



TECNÓLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2900177

Aprendiz: Brayan Santiago Guerrero Mendez

Instructor: Jhon Corredor

Neiva-Huila

Contenido

1. INNER JOIN	3
2. LEFT JOIN	4
3. RIGHT JOIN.....	5
4. FULL JOIN	6
5. CROSS JOIN	7
6. INNER JOIN A LA MISMA TABLA	11
7. INNER JOIN MULTIPLES TABLAS	13
8. SCRIPT	14

1.INNER JOIN

1.1. Realiza un query entre las tablas Empleados y Departamentos para obtener una lista de todos los empleados junto con el nombre de su departamento. Solo muestra a los empleados que están asignados a un departamento.

- El **INNER JOIN** se encarga de mostrar solo a los empleados que estén asociados a un departamento, si no hay registros asociados no se mostraran

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
---1 Ejercicio  
    SELECT e.nombre as empleados, d.nombre as departamento  
    FROM empleados e  
    INNER JOIN departamentos d  
    ON e.departamento_id = d.id;  
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

	empleados character varying (50) 🔒	departamento character varying (50) 🔒
1	Juan Manuel	Finanzas
2	María López	Finanzas
3	Ana Gómez	Marketing y Publicidad
4	Luis Rodríguez	Comercial
5	Laura Torres	Compras
6	Sofía Castro	Gestión y Administración
7	Andrés Romero	Directivo
8	Valentina Díaz	departamento tecnológico

2. LEFT JOIN

2.1. Realiza un query para obtener todos los empleados, incluso aquellos que no están asignados a un departamento, mostrando el nombre del departamento si está disponible.

- El **LEFT JOIN** muestra todos los registros de la tabla derecha en este nuestro caso la tabla **departamentos**, y los registros que estén asociados a un **empleado**, y de no estarlo completa los campos con NULL

```
////////////////////////////////////  
--2 ejercicio  
SELECT e.nombre as empleados, d.nombre as departamento  
FROM empleados e  
LEFT JOIN departamentos d  
ON e.departamento_id = d.id;  
////////////////////////////////////
```

	empleados character varying (50) 🔒	departamento character varying (50) 🔒
1	Juan Manuel	Finanzas
2	María López	Finanzas
3	Carlos Sánchez	[null]
4	Ana Gómez	Marketing y Publicidad
5	Luis Rodríguez	Comercial
6	Laura Torres	Compras
7	Diego Martínez	[null]
8	Sofía Castro	Gestión y Administración
9	Andrés Romero	Directivo
10	Valentina Díaz	departamento tecnológico

3. RIGHT JOIN

- 3.1. Realiza un query para listar todos los departamentos, incluso si no tienen empleados asignados.
- El **RIGHT JOIN** muestra todos los registros de la tabla de la izquierda en nuestro caso (**departamentos**) y valida si hay registros asociados a la tabla (**empleados**), si no hay registros asociados completa los campos con NULL

```
////////////////////////////////////  
--3 Ejercicio  
SELECT e.nombre as empleados, d.nombre as departamento  
FROM   empleados e  
RIGHT JOIN departamentos d  
ON     d.id = e.departamento_id;  
////////////////////////////////////
```

	empleados character varying (50) 🔒	departamento character varying (50) 🔒
1	Juan Manuel	Finanzas
2	María López	Finanzas
3	Ana Gómez	Marketing y Publicidad
4	Luis Rodríguez	Comercial
5	Laura Torres	Compras
6	Sofía Castro	Gestión y Administración
7	Andrés Romero	Directivo
8	Valentina Díaz	departamento tecnológico
9	[null]	ventas
10	[null]	Recursos Humanos
11	[null]	Logística

4. FULL JOIN

- 4.1. Usa un query para listar todos los empleados y departamentos, incluyendo aquellos que no tienen una relación con la otra tabla.
- El **FULL JOIN** muestra todos los registros de ambas tablas, y si algún campo de las tablas no tiene un registro asociado les coloca NULL

```
////////////////////////////////////  
-- 4 Ejercicio  
SELECT d.nombre as departamento, e.nombre as empleados  
FROM departamentos d  
FULL JOIN empleados e  
ON d.id = e.departamento_id;  
////////////////////////////////////
```

	departamento character varying (50) 🔒	empleados character varying (50) 🔒
1	Finanzas	Juan Manuel
2	Finanzas	María López
3	[null]	Carlos Sánchez
4	Marketing y Publicidad	Ana Gómez
5	Comercial	Luis Rodríguez
6	Compras	Laura Torres
7	[null]	Diego Martínez
8	Gestión y Administración	Sofía Castro
9	Directivo	Andrés Romero
10	departamento tecnológico	Valentina Díaz
11	ventas	[null]
12	Recursos Humanos	[null]
13	Logística	[null]


X



5. CROSS JOIN

5.1. Realiza un query entre las tablas de empleados y departamentos.
Explica cuál es el resultado y cómo difiere de los otros tipos de JOIN.

- El **CROSS JOIN** combina cada registro de una tabla (**empleados**) con cada registro de la tabla (**departamentos**)

```
////////////////////////////////////  
--5 Ejercicio  
SELECT e.nombre as empleados, d.nombre as departamentos  
FROM empleados e  
CROSS JOIN departamentos d  
////////////////////////////////////
```

	empleados character varying (50) 	departamentos character varying (50) 
1	Juan Manuel	Finanzas
2	María López	Finanzas
3	Carlos Sánchez	Finanzas
4	Ana Gómez	Finanzas
5	Luis Rodríguez	Finanzas
6	Laura Torres	Finanzas
7	Diego Martínez	Finanzas
8	Sofía Castro	Finanzas
9	Andrés Romero	Finanzas
10	Valentina Díaz	Finanzas
11	Juan Manuel	Recursos Humanos
12	María López	Recursos Humanos
13	Carlos Sánchez	Recursos Humanos
14	Ana Gómez	Recursos Humanos
15	Luis Rodríguez	Recursos Humanos
16	Laura Torres	Recursos Humanos
17	Diego Martínez	Recursos Humanos
18	Sofía Castro	Recursos Humanos
19	Andrés Romero	Recursos Humanos
20	Valentina Díaz	Recursos Humanos
21	Juan Manuel	Marketing y Publicidad
22	María López	Marketing y Publicidad
23	Carlos Sánchez	Marketing y Publicidad
24	Ana Gómez	Marketing y Publicidad
25	Luis Rodríguez	Marketing y Publicidad
26	Laura Torres	Marketing y Publicidad
27	Diego Martínez	Marketing y Publicidad
28	Sofía Castro	Marketing y Publicidad
29	Andrés Romero	Marketing y Publicidad
30	Valentina Díaz	Marketing y Publicidad
31	Juan Manuel	Comercial
32	María López	Comercial
33	Carlos Sánchez	Comercial

	empleados character varying (50) 	departamentos character varying (50) 
34	Ana Gómez	Comercial
35	Luis Rodríguez	Comercial
36	Laura Torres	Comercial
37	Diego Martínez	Comercial
38	Sofía Castro	Comercial
39	Andrés Romero	Comercial
40	Valentina Díaz	Comercial
41	Juan Manuel	Compras
42	María López	Compras
43	Carlos Sánchez	Compras
44	Ana Gómez	Compras
45	Luis Rodríguez	Compras
46	Laura Torres	Compras
47	Diego Martínez	Compras
48	Sofía Castro	Compras
49	Andrés Romero	Compras
50	Valentina Díaz	Compras
51	Juan Manuel	Logística
52	María López	Logística
53	Carlos Sánchez	Logística
54	Ana Gómez	Logística
55	Luis Rodríguez	Logística
56	Laura Torres	Logística
57	Diego Martínez	Logística
58	Sofía Castro	Logística
59	Andrés Romero	Logística
60	Valentina Díaz	Logística
61	Juan Manuel	Gestión y Administración
62	María López	Gestión y Administración
63	Carlos Sánchez	Gestión y Administración
64	Ana Gómez	Gestión y Administración
65	Luis Rodríguez	Gestión y Administración
66	Laura Torres	Gestión y Administración

67	Diego Martínez	Gestión y Administración
68	Sofía Castro	Gestión y Administración
69	Andrés Romero	Gestión y Administración
70	Valentina Díaz	Gestión y Administración
71	Juan Manuel	Directivo
72	María López	Directivo
73	Carlos Sánchez	Directivo
74	Ana Gómez	Directivo
75	Luis Rodríguez	Directivo
76	Laura Torres	Directivo
77	Diego Martínez	Directivo
78	Sofía Castro	Directivo
79	Andrés Romero	Directivo
80	Valentina Díaz	Directivo
81	Juan Manuel	departamento tecnológico
82	María López	departamento tecnológico
83	Carlos Sánchez	departamento tecnológico
84	Ana Gómez	departamento tecnológico
85	Luis Rodríguez	departamento tecnológico
86	Laura Torres	departamento tecnológico
87	Diego Martínez	departamento tecnológico
88	Sofía Castro	departamento tecnológico
89	Andrés Romero	departamento tecnológico
90	Valentina Díaz	departamento tecnológico
91	Juan Manuel	ventas
92	María López	ventas
93	Carlos Sánchez	ventas
94	Ana Gómez	ventas
95	Luis Rodríguez	ventas
96	Laura Torres	ventas
97	Diego Martínez	ventas
98	Sofía Castro	ventas
99	Andrés Romero	ventas
100	Valentina Díaz	ventas

6. INNER JOIN A LA MISMA TABLA

6.1. Usando un query que muestra a los empleados cuyo salario sea mayor que el de otros empleados. Este ejercicio requiere que compares la tabla consigo misma.

- Se realiza un **INNER JOIN** sobre la misma tabla para poder hacer una validación la cual permita conocer cual empleado gana más que otros.

```
SELECT e.nombre AS nombre_tabla1,  
       e.salario AS salario_tabla2,  
       em.nombre AS nombre_tabla2,  
       em.salario AS salario_tabla2  
FROM empleados e  
INNER JOIN empleados em  
ON e.salario > em.salario  
ORDER BY e.salario DESC, em.salario DESC
```

	nombre_tabla1 character varying (50) 🔒	salario_tabla2 numeric (10,2) 🔒	nombre_tabla2 character varying (50) 🔒	salario_tabla2 numeric (10,2) 🔒
1	Luis Rodríguez	10000000.00	Ana Gómez	9203000.00
2	Luis Rodríguez	10000000.00	Valentina Díaz	9000000.00
3	Luis Rodríguez	10000000.00	María López	8802000.00
4	Luis Rodríguez	10000000.00	Diego Martínez	8400000.00
5	Luis Rodríguez	10000000.00	Sofía Castro	8203200.00
6	Luis Rodríguez	10000000.00	Andrés Romero	7802400.00
7	Luis Rodríguez	10000000.00	Laura Torres	7601000.00
8	Luis Rodríguez	10000000.00	Juan Manuel	7000000.00
9	Luis Rodríguez	10000000.00	Carlos Sánchez	6000000.00
10	Ana Gómez	9203000.00	Valentina Díaz	9000000.00
11	Ana Gómez	9203000.00	María López	8802000.00
12	Ana Gómez	9203000.00	Diego Martínez	8400000.00
13	Ana Gómez	9203000.00	Sofía Castro	8203200.00
14	Ana Gómez	9203000.00	Andrés Romero	7802400.00
15	Ana Gómez	9203000.00	Laura Torres	7601000.00
16	Ana Gómez	9203000.00	Juan Manuel	7000000.00
17	Ana Gómez	9203000.00	Carlos Sánchez	6000000.00
18	Valentina Díaz	9000000.00	María López	8802000.00
19	Valentina Díaz	9000000.00	Diego Martínez	8400000.00
20	Valentina Díaz	9000000.00	Sofía Castro	8203200.00
21	Valentina Díaz	9000000.00	Andrés Romero	7802400.00
22	Valentina Díaz	9000000.00	Laura Torres	7601000.00
23	Valentina Díaz	9000000.00	Juan Manuel	7000000.00
24	Valentina Díaz	9000000.00	Carlos Sánchez	6000000.00
25	María López	8802000.00	Diego Martínez	8400000.00
26	María López	8802000.00	Sofía Castro	8203200.00
27	María López	8802000.00	Andrés Romero	7802400.00
28	María López	8802000.00	Laura Torres	7601000.00
29	María López	8802000.00	Juan Manuel	7000000.00
30	María López	8802000.00	Carlos Sánchez	6000000.00
31	Diego Martínez	8400000.00	Sofía Castro	8203200.00
32	Diego Martínez	8400000.00	Andrés Romero	7802400.00
33	Diego Martínez	8400000.00	Laura Torres	7601000.00
34	Diego Martínez	8400000.00	Juan Manuel	7000000.00
35	Diego Martínez	8400000.00	Carlos Sánchez	6000000.00
36	Sofía Castro	8203200.00	Andrés Romero	7802400.00
37	Sofía Castro	8203200.00	Laura Torres	7601000.00
38	Sofía Castro	8203200.00	Juan Manuel	7000000.00
39	Sofía Castro	8203200.00	Carlos Sánchez	6000000.00
40	Andrés Romero	7802400.00	Laura Torres	7601000.00
41	Andrés Romero	7802400.00	Juan Manuel	7000000.00
42	Andrés Romero	7802400.00	Carlos Sánchez	6000000.00
43	Laura Torres	7601000.00	Juan Manuel	7000000.00
44	Laura Torres	7601000.00	Carlos Sánchez	6000000.00
45	Juan Manuel	7000000.00	Carlos Sánchez	6000000.00

7. INNER JOIN MULTIPLES TABLAS

7.1. Realiza un query con más de dos tablas. Agrega una tercera tabla llamada Proyectos (ID, Nombre, DepartamentoID) y realiza un INNER JOIN entre Empleados, Departamentos y Proyectos para mostrar todos los empleados, sus departamentos y los proyectos en los que están involucrados.

- Se usa un INNER JOIN para unir las tablas de **empleados**, **departamentos** con la tabla de **proyectos** y poder mostrar los empleados asociados a un proyecto en cada departamento

```
////////////////////////////////////  
--7 Ejercicio  
SELECT e.nombre as empleados, d.nombre as departamentos, p.nombre as proyectos  
FROM empleados e  
INNER JOIN departamentos d  
ON e.departamento_id = d.id  
INNER JOIN proyectos p  
ON p.departamento_Id = d.id  
////////////////////////////////////
```

	empleados character varying (50) 🔒	departamentos character varying (50) 🔒	proyectos character varying (100) 🔒
1	Juan Manuel	Finanzas	Proyecto Alpha
2	María López	Finanzas	Proyecto Alpha
3	Ana Gómez	Marketing y Publicidad	Campaña de Marketing
4	Luis Rodríguez	Comercial	Desarrollo de Producto
5	Laura Torres	Compras	Optimización de Compras
6	Sofía Castro	Gestión y Administración	Mejora Administrativa
7	Andrés Romero	Directivo	Reestructuración Directiva
8	Valentina Díaz	departamento tecnológico	Innovación Tecnológica

8. SCRIPT

--CREAR TABLAS

```
CREATE TABLE departamentos (  
    id INT NOT NULL,  
    nombre VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE empleados(  
    id INT NOT NULL,  
    nombre VARCHAR(50) NOT NULL,  
    departamento_Id INT,  
    salario DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE proyectos(  
    id serial NOT NULL,  
    nombre varchar(100) NOT NULL,  
    departamento_Id integer NOT NULL,  
  
    PRIMARY KEY(id)  
);
```

-- Alterar tablas llaves foranes

```
ALTER TABLE empleados  
ADD CONSTRAINT fk_departamento
```

```
FOREIGN KEY (departamento_id)
REFERENCES departamentos(id);
```

```
ALTER TABLE proyectos
ADD CONSTRAINT fk_proyectos
FOREIGN KEY (departamento_id)
REFERENCES departamentos(id)
```

-- Insertar datos en las tablas

--Departamentos

```
INSERT INTO departamentos
      (id, nombre)
VALUES  (1,'Finanzas'),
        (2,'Recursos Humanos'),
        (3,'Marketing y Publicidad'),
        (4,'Comercial'),
        (5,'Compras'),
        (6,'Logística'),
        (7,'Gestión y Administración'),
        (8,'Directivo'),
        (9,'departamento tecnológico'),
        (10,'ventas');
```

-- Empleados

INSERT INTO empleados

(id, nombre, departamento_Id, salario)

VALUES (1, 'Juan Manuel', 1, 7000000),
(2, 'María López', 1, 8802000),
(3, 'Carlos Sánchez', NULL, 6000000),
(4, 'Ana Gómez', 3, 9203000),
(5, 'Luis Rodríguez', 4, 10000000),
(6, 'Laura Torres', 5, 7601000),
(7, 'Diego Martínez', NULL, 8400000),
(8, 'Sofía Castro', 7, 8203200),
(9, 'Andrés Romero', 8, 7802400),
(10, 'Valentina Díaz', 9, 9000000);

--Proyectos

INSERT INTO proyectos (id, nombre, departamento_Id)

VALUES (1, 'Proyecto Alpha', 1),
(2, 'Iniciativa Beta', 2),
(3, 'Campaña de Marketing', 3),
(4, 'Desarrollo de Producto', 4),
(5, 'Optimización de Compras', 5),
(6, 'Logística Eficiente', 6),
(7, 'Mejora Administrativa', 7),
(8, 'Reestructuración Directiva', 8),
(9, 'Innovación Tecnológica', 9),
(10, 'Estrategia de Ventas', 10);

-- SELECCIONAR DATOS

SELECT * FROM departamentos;

SELECT * FROM empleados;

SELECT * FROM proyectos;

-- INNSERTs

-- INNER JOIN

////////////////////////////////////

---1 Ejercicio

SELECT e.nombre as empleados, d.nombre as departamento

FROM empleados e

INNER JOIN departamentos d

ON e.departamento_id = d.id;

////////////////////////////////////

--2 ejercicio

SELECT e.nombre as empleados, d.nombre as departamento

FROM empleados e

LEFT JOIN departamentos d

ON e.departamento_id = d.id;

////////////////////////////////////

--3 Ejercicio

SELECT e.nombre as empleados, d.nombre as departamento

FROM empleados e

```
RIGHT JOIN departamentos d
ON d.id = e.departamento_id;
```

```
////////////////////////////////////
```

```
-- 4 Ejercicio
```

```
SELECT d.nombre as departamento, e.nombre as empleados
FROM departamentos d
FULL JOIN empleados e
ON d.id = e.departamento_id;
```

```
////////////////////////////////////
```

```
--5 Ejercicio
```

```
SELECT e.nombre as empleados, d.nombre as departamentos
FROM empleados e
CROSS JOIN departamentos d
```

```
////////////////////////////////////
```

```
--6 Ejercicio
```

```
SELECT e.nombre as empleados , e.salario as salario
FROM empleados e
INNER JOIN empleados es
ON e.salario>= es.salario
GROUP BY e.id
ORDER BY e.salario DESC;
```

```
////////////////////////////////////
```

```
--7 Ejercicio
```

```
SELECT e.nombre as empleados, d.nombre as departamentos, p.nombre
as proyectos
FROM empleados e
```

INNER JOIN departamentos d

ON e.departamento_id = d.id

INNER JOIN proyectos p

ON p.departamento_id = d.id