Introducció

En aquesta pràctica treballarem la importació massiva de dades en Oracle utilitzant Python, les gestionarem i avaluarem l'impacte dels índexs en l'execució de consultes. Per la importació de dades s'haurà d'implementar un script de Python que permeti inserir les dades del repositori de la UCI[1]. Aquest repositori conté conjunts de dades de diferents tipus i característiques que s'utilitzen per avaluar mètodes d'aprenentatge computacional. Per la gestió de dades s'haurà d'adaptar el script de Python lliurat amb la pràctica per que sigui capaç de carregar el conjunt de dades, realitzar el experiment que s'especifiqui i guardar els resultats obtinguts a la base de dades. Finalment, amb tot el volum de dades generat, s'avaluarà l'impacte dels índexs en les consultes a fer.

Materials i Recursos

Per fer el lliurament d'aquest part disposeu dels següents materials i recursos:

- Aquest enunciat.
- Scripts Python de referència.
- Dades d'exemple de la UCI: Iris, Ionosphere, breast-cancer, letter-recognition.

A més, necessitareu els següents programes per a realitzar les pràctiques i monitoritzar els SGBD: SQLDeveloper, Pycharm (IDE per programar en Python). Caldrà que disposeu d'un client SSH per connectar-vos a les màquines de les pràctiques. A més a la màquina main del projecte teniu instal·lat un intèrpret de Python amb les llibreries Oracle Instant Client i els paquets bàsics per executar els vostres scripts. A més teniu instal·lada la comanda **tmux**[2], aquesta comanda permet obrir sessions en la terminal i recuperar-la en una nova sessió. És a dir, us permet entrar i sortir de la màquina main per ssh mantenint el que havia en la sessió anterior.

Sessió 1

Exercici 1. **Importació de Dades.** Implementeu el script en Python, insertData.py, que importi les dades de la UCI. Executeu el script amb el paràmetre –help per veure els paràmetres a passar en el moment de l'execució com es mostra en la imatge de sota des del vostre ordinador o la màquina main de la pràctica.

```
[student@main ~]  python3 src/insertData.py --help
usage: insertData.py [-h] [-C COLUMNCLASS] [--user USER] [--passwd PASSWD]
[--hostname HOSTNAME]
                     [--port PORT] [--serviceName SERVICENAME] [--
ssh_tunnel SSH_TUNNEL]
                     [--ssh_user SSH_USER] [--ssh_password SSH_PASSWORD]
[--ssh_port SSH_PORT]
                     datasetName fileName

This script insert data from the UCI repositori.
```

[1] Repositori UCI: https://archive.ics.uci.edu/ml/

[2] tmux: https://tmuxguide.readthedocs.io/en/latest/tmux/tmux.html

```
positional arguments:
  datasetName           Name of the imported dataset.
  fileName              file where data is stored.

optional arguments:
  -h, --help            show this help message and exit
  -C COLUMNCLASS, --columnClass COLUMNCLASS
                        index to denote the column position of class
label.
  --user USER           string with the user used to connect to the Oracle
DB.
  --passwd PASSWD       string with the password used to connect to the
Oracle DB.
  --hostname HOSTNAME   name of the Oracle Server you want to connect
  --port PORT           Oracle Port connection.
  --serviceName SERVICENAME
                        Oracle Service Name
  --ssh_tunnel SSH_TUNNEL
                        name of the Server you want to create a ssh tunnel
  --ssh_user SSH_USER   SSH user
  --ssh_password SSH_PASSWORD
                        SSH password
  --ssh_port SSH_PORT   SSH port
```

Aquest fitxer està preparat per el que pugueu executar tant des del vostre ordinador com des de la màquina main del projecte. En funció de la màquina des d'on l'executeu haureu de passar uns paràmetres de connexió o uns altres. Penseu que si executeu el script des del vostre ordinador haureu de passar els paràmetres per fer el túnel SSH. En canvi, executat des de la màquina main us podreu connectar a oracle-1.grupXX.gabd directament. Els paràmetres **datasetName** i **fileName** son, respectivament, el nom del conjunt de dades de la UCI a inserir i la ruta (relativa al script) d'on es troba el fitxer de dades a importar. Penseu en copiar les dades a la màquina main si voleu executar el script des d'allà. A més, el paràmetre **ColumnClass**, indica la posició de la classe de cada mostra. En general, aquesta acostuma ser la última columna (-1 en Python) però hi ha conjunt de dades que està en la primera. En funció de les dades que importeu ho haureu d'especificar.

Del fitxer inserData.py haureu d'acabar d'implementar la funció **insertVectorDataset**, aquesta funció ja rep les dades correctament instanciades a partir dels paràmetres que heu especificat a la crida del script d'inserció. Dins d'aquesta funció trobareu la crida a la funció **readVectorDataFile** com es mostra a continuació:

```
df,ids = readVectorDataFile(fileName,  label_pos=label_pos)
```

El paràmetre label_pos contindrà el valor del paràmetre **ColumnClass**. Els paràmetres de sortida són un Dataframe de Pandas amb les dades del fitxer i un diccionari ids on, per cada classe del conjunt de dades hi ha el índex de la mostra.

Per ajudar-vos a fer la inserció de dades, disposeu del script en sql **UCIExperimentsDB.sql** que conté un esquema bàsic de la BD a crear. Com es pot veure a la Figura 1, aquest esquema conté dos taules: Samples i Dataset. La taula Samples conté les dades a import de la UCI. El camp **Features**, de tipus BLOB, conté les característiques importades i codificades en aquest tipus de dades binari. El camp label, la classe a la que pertany. La taula Dataset conté la informació bàsica de cada conjunt de dades. Feat_size representa la dimensió dels vectors de característiques, numclasses el nombre de classes i **info** és un camp (opcional) de tipus JSON que permet introduir altres metadades informatives que pugui tenir cada conjunt de dades. Haureu de consultar tant a la documentació de OracleDB com a la de Oracle per veure com manipular aquests tipus de dades i com passar un diccionari de Python a un element de tipus JSON en Oracle 21c. Si executeu en local el script haureu de posar especial atenció a les versions instal·lades de les llibreries del Oracle Instant Client.

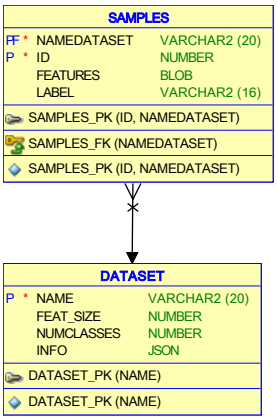

Figura 1. Esquema bàsic de la BD a crear.

Per inserir les dades a la taula Dataset haureu d'utilitzar el mètode **prepare** de oracleDB per optimitzar l'execució del codi i fer un bucle que recorri línia per línia el dataframe s'han carregat les dades llegides. Abans de fer la inserció haureu de tractar el vector de característiques i convertir-lo en un **BLOB**, d'un mode semblant a com heu tractat les dades de tipus JSON del camp info de la taula Dataset. Per convertir la llista de números a binari podeu fer servir el mètode **tobytes()** de numpy.

Un cop implementat el codi, haureu d'importar les dades dels següents conjunts de dades de la UCI: Iris, breast-cancer, ionosphere i letter-recognition. El script s'haurà d'executar amb un usuari amb nom: **GestorUCI** que haurà de tenir el rol de Gestor definit a la pràctica 1. Si us cal afegir més privilegis, en el rol Gestor (o Desenvolupador) els afegiu i ho comenteu en l'informe. El script s'haurà de poder executar tantes vegades com es desitgi. En cas que s'intentin inserir el mateix conjunt de dades 2 o més cops, el script ho haurà de controlar, finalitzant correctament, però sense inserir les dades duplicades. Tota l'execució del script s'ha de considerar com una sola transacció.

Exercici 2. **Disseny de la Base de Dades Completa.** En l'exercici anterior hem treballat amb una versió parcial de la base de dades. Aquesta només permet guardar els conjunts de dades de la UCI però no els resultats dels experiments que es faran amb el script **testExempleUCI.py**. En la Figura 2 trobareu el disseny Entitat-Relació de la base de dades amb els resultats dels experiments sobre els conjunts de dades de la UCI. Els atributs que apareixen en el disseny, són els que heu inserit en l'Exercici 1 i els que genera el script Python de exemple. En aquest exercici heu de fer la conversió al model relacional. Per cadascun dels atributs heu d'escollir el tipus més adequat. A l'hora de fer la conversió al model relacional tingueu en compte les següents consideracions:

- Fixeu-vos que el tipus de l'atribut **valors** de l'entitat paràmetres, en Python, és un tipus de dades complex (un diccionari). Si bé el oracle us permet definir atributs amb tipus JSON, els atributs d'aquest tipus no poden clau primària. A l'hora de trobar un atribut que sigui clau primària, l'ús de seqüències no és recomanable. En canvi, us podeu ajudar de la funció **hash** de Python per crear una signatura del diccionari que pugueu utilitzar com a clau primària.
- A l'hora de fer la conversió al model relacional podeu prendre decisions de disseny que redueixin el nombre de taules finals.
- Les decisions de disseny que preneu en els punts anteriors, així com qualsevol altre decisió que preneu per a fer la conversió haurà d'estar suficientment, i correctament, raonada per que es doni per vàlida la conversió proposada.
- Heu de generar el script sql, **BDUCI.sql**, que implementi la base de dades completes i executar-lo amb l'usuari **GestorUCI**.

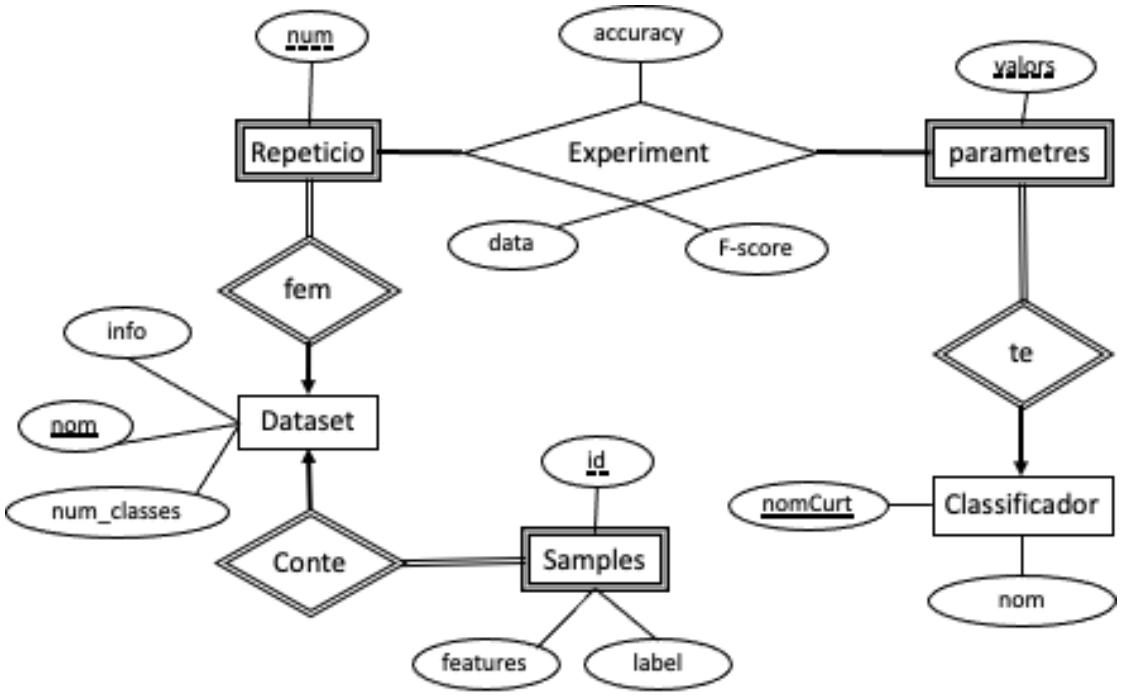

Figura 2. Disseny Entitat-Relació de la BD d'experiments.

Exercici 3. **Experiments**. A partir del script d'exemple que fa els experiments, feu un altre script de Python, **testUCI.py**, que es connecti a la base de dades on tingueu els datasets de la UCI i les recuperi. Un cop les hagi recuperat, haurà de fer els experiments i guardar-los en les taules que hagueu definit en l'exercici anterior a aquest efecte. Algunes consideracions a l'hora d'implementar aquest script:

- La interfície del script haurà de ser semblant a la que us donem pel script **inserData.py**. En concret els paràmetres opcionals de connexió a la base de dades hauran de ser els mateixos. Pel que fa als paràmetres posicionals també tindrà dos: el nom del data set i el nombre de repeticions. Podeu copiar els fragments que us interessi del insertData.py i adaptar-lo als nous paràmetres. No us oblideu d'esborrar els paràmetres opcionals del insertData.py que no siguin necessaris pel nou script.
- Haureu d'implementar una funció en Python que carregui les dades des d'oracle.
- Haureu d'implementar una funció en Python que escrigui les dades dels resultats a Oracle. Aquesta funció haurà de cridar, a una funció en PL/SQL que implementareu vosaltres mateixos i que serà l'encarregada de inserir les dades en l'estructura de dades que hagueu definit en l'exercici anterior. Aquesta funció s'implementarà amb un usuari d'oracle que tingui el rol Desenvolupador. El valor que retorni la funció haurà d'indicar si s'ha pogut inserir, o actualitzar, les dades dels experiments correctament. El script de Python l'haurà de recollir i tractar com correspongui.
- El script s'haurà d'executar amb un usuari amb nom **TestUCI**, aquest usuari tindrà el rol de test. A més dels privilegis propis d'usuaris amb aquest rol us assegurareu que aquest usuari no pot modificar les dades que estan en les entitats Dataset i Samples. En canvi haurà de poder inserir i modificar les dades dels experiments. No les podrà esborrar.
- Per cada experiment s'hauran de fer fins a 50 repeticions.

Sessió 2

Exercici 4. **Vistes i plans d'excisió**. Feu una vista materialitzada sobre els resultats que mostri per cada experiment, classificador i valors de paràmetres, la data del experiment, la mitja i desviació típica[3] de les mètriques de rendiment accuracy i f-score. El valor a mostrar per la vista serà el que es mostra en la taula següent:

Dataset	Classificador	Parametres	Accuracy	F-Score
Iris	SVM	{'kernel':'linear', 'gamma':10}	.98 +/- 0.1	.97 +/- 0.1

Un cop hagueu fet aquesta vista materialitzada, feu un altre vista que retorni, per cada conjunt de dades i classificador, el conjunt de paràmetres amb la millor accuracy (és a dir, amb el valor més alt). A l'hora de fer aquesta vista tingueu en compte les següents consideracions:

- Construïu la vista sobre els resultats de la vista materialitzada.
- Definiu indexs en la vista materialitzada i feu el pla d'execució de la consulta que implementa la vista amb i sense índexs. Comenteu les diferencies i comenteu les conclusions a les que arribeu.
- Les vistes les haureu de fer amb el mateix usuari GestorUCI, amb rol Gestor, que heu utilitzat per fer l'exercici 1.

Material a lliurar i criteris d'avaluació

La puntuació total d'aquesta pràctica serà sobre **10** punts. Caldrà lliurar un informe (format pdf) on es raoni la resolució de cada exercici, amb els scripts implementats per a resoldre'ls. Tot plegat haurà d'anar en un fitxer .zip. En l'informe caldrà evitar l'excés de captures, i totes les figures caldrà referenciar-les en el text. Es mirarà que els fitxers que s'han demanat s'han creat amb els noms indicats a l'enunciat i són fàcilment identificables. Si no es lliura algun dels scripts que es demana es puntuarà amb un 0 l'exercici on es demani. A més, caldrà lliurar una fitxa d'autoavaluació d'acord amb les puntuacions i descriptius que es mostren en la taula següent. En la fitxa d'autoavaluació caldrà indicar les fites que penseu que heu assolit en cadascun dels exercicis.

Exercici	Necessari per aprovar l'exercici	Necessari per obtenir puntuació màxima	Puntuació màxima
1	S'han inserit les dades dels datasets. Els scripts i procediments de PL/SQL es creen i/o executen amb l'usuari indicat i aquest te els privilegis que li correspon segons la definició que es va donar en la pràctica 1. Si es torna a executar el script d'inserció amb les dades ja inserides el comportament és	S'han inserit les dades dels datasets. Es modifica la funció readVectorDataFile per inserir les dades a Oracle a mesura que es va llegint el fitxer i no s'utilitza, en conseqüència, el DataFrame de pandas S'ha fet un bon ús dels mètodes de oracleDB i/o PL/SQL per fer la càrrega el més eficient possible. S'implementa un script auxiliar en sql per mostrar que els resultats s'han inserit	3

[3] Podeu utilizar la funció stddev d'oracle:
https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions159.htm

	l'esperat (no es dupliquen les dades, no finalitza amb error) Les explicacions de l'informe són mínimament convincents.	La presentació de l'informe en contingut i forma és molt bona.	
2	S'ha fet una conversió vàlida. Els scripts i procediments de PL/SQL es creen i/o executen amb l'usuari indicat i aquest te els privilegis que li correspon segons la definició que es va donar en la pràctica 1. Les explicacions de l'informe són mínimament convincents.	S'han pres decisions de disseny encertades que milloren el rendiment de la base de dades. Les decisions estan ben argumentades i són correctes. La presentació de l'informe en contingut i forma és molt bona.	1
3	S'han inserit les dades dels experiments. Si s'executa més d'una vegades el script d'experiment els resultats són els esperats (s'actualitza els valors de la relació "experiments) Els scripts i procediments de PL/SQL es creen i/o executen amb l'usuari indicat i aquest te els privilegis que li correspon segons la definició que es va donar en la pràctica 1. Les explicacions de l'informe són mínimament convincents.	S'han inserit les dades dels experiments. S'ha fet un bon ús dels mètodes de OracleDB i/o PL/SQL per fer la càrrega el més eficient possible. El codi PL/SQL està modulat i es fa un ús correcte de procediments i funcions. La presentació de l'informe en contingut i forma és molt bona.	4
4	S'implementen correctament la vista materialitzada i la vista encara que no hi hagi dades a mostrar. Els scripts i procediments de PL/SQL es creen i/o executen amb l'usuari indicat i aquest te els privilegis que li correspon segons la definició que es va donar en la pràctica 1. Les explicacions de l'informe són mínimament convincents.	S'implementen correctament la vista materialitzada i la vista i es mostren les dades. La definició d'índexs en la vista materialitzada és pertinent i està ben argumentada. L'anàlisi del pla d'execució de les consultes associades a les vistes és pertinent, així com les conclusions i decisions que se'n deriven. La presentació de l'informe en contingut i forma és molt bona.	2