

Taller repaso SQL

Entregado por:

Johan Fernando Burbano Calpa

Entregado a:

Ing. Brayan Arcos

Bases de datos y almacenamiento masivo

Octavo semestre

Instituto Tecnológico del Putumayo

Mocoa – Putumayo

2024

Contenido

RESUMEN	3
INTRODUCCION	4
ALCANCE Y OBJETIVOS.....	5
METODOLOGIA Y PROCEDIMIENTOS.....	6
DESARROLLO.....	7
CONSULTAS SQL	9
ESQUEMA DE LA BASE DE DATOS	13
ANALISIS Y DISCUSIÓN	14
CONCLUSIONES	14
RECOMENDACIONES.....	14
REFERENCIAS.....	15
LINK.....	15

RESUMEN

En el presente informe abordaremos un análisis de consultas SQL en una base de datos relacional de un pequeño sistema académico. Se evaluaron las diferentes cargas académicas de los estudiantes a través de 3 consultas que permiten filtrar personas con una cantidad específica de materias, analizar la cantidad de estudiantes matriculados en todos los semestres o uno en específico. El informe incluye un análisis del diseño de la base de datos, destacando la elección de claves y la estructura de las tablas.

INTRODUCCION

En la mayoría de instituciones educativas la gestión de las cargas académicas y la optimización de esta información relacionada con los estudiantes es fundamental para un seguimiento eficiente. El análisis de esta información permite una mejor toma de decisiones en cuanto a la distribución de materias y profesores asignados a cada una.

ALCANCE Y OBJETIVOS

Este informe se enfoca en ejecuciones y repaso de consultas SQL, donde con estas consultas analizan la cantidad de cargas académicas de los estudiantes, identificar patrones y determinar cómo estas consultas pueden ayudar a optimizar la gestión académica.

El objetivo principal de este informe es analizar los datos académico a través de consultas SQL donde permitan:

1. Filtrar a los estudiantes por cargas académicas.
2. Identificar el numero de estudiantes matriculados en un solo semestres
3. Identificar el numero de estudiantes matriculados en todos los semestres.
4. Listar las materias mayores cursadas en cualquier semestre.

METODOLOGIA Y PROCEDIMIENTOS

Herramientas utilizadas:

1. **MySQL:** sistema donde se gestiono la base de datos relacional.

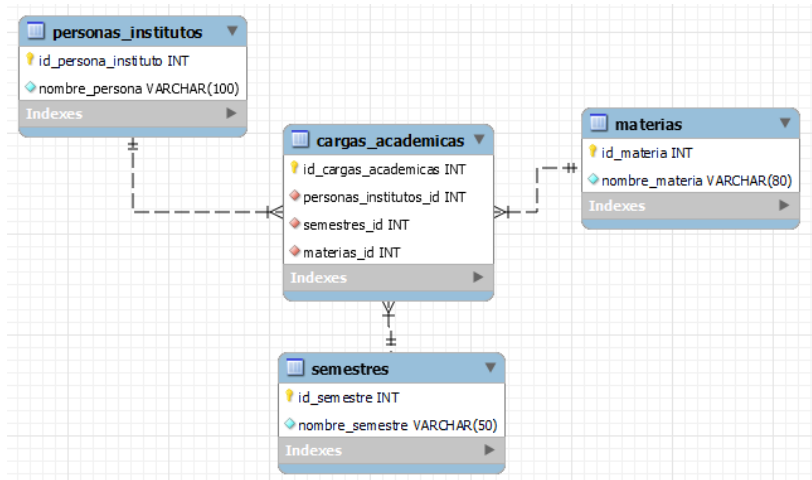


Figura 1

2. **MySQL Server:** se permitió almacenar los datos de manera local para desarrollar las diferentes consultas y la organización de la información

```

-- Crear la base de datos
• CREATE DATABASE IF NOT EXISTS insti;
• USE insti;

-- Crear la tabla semestres
• DROP TABLE IF EXISTS `semestres`;
• CREATE TABLE `semestres` (
  `id_semestre` INT NOT NULL AUTO_INCREMENT,
  `nombre_semestre` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id_semestre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
  
```

Figura 2

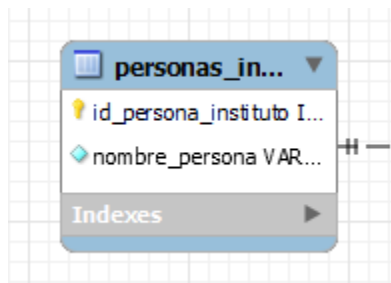
En el procedimiento de este análisis se creo una base de datos llamada **insti** con tablas que incluyen **personas_institutos**, **semestres**, **materias** y **cargas_academicas**. Se lleno la información de las tablas con datos no verdaderos para la ejecución de diversas consultas SQL para analizar la información de las cargas académicas.

DESARROLLO

La base de datos **insti** incluye las siguientes tablas:

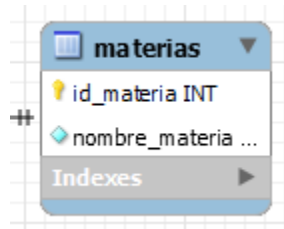
- **Personas_institutos**: en esta tabla se llena la información básica los estudiantes.

Su llave primaria es “**id_persona_instituto**”.



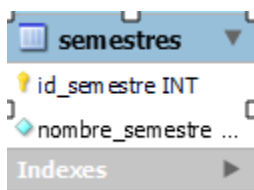
- **Materias**: Se llena las diferentes materias de todos los semestres.

Su llave primaria es “**id_materia**”




- **Semestres:** Todos los semestres o niveles formativos de la institución.

Su llave primaria es “**id semestre**”.



- **Cargas_academicas:** Tabla principal donde se realizan 3 relaciones de 1 a muchos.

 Table Name: <input type="text" value="cargas_academicas"/>	
Foreign Key Name	Referenced Table
fk_carga_academica_per...	'insti', 'personas_institutos'
fk_carga_academica_se...	'insti', 'semestres'
fk_carga_academica_mat...	'insti', 'materias'

CONSULTAS SQL

1. Esta consulta filtra una lista de personas que tienen exactamente la cantidad de cargas académicas que quisiéramos filtrar donde se muestra el semestre y la materia asociada que tiene dicha persona.

```
SELECT pi.nombre_persona, s.nombre_semestre, m.nombre_materia
FROM cargas_academicas ca
JOIN personas_institutos pi ON ca.personas_institutos_id = pi.id_persona_instituto
JOIN semestres s ON ca.semestres_id = s.id_semestre
JOIN materias m ON ca.materias_id = m.id_materia
WHERE pi.id_persona_instituto IN (
    SELECT ca_sub.personas_institutos_id
    FROM cargas_academicas ca_sub
    GROUP BY ca_sub.personas_institutos_id
    HAVING COUNT(*) = 1
)
ORDER BY pi.nombre_persona, s.nombre_semestre, m.nombre_materia;
```

Como primera parte hacemos JOIN para combinar datos de diferentes tablas, navegamos en personas_institutos, materias y semestres que están relacionadas con cargas_academicas, de las cuales traeremos el nombre del estudiante, el nombre del semestre y el nombre de la materia, después de esto pasamos a una subconsulta donde con un **WHERE** selecciona solo los estudiantes que tiene exactamente una materia asignada primero con el **GROUP BY** a los estudiantes según el id de las personas luego con el **HAVING** y el **COUNT** hacemos un filtro para aquellos estudiantes que tiene solo una carga académica. Después finalmente con el otro **ORDEN BY** se ordena los resultados del nombre del estudiante, seguidos del nombre del semestre y luego el nombre de la materia.

resultado

Result Grid		Filter Rows:		Export:
	nombre_persona	nombre_semestre	nombre_materia	
▶	Johan Burbano	Semestre 1	Matemáticas	

2. En la siguiente consulta tenemos como resultado la cantidad de cargas académicas que tenga cierto semestre que queramos filtrar.

```
SELECT s.nombre_semestre AS Semestre, COUNT(*) as Cantidad
FROM cargas_academicas ca
JOIN semestres s ON ca.semestres_id = s.id_semestre
WHERE s.nombre_semestre = 'Semestre 1';
```

En esta subconsulta parte de **cargas_academicas** donde con el contador se sabrá cuantas cargas académicas hay en ese semestre, con los **JOIN** entraremos a las tablas de semestre mediante el campo común **semestre_id** esto nos permite conectar las cargas académicas con los semestres correspondientes, finalmente con nuestro **WHERE** se indica que en la tabla **semestres** del campo **nombre_semestre** solo será llamado **Semestre 1**.

Result Grid		Filter Rows:		Export:
	Semestre	Cantidad		
▶	Semestre 1	5		

3. esta subconsulta es similar a la anterior, pero esta ya llama a todos los semestres con su cantidad de estudiantes matriculados.

```
SELECT s.nombre_semestre, COUNT(ca.personas_institutos_id) AS total_estudiantes
FROM cargas_academicas ca
JOIN semestres s ON ca.semestres_id = s.id_semestre
GROUP BY s.id_semestre;
```

Así como la anterior busca dentro de la tabla **cargas_academicas** mediante el contador busca cada id de las personas donde los almacena en nuestra columna llamada **total_estudiantes** y ya solo basta agrupar la tabla **semestre** por medio de los id de cada uno.

nombre_semestre	total_estudiantes
Semestre 1	5
Semestre 2	4
Semestre 3	2
Semestre 4	2
Semestre 5	4
Semestre 6	4
Semestre 7	4
Semestre 8	4
Semestre 9	3
Semestre 10	3
Semestre 11	3
Semestre 12	3

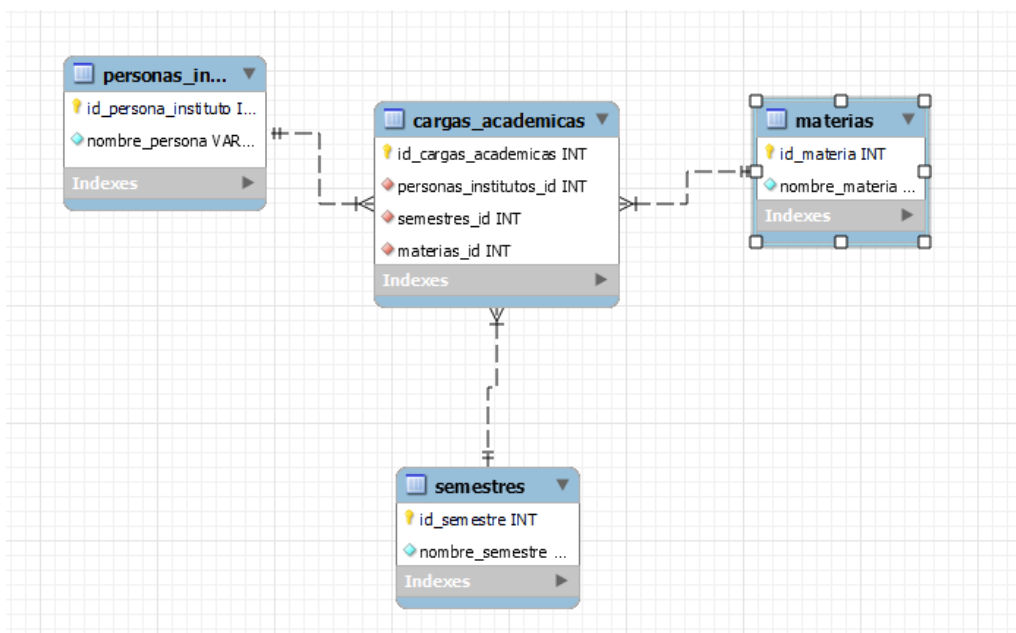
4. Con esta última subconsulta listaremos las materias que están siendo cursadas por mas estudiantes en cualquier semestre.

```
SELECT m.nombre_materia, COUNT(ca.id_cargas_academicas) AS total_estudiantes
FROM cargas_academicas ca
JOIN materias m ON ca.materias_id = m.id_materia
GROUP BY m.nombre_materia
HAVING COUNT(ca.id_cargas_academicas) > 5;
```

Para empezar esta consulta entramos a nuestra tabla principal donde mediante los **JOIN** consumiremos la tabla materias para traer el nombre y con nuestro contador mediante los id de las cargas académicas almacenaremos esa información para luego pasar a un **HAVING** donde filtra los resultados para mostrar solo aquellas materias que están siendo cursadas por más estudiantes.

	nombre_materia	total_estudiantes
▶	Matemáticas	6
	Física	6
	Biología	6

ESQUEMA DE LA BASE DE DATOS



Consideraciones de Diseño

Se eligieron claves primarias autoincrementales para garantizar la unicidad en las tablas, como **id_persona_instituto**, **id_semestre**, e **id_materia**. Las claves foráneas en la tabla **cargas_academicas** aseguran la integridad referencial entre los estudiantes, semestres y materias.

ANÁLISIS Y DISCUSIÓN

Los resultados muestran que la mayoría de los estudiantes están distribuidos equitativamente entre los semestres. Además, las materias con más de un estudiante inscrito revelan tendencias en las preferencias académicas, lo que puede ser útil para la planificación de recursos y docentes.

CONCLUSIONES

El análisis de las cargas académicas proporciona información clave sobre la distribución de estudiantes y materias. Este tipo de análisis permite optimizar la oferta académica y gestionar de manera más efectiva los recursos de la institución mediante la ayuda de software especializados con un buen manejo de base de datos.

RECOMENDACIONES

- Implementar consultas adicionales para identificar a los estudiantes con múltiples cargas académicas y analizar su rendimiento.
- Ampliar el análisis para incluir otras métricas, como el desempeño académico de los estudiantes por semestre o materia.
- Cuestionares mensuales referentes a las clases y materias.

REFERENCIAS

<https://dev.mysql.com/doc/workbench/en/>

<https://www.chucksacademy.com/es/topic/db-basic/advanced-sql-queries>

LINK

<https://youtu.be/dOTtMJI4uOA>