

PREENTREGA 2 PROYECTO SQL

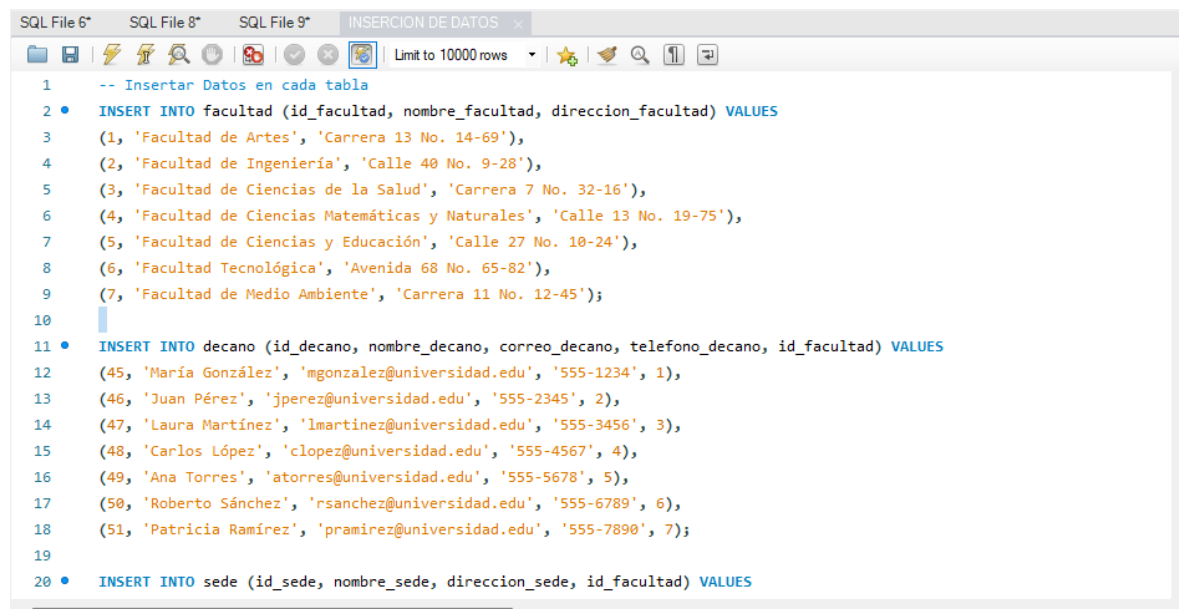
CODERHOUSE

Diana Milena Alvarado Pinzon

CONTENIDO

- Script de inserción de datos en las bases.
- Listado de Vistas más una descripción detallada, su objetivo, y qué tablas las componen.
- Listado de Funciones que incluyan una descripción detallada, el objetivo para la cual fueron creadas y qué datos o tablas manipulan y/o son implementadas.
- Listado de Stored Procedures con una descripción detallada, qué objetivo o beneficio aportan al proyecto, y las tablas que lo componen y/o tablas con las que interactúa.

Script de inserción:



```
SQL File 6*  SQL File 8*  SQL File 9*  INSERCIÓN DE DATOS
Limit to 10000 rows

1  -- Insertar Datos en cada tabla
2  • INSERT INTO facultad (id_facultad, nombre_facultad, direccion_facultad) VALUES
3  (1, 'Facultad de Artes', 'Carrera 13 No. 14-69'),
4  (2, 'Facultad de Ingeniería', 'Calle 40 No. 9-28'),
5  (3, 'Facultad de Ciencias de la Salud', 'Carrera 7 No. 32-16'),
6  (4, 'Facultad de Ciencias Matemáticas y Naturales', 'Calle 13 No. 19-75'),
7  (5, 'Facultad de Ciencias y Educación', 'Calle 27 No. 10-24'),
8  (6, 'Facultad Tecnológica', 'Avenida 68 No. 65-82'),
9  (7, 'Facultad de Medio Ambiente', 'Carrera 11 No. 12-45');
10
11 • INSERT INTO decano (id_decano, nombre_decano, correo_decano, telefono_decano, id_facultad) VALUES
12 (45, 'María González', 'mgonzalez@universidad.edu', '555-1234', 1),
13 (46, 'Juan Pérez', 'jperez@universidad.edu', '555-2345', 2),
14 (47, 'Laura Martínez', 'lmartinez@universidad.edu', '555-3456', 3),
15 (48, 'Carlos López', 'clopez@universidad.edu', '555-4567', 4),
16 (49, 'Ana Torres', 'atorres@universidad.edu', '555-5678', 5),
17 (50, 'Roberto Sánchez', 'rsanchez@universidad.edu', '555-6789', 6),
18 (51, 'Patricia Ramírez', 'pramirez@universidad.edu', '555-7890', 7);
19
20 • INSERT INTO sede (id_sede, nombre_sede, direccion_sede, id_facultad) VALUES
```

Para la inserción de los datos me guie en la universidad que estudio actualmente en Bogotá Colombia, hay 7 facultades, 1 decano por facultad, 8 sedes, añadi 30 carreras, 60 estudiantes y 20 profesores.

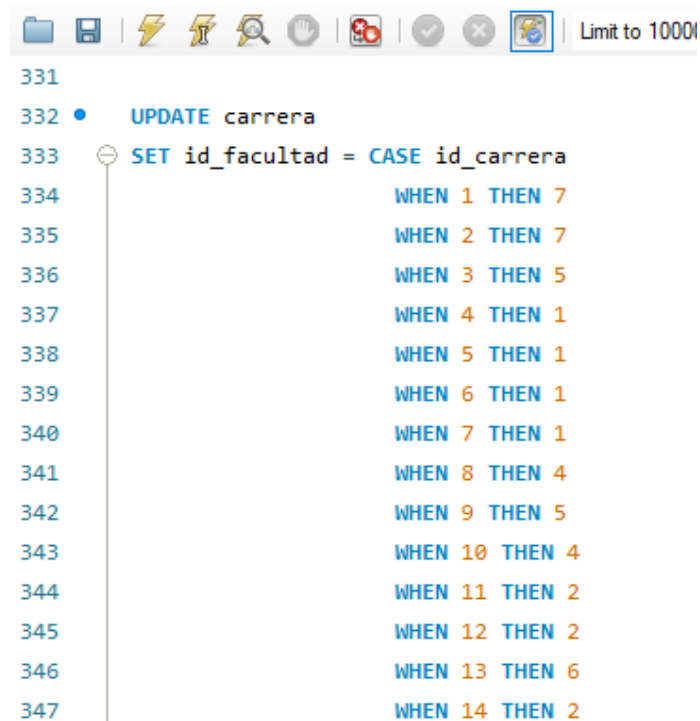
Para hacer las vistas tuve que crear otra tabla(asignatura_estudiante) que también contendrá una columna 'fecha_inscripcion' para registrar la fecha en que

cada estudiante se inscribió en cada asignatura, esto para una creación de una vista más detallada, con más información.

```
-- Creacion de tabla intermedia
CREATE TABLE asignatura_estudiante (
    id_asignatura INT,
    id_estudiante INT,
    fecha_inscripcion DATE,
    PRIMARY KEY (id_estudiante, id_asignatura),
    FOREIGN KEY (id_asignatura) REFERENCES asignatura(id_asignatura),
    FOREIGN KEY (id_estudiante) REFERENCES estudiante(id_estudiante)
);
```

Además, modifique la tabla carrera, agregándole una nueva columna (id_facultad) para que haya una relación de carrera con facultad, solo para mejorar la tabla. Todo esta anexo. (INSERCIÓN DE DATOS)

```
-- Modifique tabla carrera
ALTER TABLE carrera
ADD COLUMN id_facultad INT;
```



The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. Below the toolbar, a SQL query is displayed with line numbers 331 through 347. The query is an UPDATE statement for the 'carrera' table, using a CASE statement to assign values to the 'id_facultad' column based on the 'id_carrera' values. The values are: 1 and 2 map to 7, 3 maps to 5, 4 maps to 1, 5 and 6 map to 1, 7 maps to 1, 8 maps to 4, 9 maps to 5, 10 maps to 4, 11 and 12 map to 2, 13 maps to 6, and 14 maps to 2.

```
331
332 • UPDATE carrera
333 SET id_facultad = CASE id_carrera
334     WHEN 1 THEN 7
335     WHEN 2 THEN 7
336     WHEN 3 THEN 5
337     WHEN 4 THEN 1
338     WHEN 5 THEN 1
339     WHEN 6 THEN 1
340     WHEN 7 THEN 1
341     WHEN 8 THEN 4
342     WHEN 9 THEN 5
343     WHEN 10 THEN 4
344     WHEN 11 THEN 2
345     WHEN 12 THEN 2
346     WHEN 13 THEN 6
347     WHEN 14 THEN 2
```

Listado de vistas

- **Vista 1:** vista_estudiantes_facultades

Descripción: Proporciona información sobre los estudiantes junto con el nombre de la carrera y la facultad a la que pertenecen.

Tablas involucradas: estudiante, carrera, carrera_facultad, facultad.

- **Vista 2:** vista_detalle_asignaturas

Descripción: Esta vista muestra detalles completos de las asignaturas, incluyendo el ID y nombre de la asignatura, el número de créditos, el nombre de la carrera a la que pertenece, el nombre del profesor que imparte la asignatura y el nombre de la facultad a la que pertenece la carrera.

Tablas involucradas: asignatura, carrera, profesor, facultad

- **Vista 3:** vista_asig_inscr_junio

Descripción: Esta vista muestra las asignaturas en las que se inscribieron los estudiantes durante el mes de junio.

Tablas involucradas: estudiante, asignatura, asignatura_estudiante.

Listado de Funciones

- **Función 1:** ObtenerNombreCarrera

Propósito: La función ObtenerNombreCarrera está diseñada para obtener el nombre de la carrera a la que pertenece un estudiante específico en función del id_estudiante. Esta función es útil para consultar rápidamente la carrera de un estudiante sin necesidad de escribir consultas SQL complejas cada vez que se necesite esta información.

Parámetros de Entrada:

- *p_id_estudiante (INT)*: El identificador único del estudiante cuyo nombre de carrera se desea obtener.

Valor de Retorno:

- *VARCHAR(250)*: El nombre de la carrera a la que pertenece el estudiante.

Tablas Involucradas:

Carrera: Esta tabla contiene la información de las carreras ofrecidas por la institución.

- Columnas relevantes: *id_carrera*, *nombre_carrera*

Estudiante: Esta tabla contiene la información de los estudiantes inscritos en la institución.

- Columnas relevantes: *id_estudiante*, *id_carrera*

- **Función 2: ContarAsignaturasInscritas**

Propósito: La función *ContarAsignaturasInscritas* cuenta el número de asignaturas en las que está inscrito un estudiante específico, basado en su *id_estudiante*.

Parámetros de Entrada:

- *p_id_estudiante (INT)*: El identificador único del estudiante del cual se desea contar las asignaturas inscritas.

Valor de Retorno:

- *INT*: El número de asignaturas inscritas por el estudiante.

Tablas Involucradas:

Asignatura_estudiante: Esta tabla relaciona los estudiantes con las asignaturas en las que están inscritos.

- Columnas relevantes: *id_asignatura*, *id_estudiante*

- **Función 3: ContarEstudiantesPorCarrera**

Propósito: La función *ContarEstudiantesPorCarrera* cuenta el número de estudiantes inscritos en una carrera específica, identificada por su *id_carrera*.

Parámetros de Entrada:

- *p_id_carrera (INT)*: El identificador único de la carrera para la cual se desea contar los estudiantes inscritos.

Valor de Retorno:

- *INT*: Número de estudiantes inscritos en la carrera especificada por *p_id_carrera*.

Tablas Involucradas:

Estudiante: Tabla que contiene la información de todos los estudiantes.

- Columnas relevantes: *id_estudiante*, *nombre_estudiante*, *id_carrera*

Listado de Stored Procedures

- **Procedimiento 1: EliminarProfesor**
 - **Descripción:** Este procedimiento elimina un profesor de la base de datos dado su *id_profesor*.
 - **Tablas involucradas:** *profesor*: Contiene los datos de los profesores, incluyendo *id_profesor*, *id_facultad*, *telefono_profesor*, *correo_profesor*, y *nombre_profesor*.
- **Procedimiento 2: ActualizarCorreoEstudiante**
 - **Descripción:** permite actualizar el correo electrónico de un estudiante en la base de datos mediante su ID (*p_id_estudiante*) y el nuevo correo electrónico (*p_nuevo_correo*).
 - **Tablas involucradas:** *Tabla estudiante*: Contiene los datos de los estudiantes, incluyendo *id_estudiante*, *nombre_estudiante*, *correo_estudiante*, *direccion_estudiante*, y *id_carrera*.
- **Procedimiento 3: RegistrarFacultad**
 - **Descripción:** permite registrar una nueva facultad en la base de datos proporcionando su ID (*p_id_facultad*), nombre (*p_nombre_facultad*), y dirección (*p_direccion_facultad*).
 - **Tablas involucradas:** Contiene los datos de las facultades, incluyendo *id_facultad*, *nombre_facultad*, y *direccion_facultad*.

Listado Triggers:

Trigger 1: *RegistrarFechaInscripcion*

- **Descripción:** Registra la fecha de inscripción cuando un estudiante se inscribe en una asignatura.
- **Tablas Involucradas:** asignatura_estudiante.

Trigger 2: EvitarEliminacionEstudiante

- **Descripción:** Previene la eliminación de un estudiante si tiene asignaturas asignadas.
- **Tablas Involucradas:** estudiante, asignatura_estudiante.

ARCHIVO SQL:

https://github.com/Dianalvaradop/proyectoSQL_coder_house.git