Sessió Lab 3 – Consultes sobre la base de dades

Executeu l'script biblioteca.sql que hi ha al Moodle. Aquest script us crearà una base de dades anomenada biblioteca, amb totes les seves taules i amb alguns registres de prova.

Comanda SELECT

La comanda d'SQL per realitzar consultes és la comanda SELECT:

```
SELECT camps
FROM taula
WHERE condició
ORDER BY camps_per_ordenar;
```
[1]

Les clàusules WHERE i ORDER BY són opcionals (només les posem si volem filtrar o ordenar, respectivament.

On:

camps: llistat dels camps que es volen mostrar, separats per comes. Es pot substituir la llista de camps per un asterisc (*) que indica que volem tots els camps.

Taula: nom de la taula de la que volem treure informació

condició: condició booleana que s'utilitza per filtrar els registres, de manera que només es mostrin aquells registres que compleixin la condició. Permet posar diferents operadors els quals es detallaran posteriorment.

camps_per_ordenar: llistat dels camps pels quals en volem ordenar el resultat. En cas de posar més d'un camp, estarem ordenant pel primer camp, i utilitzant el segon per ordenar aquells registres que tinguin el mateix valor en el primer camp, és a dir, el camp2 s'utilitzaria per desempatar. Després de cada un dels camps es pot afegir la paraula DESC per indicar que volem ordre descendent (si no s'indica, el pressuposa la paraula ASC que indica ordre ascendent).

Exemple:

La següent consulta mostra el DNI i nom dels socis donats d'alta abans del 2005, ordenats pel seu nom (en ordre ascendent) i, en cas d'empat, s'ordena pel DNI (en ordre descendent).

```
SELECT dni, nom
FROM Socis
WHERE data_alta < '2005-01-01'
ORDER BY nom, dni DESC;
```

Exercici:

1. Realitza una consulta per a mostrar per pantalla les dades dels llibres que són de la primera edició i que s'han editat després de l'any 1999.

1 Existeixen també les clàusules GROUP BY i HAVING les quals seran introduïdes més endavant.

Clàusula DISTINCT

Just després de la paraula SELECT, abans de la llista de camps a mostrar, es pot afegir la paraula clau DISTINCT per tal d'evitar que apareguin files repetides.

La següent consulta obté un llistat de les poblacions (sense repetits) on viuen els socis de la biblioteca:

```
SELECT DISTINCT poblacio
FROM Socis;
```

IMPORTANT: La operació d'eliminar repetits consumeix recursos. Així doncs, hauriem d'utilitzar la clàusula DISTINCT només quan sigui realment necessari.

Exercici:

2. Fes una consulta per a mostrar per pantalla de quins temes tenim llibres
3. Fes una consulta per a obtenir sense repetits els DNIs dels socis

Operadors de la clàusula WHERE

La condició de la clàusula WHERE admet els següents operadors:

- De comparació: `<`, `>`, `<=`, `>=`, `=`, `<>`
- Tractament de nuls: `IS NULL`, `IS NOT NULL`
- Lògics: `AND`, `OR`, `NOT`
- Operadors específics de SQL:
  - `IN`, `NOT IN`
  - `BETWEEN` ... `AND` ...
  - `LIKE`
- Funcions específiques
  - Funcions matemàtiques (sin, cos, sqrt, ...)
  - Funcions de date i time (curdate, year, month, day, ...)
  - Funcions de strings (char_length, left, replace, ...)

És important que no utilitzeu mai els operadors de comparació per comparar amb NULLs. Per saber si un camp és null cal utilitzar sempre els operadors `IS NULL` o `IS NOT NULL`.

Exemple:

Obtenir tots els socis pels quals en coneixem la seva població:

```
SELECT nom, poblacio
FROM Socis
WHERE poblacio IS NOT NULL;
```

Els operadors IN i NOT IN s'utilitzen per comparar si el valor del camp està contingut en un conjunt de valors:

Exemples:

Obtenir els socis que viuen en alguna capital de província de Catalunya:

```
SELECT nom, poblacio
FROM Socis
WHERE poblacio IN ('Barcelona','Tarragona','Lleida','Girona');
```

Obtenir els socis que no viuen ni a Barcelona ni a Tarragona:

```
SELECT nom, poblacio
FROM Socis
WHERE poblacio NOT IN ('Barcelona','Tarragona');
```

L'operador between no és realment necessari, però resulta còmode quan volem consultar si un camp conté un valor dins d'un rang:

```
SELECT ISBN
FROM Edicions
WHERE num_planes BETWEEN 900 AND 1200;
```

L'anterior consulta és equivalent a:

```
SELECT ISBN
FROM Edicions
WHERE num_planes >= 900
  AND num_planes <= 1200;
```

L'operador LIKE s'utilitza per comparar textos amb un text patró, en el qual hi utilitzem caracters comodí:

%: Actua com a un conjunt de caràcters (de 0 a N caràcters)

_ : Actua com a exactament un caràcter.

Exemple:

Obtenir els noms dels socis que comencen per A:

```
SELECT nom
FROM Socis
WHERE nom LIKE 'A%';
```

Obtenir els noms dels socis que acaben per S i la seva segona lletra és una 'o':

```
SELECT nom
FROM Socis
WHERE nom LIKE '_O%S';
```

A banda dels operadors estandards de SQL, cada sistema gestor de base de dades incorpora un conjunt de funcions de strings, dates, etc... La sintaxis i funcionalitat d'aquestes funcions poden variar en cada SGBD i per tant es recomana buscar-ne informació en el manual.

Exemple: Obtenir l'edat dels Socis:

```
SELECT nom, data_naix,
  CASE
     WHEN (MONTH(data_naix) < MONTH(CURDATE())) THEN YEAR(CURDATE()) - YEAR(data_naix)
     WHEN (MONTH(data_naix) = MONTH(CURDATE())) AND (DAY(data_naix) <= DAY(CURDATE())) THEN
YEAR(CURDATE()) - YEAR(data_naix)
     ELSE (YEAR(CURDATE()) - YEAR(data_naix)) - 1
   END AS edat
FROM Socis
```

Exercicis

3. Obtenir els socis que es van apuntar a la biblioteca durant el mes d'abril (de qualsevol any).

4. Obtenir els préstecs actius. Concretament volem mostrar quants dies fa que es van fer, i quants dies falten per tornar el llibre, suposant que els préstecs són de 15 dies com a màxim.

5. Obtenir una previsó de l'ordre en que els llibres en préstec seran tornats. Mostrar el títol del llibre i el nom del soci.

6. Obtenir els noms dels socis que es diuen Joan i que es van fer socis entre l'1/1/2005 i el 31/12/2008.

7. Obtenir els noms dels socis que són de Reus o Tarragona.

Join

En la clàusula FROM, realment, s'hi pot posar un conjunt de taules (separades per comes). Si fem això, sense fer res més, estarem obtenint el producte cartesià (és a dir, totes les combinacions possibles) dels registres d'aquestes taules (per exemple entre Volums i Eicions).

Si el que volem és obtenir edició amb la informació del seu volum, hem d'afegir la condició de join a la clàusula WHERE:

```
SELECT isbn, titol, num_edicio, num_volum
FROM Edicions, Volums
WHERE num_volum_edició = num_volum;
```

En l'anterior exemple estem suposant que el camp num_volum_edició és el camp de la taula Edicions que conté el número de volum. Si aquest camp es diu igual que en la taula Volums (tal i com passa a la nostra base de dades), hauriem d'utilitzar àlies a les taules, cosa que també és útil si, pel camp num_edicio per ser un camp ambigu (està a les 2 taules amb el mateix nom):

```
SELECT isbn, e.num_edicio, e.num_volum, titol
FROM Edicions AS e, Volums AS v
WHERE e.num_volum = v.num_volum
  AND e.codi_obra = v.codi_obra;
```

NOTA: Per definir un àlies, la paraula AS és opcional, de manera que també es podria fer així:

```
SELECT isbn, e.num_edicio, e.num_volum, titol
FROM Edicions e, Volums v
WHERE e.num_volum = v.num_volum
  AND e.codi_obra = v.codi_obra;
```

Exercicis

8. Fes una consulta que mostri els títols de les obres que tenen algun exemplar de consulta en sala.

9. Obtenir els noms dels autors que han escrit el llibre titulat “Programación en C”. Fes que els autors apareguin en el mateix ordre en que apareix en la portada del llibre.

Funcions d'agregat

Hi ha vegades en que en comptes d'obtenir un llistat de registres, el que volem és obtenir algun resum d'aquests registres, com ara obtenir quants socis compleixen una certa condició, o quin és el sou més alt d'un empleat. Per a fer això disposem de les següents funcions d'agregat, que són estandards d'SQL:

Count(*): Obté el número de registres

Sum(camp): Obté la suma dels valors del camp

Avg(camp): Obté el promig dels valors del camp

Min(camp): Obté el valor mínim del camp

Max(camp): Obté el valor màxim del camp.

La funció Count accepta també posar-hi un camp. En aquest cas comptarà quants registres contenen un valor per a aquest camp, és a dir, no comptarà els registres que tinguin un valor NULL per a aquest camp. També és possible posar la paraula DISTINCT davant del camp de dins el Count per indicar que només volem comptar valors diferents.

És molt habitual utilitzar àlies en les columnes on s'utilitza una funció d'agregat.

Exemple (en una hipotètica base de dades d'empleats):

```
SELECT Max(sou) as sou_maxim, Min(sou) as sou_minim,
       Sum(sou) as cost_total,
       Count(*) as n_empleats,
       Count(poblacio) as empleats_amb_poblacio,
       Count(Distinct poblacio) as n_poblacions
FROM Empleats
WHERE num_dpt=1;
```

Exercicis

10. Fes una consulta per obtenir quants llibres estan en préstec

11. Fes una consulta per obtenir el número total d'exemplars que té la biblioteca, juntament amb el número total de pàgines que hi ha a la biblioteca (la suma de pàgines de tots els exemplars).

12. Fes una consulta que obtingui quantes pàgines té el llibre que té més pàgines

13. Fes una consulta per obtenir en quantes poblacions la biblioteca té algun soci que hi viu

La clàusula GROUP BY

Hi ha vegades en que aquestes funcions d'agregats les volem calcular per cada valor d'un camp concret. Per exemple, ens pot interessar obtenir el número de socis per cada població. Quan això passa cal agrupar els registres (en aquest cas socis) en grups (en aquest cas poblacions). Per fer-ho s'utilitza la clàusula GROUP BY:

```
SELECT poblacio, count(*)
FROM Socis
GROUP BY poblacio;
```

Quan utilitzem la clàusula GROUP BY, la SELECT només pot contenir funcions d'agregat i els camps del GROUP BY, però no podem posar altres camps. Aquesta restricció també es compleix si no posem la clàusula GROUP BY però tenim funcions d'agregats a la SELECT (en aquest cas la SELECT només pot tenir funcions d'agregat).

Exercicis

14. Fes una consulta que mostri, per cada població, quants socis en tenim. Concretament volem mostrar el nom de la població i el número de socis que hi ha en ella. Volem el llistat ordenat alfabèticament per població.

15. Fes una consulta per mostrar quants llibres té la biblioteca de cada editorial. Mostreu el nom de l'editorial i el número de llibres.

16. Fes una consulta per obtenir el títol de cada llibre juntament amb la quantitat d'autors que aquest té. Fes que s'ordeni descendentment per quantitat d'autors.

La clàusula HAVING

Informalment es diu que el HAVING és el "WHERE del GROUP BY". És a dir, permet filtrar grups. A efectes pràctics això significa que podem posar funcions d'agregat en la condició de filtre. Per la resta la condició, funciona exactament igual que en el WHERE.

Exemple:

Obtenir el nom de les poblacions que tinguin més de 3 socis

```
SELECT poblacio
FROM Socis
GROUP BY poblacio
HAVING count(*)>3;
```

Fixeu-vos que no té sentit posar la clàusula HAVING si no hi ha la clàusula GROUP BY (si ho fessim, l'SGBD ens retornaria un error).

Exercicis

17. Obtenir els títols dels llibres que tenen més d'un autor.

18. Obtenir els noms dels socis que tenen més d'un llibre en préstec

19. Obtenir un llistat alfabètic de les editorials que tenen algun llibre amb menys de 1000 pàgines.

SUBCONSULTES

És possible posar una sentència SELECT dins d'una consulta, de manera que aquesta SELECT substitueixi un valor concret. A aquesta SELECT posada dins d'una consulta se li diu subconsulta, i ha d'anar sempre entre parèntesis.

Això s'utilitza per realitzar consultes complexes, i ens permet resoldre-les per parts.

Per exemple, si ens demanessin d'obtenir l'ISBN del llibre que té més planes, podríem fer la següent consulta:

```
SELECT isbn
FROM Edicions
WHERE num_planes = (
                SELECT Max(num_planes)
                FROM Edicions
              );
```

On podem veure que la subconsulta (en negreta) obté quin és el número màxim de planes, per després obtenir (a la consulta mare) quin és el llibre que té aquest número de planes màxim.

En l'exemple anterior la consulta ha de retornar exactament un valor, ja que s'ha posat en el lloc on hi aniria un valor (en aquest exemple, després d'un comparador d'igualtat). No obstant, és possible també fer consultes que retornin diversos valors i posar-la dins d'un IN.

La següent consulta obté els socis que no viuen en cap ciutat de més de 100.000 habitants:

```
SELECT nom
FROM Socis
WHERE poblacio NOT IN (
                          SELECT nom
                          FROM Poblacions
                          WHERE n_hab > 100000
                       );
```

Com es pot observar, s'ha substituït el llistat de valors de dins l'operador IN (NOT IN en aquest exemple) per una SELECT d'una columna que pot retornar múltiples files. Posar una subconsulta dins d'un NOT IN és molt habitual, i s'utilitza per fer el que en àlgebra relacional es coneix amb el nom de diferència o resta.

Si en la diferència que volem fer estessin implicats dos camps (per exemple, si la població estigués identificada pel seu nom conjuntament amb el país al que pertany) aleshores hauríem d'utilitzar l'operador EXISTS que permet posar subconsultes de més d'un camp. En aquest cas s'utilitzaria aquesta sintaxis:

```
SELECT nom
FROM Empleats e
WHERE NOT EXISTS (
      SELECT * -- Aquí hi podem posar qualsevol combinació de camps
      FROM Poblacions p
      WHERE p.n_hab > 100000
        AND p.nom = e.poblacio
        AND p.pais = e.pais
      );
```

NOT EXISTS retorna cert a la consulta mare si la subconsulta retorna 0 registres, i retorna fals si la subconsulta retorna 1 o més registres. Observeu que des de la subconsulta es pot accedir als camps de la consulta mare.

IMPORTANT: Algunes consultes es poden resoldre utilitzant subconsultes o sense utilitzar-les. Només hauriem d'utilitzar subconsultes quan és necessari.

Per exemple, si ens diuen d'obtenir els empleats que cobren més que l'empleat número 4, podem fer-ho amb una subconsulta:

```
SELECT nom
FROM Empleats e
WHERE e.sou > (
               SELECT sou
               FROM Empleats
               WHERE num = 4
              );
```

Però també es podria resoldre així:

```
SELECT nom
FROM Empleats e, Empleats e4
WHERE e4.num = 4
  AND e.sou > e4.sou;
```

Aquesta segona solució és millor perquè utilitza una join en comptes d'una subconsulta.

Les subconsultes es poden utilitzar a la clàusula WHERE de qualsevol instrucció (no només SELECT: també UPDATE o DELETE).

Les subconsultes es poden utilitzar també a la clàusula HAVING.

Exercicis

20. Obtenir els noms dels socis que no tenen actualment cap llibre en préstec.

21. Obtenir quin és l'autor que ha escrit més llibres. Mostrar-ne només el seu nom.

22. Obtenir els noms dels socis que han llegit el llibre que té més pàgines.

23. Obtenir els noms dels socis que tenen més llibres en préstec que el soci núm. 3.

24. En apuntar el número de planes dels llibres escrits per l'autor Paul Dubois, ens hem descuidat de comptar les 10 primeres pàgines que formen sempre l'index d'aquests llibres. Fes una modificació a la base de dades que incrementi en 10 el número de planes de tots els llibres en els que l'autor Paul Dubois ha intervingut.

25. Obtenir el llistat de llibres (obres) de l'editorial que té més llibres. Ens interessen els títols d'aquestes obres ordenats alfabèticament.

26. Obtenir el llistat de llibres (edicions) que tenen més planes que el llibre que té el codi isbn 059652708X