

Activitat 3.1 - Introducció de la programació

Realitzar les tasques per adquirir els coneixements per aprendre a programar PL/SQL en una base de dades usant el llenguatge SQL. Podeu consultar la documentació DCL des de l'apartat 1, 2, 3.1 i 5.

Enunciat

A partir de la BD [pizzeria-creació.sql](#), les dades [pizzeria-insert_select.sql](#) i juntament amb el [diagrama de taules](#). Programa a PL/SQL la solució de cada tasca.

Temps estimat: 2 hores.

Sense Avaluació: solució a classe.

Tasca 1.

Mostra quantes comandes es van fer l'11/01/2017. Fes servir una variable.

```
SET @numero_comandes = (SELECT COUNT(*)  
                        FROM comandes  
                        WHERE      DATE(data_hora)      =  
DATE('2017-01-11'));  
SELECT @numero_comandes;
```

Tasca 2.

Mostra el nom del clients i un contador del registre alhora que es mostren els valors de la consulta ordenats per nom.

```
SET @contador := 0;  
SELECT nom, @contador := @contador + 1  
FROM clients  
ORDER BY nom;
```

Tasca 3.

Enmagatzema la quantitat en euros que ha facturat l'empleat amb major facturació. Mostra posteriorment la dada enmagatzemada a la variable amb el valor resultant.

```
SELECT MAX(facturat.facturacio) INTO @facturacio
FROM empleats em
    INNER JOIN (
        SELECT e.id_empleat, SUM(cp.quantitat * p.preu) AS facturacio
        FROM empleats e
            INNER JOIN comandes co ON e.id_empleat = co.empleat_id
            INNER JOIN comandes_productes cp ON co.numero = cp.numero
            INNER JOIN productes p ON cp.id_producte = p.id_producte
        GROUP BY e.id_empleat) AS facturat ON em.id_empleat =
facturat.id_empleat;
SELECT @facturacio;
```

Tasca 4.

A partir d'una variable que contingui el valor d'increment de preus 1%, aplica l'increment de preus a tots els postres.

```
SET @increment = CAST(0.01 AS DECIMAL(4,2));
UPDATE productes
SET preu = preu + (preu * @increment)
WHERE id_producte IN (SELECT id_producte FROM postres po);
SELECT *
FROM productes p
    INNER JOIN postres po ON p.id_producte = po.id_producte;
```

Tasca 5.

Localitza el client que més ha comprat (valor total de les seves comandes). Enmagatzema el nom del client i el preu total de totes les seves comandes, posteriorment mostra les dades enmagatzemades de les variables.

```
SELECT cl.nom, SUM(cp.quantitat * p.preu) AS facturacio INTO @client, @facturacio
FROM clients cl
      INNER JOIN comandes co ON cl.id_client = co.client_id
      INNER JOIN comandes_productes cp ON co.numero = cp.numero
      INNER JOIN productes p ON cp.id_producte = p.id_producte
GROUP BY cl.id_client
ORDER BY facturacio DESC
LIMIT 1;
```

Tasca 6.

Cerca que busqui totes les pizzes que tinguin com a ingredient BAC. Fes servir una variable per l'ingredient.

```
SET @ingredient = 'BAC';
SELECT DISTINCT p.nom
FROM productes p
      INNER JOIN pizzes pz ON p.id_producte = pz.id_producte
      INNER JOIN pizzes_ingredients pi ON pz.id_producte = pi.id_producte
WHERE pi.id_ingredient = @ingredient;
```

Tasca 7.

Fent l'ús de variables, modifica el preu del producte 3 a 1.65 €.

```
SET @preu = 1.65;
SET @id_producte = 3;
SELECT *
FROM productes
WHERE id_producte = @id_producte;
UPDATE productes
SET preu = @preu
```

```
WHERE id_producte = @id_producte;  
SELECT *  
FROM productes  
WHERE id_producte = @id_producte;
```

Recursos

- [Moodle M2](#)
- [DCL Account Management \(MariaDB documentation\)](#)