# Introducción a
# Bases de Datos y SQL

Módulo 5 - Resolución de los desafíos

# Ejercicio 1

1. Usando la tabla **ÚLTIMOS LANZAMIENTOS**, obtener una lista de todos aquellos temas lanzados durante el último año (año más alto que figure en la tabla). En el resultado de la consulta, mostrar solo las columnas **ARTISTA** y **TÍTULO**. Ordenar el resultado alfabéticamente por los nombres de los artistas; en el caso de que un mismo artista haya tenido más de un lanzamiento, organizar el resultado por los títulos de dichos lanzamientos.

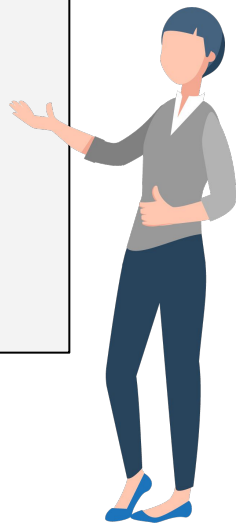
```
SELECT ARTISTA, TITULO FROM ULTIMOS_LANZAMIENTOS
WHERE ANO = (SELECT MAX(ANO) FROM ULTIMOS_LANZAMIENTOS)
ORDER BY ARTISTA, TITULO;
```

2. Utilizando la tabla **TOP SPOTIFY**, obtener una lista de todos aquellos lanzamientos correspondientes a **LADY GAGA** con mayor permanencia en la plataforma digital. En el resultado de la consulta, sólo mostrar los títulos de las canciones de la artista. Ordenar el resultado alfabéticamente por los títulos de las canciones.

```
SELECT TITULO FROM TOP_SPOTIFY
WHERE PERMANENCIA =
(SELECT MAX(PERMANENCIA) FROM TOP_SPOTIFY)
AND ARTISTA = 'LADY GAGA'
ORDER BY TITULO;
```

3. Utilizando la tabla **TOP SPOTIFY**, generar una consulta que muestre los campos **ARTISTA** y **TÍTULO**. Agregar una columna con el nombre **TIPO** en la que se muestren los valores definidos en el ejercicio. Ordenar el resultado alfabéticamente por nombres de los artistas. En el caso de que haya un artista con más de una canción en el listado, mostrar ordenados alfabéticamente los nombres de las canciones.

```
SELECT TITULO, ARTISTA,
CASE
WHEN GENERO LIKE '%POP%' THEN 'POP'
WHEN GENERO LIKE '%ELECTRO%' OR GENERO LIKE '%HOUSE%' THEN 'ELECTRÓNICA'
ELSE 'OTRO'
END AS TIPO
FROM TOP_SPOTIFY
ORDER BY ARTISTA, TITULO;
```



# Ejercicio 2

1. Obtener una lista de todos aquellas canciones que contengan en su título la palabra **BREAK**. La lista resultante debe mostrar las canciones de las tablas **TOP SPOTIFY** y **ÚLTIMOS LANZAMIENTOS**. En el resultado de la consulta, mostrar todos los campos de ambas tablas. Ordenar el resultado alfabéticamente por los nombres de las canciones. Por último, agregar a la consulta una columna con el nombre **ESTADO** en la que figure la palabra **ANTERIOR** para todos aquellos registros que provienen de la tabla **TOP SPOTIFY** y la palabra **ÚLTIMO** para todos aquellos que provienen de la tabla **ÚLTIMOS LANZAMIENTOS**.

```
SELECT *, 'ANTERIOR' AS ESTADO FROM TOP_SPOTIFY
WHERE TITULO LIKE '%BREAK%'
UNION
SELECT *, 'ULTIMO' AS ESTADO FROM ULTIMOS_LANZAMIENTOS
WHERE TITULO LIKE '%BREAK%'
ORDER BY TITULO;
```

# Ejercicio 3

1. Utilizando la tabla **LIBROS**, obtener una lista de todos aquellos productos cuyo precio supere al precio promedio de todos los libros. La lista debe contener todos los campos de la tabla. Ordenar el resultado alfabéticamente por los títulos de los libros.

```
SELECT * FROM LIBROS
WHERE PRECIO >
(SELECT AVG(PRECIO) FROM LIBROS)
ORDER BY TITULO;
```

2. Dada la tabla **LIBROS**, extraer una lista de todos aquellos libros pertenecientes a la categoría **NOVELAS** cuyo precio sea superior al libro más caro de la categoría **ENSAYOS**. La lista debe contener todos los campos de la tabla. Mostrar el resultado de la consulta ordenado de mayor a menor por los precios de los libros obtenidos.

```
SELECT * FROM LIBROS
WHERE CATEGORIA = 'NOVELAS' AND
PRECIO >
(SELECT MAX(PRECIO) FROM LIBROS
WHERE CATEGORIA = 'ENSAYOS')
ORDER BY PRECIO DESC;
```

3. Utilizando la tabla **EMPLEADOS**, obtener una lista de todos aquellos empleados con mayor permanencia dentro de la empresa.

```
SELECT * FROM EMPLEADOS
WHERE PERMANENCIA =
(SELECT MAX(PERMANENCIA)
FROM EMPLEADOS);
```

4. A partir de la tabla **EMPLEADOS**, extraer una lista de todos aquellos empleados con el puesto más alto.

```
SELECT * FROM EMPLEADOS
WHERE PUESTO_ID =
(SELECT MAX(PUESTO_ID)
FROM EMPLEADOS);
```

5. Utilizando la tabla **LIBROS**, generar una consulta que muestre los campos **LIBRO_ID**, **TÍTULO**, **CATEGORÍA** y **PRECIO_PÚBLICO**. Agregar una columna con el nombre **TIPO** en la que se muestren los valores solicitados en el ejercicio. Ordena el resultado alfabéticamente por el título de los libros. No muestres en el resultado de la consulta aquellos libros que no tienen precio (falta de stock).

```
SELECT LIBRO_ID, TITULO, CATEGORIA, PRECIO_PUBLICO,
CASE
WHEN PRECIO_PUBLICO < 15 THEN 'ECONOMICO'
WHEN PRECIO_PUBLICO <=30 THEN 'REGULAR'
ELSE 'DELUXE'
END AS TIPO
FROM LIBROS
WHERE PRECIO_PUBLICO IS NOT NULL
ORDER BY TITULO;
```

6. Obtener una lista de todos aquellos empleados que ocupen o hayan ocupado el puesto **9**. La lista debe mostrar los empleados que actualmente estén trabajando en la empresa y los que se hayan desvinculado; por lo tanto, la consulta se debe llevar a cabo en las tablas **EMPLEADOS** y **EMPLEADOS ANTERIORES**. En el resultado de la consulta, sólo debes mostrar una columna (con el nombre **EMPLEADO**) en la que figuren el nombre y el apellido de los empleados, separando ambos datos por una coma y un espacio (por ejemplo, **GARCIA, MONICA**).
Por último, ordenar el resultado alfabéticamente.

```
SELECT CONCAT(APELLIDO, ', ', NOMBRE) EMPLEADO FROM EMPLEADOS
WHERE PUESTO_ID = 9
UNION
SELECT CONCAT(APELLIDO, ', ', NOMBRE) FROM EMPLEADOS_ANTERIORES
WHERE PUESTO_ID = 9
ORDER BY EMPLEADO;
```

7. Generar una consulta que muestre las siguientes columnas: **LIBRO_ID**, **TÍTULO**, **PRECIO_PÚBLICO**, **LOCAL_ID**, **FACTURA_NRO** y **CANTIDAD**. Estos campos se encuentran en las tablas **LIBROS** y **VENTAS**. Ordenar el resultado alfabéticamente por los títulos de los libros.

```
SELECT V.LIBRO_ID, TITULO, PRECIO_PUBLICO, LOCAL_ID, FACTURA_NRO, CANTIDAD
FROM LIBROS L, VENTAS V
WHERE L.LIBRO_ID = V.LIBRO_ID
ORDER BY L.TITULO;
```

8. Modificar la consulta anterior para agregar una columna con el nombre **TOTAL** en la que se multipliquen los valores de las columnas **PRECIO_PÚBLICO** y **CANTIDAD**. Esta nueva columna debe mostrar como máximo sólo 2 decimales.

```
SELECT V.LIBRO_ID, TITULO, PRECIO_PUBLICO, LOCAL_ID, FACTURA_NRO, CANTIDAD,
ROUND(PRECIO_PUBLICO * CANTIDAD, 2) AS TOTAL
FROM LIBROS L, VENTAS V
WHERE L.LIBRO_ID = V.LIBRO_ID
ORDER BY L.TITULO;
```

9. Crear una consulta en la que se muestren sólo los títulos de aquellos libros que nunca fueron vendidos. Ordenar el resultado alfabéticamente.

```
SELECT L.TITULO FROM LIBROS L LEFT JOIN VENTAS V
ON L.LIBRO_ID = V.LIBRO_ID
WHERE FACTURA_NRO IS NULL
ORDER BY TITULO;
```

# ¡Terminaste el módulo!

## Estás listo para rendir el examen