CC3201-1

Bases de Datos

Otoño 2023

Clase 5: SQL (I)

Aidan Hogan

aidhog@gmail.com

La última vez ...

El Álgebra Relacional

El Álgebra Relacional (Mínima / Clásica)

$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R})$ $\sigma_{\text{condición}}(\mathbf{R})$ $\rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$

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

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

… ¿cómo se puede expresar este lenguaje matemático en un lenguaje computacional?

El lenguaje estructurado de consulta

STRUCTURED QUERY LANGUAGE (SQL)

Capítulo 5 Database Management Systems,
Ramakrishnan / Gehrke (Third Edition)

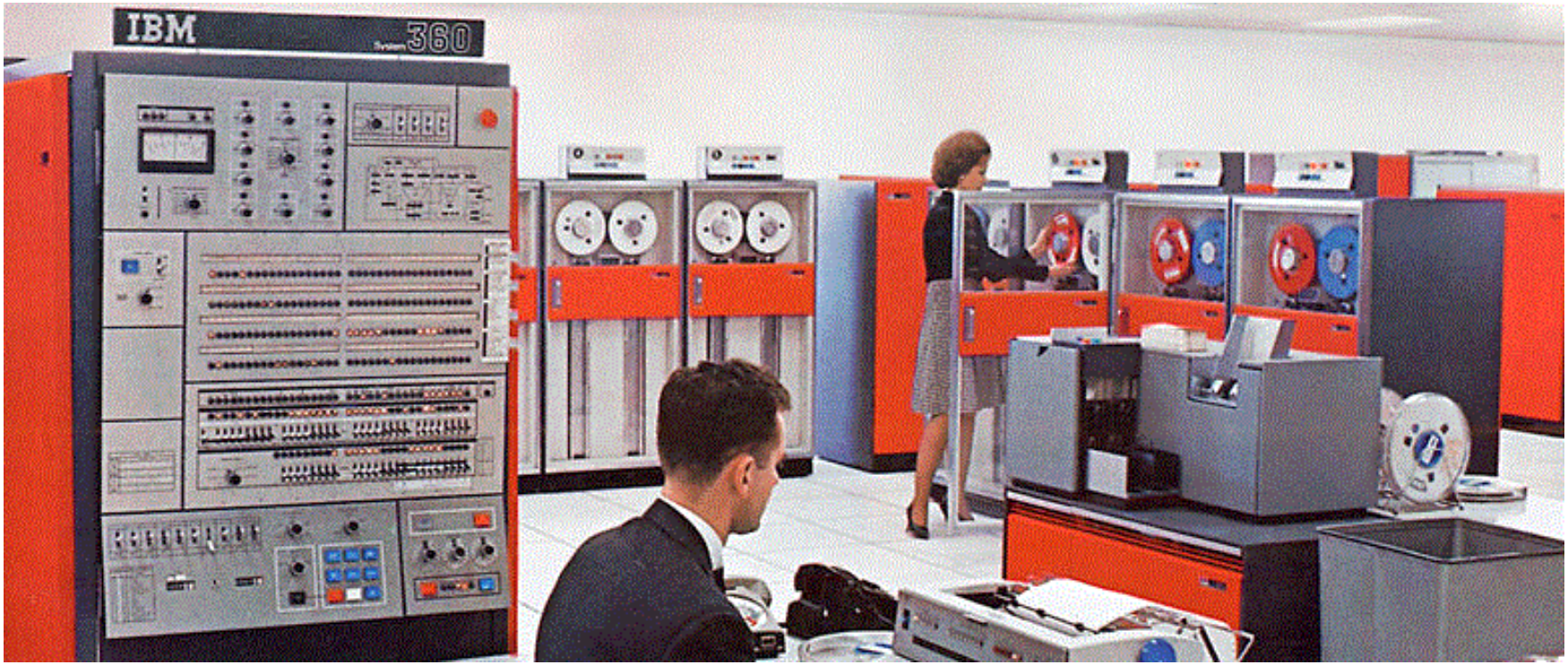
Los inicios de SQL …

Conceptualizado por

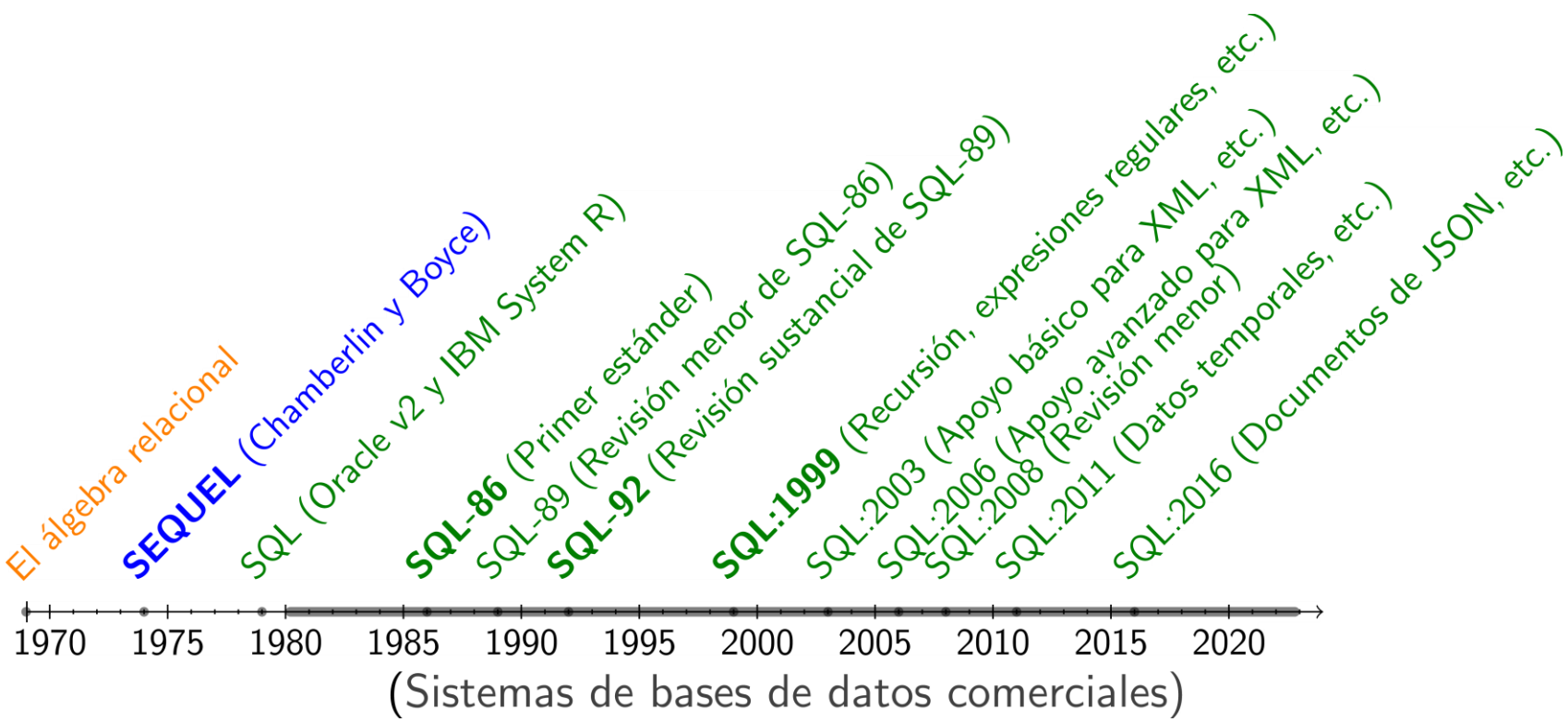
Donald Chamberlin (IBM) y Raymond F. Boyce (IBM)

en 1974

1974 ...



La evolución de SQL



Sistemas de bases de datos (con SQL)

☐ include secondary database models

166 systems in ranking, March 2023

Rank Mar 2023	Rank Feb 2023	Rank Mar 2022	DBMS	Database Model	Score Mar 2023	Score Feb 2023	Score Mar 2022
1.	1.	1.	Oracle	Relational, Multi-model	1261.29	+13.77	+9.97
2.	2.	2.	MySQL	Relational, Multi-model	1182.79	-12.66	-15.45
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	922.01	-7.08	-11.77
4.	4.	4.	PostgreSQL	Relational, Multi-model	613.83	-2.67	-3.10
5.	5.	5.	IBM Db2	Relational, Multi-model	142.92	-0.04	-19.22
6.	6.	↑ 7.	SQLite	Relational	133.82	+1.15	+1.64
7.	7.	↓ 6.	Microsoft Access	Relational	132.06	+1.03	-3.37
8.	8.	↑ 9.	Snowflake	Relational	114.40	-1.26	+28.17
9.	9.	↓ 8.	MariaDB	Relational, Multi-model	96.84	+0.03	-11.47
10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model	77.44	-1.31	-7.23
11.	11.	11.	Hive	Relational	70.91	-1.21	-10.31
12.	12.	12.	Teradata	Relational, Multi-model	63.74	+0.71	-5.11
13.	13.		Databricks	Multi-model	60.86	+0.52	
14.	↑ 15.	↑ 16.	Google BigQuery				
15.	↓ 14.	↓ 14.	FileMaker				
16.	16.	↓ 13.	SAP HANA				
17.	17.	↓ 15.	SAP Adaptive Server				
18.	18.	↓ 17.	Firebird				
19.	↑ 20.	↑ 21.	Microsoft Azure Syn				
20.	↑ 21.	↓ 19.	Informix	Relational, Multi-model	21.75	+0.05	-1.64

¡Varios sistemas pueden tener varias interpretaciones del estándar de SQL!

Pero el "core" de SQL es compatible entre los sistemas más populares.

http://db-engines.com/en/ranking/relational+dbms

Álgebra Relacional en SQL

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

Mientras tanto en Plutón …

El Álgebra Relacional

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

El Álgebra Relacional: Relación

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Relación en SQL: SELECT * FROM *R*

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 *
FROM Planeta
```

nombre	dist	radio	grav	días	años	temp	anillo
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
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
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

El Álgebra Relacional: Proyección

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Proyección en SQL: SELECT [v_1, ..., v_n]

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, dist
FROM Planeta
```

nombre	dist
Saturno	9,54
Urano	19,19
Mercurio	0,39
Venus	0,72
Tierra	1,00
Marte	1,52
Júpiter	5,20
Neptuno	30,07

Proyección en SQL: SELECT [v_1, ..., v_n]

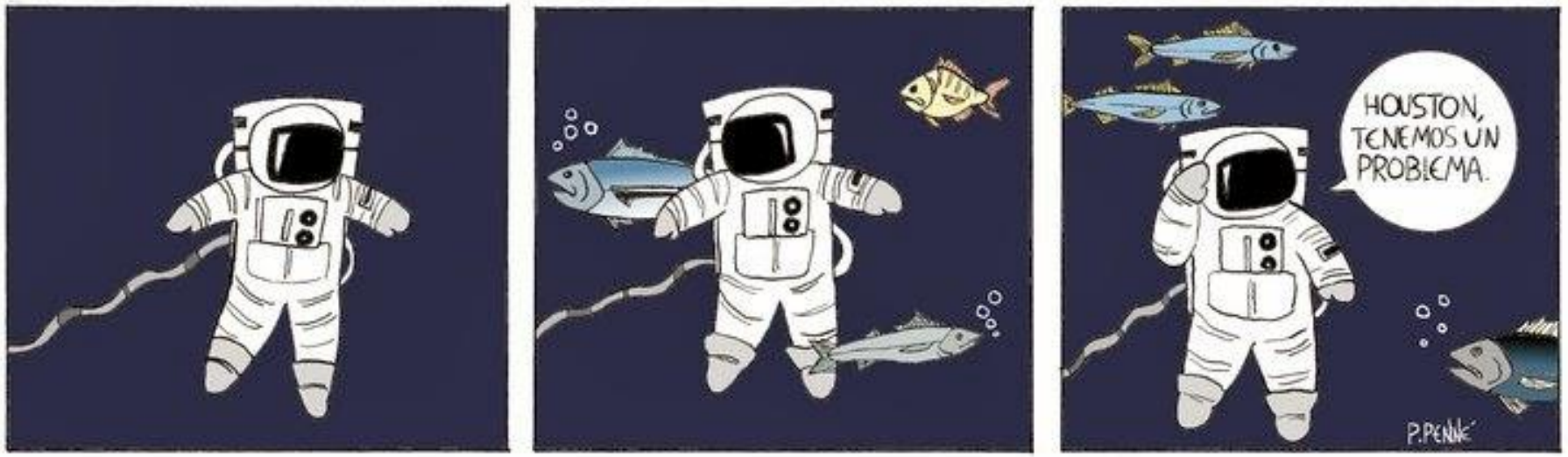
Aterrizaje

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
FROM Aterrizaje
```

planeta
Marte
Marte
Marte
Venus
Venus
Mercurio
Júpiter

¿Algún problema aquí?


HOUSTON, TENEMOS UN PROBLEMA.
P.PENNÉ

Proyección en SQL: `SELECT DISTINCT`

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 planeta
FROM Aterrizaje
```

planeta
Venus
Marte
Mercurio
Júpiter

SQL puede cambiar las reglas del álgebra relacional; por ejemplo, permite duplicados, considera orden entre las filas, etcétera.

¿Qué piensan ustedes?

¿Duplicados en tablas/resultados son útiles?

El Álgebra Relacional: Selección

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Selección en SQL: WHERE (=|<>|<|<=|etc.)

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 grav, temp
FROM Planeta
WHERE nombre = 'Venus'
```

grav	temp
8,9	730

Selección en SQL: WHERE ... AND ... (OR|NOT)

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, dist
FROM Planeta
WHERE radio > 1.0
  AND anillo IS FALSE
```

nombre	dist

¡Cuidado!

`SELECT` indica **proyección** (π)

`WHERE` indica **selección** (σ)

El Álgebra Relacional: Renombramiento

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Renombramiento en SQL: AS

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 grav AS g, nombre
FROM Planeta
WHERE grav > 9.8
```

g	nombre
22,9	Jupiter
11,0	Neptuno

El Álgebra Relacional: Unión

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \quad \mathbf{R}_1 \times \mathbf{R}_2 \quad \mathbf{R}_1 - \mathbf{R}_2$$

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

Unión en SQL: UNION

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	$\perp$	$\perp$
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 Planeta
UNION
SELECT nombre
FROM Satélite
```

nombre
Urano
Venus
Mercurio
Ío
Júpiter
...

Unión y renombrar en SQL: UNION + AS

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

```
SELECT nombre AS planeta
FROM Planeta
UNION
SELECT planeta
FROM Satélite
```

planeta
Urano
Venus
Mercurio
Tierra
Saturno
Neptuno
Júpiter
Marte

¿Cuántos resultados?

(No hay resultados duplicados)

Unión (bruta) en SQL: `UNION ALL`

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

```
SELECT nombre AS planeta
FROM Planeta
UNION ALL
SELECT planeta
FROM Satélite
```

planeta
Urano
Neptuno
Neptuno
Mercurio
Saturno
Saturno
...

Con ALL *se permiten duplicados.*

La multiplicidad será la suma; por ejemplo, Júpiter *tendrá 5 resultados, pues aparece 1 vez en* Planeta *y 4 veces en* Satélite (1 + 4 = 5).

El Álgebra Relacional: Producto Cartesiano

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \quad \mathbf{R}_1 \times \mathbf{R}_2 \quad \mathbf{R}_1 - \mathbf{R}_2$$

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

Producto Cartesiano en SQL: FROM

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, planeta, nave
FROM Satélite, Aterrizaje
```

nombre	planeta	nave
Luna	???	Messenger
...	...	...
Luna	???	Galileo
Ganímedes	???	Messenger
...	...	...
Ganímedes	???	Galileo
...	...	...

¿Algún problema aquí?

¿Cuál planeta?

Producto Cartesiano en SQL: FROM

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, Aterrizaje
```

Las tablas pueden compartir atributos, pero al momento de referenciar un atributo así, hay que desambiguarlo.

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

¿Cuántos resultados?

7 x 7 = 49

Producto Cartesiano en SQL: 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, Aterrizaje
```

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

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

Son equivalentes.

El Álgebra Relacional: Diferencia

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \quad \mathbf{R}_1 \times \mathbf{R}_2 \quad \mathbf{R}_1 - \mathbf{R}_2$$

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

Diferencia en SQL: EXCEPT

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

```
SELECT planeta
FROM Satélite
EXCEPT
SELECT nombre AS planeta
FROM Planeta
WHERE dist <= 1.00
```

Se aplica diferencia de conjuntos.

planeta
Júpiter
Saturno
Neptuno

¿Cuántos resultados?

3

Diferencia en SQL: EXCEPT ALL

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

```
SELECT planeta
FROM Satélite
EXCEPT ALL
SELECT nombre AS planeta
FROM Planeta
```

planeta
Júpiter
Júpiter
Júpiter

¿Cuántos resultados?

3

Se restan las multiplicidades.

El Álgebra Relacional: Intersección

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Intersección en SQL: INTERSECT

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 planeta
FROM Aterrizaje
INTERSECT
SELECT planeta
FROM Satélite
```

planeta
Júpiter

Se aplica intersección de conjuntos.

¿Cuántos resultados?

1

Intersección en SQL: INTERSECT ALL

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	Júpiter	ESA	2003
Galileo	Júpiter	EEUU	2003

* A modo de ejemplo

```
SELECT planeta
FROM Aterrizaje
INTERSECT ALL
SELECT planeta
FROM Satélite
```

planeta
Júpiter
Júpiter

¿Cuántos resultados?

2

Se toma la multiplicidad mínima.

El Álgebra Relacional: Join

$$\mathbf{R}$$

$$\pi_{\mathbf{A}_1,\ldots,\mathbf{A}_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{\mathbf{A}_i/\mathbf{A}_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2 \qquad \mathbf{R}_1 \times \mathbf{R}_2 \qquad \mathbf{R}_1 - \mathbf{R}_2$$

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

Joins en SQL: FROM + WHERE

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 solo '=' en el JOIN

```
SELECT nave, nombre, dist, año
FROM Planeta, Aterrizaje
WHERE 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 en SQL: 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 solo '=' 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
```

Equivalente.

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 en SQL: FROM + WHERE

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

Joins en SQL: FROM + WHERE

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.

Joins en SQL: FROM + WHERE

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
FROM Satélite S, Aterrizaje A
WHERE S.planeta = A.planeta
```

nombre
Ganímedes
Calisto
Europa
Ío

Importa la desambiguación.

Joins en SQL: 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

Join en SQL: 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 comparten (por pareja con AND).

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

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

nombre	planeta

Self Joins en SQL

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

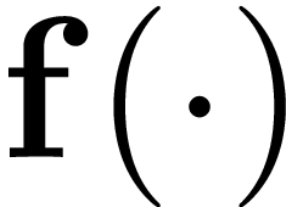
Un JOIN sobre la tabla misma

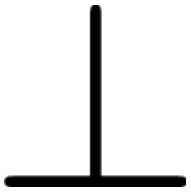
```
SELECT A1.planeta, A2.planeta
FROM Aterrizaje A1
  JOIN Aterrizaje A2
    ON A1.año = A2.año
      AND A1.planeta <> A2.planeta
```

A1.planeta	A2.planeta
Marte	Júpiter
Júpiter	Marte

Más allá del Álgebra Relacional en SQL

Más allá del Álgebra Relacional





Más allá: Condiciones avanzadas

Condiciones abreviadas en SQL: IN

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
FROM Aterrizaje
WHERE país IN ('EEUU','ESA')
```

planeta
Mercurio
Venus
Marte
Marte
Júpiter

Condiciones abreviadas en SQL: BETWEEN

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
FROM Aterrizaje
WHERE año BETWEEN 1971 AND 1978
```

planeta
Marte
Marte
Venus

Condiciones avanzadas en SQL: LIKE

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
WHERE nombre LIKE 'M%'
```

nombre
Mercurio
Marte

Condiciones avanzadas en SQL: NOT LIKE

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
WHERE nombre NOT LIKE '%no'
  AND dist > 1.00
```

nombre
Júpiter
Marte

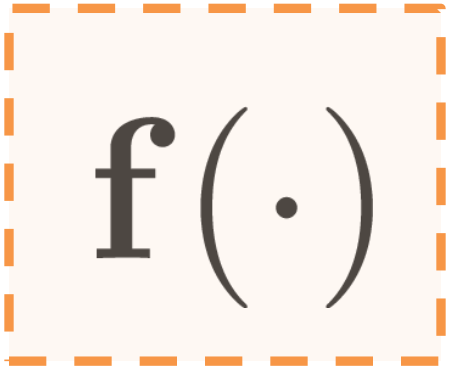
Condiciones avanzadas en SQL: LIKE

Patrón	Semántica	Ejemplo	Positivo	Negativo
%	0 o más carácteres	sat%	Saturno, SAT	asat
_	un carácter	%sat_	Satu, SATu, asatu	sat, Saturno
[charlist]	uno de los carácteres	sat[uv][r-t]%	SatUrno, SatvSno	satrno, satuurno

¡Distinción de mayúsculas depende de la configuración de un sistema en particular!

`[charlist]` *está solo en algunos motores, como SQL Server.*

Más allá: Funciones



Funciones en SQL: 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)
...
```

Funciones en SQL: 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

Funciones en SQL: 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)

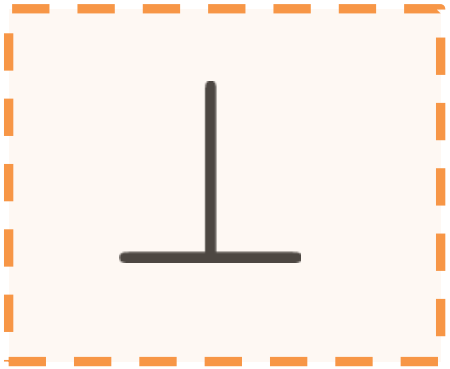
...

Funciones en SQL: Condicionales

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

Más allá: Nulos

$$\mathbf{f}(\cdot)$$



Nulos en SQL

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

¿Quién descubrió la Luna?

¿Existe forma de evitar el nulo?

¿Vale la pena evitar el nulo?

Nulos en SQL

$\perp$, $\emptyset$, $\lrcorner$, $\varnothing$, NULL

DESCONOCIDO o INAPLICABLE

(No significa FALSO)

Nulos en SQL: `IS 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 NULL
```

nombre
Luna

Nulos en SQL: `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

Nulos en SQL: Comparación con nulos

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 año > 1800
```

nombre
Tritón

Nulos en SQL: Comparación con nulos

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

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

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

nombre

Nulos en SQL: 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.

Nulos en SQL: 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 en SQL: 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

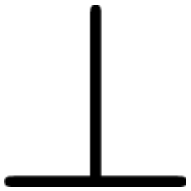
Elegir el primer valor que no sea NULL

```
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

Más allá: Orden, *rank* y limites

$\mathbf{f}(\cdot)$



Ordenar resultados: ORDER BY [DESC|ASC]

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

nave	planeta	país	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

Sistemas de bases de datos (con SQL)

☐ include secondary database models

166 systems in ranking, March 2023

Rank Mar 2023	Rank Feb 2023	Rank Mar 2022	DBMS	Database Model	Score Mar 2023	Score Feb 2023	Score Mar 2022
1.	1.	1.	Oracle	Relational, Multi-model	1261.29	+13.77	+9.97
2.	2.	2.	MySQL	Relational, Multi-model	1182.79	-12.66	-15.45
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	922.01	-7.08	-11.77
4.	4.	4.	PostgreSQL	Relational, Multi-model	613.83	-2.67	-3.10
5.	5.	5.	IBM Db2	Relational, Multi-model	142.92	-0.04	-19.22
6.	6.	↑ 7.	SQLite	Relational	133.82	+1.15	+1.64
7.	7.	↓ 6.	Microsoft Access	Relational	132.06	+1.03	-3.37
8.	8.	↑ 9.	Snowflake	Relational	114.40	-1.26	+28.17
9.	9.	↓ 8.	MariaDB	Relational, Multi-model	96.84	+0.03	-11.47
10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model	77.44	-1.31	-7.23
11.	11.	11.	Hive	Relational	70.91	-1.21	-10.31
12.	12.	12.	Teradata	Relational, Multi-model	63.74	+0.71	-5.11
13.	13.		Databricks	Multi-model	60.86	+0.52	
14.	↑ 15.	↑ 16.	Google BigQuery				
15.	↓ 14.	↓ 14.	FileMaker				
16.	16.	↓ 13.	SAP HANA				
17.	17.	↓ 15.	SAP Adaptive Server				
18.	18.	↓ 17.	Firebird				
19.	↑ 20.	↑ 21.	Microsoft Azure Syn				
20.	↑ 21.	↓ 19.	Informix	Relational, Multi-model	21.75	+0.05	-1.64

¡Varios sistemas pueden tener varias interpretaciones del estándar de SQL!

Pero el "core" de SQL es compatible entre los sistemas más populares.

http://db-engines.com/en/ranking/relational+dbms

Devolver *n* resultados: FETCH FIRST

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:2008) que se usa en Postgres y DB2.

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

nave	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

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 SQL Server y MS Access.

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

nave	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

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
OFFSET 1 LIMIT 3
```

nave	planeta	pais	año
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003
Pioneer	Venus	EEUU	1978

La Próxima Vez, Continuaremos con más del:

Structured Query Language (SQL)

Capítulo 4 & 5 | Ramakrishnan / Gehrke

Preguntas?