

CC3201-1

BASES DE DATOS

OTOÑO 2018

Clase 6: SQL (II)

Aidan Hogan

aidhog@gmail.com

Forma básica de una consulta de SQL

```
SELECT [atributos]  
FROM [tablas]  
WHERE [condición]
```

Los planetas

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

EL TEMA DE HOY...

... MÁS SQL!

El Álgebra Relacional en SQL

$$\pi_{A_1, \dots, A_n}(R) \quad \sigma_{\text{condición}}(R) \quad \rho_{A_i/A_j}(R)$$

$$R_1 \cup R_2$$

$$R_1 \times R_2$$

$$R_1 - R_2$$

$$R_1 \cap R_2$$

$$R_1 \bowtie_{\text{condición}} R_2$$

¿Hemos visto todo el álgebra en SQL?

¡Sí! (Y un poco más)

- 
- An iceberg floating in the ocean. The visible tip of the iceberg is on the left side of the frame, with a jagged, snow-covered peak. The rest of the iceberg is submerged underwater, appearing as a large, dark blue mass. The word 'SQL' is written in large, dark blue letters across the submerged part of the iceberg. The sky is a clear, light blue with some wispy clouds.
- **SELECT, FROM, WHERE**
 - **ORDER BY**
 - **JOIN** (simple)
 - **UNION, INTERSECT, EXCEPT**
 - **LIKE**
 - **IN, BETWEEN**

SQL

- Más tipos de **JOIN**
- Nulos
- Consultas anidadas
- Agregación

PRODUCTO CRUZ

Cruz: CROSS JOIN

Satélite

nombre	planeta	descubridor	año-des
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, S.planeta, nave
FROM Satélite S CROSS JOIN Aterrizaje
```

```
SELECT nombre, S.planeta, nave
FROM Satélite S, Aterrizaje
```

nombre	S.planeta	nave
Luna	Tierra	Messenger
...	...	...
Luna	Tierra	Galileo
Ganímedes	Júpiter	Messenger
...	...	...
Ganímedes	Júpiter	Galileo
...	...	...

JOINS INTERNOS

Cruzar tablas: JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, año, nave
FROM Planeta, Aterrizaje
WHERE nombre = planeta
AND dist > 1.00
AND año >= 2000
```

nombre	año	nave
Marte	2003	Beagle 2
Júpiter	2003	Galileo

Cruzar tablas: EQUI JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

EQUI JOINS usan sólo '=' en el JOIN

```
SELECT nave, nombre, dist, año
FROM Planeta, Aterrizaje
WHERE nombre = planeta
```

```
SELECT nave, nombre, dist, año
FROM Planeta JOIN Aterrizaje
ON nombre = planeta
```

nave	nombre	dist	año
Messenger	Mercurio	0,39	2015
Venera 3	Venus	0,72	1966
Pioneer	Venus	0,72	1978
Mars 2 lander	Marte	1,52	1971
Viking 1	Marte	1,52	1976
Beagle 2	Marte	1,52	2003
Galileo	Júpiter	5,20	2003

Cruzar tablas: JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

¿Esta consulta es un EQUI JOIN?

```
SELECT nombre, año, nave
FROM Planeta, Aterrizaje
WHERE nombre = planeta
AND dist > 1.00
AND año >= 2000
```

nombre	año	nave
Marte	2003	Beagle 2
Júpiter	2003	Galileo

¡Sí! Sólo la condición del join cuenta.

Cruzar tablas: JOIN USING

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, planeta
FROM Satélite
JOIN Aterrizaje USING (planeta)
```

nombre	planeta
Ganímedes	Júpiter
Calisto	Júpiter
Europa	Júpiter
Ío	Júpiter

Se puede usar JOIN USING cuando todos los atributos del JOIN tengan el mismo nombre

Cruzar tablas: NATURAL JOIN

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Un EQUI-JOIN sobre los atributos que las tablas compartan (por pareja con AND).

```
SELECT nombre, planeta
FROM Satélite
NATURAL JOIN Aterrizaje
```

```
SELECT nombre, planeta
FROM Satélite
JOIN Aterrizaje
    USING (planeta, año)
```

nombre	planeta
--------	---------

Cruzar tablas: SELF JOIN

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Un JOIN sobre la tabla misma

```
SELECT A1.planeta, A2.planeta
FROM Aterrizaje A1
JOIN Aterrizaje A2
ON A1.año = A2.año
AND A1.planeta <> A2.planeta
```

A1.planeta	A2.planeta
Marte	Júpiter
Júpiter	Marte

Cruzar tablas: INNER JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

INNER JOIN *por defecto ...*

```
SELECT nave, nombre, dist, año
FROM Planet INNER JOIN Aterrizaje
ON nombre = planeta
```

```
SELECT nave, nombre, dist, año
FROM Planeta JOIN Aterrizaje
ON nombre = planeta
```

nave	nombre	dist	año
Messenger	Mercurio	0,39	2015
Venera 3	Venus	0,72	1966
Pioneer	Venus	0,72	1978
Mars 2 lander	Marte	1,52	1971
Viking 1	Marte	1,52	1976
Beagle 2	Marte	1,52	2003
Galileo	Júpiter	5,20	2003

JOINS EXTERNOS

Joins Externos

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

¿Todos los planetas (y sus aterrizajes) sí hay datos disponibles?

Joins Externos: LEFT [OUTER] JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Se mantienen las tuplas de la izquierda si no hay datos desde la derecha

```
SELECT nave, nombre, dist, año
FROM Planeta LEFT JOIN Aterrizaje
ON nombre = planeta
```

```
SELECT nave, nombre, dist, año
FROM Planeta LEFT OUTER JOIN Aterrizaje
ON nombre = planeta
```

nave	nombre	dist	año
Messenger	Mercurio	0,39	2015
Venera 3	Venus	0,72	1966
Pioneer	Venus	0,72	1978
Mars 2 lander	Marte	1,52	1971
Viking 1	Marte	1,52	1976
Beagle 2	Marte	1,52	2003
Galileo	Júpiter	1,52	2003
⊥	Tierra	1,00	⊥
⊥	Saturno	9,54	⊥
⊥	Urano	19,19	⊥
⊥	Neptuno	30,07	⊥

Joins Externos: RIGHT [OUTER] JOIN

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Se mantienen las tuplas de la derecha si no hay datos desde la izquierda

```
SELECT nave, nombre, dist, año
FROM Aterrizaje RIGHT JOIN Planeta
ON nombre = planeta
```

```
SELECT nave, nombre, dist, año
FROM Aterrizaje RIGHT OUTER JOIN Planeta
ON nombre = planeta
```

nave	nombre	dist	año
Messenger	Mercurio	0,39	2015
Venera 3	Venus	0,72	1966
Pioneer	Venus	0,72	1978
Mars 2 lander	Marte	1,52	1971
Viking 1	Marte	1,52	1976
Beagle 2	Marte	1,52	2003
Galileo	Júpiter	1,52	2003
⊥	Tierra	1,00	⊥
⊥	Saturno	9,54	⊥
⊥	Urano	19,19	⊥
⊥	Neptuno	30,07	⊥

Joins Externos: FULL OUTER JOIN

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

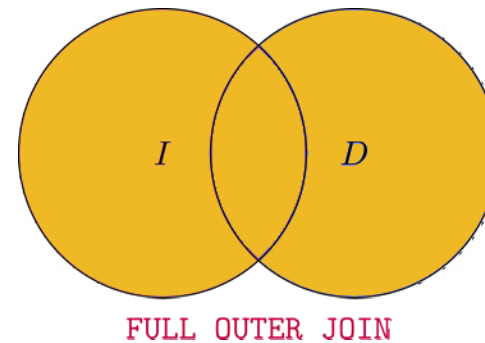
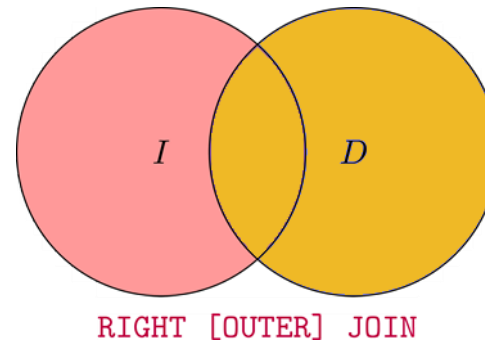
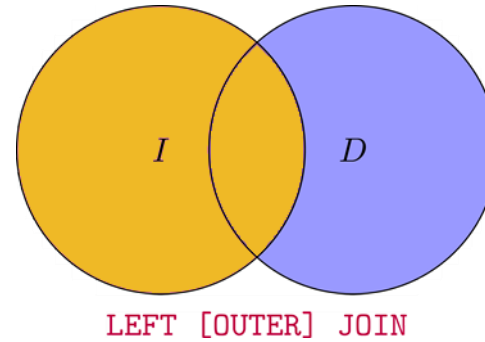
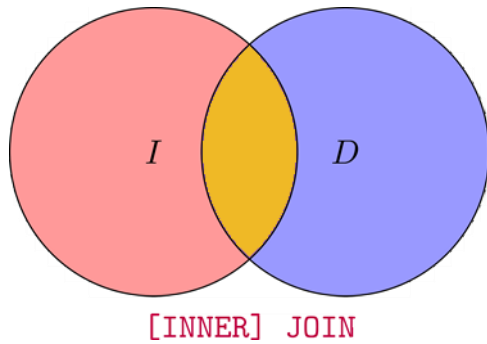
Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Se mantienen las tuplas de la derecha y la izquierda

```
SELECT planeta, nave, nombre AS satélite
FROM Satélite FULL OUTER JOIN Aterrizaje
USING (planeta)
```

planeta	nave	satélite
Tierra	⊥	Luna
Júpiter	Galileo	Ganímedes
Júpiter	Galileo	Calisto
Júpiter	Galileo	Europa
Júpiter	Galileo	Ío
Saturno	⊥	Titán
Neptuno	⊥	Tritón
Mercurio	Messenger	⊥
Venus	Venera 3	⊥
Venus	Pioneer	⊥
Marte	Mars 2 lander	⊥
Marte	Viking 1	⊥
Marte	Beagle 2 lander	⊥

Join Interno versus Joins Externos



[https://es.wikipedia.org/wiki/Null_\(SQL\)](https://es.wikipedia.org/wiki/Null_(SQL))

VALORES NULOS

Nulos

$\perp$, $\emptyset$, $\sqcup$, $\emptyset$, **NULL**

DESCONOCIDO o INAPLICABLE

(No significa FALSO)

Nulos: IS NULL

Satélite			
<u>nombre</u>	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT nombre
FROM Satélite
WHERE descubridor IS NULL
```

<u>nombre</u>
Luna

Nulos: IS NOT NULL

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT nombre
FROM Satélite
WHERE descubridor IS NOT NULL
```

nombre
Ganímedes
Calisto
Europa
Ío
Titán
Tritón

Comparación con nulos

Satélite			
<u>nombre</u>	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT nombre  
FROM Satélite  
WHERE año > 1800
```

<u>nombre</u>
Tritón

Comparación con nulos

Satélite			
<u>nombre</u>	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

¡El nulo en la consulta y el nulo en los datos son distintos!

```
SELECT nombre
FROM Satélite
WHERE año = NULL
```

nombre

Comparación con nulos

p	q	p OR q	p AND q	$p = q$
VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO	FALSO	FALSO
FALSO	VERDADERO	VERDADERO	FALSO	FALSO
FALSO	FALSO	FALSO	FALSO	VERDADERO
VERDADERO	DESCONOCIDO	???		
FALSO	DESCONOCIDO			
DESCONOCIDO	VERDADERO			
DESCONOCIDO	FALSO			
DESCONOCIDO	DESCONOCIDO			

Cuando no importa el valor del desconocido, el resultado se mantiene.
Cuando importa el valor del desconocido, el resultado es desconocido.

Comparación con nulos

p	q	p OR q	p AND q	$p = q$
VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO	FALSO	FALSO
FALSO	VERDADERO	VERDADERO	FALSO	FALSO
FALSO	FALSO	FALSO	FALSO	VERDADERO
VERDADERO	DESCONOCIDO	VERDADERO	DESCONOCIDO	DESCONOCIDO
FALSO	DESCONOCIDO	DESCONOCIDO	FALSO	DESCONOCIDO
DESCONOCIDO	VERDADERO	VERDADERO	DESCONOCIDO	DESCONOCIDO
DESCONOCIDO	FALSO	DESCONOCIDO	FALSO	DESCONOCIDO
DESCONOCIDO	DESCONOCIDO	DESCONOCIDO	DESCONOCIDO	DESCONOCIDO

Cuando no importa el valor del desconocido, el resultado se mantiene.
Cuando importa el valor del desconocido, el resultado es desconocido.

Nulos: COALESCE

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

Elegir el primer valor que no sea **NULL**

```
SELECT nombre, COALESCE(año,0) AS _año
FROM Satélite
ORDER BY _año
```

nombre	_año
Luna	0
Ganímedes	1610
Calisto	1610
Europa	1610
Ío	1610
Titán	1655
Tritón	1846

CONSULTAS ANIDADAS

Consultas Anidadas: WHERE/IN

Planeta

<u>nombre</u>	<u>dist</u>	<u>radio</u>	<u>grav</u>	<u>días</u>	<u>años</u>	<u>temp</u>	<u>anillo</u>
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, planeta
FROM Aterrizaje
WHERE planeta IN
  ( SELECT nombre
    FROM Planeta
    WHERE grav > 9.8 )
AND año > 2000
```

Subconsulta

<u>nave</u>	<u>planeta</u>
Galileo	Júpiter



Consultas Anidadas: WHERE/IN

Planeta

<u>nombre</u>	<u>dist</u>	<u>radio</u>	<u>grav</u>	<u>días</u>	<u>años</u>	<u>temp</u>	<u>anillo</u>
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, planeta
FROM Aterrizaje
WHERE planeta IN
  ( SELECT nombre
    FROM Planeta
    WHERE grav > 9.8 )
AND año > 2000
```

¿Necesitamos una consulta anidada aquí?

```
SELECT nave, P.planeta
FROM Aterrizaje A, Planeta P
WHERE A.planeta=P.nombre
      AND P.grav > 9.8
      AND año > 2000
```



Consultas Anidadas: WHERE/NOT IN

Planeta

<u>nombre</u>	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, planeta
FROM Aterrizaje
WHERE planeta NOT IN
    ( SELECT nombre
      FROM Planeta
      WHERE grav > 9.8 )
AND año > 2000
```

<u>nave</u>	<u>planeta</u>
Beagle 2	Marte
Messenger	Mercurio

Consultas Anidadas: WHERE/NOT IN

Planeta

<u>nombre</u>	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, planeta
FROM Aterrizaje
WHERE planeta NOT IN
( SELECT nombre
  FROM Planeta
  WHERE grav > 9.8 OR planeta IN
    ( SELECT planeta
      FROM Aterrizaje
      WHERE país = 'ESA'
    )
  )
AND año > 2000
```

<u>nave</u>	<u>planeta</u>
Messenger	Mercurio



Consultas Anidadas: WHERE/EXISTS

Planeta

<u>nombre</u>	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, dist
FROM Planeta
WHERE EXISTS
    ( SELECT *
      FROM Aterrizaje
      WHERE año >= 2000 AND nombre = planeta )
ORDER BY dist DESC
```

<u>nombre</u>	<u>dist</u>
Júpiter	5,20
Marte	1,52
Mercurio	0,39

Correlación:

La subconsulta depende de la consulta exterior

Consultas Anidadas: WHERE/NOT EXISTS

Planeta

<u>nombre</u>	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, dist
FROM Planeta
WHERE NOT EXISTS
    ( SELECT *
      FROM Aterrizaje
      WHERE año >= 2000 AND nombre = planeta )
ORDER BY dist DESC
```

<u>nombre</u>	<u>dist</u>
Neptuno	30,07
Urano	19,19
Saturno	9,54
Tierra	1,00
Venus	0,72

Consultas Anidadas: WHERE/(NOT) UNIQUE

Planeta

nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, dist
FROM Planeta
WHERE UNIQUE
    ( SELECT *
      FROM Aterrizaje
      WHERE nombre = planeta )
ORDER BY dist DESC
```

nombre	dist
Neptuno	30,07
Urano	19,19
Saturno	9,54
Júpiter	5,20
Tierra	1,00
Mercurio	0,39

UNIQUE (no suportado por Postgres ☹):

0 o 1 resultados

Consultas Anidadas: WHERE/ANY (o SOME)

Planeta

nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre
FROM Planeta P1
WHERE P1.grav > ANY
    ( SELECT P2.grav
      FROM Planeta P2
      WHERE P2.dist > 1.00 )
ORDER BY P1.dist DESC
```

ANY y SOME son sinónimos

nombre

Neptuno
Urano
Saturno
Júpiter
Tierra
Venus

Consultas Anidadas: WHERE/ALL

Planeta

<u>nombre</u>	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre
FROM Planeta P1
WHERE P1.grav > ALL
    ( SELECT P2.grav
      FROM Planeta P2
      WHERE P2.dist < 1.00 )
ORDER BY P1.dist DESC
```

nombre

Neptuno
Saturno
Júpiter
Tierra



MÁS CONSULTAS ANIDADAS

Consultas Anidadas: Valor

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

```
SELECT nombre
FROM Planeta P1
WHERE P1.grav >
  ( SELECT P2.grav
    FROM Planeta P2
    WHERE P2.nombre = 'Tierra' )
ORDER BY P1.dist DESC
```

nombre
Neptuno
Júpiter

La subconsulta tiene que devolver
un valor y una columna –si no...

Consultas Anidadas: Valor

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

```
SELECT nombre
FROM Planeta P1
WHERE P1.grav >
  ( SELECT P2.grav
    FROM Planeta P2
    WHERE P2.temp > 300 )
ORDER BY P1.dist DESC
```

Error:

La tabla devolió más de una fila

Consultas Anidadas: Valor

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

```
SELECT nombre
FROM Planeta P1
WHERE P1.grav >
  ( SELECT P2.grav, P2.nombre
    FROM Planeta P2
    WHERE P2.nombre = 'Tierra' )
ORDER BY P1.dist DESC
```

Error:

La tabla devolió más de una columna

Consultas Anidadas: Fila

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT S1.nombre, S1.planeta
FROM Satélite S1
WHERE (S1.año, S1.descubridor) =
( SELECT S2.año, S2.descubridor
  FROM Satélite S2
  WHERE S2.nombre = 'Ío' )
```

nombre	planeta
Ío	Júpiter
Calisto	Júpiter
Europa	Júpiter
Ganímedes	Júpiter

Consultas Anidadas: Fila

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	┘	┘
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT S1.nombre, S1.planeta
FROM Satélite S1
WHERE (S1.año, S1.descubridor) =
( SELECT S2.año, S2.descubridor, S2.nombre
  FROM Satélite S2
  WHERE S2.planeta = 'Júpiter' )
```

Error:

La subconsulta devuelve demasiadas columnas

Consultas Anidadas: Fila

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT S1.nombre, S1.planeta
FROM Satélite S1
WHERE (S1.año, S1.descubridor) =
( SELECT S2.año, S2.descubridor
  FROM Satélite S2
  WHERE S2.planeta = 'Júpiter' )
```

Error:

La tabla devolió más de una fila

Consultas Anidadas: Fila

Satélite			
nombre	planeta	descubridor	año
Luna	Tierra	⊥	⊥
Ganímedes	Júpiter	Galileo Galilei	1610
Calisto	Júpiter	Galileo Galilei	1610
Europa	Júpiter	Galileo Galilei	1610
Ío	Júpiter	Galileo Galilei	1610
Titán	Saturno	Christiaan Huygens	1655
Tritón	Neptuno	William Lassell	1846

```
SELECT S1.nombre, S1.planeta
FROM Satélite S1
WHERE (S1.año, S1.descubridor) IN
( SELECT S2.año, S2.descubridor
  FROM Satélite S2
  WHERE S2.planeta = 'Júpiter' )
```

nombre	planeta
Ío	Júpiter
Calisto	Júpiter
Europa	Júpiter
Ganímedes	Júpiter

Consultas Anidadas: FROM

Planeta

nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, grav
FROM
( SELECT A1.planeta
  FROM Aterrizaje A1, Aterrizaje A2
 WHERE A1.planeta=A2.planeta
   AND A1.país<>A2.país ) Multi
Planeta
WHERE nombre=Multi.planeta
  AND grav > 8.0
ORDER BY grav
```

El alias Multi es
obligatorio

nombre	grav
Venus	8.9
Venus	8.9

AGREGACIÓN

Operadores de agregación

- COUNT ([DISTINCT] A)
- SUM ([DISTINCT] A)
- AVG ([DISTINCT] A)
- MAX (A)
- MIN (A)

Agregación: COUNT

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT COUNT(planeta) AS conteo  
FROM Aterrizaje
```

conteo
7

Agregación: COUNT (DISTINCT afuera)

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT DISTINCT COUNT(planeta) AS conteo  
FROM Aterrizaje
```

conteo
7

Agregación: COUNT DISTINCT

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT COUNT(DISTINCT planeta) AS conteo
FROM Aterrizaje
```

<u>conteo</u>
4

Agregación: COUNT(*)

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT COUNT(*) AS conteo  
FROM Aterrizaje
```

conteo
7

Agregación: AVG

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT AVG(año) AS promedio  
FROM Aterrizaje
```

Depende del sistema

conteo

1987,429

Postgres

conteo

1987

Agregación: AVG DISTINCT

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT AVG(DISTINCT año) AS promedio  
FROM Aterrizaje
```

Depende del sistema

conteo

1984,833

Postgres

conteo

1984

conteo

1985

Agregación: AVG (con casting)

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT AVG(CAST(año AS FLOAT)) AS promedio  
FROM Aterrizaje
```

<u>conteo</u>
1987,429

Agregación: MIN

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT MIN(año) AS mínimo
FROM Aterrizaje
```

<u>año</u>
1966

Agregación: MIN

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT MIN(año) AS mínimo, planeta
FROM Aterrizaje
```

Error:

Si hay un operador de agregación solo se puede devolver el resultado de ese operador (o de un operador (GROUP BY))

Agregación: MIN

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT A1.planeta, A1.año
FROM Aterrizaje A1
WHERE A1.año =
    ( SELECT MIN(A2.año)
      FROM Aterrizaje A2
    )
```

planeta	año
Venus	1966

Agregación por planeta: explícitamente

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT A1.planeta, conteo
FROM Aterrizaje A1,
    ( SELECT COUNT(*) AS conteo
      FROM Aterrizaje A2
      WHERE A2.planeta = 'Mercurio'
    ) Mercurio
WHERE A1.planeta = 'Mercurio'
UNION
SELECT A1.planeta, conteo
FROM Aterrizaje A1,
    ( SELECT COUNT(*) AS conteo
      ...
    )
```

planeta	conteo
Mercurio	1
Venus	2
...	...

Agregación por planeta: GROUP BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
```

planeta	conteo
Mercurio	1
Venus	2
Marte	3
Júpiter	1

Agregación por planeta: GROUP BY/HAVING

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
HAVING MAX(año)<2000
```

planeta	conteo
Venus	2

Agregación por planeta: HAVING/EVERY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
HAVING EVERY(año BETWEEN 2000 AND 2005)
```

planeta	conteo
Júpiter	1

Agregación por planeta: HAVING/ANY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
HAVING ANY(año BETWEEN 2000 AND 2005)
```

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
HAVING bool_or(año BETWEEN 2000 AND 2005)
```

Postgres

planeta	conteo
Júpiter	1
Marte	3

Más detalles:

[https://en.wikipedia.org/wiki/Select_\(SQL\)#Limiting_result_rows](https://en.wikipedia.org/wiki/Select_(SQL)#Limiting_result_rows)

LIMITAR RESULTADOS

Sistemas de bases de datos (con SQL)

123 systems in ranking, October 2016

Rank			DBMS	Database Model	Score		
Oct 2016	Sep 2016	Oct 2015			Oct 2016	Sep 2016	Oct 2015
1.	1.	1.	Oracle	Relational DBMS	1417.10	-8.46	-49.85
2.	2.	2.	MySQL +	Relational DBMS	1362.65	+8.62	+83.69
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1214.18	+2.62	+90.95
4.	4.	4.	PostgreSQL	Relational DBMS	318.69	+2.34	+36.56
5.	5.	5.	DB2	Relational DBMS	180.56	-0.62	-26.25
6.	6.	6.	Microsoft Access	Relational DBMS	124.68	+1.36	-17.16
7.	7.	7.	SQLite	Relational DBMS	108.57	-0.05	+5.90
8.	8.	↑ 9.	Teradata	Relational DBMS	76.23	+3.17	+2.79
9.	9.	↓ 8.	SAP Adaptive Server	Relational DBMS	69.48	+0.32	-16.16
10.	10.	↑ 11.	FileMaker	Relational DBMS	54.95	-0.40	+5.17
11.	11.	↓ 10.	Hive	Relational DBMS	49.20	+0.38	-4.36
12.	12.	12.	SAP HANA +	Relational DBMS	45.77	+2.35	+6.67
13.	13.	↑ 14.	MariaDB	Relational DBMS	40.28	+1.74	+15.65
14.	14.	↓ 13.	Informix	Relational DBMS	39.66	-0.13	-0.12
15.	15.	↑ 17.	Vertica	Relational DBMS	36.56	+2.34	+36.56
16.	↑ 17.	↑ 18.	Microsoft Azure SQL	Relational DBMS	31.87	+2.34	+36.56
17.	↓ 16.	↓ 16.	Netezza	Relational DBMS	26.25	-0.62	-26.25
18.	18.	↓ 15.	Firebird	Relational DBMS	23.17	+3.17	+2.79
19.	19.	↑ 22.	Amazon Redshift	Relational DBMS	22.79	+3.17	+2.79
20.	20.	↑ 21.	dBASE	Relational DBMS	21.16	+0.32	-16.16

*¡Varios sistemas pueden tener varias interpretaciones del estándar de SQL!
Pero normalmente el “core” de SQL es compatible en los sistemas más populares.*

Ordenar resultados: ORDER BY [DESC|ASC]

Aterrizaje

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT *  
FROM Aterrizaje  
ORDER BY año DESC, nave
```

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003
Pioneer	Venus	EEUU	1978
Viking 1	Marte	EEUU	1976
Mars 2 lander	Marte	URRS	1971
Venera 3	Venus	URRS	1966

Devolver n resultados: FETCH FIRST

Aterrizaje

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión estándar (desde SQL:2008) que se usa en Postgres y DB2.

```
SELECT *  
FROM Aterrizaje  
ORDER BY año DESC, nave  
FETCH FIRST 3 ROWS ONLY
```

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Devolver n resultados: LIMIT

Aterrizaje			
nave	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión no estándar que se usa en Postgres, SQLite y MySQL.

```
SELECT *  
FROM Aterrizaje  
ORDER BY año DESC, nave  
LIMIT 3
```

nave	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Devolver n resultados: TOP

Aterrizaje

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión no estándar que se usa en SQL Server y MS Access.

```
SELECT TOP 3 *  
FROM Aterrizaje  
ORDER BY año DESC, nave
```

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Devolver n resultados: ROW_NUMBER()

Aterrizaje

nave	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión estándar (desde SQL:2003) que se usa en Postgres, DB2, MS Access, Oracle

```
SELECT * FROM (  
    SELECT ROW_NUMBER()  
        OVER (ORDER BY año DESC, nave)  
        AS row, *  
    FROM Aterrizaje  
) AS Ans  
WHERE row <= 3
```

row	nave	planeta	pais	año
1	Messenger	Mercurio	EEUU	2015
2	Beagle 2	Marte	ESA	2003
3	Galileo	Júpiter	EEUU	2003

Devolver empates: RANK()

Aterrizaje

nave	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión estándar (desde SQL:2003) que devuelva *empates* en el orden.

```
SELECT * FROM (  
  SELECT RANK()  
    OVER (ORDER BY año DESC)  
  AS rnk, *  
  FROM Aterrizaje  
) AS Ans  
WHERE rnk <= 2
```

rnk	nave	planeta	pais	año
1	Messenger	Mercurio	EEUU	2015
2	Beagle 2	Marte	ESA	2003
2	Galileo	Júpiter	EEUU	2003

Saltar n resultados: LIMIT + OFFSET

Aterrizaje

<u>nave</u>	planeta	pais	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Una versión no estándar que se usa en Postgres, SQLite y MySQL.

```
SELECT *  
FROM Aterrizaje  
ORDER BY año DESC, nave  
OFFSET 1 LIMIT 3
```

<u>nave</u>	planeta	pais	año
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003
Pioneer	Venus	EEUU	1978

MÁS FUNCIONES

¡Dependen mucho del sistema particular!

Aritmético

$+$, $-$, $/$, $*$, $\%$

ABS(a)

CEIL(a) o **CEILING**(a)

FLOOR(a)

EXP(a, b) o **POWER**(a, b)

ROUND(a) o **ROUND**(a, b)

SQRT(a)

...

Aritmético

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

```
SELECT nombre,  
       ABS(dist-1.0) AS distDeTierra  
FROM Planeta  
ORDER BY distDeTierra
```

nombre	distDeTierra
Tierra	0,00
Venus	0,28
Martes	0,52
Mercurio	0,61
Júpiter	4,20
Saturno	8,54
Urano	18,19
Neptuno	29,07

Strings

LOWER(a) o LOWERCASE(a) o LCASE(a)

UPPER(a) o UPPERCASE(a) o UCASE(a)

TRIM(a)

SUBSTRING(a,b) o SUBSTRING(a,b,c)

STARTSWITH(a,b)

...

Condicionales

```
IF ... THEN ... [ ELSE IF ... ] * [ELSE]  
CASE ... [WHEN ... THEN ...] * [ELSE ...]  
...
```

CONSULTAS DIRECTAS VS. CONSULTAS ANIDADAS

SQL tiene mucha redundancia



Salmón ahumado

El salmón ahumado es un producto ahumado del salmón



Consultas directas vs. consultas anidadas

Nombres y géneros de los co-actores de Liv Tyler.

(1) Selección/producto:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A,
     personaje P1, personaje P2
WHERE P1.a_nombre='Tyler, Liv'
      AND P1.p_nombre = P2.p_nombre
      AND P1.p_anho = P2.p_anho
      AND A.nombre = P2.a_nombre
```

(2) Join explícito:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A NATURAL JOIN
  ( SELECT DISTINCT P2.a_nombre AS nombre
    FROM personaje P2 NATURAL JOIN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      ) PLT
    ) CLT
```

(3) Consulta anidada (FROM):

```
SELECT DISTINCT A.nombre, A.genero FROM
  ( SELECT DISTINCT P2.a_nombre FROM
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre='Tyler, Liv'
    ) PLT, personaje P2
    WHERE PLT.p_nombre = P2.p_nombre
      AND PLT.p_anho = P2.p_anho
  ) CLT, actor A
WHERE CLT.a_nombre = A.nombre
```

(4) Consulta anidada (WHERE/IN):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
  ( SELECT DISTINCT P2.a_nombre
    FROM personaje P2
    WHERE (P2.p_nombre,P2.p_anho) IN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      )
  )
```

(5) [Hay más opciones] | ¿Son equivalentes pero cuál es más eficiente?

Consultas directas vs. consultas anidadas

Nombres y géneros de los co-actores de Liv Tyler.

(1) Selección/producto:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A,
     personaje P1, personaje P2
WHERE P1.a_nombre='Tyler, Liv'
     AND P1.p_nombre = P2.p_nombre
     AND P1.p_anho = p2.p_anho
     AND A.nombre = P2.a_nombre
```

10 ms

(2) Join explícito:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A NATURAL JOIN
  ( SELECT DISTINCT P2.a_nombre AS nombre
    FROM personaje P2 NATURAL JOIN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      ) PLT
    ) CLT
```

9 ms

(3) Consulta anidada (FROM):

```
SELECT DISTINCT A.nombre, A.genero FROM
  ( SELECT DISTINCT P2.a_nombre FROM
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre='Tyler, Liv'
    ) PLT, personaje P2
    WHERE PLT.p_nombre = P2.p_nombre
      AND PLT.p_anho = P2.p_anho
  ) CLT, actor A
WHERE CLT.a_nombre = A.nombre
```

11 ms

(4) Consulta anidada (WHERE/IN):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
  ( SELECT DISTINCT P2.a_nombre
    FROM personaje P2
    WHERE (P2.p_nombre,P2.p_anho) IN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      )
    )
```

12 ms

¡Hay poca diferencia!

Consultas directas vs. consultas anidadas

Nombres y géneros de co-actores de personas con una apellida "L%".

(1) Selección/producto:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A,
     personaje P1, personaje P2
WHERE P1.a_nombre LIKE '%, L%'
      AND P1.p_nombre = P2.p_nombre
      AND P1.p_anho = P2.p_anho
      AND A.nombre = P2.a_nombre
```

160 ms

(2) Join explícito:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A NATURAL JOIN
  ( SELECT DISTINCT P2.a_nombre AS nombre
    FROM personaje P2 NATURAL JOIN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre LIKE '%, L%'
      ) PLT
  ) CLT
```

169 ms

(3) Consulta anidada (FROM):

```
SELECT DISTINCT A.nombre, A.genero FROM
  ( SELECT DISTINCT P2.a_nombre FROM
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre LIKE '%, L%'
    ) PLT, personaje P2
    WHERE PLT.p_nombre = P2.p_nombre
      AND PLT.p_anho = P2.p_anho
  ) CLT, actor A
WHERE CLT.a_nombre = A.nombre
```

167 ms

(4) Consulta anidada (WHERE/IN):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
  ( SELECT DISTINCT P2.a_nombre
    FROM personaje P2
    WHERE (P2.p_nombre, P2.p_anho) IN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre LIKE '%, L%'
      )
  )
```

48 ms

¡Hay una diferencia (pero es poco predecible)!

SQL es un lenguaje **declarativo**

Uno dice lo que quiere, no cómo se debería computar

(1) Selección/producto:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A,
     personaje P1, personaje P2
WHERE P1.a_nombre='Tyler, Liv'
      AND P1.p_nombre = P2.p_nombre
      AND P1.p_anho = p2.p_anho
      AND A.nombre = P2.a_nombre
```

(2) Join explícito:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A NATURAL JOIN
  ( SELECT DISTINCT P2.a_nombre AS nombre
    FROM personaje P2 NATURAL JOIN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      ) PLT
    ) CLT
```

(3) Consulta anidada (FROM):

```
SELECT DISTINCT A.nombre, A.genero FROM
  ( SELECT DISTINCT P2.a_nombre FROM
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre='Tyler, Liv'
    ) PLT, personaje P2
    WHERE PLT.p_nombre = P2.p_nombre
      AND PLT.p_anho = P2.p_anho
  ) CLT, actor A
WHERE CLT.a_nombre = A.nombre
```

(4) Consulta anidada (WHERE/IN):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
  ( SELECT DISTINCT P2.a_nombre
    FROM personaje P2
    WHERE (P2.p_nombre,P2.p_anho) IN
      ( SELECT DISTINCT P1.p_nombre, P1.p_anho
        FROM personaje P1
        WHERE P1.a_nombre='Tyler, Liv'
      )
  )
```

Caja Negra



nombre	genero
Abbott, Jane (II)	F
Acevedo, Gino (I)	M
Allpress, Bruce	M
Appleby, Noel	M
Appleton, Matt (I)	M
Astin, Ali	F
Astin, Sean	M
Aston, David (I)	M
Bach, John (I)	M
Baker, Sala	M
Bartlett, Timothy	M
Bean, Sean	M
Benzon, Jarl	M
Benzon, Jørn	M
Beynon-Cole, Victoria	F
Blanchett, Cate	F
Bloom, Orlando	M
Boyd, Billy (I)	M
Boyens, Callum	M
Britton, Ben (I)	M
Britton, Ben (VI)	M
Brophy, Jed	M
Brophy, Riley	M
Brophy, Sadwyn	M
Browning, Alistair	M
Bryson, Paul (I)	M
Burnyeat, Luke	M
Clentworth, Rachel	F
Comery, Sam	M
Corrigan, Peter (II)	M
Crossen, Sabine	F

SQL es un lenguaje **declarativo**

Uno dice lo que quiere, no cómo se debería computar

— *Idealmente*, el motor puede elegir el mejor plan de ejecución independientemente de su expresión particular

Caja Negra

(2) Join explícito:

- Pero, esto es caro, entonces *en la práctica*, hay diferencias

— Regresaremos al tema de rendimiento y optimización más adelante en el curso

— Pero en general, se puede expresar una consulta en la forma “más natural” y dejar la ejecución al motor

(1) Selección/proyección

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A,
     personaje P1, personaje P2
WHERE P1.a_nombre=A.nombre
AND P1.p_nombre = A.nombre
AND P1.p_anho = A.p_anho
AND A.nombre = P2.a_nombre
```

(2) Join explícito:

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A NATURAL JOIN
( SELECT DISTINCT P2.a_nombre, P2.p_nombre, P2.p_anho
  FROM personaje P2 NATURAL JOIN
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre='Tyler, Liv'
    ) PLT
  ) CLT
```

(3) Consulta anidada (nested):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
( SELECT DISTINCT A.nombre
  FROM personaje P
  WHERE P1.a_nombre='Tyler, Liv'
  ) PLT, personaje P
WHERE PLT.p_nombre = A.nombre
AND PLT.p_anho = A.p_anho
) CLT, actor A
WHERE CLT.a_nombre = A.nombre
```

(4) Consulta anidada (WHERE...IN):

```
SELECT DISTINCT A.nombre, A.genero
FROM actor A
WHERE A.nombre IN
( SELECT DISTINCT A.nombre
  FROM personaje P2
  WHERE (P2.p_nombre, P2.p_anho) IN
    ( SELECT DISTINCT P1.p_nombre, P1.p_anho
      FROM personaje P1
      WHERE P1.a_nombre='Tyler, Liv'
    )
  )
```

nombre	genero
Abbott, Jane (II)	F
Acavedo, Gina	M
Appelby, Noel	M
Appleton, Matt (I)	M
Astin, Ali	F
Astin, Sean	M
Aston, David (I)	M
Ashby, Ian (I)	M
Bartlett, Timothy	M
Bean, Sean	M
Benson, Jarl	M
Benson, Jarn	M
Benson-Cole, Victoria	F
Blackchett, Cate	F
Bloom, Orlando	M
Blyd, Billy (I)	M
Boydens, Callum	M
Britton, Ben (I)	M
Bryson, Paul (I)	M
Burnyeat, Luke	M
Clentworth, Rachel	F
Comery, Sam	M
Corrigan, Peter (II)	M
Crossen, Sabine	F

- 
- The background of the slide is a large iceberg floating in the ocean. The tip of the iceberg, which is visible above the water, is white and jagged. The much larger part of the iceberg is submerged underwater, appearing as a dark blue mass. The word 'SQL' is written in large, light blue letters across the submerged part of the iceberg. In the top right corner, there is a list of SQL keywords. In the bottom right corner, there is a small inset image of a man and a woman in a close embrace, with the man looking down at the woman.
- **SELECT, FROM, WHERE**
 - **ORDER BY**
 - **JOIN** (simple)
 - **UNION, INTERSECT, EXCEPT**
 - **LIKE**
 - **IN, BETWEEN**

SQL

- Más tipos de **JOIN**
- Nulos
- Consultas anidadas
- Agregación



Preguntas?

