**Alumne/a……………………………………... 1er DAW**

**MÒDUL 2: BASES DE DADES (I) 30/05/19**

**UF3: Llenguatges SQL: DCL i extensió procedimental**

*Les faltes d’ortografia descompten seguint la normativa de centre.*

**Usuaris**

**Exercici 1 (1 punt)** Crea un usuari que es digui examen3 amb privilegis necessaris per poder crear taules, procediments, triggers… fer inserts,...  
Un cop fet, crea una nova connexió que es digui també examen3.

Quant acabis, passa’m la query o querys per crear l’usuari, i un pantallasso on es vegi la connexió creada amb les seves dades de connexió

|  |
| --- |
| RESULTAT: |

Fes la resta de l’examen amb la nova connexió. Si no has pogut fer l’exercici anterior, fes-lo amb la connexió habitual que tinguis.

**Blocs anònims**

**Exercici 2 - (1 punt)**

Crea la següent taula:

create table productes(

id\_producte number primary key,

preu\_producte NUMBER(8,2),

stock NUMBER

);

I fes un bloc anònim que empleni la taula amb 500 productes amb els id\_producte d’1 a 500, amb un preu random entre 100 i 1000, i amb un stock random d’entre 0 i 100.  
Per als random, podeu fer servir la funció DBMS\_RANDOM.Value i per arodonir la funció ROUND

|  |
| --- |
| RESULTAT: |

**Procediments**

**Exercici 3 (1,5 punt)**

**3.1 (1 punts)** Crea un procediment que es digui productesPreus.  
El procediment rebrà dos paràmetres: preu\_minim i preu\_maxim.  
I el que ha de fer és obtenir el número de productes entre aquests dos valors, i la suma dels seus preus (en una sola select).

Després, amb aquestes dues dades, es mostrarà per pantalla la següent informació:

* El número d’elements
* La suma dels preus dels productes

*El número dels productes es: 50*

*La suma dels productes es: 7857,23*

|  |
| --- |
| RESULTAT: |

Penja només el procediment final

**3.2 (0.5 punts)** Finalment, amb un bloc anònim, feu que es demani per teclat els dos preus i faci la crida del procedure.

|  |
| --- |
| RESULTAT: |

Penja el bloc anònim

**Procediments, funcions, cursors i excepcions**

**Exercici 4 (3 punts)**

Crea la següent taula:  
  
create table comandes(

id\_producte number,

data\_comanda date,

stock\_a\_demanar NUMBER

);

Recorda d’enganxar a sota de cada punt el resultat abans de fer el següent.

**4.1 (1)** Fes un procediment que es digui *fesComanda* amb el que, mitjançant un cursor, recorri la taula de productes, de tots aquells que tinguin menys de 10 d’stock i que, per cada producte faci una comanda (insereixi un registre a la taula comandes), guardant la data d’avui al camp data\_comanda i com a stock\_a\_demanar ha de guardar 50-stock actual. El cursor s’ha de recorrer fent servir un “WHILE”

|  |
| --- |
| RESULTAT: |

Penja el procediment

**4.2 (0.75)** Ara volem fer una comanda on l’stock a demanar vagi en funció del preu del producte. De manera que si el preu està entre 100 i 400 demanem 100 productes, si està entre 401 i 700 en demanem 50 i si està per sobre de 700 en demanem 20.  
Fes una funció que es digui getStockDemanar a la que li passarem el preu del producte per paràmetre, i ens retornarà els productes a demanar.  
Modifica el procediment anterior per a que cridi la funció i en comptes del “50” fixos, agafi el valor que retorna la funció

|  |
| --- |
| RESULTAT: |

Penja el procediment modificat i la funció

**4.3**  **(0.5)** Realitza els canvis necessaris per a que la funció que acabes de fer, sigui un procediment i tot funcioni igual.

|  |
| --- |
| RESULTAT: |

Penja el procediment modificat i la funció transformada en procediment

**4.4 (0.75)** Per últim, realitza els canvis necessaris per tal de que, si ja existeix una comanda d’un determinat producte, llenci una excepció que digui “*La comanda no s’ha pogut realitzar per que ja hi ha una comanda pendent del producte amb id xxxx*”  
Però s’ha de capturar l’exepció, no ha pas de sortir l’error, sinó el text

|  |
| --- |
| RESULTAT: |

**Exercici 5 (1.5 punt)**

Fer un procediment que es digui “mostraStock” que printi per pantalla l’id dels productes i l’stock.  
També haurà de mirar, per cada producte, si te alguna comanda realitzada. Si te comandes, ha de mostrar també les dates de les comandes, i el número de productes demanats. Podeu fer servir el recorregut de cursors que més us agradi.  
Pista: 2 cursors aniuats.

El resultat hauria de ser una cosa així:

. . . . . .  
 ID\_producte: 17 stock actual: 92

ID\_producte: 18 stock actual: 28

ID\_producte: 19 stock actual: 20

--->Comanda realitzada el : 26/05/2019 stock a demanar: 15

ID\_producte: 20 stock actual: 5

--->Comanda realitzada el : 26/05/2019 stock a demanar: 45

. . . . . .

|  |
| --- |
| RESULTAT: |

Pegam aquí només el procediment, no cal pantallasso de la crida.

**Triggers**

**Exercici 6 (2 punt)**

**6.1 (0.75 punt)** Crea la següent taula:

create table registre\_accessos(

usuari varchar2(50),

ip\_client varchar2(15),

hora DATE,

accio varchar2(50)

);

I fes els triggers necessaris per a que registri quin usuari es conecta i desconecta a la base de dades, , la seva ip i a quin hora.

|  |
| --- |
| RESULTAT: |

Pegam aquí només el trigger.

**6.2 (1.25 punt)** Crea la següent taula:

create table registre\_canvis\_productes(

usuari varchar2(50),

hora DATE,

accio varchar2(50),

id\_producte NUMBER,

preu\_anterior number (8,2),

preu\_posterior number (8,2),

stock\_anterior number,

stock\_posterior number

);

**I** crea un altre tirgger que faci un insert a la taula per registrar quin usuari fa canvis a la taula productes, indicant l’hora i el tipus de canvi realitzat (insert, update o delete), així com els valors anteriors i posteriors al canvi.

Quan acabeu, llenceu aqueses querys:

update productes set stock = stock+5, preu\_producte=preu\_producte+50 where id\_producte in (15, 20);  
Insert into productes values (501, 210.25, 54);

I fes una caputra del resultat de fer la select \* from registre\_canvis\_productes; a més del trigger.

|  |
| --- |
| RESULTAT: |

Pegam aquí el trigger i la captura.