

#CrecimientoProfesional
#AprendeConLosPioneros

Online |  Perú



CAPACITACIÓN
PROFESIONAL

SESIÓN V

Docente: Victor Gutierrez
Data Architect

Agenda

CONSULTAS MULTITABLA

- De tipo unión.
- De tipo inner join.
- De tipo left join.
- De tipo right join.
- De tipo full outer join.
- De tipo subconsulta.

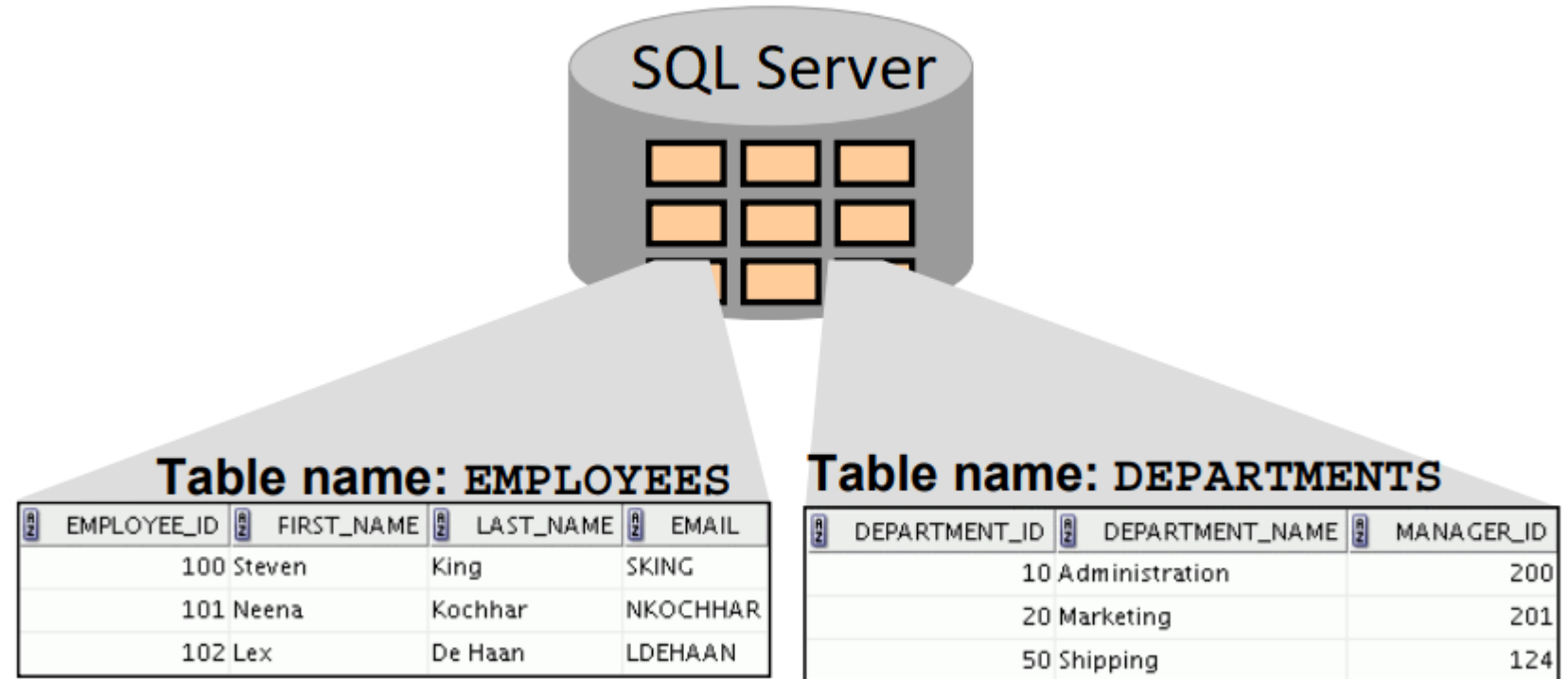
Modelo Relacional

Una base de datos relacional es una colección de relaciones o tablas BI dimensionales.

Por ejemplo, es posible que desee almacenar información sobre todos los empleados de su empresa. En una base de datos relacional, puede crear varias tablas para almacenar diferentes datos sobre sus empleados, una tabla de departamentos, una tabla de salarios, y una tabla con los empleados

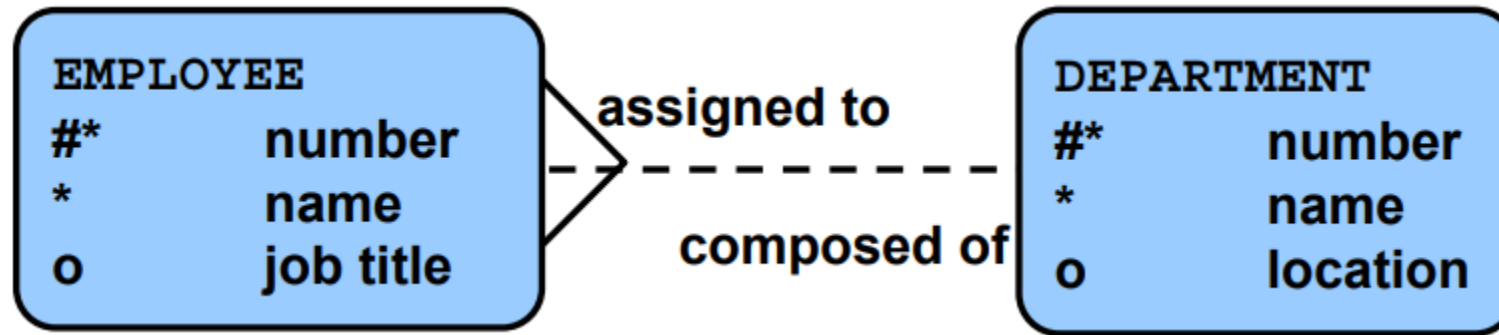
Propósito

- Comunicar
- Categorizar
- Describir
- Especificar
- Investigar
- Evolucionar
- Analizar



Modelo Entidad Relación

Las tablas (entidades) se relacionan entre si:



Escenario:

- *Asignar uno o mas empleados a un departamento*
- *Algunos departamentos no tiene asignados empleados*

Modelo Entidad Relación

Capacidades SQL

Proyección: Seleccionar las columnas. Unas o muchas

Selección: Obtener filas de una tabla que cumplen una condición.

Join: Utilizado para obtener data de diferentes tablas

Proyeccion

Tabla 1

Seleccion

Tabla 1

Tabla 1

Join




Tabla 2

Joins

Cuando se requiere obtener información de dos o mas tablas, relacionadas entre si:

- ID de los empleados que existen en la tabla de empleados
- ID de los departamentos que existen en la tabla de empleados y departamentos.
- Nombres de los departamentos que existen en la tabla departamentos

EMPLOYEES				DEPARTMENTS			
	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	
1	200	Whalen	10	1	Administration	1700	
2	201	Hartstein	20	2	Marketing	1800	
3	202	Fay	20	3	Shipping	1500	
...				4	IT	1400	
18	174	Abel	80	5	Sales	2500	
19	176	Taylor	80	6	Executive	1700	
20	178	Grant	(null)	7	Accounting	1700	
				8	Contracting	1700	



	EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	200	10	Administration
2	201	20	Marketing
3	202	20	Marketing
4	124	50	Shipping

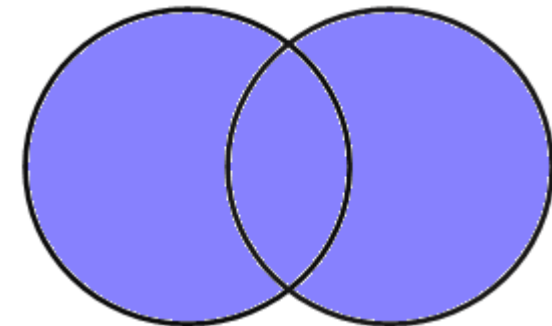
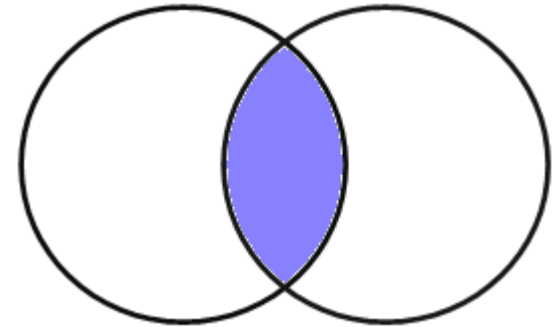
Joins

(INNER) JOIN: devuelve registros que tienen valores coincidentes en ambas tablas

FULL (OUTER) JOIN: devuelve todos los registros cuando hay una coincidencia en la tabla izquierda o derecha

```
SELECT column_name(s)
FROM table1
INNER JOIN table2 ON
table1.column_name=table2.column_name
```

```
SELECT column_name(s)
FROM table1
FULL JOIN table2 ON
table1.column_name=table2.column_name
```



Joins

(INNER) JOIN: devuelve registros que tienen valores coincidentes en ambas tablas

FULL (OUTER) JOIN: devuelve todos los registros cuando hay una coincidencia en la tabla izquierda o derecha

```

SELECT e.last_name, d.department_id, d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
  
```

	A2	LAST_NAME	A2	DEPARTMENT_ID	A2	DEPARTMENT_NAME
1		Whalen		10		Administration
2		Hartstein		20		Marketing
3		Fay		20		Marketing
4		Higgins		110		Accounting

...

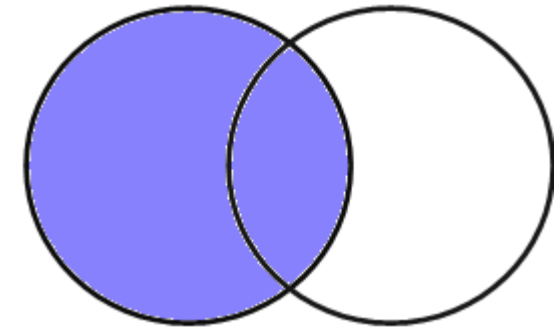
17		Zlotkey		80		Sales
18		Abel		80		Sales
19		Taylor		80		Sales
20		Grant		(null)	(null)	
21		(null)		190		Contracting

Joins

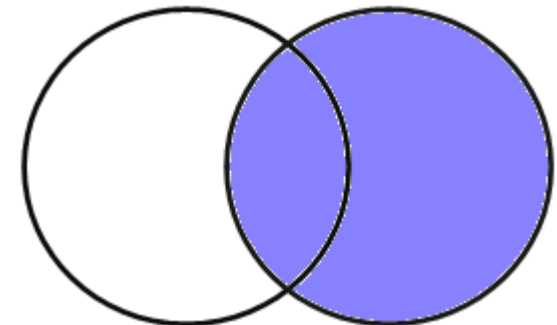
LEFT JOIN: devuelve todos los registros de la tabla izquierda y los registros coincidentes de la tabla derecha

RIGHT JOIN: devuelve todos los registros de la tabla derecha y los registros coincidentes de la tabla izquierda

```
SELECT column(s)  
FROM table1  
LEFT JOIN table2 ON table1.column=table2.column
```



```
SELECT column(s)  
FROM table1  
RIGHT JOIN table2 ON table1.column=table2.column
```



Joins

LEFT JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Vargas	50	Shipping
5	Matos	50	Shipping

...

16	Kochhar	90	Executive
17	King	90	Executive
18	Gietz	110	Accounting
19	Higgins	110	Accounting
20	Grant	(null)	(null)

RIGHT JOIN

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Hartstein	20	Marketing
3	Fay	20	Marketing
4	Davies	50	Shipping
5	Vargas	50	Shipping
6	Rajs	50	Shipping
7	Mourgos	50	Shipping
8	Matos	50	Shipping

...

18	Higgins	110	Accounting
19	Gietz	110	Accounting
20	(null)	190	Contracting

Joins

```
SELECT Customers.CompanyName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
where Orders.OrderID is null
```

CompanyName	OrderID
FISSA Fabrica Inter. Salchichas S.A.	NULL
Paris spécialités	NULL

```
SELECT Customers.CompanyName, Orders.OrderID  
FROM Customers  
RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
where orders.OrderID is null
```

CompanyName	OrderID
-------------	---------

Joins

INNER vs OUTER JOIN

- SQL 1999 el join entre dos tablas retorna solo las filas que hacen match (INNER)
- Un join entre dos tablas que retornan el con filas que no hicieron match (LEFT o RIGTH) es llamado OUTER JOIN
- Un Join entre dos tablas que retorna un INNER JOIN y también los resultados LEFT y RIGHT JOIN se llama FULL OUTER JOIN

DEPARTMENTS

	DEPARTMENT_NAME	DEPARTMENT_ID
1	Administration	10
2	Marketing	20
3	Shipping	50
4	IT	60
5	Sales	80
6	Executive	90
7	Accounting	110
8	Contracting	190

There are no employees in department 190.

Employee "Grant" has not been assigned a department ID.

Equijoin with EMPLOYEES

	DEPARTMENT_ID	LAST_NAME
1	10	Whalen
2	20	Hartstein
3	20	Fay
4	110	Higgins
5	110	Gietz
6	90	King
7	90	Kochhar
8	90	De Haan
9	60	Hunold
10	60	Ernst

...

18	80	Abel
19	80	Taylor

De Tipo Sub consulta

Quienes tienen el salario mas alto que Joel?

Query principal

Cuales son los empleados que tienen el salario mayor al de Joel?

SubQuery

Cual es el salario de Joel?

```
SELECT last_name, salary
FROM employees
WHERE salary > 11000
      (SELECT salary
       FROM employees
       WHERE last_name = 'Abel');
```

	LAST_NAME	SALARY
1	Hartstein	13000
2	Higgins	12000
3	King	24000
4	Kochhar	17000
5	De Haan	17000

Para el uso de SubQueries:

- Incluir subconsultas entre paréntesis
- Coloque las subconsultas en el lado derecho de la condición de comparación para facilitar la lectura. (Sin embargo, las subconsultas pueden aparecer a ambos lados del operador de comparación)
- Utilice operadores de una sola fila con subconsultas de una sola fila y operadores de varias filas con subconsultas de varias filas.

De Tipo Sub consulta

Un **SubQuery** es una sentencia SELECT anidada en una clausula de otra sentencia SELECT

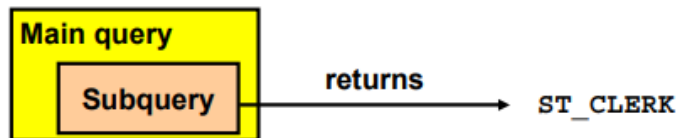
```

SELECT select_list
FROM table
WHERE expr operator ( SELECT select_list
                        FROM table )
    
```

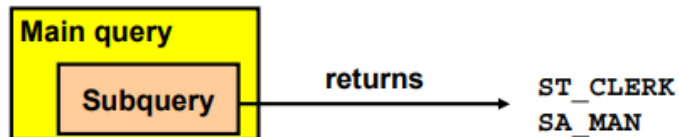
Se debería usar en las clausulas:

From
Where
Having

Single-row subquery



Multiple-row subquery



Single Row	Multiple Row
Retorna solo una fila	Retorna mas de una fila
Usa el operador de comparación de una sola fila = > < >= <= <>	Usa los operadores de comparación para múltiples filas IN, ANY, ALL

La subconsulta se ejecuta **antes** de la consulta principal. El resultado de la subconsulta es utilizado por la consulta principal

REFERENCIAS

Joins

<https://docs.microsoft.com/es-es/sql/relational-databases/performance/joins?view=sql-server-ver15>

Selects

<https://docs.microsoft.com/es-es/sql/t-sql/queries/select-transact-sql?view=sql-server-ver15>

Sub Consultas

<https://docs.microsoft.com/es-es/sql/relational-databases/performance/subqueries?view=sql-server-ver15>



CAPACITACIÓN
PROFESIONAL