



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:
Educación

PROGRAMA INGENIERIA DE SISTEMAS

Página 1 | 18

Informe SQL (CodeMusic)

Presentado por: Cristian Camilo Vallejos Bastidas

Presentado: Brayan Arcos



El **Saber** como **Arma** de **Vida**



PROGRAMA INGENIERIA DE SISTEMAS

Página 2 | 18

índice

| | |
|--|----|
| 1. Resumen Ejecutivo..... | 3 |
| 2. Introducción | 4 |
| Contexto y Motivación..... | 4 |
| Alcance del Informe | 4 |
| Objetivos..... | 5 |
| 3. Metodología | 6 |
| Herramientas Utilizadas | 6 |
| Procedimientos | 6 |
| 4. Desarrollo del Informe..... | 8 |
| Consultas SQL | 8 |
| Definición de la Base de Datos | 12 |
| Definición de campos de la tabla “ARTISTA” | 12 |
| Definición de campos de la tabla “CANCION” | 13 |
| Definición de campos de la tabla “ALBUM” | 14 |
| Definición de campos de la tabla “GENEROS” | 14 |
| 5. Análisis y Discusión | 15 |
| 6. Conclusiones | 16 |
| 7. Recomendaciones..... | 17 |
| 8. Referencias..... | 18 |

PROGRAMA INGENIERIA DE SISTEMAS

Página 3 | 18

1. Resumen Ejecutivo

Este informe muestra los principios fundamentales y aplicaciones de bases de datos relacionales en MySQL, enfocándose en la implementación práctica de conceptos clave como DDL (Data Definition Language), DML (Data Manipulation Language), tipos de datos SQL, y funciones y operadores SQL. Se analiza cómo DDL se utiliza para definir y estructurar las bases de datos, incluyendo la creación de tablas y la definición de relaciones entre ellas. Se examina DML para la manipulación de datos, destacando operaciones esenciales como la inserción, actualización, y eliminación de registros. Además, se abordan los tipos de datos SQL utilizados para almacenar diversos tipos de información, y se exploran funciones y operadores SQL avanzados para realizar consultas complejas y optimizar el rendimiento. Este informe proporciona una visión integral de cómo los conceptos teóricos se aplican en el entorno de desarrollo de MySQL para gestionar y manipular datos de manera efectiva.

El Saber como Arma de Vida

PROGRAMA INGENIERIA DE SISTEMAS

Página 4 | 18

2. Introducción

Contexto y Motivación

Las bases de datos relacionales hacen un papel importante en la gestión de la información en muchos campos, desde sistemas empresariales hasta aplicaciones web. MySQL, como uno de los sistemas de gestión de bases de datos más utilizados, ofrece una plataforma robusta y flexible para el almacenamiento, manipulación y consulta de grandes volúmenes de datos. Este informe surge como una necesidad de comprender a fondo cómo los conceptos teóricos de bases de datos relacionales se aplican en un entorno de desarrollo real, permitiendo la optimización y control eficaz de la información. La motivación principal radica en desarrollar habilidades prácticas en la creación y gestión de bases de datos utilizando MySQL, abarcando desde la definición de estructuras hasta la realización de consultas complejas.

Alcance del Informe

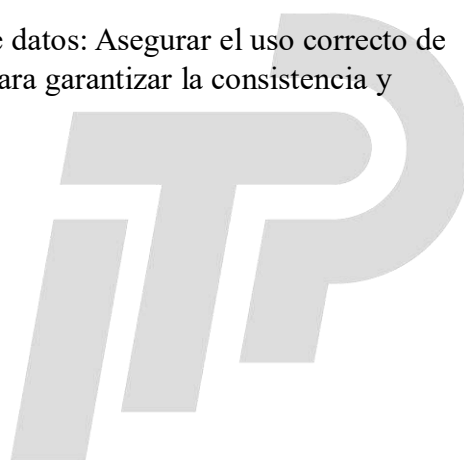
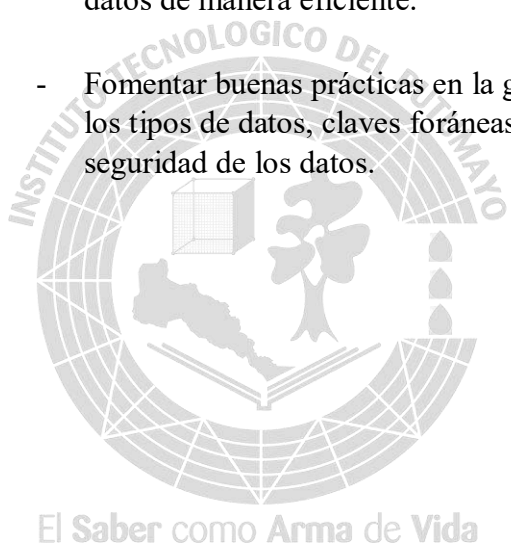
El informe abarca el estudio y la aplicación de los conceptos fundamentales de bases de datos relacionales en el entorno de MySQL. Se explorarán las distintas áreas de SQL, incluyendo el uso de DDL (Data Definition Language) para la creación y modificación de estructuras de datos, DML (Data Manipulation Language) para la manipulación de datos, tipos de datos SQL y funciones para consultas avanzadas. El informe también abordará buenas prácticas en el diseño de bases de datos, centrándose en su integridad y rendimiento. No se incluye el análisis de otros sistemas de gestión de bases de datos ni se profundiza en la administración avanzada de MySQL.

PROGRAMA INGENIERIA DE SISTEMAS

Página 5 | 18

Objetivos

- Aplicar conceptos fundamentales de bases de datos relacionales: Comprender y utilizar DDL, DML, y otros componentes de SQL en la creación y gestión de bases de datos en MySQL.
- Diseñar y optimizar estructuras de bases de datos: Definir esquemas relacionales, garantizando la integridad referencial y optimizando el rendimiento de consultas.
- Implementar consultas SQL avanzadas: Desarrollar habilidades en el uso de funciones y operadores SQL para la ejecución de consultas complejas que extraigan y manipulen datos de manera eficiente.
- Fomentar buenas prácticas en la gestión de bases de datos: Asegurar el uso correcto de los tipos de datos, claves foráneas, y restricciones para garantizar la consistencia y seguridad de los datos.



PROGRAMA INGENIERIA DE SISTEMAS

Página 6 | 18

3. Metodología

Herramientas Utilizadas

Para el desarrollo y gestión de la base de datos relacional, se emplearon las siguientes herramientas:

1. **MySQL:** Sistema de gestión de bases de datos relacional utilizado para crear, gestionar y manipular la base de datos. MySQL proporciona una plataforma robusta y ampliamente adoptada para el almacenamiento y recuperación de datos.
2. **MySQL Workbench:** Entorno gráfico de diseño y administración de bases de datos que permite la creación de esquemas, ejecución de consultas SQL y administración de servidores MySQL de manera intuitiva.
3. **SQL (Structured Query Language):** Lenguaje estándar para la definición, manipulación y consulta de datos dentro de la base de datos. Se utilizaron sus sublenguajes DDL (Data Definition Language) y DML (Data Manipulation Language) para crear tablas, insertar datos y realizar consultas.
4. **Documentación MySQL:** Fuente de referencia para asegurar el correcto uso de las funciones, operadores y características específicas del sistema de gestión de bases de datos.

Procedimientos

El desarrollo de la base de datos relacional en MySQL siguió los siguientes pasos:

1. **Creación de la Base de Datos:**
 - Se inició el proceso utilizando el comando CREATE DATABASE para definir la base de datos principal. Este comando permite establecer un espacio lógico donde se almacenarán las tablas y demás objetos de la base de datos.
2. **Definición de Esquemas y Tablas:**

PROGRAMA INGENIERIA DE SISTEMAS

Página 7 | 18

- Se utilizaron comandos DDL para definir las tablas que conforman la base de datos.
- Se crearon las tablas correspondientes a géneros, artistas, álbumes y canciones, especificando los tipos de datos adecuados para cada campo (como INT, VARCHAR, DATE, y TIME), y estableciendo relaciones mediante claves foráneas para garantizar la integridad referencial.

3. Definición de Relaciones:

- Se implementaron relaciones entre las tablas utilizando claves foráneas (FOREIGN KEY), lo que garantiza que los datos estén correctamente vinculados entre sí. Se utilizaron acciones ON DELETE CASCADE y ON DELETE SET NULL para gestionar la eliminación de registros y mantener la coherencia en las relaciones.

4. Inserción de Datos:

- Con el uso de comandos DML (INSERT INTO), se crearon las tablas con datos de prueba que permiten realizar las consultas y manipulaciones necesarias para la evaluación de la base de datos.

5. Consultas SQL:

- Se realizaron consultas avanzadas utilizando el lenguaje SQL para extraer información relevante de la base de datos. Estas consultas incluyeron el uso de funciones agregadas (COUNT, SUM, etc.), operadores (WHERE, JOIN, etc.), y la optimización de las consultas mediante índices y ajustes en las estructuras de las tablas.

6. Pruebas de Integridad y Rendimiento:

- Finalmente, se llevaron a cabo pruebas para verificar la integridad referencial y la correcta ejecución de las consultas, asegurando que los datos se mantengan consistentes y que las relaciones entre las tablas funcionen adecuadamente.

PROGRAMA INGENIERIA DE SISTEMAS

Página 8 | 18

4. Desarrollo del Informe**Consultas SQL****1. Imprimir las 5 canciones con mayor duración**

```
SELECT sub.titulo_cancion, sub.nombre_artista, sub.duracion
FROM (
    SELECT c.titulo_cancion, ar.nombre_artista, c.duracion
    FROM canciones c
    JOIN artistas ar ON c.id_artista = ar.id_artista
    ORDER BY c.duracion DESC
    LIMIT 5
) AS sub
ORDER BY sub.titulo_cancion;
```

En esta subconsulta, el objetivo es obtener las cinco canciones más largas en términos de duración y mostrar, junto a estas, los nombres de las canciones y de los artistas correspondientes. La consulta se estructura de la siguiente manera:

- Primero, se seleccionan las canciones con su título (titulo_cancion), el nombre del artista (nombre_artista) y la duración de la canción (duracion). Esto se logra mediante una combinación de las tablas canciones y artistas usando un JOIN en la columna id_artista.
- Luego, se ordenan los resultados por la duración de las canciones en orden descendente (de mayor a menor), utilizando ORDER BY c.duracion DESC, y se limita el número de resultados a cinco (LIMIT 5).
- Finalmente, se ordenan estas cinco canciones alfabéticamente por título (ORDER BY sub.titulo_cancion), para presentar los resultados de manera ordenada.

| | titulo_cancion | nombre_artista | duracion |
|---|--------------------|----------------|----------|
| ► | Bendita tu Luz | Maná | 00:04:31 |
| | Flor Pálida | Marc Anthony | 00:04:47 |
| | Labios Compartidos | Maná | 00:05:17 |
| | No | Shakira | 00:04:45 |
| | Safaera | Bad Bunny | 00:04:55 |

PROGRAMA INGENIERIA DE SISTEMAS

Página 9 | 18

2. Imprimir canciones de un artista específico y su álbum si la duración es mayor a 180 segundos

```
SELECT *  
FROM (  
    SELECT c.titulo_cancion, c.duracion, a.titulo_album  
    FROM canciones c  
    JOIN albumes a ON c.id_album = a.id_album  
    JOIN artistas ar ON c.id_artista = ar.id_artista  
    WHERE ar.nombre_artista = 'Shakira'  
) AS sub  
WHERE sub.duracion > 180;
```

Aquí el propósito es mostrar las canciones de un artista específico, en este caso "Shakira", junto con la duración de cada canción y el título del álbum al que pertenece, pero sólo si la duración de la canción es mayor a 180 segundos.

- En primer lugar, se hace una selección de las canciones, sus duraciones y los títulos de los álbumes a los que pertenecen, utilizando un JOIN entre las tablas canciones, albumes y artistas, para obtener la relación entre cada uno.
- Luego, se filtran los resultados para mostrar únicamente las canciones del artista "Shakira" (WHERE ar.nombre_artista = 'Shakira').
- Por último, se aplica una condición adicional para que solo se incluyan las canciones cuya duración sea mayor a 180 segundos (WHERE sub.duracion > 180).

| | titulo_cancion | duracion | titulo_album |
|---|----------------|----------|----------------------|
| ▶ | La Tortura | 00:03:33 | Fijación Oral Vol. 1 |
| | No | 00:04:45 | Fijación Oral Vol. 1 |

PROGRAMA INGENIERIA DE SISTEMAS

Página 10 | 18

3. Imprimir álbum y género excluyendo el género "Pop"

```
SELECT sub.titulo_album, sub.nombre_genero
FROM (
    SELECT a.titulo_album, g.nombre_genero
    FROM albumes a
    JOIN artistas ar ON a.id_artista = ar.id_artista
    JOIN generos g ON ar.id_genero = g.id_genero
) AS sub
WHERE sub.nombre_genero != 'Pop';
```

En esta consulta, se busca obtener el título de los álbumes junto con su género, pero excluyendo aquellos que pertenecen al género "Pop".

- Para lograr esto, se realiza una selección de los títulos de los álbumes y los géneros musicales asociados, mediante una combinación de las tablas albumes, artistas y generos.
- Posteriormente, se filtran los resultados para excluir los géneros que sean "Pop" (WHERE sub.nombre_genero != 'Pop').

| | titulo_album | nombre_genero |
|---|-----------------------|---------------|
| ▶ | YHLQMDLG | Reggaeton |
| | Amar es Combatir | Rock |
| | Cómo Te Voy a Olvidar | Cumbia |
| | 3.0 | Salsa |

PROGRAMA INGENIERIA DE SISTEMAS

Página 11 | 18

4. Mostrar todas las canciones y sus álbumes, incluyendo las canciones sin álbum

```
SELECT c.titulo_cancion, a.titulo_album
FROM canciones c
LEFT JOIN albumes a ON c.id_album = a.id_album
UNION
SELECT c.titulo_cancion, a.titulo_album
FROM canciones c
RIGHT JOIN albumes a ON c.id_album = a.id_album;
```

Esta subconsulta tiene el objetivo de mostrar todas las canciones junto con el álbum al que pertenecen, incluyendo aquellas canciones que no están asociadas a ningún álbum.

- Se utiliza una combinación de un LEFT JOIN y un RIGHT JOIN para asegurar que se obtienen todas las canciones, incluso las que no tienen un álbum asignado.
- El LEFT JOIN asegura que todas las canciones de la tabla canciones sean incluidas, incluso si no tienen un álbum relacionado. Mientras que el RIGHT JOIN asegura que todas las canciones estén presentes, aunque no tengan un álbum asociado.

El Saber como Arma de Vida

| | titulo_cancion | titulo_album |
|---|-----------------------|-----------------------|
| ▶ | La Tortura | Fijación Oral Vol. 1 |
| | No | Fijación Oral Vol. 1 |
| | Safaera | YHLQMDLG |
| | La Difícil | YHLQMDLG |
| | Labios Compartidos | Amar es Combatir |
| | Bendita tu Luz | Amar es Combatir |
| | El Listón de tu Pelo | Cómo Te Voy a Olvidar |
| | Cómo Te Voy a Olvidar | Cómo Te Voy a Olvidar |
| | Vivir mi Vida | 3.0 |
| | Flor Pálida | 3.0 |

PROGRAMA INGENIERIA DE SISTEMAS

Página 12 | 18

Definición de la Base de Datos

Definición de campos de la tabla “ARTISTA”

| Clave/Campo | Tipo | Descripción |
|----------------|----------|---|
| id_artista | INT (PK) | Clave primaria única que identifica de forma inequívoca a cada artista en la base de datos. |
| nombre_artista | VARCHAR | Nombre del artista o grupo musical. |
| id_genero | INT (FK) | Clave foránea que referencia el género musical del artista, asociada a la tabla “géneros”. Permite clasificar a los artistas por su estilo. |

El **Saber** como **Arma** de **Vida**

PROGRAMA INGENIERIA DE SISTEMAS

Página 13 | 18

Definición de campos de la tabla “CANCION”

| Clave/Campo | Tipo | Descripción |
|----------------|----------|--|
| id_cancion | INT (PK) | Clave primaria única que identifica a cada canción en la base de datos. |
| titulo_cancion | VARCHAR | Nombre o título de la canción. |
| duracion | TIME | Duración de la canción (en formato HH:MM). |
| id_album | INT (FK) | Clave foránea que referencia el álbum al que pertenece la canción, asociada a la tabla albumes. |
| id_artista | INT (FK) | Clave foránea que referencia al artista que interpreta la canción, asociada a la tabla artistas. |

El **Saber** como **Arma** de **Vida**

PROGRAMA INGENIERIA DE SISTEMAS

Página 14 | 18

Definición de campos de la tabla “ALBUM”

| Clave/Campo | Tipo | Descripción |
|-------------------|----------|--|
| id_album | INT (PK) | Clave primaria única que identifica de forma inequívoca a cada álbum. |
| Título_album | VARCHAR | Título o nombre del álbum. |
| fecha_lanzamiento | DATE | Año en que se lanzó el álbum. |
| id_artista | INT (FK) | Clave foránea que referencia al artista que creó el álbum, asociada a la tabla artistas. |

Definición de campos de la tabla “GENEROS”

| Clave/Campo | Tipo | Descripción |
|---------------|----------|--|
| id_genero | INT (PK) | Clave primaria única que identifica de forma inequívoca a cada género musical. |
| nombre_genero | VARCHAR | Nombre del género musical (por ejemplo: Pop, Rock, Reggaetón, etc.). |

PROGRAMA INGENIERIA DE SISTEMAS

Página 15 | 18

5. Análisis y Discusión

El desarrollo de la base de datos relacional utilizando MySQL permitió la aplicación práctica de diversos conceptos fundamentales en el manejo de datos. Este proyecto se destacó en la importancia de las estructuras de datos bien diseñadas y las relaciones entre tablas para garantizar la integridad y el correcto funcionamiento de las operaciones de consulta.

Uno de los aspectos más relevantes fue la implementación de claves foráneas para establecer relaciones entre las tablas. Esto permitió vincular entidades como géneros, artistas, álbumes y canciones de manera coherente y eficiente. El uso de tipos de datos adecuados fue también un elemento clave en el diseño. La elección de VARCHAR, INT, DATE, y TIME fue realizada de manera consciente para asegurar un manejo óptimo del almacenamiento y la manipulación de datos. Pero, un análisis más profundo sugiere que, en futuros proyectos, podría considerarse la normalización de los datos en mayor detalle, especialmente si la base de datos crece en complejidad y volumen de información. Esto no solo evitaría la redundancia, sino que también mejoraría el rendimiento de las consultas.

Otro aspecto discutido en el informe fue el uso de consultas SQL avanzadas. Las funciones y operadores SQL utilizados permitieron la extracción de datos específicos de una manera eficiente. No obstante, en bases de datos más grandes, sería recomendable la implementación de índices en las tablas para optimizar el rendimiento de las consultas, especialmente en aquellas que involucran múltiples tablas mediante JOIN.

PROGRAMA INGENIERIA DE SISTEMAS

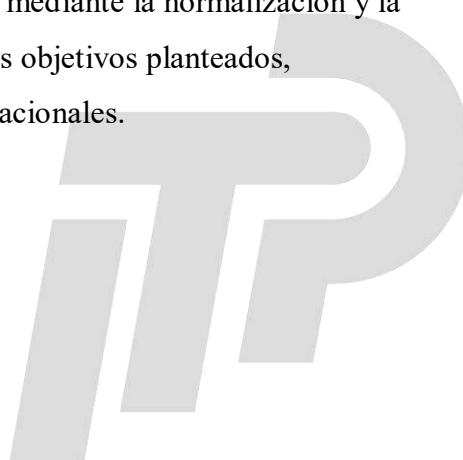
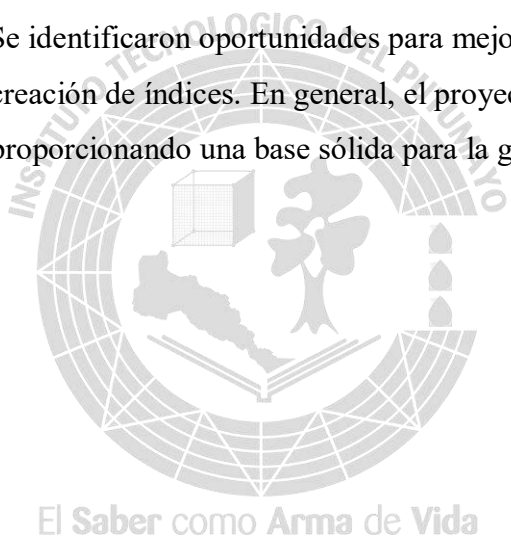
Página 16 | 18

6. Conclusiones

El desarrollo de la base de datos relacional en MySQL permitió aplicar de manera efectiva los conceptos de DDL, DML, y SQL en general, demostrando la importancia de un buen diseño de tablas y relaciones para garantizar la integridad de los datos.

La implementación de claves foráneas y la correcta elección de tipos de datos fueron factores clave para mantener la coherencia y eficiencia en las operaciones.

Se identificaron oportunidades para mejorar el rendimiento mediante la normalización y la creación de índices. En general, el proyecto cumplió con los objetivos planteados, proporcionando una base sólida para la gestión de datos relacionales.



PROGRAMA INGENIERIA DE SISTEMAS

Página 17 | 18

7. Recomendaciones

Se recomienda continuar aplicando técnicas avanzadas de optimización de bases de datos, como la implementación de índices para mejorar el rendimiento en consultas complejas, especialmente en bases de datos con mayor volumen de datos. Sería beneficioso aprender mecanismos adicionales de validación de datos, como restricciones más específicas y triggers, para garantizar la calidad e integridad de la información almacenada. En conclusion , la práctica de normalización debería profundizarse en proyectos futuros, con el fin de evitar redundancias y asegurar una estructura de datos más eficiente y escalable.



PROGRAMA INGENIERIA DE SISTEMAS

Página 18 | 18

8. Referencias

Repositorio (*Carpeta SQL*): https://github.com/Camilo138/repository_sql.git

Video: <https://www.loom.com/share/8903649e88ee4afd9219a512e86d9694?sid=05f5afbd-0185-44f2-9656-c8c9fa4e49b5>

