




QUERY COMMANDS


```
SELECT * FROM [table];
SELECT [atr1], [atr2] AS [nou nom] FROM [taula];
SELECT * FROM [taula] WHERE [atr][comparador][value];
SELECT * FROM [taula] ORDER BY [atr] ASC, [atr] DESC;
SELECT [atr1], [atr2] FROM [taula] ORDER BY 1 ASC, 2 DESC;
SELECT * FROM [table] WHERE [atr] > (SELECT avg ([atr]) FROM [table]);
```

JOIN COMMANDS


 SELECT * FROM [tableL] INNER JOIN [tableR] ON [tableL].[atr]=[tableR].[atr];


 SELECT * FROM [tableL] LEFT JOIN [tableR] USING();


 SELECT * FROM [tableL] RIGH JOIN [tableR] USING();

 SELECT * FROM [tableL] FULL JOIN [tableR] USING();

SELECT * FROM [tableL] INNER JOIN [tableR] USING()
INNER JOIN [table3] ON [tableX].[atr] = [tableZ].[atr]
...
WHERE [condition];

 SELECT * FROM [table_A] CROSS JOIN [table_B];

 SELECT * FROM [taula][UNION | UNION ALL] SELECT * FROM [taula];

 SELECT * FROM [taula1] NATURAL [__|inner|left|right] JOIN [taula2];

SEQUÈNCIES

CREATE SEQUENCE [nom_seq] start with 1000 increment by 1 max value 9999 min value 1 cycle ;	DROP SEQUENCE [nom];
--	-----------------------------

```
CREATE TABLE [nom] (ID nextvalue('[nom_sequencia]'), ...);
```

UPDATE/DELETE EN CASCADA

```
CREATE TABLE [nom] (ID integer reference [taula_orig] ON DELETE CASCADE, ...);
```

```
ALTER TABLE [taula] ADD CONSTRAINT [nom_constr] [taula_conn]  
FOREIGN KEY ([atr_dst]) REFERENCES [taula_dst] ([atr_src])  
ON [UPDATE | DELETE] CASCADE;
```

```
CREATE TABLE clientes_mad (LIKE clientes including ALL);
```

```
INSERT INTO clientes_mad SELECT * FROM clientes WHERE cod_ciudad=5;
```

```
SELECT nombre, poblacion FROM localidades  
WHERE id_localidad NOT IN (SELECT id_localidad FROM cap_prov)  
AND poblacion > 200000 ORDER By poblacion DESC;
```

```
SELECT nombre, apellido FROM clientes WHERE  
(SELECT min(substring(dni,1,8)::int) FROM clientes  
INNER JOIN ciudades using(cod_ciudad)  
GROUP BY cod_ciudad HAVING cod_ciudad=6)>substring(dni,1,8)::int;
```

```
SELECT dni FROM clientes WHERE  
(SELECT avg(substring(dni,1,8)::int) FROM CLIENTES) > substring(dni,1,8)::int;
```

```
SELECT comunidades.nombre,sum(localidades.poblacion) FROM comunidades  
INNER JOIN provincias USING (id_comunidad)  
INNER JOIN cap_prov USING (id_provincia)  
INNER JOIN localidades USING (id_localidad)  
GROUP BY comunidades.nombre, comunidades.poblacion  
HAVING ((sum(localidades.poblacion)*100)/comunidades.poblacion)>45;
```

```
SELECT comunidades.nombre, localidades.nombre FROM comunidades  
INNER JOIN cap_com USING(id_comunidad)  
INNER JOIN localidades USING(id_localidad)  
INNER JOIN provincias USING (id_comunidad)  
WHERE (select count(*) FROM provincias  
WHERE provincias.id_comunidad=comunidades.id_comunidad  
=1;
```

```
SELECT comunidades.nombre, localidades.nombre FROM comunidades  
INNER JOIN cap_com USING(id_comunidad)  
INNER JOIN localidades USING(id_localidad)  
INNER JOIN provincias USING (id_comunidad)  
WHERE (select count(*) FROM provincias  
WHERE provincias.id_comunidad=comunidades.id_comunidad  
)=1;
```