

**Evidencia 7.1**

**Base de datos *Sakila***

******

* **Tema:** Distintas formas de JOINS entre tablas en MySQL
* **Entregables:**

1. Modelo relacional de la base de datos
2. Diccionario de datos con campos documentados (Generado)
3. Archivo **.sql** (DDL)
4. Tabla consultas: Capturas de pantalla de resultados de cada consulta

* **Actividad:**



|  |
| --- |
| Esta base de datos supone una tienda de alquiler de películas que puede tener sucursales y vendedores.  Es una base de datos de código abierto creada originalmente por MySQL y distribuida bajo la licencia BSD. Su esquema permite modelar conceptos como películas, actores, relaciones entre películas y actores, y una tabla de inventario central que conecta películas, tiendas y alquileres. |

1. **Familiarizarse con la base de datos**

Entregar***capturas de pantalla*** de cada ítem y el texto de la consulta.

### *Imprima la lista de las tablas.existentes en la base de datos*

### *Describa la estructura de TODAS las tablas.*

### *Imprima los datos almacenados en TODAS las tablas.*

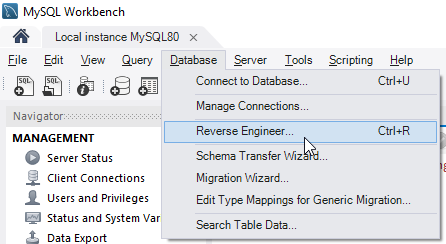
### *Genere, analice y explique el modelo de datos*

### *Genere su diccionario de datos documentado.*

### *Tenga en cuenta:*

### *Recuerde: Ingeniería inversa (Organizar Modelo).*

*“La ingeniería inversa o retroingeniería es el proceso llevado a cabo con el objetivo de obtener información o un diseño a partir de un producto, con el fin de determinar cuáles son sus componentes y de qué manera interactúan entre sí y cuál fue el proceso de fabricación.”*



1. **Consultas a la base de datos:**

Luego de realizar la consulta de la ***evidencia No. 6.1 (***[***Evidencia 6.1 - Conferencias JOINS\_Subconsultas***](https://docs.google.com/document/d/1LaaMoU8hD2_zOqRKV55f0O_TmfNgjqMKUchjVVrvhgg/edit?usp=sharing)***)***, cada aprendiz desarrollará dos consultas que implementen JOINS.

|  |  |  |
| --- | --- | --- |
| **Enunciado** | | **Query** |
| **MySQL INNER JOIN, Self Join y Operador UNION** | | |
| **1** | Clientes y sus direcciones | *Texto de la consulta:*  SELECT c.first\_name, c.last\_name, a.address FROM customer c INNER JOIN address a ON c.address\_id = a.address\_id;  *Captura de pantalla:* |
| **2** | Comparar ciudades dentro del mismo país | *Texto de la consulta:*  SELECT c1.city AS ciudad1, c2.city AS ciudad2, co.country FROM city c1 JOIN city c2 ON c1.country\_id = c2.country\_id AND c1.city\_id < c2.city\_id JOIN country co ON c1.country\_id = co.country\_id;  *Captura de pantalla:* |
| **3** | Nombres de actores y staff (solo nombres) | *Texto de la consulta:*  SELECT first\_name, last\_name FROM actor UNION SELECT first\_name, last\_name FROM staff;  *Captura de pantalla:* |
| **MySQL LEFT JOIN** | | |
| **4** | Todos los clientes y su información de alquiler (incluso si no han alquilado) | *Texto de la consulta:*  SELECT c.first\_name, r.rental\_date FROM customer c LEFT JOIN rental r ON c.customer\_id = r.customer\_id;  *Captura de pantalla:* |
| **5** | Películas y su categoría (aunque no tengan categoría asignada) | *Texto de la consulta:*  SELECT f.title, c.name AS categoria FROM film f LEFT JOIN film\_category fc ON f.film\_id = fc.film\_id LEFT JOIN category c ON fc.category\_id = c.category\_id;  *Captura de pantalla:* |
| **MySQL RIGHT JOIN** | | |
| **6** | Películas y su categoría (enfocado en que salgan todas las categorías) | *Texto de la consulta:*  SELECT f.title, c.name AS categoria FROM film f RIGHT JOIN film\_category fc ON f.film\_id = fc.film\_id RIGHT JOIN category c ON fc.category\_id = c.category\_id;  *Captura de pantalla:* |
| **7** | Clientes y direcciones (todas las direcciones aunque no tengan clientes) | *Texto de la consulta:*  SELECT c.first\_name, a.address FROM customer c RIGHT JOIN address a ON c.address\_id = a.address\_id;  *Captura de pantalla:* |
| **MySQL CROSS JOIN** | | |
| **8** | Dos idiomas con dos categorías (combinaciones posibles) | *Texto de la consulta:*  SELECT l.name AS idioma, c.name AS categoria FROM language l CROSS JOIN category c LIMIT 10;  *Captura de pantalla:* |
| **9** | Dos actores y dos películas (todas las combinaciones posibles) | *Texto de la consulta:*  SELECT a.first\_name, f.title FROM actor a CROSS JOIN film f LIMIT 10;  *Captura de pantalla:* |
| **Subconsultas:**  **a. En cláusula SELECT** | | |
| **10** | Número total de actores en cada fila | *Texto de la consulta:*  SELECT first\_name, last\_name, (SELECT COUNT(\*) FROM actor) AS total\_actores FROM actor LIMIT 5;  *Captura de pantalla:* |
| **11** | Total de películas por cada actor (sencilla) | *Texto de la consulta:*  SELECT a.first\_name, a.last\_name, (SELECT COUNT(\*) FROM film\_actor fa WHERE fa.actor\_id = a.actor\_id) AS total\_peliculas FROM actor a LIMIT 5;  *Captura de pantalla:* |
| **Subconsultas:**  **b. En cláusula FROM** | | |
| **12** | Subconsulta como tabla derivada de películas recientes | *Texto de la consulta:*  SELECT \*FROM ( SELECT title, release\_year FROM film WHERE release\_year >= 2005 ) AS peliculas\_recientes;  *Captura de pantalla:* |
| **13** | Promedio de duración por subconsulta | *Texto de la consulta:*  SELECT avg\_length.promedio FROM (SELECT AVG(length) AS promedio FROM film) AS avg\_length;  *Captura de pantalla:* |
| **Subconsultas:**  **c. En cláusula WHERE** | | |
| **14** | Clientes que hayan alquilado alguna película | *Texto de la consulta:*  SELECT first\_name, last\_name FROM customer WHERE customer\_id IN (SELECT customer\_id FROM rental);  *Captura de pantalla:* |
| **15** | Películas con duración mayor al promedio | *Texto de la consulta:*  SELECT title, length FROM film WHERE length > (SELECT AVG(length) FROM film);  *Captura de pantalla:* |

1. **Actividad en clase: Consultas a la base de datos *(Miércoles, Junio 18 de 2025)***

|  |  |  |
| --- | --- | --- |
|  | **Enunciado** | **Query** |
| **1** | Imprima las tablas existentes en la base de datos | *Texto de la consulta:*  Show tables  *Captura de pantalla:* |
| **2** | ¿Cuántas ciudades hay registradas en la base de datos? | *Texto de la consulta:*  SELECT COUNT(\*) AS total\_ciudades FROM city;  *Captura de pantalla:* |
| **3** | ¿Cuántos países se encuentran registrados? | *Texto de la consulta:*  SELECT COUNT(\*) AS total\_paises FROM country;  *Captura de pantalla:* |
| **4** | Imprima la lista de los clientes de la empresa (nombres y datos de contacto). | *Texto de la consulta:*  SELECT c.first\_name, c.last\_name, c.email, a.address, a.phone FROM customer c JOIN address a ON c.address\_id = a.address\_id;  *Captura de pantalla:* |
| **5** | ¿Cuál es el total acumulado de todos los pagos realizados en la empresa?    Concat - format | *Texto de la consulta:*  SELECT SUM(amount) AS total\_pagado FROM payment;  *Captura de pantalla:* |
| **6** | ¿Cuál es el importe de pago más alto? | *Texto de la consulta:*  SELECT MAX(amount) AS pago\_maximo FROM payment;  *Captura de pantalla:* |
| **7** | ¿Cuál es el promedio de los importes de pago? | *Texto de la consulta:*  SELECT ROUND(AVG(amount), 2) AS promedio\_pago FROM payment;  *Captura de pantalla:* |
| **8** | ¿Cuáles son las películas que alquilan entre 3 y 5 días?  \*Between    etc | *Texto de la consulta:*  SELECT title, rental\_duration FROM film WHERE rental\_duration BETWEEN 3 AND 5;  *Captura de pantalla:* |
| **9** | 1. ¿Cuántas películas clasificadas solo para adultos tienen para alquilar?          SELECT rating as clasificación         FROM film;    Imprima la lista de películas clasificadas solo para adultos que están disponibles para alquilar. | SELECT COUNT(\*) AS peliculas\_adultos FROM film WHERE rating IN ('R', 'NC-17');  SELECT DISTINCT f.title, f.rating FROM film f JOIN inventory i ON f.film\_id = i.film\_id WHERE f.rating IN ('R', 'NC-17'); |
| **10** | ¿Cuánto cuesta en promedio alquilar una película para público infantil?  *Recuerde siempre usar el formato para el dinero.* | SELECT FORMAT(AVG(rental\_rate), 2) AS promedio\_renta\_infantil FROM film WHERE rating LIKE ‘G’; |
| **11** | Imprima la lista de películas que tienen detrás de cámaras. | SELECT title, special\_features FROM film WHERE special\_features LIKE '%Behind the Scenes%';    *SELECT title, special\_features FROM film WHERE find\_in\_set('Behind the Scenes',special\_features)>0;* |
| **12** | 1. ¿Cuántas películas tienen la característica de que le borraron escenas?      1. Imprima la lista de las películas con escenas eliminadas. | SELECT count(\*) FROM film WHERE find\_in\_set('Deleted Scenes,special\_features)>0;    SELECT title, special\_features FROM film WHERE find\_in\_set('Deleted Scenes',special\_features)>0;    SELECT title, special\_features FROM film WHERE special\_features LIKE '%Deleted Scenes%';    SELECT count(\*) FROM film WHERE special\_features LIKE '%Deleted Scenes%'; |
| **13** | a. Imprima nombre de las películas solo si èstas empiezan por la letra **k**  b. Liste las películas cuyos nombres terminan con vocal **a**  c. Imprima los nombres de las ciudades que contengan la letra **"r"** en cualquier posición.  d. Imprima los nombres de las ciudades que tengan la vocal "**o**" en la segunda posición.  e. Imprima el nombre de las ciudades que empiecen por la letra **“p”** y tengan seis letras únicamente.  f. Liste las ciudades cuyos nombres empiezan con **"a"** y tengan al menos 3 caracteres de longitud total.  g. Encuentre las ciudades cuyos nombres empiezan con **"a"** y terminan con **"a".** | SELECT title FROM film WHERE title LIKE 'K%';    SELECT title FROM film WHERE title LIKE '%a';    SELECT city FROM city WHERE city like '%r%';    SELECT city FROM city WHERE city like '\_\_o%';    SELECT city FROM city WHERE city like 'P\_\_\_\_\_';    SELECT city FROM city WHERE city like 'A\_\_%';  SELECT city FROM city WHERE city like 'A%' and length(city)>3; |
| **14** | Imprima el listado que muestra la cantidad total de clientes registrados en cada ciudad, agrupando los datos para facilitar el análisis por ubicación geográfica. | SELECT ci.city AS ciudad, COUNT(c.customer\_id) AS total\_clientes  FROM customer c  JOIN address a ON c.address\_id = a.address\_id  JOIN city ci ON a.city\_id = ci.city\_id  GROUP BY ci.city  ORDER BY total\_clientes DESC; |
| **15** |  |  |