

## PROPOSTA DE SOLUCIÓ

### PRÀCTICA 2 - 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 algunes variacions. Entre elles trobareu que s'han ampliat els atributs de les taules, i que s'han variat cardinalitats i relacions.

A partir de la càrrega del model físic i d'un conjunt de dades reals i 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.

#### Prerequisits

Executeu la instrucció següent a l'SQLDeveloper.

```
SELECT * FROM v$version;
```

Mostreu una captura de pantalla amb el resultat retornat.

```
Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production  
PL/SQL Release 11.2.0.2.0 - Production  
CORE 11.2.0.2.0 Production  
TNS for 32-bit Windows: Version 11.2.0.2.0 - Production  
NLSRTL Version 11.2.0.2.0 - Production
```

En el cas que el resultat de la línia on consta la versió de la 'Database', no sigui: **Oracle Database 11g Express Edition Release 11.2.0.2.0 – Production** i on consta la versió de 'PL/SQL' no aparegui **PL/SQL Release 11.2.0.2.0 – Production** haureu de revisar el fitxer origen utilitzat per a fer la instal·lació, desinstal·lar la versió actual, i tornar a fer la instal·lació amb la versió correcta.

No continueu amb la pràctica fins que no tingueu la versió XE (eXpress Edition) 11.2 **versió 32 bits** instal·lada, ja que els resultats de l'optimitzador poden diferir dels esperats i provocar que les consultes no s'executin com s'espera.



## PROPOSTA DE SOLUCIÓ

### Descripció de la pràctica

#### Exercici 1

##### Apartat A

Creeu un usuari amb nom 'T18'. 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 l'SQL utilitzat per a crear l'usuari amb tots els requisits indicats:

```
CREATE USER T18 IDENTIFIED BY T18
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON users;
```

Atorgueu permisos a l'usuari T18 per a que pugui crear sessions, taules, usuaris, vistes i vistes materialitzades. Mostreu l'SQL utilitzat:

```
GRANT CREATE SESSION, CREATE TABLE, CREATE USER, CREATE VIEW, CREATE
MATERIALIZED VIEW TO T18;
```

##### Apartat B

Seguidament es demana carregar les dades del fitxer de backup subministrat (T18\_backup.dmp). Per a fer la càrrega del fitxer .dmp s'ha d'utilitzar la sintaxi següent:

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

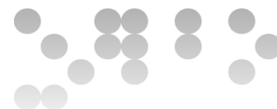
Anoteu aquí la instrucció utilitzada per a fer la importació de les dades amb els paràmetres utilitzats en el vostre sistema:

```
IMP T18/T18 LOG=import.log FILE=T18_backup
```

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

	Nom Taula	Número de registres
1	Tyre	169
2	ChargeStation	1457
3	VehicleUnit	142366

##### Apartat C



## PROPOSTA DE SOLUCIÓ

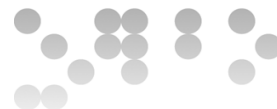
Es demana que creeu la taula 'Charge' segons la descripció que trobareu més avall, i que carregueu en aquesta el contingut del fitxer de text 'CHARGE.CSV'

La definició del fitxer és la següent:

<b>Arxiu:</b> CHARGE.CSV				
<b>Format:</b> Variable, separat per tabuladors				
<b>Codificació:</b> UTF8				
<b>Contingut:</b> Recàrregues que han realitzat un conjunt de vehicles entre dates				
Camp	Significat	Tipus original	Amplada	Pot ser nul?
idChargeStation	Identificador estació càrrega	Numèric		
idVehicleUnit	Identificador vehicle	Numèric		
chargeDate	Data càrrega	Data		
powerKwH	KwH carregats	Numèric	3 enters 2 decimals	NO
effectiveChargeTime	Temps de càrrega	Numèric		NO
totalPluggedTime	Temps ocupació lloc càrrega	Numèric		NO
odometer	Km que té el vehicle	Numèric		NO
kmFromLast	Km des d'última recàrrega	Numèric		
<b>Comentaris:</b> El camp idChargeStation s'ha de relacionar amb la clau principal de ChargeStation El camp idVehicleUnit s'ha de relacionar amb la clau principal de VehicleUnit La clau primària de la taula serà composta per idChargeStation+idVehicleUnit+chargeDate				

Mostreu l'SQL utilitzat per a crear la taula 'Charge'. Per a crear les restriccions que siguin necessàries, tingueu en compte tots els detalls indicats en la definició del fitxer. Doneu nom a totes les restriccions.

```
CREATE TABLE CHARGE
(
  IDCHARGESTATION INTEGER,
  IDVEHICLEUNIT INTEGER,
  CHARGEDATE DATE,
  POWERKWH DECIMAL(5,2) CONSTRAINT NN_Charge_PowerKwh NOT NULL,
  EFFECTIVECHARGETIME INTEGER CONSTRAINT NN_Charge_EffectiveChargeTime NOT NULL,
  TOTALPLUGGEDTIME INTEGER CONSTRAINT NN_Charge_TotalPluggedTime NOT NULL,
  ODOMETER INTEGER CONSTRAINT NN_Charge_Odometer NOT NULL,
  KMFROMLAST INTEGER,
  CONSTRAINT PK_Charge PRIMARY KEY (IDCHARGESTATION, IDVEHICLEUNIT, CHARGEDATE),
```



## PROPOSTA DE SOLUCIÓ

```
CONSTRAINT FK_Charge_ChargeStation FOREIGN KEY (IDCHARGESTATION)
REFERENCES CHARGESTATION(IDCHARGESTATION),
CONSTRAINT FK_Charge_VehicleUnit FOREIGN KEY (IDVEHICLEUNIT) REFERENCES
VEHICLEUNIT(IDVEHICLEUNIT)
);
```

Carregueu les dades del fitxer 'CHARGE.CSV' a la taula Charge. Mostreu el contingut del fitxer .ctl que heu fet servir per a carregar les dades amb l'sqlloader. Eviteu carregar el contingut de la primera fila (capçalera amb el nom de les columnes). No editeu el fitxer 'CHARGE.CSV' per a modificar el contingut!

```
OPTIONS(SKIP=1)
LOAD DATA INFILE charge.csv
APPEND INTO TABLE CHARGE
FIELDS TERMINATED BY '\t' OPTIONALLY ENCLOSED BY '"'
(IDCHARGESTATION, IDVEHICLEUNIT, CHARGEDATE, POWERKWH,
EFFECTIVECHARGETIME, TOTALPLUGGEDTIME, ODOMETER, KMFROMLAST)
```

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

```
sqlldr T18/T18 control=PRA2_charge.ctl log=charge.log
```

Indiqueu el nombre de registres carregats a la taula Charge:

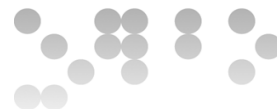
Nom taula		Número de registres
1	Charge	<b>816971</b>

**NOTA:** a efectes facilitar el desenvolupament de la pràctica, es pot considerar que existeix la limitació de fer una sola càrrega diària d'energia en les estacions de càrrega de TESMA (res impediria utilitzar hores en la data per a variar aquesta simplificació en la preparació de dades i resolució de consultes).

**IMPORTANT:** 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 instrucció següent:

```
BEGIN
Dbms_Stats.Gather_Schema_Stats (
    ownname => 'T18',
    estimate_percent => 100);
END;
```

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, com ara la població massiva de dades, creació/modificació de taules, creació d'índexs ...



## PROPOSTA DE SOLUCIÓ

### Apartat D

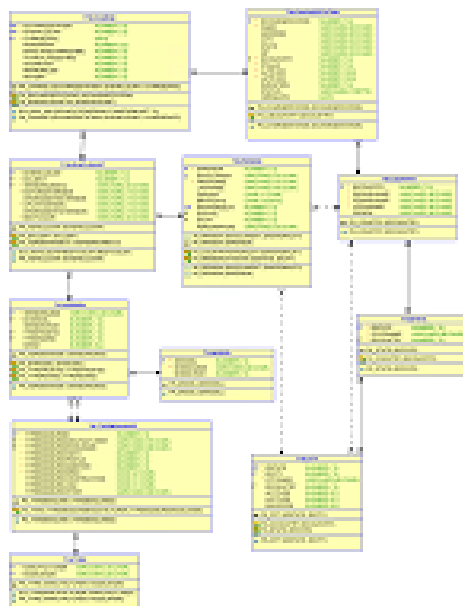
Atès que hi ha variacions en la BBDD dissenyada a la primera pràctica, es desitja obtenir de nou el disseny de les taules. Amb l'objectiu d'evitar l'actualització manual, i disposar de la versió gràfica ER, es proposa generar aquest diagrama per enginyeria inversa utilitzant les opcions que incorpora l'SQLDeveloper.

Seguiu els passos següents per a obtenir el diagrama actualitzat (suposarem que el teniu instal·lat en castellà) :

- 1) Des de l'SQLDeveloper menú 'Archivo', opció 'Data modeler', opció 'Importar', opció 'Diccionario de Datos'.
- 2) En la finestra de l'assistent, trieu l'usuari T18. Introduïu la contrasenya de connexió quan se us demani. Pitgeu el botó 'Siguiente'.
- 3) Quan se us demani triar un esquema, marqueu el 'T18'. Pitgeu sobre el botó 'Siguiente'. Quan us aparegui la finestra on es demani els objectes a importar, marqueu totes les taules. Pitgeu sobre el botó 'Siguiente' i després sobre el botó 'Terminar'.

Generat el diagrama ER, ordeneu mínimament les taules per a evitar els excessius encreuaments de les línies de les relacions.

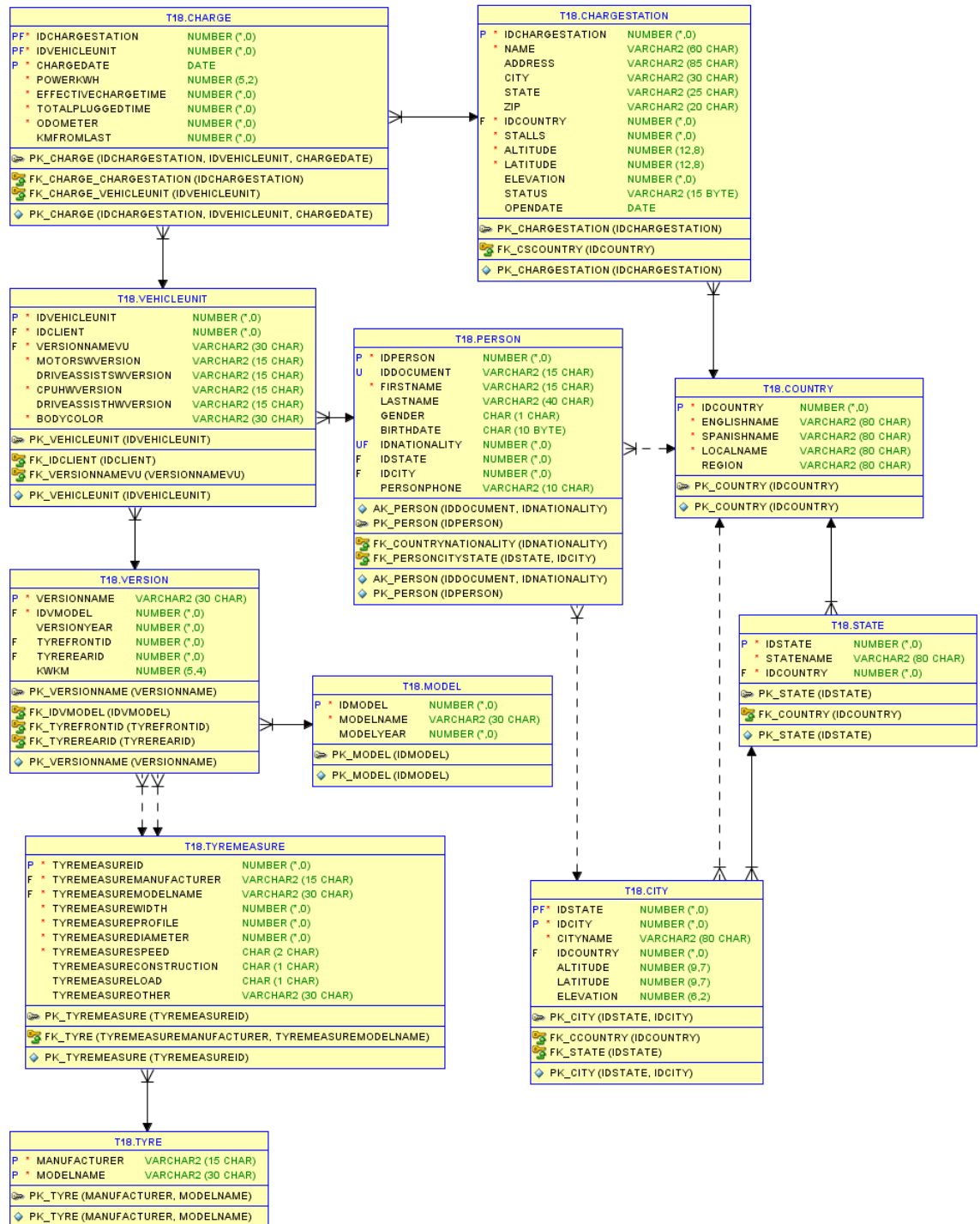
NOTA: Ha de quedar un diagrama ER similar al següent, però on es puguin veure tots els detalls!





# PROPOSTA DE SOLUCIÓ

Enganxeu aquí una captura de pantalla amb el diagrama obtingut:





## PROPOSTA DE SOLUCIÓ

### Apartat E

Revisant l'estructura i les relacions de les taules importades, s'observa que el model de pneumàtic per a cada eix (davanter i posterior) s'anota a l'entitat Version. El departament comercial de TESMA ens diu que volen que els clients puguin triar lliurement els pneumàtics del seu vehicle, pel que aquesta dada haurà de constar a partir de determinat moment a VehicleUnit.

És per això que es traspasarà la informació que consta actualment a Version a VehicleUnit, i es demanarà fer les variacions necessàries a les taules.

Es demana:

- a) Afegiu els camps tyreFrontId i tyreRearId, amb tipus INTEGER a la taula VehicleUnit. Mostreu les dues instruccions 'alter table'

```
ALTER TABLE VEHICLEUNIT
ADD TYREFRONTID NUMBER;
```

```
ALTER TABLE VEHICLEUNIT
ADD TYREREARID NUMBER;
```

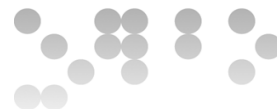
- b) Afegiu les restriccions de clau forana de tyreFrontId i tyreRearId de VehicleUnit a la taula TyreMeasure. Anomeneu a les noves restriccions FK\_VUTyreFrontId i FK\_REAR respectivament. Mostreu les instruccions SQL.

```
ALTER TABLE VEHICLEUNIT
ADD CONSTRAINT FK_VUTyreFrontId
FOREIGN KEY (TYREFRONTID)
REFERENCES TYREMEASURE(TYREMEASUREID);
```

```
ALTER TABLE VEHICLEUNIT
ADD CONSTRAINT FK_VUTyreRearId
FOREIGN KEY (TYREREARID)
REFERENCES TYREMEASURE(TYREMEASUREID);
```

- c) Mostreu la instrucció UPDATE que ha de servir per a poblar el tyreFrontId a VehicleUnit, a partir de les dades que hi ha per als pneumàtics davanters a Version.

```
UPDATE VEHICLEUNIT vu
SET
(TYREFRONTID) = (SELECT
    vr.tyrefrontid
FROM
    VERSION vr
WHERE
```



## PROPOSTA DE SOLUCIÓ

```

        vu.versionnamevu = vr.versionname
    )
WHERE EXISTS (SELECT VERSION.tyrefrontid
              FROM VERSION
              WHERE VERSION.versionname = vu.versionnamevu);

```

Repetiu el UPDATE per a poblar el camp tyreRearId de VehicleUnit.

```

UPDATE VEHICLEUNIT vu
SET
    (TYREREARID) = (SELECT
                    vr.tyrrrearid
                    FROM
                        VERSION vr
                    WHERE
                        vu.versionnamevu = vr.versionname
                    )
WHERE EXISTS (SELECT VERSION.tyrrrearid
                FROM VERSION
                WHERE VERSION.versionname = vu.versionnamevu);

```

- d) Una vegada els camps tyreFrontId i tyreRearId estiguin correctament poblats, s'han d'eliminar els camps corresponents de la taula Version (les FK corresponents s'esborraran automàticament). Mostreu les instruccions per a eliminar aquests dos camps (executeu-les).

```

-- Primer borro els 2 indexs
ALTER TABLE VERSION
DROP CONSTRAINT FK_TYREFRONTID;

ALTER TABLE VERSION
DROP CONSTRAINT FK_TYREREARID;

-- Segon borro les 2 columnes
ALTER TABLE VERSION
DROP COLUMN TYREFRONTID;

ALTER TABLE VERSION
DROP COLUMN TYREREARID;

```

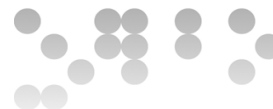
Per a comprovar la correcta execució de tots els passos, mostreu el resultat d'executar l'SQL següent:

```

SELECT
    VehicleUnit.idVehicleUnit,
    VehicleUnit.idClient,
    Front.tyreMeasureModelName FrontModel,
    Front.tyreMeasureWidth FrontWidth,
    Rear.tyreMeasureModelName RearModel,
    Rear.tyreMeasureWidth RearWidth

```





## PROPOSTA DE SOLUCIÓ

```
FROM
    VehicleUnit,
    TyreMeasure Front,
    TyreMeasure Rear
WHERE
    VehicleUnit.tyreFrontId = Front.tyreMeasureId AND
    VehicleUnit.tyreRearId = Rear.tyreMeasureId AND
    ROWNUM < 11
ORDER BY VehicleUnit.idVehicleUnit DESC;
```

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

	IDVEHICLEUNIT	IDCLIENT	FRONTMODEL	FRONTWIDTH	REARMODEL	REARWIDTH
1	185083	3160	PILOT SPORT 3	265	PILOT SPORT 3	285
2	185082	248272	PILOT SPORT 3	265	PILOT SPORT 3	285
3	185081	305613	PILOT SPORT 3	265	PILOT SPORT 3	285
4	185080	212312	PILOT SPORT 3	255	PILOT SPORT 3	275
5	185079	246628	PILOT SPORT 3	265	PILOT SPORT 3	285
6	185078	20141	PILOT SPORT 3	265	PILOT SPORT 3	285
7	185077	156617	PILOT SPORT 3	255	PILOT SPORT 3	275
8	185076	146619	PILOT SPORT 3	255	PILOT SPORT 3	275
9	185075	152522	PILOT SPORT 3	255	PILOT SPORT 3	275
10	185074	344122	PILOT SPORT 3	255	PILOT SPORT 3	275



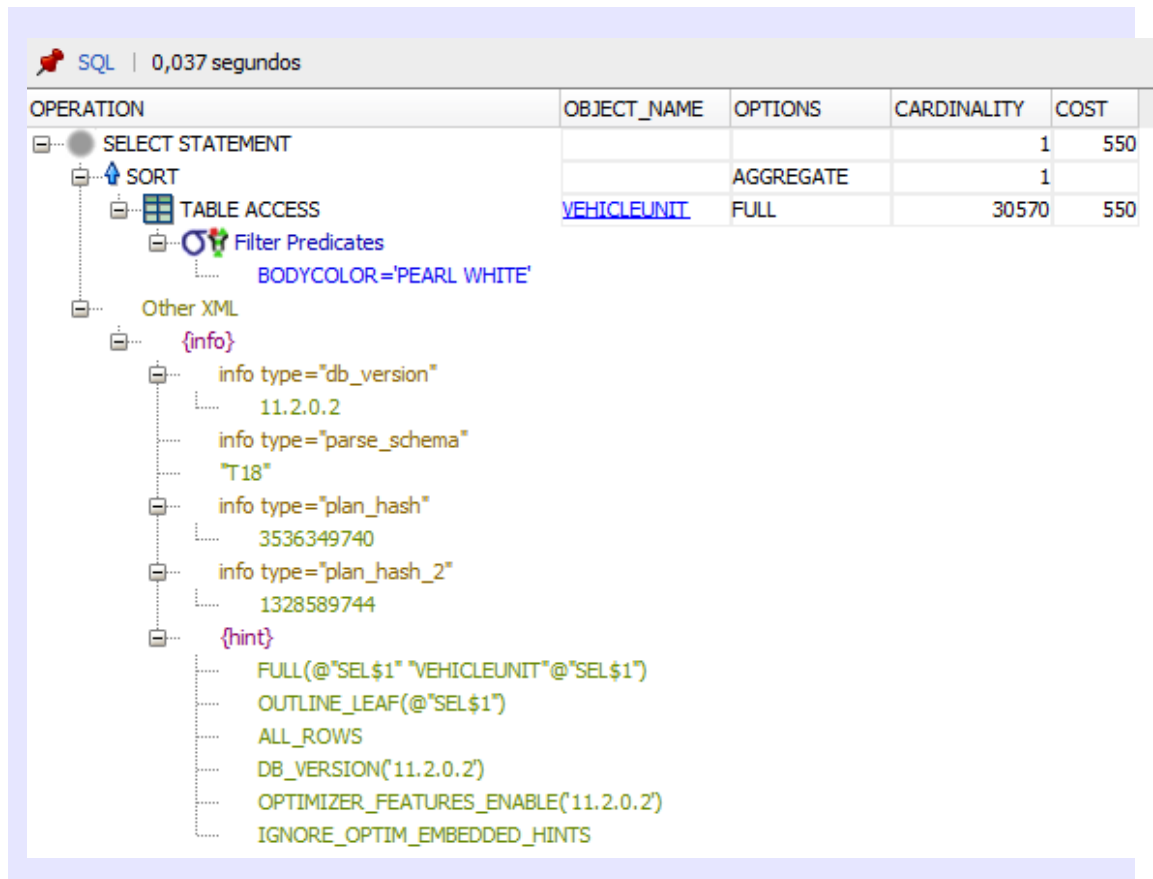
## PROPOSTA DE SOLUCIÓ

### Exercici 2

#### Apartat A

Executeu la consulta següent i mostreu-ne el pla d'execució (utilitzeu la tecla de funció F10 a l'SQL Developer per a obtenir l'esmentat pla d'execució):

```
SELECT COUNT(*) FROM VehicleUnit WHERE bodyColor = 'PEARL WHITE';
```



Amb l'objectiu de millorar el temps de resposta de la consulta, creeu l'índex que considereu oportú. Mostreu la instrucció SQL utilitzada i el nou pla d'accés resultant de tornar a executar la consulta després d'haver creat l'índex.

```
CREATE INDEX vu_bodyColor ON VehicleUnit(bodyColor) COMPUTE STATISTICS;
```



## PROPOSTA DE SOLUCIÓ

SQL   0 segundos				
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	92
SORT		AGGREGATE	1	
INDEX	VU_BODYCOLOR	RANGE SCAN	30570	92
Access Predicates				
BODYCOLOR='PEARL WHITE'				
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"T18"				
info type="plan_hash"				
3748969924				
info type="plan_hash_2"				
3257109578				
{hint}				
INDEX(@"SEL\$1" "VEHICLEUNIT"@"SEL\$1" ("VEHICLEUNIT", "BODYCOLOR"))				
OUTLINE_LEAF(@"SEL\$1")				
ALL_ROWS				
DB_VERSION("11.2.0.2")				
OPTIMIZER_FEATURES_ENABLE("11.2.0.2")				
IGNORE_OPTIM_EMBEDDED_HINTS				

### Apartat B

Continuant amb el procés d'optimització de les consultes, ens diuen que tot sovint es poden executar consultes similars a la següent:

```
SELECT * FROM VehicleUnit WHERE bodyColor = 'DEEP BLUE METALLIC';
```

Donat la consulta filtra pel camp bodyColor creeu un índex per aquest camp, si encara no ho heu fet.

Mostreu el pla d'execució que retornarà l'execució de la consulta, amb l'índex creat.



## PROPOSTA DE SOLUCIÓ

SQL | 0 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				13741 550
TABLE ACCESS	VEHICLEUNIT	FULL		13741 550
Filter Predicates				
BODYCOLOR='DEEP BLUE METALLIC'				
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"T18"				
info type="plan_hash"				
1810773069				
info type="plan_hash_2"				
945430054				
{hint}				
FULL(@"SEL\$1" "VEHICLEUNIT"@"SEL\$1")				
OUTLINE_LEAF(@"SEL\$1")				
ALL_ROWS				
DB_VERSION('11.2.0.2')				
OPTIMIZER_FEATURES_ENABLE('11.2.0.2')				
IGNORE_OPTIM_EMBEDDED_HINTS				

Que s'observa? S'utilitza l'índex creat? A què és degut? Quina diferència hi ha en el pla d'execució en comparació al de l'apartat A? A què és degut?

La consulta no fa servir l'índex creat. Això és degut a que fem una selecció (SELECT \*) de tots els camps de la taula. Llavors el motor d'execució de l'Oracle decideix que al no treure cap profit de l'índex es millor no fer-lo servir.

La principal diferència amb l'apartat A és que aquest sí que fa servir l'índex creat. Això es pel fet que en aquella query filtrem pel camp de l'índex i fem un agregat (COUNT). Llavor el motor d'execució d'Oracle sí que determina que pot fer servir l'índex per optimitzar la consulta.

### Apartat C

Ens diuen que tot sovint s'executaran consultes que impliquin recuperar el model del fabricant dels pneumàtics dels vehicles. Tal com la següent consulta bàsica.

```
SELECT * FROM Tyre WHERE manufacturer = 'BRIDGESTONE';
```

Per a intentar millorar el temps d'accés, es demana crear un índex pel camp manufacturer. Mostreu la instrucció per a la creació d'aquest índex:

```
CREATE INDEX tyre_manufacturer ON Tyre(manufacturer) COMPUTE STATISTICS;
```

Executeu la instrucció per a crear l'índex.

Mostreu el pla d'execució resultant d'executar la consulta mostrada:



## PROPOSTA DE SOLUCIÓ

SQL   0 segundos				
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				42 1
INDEX	PK_TYRE	RANGE SCAN		42 1
Access Predicates				
MANUFACTURER='BRIDGESTONE'				
Other XML				
{info}				
info type="db_version"				
info type="parse_schema"				
"T18"				
info type="plan_hash"				
3223887785				
info type="plan_hash_2"				
2845474955				
{hint}				
INDEX(@"SEL\$1" "TYRE"@"SEL\$1" ("TYRE", "MANUFACTURER" "TYRE", "MODELNAME"))				
OUTLINE_LEAF(@"SEL\$1")				
ALL_ROWS				
DB_VERSION("11.2.0.2")				
OPTIMIZER_FEATURES_ENABLE("11.2.0.2")				
IGNORE_OPTIM_EMBEDDED_HINTS				

S'utilitza l'índex recentment creat? A què és degut aquest fet? Argumenteu els motius.

No es fa servir l'índex recentment creat. Això és degut al fet que l'índex només l'he creat pel camp manufacturer. En canvi la PRIMARY KEY també incorpora un índex que si que inclou tots els camps de la taula (MANUFACTURER i MODELNAME). Al fer una consulta de tots els camps (SELECT \*) el motor d'execució determina que és millor fer servir el index PK\_TYRE

### Apartat D

Analitzant la taula ChargeStation ens adonem que no està ben relacionada amb la taula dels països i províncies/estats, donat si bé està relacionada amb Country, no ho està amb State.

Expliqueu textualment els problemes que pot comportar el fet que la taula ChargeStation tingui un camp textual per anotar la província/estat, i aquesta falta de relació (FK) amb la taula State.

Tal com es diu a l'enunciat el principal problema és la falta d'un identificador numèric de províncies/estats per relacionar una ChargeStation amb la província/estat on està situada. En ser un camp de text les consultes es veuran afectades de forma negativa. Quan fem una unió (JOIN) amb la taula d'estats per mostrar qualsevol dels altres camps d'aquesta taula la consulta es veurà afectada. La falta d'una FOREIGN KEY de la taula ChargeStation cap a la taula State provocarà un problema de rendiment en qualsevol consulta que inclogui una unió (JOIN) amb la taula State.

Per últim, els valors del camp State són de dos tipus diferents: les dues inicials de l'estat de l'adreça (si el tenim present), null si l'adreça no disposa d'estat i només



## PROPOSTA DE SOLUCIÓ

indica país i en alguns casos la ciutat sí l'adreça no disposa d'estat. Podem dir que l'entrada de valors d'aquesta columna és arbitrària i serà molt difícil poder-la vincular amb la taula State.

Per a solucionar aquests possibles problemes, anoteu textualment els passos que caldria dur a terme per a crear els vincles entre les dues taules. Assegureu d'anotar els passos seqüencialment i amb detall (traspàs de dades, creació restriccions, esborrat de dades...). Anoteu també els possibles problemes que poden sorgir al fer aquest procés de relació.

Sota el meu parer els passos a seguir són:

- 1.- Crear un nou camp anomenat IDSTATE. Aquest ha de permetre la inserció de valors NULLS (més endavant explicaré el motiu).
- 2.- Crear una clau foranea (FK) cap a la taula State vinculant el nou camp IDSTATE
- 3.- Per intentar traspasar les dades, primer hem de tenir en compte que la taula STATE només conté estats pels països: Espanya (199), Regne Unit (225) i dels Estats Units (226). Per últim només disposem de totes les provincies d'Espanya. De la resta només en tenim una part. Això fa que només puguem tenir un traspàs complet amb les provincies d'Espanya i que puguem intentar tenir un traspàs parcial dels altres dos països. Això implica la inserció de valors NULLs al camp IDSTATE.
- 4.- Fem una actualització de les files mitjançant una actualització amb consulta. La consulta es farà sobre la taula STATE on filtrarem pel valor del camp IDCOUNTRY de ChargeStation i pel camp STATE. Per les coincidències intentarem inserir el valor d'IDSTATE al camp IDSTATE de la taula ChargeStation.
- 5.- Esborrar la columna STATE.

NOTA: Només podrem relacionar una part molt petita dels estats.

### Apartat E

Ens demanen fer una consulta SQL que retorni els kWh carregats per Km recorregut, per a cadascuna de les recàrregues que consten a Charge.

Les columnes a retornar són idChargeStation, idVehicleUnit, chargeDate i el càlcul de kWh carregats per Km recorregut (powerKwH/kmFromLast). Mostreu aquesta última xifra utilitzant dos decimals. Afegiu un filtre a la consulta per a que sols retorni les càrregues que siguin de més de 1KwH per km recorregut. Mostreu el codi SQL.

```
SELECT
    idChargeStation, idVehicleUnit, chargeDate, ROUND((powerKwH/kmFromLast), 2)
    KWH_Km
FROM
    CHARGE
WHERE
    (powerKwH/kmFromLast) > 1;
```

Mostreu el resultat de la consulta:



## PROPOSTA DE SOLUCIÓ

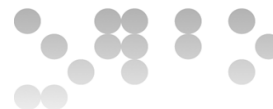
	IDCHARGESTATION	IDVEHICLEUNIT	CHARGEDATE	KWH_KM
1	706	68718	25/11/17	1,06
2	706	68718	26/11/17	1,19
3	706	68718	27/11/17	1,6
4	706	68718	30/11/17	1,1
5	712	124324	23/11/17	5,85
6	712	124324	24/11/17	4,94
7	712	124324	25/11/17	2,07
8	712	124324	26/11/17	1,03
9	712	124324	27/11/17	2,6
10	712	124324	28/11/17	2,92
11	712	124324	29/11/17	5,18
12	712	124324	30/11/17	3,73
13	696	93705	24/11/18	3,37
14	696	93705	25/11/18	2,35

Mostreu el pla d'execució de la consulta:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				51601 1280
TABLE ACCESS	CHARGE	FULL		51601 1280
Filter Predicates				
POWERKWH/KMFROMLAST>1				
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"T18"				
info type="plan_hash"				
109723134				
info type="plan_hash_2"				
3197809030				
{hint}				
FULL(@SEL\$1 "CHARGE"@SEL\$1)				
OUTLINE_LEAF(@SEL\$1)				
ALL_ROWS				
DB_VERSION('11.2.0.2')				
OPTIMIZER_FEATURES_ENABLE('11.2.0.2')				
IGNORE_OPTIM_EMBEDDED_HINTS				

Per a intentar millorar el temps de resposta ens proposen crear un índex calculat. Aquest índex s'ha de crear utilitzant el mateix càlcul que s'utilitza en la condició de la consulta, a efectes l'optimitzador detecti que és compatible amb la consulta.

Mostreu l'SQL utilitzat per a crear-lo:



## PROPOSTA DE SOLUCIÓ

```
CREATE INDEX KWW_per_KM_IDX
ON CHARGE (ROUND((powerKwH/kmFromLast), 2));
```

Mostreu el pla d'execució resultant de tornar a executar la consulta:

SQL | 0 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				51601 1280
TABLE ACCESS	CHARGE	FULL		51601 1280

Filter Predicates  
POWERKWH/KMFROMLAST>1

Other XML

{info}

- info type="db\_version"  
11.2.0.2
- info type="parse\_schema"  
"T18"
- info type="plan\_hash"  
109723134
- info type="plan\_hash\_2"  
3197809030

{hint}

- FULL(@"SEL\$1" "CHARGE"@"SEL\$1")
- OUTLINE\_LEAF(@"SEL\$1")
- ALL\_ROWS
- DB\_VERSION("11.2.0.2")
- OPTIMIZER\_FEATURES\_ENABLE("11.2.0.2")
- IGNORE\_OPTIM\_EMBEDDED\_HINTS

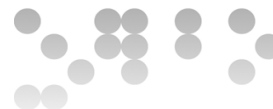
L'índex recentment creat s'utilitza? Si no s'utilitza, anoteu els possibles motius.

El índex creat recentment no s'utilitza. Això es degut al fet que en el filtre (WHERE) no estic fent servir la funció ROUND. Aquesta funció l'he feta servir alhora de crear l'índex calculat i ha fet que l'índex guardi els valors només amb dos decimals. Els càlculs són diferents i per això no es fa servir. Per fer servir l'índex calculat he de canviar la condició del WHERE i fer servir la funció ROUND.

```
SELECT
    idChargeStation, idVehicleUnit, chargeDate, ROUND((powerKwH/kmFromLast), 2)
    KWH_Km
FROM
    CHARGE
WHERE
    ROUND((powerKwH/kmFromLast), 2)> 1;
```

Ara sí es fa servir:





## PROPOSTA DE SOLUCIÓ

SQL | 0,009 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				40849 59
TABLE ACCESS	CHARGE	BY INDEX ROWID		40849 59
INDEX	KWW_PER_KM...	RANGE SCAN		7353 17

Access Predicates  
 $\text{ROUND}(\text{POWERKWH}/\text{KMFROMLAST}, 2) > 1$

Other XML

```

{info}
  info type="db_version"
    11.2.0.2
  info type="parse_schema"
    "T18"
  info type="plan_hash"
    3199314234
  info type="plan_hash_2"
    2528251552
{hint}
  INDEX_RS_ASC(@"SEL$1" "CHARGE"@"SEL$1" "KWW_PER_KM_IDX")
  OUTLINE_LEAF(@"SEL$1")
  ALL_ROWS
  DB_VERSION("11.2.0.2")
  OPTIMIZER_FEATURES_ENABLE("11.2.0.2")
  IGNORE_OPTIM_EMBEDDED_HINTS
  
```

### Apartat F

Ara ens diuen que la consulta de l'apartat E sols s'executarà cada final de mes, encara que en aquells moments, s'executarà moltes vegades. Trobaríeu adient crear una vista materialitzada amb una columna addicional amb aquest càlcul, i un índex sobre aquest camp de la vista materialitzada, a efectes optimitzar els accessos? Anoteu els possibles avantatges i els possibles inconvenients que tindria crear aquesta vista materialitzada.

Trobo adient fer servir una vista materialitzada per donar suport a la consulta que s'executarà moltes vegades a final de mes. Les principals avantatges de fer-la servir és que les dades de la consulta es trobarien emmagatzemades a la caché del Oracle. Tindriem una 'foto' de les dades. Al només executar-se a final de mes podríem tenir una foto d'aquells més. A l'executar-se moltes vegades l'Oracle no hauria d'executar moltes vegades la consulta. Faria servir la cache de quan es va crear la vista o de quan es va actualitzar.

L'inconvenient principal el tenim que una vista materialitzada és una 'foto' de la creació o de l'última actualització. Si es modifiquen els registres de la taula/es d'on surt la informació (inserció, modificació i/o esborrat) no és veurà reflectit fins que és torni a modificar. Encara que existeixen polítiques per poder actualitat la VM a cada commit.

Per a provar el rendiment d'aquesta proposta, creeu una vista materialitzada bàsica, amb nom VM\_Charge\_KwH\_Km (utilitzant els mínims paràmetres), que retorni les columnes idChargeStation, idVehicleUnit, chargeDate, powerKwH i



## PROPOSTA DE SOLUCIÓ

kmFromLast de Charge, més una columna amb nom kWh\_KM amb el resultat de l'operació  $\text{ROUND}((\text{powerKwH}/\text{kmFromLast}), 2)$ .

```
CREATE MATERIALIZED VIEW VM_Charge_KwH_Km
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
AS
SELECT
    idChargeStation, idVehicleUnit, chargeDate, powerKwH, kmFromLast,
    ROUND((powerKwH/kmFromLast), 2) kWh_Km
FROM
    CHARGE;
```

Creeu un índex, amb nom IDX\_VM\_Charge\_KwH\_Km, per la columna kWh\_Km;

```
CREATE INDEX IDX_VM_Charge_KwH_Km ON VM_Charge_KwH_Km(kwh_Km)
COMPUTE STATISTICS;
```

Mostreu el pla d'execució de la consulta:

```
SELECT idChargeStation, idVehicleUnit, chargeDate, kWh_Km
FROM VM_Charge_KwH_Km
WHERE kWh_Km > 1;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				42 51
MAT_VIEW ACCESS	VM_CHARGE_K...	BY INDEX ROWID		42 51
INDEX	IDX_VM_CHAR...	RANGE SCAN	8202	17
Access Predicates				
KWH_KM>1				
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"T18"				
info type="dynamic_sampling"				
2				
info type="plan_hash"				
2440584144				
info type="plan_hash_2"				
41796263				
{hint}				
INDEX_RS_ASC(@"SEL\$1" "VM_CHARGE_KWH_KM"@"SEL\$1" ("VM_CHARGE_KWH_KM", "KWH_KM"))				
OUTLINE_LEAF(@"SEL\$1")				
ALL_ROWS				
DB_VERSION("11.2.0.2")				

Comproveu les diferències en el cost d'execució, creareu aquesta vista materialitzada i l'índex addicional?



## PROPOSTA DE SOLUCIÓ

Per aquesta consulta no crearia la vista materialitzada i tampoc l'índex. Crec que la diferència és mínima. Aquesta vista implica un cost adicional i hem de mantenir la vista actualitzada al final de cada mes. A més, si no es fa correctament podem tenir problemes de files no actualitzades.

Proposeu un cas en que per a optimitzar una consulta sobre una dada calculada, no fos possible crear un índex calculat i s'hagués d'utilitzar la tècnica de creació de vista materialitzada i creació d'índex sobre un camp d'aquesta.

### Apartat G

Mostreu els índexs que existeixen per a la taula Version, adaptant la instrucció següent per a que retorni els índexs d'aquesta taula:

```
SELECT table_name, constraint_name, index_name
FROM all_constraints
WHERE table_name = 'TABLE-NAME';
```

Mostreu la consulta SQL:

```
SELECT table_name, constraint_name, index_name
FROM all_constraints
WHERE table_name = 'VERSION';
```

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

	TABLE_NAME	CONSTRAINT_NAME	INDEX_NAME
1	VERSION	FK_IDVMODEL	(null)
2	VERSION	PK_VERSIONNAME	PK_VERSIONNAME
3	VERSION	NN_VMODEL	(null)

### Apartat H

Revisant l'espai ocupat pels índexs, s'observa que PK\_versionName té un volum important per la quantitat de registres existents.

Personalitzeu la consulta següent per a que retorni els valors de l'índex de la clau primària d'aquesta taula:

```
SELECT
index_name as "Índex",
```



## PROPOSTA DE SOLUCIÓ

```
leaf_blocks AS "blocs plens -fulles-",
leaf_blocks*8 as "Kbyte plens",
(leaf_blocks*8)/1024 AS "MB plens"
FROM User_Indexes
WHERE index_name IN ('INDEX-NAME');
```

Mostreu la consulta i el resultat de la seva execució:

[Consulta]

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

[Execució]

index	blocs plens -fulles-	Kbyte plens	MB plens
1 PK_VERSIONNAME	1	8	0,0078125

Quines implicacions pot tenir en el rendiment de les consultes on es combini la taula Version amb altres, el fet de que versionName estigui definit com un VARCHAR2(30 CHAR)? Si poguéssiu fer variacions sobre la definició de la taula Version, faríeu alguna proposta de canvi per a millorar el rendiment d'aquest tipus de consultes?

Al ser un camp de tipus cadena pot ocasionar problemes de rendiment quan altres taules vulguin accedir al registre relacionat.

Jo afegiria un camp anomenat IDVERSION de tipus numèric que servir per identificar de forma única a una versió.

### Exercici 3

#### Apartat A

Per a fer el seguiment de les tasques del departament que gestiona els carregadors de TESMA, en relació a la prohibició d'ús dels carregadors per part d'aquells usuaris que han fet un abús de l'energia que aquests subministren de forma gratuïta (deixant al maleter ordinadors 'minant bitcoins', per exemple), es vol fer una consulta que retorni aquells usuaris que han tingut el vehicle estacionat als carregadors una mitjana de temps superior al 600% del temps mitjà de totes les operacions de càrrega que consten a Charge, i que



## PROPOSTA DE SOLUCIÓ

a la vegada, els kWh carregats per Km recorregut, sobrepassin la unitat.

Per a cada fila retornada, es desitja que es mostrin les columnes idChargeStation, idVehicleUnit, chargeDate, la mitjana de totalPluggedTime, versionNameVU (model del vehicle), i el firstName i el lastName de Person (propietari del vehicle)

Ordeneu els resultats per data de càrrega, de forma descendent.

Mostreu la consulta SQL:

```
SELECT
  ch.idchargestation,
  ch.idvehicleunit,
  ch.chargeDate,
  vu.versionnamevu,
  cl.firstName,
  cl.lastName
FROM
  charge ch,
  vehicleunit vu,
  person cl
WHERE
  vu.idvehicleunit = ch.idvehicleunit
  AND cl.idperson = vu.idclient
  AND ( (totalpluggedtime * 100 ) / (
    SELECT
      ROUND(SUM(TOTALPLUGGEDTIME) / count(*), 2)
    FROM
      CHARGE C
  )) > 600
  AND ROUND((powerKwH/kmFromLast), 2) > 1
ORDER BY
  ch.chargeDate DESC;
```

Mostreu una captura de pantalla amb el resultat d'executar la consulta SQL:

	IDCHARGESTATION	IDVEHICLEUNIT	CHARGEDATE	VERSIONNAMEVU	FIRSTNAME	LASTNAME
1	696	93705	25/11/18	MODEL 3 P75 Long Range	Mònica	Ciriano Donatelli
2	696	93705	24/11/18	MODEL 3 P75 Long Range	Mònica	Ciriano Donatelli
3	712	124324	30/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
4	706	68718	30/11/17	MODEL S 90D 4WD	Joan David	Llevat Garriga
5	712	124324	29/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
6	712	124324	28/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
7	712	124324	27/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
8	706	68718	27/11/17	MODEL S 90D 4WD	Joan David	Llevat Garriga
9	706	68718	26/11/17	MODEL S 90D 4WD	Joan David	Llevat Garriga
10	712	124324	26/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
11	706	68718	25/11/17	MODEL S 90D 4WD	Joan David	Llevat Garriga
12	712	124324	25/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
13	712	124324	24/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide
14	712	124324	23/11/17	MODEL 3 P50 Standard	Ana Maria	Balda Alcaide



## PROPOSTA DE SOLUCIÓ

### Apartat B

Per assegurar que els clients poden fer les càrregues d'energia sense esperes, es vol realitzar una consulta que retorni els dies que les estacions de càrrega ubicades a 'Espanya' han funcionat per damunt la seva capacitat. En concret a partir del 150% d'ús per a cada dia que hagi sobrepassat aquesta xifra.

La consulta ha de retornar les columnes idChargeStation, chargeDate i occupation, essent aquesta última xifra el percentatge d'ocupació calculat.

Es pot calcular el percentatge d'ocupació dividint la suma de recàrregues diàries d'una estació de càrrega, pel nombre de punts de recàrrega que té, multiplicat per 10 (que és la xifra que s'estima que pot acceptar un punt de càrrega sense que pateixi saturació). La següent fórmula ho exemplifica:

```
ROUND((COUNT(*)/(ChargeStation.stalls*10)*100),2)
```

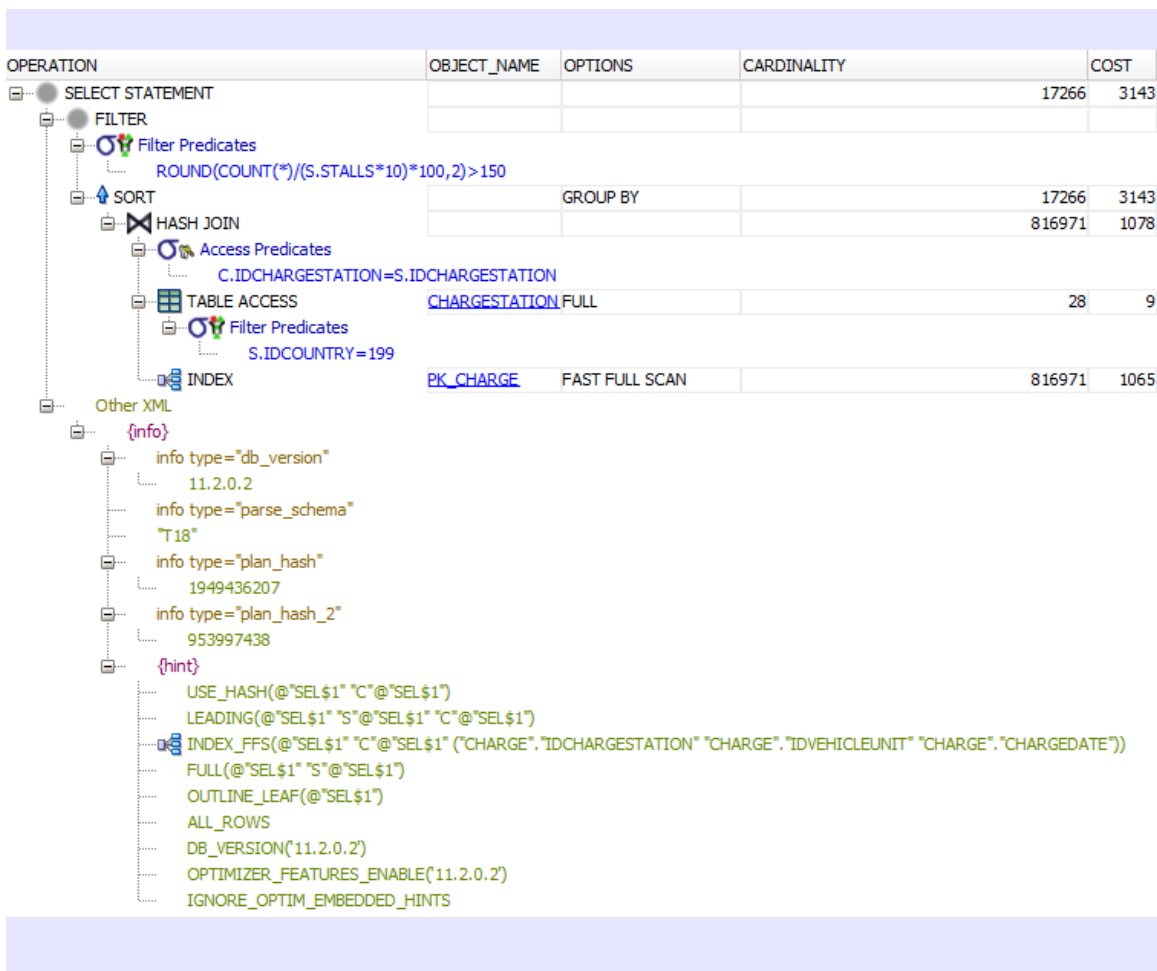
Mostreu la consulta SQL:

```
SELECT
  c.idchargestation,
  c.chargedate,
  ROUND((count(*)/(s.stalls*10)*100),2) occupation
FROM
  charge c,
  chargestation s
WHERE
  c.idchargestation = s.idchargestation
  AND s.idcountry = 199
GROUP BY
  c.idchargestation, c.chargedate, s.stalls
HAVING
  ROUND((count(*)/(s.stalls*10)*100),2) > 150
ORDER BY
  c.idchargestation, c.chargedate;
```

Mostreu una captura de pantalla que mostri el resultat d'executar la consulta:



## PROPOSTA DE SOLUCIÓ



### Exercici 4

Ens diuen que el departament comercial de TESMA desitja afegir a la intranet a la que els clients tenen accés, els llistat d'estacions de càrrega que són més properes a la ciutat en la que cadascun d'aquests clients resideix.

Per a emular aquesta operativa, caldrà que en primer lloc es creï una taula on poder fer constar l'identificador dels clients que accedeixen a la intranet per a realitzar consultes.

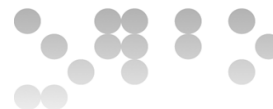
Creeu una taula anomenada WebClient amb els camps següents:

nick VARCHAR que pugui contenir fins a 15 caràcters  
 idWebClient INTEGER, no ha de poder contenir nulls  
 startDate data

NOTA: nick serà l'identificador únic (clau primària) de la taula.

Afegiu les restriccions que creieu necessàries, donant nom a totes les restriccions. Relacioneu idWebClient amb el camp que identifica les persones a Person.

Mostreu l'SQL per a la creació de la taula:



## PROPOSTA DE SOLUCIÓ

```
CREATE TABLE WEBCLIENT
(
  NICK VARCHAR(15),
  IDWEBCLIENT INTEGER CONSTRAINT NN_webclient_idwebclient NOT NULL,
  STARTDATE DATE,
  CONSTRAINT PK_webclient PRIMARY KEY (NICK),
  CONSTRAINT FK_webclient_person FOREIGN KEY (IDWEBCLIENT) REFERENCES
  PERSON(IDPERSON)
);
```

Inseriu en aquesta taula les dades següents:

bcorrea	19678	14/06/2018
fesquena	215899	19/07/2018

Mostreu les dues instruccions INSERT utilitzades:

```
INSERT INTO
WEBCLIENT (NICK, IDWEBCLIENT, STARTDATE)
VALUES ('bcorrea', 19678, '14/06/2018');
```

```
INSERT INTO
WEBCLIENT (NICK, IDWEBCLIENT, STARTDATE)
VALUES ('fesquena', 215899, '19/07/2018');
```

Creeu una consulta que mostri per a una persona determinada, segons les coordenades altitud i latitud, els punts de càrrega que té més propers, ordenats per proximitat, de més propers a més llunyans.

Mostreu les següent columnes en els resultats:

Id de l'estació de càrrega

Nom de l'estació de càrrega

Adreça de l'estació de càrrega

Ciutat que consta per a l'estació de càrrega

'altitude' de l'estació de càrrega

'latitude' de l'estació de càrrega

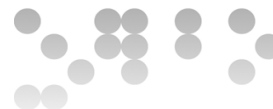
'elevation' de l'estació de càrrega

La quantitat de punts de càrrega que té l'estació (camp 'stalls')

El resultat de l'operació que calcula la distància *relativa* amb la ciutat de residència.

Per a fer un càlcul de distància *relativa* entre la ciutat de residència i una estació de càrrega (de forma bàsica i senzilla), resteu la 'altitude' de la ciutat de la





## PROPOSTA DE SOLUCIÓ

'altitude' del punt de càrrega. Feu el mateix amb la 'latitude' de la ciutat i la de l'estació de càrrega. Apliqueu la funció ABS() sobre els dos valors resultants (a efectes eliminar el possible signe negatiu), i sumeu aquests dos valors.

Per a cada 'ciutat i estació de càrrega' tindreu un valor que representa la distància entre els dos punts. Com més baix sigui el valor resultant, més propera estarà l'estació de càrrega de la ciutat.

Mostreu la consulta SQL. Feu que aquesta filtri les dades per a mostrar únicament les dades de l'usuari que té per nick 'bcorrea':

```
SELECT
  cs.idchargestation,
  cs.name as chargestationname,
  cs.address,
  cs.city,
  cs.altitude cs_alitude,
  cs.latitude cs_latitude,
  cs.elevation,
  cs.stalls,
  ABS(cs.altitude - cy.altitude) + ABS(cs.latitude - cy.latitude) as proximity
FROM
  person p,
  city cy,
  chargestation cs,
  webclient wc
WHERE
  cy.idcity = p.idcity
  AND cy.idstate = p.idstate
  AND cy.idcountry = cs.idcountry
  AND wc.idwebclient = p.idperson
  AND wc.nick = 'bcorrea'
ORDER BY proximity ASC;
```

Mostreu les deu primeres files del resultat de l'execució de la consulta:

<div> <div></div> <div>SQL</div> </div> <div>Todas las Filas Recuperadas: 28 en 0,004 segundos</div>									
IDCHARGESTATION	CHARGESTATIONNAME	ADDRESS	CITY	CS_ALITUDE	CS_LATITUDE	ELEVATION	STALLS	PROXIMITY	
1	697 Sant Cugat, Spain	08174 Sant Cugat del Vallès	Barcelona	41,498135	2,072674	209	12	0,207541	
2	700 Caldes de Malavella, Spain	Carretera N-II, km. 701	Caldes de Malavella	41,859029	2,767237	132	10	1,068346	

Creeu una vista a partir d'aquesta consulta, amb nom V\_ChargeStationDistance, substituint a la comparació del WHERE, WebClient.nick = 'bcorrea', per UPPER(WebClient.nick) = UPPER(USER).



## PROPOSTA DE SOLUCIÓ

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

```
CREATE VIEW V_ChargeStationDistance AS
SELECT
  cs.idchargestation,
  cs.name as chargestationname,
  cs.address,
  cs.city,
  cs.altitude cs_altitude,
  cs.latitude cs_latitude,
  cs.elevation,
  cs.stalls,
  ABS(cs.altitude - cy.altitude) + ABS(cs.latitude - cy.latitude) as proximity
FROM
  person p,
  city cy,
  chargestation cs,
  webclient wc
WHERE
  cy.idcity = p.idcity
  AND cy.idstate = p.idstate
  AND cy.idcountry = cs.idcountry
  AND wc.idwebclient = p.idperson
  AND UPPER(wc.nick) = UPPER(USER)
ORDER BY proximity ASC;
```

Inicieu sessió al SQLLoader amb l'usuari SYSTEM. Creeu un usuari amb l'identificador 'bcorrea'. L'usuari ha de tenir el password "ABC1234". Mostreu la instrucció SQL utilitzada per crear l'usuari:

```
CREATE USER bcorrea IDENTIFIED BY ABC1234;
```

Doneu al nou usuari permisos d'inici de sessió i de consulta de la nova vista V\_ChargeStationDistance. Mostreu les instruccions SQL corresponents:

```
GRANT CREATE SESSION TO "BCORREA";
GRANT SELECT ON "T18"."V_CHARGESTATIONDISTANCE" TO "BCORREA";
```

Inicieu sessió amb el nou usuari i utilitzeu la vista per a consultar les dades de les estacions de càrrega que té més properes.

Mostreu la instrucció SQL utilitzada per a fer la consulta:

```
SELECT
  idchargestation,
  chargestationname,
```



## PROPOSTA DE SOLUCIÓ

```
address,
city,
cs_altitude,
cs_latitude,
elevation,
stalls,
proximity
FROM
T18.v_chargestationdistance;
```

Mostreu les 10 primeres files del resultat de l'execució de la consulta:

IDCHARGESTATION	CHARGESTATIONNAME	ADDRESS	CITY	CS_ALTITUDE	CS_LATITUDE	ELEVATION	STALLS	PROXIMITY
1	697 Sant Cugat, Spain	08174 Sant Cugat del Vallès	Barcelona	41,498135	2,072674	209	12	0,207541
2	700 Caldes de Malavella, Spain	Carretera N-II, km. 701	Caldes de Malavella	41,859029	2,767237	132	10	1,068346
3	709 Tarragona, Spain	La Boella Tarragona exit 12 Autovia T-11	Tarragona	41,134294	1,1683	52	3	1,255326
4	710 Girona, Spain	Calle Francesc Ferrer, 16-18 17006	Girona	41,998411	2,817295	83	5	1,257786
5	687 La Seu d Urgell, Spain	Carrer del Bisbe Benlloch, 34	La Seu d Urgell	42,359108	1,462297	690	10	1,678891
6	706 Lleida, Spain	S N La Variant Torrefarera	Catalunya	41,663959	0,605371	203	6	1,840668
7	708 L'Aldea, Spain	L'Aldea, Spain 43896	L'Aldea	40,753019	0,606421	32	6	2,19848
8	696 Zaragoza, Spain	Estación de Servicio EL CISNE, A2, Km 309	Zaragoza	41,625707	-1,009618	321	10	3,417405
9	702 Ariza, Spain	Calle Carr. de Madrid, 197	Ariza	41,313192	-2,002201	726	6	4,246929
10	707 Valencia, Spain	S/N Carrer de Leonardo Da Vinci Paterna	Valencia	39,542965	-0,451521	107	4	4,466476

## FI DE LA PRÀCTICA



## PROPOSTA DE SOLUCIÓ

### 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: 25%
- Exercici 2: 25%
- Exercici 3: 25%
- Exercici 4: 25%

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. Utilitzeu el format original del document i els espais reservats, per a respondre cada exercici. 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. **Creeu les taules amb una única instrucció CREATE, no afegiu restriccions a posteriori (ALTER TABLE).**

**Retalleu les captures de pantalles per a mostrar allò que es demana (no feu captures de pantalla de tot l'escriptori). Assegureu que en totes les captures de pantalla en les que es demana mostrar un 'pla d'execució' es veuen correctament les columnes 'cardinality' i 'cost'.**

No envieu cap fitxer addicional 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.



## PROPOSTA DE SOLUCIÓ

La data límit per lliurar la segona pràctica és dijous dia 13 de desembre.



## PROPOSTA DE SOLUCIÓ

### **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 el 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, d'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.