

NOM ALUMNE:

DISSENY DE BBDD

Presentació

Aquesta pràctica segueix la temàtica de la primera pràctica de l'assignatura i pren com a punt de partida un disseny similar al de la solució proposada amb lleugeres variacions. A partir de la càrrega del model físic i d'un conjunt de dades sintètiques es demanarà fer manipulacions de dades i adaptar el disseny a l'SGBD específic per tal que les consultes, a més de retornar els resultats esperats, optimitzin els recursos. Amb l'objectiu d'assegurar aquest últim punt, s'han de saber analitzar les relacions entre les diferents taules, els volums de dades, i reformular les consultes o proposar, en cas de ser necessari, la creació d'índexs o altres tipus de millores.

Objectius i competències

La realització de la pràctica ha de demostrar que s'han treballat les competències següents i que s'han assolit els objectius que s'indiquen a continuació:

- Ser capaç de fer i manipular el disseny físic de la base de dades a partir d'un disseny lògic, adaptant-lo a les característiques d'un SGBD concret.
- Saber utilitzar les utilitats que incorpora Oracle per a realitzar càrregues de dades, tals com `sqlldr` i `imp`, i practicar la creació de taules i el seu poblament a partir de dades tipus 'bulk'.
- Saber definir els índexs necessaris i convenients en cada taula per tal que les aplicacions tinguin un bon rendiment quan accedeixen a la base de dades.
- Saber els mecanismes de processament i optimització de consultes.
- Ser capaç de fer ús de les vistes com a elements de disseny extern per tal de millorar el disseny de la base de dades.
- Saber l'abast dels mecanismes de seguretat d'una base de dades.



NOM ALUMNE:

Descripció de la pràctica

Exercici 1:

Heu de carregar a la BBDD l'arxiu t14.dmp adjunt a la pràctica. Per això primer haureu de crear l'usuari amb el que es va realitzar l'exportació. Si no es fes així, caldria atorgar permisos addicionals a l'usuari importador i l'operació seria més complexa.

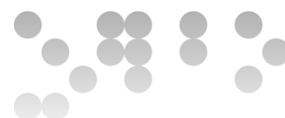
Es demana:

Creeu un usuari amb nom 't14'. Aquest ha d'utilitzar per defecte el tablespace 'users', ha de tenir quota il·limitada, i ha de tenir com a tablespace temporal l'anomenat 'temp'. Mostreu el SQL utilitzat per a crear l'usuari amb tots els requisits indicats:

```
CREATE USER t14  
IDENTIFIED BY abc1234  
DEFAULT TABLESPACE users  
QUOTA UNLIMITED ON users  
TEMPORARY TABLESPACE temp;
```

Atorgueu permisos a l'usuari perquè pugui crear sessions, taules, usuaris, vistes i seqüències. Mostreu l'SQL utilitzat:

```
GRANT  
CREATE SESSION,  
CREATE USER,  
CREATE TABLE,  
CREATE VIEW,  
CREATE SEQUENCE  
TO t14;
```



NOM ALUMNE:

Exercici 2:

Com a primer pas per al poblament de dades de les taules del disseny SmartCity, es demana crear una taula que permeti emmagatzemar les dades dels aparcaments existents a la ciutat.

Donat el disseny es provarà amb les dades d'una ciutat, que addicionalment és OpenData. Es demana descarregar les dades per a omplir la taula ParkingArea de l'adreça web:

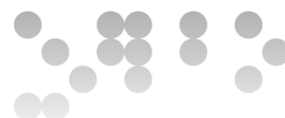
<http://opendata.bcn.cat/opendata/es/catalog/APARCAMENTS>

NOTA: Cal descarregar les dades en format CSV.

Es demana que creeu una taula anomenada ParkingArea i que carregueu a la mateixa el contingut del fitxer CSV descarregat.

La definició del camp a crear, segons la definició del fitxer, és la següent:

Arxiu: ParkingArea.txt				
Format: Variable, camps separats per punt i coma; ','				
Contingut: Dades dels aparcaments de Barcelona				
Nom del camp	Significat	Tipus original	Longitud	Pot ser nul?

**NOM ALUMNE:**

layerCode	Codi	Caràcter fix	4	SÍ
layerNameCat	Nom capa en Català	Caràcter	20	SÍ
layerNameCas	Nom capa en Castellà	Caràcter	20	SÍ
layerNameIng	Nom capa en Anglès	Caràcter	20	SÍ
coordX	Latitud	Número	10 (amb 3 decimals)	NO
coordY	Altitud	Número	10 (amb 3 decimals)	NO
Name	Nom	Caràcter	100	NO
phoneNumber	Número telèfon	Caràcter	9	SÍ
Address	Adreça	Caràcter	70	NO

Comentaris:

name no es repeteix i és la clau identificativa.

address no es repeteix.

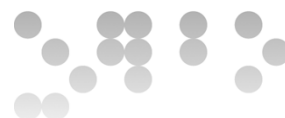
En unes coordenades X,Y únicament hi pot haver un aparcament.

Mostreu aquí la sentència SQL que utilitzeu per a crear la taula ParkingArea. Tingueu en compte tots els detalls indicats en la definició del fitxer per a crear les restriccions que siguin necessàries. Doneu nom a totes les restriccions. No utilitzeu la sentència ALTER TABLE per afegir restriccions a posteriori.

```
CREATE TABLE ParkingArea (
  layerCode CHAR(4),
  layerNameCat VARCHAR2(20 CHAR),
  layerNameCas VARCHAR2(20 CHAR),
  layerNameIng VARCHAR2(20 CHAR),
  coordX NUMBER(10,3) CONSTRAINT NN_coordX NOT NULL,
  coordY NUMBER(10,3) CONSTRAINT NN_coordY NOT NULL,
  name VARCHAR2(100 CHAR) CONSTRAINT PK_ParkingArea PRIMARY KEY,
  phoneNumber VARCHAR2(9 CHAR),
  address VARCHAR2(70 CHAR) CONSTRAINT NN_addressPA NOT NULL CONSTRAINT
  AK_addressPA UNIQUE,
  CONSTRAINT AK_coordXY UNIQUE(coordX, coordY)
);
```

Mostreu aquí el contingut del fitxer .ctl utilitzat per a carregar amb l'sqlloader les dades del fitxer CSV a la taula ParkingArea. Indiqueu en aquest fitxer que no es desitja carregar les dades existents a la primera línia del fitxer (descripció de les columnes).

```
OPTIONS (SKIP=1)
LOAD DATA
```



NOM ALUMNE:

```
CHARACTERSET 'UTF8'
INFILE 'c:\uoc\dbd\pra2\aparcaments-UTF8.CSV'
INTO TABLE ParkingArea
FIELDS TERMINATED BY ';'
TRAILING NULLCOLS
(
  layercode,
  layerNameCat,
  layerNameCas,
  layerNameIng,
  coordX ,
  coordY ,
  name,
  phoneNumber,
  address
)
```

NOTA: per assegurar que el tipus de caràcter (character set) del fitxer descarregat és UTF8, l'habitual en les instal·lacions de l'Oracle 11XE, carregueu les dades en un editor de text (bloc de notes de Windows, per exemple) i feu un 'save as', indicant que s'han d'emmagatzemar amb format UTF8.

Anoteu aquí els paràmetres utilitzats amb la instrucció sqlldr, per a carregar les dades segons el fitxer .ctl

```
sqlldr userid=t14/abc1234 control=C:\load_ParkingArea.ctl
```

Indiqueu el nombre de registres carregats a la taula ParkingArea:

Nom Taula		Número de registres
1	ParkingArea	406*

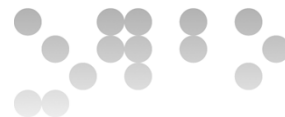
*segons comptador d'errors màxims establert a l'sqlloader

El següent pas serà carregar la resta de dades a partir d'un fitxer de backup. Per a fer la càrrega del fitxer .dmp s'ha d'utilitzar la sintaxi següent;

```
IMP usuari/contrasenya LOG=nomFitxer.log FILE=nomFitxer.dmp
```

Anoteu aquí la instrucció utilitzada per a fer la importació de les dades amb els paràmetres utilitzats en el vostre sistema:

```
IMP t14/abc1234 log=result.log file=t14_backup.dmp
```

**NOM ALUMNE:**

Es demana que anoteu el nombre de registres que conté cadascuna de les taules indicades en la graella següent:

	Nom Taula	Número de registres
1	Advantage	64
2	Bank	321
3	Card	142657
4	CreditCard	30501
5	Parking	12
6	Replaces	6
7	isAvailable	4760
8	Offers	3819403
9	SmartUser	63070

Per assegurar que l'SGBD té actualitzades les estadístiques i utilitza els recursos segons el contingut de les taules i l'existència d'índexs, caldrà executar la següent instrucció:

```
BEGIN
Dbms_Stats.Gather_Schema_Stats (
    ownname => 'T14',
    estimate_percent => 100);
END;
```

NOTA: És obligatori executar el procés d'actualització de les estadístiques (pot trigar una estona) cada vegada que es facin canvis d'importància en la BBDD, tal com població massiva de dades, creació/modificació de taules, creació d'índexs, ...

PUNT DE CONTROL: Abans de continuar, assegureu que en el procés d'importació de dades no han aparegut errors i que el resultat de la consulta:

```
SELECT COUNT(*) FROM ParkingArea, isAvailable WHERE ParkingArea.name = isAvailable.park;
```

indica que hi ha 4760 registres.



NOM ALUMNE:

Exercici 3

Apartat A

Comprovant el rendiment de les consultes més habituals que es poden fer a la BBDD, ens adonem que al consultar la taula isAvailable pel camp adv no obtenim els resultats esperats.

Mostreu el pla d'execució de la següent consulta:

```
SELECT * FROM isAvailable WHERE adv = '39';
```

Mostreu la captura de pantalla amb el pla d'execució:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		398	9
TABLE ACCESS (FULL)	ISAVAILABLE	398	9
Filter Predicates			
ADV=39			

Observem que no s'utilitza cap índex per no existir ni índex ni clau primària a la taula.

Per a subsanar l'error, es demana crear un clau primària composta pels camps adv i park.

A més es demana que:

Mostreu la consulta que utilitzareu per a comprovar quins són els duplicats que existeixen per la clau (adv, park).

```
SELECT adv,park, COUNT(*) FROM isAvailable GROUP BY (adv,park) HAVING COUNT(*)>1 ORDER BY COUNT(*) DESC;
```

Elimineu els registres duplicats. Mostreu la instrucció utilitzada.

```
DELETE FROM isAvailable WHERE adv = '28' AND park ='Aparcament Elizalde';
```

Mostreu la instrucció per a afegir la restricció de clau primària (ALTER TABLE ...) a la taula isAvailable, pels camps (adv, park). La restricció ha de tenir per nom PK_isAvailable.

```
ALTER TABLE isAvailable ADD constraint PK_isAvailable PRIMARY KEY (adv, park);
```



NOM ALUMNE:

Mostreu el pla d'execució que s'utilitzarà ara per a executar la consulta

```
SELECT * FROM isAvailable WHERE adv = '39';
```

Pla d'execució:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		398	4
INDEX (RANGE SCAN)	PK_ISAVAILABE	398	4
Access Predicates			
ADV=39			

Per a comprovar l'espai que ocupa l'índex en disc, s'ha d'executar la següent instrucció:

```
SELECT
index_name as "índex",
leaf_blocks AS "blocs plens -fulles-",
leaf_blocks*8 as "KB plens",
(leaf_blocks*8)/1024 AS "MB plens"
FROM User_Indexes
WHERE index_name IN ('PK_ISAVAILABLE');
```

Mostreu la captura de pantalla amb el resultat de la consulta:

	índex	blocs plens -fulles-	KB plens	MB plens
1	PK_ISAVAILABLE	28	224	0,21875

Apartat B

Revisades les estructures de les taules, s'observa que s'ha triat com a clau primària de la taula ParkingArea el camp name.

Creieu que és adequada la decisió? Argumenteu aquest fet i anoteu els avantatges o els inconvenients que aquesta decisió pot tenir sobre el rendiment de les consultes que es facin sobre aquesta taula.

La clau primària ocuparà molt d'espai, blocs de disc, ja que el camp name és de tipus VARCHAR2(100), i el seu contingut és habitualment llarg. És per això que comparativament a

**NOM ALUMNE:**

utilitzar una clau primària de mida molt reduïda, s'hauran de carregar més blocs de disc a l'hora de fer les consultes.

Sense tenir en compte les conclusions del punt anterior, es decideix que la clau primària de ParkingArea ha de ser una clau primària sintètica, pel que es demana fer algunes operacions per afegir el camp, crear una seqüència, canviar la taula isAvailable en conseqüència, i tornar-la a vincular.

Afegiu un camp anomenat id, tipus integer, a la taula ParkingArea.

Anoteu la instrucció utilitzada per a dur a terme aquest pas:

```
ALTER TABLE ParkingArea ADD (id INTEGER);
```

Creeu una seqüència amb la següent instrucció:

```
CREATE SEQUENCE seq_ParkingArea INCREMENT BY 1 START WITH 1;
```

Executeu la següent instrucció per a afegir el valor de la seqüència al camp id de ParkingArea.

```
UPDATE ParkingArea SET id = seq_ParkingArea.NEXTVAL;
```

Creeu ara la taula isAvail, poblada de dades, amb la següent instrucció:

```
CREATE TABLE isAvail AS  
  SELECT isAvailable.adv, ParkingArea.id  
  FROM isAvailable, ParkingArea  
  WHERE ParkingArea.name = isAvailable.park;
```

Analitzeu les restriccions i relacions existents de la resta de taules amb la taula isAvailable i desactiveu-les si és necessari.

```
ALTER TABLE isAvailable DROP CONSTRAINT FK_parkIA;
```

```
ALTER TABLE isAvailable DROP CONSTRAINT FK_idIA;
```

Elimineu la restricció de clau primària de ParkingArea, i afegiu com a nova clau primària el nou camp id. Creeu també una clau alternativa a ParkingArea, pel camp name.

Mostreu les instruccions SQL utilitzades:

```
ALTER TABLE ParkingArea DROP CONSTRAINT PK_ParkingArea;
```

```
ALTER TABLE ParkingArea ADD CONSTRAINT PK_ParkingArea PRIMARY KEY (id);
```

**NOM ALUMNE:**

```
ALTER TABLE ParkingArea ADD CONSTRAINT AK_ParkingArea UNIQUE (name);
```

Afegiu a isAvail les restriccions de clau primària i claus foranes amb Parking i ParkingArea.

```
ALTER TABLE isAvail ADD CONSTRAINT FK_idISAVAIL FOREIGN KEY (adv) REFERENCES Parking(id);
```

```
ALTER TABLE isAvail ADD CONSTRAINT FK_parkISAVAIL FOREIGN KEY (id) REFERENCES ParkingArea(id);
```

```
ALTER TABLE isAvail ADD CONSTRAINT PK_isAvail PRIMARY KEY (adv, id);
```

Mostreu una captura de pantalla amb el resultat d'una consulta que retorni el nombre total de registres que conté la taula isAvail:

	COUNT(*)
1	4758

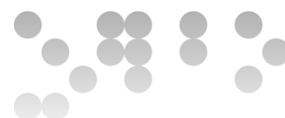
Ara consultarem l'espai ocupat pels índex de la taula ParkingArea i l'espai ocupat per la taula isAvail. Executeu les següent instruccions (donant els noms utilitzats allí on calgui):

```
SELECT
index_name as "índex",
leaf_blocks AS "blocs plens -fulles-",
leaf_blocks*8 as "KB plens",
(leaf_blocks*8)/1024 AS "MB plens"
FROM User_Indexes
WHERE index_name IN ('NOM_CLAU_PRIMARIA', 'NOM_CLAU_ALTERNATIVA');
```

Mostreu una captura de pantalla amb el resultat:

	índex	blocs plens -fulles-	KB plens	MB plens
1	AK_PARKINGAREA	3	24	0,0234375
2	PK_PARKINGAREA	1	8	0,0078125

```
SELECT table_name as taula, blocks as "blocs plens",
blocks*8 as "KB plens", (blocks*8)/1024 as "MB plens"
```



NOM ALUMNE:

```
FROM User_Tables
WHERE table_name IN ('ISAVAIL');
```

Mostreu una captura de pantalla amb el resultat:

	TAULA	blocs plens	KB plens	MB plens
1	ISAVAIL	11	88	0,0859375

Vist l'espai ocupat pels índex de la clau primària de parkingArea, de l'ara clau alternativa, i l'espai que ocupa l'índex de isAvail, com creieu que influiran aquestes diferències en el rendiment les consultes que es realitzin? Anoteu el motiu d'aquest fet, no us limiteu a dir 'anirà més ràpid' o 'anirà més lent',

La quantitat d'espai ocupat pels índex de la taula ParkingArea serà superior a l'inicial, ja que ara hi ha un índex per a la clau primària i un índex per a la clau alternativa. No obstant això, com que únicament s'utilitzarà un dels índex, segons accés, el rendiment serà habitualment millor, ja que al fer consultes on es relacioni la taula PakingArea amb Parking, s'utilitzarà el camp id, que ocupa molt menys espai, i per tant, hi haurà menys moviment de dades entre el subsistema d'emmagatzemament i la memòria.

Comparant l'espai ocupat pel índex de isAvail, amb l'anteriorment ocupat (apartat A del tercer exercici), s'observa que també ha disminuït molt l'espai ocupat, passa dels 28 als 11 blocs, cosa que corrobora que els accessos seran més ràpids.

Exercici 4:

Apartat A

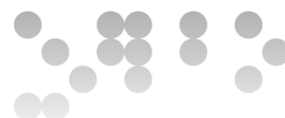
Els promotors dels sistema d'avantatges de l'SmartCity volen conèixer la quantitat d'avantatges a les que a l'actualitat pot accedir cada usuari registrat al sistema.

Per aquest motiu es desitja que feu una consulta que retorni el DNI, el nom, el cognom i el nombre d'avantatges que té disponibles cada usuari. La consulta ha de retornar els resultats ordenats per DNI, de menor a major, i per la quantitat d'avantatges disponibles, de més a menys.

Assegureu que el cost de la consulta sigui el més baix possible.

Mostreu la consulta SQL:

```
SELECT SmartUser.DNI, SmartUser.name, SmartUser.Surname, COUNT(*)
FROM Advantage, Offers, Card, SmartUser
WHERE
  Offers.adv = Advantage.id AND
  Offers.card = Card.DNI AND
```

**NOM ALUMNE:**

```

Offers.beginDate = Card.beginDate AND
Card.DNI = SmartUser.DNI AND
Card.ISACTIVE = 'Y'
GROUP BY SmartUser.DNI, SmartUser.name, SmartUser.SurName
ORDER BY DNI ASC, COUNT(*) DESC;

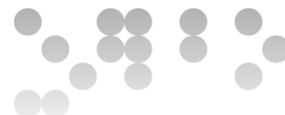
```

Mostreu una captura de pantalla amb com a mínim els primers 10 resultats retornats:

	DNI	NAME	SURNAME	COUNT(*)
1	X0000125F	Ermencinta Rocinau	Llabore	39
2	X0001088K	Octavi Sabucedo	Ludtke	39
3	X0001849C	Gedeon Sentanyes	Nunziante	43
4	X0002738Q	Lauren Romar	Lletxa	34
5	X0004066R	Zdenka Scarna	Marsinyach	37
6	X0004746D	Vicencio Ungaro	Provedo	37
7	X0004778M	Amabilio Vincke	Rigual	39
8	X0004970V	Corina Villeta	Ricondo	49
9	X0005412K	Amar Sanjulian	Mugarra	43
10	X0007620J	Gerardo Sojo	Micaela	27
11	X0007675A	Conxa Schermann	Martinalbo	23
12	X0007945K	Jamil Vicario	Begoner	23

Mostreu una captura de pantalla amb el pla d'execució i el cost de la consulta:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1219485	51568
SORT (ORDER BY)		1219485	51568
HASH (GROUP BY)		1219485	51568
HASH JOIN		1219485	9998
Access Predicates			
AND			
OFFERS.CARD=CARD.DNI			
OFFERS.BEGINDATE=CARD.BEGINDATE			
HASH JOIN		45549	522
Access Predicates			
CARD.DNI=SMARTUSER.DNI			
TABLE ACCESS (FULL)	CARD	45549	141
Filter Predicates			
CARD.ISACTIVE='Y'			
TABLE ACCESS (FULL)	SMARTUSER	63070	173
TABLE ACCESS (FULL)	OFFERS	3819403	3880



NOM ALUMNE:

Vist l'alt cost de la consulta, el volum de registres de cada taula, i en el cas que l'SGBD utilitzat permetés particionament de taules: proposaríeu particionar alguna taula? Si fos així, quina taula particionariu? Per quin conjunt de dades (camp) i per quin criteri proposaríeu fer la partició?

NOTA: No s'espera cap codi SQL ni prova d'execució com a resultat d'aquest apartat. Respongueu fent argumentacions teòriques, consultant, si és necessari, fonts d'informació addicionals.

Les particions serveixen per a dividir les dades d'una taula en varies parts, segons unes regles definides, amb la idea de que l'SGBD accedeixi a una o altra partició –del total on s'emmagatzema la informació de la taula–, segons calculi on han d'estar les dades. Amb aquest funcionament, per exemple, es poden separar les dades de facturacions històriques d'una companyia que generi milions de factures mensuals per mesos/anys, o en una empresa que gestioni milions de vehicles, les matrícules dels vehicles, per l'última xifra o lletra d'aquestes.

El factor de partició, o indicació de les dades que han d'anar en una altra partició és important, ja que s'ha de intentar que les dades estiguin clarament distribuïdes. Segons el motiu del particionament, que pot ser distribuir dades uniformement entre particions (per treballar amb molts milions de registres), o deixar les dades històriques en unes particions concretes, les dades s'hauran de distribuir uniformement entre les particions (a efectes cadascuna d'elles tingui una part proporcional de les dades) o s'hauran de col·locar segons un criteri d'antiguitat o data, per tal d'evitar l'accés continuat a particions que continguin dades històriques (les quals poden estar, fins i tot, en sistemes d'emmagatzemament més econòmics – discs durs de baixes prestacions).

Vist el funcionament de les particions, es podria optar (entre altres propostes) per separar les dades de la taula Card pel contingut del camp isActive per tenir les dades de les targetes actives (amb el 'Y' informat) en una partició, i les dades de les targetes no actives, en una altra. Com que habitualment es consultaran les dades de les targetes actives, l'SGBD faria un esforç molt menor a l'accedir a les dades, especialment en el casos en que hagués de fer un recorregut 'full' per la taula Card.

Apartat B

Ara desitgem consultar els usuaris que no tenen informat el camp correu electrònic per tal de demanar-los-el Optem per a executar la següent consulta:

```
SELECT * FROM SmartUser WHERE email IS NULL;
```

Vist el pla d'execució, constatem que es realitza un accés de tipus 'ACCESS FULL'.

Milloraria l'accés de crear un índex pel camp email? Creeu l'índex i anoteu el tipus d'accés que s'utilitza amb l'índex recent creat. Argumenteu els resultats obtinguts i/o les possibles millores del pla d'execució.

**NOM ALUMNE:**

El tipus d'accés no variarà, ja que els índexs no s'utilitzen en realitzar-se consultes on intervé la clàusula NULL.

Apartat C

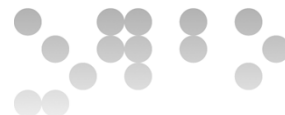
Ens diuen la consulta SQL per conèixer els usuaris que tenen el compte bancari de la seva tarjeta de crèdit en un banc concret no és gaire òptima, i ens proposen de crear un índex pel camp isActive.

```
SELECT * FROM SmartUser, Card, CreditCard, Bank
WHERE
  Card.DNI = SmartUser.DNI AND
  CreditCard.DNI = Card.DNI AND
  CreditCard.BANK = Bank.CODE AND
  Bank.Code = 'ES0073' AND
  Card.isActive = 'Y';
```

Mostreu la instrucció per a crear l'índex, el nou pla d'execució, i comenteu argumentadament els resultats obtinguts i el motiu o motius d'aquests (si n'hi pot haver més d'un).

Instrucció per a crear l'índex:
CREATE INDEX IDX_isActive ON Card(isActive);

Pla d'execució:



NOM ALUMNE:

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		2037	386
HASH JOIN		2037	386
Access Predicates			
CARD.DNI=SMARTUSER.DNI			
HASH JOIN		2037	213
Access Predicates			
CREDITCARD.DNI=CARD.DNI			
NESTED LOOPS		1426	71
TABLE ACCESS (BY INDEX ROWID)	BANK	1	1
INDEX (UNIQUE SCAN)	PK_BANK	1	0
Access Predicates			
BANK.CODE='E50073'			
TABLE ACCESS (FULL)	CREDITCARD	1426	70
Filter Predicates			
CREDITCARD.BANK='E50073'			
TABLE ACCESS (FULL)	CARD	45549	141
Filter Predicates			
CARD.ISACTIVE='Y'			
TABLE ACCESS (FULL)	SMARTUSER	63070	173

S'observa que després de crear l'índex l'SGBD no l'utilitza al fer les consultes. Aquest motiu és degut principalment al gran nombre de registres que compleixen el criteri `isActive = 'Y'`. L'optimitzador calcula que és més ràpid fer un recorregut de tota la taula, carregant la informació de tots els blocs en unes poques operacions de E/S, que no pas fer múltiples accessos per a comprovar l'índex i després accedir ordenadament a les dades de la taula.

La poca variació de valors que s'emmagatzemen en aquest camp 'Y' o 'N', també fa que l'optimitzador no consideri utilitzar índex per a l'accés.



NOM ALUMNE:

Exercici 5:

Els gestors del sistema d'avantatge SmartCity volen crear, per als bancs, un accés a les dades dels usuaris respectius de targetes de crèdit.

L'objectiu és que aquests puguin controlar que els càrrecs que reben corresponen als avantatges que realment tenen disponibles els seus clients.

Així, cal crear una consulta que, per a un banc donat i un usuari concret, retorni si té carnet actiu. La consulta ha de retornar el contingut del camp DNI, en el cas que l'usuari tingui una targeta activada. En cas contrari no retornarà res.

Mostreu la consulta amb les condicions per a què retorni els valors del banc ES8769, per l'usuari que té DNI 'X4960016P'.

```
SELECT *  
FROM Card, CreditCard, Bank  
WHERE  
    CreditCard.DNI = Card.DNI AND  
    CreditCard.beginDate = Card.beginDate AND  
    Card.IsActive = 'Y' AND  
    Card.DNI = 'X4960016P' AND  
    Bank.code = 'ES8769' AND  
    CreditCard.bank = Bank.code;
```

Creeu a partir de la consulta una vista anomenada V_ActiveSmartUser, on s'haurà de substituir el valor 'ES8769', per la variable del sistema 'USER'. No indiqueu en aquesta consulta la condició que limita el resultat a un usuari concret (camp DNI). Indiqueu que el camp retornat té àlies 'DNI'.

NOTA: Per a comprovar l'ús de la variable USER executeu la següent sentència SQL:

```
SELECT USER FROM DUAL;
```

Mostreu la sentència SQL per a la creació de la vista:

```
CREATE VIEW V_ActiveSmartUser AS  
SELECT Card.DNI AS DNI  
FROM Card, CreditCard, Bank  
WHERE  
    CreditCard.DNI = Card.DNI AND  
    CreditCard.beginDate = Card.beginDate AND  
    Bank.code = USER AND  
    CreditCard.bank = Bank.code AND  
    Card.ISACTIVE = 'Y'  
GROUP BY Card.DNI;
```




NOM ALUMNE:

Inicieu ara sessió amb l'usuari SYSTEM i creeu un nou usuari anomenat 'ES8769'. Doneu-li permisos per a iniciar sessió i per a consultar la vista V_ActiveSmartUser amb opció de donar permisos sobre aquesta vista.

Mostreu la instrucció per crear el nou usuari:

```
CREATE USER ES8769 IDENTIFIED BY abc1234;
```

Mostreu les instruccions per atorgar-li els permisos:

```
GRANT CREATE SESSION TO ES8769;  
GRANT SELECT ON t14.V_ActiveSmartUser TO ES8769 WITH GRANT OPTION;
```

Inicieu sessió amb el nou usuari ES8769. Des d'aquest usuari executeu les següents instruccions.

```
SELECT * FROM t14.v_ActiveSmartUser WHERE DNI = 'X4960016P';  
SELECT * FROM t14.v_ActiveSmartUser WHERE DNI = '61622070X';
```

Mostreu una captura de pantalla amb el resultat de cada consulta i argumenteu si són correctes i el motiu de les mateixes.

La primera consulta retorna:

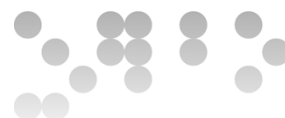
DNI

La segona consulta retorna:

DNI

Les dues consultes retornen el mateix resultat, donat no es compleixen les condicions pels DNIs anotats. En el cas de fer la consulta per a un DNI que sí fos client i estigués actiu, tal com el 'X1251676J', apareixeria aquest en el resultat, confirmant que és client del banc i està actiu.

FI DE LA PRÀCTICA



NOM ALUMNE:

Recursos

Per resoldre aquesta pràctica és necessari utilitzar els continguts dels mòduls 1 a 4, i part del 5, del material docent.

Criteris de valoració

El pes dels exercicis en la nota total de la pràctica és el següent:

- Exercici 1: 5%
- Exercici 2: 10%
- Exercici 3: 20%
- Exercici 4: 45%
- Exercici 5: 20%

Aquesta pràctica té un pes del 50% en la nota de pràctiques de l'assignatura.

Recordem que és obligatori realitzar la pràctica per aprovar l'assignatura.

Format i data de lliurament

El format del fitxer ha de ser Word o OpenOffice, i s'ha de lliurar una versió del mateix fitxer en PDF. Tot el codi SQL demanat s'ha de poder seleccionar per a copiar i poder provar (no han de constar com a captura de pantalla). **Indenteu (tabuleu) adequadament les consultes SQL per a fer-les el màxim de llegibles. Utilitzeu grafia Pascal pels noms de les taules i grafia Camel pels noms dels camps de les taules.** No envieu cap fitxer adicional als demanats (ni amb els SQL, ni amb les captures de pantalla, ...). Feu enviaments independents per a la versió doc/odt i el fitxer PDF.

El nom del fitxer tindrà el format següent:

DBD_PRA2_Cognom1_Cognom2_Nom.extensió

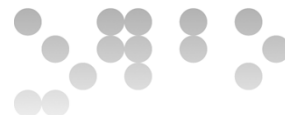
Els cognoms s'escriuran sense accents. Per exemple, un estudiant que es digui Alfredo García Melgar posaria el següent nom a l'arxiu:

DBD_PRA2_Garcia_Melgar_Alfredo.doc (u .odt) i .pdf

IMPORTANT: El nom i cognoms de l'estudiant també han d'aparèixer a l'espai reservat a la capçalera del document.

És responsabilitat de l'estudiant assegurar-se que els documents lliurats s'han pujat correctament i corresponen a l'activitat que cal presentar.

La data límit per lliurar la segona pràctica és dijous dia 11 de desembre.



NOM ALUMNE:

Nota: Propietat intel·lectual

Sovint és inevitable, en produir una obra, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis del Grau d'Informàtica, sempre que això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (CreativeCommons, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

Un altre punt a considerar és que qualsevol pràctica que faci ús de recursos protegits pel copyright no podrà en cap cas publicar-se en altres mitjans, a no ser que els propietaris dels drets intel·lectuals donin la seva autorització explícita.