

Actividad de INNER JOIN, LEFT JOIN y RIGHT JOIN:

1. **Empleados y sus Departamentos:** Liste el emp_no, first_name, last_name y dept_no de todos los empleados.

```
SELECT
  e.emp_no,
  e.first_name,
  e.last_name,
  de.dept_no
FROM
  employees e
  INNER JOIN dept_emp de ON e.emp_no = de.emp_no
--
```

2. **Departamentos y Jefes de Departamento:** Liste el dept_name, emp_no, first_name y last_name de todos los departamentos junto con sus actuales jefes.

```
SELECT
  d.dept_name,
  e.emp_no,
  e.first_name,
  e.last_name
FROM
  departments d
  INNER JOIN dept_manager dm ON d.dept_no = dm.dept_no
  INNER JOIN employees e ON dm.emp_no = e.emp_no
```

3. **Películas no Rentadas** Identifique las películas (film_id, title) que nunca han sido rentadas.

```
SELECT
  f.film_id,
  f.title
FROM
  film f
  LEFT JOIN inventory i ON f.film_id = i.film_id
  LEFT JOIN rental r ON i.inventory_id = r.inventory_id
WHERE
  r.rental_id IS NULL
```

Tarea de INNER JOIN, LEFT JOIN y RIGHT JOIN:

1. **Empleados y sus Departamentos (con nombre):**

- Liste el emp_no, first_name, last_name y dept_name de todos los empleados.
- Ordenar resultado por emp_no DESC.

```
SELECT
  e.emp_no,
  e.first_name,
  e.last_name,
  d.dept_name
FROM
  employees e
  JOIN dept_emp de ON e.emp_no = de.emp_no
  JOIN departments d ON de.dept_no = d.dept_no
ORDER BY
  e.emp_no DESC;
```

2. **Empleados y Salarios Vigentes:**

- Muestra el first_name, last_name y el salario vigente de los empleados que tienen salarios vigentes, es decir, aquellos cuya fecha to_date es '9999-01-01'.
- Ordenar resultado por first_name ASC.

```

SELECT
  e.emp_no,
  e.first_name,
  e.last_name,
  s.salary
FROM
  employees e
  INNER JOIN salaries s ON e.emp_no = s.emp_no
  AND s.to_date = '9999-01-01'
ORDER BY
  e.first_name ASC;

```

3. Departamentos y Jefes de Departamento con Sueldos Altos:

- Liste el número de departamento (dept_no), el nombre del departamento (dept_name), el nombre y apellido de los jefes (first_name, last_name) y sus salarios (salary). Solo incluya los jefes que tienen un salario superior a \$80,000.
- Ordenar resultado por dept_no DESC.

```

SELECT
  d.dept_no,
  d.dept_name,
  e.first_name,
  e.last_name,
  s.salary
FROM
  departments d
  INNER JOIN dept_manager dm ON d.dept_no = dm.dept_no
  INNER JOIN employees e ON dm.emp_no = e.emp_no
  INNER JOIN salaries s ON e.emp_no = s.emp_no
WHERE
  s.salary > 80000
ORDER BY
  d.dept_no DESC;

```

4. Películas Sin Inventarios. (film, inventory):

- Identifique las películas (film_id, title) que no tienen inventario en ninguna tienda.
- Ordenar resultado por film_id ASC.

```

SELECT
  f.film_id,
  f.title
FROM
  film f
  LEFT JOIN inventory i ON f.film_id = i.film_id
WHERE
  i.inventory_id IS NULL
ORDER BY
  film_id ASC;

```

5. Alquileres y Detalles de Pagos. (rental, payment):

- Liste los alquileres (rental_id, rental_date) junto con los detalles de los pagos realizados (payment_id, amount, payment_date).
- Ordenar resultado por rental_id ASC.

```

SELECT
  r.rental_id,
  r.rental_date,
  p.payment_id,
  p.amount,
  p.payment_date
FROM
  rental r
  INNER JOIN payment p ON r.rental_id = p.rental_id
ORDER BY
  r.rental_id ASC;

```

6. Actores y Películas. (actor, film_actor, film):

- Muestre todos los actores (actor_id, first_name, last_name) y las películas (film_id, title) en las que han actuado.
- Ordenar resultado por actor_id ASC.

```

SELECT
  a.actor_id,
  a.first_name,
  a.last_name,
  f.film_id,
  f.title
FROM
  actor a
  INNER JOIN film_actor fa ON a.actor_id = fa.actor_id
  INNER JOIN film f ON fa.film_id = f.film_id
ORDER BY
  a.actor_id ASC;

```

7. Películas sin Actores.

- Muestre todas las películas (film_id, title) que no tienen actores asociados.
- Ordenar resultado por film_id ASC.

```

SELECT
  f.film_id,
  f.title
FROM
  film f
  LEFT JOIN film_actor fa ON f.film_id = fa.film_id
WHERE
  fa.actor_id IS NULL
ORDER BY
  f.film_id ASC;

```

Ejercicios de GROUP BY y HAVING:

1. Liste el número de identificación de los empleados junto con el salario más bajo que hayan percibido, considerando únicamente a aquellos empleados de género masculino.

```

SELECT
  e.emp_no,
  MIN(s.salary) salario_minimo
FROM
  employees e
  INNER JOIN salaries s ON e.emp_no = s.emp_no
WHERE
  e.gender = 'M'
GROUP BY
  e.emp_no;

```

2. Identifique aquellos empleados que han tenido más o igual de tres roles o títulos diferentes a lo largo de su carrera en la empresa. Muestre el número de identificación del empleado y la cantidad de roles que ha tenido.

```

SELECT
  e.emp_no,
  COUNT(DISTINCT t.title) titulos
FROM
  employees e
  INNER JOIN titles t ON e.emp_no = t.emp_no
GROUP BY
  e.emp_no
HAVING
  titulos >= 3;

```

3. Liste todos los departamentos cuyo salario promedio de sus empleados sea mayor a \$60,000. Muestre el nombre del departamento y el salario promedio correspondiente.

```

SELECT
  d.dept_name,
  AVG(s.salary) salario_promedio
FROM
  departments d
  INNER JOIN dept_emp de ON d.dept_no = de.dept_no
  INNER JOIN salaries s ON de.emp_no = s.emp_no
GROUP BY
  d.dept_name
HAVING
  salario_promedio > 60000;

```

Tarea de GROUP BY y HAVING:

1. Número de empleados por género:

Dentro de la base de datos de la compañía, es fundamental tener una visión clara de la distribución demográfica de los empleados. Calcule y muestre la cantidad total de empleados separados por su género, indicando cuántos empleados masculinos y femeninos hay en la empresa.

```
SELECT
    e.gender,
    COUNT(e.emp_no) numero_empleados
FROM
    employees e
GROUP BY
    e.gender;
```

2. Departamentos con más de 1000 empleados:

Para determinar qué departamentos requieren una gestión más detallada debido a la magnitud de empleados que contienen, liste todos los departamentos con más de 1000 empleados. Los resultados deben mostrar el nombre del departamento y la cantidad de empleados en ese departamento.

```
SELECT
    d.dept_name,
    COUNT(de.emp_no) numero_empleados
FROM
    departments d
    INNER JOIN dept_emp de ON d.dept_no = de.dept_no
GROUP BY
    d.dept_name
HAVING
    numero_empleados > 1000;
```

3. Los 5 títulos más comunes:

Cada empleado tiene un rol o título que define sus responsabilidades y posición dentro de la empresa. Para entender mejor la distribución de roles, identifique los cinco títulos de trabajo más frecuentes en la organización y muestre cuántos empleados tienen asignados cada uno de esos títulos.

```
SELECT
    t.title,
    COUNT(t.emp_no) numero_empleados
FROM
    titles t
GROUP BY
    t.title
ORDER BY
    numero_empleados DESC
LIMIT 5;
```

4. Identificación de Departamentos Sin Asignaciones Actuales de Empleados

Dentro de la estructura de la base de datos de la empresa, los empleados asignados a departamentos tienen una columna llamada `to_date` en la tabla `dept_emp`. Esta columna señala hasta cuándo está programada la asignación del empleado a ese departamento. Si un empleado está actualmente asignado a un departamento, la columna `to_date` tendrá el valor '9999-01-01', indicando que esa asignación no tiene una fecha de finalización programada y está en vigor. Basándose en esta estructura, identifique aquellos departamentos que, según los registros actuales, no tienen ningún empleado asignado a ellos. Muestre el nombre de cada uno de esos departamentos y la cantidad de empleados actuales en cada departamento, que en este caso sería cero.

```
SELECT
    d.dept_name,
    COUNT(de.emp_no) AS numero_empleados_actuales
FROM
    departments d
    LEFT JOIN dept_emp de ON d.dept_no = de.dept_no
    AND de.to_date = '9999-01-01'
GROUP BY
    d.dept_name
HAVING
    numero_empleados_actuales = 0;
```

Tarea de SUBCONSULTAS:

1. Número de empleados por género:

Dentro de la base de datos de la compañía, es fundamental tener una visión clara de la distribución demográfica de los empleados. Calcule y muestre la cantidad total de empleados separados por su género, indicando cuántos empleados masculinos y femeninos hay en la empresa.

```

SELECT
    catalogo_generos.gender,
    (
        SELECT
            COUNT(e.emp_no)
        FROM
            employees e
        WHERE
            e.gender = catalogo_generos.gender
    ) numero_empleados
FROM
    (
        SELECT
            e.gender
        FROM
            employees e
        GROUP BY
            e.gender
    ) catalogo_generos;

```

2. Departamentos con más de 1000 empleados:

Para determinar qué departamentos requieren una gestión más detallada debido a la magnitud de empleados que contienen, liste todos los departamentos con más de 1000 empleados. Los resultados deben mostrar el nombre del departamento y la cantidad de empleados en ese departamento.

```

SELECT
    d.dept_name,
    (
        SELECT
            COUNT(de.emp_no)
        FROM
            dept_emp de
        WHERE
            de.dept_no = d.dept_no
    ) numero_empleados
FROM
    departments d
HAVING
    numero_empleados > 1000;

```

3. Los 5 títulos más comunes:

Cada empleado tiene un rol o título que define sus responsabilidades y posición dentro de la empresa. Para entender mejor la distribución de roles, identifique los cinco títulos de trabajo más frecuentes en la organización y muestre cuántos empleados tienen asignados cada uno de esos títulos.

```

SELECT
    catalogo_titulos.title,
    (
        SELECT
            COUNT(t.emp_no)
        FROM
            titles t
        WHERE
            t.title = catalogo_titulos.title
    ) numero_empleados
FROM
    (
        SELECT
            DISTINCT t.title
        FROM
            titles t
    ) catalogo_titulos
ORDER BY
    numero_empleados DESC
LIMIT 5;

```

4. Identificación de Departamentos Sin Asignaciones Actuales de Empleados

Dentro de la estructura de la base de datos de la empresa, los empleados asignados a departamentos tienen una columna llamada `to_date` en la tabla `dept_emp`. Esta columna señala hasta cuándo está programada la asignación del empleado a ese departamento. Si un empleado está actualmente asignado a un departamento, la columna `to_date` tendrá el valor '9999-01-01', indicando que esa asignación no tiene una fecha de finalización programada y está en vigor. Basándose en esta estructura, identifique aquellos departamentos que, según los registros actuales, no tienen ningún empleado asignado a ellos. Muestre el nombre de cada uno de esos departamentos y la cantidad de empleados actuales en cada departamento, que en este caso sería cero.

```
SELECT
    *
FROM (
    SELECT
        d.dept_name,
        COUNT(de.emp_no) AS numero_empleados_actuales
    FROM
        departments d
        LEFT JOIN dept_emp de ON d.dept_no = de.dept_no
        AND de.to_date = '9999-01-01'
    GROUP BY
        d.dept_name
    HAVING
        numero_empleados_actuales = 0
) resultado;
```