

CC3201-1

BASES DE DATOS

OTOÑO 2025

Clase 6: SQL (II)

Aidan Hogan

aidhog@gmail.com

LA ÚLTIMA VEZ ...

El Álgebra Relacional en SQL

R

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

$$\mathbf{R}_1 \cup \mathbf{R}_2$$

$$\mathbf{R}_1 \times \mathbf{R}_2$$

$$\mathbf{R}_1 - \mathbf{R}_2$$

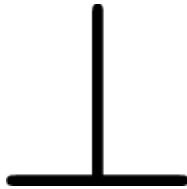
$$\mathbf{R}_1 \cap \mathbf{R}_2$$

$$\mathbf{R}_1 \bowtie_{\text{condición}} \mathbf{R}_2$$

Aún más allá del Álgebra Relacional en SQL



$f(\cdot)$

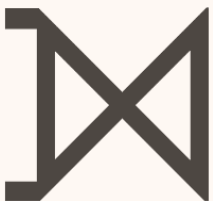
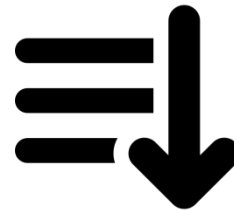
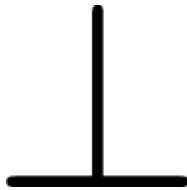


HOY ...

Aún más allá del Álgebra Relacional en SQL



$f(\cdot)$



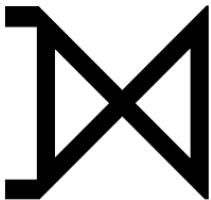
Los planetas

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

JOINS EXTERNOS



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

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 si 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

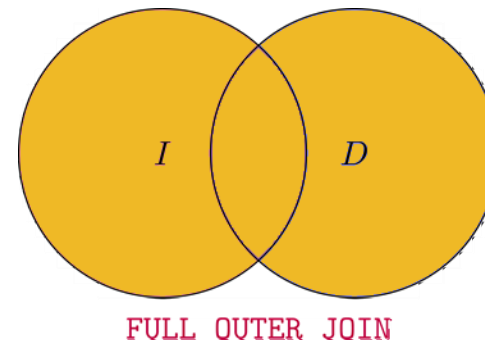
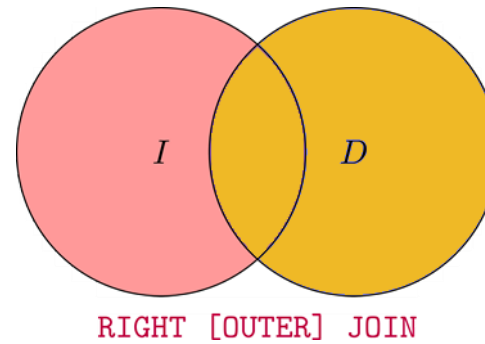
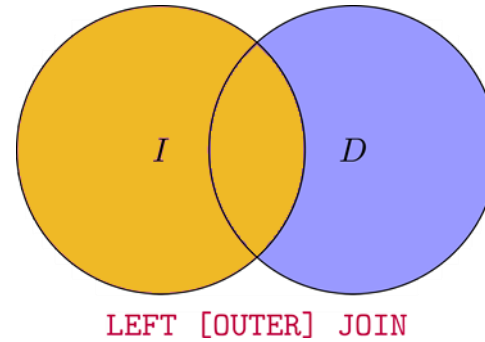
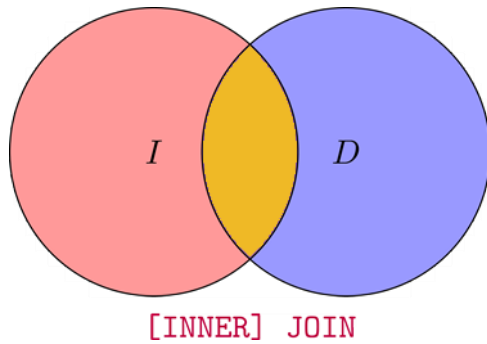
Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Se mantienen las tuplas de la derecha y de 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



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
```

¡No!

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							
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 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
```

nave	planeta
Messenger	Mercurio



Consultas Anidadas: WHERE/EXISTS

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

nombre	dist
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>	<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 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/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
```

nombre

Neptuno
Urano
Saturno
Júpiter
Tierra
Venus

ANY y SOME son sinónimos

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

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 devolvió 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
```

Un alias (como Multi)
Es obligatorio

nombre	grav
Venus	8.9
Venus	8.9

AGREGACIONES



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
Venera 4	Venus	URRS	1967
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
8

Agregación: COUNT (DISTINCT afuera)

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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

8



Agregación: COUNT DISTINCT

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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
```

conteo

4

Agregación: COUNT(*)

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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
8

Agregación: COUNT (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 COUNT(descubridor) AS conteo
FROM Satélite
```

Se ignoran los nulos

conteo
6

Agregación: COUNT (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 COUNT(descubridor) AS conteo
FROM Satélite
WHERE nombre = 'Luna'
```

conteo
0

Agregación: AVG

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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

promedio

1984,875

Postgres

promedio

1985

promedio

1984

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
Venera 4	Venus	URRS	1967
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

promedio

1982,285

Postgres

promedio

1982

Agregación: AVG (con casting)

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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
```

promedio

1984,875

Agregación: MIN

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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>mínimo</u>
1966

Agregación: MIN

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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
Venera 4	Venus	URRS	1967
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

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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	3
...	...

Agregación por planeta: GROUP BY

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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	3
Marte	3
Júpiter	1

Agregación por planeta: GROUP BY múltiple

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta, país
```

planeta	país	conteo
Mercurio	EEUU	1
Venus	URRS	2
Venus	EEUU	1
Marte	URRS	1
Marte	EEUU	1
Marte	ESA	1
Júpiter	EEUU	1

Agregación por planeta: GROUP BY/HAVING

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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	3

Agregación por planeta: HAVING/EVERY

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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)
```

<u>planeta</u>	<u>conteo</u>
Júpiter	1

Agregación por planeta: HAVING/ANY

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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

FUNCIONES DE VENTANA



Agregación

Aterrizaje							
nave	planeta	país	año		planeta	$A_1(\cdot)$	$\dots$ $A_n(\cdot)$
Messenger	Mercurio	EEUU	2015		Mercurio	...	...
Venera 3	Venus	URRS	1966		Venus	...	...
Venera 4	Venus	URRS	1967				
Pioneer	Venus	EEUU	1978				
Mars 2 lander	Marte	URRS	1971		Marte	...	...
Viking 1	Marte	EEUU	1976				
Beagle 2	Marte	ESA	2003				
Galileo	Júpiter	EEUU	2003		Júpiter	...	...

Agregación: Hay un resultado para cada grupo de entrada.

Se calculan las columnas nuevas usando los valores de cada grupo.

Ventanas

Ventanas: Hay un resultado para cada resultado de entrada.

Se calculan las columnas nuevas usando los valores de cada grupo.

Aterrizaje											
nave	planeta	país	año		nave	planeta	país	año	$V_1(\cdot)$	$\dots$	$V_n(\cdot)$
Messenger	Mercurio	EEUU	2015	→	Messenger	Mercurio	EEUU	2015	...	...	...
Venera 3	Venus	URRS	1966	→	Venera 3	Venus	URRS	1966	...	...	...
Venera 4	Venus	URRS	1967	→	Venera 4	Venus	URRS	1967	...	...	...
Pioneer	Venus	EEUU	1978	→	Pioneer	Venus	EEUU	1978	...	...	...
Mars 2 lander	Marte	URRS	1971	→	Mars 2 lander	Marte	URRS	1971	...	...	...
Viking 1	Marte	EEUU	1976	→	Viking 1	Marte	EEUU	1976	...	...	...
Beagle 2	Marte	ESA	2003	→	Beagle 2	Marte	ESA	2003	...	...	...
Galileo	Júpiter	EEUU	2003	→	Galileo	Júpiter	EEUU	2003	...	...	...

Agregación vs. Ventanas

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

planeta	$A_1(\cdot)$	...	$A_n(\cdot)$
Mercurio	...	...	...
Venus	...	...	...
Marte	...	...	...
Júpiter	...	...	...

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

nave	planeta	país	año	$V_1(\cdot)$	...	$V_n(\cdot)$
Messenger	Mercurio	EEUU	2015	...	...	...
Venera 3	Venus	URRS	1966	...	...	...
Venera 4	Venus	URRS	1967	...	...	...
Pioneer	Venus	EEUU	1978	...	...	...
Mars 2 lander	Marte	URRS	1971	...	...	...
Viking 1	Marte	EEUU	1976	...	...	...
Beagle 2	Marte	ESA	2003	...	...	...
Galileo	Júpiter	EEUU	2003	...	...	...

Operadores de agregación (en ventanas)

- COUNT (A)
- SUM (A)
- AVG (A)
- MAX (A)
- MIN (A)

Agregación: GROUP BY / MIN

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, MIN(año) AS primero
FROM Aterrizaje
GROUP BY planeta
```

planeta	primero
Mercurio	2015
Venus	1966
Marte	1971
Júpiter	2003

Ventana: PARTITION BY / MIN

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, MIN(año) OVER (PARTITION BY planeta) AS primero
FROM Aterrizaje
```

nave	planeta	año	primero
Messenger	Mercurio	2015	2015
Venera 3	Venus	1966	1966
Venera 4	Venus	1967	1966
Pioneer	Venus	1978	1966
Mars 2 lander	Marte	1971	1971
Viking 1	Marte	1976	1971
Beagle 2	Marte	2003	1971
Galileo	Júpiter	2003	2003

Ventana: PARTITION BY / MAX múltiple

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, MAX(año) OVER (PARTITION BY planeta, país) AS último
FROM Aterrizaje
```

<u>nave</u>	planeta	país	último
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1967
Venera 4	Venus	URRS	1967
Pioneer	Venus	EEUU	1978
Mars 2 lander	URRS	1971	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

Ventana: OVER vacío

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, AVG(año) OVER () AS promedio
FROM Aterrizaje
```

Si **OVER** está vacío, la ventana es la tabla entera.

Tenemos que poner **OVER** igual.

nave	año	promedio
Messenger	2015	1984,875
Venera 3	1966	1984,875
Venera 4	1967	1984,875
Pioneer	1978	1984,875
Mars 2 lander	1971	1984,875
Viking 1	1976	1984,875
Beagle 2	2003	1984,875
Galileo	2003	1984,875

Ventana: PARTITION BY / COUNT DISTINCT

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, COUNT(DISTINCT país) OVER (PARTITION BY planeta) AS países
FROM Aterrizaje
```

Depende del sistema.
En Postgres, no se puede.

Error:

No se permite DISTINCT en ventanas

Funciones de ventana: Rank

Donde k es el número de elementos en la ventana:

- **ROW_NUMBER()** Da un número $\in [1, k]$
- **RANK()** Da un número $\in [1, k]$ con empates y brechas
- **DENSE_RANK()** Da un número $\in [1, k]$ con empates, sin brechas
- **PERCENT_RANK()** Da $\frac{\text{RANK}() - 1}{k - 1} \in [0, 1]$
- **CUME_DIST()** Da $\frac{\text{RANK}()}{k} \in [\frac{1}{k}, 1]$
- **NTILE(n)** Da el n -tile $\in [1, n]$ (p.ej., $n = 4$ daría cuartiles)

Ventana: ROW NUMBER

Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, ROW_NUMBER() OVER () AS num
FROM Aterrizaje
```

<u>nave</u>	<u>num</u>
Messenger	1
Venera 3	2
Venera 4	3
Pioneer	4
Mars 2 lander	5
Viking 1	6
Beagle 2	7
Galileo	8

Ventana: ROW NUMBER / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, ROW_NUMBER() OVER (ORDER BY año) AS num
FROM Aterrizaje
```

nave	año	num
Venera 3	1966	1
Venera 4	1967	2
Mars 2 lander	1971	3
Viking 1	1976	4
Pioneer	1978	5
Beagle 2	2003	6
Galileo	2003	7
Messenger	2015	8

Ventana: RANK / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, RANK() OVER (ORDER BY año) AS num
FROM Aterrizaje
```

nave	año	num
Venera 3	1966	1
Venera 4	1967	2
Mars 2 lander	1971	3
Viking 1	1976	4
Pioneer	1978	5
Beagle 2	2003	6
Galileo	2003	6
Messenger	2015	8

Ventana: DENSE_RANK / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, DENSE_RANK() OVER (ORDER BY año) AS num
FROM Aterrizaje
```

nave	año	num
Venera 3	1966	1
Venera 4	1967	2
Mars 2 lander	1971	3
Viking 1	1976	4
Pioneer	1978	5
Beagle 2	2003	6
Galileo	2003	6
Messenger	2015	7

Ventana: PERCENT_RANK / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, PERCENT_RANK() OVER (ORDER BY año) AS perc  
FROM Aterrizaje
```

nave	año	perc
Venera 3	1966	0
Venera 4	1967	0,142
Mars 2 lander	1971	0,286
Viking 1	1976	0,429
Pioneer	1978	0,571
Beagle 2	2003	0,714
Galileo	2003	0,714
Messenger	2015	1

Ventana: CUME_DIST / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, CUME_DIST() OVER (ORDER BY año) AS cdist
FROM Aterrizaje
```

nave	año	cdist
Venera 3	1966	0,125
Venera 4	1967	0,25
Mars 2 lander	1971	0,375
Viking 1	1976	0,5
Pioneer	1978	0,625
Beagle 2	2003	0,75
Galileo	2003	0,75
Messenger	2015	1

Ventana: NTILE / ORDER BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, NTILE(4) OVER (ORDER BY año) AS cuartil
FROM Aterrizaje
```

En Postgres, valores con
empates pueden estar en
ntiles diferentes.

nave	año	cuartil
Venera 3	1966	1
Venera 4	1967	1
Mars 2 lander	1971	2
Viking 1	1976	2
Pioneer	1978	3
Beagle 2	2003	3
Galileo	2003	4
Messenger	2015	4

Ventana: RANK / ORDER BY desc., múltiple

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, año, RANK() OVER (ORDER BY año DESC, país) AS num
FROM Aterrizaje
```

nave	año	num
Messenger	2015	1
Galileo	2003	2
Beagle 2	2003	3
Pioneer	1978	4
Viking 1	1976	5
Mars 2 lander	1971	6
Venera 4	1967	7
Venera 3	1966	8

Ventana: RANK / ORDER BY / PARTITION BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año, RANK() OVER (PARTITION BY país ORDER BY año) AS num
FROM Aterrizaje
ORDER BY país
```

nave	país	año	num
Viking 1	EEUU	1976	1
Pioneer	EEUU	1978	2
Galileo	EEUU	2003	3
Messenger	EEUU	2015	4
Beagle 2	ESA	2003	1
Venera 3	URRS	1966	1
Venera 4	URRS	1967	2
Mars 2 lander	URRS	1971	3

Funciones de ventana: Posición

Con respecto a cada ventana:

- **FIRST_VALUE**(**A**) Da el primer valor
- **LAST_VALUE**(**A**) Da el último valor
- **NTH_VALUE**(**A**, n) Da el n -ésimo valor
- **LAG**(**A**) Da el valor previo
- **LEAD**(**A**) Da el próximo valor

Si no existe el valor, da **NULL**.

Ventana: FIRST_VALUE / ORDER BY / PARTITION BY

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       FIRST_VALUE(año) OVER (PARTITION BY país ORDER BY año) AS primero  
FROM Aterrizaje  
ORDER BY país
```

nave	país	año	primero
Viking 1	EEUU	1976	1976
Pioneer	EEUU	1978	1976
Galileo	EEUU	2003	1976
Messenger	EEUU	2015	1976
Beagle 2	ESA	2003	2003
Venera 3	URRS	1966	1966
Venera 4	URRS	1967	1966
Mars 2 lander	URRS	1971	1966

Ventana: LAST_VALUE / ORDER BY / PARTITION BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       LAST_VALUE(nave) OVER (PARTITION BY país ORDER BY año) AS último  
FROM Aterrizaje  
ORDER BY país
```

Por defecto, solo considera las filas hasta la fila actual (inclusive) en la ventana y sus empates en el orden especificado.

¿Y si quisiéramos el valor máximo de la ventana?

nave	país	año	último
Viking 1	EEUU	1976	Viking 1
Pioneer	EEUU	1978	Pioneer
Galileo	EEUU	2003	Galileo
Messenger	EEUU	2015	Messenger
Beagle 2	ESA	2003	Beagle 2
Venera 3	URRS	1966	Venera 3
Venera 4	URRS	1967	Venera 4
Mars 2 lander	URRS	1971	Mars 2 lander

Ventana: FIRST_VALUE / ORDER BY / PARTITION BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       FIRST_VALUE(nave) OVER (PARTITION BY país ORDER BY año DESC) AS último  
FROM Aterrizaje  
ORDER BY país
```

nave	país	año	último
Viking 1	EEUU	1976	Messenger
Pioneer	EEUU	1978	Messenger
Galileo	EEUU	2003	Messenger
Messenger	EEUU	2015	Messenger
Beagle 2	ESA	2003	Beagle 2
Venera 3	URRS	1966	Mars 2 lander
Venera 4	URRS	1967	Mars 2 lander
Mars 2 lander	URRS	1971	Mars 2 lander

Ventana: NTH_VALUE / ORDER BY / PARTITION BY

Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       NTH_VALUE(nave,2) OVER (PARTITION BY país ORDER BY año) AS segundo  
FROM Aterrizaje  
ORDER BY país
```

Por defecto, solo considera las filas hasta la fila actual (inclusive) en la ventana y sus empates en el orden especificado.

nave	país	año	segundo
Viking 1	EEUU	1976	⊥
Pioneer	EEUU	1978	Pioneer
Galileo	EEUU	2003	Pioneer
Messenger	EEUU	2015	Pioneer
Beagle 2	ESA	2003	⊥
Venera 3	URRS	1966	⊥
Venera 4	URRS	1967	Venera 4
Mars 2 lander	URRS	1971	Venera 4

Ventana: Su rango

Se define la ventana con **RANGE BETWEEN** y:

- **UNBOUNDED PRECEDING** Desde la primera tupla de la partición
- **UNBOUNDED FOLLOWING** Hasta la última tupla de la partición
- **CURRENT ROW** Desde/hasta la tupla actual
- n **PRECEDING** Desde n tuplas antes de la tupla actual
- n **FOLLOWING** Hasta n tuplas luego de la tupla actual

Ventana: LAST_VALUE / ORDER BY / PARTITION BY

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       LAST_VALUE(nave) OVER  
       ( PARTITION BY país ORDER BY año  
         RANGE BETWEEN  
           UNBOUNDED PRECEDING AND  
           UNBOUNDED FOLLOWING  
       ) AS último  
FROM Aterrizaje  
ORDER BY país
```

nave	país	año	último
Viking 1	EEUU	1976	Messenger
Pioneer	EEUU	1978	Messenger
Galileo	EEUU	2003	Messenger
Messenger	EEUU	2015	Messenger
Beagle 2	ESA	2003	Beagle 2
Venera 3	URRS	1966	Mars 2 lander
Venera 4	URRS	1967	Mars 2 lander
Mars 2 lander	URRS	1971	Mars 2 lander

Ventana: LAG / ORDER BY / PARTITION BY

Aterrizaje

<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
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, país, año,  
       LAG(nave) OVER (PARTITION BY país ORDER BY año) AS previo  
FROM Aterrizaje  
ORDER BY país
```

nave	país	año	previo
Viking 1	EEUU	1976	⊥
Pioneer	EEUU	1978	Viking 1
Galileo	EEUU	2003	Pioneer
Messenger	EEUU	2015	Galileo
Beagle 2	ESA	2003	⊥
Venera 3	URRS	1966	⊥
Venera 4	URRS	1967	Venera 3
Mars 2 lander	URRS	1971	Venera 4

Ventana: LEAD / ORDER BY / PARTITION BY

Aterrizaje

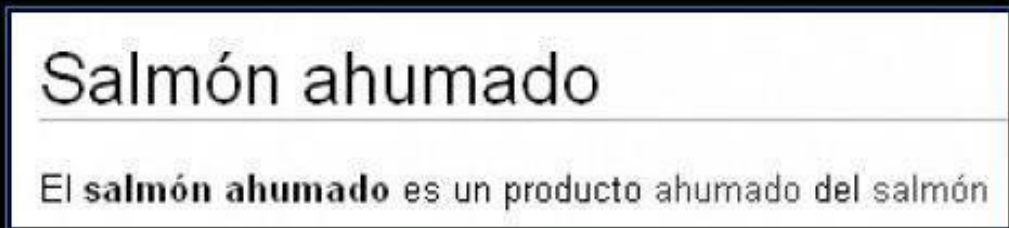
<u>nave</u>	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Venera 4	Venus	URRS	1967
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, país, año,  
       LEAD(nave) OVER (PARTITION BY país ORDER BY año) AS próximo  
FROM Aterrizaje  
ORDER BY país
```

nave	país	año	próximo
Viking 1	EEUU	1976	Pioneer
Pioneer	EEUU	1978	Galileo
Galileo	EEUU	2003	Messenger
Messenger	EEUU	2015	⊥
Beagle 2	ESA	2003	⊥
Venera 3	URRS	1966	Venera 4
Venera 4	URRS	1967	Mars 2 lander
Mars 2 lander	URRS	1971	⊥

SQL: UN LENGUAJE DECLARATIVO

SQL tiene mucha redundancia



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 debería ser computado

(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 debería ser computado

(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 JOIN
( SELECT DISTINCT P2.a_nombre AS nombre
  FROM personaje P2 JOIN
( 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
```

(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

Idealmente, el motor puede elegir el mejor plan de ejecución independientemente de la expresión particular de la consulta

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
Baker, Sala	M
Bartlett, Timothy	M
Benson, Jari	M
Benson, 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

Preguntas?

