**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. Nexe**

És el punt central del nostre script es fa la cirda 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:

1. 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 diferements 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.

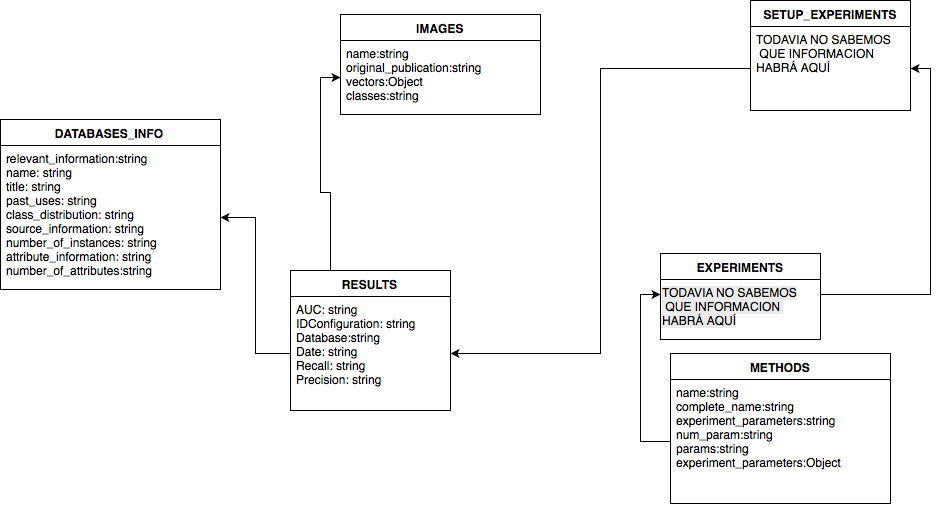
G. IMAGES

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 caraterístiques que estàn associats.

**1.2.6 Observacions en la generació:**

En la creació de la base de dades, ens hem asonat 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 tamanys 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.*

***TODO***

***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 relacionatts amb la supervisió de la base de dades.*

*La mida de les col·leccions de la base de dades, la podrem 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 comprobar 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:***

***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.*

**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.

**TODO: explicar por encima esas adaptacions**

**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 seguent:

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

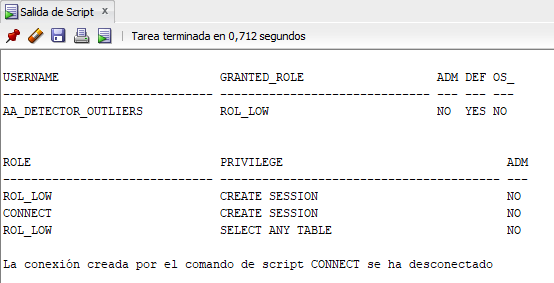
Adjuntem l’ script “SQL\_SECURITY.sql” el qual conté tota la creació de l’estructura dels usuaris.

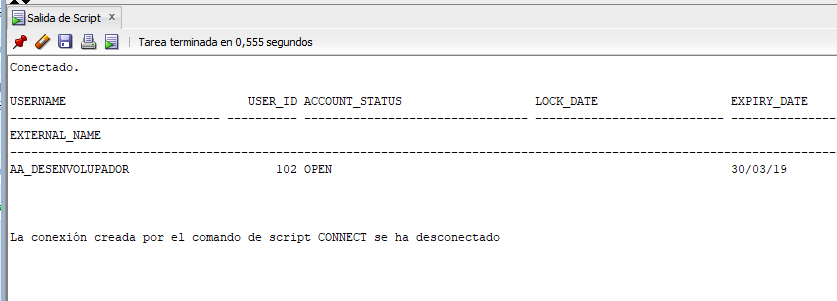
***2. Feu un script que mostri, per a cada usuari i rol que hageu creat, els privilegis assignats a cadascun d’ells. En cas que creeu algun perfil, el script que creeu també ho haurà de mostrar així com els paràmetres que hagueu configurat. Consulteu la documentació associada a aquesta sessió per 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.***

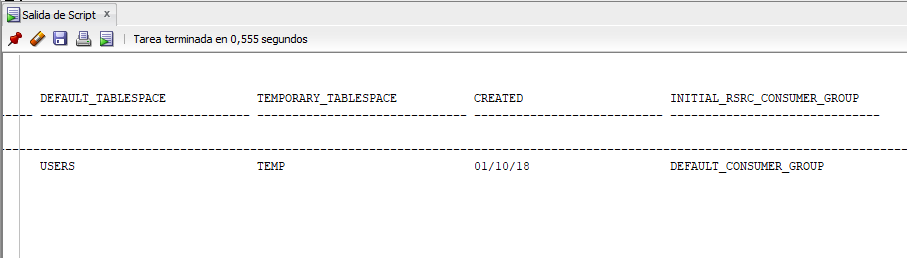
Fent la connexió amb l’usuari desitjat i consultant les taules cada usuari pot mostrar els seus privilegis, rols i paràmetres de perfil que té cadascun. Aquestes taules son: USER\_ROLE\_PRIVS;

ROLE\_SYS\_PRIVS i USER\_USERS.

A continuación podem veure la sortida on mostra tota la información demanada de l’usuari connectat.







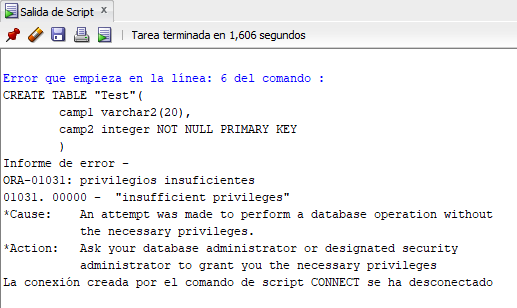
Adjuntem l’script corresponent aquest exercici amb el nom de “SHOW\_PRIVILEGES\_ROLES\_PROFILES.sql”.

***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 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 cambiar d’usuari està especificat a l’script amb 4 connects 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.



***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\_tablespaces, dba\_segments).***

--resultado en megas

--primer select devuelve total megas

--segundo select devuelve total megas libres

--la resta del total con los libres da los ocupados

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')

FROM dual;

***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.***

CREATE TABLESPACE PROVA

DATAFILE 'prova.dbf'

SIZE 5M;

CREATE TABLE Test (

camp1 varchar2(255),

camp2 number

)

TABLESPACE PROVA;

SELECT ROUND(sum(bytes)/1024/1024,0)

FROM dba\_free\_space

WHERE tablespace\_name='PROVA';

--esto 20 veces

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

INSERT INTO Test

(camp1, camp2)

VALUES ('a', 1);

SELECT ROUND(sum(bytes)/1024/1024,0)

FROM dba\_free\_space

WHERE tablespace\_name='PROVA';

drop tablespace prova INCLUDING CONTENTS CASCADE CONSTRAINTS;

drop table test purge;

***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***.

*--amb user sys*

*CREATE TABLESPACE outliersDB*

*SIZE 10M;*

*--amb user developer*

*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,*

*nombre varchar2(255),*

*descripcion varchar2(255),*

*Important\_Information varchar2(255),*

*Significate\_Atributes varchar2(255),*

*Past\_Uses varchar2(255),*

*Classes varchar2(255),*

*Citations varchar2(255),*

*Resources 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;*

*para hacer un drop de todo:*

*drop tablespace outliersDB INCLUDING CONTENTS CASCADE CONSTRAINTS;*

*drop table Results\_Vector purge;*

*drop table SetupExperimentHASExperiment purge;*

*drop table Experiments purge;*

*drop table SetUpExperiment purge;*

*drop table Vector purge;*

*drop table DatabaseInfo purge;*

*drop table Methods purge;*

*drop table Results\_Images purge;*

*drop table Images purge;*

*drop table NeuralNets purge;*

***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.***