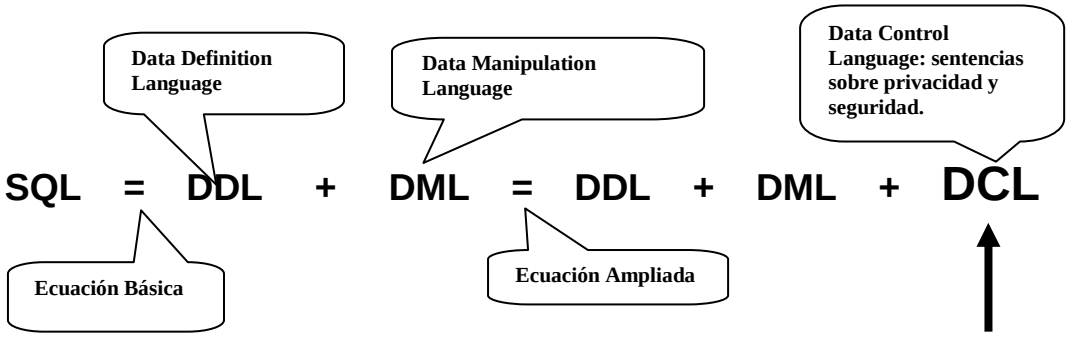
SUBLLENGUATGE DCL
GRUPS, USUARIS I PERMISOS

A. Moll

1. INTRODUCCIÓ



El que veurem…

2. USUARIS I GRUPS (*ROLES*)

The concept of roles subsumes the concepts of “users” and “groups”. In PostgreSQL versions before 8.1, users and groups were distinct kinds of entities, but now there are only roles. Any role can act as a user, a group, or both.

Els *rols* són globales a tot el *cluster* i no a una bd concreta.

The right to modify or destroy an object is always the privilege of the owner only. La forma de quitarle derechos al propietario es cambiar el propietario con la orden ALTER

```
CREATE ROLE name [ListaOpcion]

Opcion::=
     SUPERUSER | NOSUPERUSER
   | CREATEDB | NOCREATEDB
   | CREATEROLE | NOCREATEROLE
   | CREATEUSER | NOCREATEUSER
   | INHERIT | NOINHERIT
   | LOGIN | NOLOGIN
   | CONNECTION LIMIT n ←per defecte no hi ha limit a les connexions obertes
   | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
   | VALID UNTIL 'timestamp' ← per defecte indefinidament
   | IN ROLE rolename [, ...]
   | IN GROUP rolename [, ...]
   | ROLE rolename [, ...] ← Llista de membres del nou role
   | ADMIN rolename [, ...] ← Permet als membres citats açí incloure en aquest role
que estem definint a tercers roles.
   | USER rolename [, ...]
   | SYSID uid
```

A role with the ***INHERIT*** attribute can automatically use whatever database privileges have been granted to all roles it is directly or indirectly a member of

A role having the ***LOGIN*** attribute can be thought of as a user. Roles without this attribute are useful for managing database privileges, but are not users in the usual sense of the word

The ***VALID UNTIL*** clause defines an expiration time for a password only, not for the role *per se*. In particular, the expiration time is not enforced when logging in using a non-password-based authentication method.

3. PERMISOS SOBRE TAULES

3.1. Concessió ➔ GRANT

Cuando se crea una tabla sólo el superusuario y quien la ha creado (propietario) tienen permisos para manipularla. Cualquier otro usuario que quiera acceder a ella deberá recibir, previamente, la pertinente concesión de permisos.

Grant del PostgreSQL

```
GRANT { {SELECT|INSERT|UPDATE|DELETE|REFERENCES|TRIGGER} [,...] | ALL [PRIVILEGES]}
    ON [ TABLE ] tablename [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...]
[ WITH GRANT OPTION ]
```

Sintaxi que reescrivim…

```
GRANT {ComaListaPermiso | ALL}
    ON ComaListaTabla
    TO {ComaListaRoleName | PUBLIC}
[WITH GRANT OPTION]
```

PostgreSQL does not support the SQL-standard functionality of setting privileges for individual columns.

L'estàndard 99 sí reconeix permisos a nivell de columna. Vejam:

```
Permiso ::=
SELECT | SELECT ( ComaListaColumna )
| SELECT ( <privilege method list> )
| DELETE
| INSERT [ ( ComaListaColumna ) ]
| UPDATE [ ( ComaListaColumna ) ]
| REFERENCES [ ( ComaListaColumna ) ]
| USAGE | TRIGGER | UNDER | EXECUTE
```

Per exemple, en l'estàndard sí podem escriure:

o Grant Insert(Nexpd, nom) /* No estan explícitos todos los campos de la tabla Alumnes
On Alumnes
To Pepe

3.2. Revocació ➔ REVOKE

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | REFERENCES | TRIGGER }
    [,...] | ALL [ PRIVILEGES ] }
    ON [ TABLE ] tablename [, ...]
    FROM { [ GROUP ] rolename | PUBLIC } [, ...]
  [ CASCADE | RESTRICT ]
```

Que reescrivim, omitint el superflu:

```
REVOKE [ GRANT OPTION FOR ] {ComaListaPermiso | ALL}
    ON ComaListaTabla
    FROM {ComaListaRoleName | ALL}
[CASCADE | RESTRICT]
```

4. EXERCICIS

1. Qué executen les ordres següents.

Ordre	Significat
Grant all privilegies On Alumnes To Pepe	
Grant all On Alumnes To Pepe	
Grant Select, Update On Alumnes, Notes, Assignatures To Pepe With Grant Option	
Grant Select, Insert, Update, Trigger On Alumnes To group *UsuarisAvançats*	
Grant Select, Insert, Update	

On Alumnes, Assignatures To Pepe, group *UsuarisAvançats*
Grant all On Alumnes To public With Grant Option

2. Creem l'usuari *Primer* mitjançant **create user primer**.
2.1 ¿Té password aquest usuari?. ¿Ens podem connectar?.
2.2 Modifica l'usuari incorporant-li una contrassenya ➔ alter user primer password 'prova'.
2.3 Intenta crear la base de dades PrimeraBD. ¿Qué ha passat?.
2.4 ¿Pot aquest usuari obrir simultàniament 3 connexions?. Limita les possibles connexions a 2. →
Alter User primer connection limit 2

3. ¿Com saber quin és l'usuari que té el control de la consola?.

4. Pot un superusuari esborrar qualsevol taula, siga qui siga el propietari.

5. ¿Pot l'usuari *postgres* esborrar un role que és propietari d'algun objecte?

6. ¿Pot un usuari normal crear un altre usuari amb més "poders" que ell mateix?

7. Fins i tot ¿podría un usuari "normal" crear un superuser?

8. ¿Un usuari es pot suicidar?

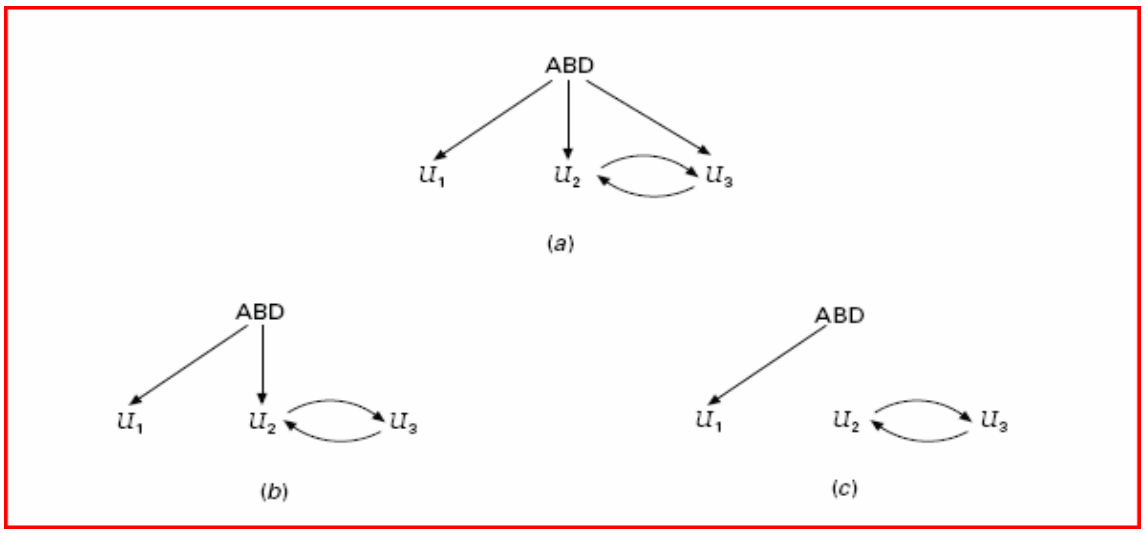
9. ¿Pot el superuser distinguit postgres esborrar un usuari actualment connectat?.

10. ¿Pot el postgres canviar el password a un altre usuari?. . ¿I si està en eixos moments connectat?.

11. ¿Es pot esborrar el superusari postgres

12. ¿I llevar-li poders a postgres?. Per exemple, ¿prohibir-li que puga crear BD?

13. ¿Es pot el.ludir la retirada d'autoritzacions en Postgres exemplificada en la figura següent?



L'ABD dona permís *p2* a U2 with grant option i permís *p3* a U3 tbé with grant option. Posteriorment, U2 transfereix *p2* a U3 i viceversa. L'ABD revoca els permisos a ambdós.

- *¿Poden quedar-se tots dos amb la parella {p2.p3}.*

A. Moll

- *¿Dependrà tot de la clàusula CASCADE de l'ordre REVOKE?.*

14. ¿Com podem alterar la *durabilitat* d'un determinat role espeficada amb ***Valid Until***

15. ¿Quina diferència hi ha entre ENCRYPTED PASSWORD i UNENCRYPTED PASSWORD?