequem el nostre mongo amb les opcions :

```
—dbpath /home/student/data/configdb
```

...

Per a crear l'usuari, hem de connectarnos a través d'un client mongo i executar la següent comanda:

```
db.createUser(
{
  user: "sys",
  pwd: "sys123",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
}
)
```

Després per a configurar cada node com a servidor de configuracio, hem de crear una copia del fitxer Master-Slave que tenien amb la comanda.

```
cp Master-Slave.conf ./data/configdb/configReplSet.conf
```

(suposant que estem situats a /home/student/).

Un cop hem fet ixò, hem de fer:

```
cd data/configdb
vi configReplSet.conf
```

i a la part de sharding, afegim les 2 línies següents:

```
clusterRole: configsvr
configsvrMode: sccc
```

i amb això ja quedaria configurat el nostre sistema de server config.

Exercici 5. Al node main, haureu d'iniciar el servei de mongos amb la següent instrucció:
`mongos --configdb configReplSet/main.<grup>.gabd:27019,mongo-1.<grup>.gabd:27019,mongo-2.<grup>.gabd:27019 --bind_ip_all --fork -- logpath ./mongos.log --keyFile ssl/mongoddb.pem`
on `mongoddb.pem` és el fitxer de claus necessari per habilitar l'autenticació i que es troba a la carpeta `ssl`. Si heu fet servir un altre nom i ubicació haureu de posar la vostra.

Execució de la següent comanda al main, no fa falta tocar el path del fitxer de claus ja que s'ha creat al path que tocava (/home/student/ssl)

```
mongos --configdb configReplSet/main.grup-18.gabd:27019,mongo-1.grup-18.gabd:27019,mongo-2.grup-18.gabd:27019 --bind_ip_all --fork -- logpath ./mongos.log --keyFile ssl/mongoddb.pem
```

Exercici 6 Afegiu els shards. Aquesta operació es fa en el mongos que us haureu de connectar amb les credencials que heu creat en l'exercici 4. La instrucció que haureu de fer serà una cosa del estil:

```
mongo main.<YOUR GROUP>.gabd:27017/admin -u sys -p <YOUR_PASSWORD> -- authenticationDatabase admin
```

Haureu d'afegir els 2 shards. El format pel replica set main i mongo-1 i el format únicament per mongo-2. No us oblideu d'especificar els ports que correspon a cada servidor. Feu servir `rs.status()` per saber els ports i les màquines del replica set.

Execució de la comanda

```
mongo main.grup-18.gabd:27019/admin -u sys -p sys123 -- authenticationDatabase admin
```

Un cop hem fet això, el que necessitem es fer utilitzar la comanda:

```
rs.add("mongo-1.grup18.gabd:27018")  
rs.add("mongo-2.grup18.gabd:27018")
```

per a poder insertar de nou els dos nodes en el servidor de sharding. I per últim amb la comanda `rs.status()` veiem quin es l'estat actual del sistema.

Exercici 7. Encara en el mongos, habiliteu el sharding per les bases de dades "test" i "outliers". Afegiu la col·lecció Mostres i segui les instruccions de la secció Shard a Collection del tutorial que estem seguint

de referencia de la documentació de mongoDB a partir del punt 3.

Per a habilitar el sharding un cop hem fet tot això per a les bases de dades test i outliers, necessitem de la següent comanda:

```
sh.enableSharding("gabd.test")  
sh.enableSharding("gabd.outliers")
```

Apèndix
Distribució de tasques

	Distribució de tasques	Volum de feina feta
Marc	Oracle	30%
Nil	Mongo	40%
Oscar	Oracle	30%
Ibai	Abandonament	0%