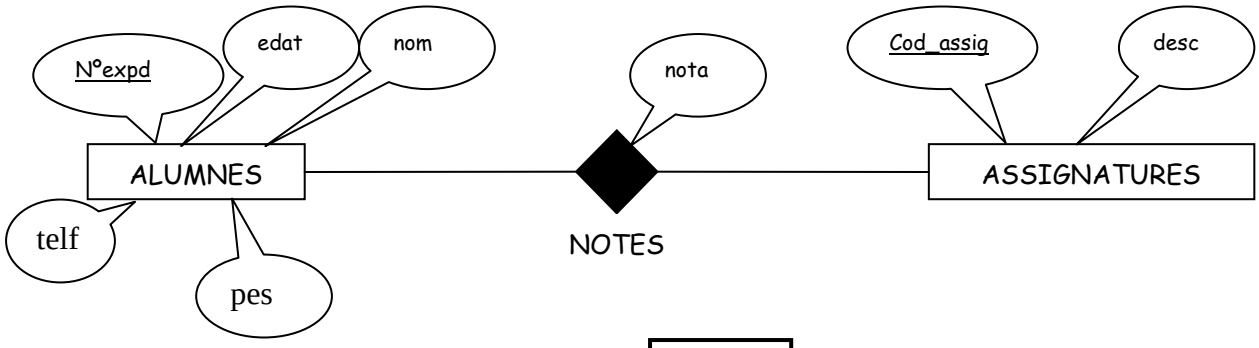
PRÀCTICA

SENTÈNCIES SELECT

A. Moll

La següent pràctica dissenya i implementa una senzilla base de dades amb 3 taules per tal de fer-li, posteriorment, una serie de consultes que aniràn tocant paulatinament molts conceptes de la sentència SELECT. La pràctica és, doncs, prou conceptual. Es per això que no hi ha restriccions en les taules i els tipus de dades emprats són, com es veurà, senzills i fonamentals.



a) Disseny Lògic

ALUMNES = Nºexpd + edat + nom + tlf + pes.
ASIGNATURA = CodAsig + desc
NOTES = Nºexped + CodAsig + nota

C.Aj { Nºexped → ALUMNE
CodAsig → ASSIGNATURA

Esquema lògic relacional

Implementació en DDL (SQL)

Create Table *Alumnes* (Nexpd **char(4) Primary Key**, Edat **int2**, Nom **VarChar**(30), Tlf **char**(9), Pes **numeric(4,1))**;	**Create Table** *Assignatures* (CodAssig **char(4) Primary Key**, Descripcio **varchar(56)**);

Create Table *Notes*

(Nexpd **char(4) References** *Alumnes*,

CodAssig **char**(4) **References** *Assignatures*,

Nota **numeric(4,2)**,

Primary Key (Nexpd, CodAssig))

Ens situem, per exemple, en la carpeta **data**. Així serà aquest el nostre "working directory".

psql template1 pepito ➔ Ens connectem, per exemple, a la base de dades *template1* com superusuaris. Teniu cura: Es case sensitive!!.

Create database notes Però la base de dades activa continua essent *template1*.

\c notes ➔ Ens connectem a la base de dades buida notes

\i notes.sql ➔ notes.sql el crearem amb el Bloc de Notes i contindrà les ordres DDL de la base de dades així com els inserts de les tuples.

A partir d'ara estem en condicions de resoldre les consultes següents. Les resoldrem en l'estàndart SQL92 i també en Postgres. Malgrat que aquest últim respecta prou l'estàndart ens trobarem amb alguna diferència.

1. Tota la informació dels alumnes

2. Nom i tlf dels alumnes entre 25 i 30 anys

3. Nexpd dels alumnes matriculats amb alguna nota de les següents: 5, 5.5, 6, 7.8

4. Edat i pes en grams dels alumnes amb nom desconegut i per als que, a més, els telèfons comencen amb 96 i siga un 5 el penúltim dígit.

5. Nexpd, nom i nota que han tret els alumnes.

```
select AL.Nexpd, Nom, Nota
from Alumnes AS AL, Notes AS N
where AL.Nexpd=N.Nexpd;
```

Incorpora dos taules que s'hauran de vincular. A més amb *alies*, on hem ficat en la projecció el que és necessari (AL.Nexpd).

En lloc de fer nosaltres la vinculació ens la pot fer el SQL si muntem una concatenació natural. El següent query és equivalent:

```
select al.nexpd, nom, nota
from alumnes as al NATURAL JOIN notes as n;
```

6. Nexpd i nom dels alumnes majors de 30 anys juntament amb la descripció de les assignatures en que s'han matriculat.

7. Nom d'alumnes amb la mateixa edat que l'alumne 0002

8. Nom i edat dels alumnes matriculats en assignatures interessants (camp descripció)

```
select al.nom, al.edat
from alumnes as al, notes as n, assignatures as a
where al.nexpd=n.nexpd and n.codassig=a.codassig and descripcio='Interessant';
```

En l'anterior solució si un alumne té 2 assignatures interessants doncs eixiria dos vegades!!.

```
select distinct al.nom, al.edat
from alumnes as al, notes as n, assignatures as a
where al.nexpd=n.nexpd and n.codassig=a.codassig and descripcio='Interessant';
```

Arreglem el problema d'abans PERÒ ara si dona la casualitat de que tinc dos alumnes amb el mateix nom i edat doncs només surtiria un dels dos.

```
select nom,edat
from alumnes
where nexpd in (select nexpd
                from notes
                where codassig in (select codassig
                                   from assignatures
                                   where descripcio='Interessant'));
```

Solució correcta. Exemple amb anidadament de segon nivell.

9. Descripció d'aquelles assignatures on les notes han estat totes majors que 9

```
select descripcio
from assignatures as a
where 9 < all (select nota from notes as n where n.codassig=a.codassig);
```

[ExprEscalar | (Subquery)] OpRel ALL (subquery)

Manual del Postgres ➔Si el subquery de la dreta no contesta res el predicat ALL contesta amb un TRUE. Ara bé, si hi ha valors de retorn i, a més, el subquery de la dreta contesta amb algún NULL, aleshores el predicat ALL s'avalua a INDEFINIT.

Aleshores l'anterior query llistaria les assignatures de les que no s'ha matriculat cap alumne. I al query següent també:

```
select descripcio
from assignatures
where codassig not in (select codassig from notes where nota<=9);
```

10. Nom d'alumnes on no hi siga l'alumne amb Nexpd més baix.

11. Codis d'assignatures interessants en les que s'han matriculat els alumnes majors d'edat

12. Nom i tlf d'aquells que sempre trauen més d'un 7 en totes les assignatures que s'han matriculat.

13. Pes mitjà dels alumnes majors de 20 anys.

14. Nexpd i nota mitjana de cada alumne

```
select nexpd, to_char(avg(nota),'99.9') as NotaMitjana
from notes
group by Nexpd;
```

Només eixirien els alumnes "matriculats" en alguna assignatura.

15. Nom d'alumnes matriculats en més assignatures que l'alumne 0002

16. Nexpd i nom de tots els alumnes juntament amb la quantitat d'assignatures que porten endavant.

```
select al.nexpd, nom, count(codassig) as NombreAssignatures
from alumnes as al LEFT JOIN notes as n ON al.nexpd=n.nexpd
group by al.nexpd, al.nom;
```

Primer exemple de CONCATENACIÓ EXTERNA (per l'esquerra). Els alumnes que no s'han matriculat en res surten i a més els fica un zero en la tercera columna. Ací teniu l'explicació clara i directa que dona l'*User Guide* del Postgres per al LEFT JOIN

> o First, an INNER JOIN is performed. Then, for each row in T1 that does not satisfy the join condition with any row in T2, a joined row is returned with NULL values in columns of T2. Thus, the joined table unconditionally has at least one row for each row in T1.

Consideracions:

- LEFT JOIN ≡ LEFT OUTER JOIN. Es a dir, OUTER aporta tan sols legibilitat.
- La clàusula ON ... és obligada
- També tenim el RIGHT [OUTER] JOIN i el FULL [OUTER] JOIN

Per cert, ¿qué passaria si en el count de la projecció fiquem **count(*)** en lloc de **count(codassig)**?

17. Nom d'alumnes que han tret la mateixa nota en, si més no, dos assignatures.

18. De tots els alumnes (inclosos els que no tenen cap assignatura) llisteu Nexpd, Nom, Nombre d'assignatures i Nota mitjana, ordenant el llistat per la nota mitjana (ascendentment) i després pel nom de l'alumne (descendentment).

PECULIARITATS DE POSTGRES EN LES SELECTS

```
Select *
from alumnes
Limit 10
```

Postgres no té la famosa taula predefinida "d'experiments" d'Oracle ➔ ***dual***		
Select current_user;	select random();	select 'Hola mon';

```
select nexpd, coalesce(nom, '<buit>'), coalesce(tlf,'<buit>')
from alumnes;
```

```
select (select 4) = 5;     // Admet selects en la projecció i avalua expressions booleanes
```

```
select *
from A
where a0 ilike ‘xyz’;  // ILIKE no distingueix entre minúscules i majúscules
```

```
select *
from (select ……) AS T
Where T.x⊙⊙⊙;
```