

**Base de datos con SQL SERVER**

**Proyecto Final:**

**Gestión de Matrícula Universitaria**

**Autor:**

Prado Acuña, Robert Anthony

**Docente:**

Ing. Rivera Vergaray, Kevin

Lima, Perú

2024

1. **Iniciación: Definición de los objetivos y el alcance inicial de la base de datos.**

**Título del Problema: Gestión de Matrícula Universitaria**

**Descripción del Problema:**

La Universidad necesita un sistema para gestionar la matrícula de sus estudiantes en diversas carreras y materias. Este sistema debe registrar información sobre estudiantes, profesores, materias, programas y períodos académicos. Además, se espera que el sistema facilite la asignación de calificaciones y permita generar reportes de desempeño por estudiante y curso.

**Requerimientos Funcionales:**

1. Registrar información básica de los estudiantes (nombre completo, fecha de nacimiento, email, teléfono, dirección y carrera inscrita).
2. Gestionar el proceso de matrícula:

* Fecha de inicio y fin de matrícula.
* Tener referencia de la cantidad de estudiantes que se van a matricular
* Saber que tipo de matricula se esta efectuando

1. Verificar la cantidad de materias y/o cursos que se habilitaran por cada periodo académico.
2. Gestionar la plantilla de profesores que dictaran cada materia
3. Habilitar la curricula para cada estudiante
4. Registrar a que período académico se está matriculando los estudiantes.
5. Generar reportes hacia cada escuela profesional sobre la cantidad de alumnado se ha matriculado respecto a la curricula enviada.

### 

### **Entidades Identificadas:**

1. **Estudiante:** Información básica del estudiante.
2. **Matrícula:** Información sobre las fechas de inicio y fin.
3. **Profesor:** Información básica del profesor.
4. **Materia:** Información básica de la materia con una pequeña descripción.
5. **Periodo Académico:** Información sobre cada periodo académico.

**Tablas y Campos Relevantes:**

**1. Tabla Estudiantes:** Contiene la información básica de todos los estudiantes.

* Nombre completo
* Código de estudiante
* Fecha de nacimiento
* Email
* Teléfono
* Dirección
* Carrera Inscrita

**2. Tabla Profesores:** Contiene la información básica de los profesores.

* Nombre completo
* Código de profesor
* Email
* Teléfono
* Especialidad

**3. Tabla Curricula:** Contiene la información básica de cada curricula.

* Nombre
* Descripción

**4. Tabla Escuela Profesional:** Contiene la información básica de cada escuela profesional.

* Código de Escuela Profesional
* Nombre
* Dirección

**5. Tabla Matrícula:** Contiene la información básica de cada matrícula

* Fecha de inicio
* Fecha fin
* Created\_at
* Updated\_at
* Deleted\_at
* Created\_by
* Updated\_by
* Deleted\_by

**6. Tabla de Materia:** Contiene información básica de cada materia.

* Código de materia
* Nombre
* Número de créditos

**7. Tabla de Periodo Académico:** Contiene información básica de cada periodo académico.

* Número de periodo
* Fecha de inicio
* Fecha fin

**8. Tabla de Tipo de Matrícula:** Contiene información básica de cada tipo de matrícula.

* Nombre
* Descripción

**Desafíos:**

* **Volumen de Estudiantes:** Durante los periodos de matrícula, el alto volumen de estudiantes puede sobrecargar los sistemas y recursos disponibles, lo que puede llevar a fallos técnicos o largos tiempos de espera.
* **Personalización y flexibilidad:** Los estudiantes tienen necesidades diversas, como horarios personalizados, materias optativas y requisitos de programas específicos.
* **Gestión de cupos y disponibilidad:** Asegurar que los cupos en materias o secciones no se llenen de manera desigual o antes de que todos los estudiantes tengan la oportunidad de inscribirse.
* **Integración con otros sistemas:** La matrícula no es un proceso aislado; debe estar conectada con sistemas de pagos, registros académicos y calendarios institucionales.

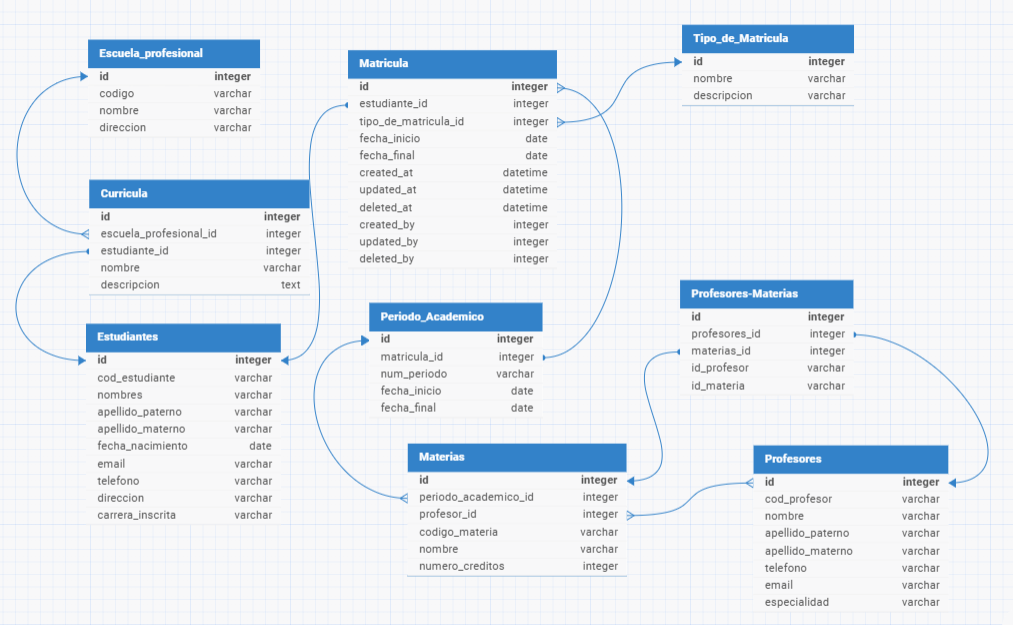
**Beneficios Esperados:**

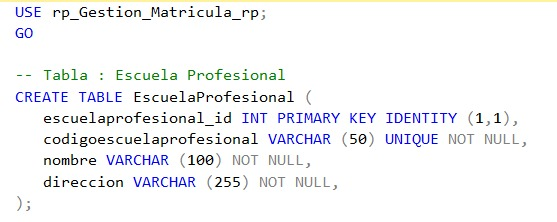
* Implementar sistemas de gestión escalables, como plataformas en la nube, y establecer horarios escalonados para la matrícula.
* Utilizar software de gestión que permita personalización y escenarios dinámicos según el perfil del estudiante.
* Implementar estrategias de asignación de cupos, como matrículas por prioridad, y utilizar sistemas de actualización en tiempo real.
* Usar plataformas integradas que faciliten la comunicación entre diferentes módulos (ERP académico).

1. **Análisis: Diseño Conceptual de la base de datos.**

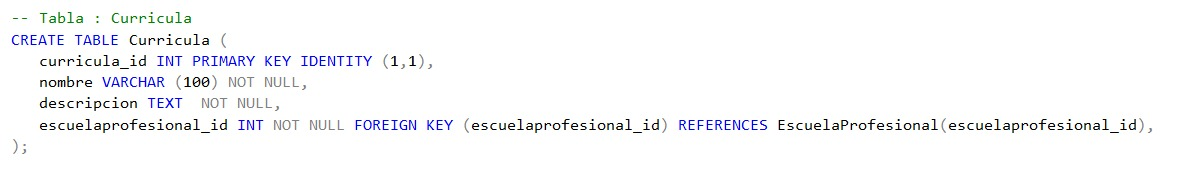
**GESTIÓN DE MATRICULA UNIVERSITARIA**

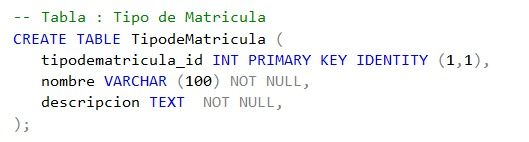


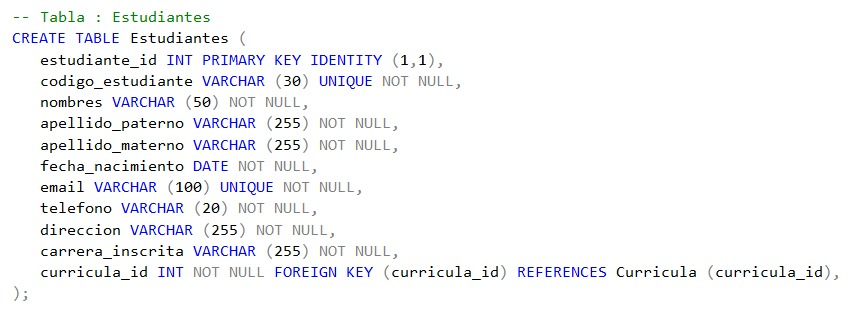
1. **Diseño**: **Definición del Modelo Lógico de la Base de Datos.**
2. **Construcción: Desarrollo de las sentencias SQL que permiten la construcción de las tablas y demás estructuras de la Base de Datos en SQL SERVER.**

Creación de la tabla **Escuela Profesional**:

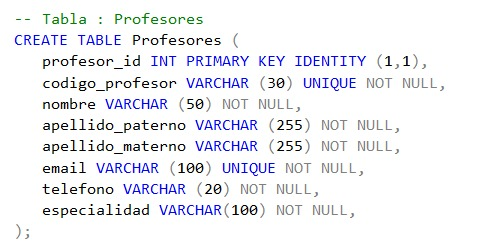
Creación de la tabla **Curricula**:



Creación de la tabla **Tipo de Matrícula**:

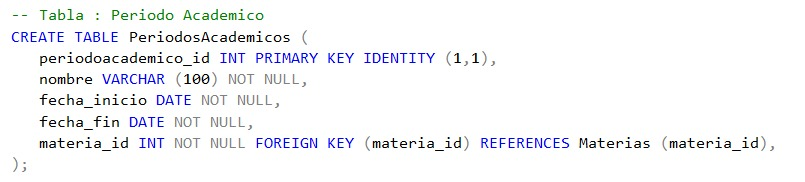
Creación de la tabla **Estudiantes**:

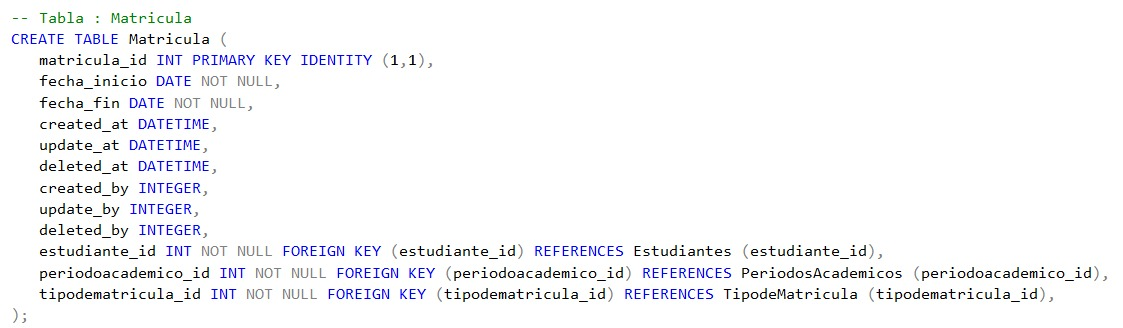
Creación de la tabla **Profesores**:

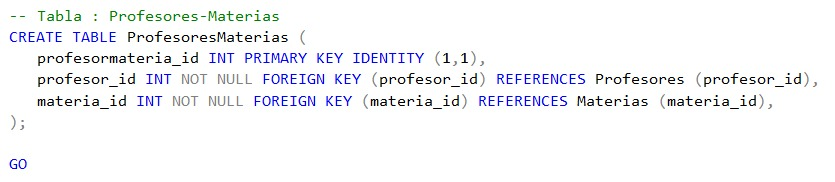


Creación de la tabla **Materias**:

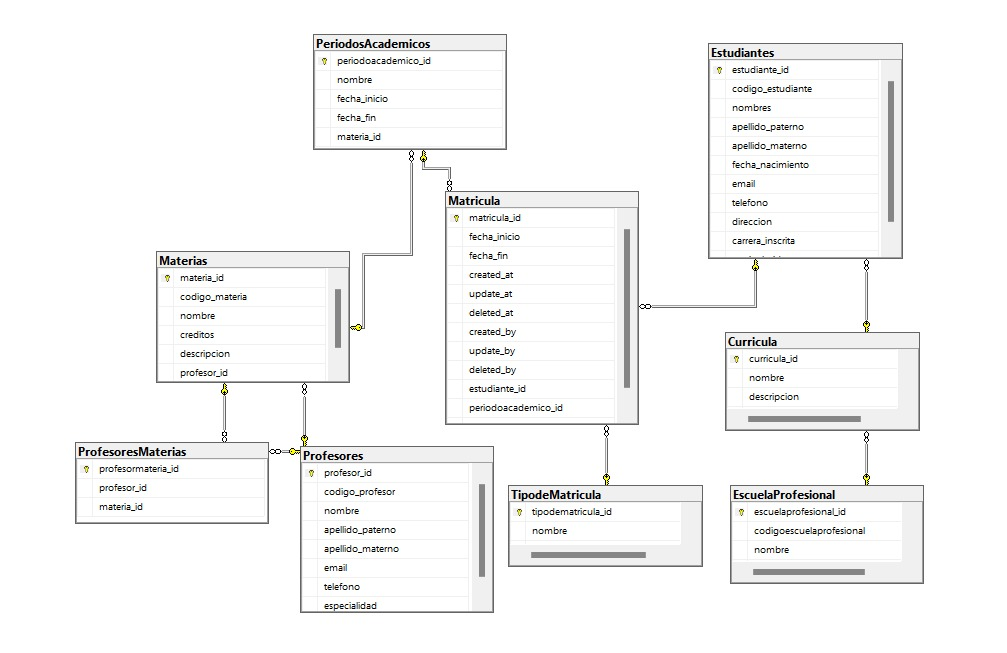
Creación de la tabla **Periodo Académico:**

****

Creación de la tabla **Matrícula**:

Creación de la tabla **Profesores-Materias**:

**V. Producto: Diseño Físico, Script de la Base de datos con inserción de datos (Mínimo 5 registros por tablas independientes o no transaccionales y mínimo 15 registros por cada tabla transaccional).**

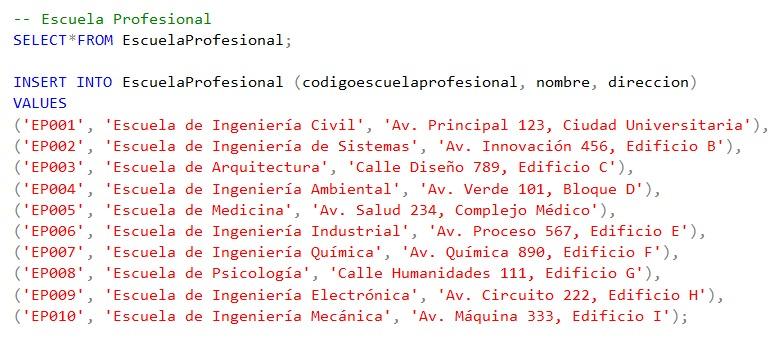
DISEÑO FÍSICO

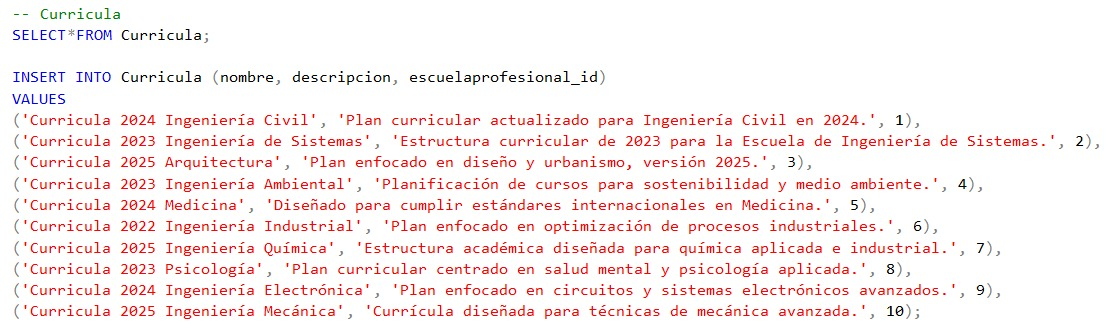
**Link del SCRIPT**

Se adjunta link de mi repositorio en **GITHUB** donde está incluido el Script:

* https://github.com/RobertPrado01/rp\_Gestion\_Matricula\_Universitaria\_bda\_6

INSERCIÓN DE DATOS

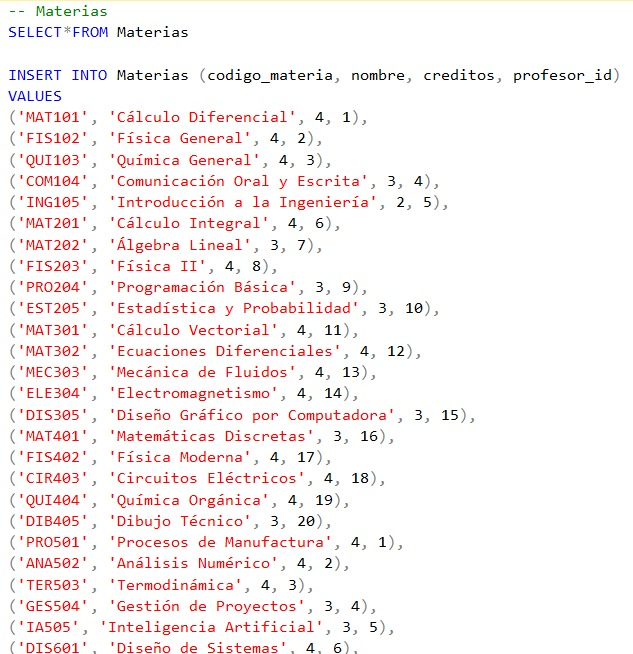
****Tabla **Escuela Profesional**

****Tabla **Curricula**

****Tabla **Estudiantes**

****Tabla **Profesores**

Tabla **Materias**

****

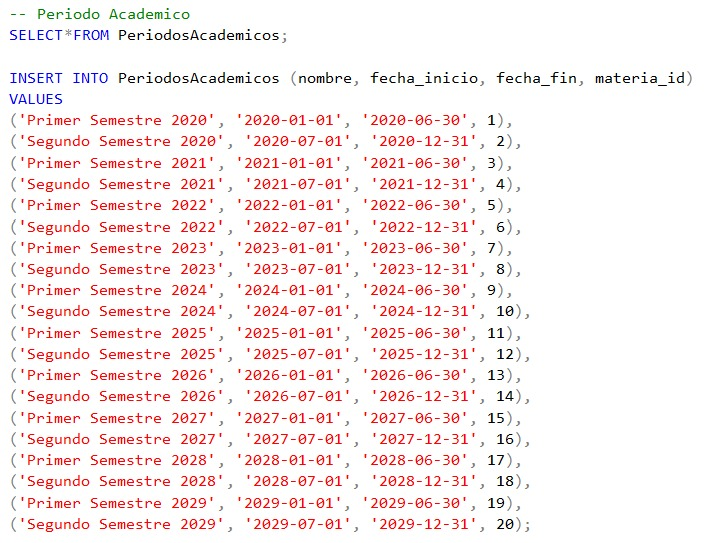
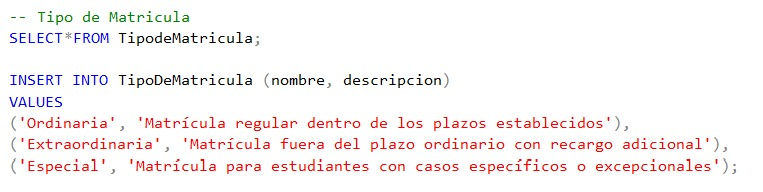
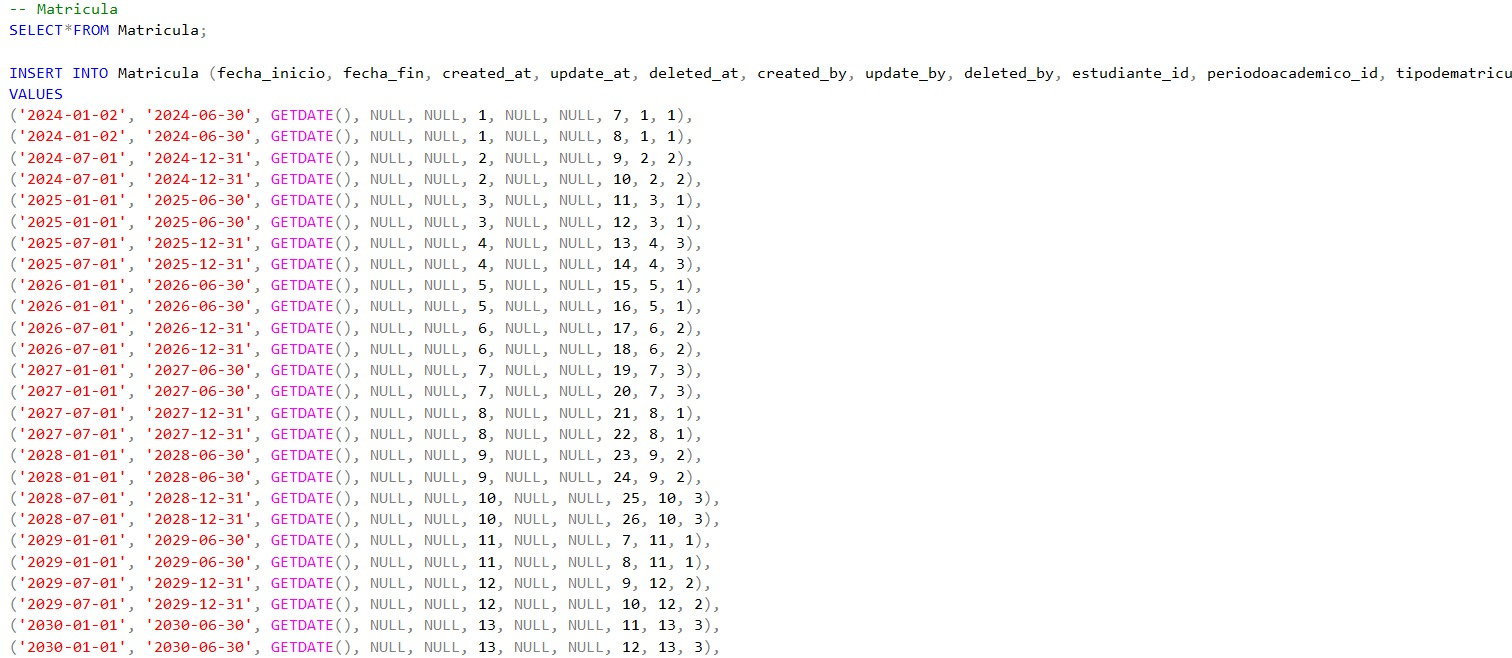
****Tabla **Periodo Académico**

Tabla **Tipo de Matricula**

****

****Tabla **Matricula**

1. **Anexos: Evidencias de Consultas, funciones, procedimientos almacenados, vistas, etc. Realizados sobre la base de datos implementada. (5 ejercicios por integrante del grupo, Integrantes individuales (7 ejercicios))**

**-- CONSULTAS SQL**

**-- 1° CONSULTA, TODAS LAS ESCUELAS PROFESIONALES CON SUS NOMBRES Y DIRECCIONES ORDENADOS ALFABETICAMENTE.**

SELECT

codigoescuelaprofesional, nombre, direccion

FROM EscuelaProfesional

ORDER BY nombre;

**-- 2° CONSULTA, OBTEN EL LISTADO DE ESTUDIANTES JUNTO CON EL NOMBRE DE LA CURRICULA A LA QUE ESTAN INSCRITOS.**

SELECT

E. codigo\_estudiante,

CONCAT (E. nombres, ' ', E. apellido\_paterno,' ', E. apellido\_materno) AS 'Nombre completo',

C.nombre AS curricula\_nombre

FROM Estudiantes E

INNER JOIN Curricula C ON E. curricula\_id = C.curricula\_id;

**-- 3° CONSULTA, MUESTRA EL LISTADO DE PROFESORES Y LAS MATERIAS QUE DICTAN, INCLUYENDO SUS CREDITOS.**

SELECT

CONCAT (P. nombre,' ',P.apellido\_paterno,' ',P.apellido\_materno) AS 'Profesor nombre completo',

P. especialidad,

M.nombre AS 'materia',

M.creditos

FROM Profesores P

INNER JOIN Materias M ON P. profesor\_id = M. profesor\_id;

**-- 4° CONSULTA, OBTEN EL NOMBRE COMPLETO DE LOS ESTUDIANTES Y LAS FECHAS DE MATRICULA DEL PERIODO ACADEMICO ACTUAL.**

SELECT

CONCAT (E. nombres, ' ', E. apellido\_paterno,' ', E. apellido\_materno) AS 'Nombre completo',

M.fecha\_inicio,

M.fecha\_fin

FROM Estudiantes E

INNER JOIN Matrícula M ON E. estudiante\_id = M. estudiante\_id

INNER JOIN PeriodosAcademicos PA ON M. periodoacademico\_id = PA. periodoacademico\_id;

**-- 5° CONSULTA, MUESTRA LAS MATERIAS QUE SE DICTAN EN EL PERIODO ACADEMICO ACTUAL CON SUS FECHAS DE INICIO Y FIN.**

SELECT

PA. nombre AS 'Periodo Academico',

M.nombre AS 'Materia',

PA. fecha\_inicio,

PA. fecha\_fin

FROM PeriodosAcademicos PA

INNER JOIN Materias M ON PA. materia\_id = M. materia\_id;

**-- 6° CONSULTA, CAMBIA EL NUMERO DE CREDITOS DE LA MATERIA "CALCULO DIFERENCIAL" A 6.**

UPDATE Materias

SET creditos = 6

WHERE nombre = 'Cálculo Diferencial';

**-- 7° CONSULTA, OBTENER EL NUMERO TOTAL DE ESTUDIANTES POR CADA CURRICULA.**

SELECT

C.nombre AS 'Curricula',

COUNT (E. estudiante\_id) AS 'Total estudiantes'

FROM Curricula C

LEFT JOIN Estudiantes E ON C. curricula\_id = E. curricula\_id

GROUP BY C. nombre;

**-- 8° CONSULTA, OBTEN EL LISTADO DE ESTUDIANTES QUE AUN NO TIENEN MATRICULA REGISTRADA.**

SELECT

E. estudiante\_id,

CONCAT (E. nombres, ' ', E. apellido\_paterno,' ', E. apellido\_materno) AS 'Nombre completo'

FROM Estudiantes E

LEFT JOIN Matricula M ON E. estudiante\_id = M. estudiante\_id

WHERE M. matricula\_id IS NULL;

**-- 9° CONSULTA, MUESTRA EL ESTUDIANTE JUNTO CON EL TOTAL DE CREDITOS DE LAS MATERIAS QUE ESTAN MATRICULADOS.**

SELECT

CONCAT (E. nombres, ' ', E. apellido\_paterno,' ', E. apellido\_materno) AS 'Nombre completo',

SUM (M. creditos) AS 'Total creditos'

FROM Estudiantes E

JOIN Matricula MA ON E. estudiante\_id = MA. estudiante\_id

JOIN PeriodosAcademicos PA ON MA. periodoacademico\_id = PA. periodoacademico\_id

JOIN Materias M ON PA. materia\_id = M. materia\_id

GROUP BY E. estudiante\_id, E. nombres, E. apellido\_paterno, E. apellido\_materno;

**-- 10° CONSULTA, ENCUENTRA AL PROFESOR CON MAS MATERIAS ASIGNADAS Y MUESTRA SU NOMBRE COMPLETO JUNTO CON EL NUMERO DE MATERIAS.**

SELECT TOP 1

CONCAT (P. nombre,' ',P.apellido\_paterno,' ',P.apellido\_materno) AS 'Profesor nombre completo',

COUNT (M. materia\_id) AS 'Total materias'

FROM Profesores P

INNER JOIN Materias M ON P. profesor\_id = M. profesor\_id

GROUP BY P. profesor\_id, P. nombre, P. apellido\_paterno, P. apellido\_materno

ORDER BY 'Total materias' DESC;

**-- 11° CONSULTA, LISTA EL NOMBRE DE LOS ESTUDIANTES JUNTO CON LAS MATERIAS QUE ESTAN CURSANDO ACTUALMENTE.**

SELECT

CONCAT (E. nombres, ' ', E. apellido\_paterno,' ', E. apellido\_materno) AS 'Nombre completo',

M.nombre AS 'Materia'

FROM Estudiantes E

INNER JOIN Matrícula MA ON E. estudiante\_id = MA. estudiante\_id

INNER JOIN PeriodosAcademicos PA ON MA. periodoacademico\_id = PA. periodoacademico\_id

INNER JOIN Materias M ON PA. materia\_id = M. materia\_id

WHERE GETDATE () BETWEEN PA. fecha\_inicio AND PA. fecha\_fin;