

# DCL

Lenguaje de control de datos

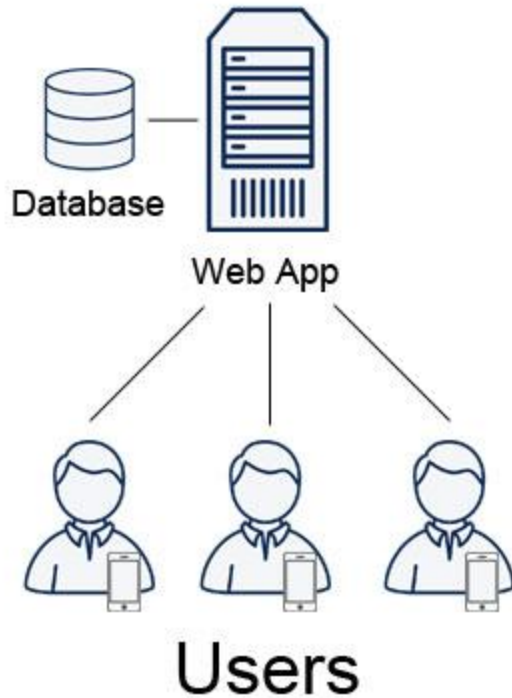


# Data Control Language (DCL)

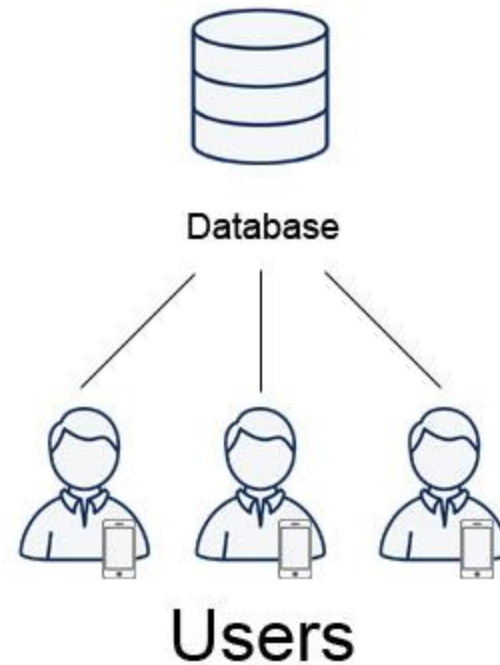
- Permite **controlar el acceso a los objetos**, mediante el uso de:
  - **Usuarios o roles.**
  - **Permisos.**



# App users vs DB users



Only the webapp connects to the database



Every user connects directly to the database

¡mayor interés para administradores que para desarrolladores!

# Usuarios o roles



# Creación de usuarios o roles

## Estándar:

```
> CREATE USER name [ [ WITH ] opciones [ ... ] ]
```

donde opciones pueden ser:

```
SUPERUSER | NOSUPERUSER  
| LOGIN | NOLOGIN  
| CONNECTION LIMIT connlimit  
|[ ENCRYPTED ] PASSWORD 'password'  
| VALID UNTIL 'timestamp'
```



# Creación de usuarios o roles

## Ejemplo PostgreSQL:

```
su postgres - Nos conectamos como administradores  
psql
```

```
\du  
CREATE ROLE usuarioLimitado  
WITH LOGIN PASSWORD 'rymjuegos';
```

# Modificación de usuarios o roles

## Estándar:

```
> ALTER ROLE role_specification [ WITH ] option [ ... ]
```

donde opciones pueden ser:

```
SUPERUSER | NOSUPERUSER  
| LOGIN | NOLOGIN  
| CONNECTION LIMIT connlimit  
|[ ENCRYPTED ] PASSWORD 'password'  
| VALID UNTIL 'timestamp'
```



# Modificación de usuarios o roles

## Ejemplo PostgreSQL:

```
ALTER ROLE usuariolimitado WITH PASSWORD 'hu8jmn3';
```

```
ALTER ROLE usuariolimitado  
VALID UNTIL 'May 4 12:00:00 2015 +1';
```

```
ALTER ROLE usuariolimitado CREATEROLE CREATEDB;
```



# Eliminación de usuarios o roles

## Estándar:

```
> DROP ROLE [ IF EXISTS ] name [, ...]
```

Un rol no puede ser eliminado si algún objeto de la base de datos lo referencia.

## Ejemplo PostgreSQL:

```
DROP ROLE IF EXISTS usuariolimitado;
```

# Privilegios



# GRANT

## Estándar:

```
> GRANT lista_privilegios | ALL  
    ON nombre_tabla  
    TO nombre_role;
```

donde `lista_privilegios` pueden ser:

```
SELECT |  
INSERT |  
UPDATE |  
DELETE |  
TRUNCATE |  
...
```



# Ejemplo GRANT

```
CREATE ROLE usuariolimitado  
WITH LOGIN PASSWORD 'rymjuegos';
```

```
GRANT SELECT ON employees TO  
usuariolimitado;
```

```
psql -U usuariolimitado -W -h 127.0.0.1 hr  
- contraseña rymjuegos
```

```
SELECT * FROM employees;  
- ERROR: permiso denegado a la tabla employees
```

```
SELECT * FROM employees FETCH FIRST 5 ROWS  
ONLY;  
- Muestra 5 primeras filas de employees
```

# Ejemplo GRANT II

```
GRANT SELECT, INSERT, UPDATE, DELETE  
ON jobs TO usuariolimitado;
```

```
GRANT SELECT  
ON ALL TABLES IN SCHEMA "public"  
TO usuariolimitado  
WITH GRANT OPTION;
```

```
INSERT INTO jobs  
VALUES( 'TCH', 'Teacher', 2000, 3000);  
- ERROR: permiso denegado a la tabla jobs
```

```
INSERT INTO jobs  
VALUES( 'TCH', 'Teacher', 2000, 3000);
```

```
DELETE FROM jobs WHERE job_id = 'TCH';
```

# REVOKE

## Estándar:

```
> REVOKE privilegio | ALL  
    ON TABLE nombre_tabla | ALL TABLES  
    IN SCHEMA nombre_esquema  
    FROM nombre_role;
```

# Ejemplo REVOKE

```
REVOKE SELECT ON jobs FROM usuariolimitado;
```

```
REVOKE ALL  
ON ALL TABLES IN SCHEMA public FROM  
usuariolimitado;
```

```
SELECT * FROM jobs;  
- ERROR: permiso denegado a la tabla jobs
```

```
SELECT * FROM employees FETCH FIRST 5 ROWS  
ONLY;  
-ERROR: permiso denegado a la tabla  
employees
```