

Comience desde la **base de datos 'publications'** que creó en el create\_database ejercicio para responder las siguientes preguntas:

### Pregunta 1

Su gerente le solicita que entregue lo siguiente:

- Una tabla con el nombre y apellido del autor, dirección y número de teléfono.

```
SELECT  
  
    au_fname AS nombre,  
  
    au_lname AS apellido,  
  
    address AS direccion,  
  
    phone AS telefono  
  
FROM authors;
```

- Una tabla con el nombre y la dirección de las tiendas.

```
SELECT  
  
    stor_name AS tienda,  
  
    CONCAT_WS(' ', stor_address, city, state, zip) AS direccion_completa  
  
FROM stores;
```

- ¿En qué estados tenemos tiendas?

```
SELECT DISTINCT state  
  
FROM stores  
  
WHERE state IS NOT NULL  
  
ORDER BY state;
```

- ¿Cuántas tiendas tenemos por estado?

```
SELECT  
  
    state,  
  
    COUNT(*) AS num_tiendas  
  
FROM stores  
  
GROUP BY state  
  
ORDER BY num_tiendas DESC, state;
```

### Pregunta 2

Se le pide que entregue lo siguiente trabajando sobre la tabla jobs.

- ¿Qué descripción de trabajo tiene un salario más alto (max\_lvl)?

```
SELECT job_desc, max_lvl
FROM jobs
ORDER BY max_lvl DESC, min_lvl DESC
LIMIT 1;
```

Ordene la tabla del salario más alto al más bajo, si el salario máximo es igual entre dos trabajos, utilice el salario mínimo para ordenar.

```
SELECT job_desc, min_lvl, max_lvl
FROM jobs
ORDER BY max_lvl DESC, min_lvl DESC, job_desc;
```

### Pregunta 3

- Modifique el formato de la ord\_date columna de la sales tabla para mostrar el día, el mes y el año. Actualmente, la fecha y hora del pedido siempre indican las 00:00, lo que no proporciona información útil adicional.

```
SELECT
    ord_num,
    DATE_FORMAT(ord_date, '%d-%m-%Y') AS ord_date_dmy,
    stor_id,
    title_id,
    qty,
    payterms
FROM sales;
```

- ¿Cuántos pedidos se realizaron al año?

```
SELECT
    YEAR(ord_date) AS anio,
    COUNT(DISTINCT ord_num) AS pedidos_distintos
FROM sales
GROUP BY YEAR(ord_date)
ORDER BY anio;
```

- ¿Cuál es la cantidad de pedidos realizados al año?

```
SELECT
    YEAR(ord_date) AS anio,
```

```
SUM(qty) AS unidades_vendidas
```

```
FROM sales
```

```
GROUP BY YEAR(ord_date)
```

```
ORDER BY anio;
```

- ¿Cuál es la única tienda que ofrece descuentos?

```
SELECT DISTINCT
```

```
s.stor_id,
```

```
s.stor_name,
```

```
s.state
```

```
FROM stores s
```

```
JOIN discounts d
```

```
ON d.stor_id = s.stor_id
```

```
ORDER BY s.stor_id;
```

- En la tabla de ventas, por ejemplo, se vendieron 5 artículos de BU1032 con un precio de 19,99 \$ cada uno en la tienda 6380 y también se vendieron 10 unidades del mismo artículo en la tienda 8042. La pregunta es: ¿cuál title generó más ingresos? Por favor, especifique también el nombre y el estado de la tienda.

```
SELECT
```

```
t.title,
```

```
s.stor_name,
```

```
s.state,
```

```
SUM(sa.qty * t.price * (1 - COALESCE(
```

```
    CASE
```

```
        WHEN d.stor_id IS NOT NULL
```

```
            AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)
```

```
            AND (d.highqty IS NULL OR sa.qty <= d.highqty)
```

```
        THEN d.discount
```

```
        ELSE 0
```

```
    END, 0))) AS ingresos
```

```
FROM sales sa
```

```
JOIN titles t ON t.title_id = sa.title_id
```

```
JOIN stores s ON s.stor_id = sa.stor_id
```

```

LEFT JOIN discounts d
    ON d.stor_id = s.stor_id
    AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)
    AND (d.highqty IS NULL OR sa.qty <= d.highqty)
GROUP BY t.title, s.stor_name, s.state
ORDER BY ingresos DESC
LIMIT 1;

```

- ¿Cuánto dinero tiene la tienda 8042 de descuento en total? Pista: Busca el descuento de la tienda y multiplícalo por la cantidad de artículos de esa tienda y por sus precios.

```

SELECT
    s.stor_id,
    s.stor_name,
    SUM(sa.qty * t.price * COALESCE(
        CASE
            WHEN d.stor_id IS NOT NULL
                AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)
                AND (d.highqty IS NULL OR sa.qty <= d.highqty)
            THEN d.discount
            ELSE 0
        END, 0)) AS descuento_total
FROM sales sa
JOIN titles t ON t.title_id = sa.title_id
JOIN stores s ON s.stor_id = sa.stor_id
LEFT JOIN discounts d
    ON d.stor_id = s.stor_id
    AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)
    AND (d.highqty IS NULL OR sa.qty <= d.highqty)
WHERE s.stor_id = '8042'
GROUP BY s.stor_id, s.stor_name;

```

- ¿Qué tipo de título es más caro en promedio? Consejo: Calcula el precio promedio por tipo de título y ordénalo de mayor a menor.

```

SELECT

s.stor_id,

s.stor_name,

SUM(sa.qty * t.price * COALESCE(

    CASE

        WHEN d.stor_id IS NOT NULL

            AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)

            AND (d.highqty IS NULL OR sa.qty <= d.highqty)

        THEN d.discount

        ELSE 0

    END, 0)) AS descuento_total

FROM sales sa

JOIN titles t ON t.title_id = sa.title_id

JOIN stores s ON s.stor_id = sa.stor_id

LEFT JOIN discounts d

    ON d.stor_id = s.stor_id

    AND (d.lowqty IS NULL OR sa.qty >= d.lowqty)

    AND (d.highqty IS NULL OR sa.qty <= d.highqty)

WHERE s.stor_id = '8042'

GROUP BY s.stor_id, s.stor_name;

```

#### **Pregunta 4**

¿Cuál es la relación entre... (uno a uno, uno a muchos, muchos a muchos)?

- employee y job?

```

SELECT

j.job_id,

j.job_desc,

COUNT(e.emp_id) AS empleados_en_este_job

FROM jobs j

LEFT JOIN employee e

    ON e.job_id = j.job_id

GROUP BY j.job_id, j.job_desc

```

```
ORDER BY empleados_en_este_job DESC, j.job_id;
```

- title y author?

```
-- Autores por título
```

```
SELECT
```

```
    t.title_id,
```

```
    t.title,
```

```
    COUNT(ta.au_id) AS num_autores
```

```
FROM titles t
```

```
LEFT JOIN titleauthor ta
```

```
    ON ta.title_id = t.title_id
```

```
GROUP BY t.title_id, t.title
```

```
ORDER BY num_autores DESC, t.title;
```

```
-- Títulos por autor
```

```
SELECT
```

```
    a.au_id,
```

```
    CONCAT(a.au_fname, ' ', a.au_lname) AS autor,
```

```
    COUNT(ta.title_id) AS num_titulos
```

```
FROM authors a
```

```
LEFT JOIN titleauthor ta
```

```
    ON ta.au_id = a.au_id
```

```
GROUP BY a.au_id, autor
```

```
ORDER BY num_titulos DESC, autor;
```