**Fase de análisis**

Registro Académico Facultad de Ingeniería USAC

Entidades Identificadas en el proyecto

1. Carrera
2. Plan
3. Estudiante
4. Curso
5. Catedrático
6. Salón
7. Sección
8. Horario
9. Inscripción
10. Pensum
11. Prerrequisito

Posibles Relaciones entre entidades:

* Una Carrera tiene muchos Planes.
* Un Plan pertenece a una Carrera.
* Un Estudiante se inscribe en uno o más Planes.
* Un Estudiante está matriculado en uno o más Cursos.
* Un Curso es enseñado por uno o más Catedráticos.
* Un Curso se lleva a cabo en una o más Secciones.
* Una Sección está asociada a un Curso y tiene asignado un Catedrático.
* Un Horario está asignado a una Sección en un Salón.
* Una Inscripción corresponde a un Estudiante en un Plan.
* Un Plan está compuesto por un Pensum.
* Un Curso puede tener varios Prerrequisitos

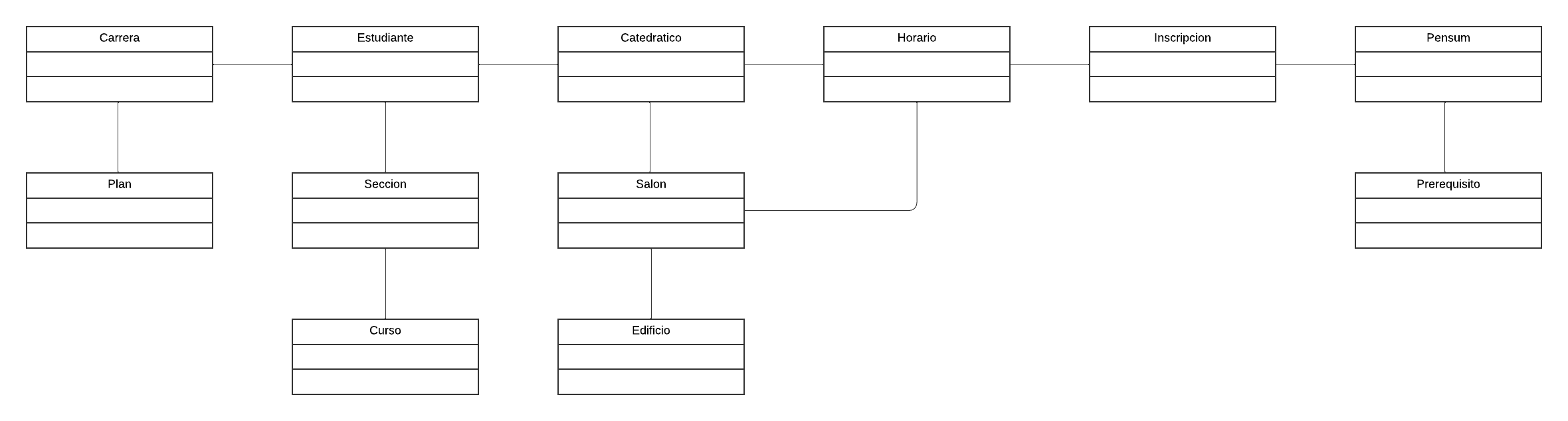
Entidades

1. **Carrera**
   * Atributos:
     + Código de Carrera (PK)
     + Nombre de Carrera
2. **Plan**
   * Atributos:
     + Código de Plan (PK)
     + Código de Carrera (FK)
     + Nombre de Plan
     + Año de Inicio
     + Ciclo de Inicio
     + Año de Fin
     + Ciclo de Fin
     + Créditos Necesarios
   * Relación: Una Carrera tiene muchos Planes.
3. **Estudiante**
   * Atributos:
     + Número de Carnet (PK)
     + Nombre Completo
     + Ingreso Familiar
     + Fecha de Nacimiento
4. **Curso**
   * Atributos:
     + Código de Curso (PK)
     + Nombre de Curso
5. **Catedrático**
   * Atributos:
     + Código de Catedrático (PK)
     + Nombre Completo
     + Sueldo Mensual
6. **Salón**
   * Atributos:
     + Código de Salón (PK)
     + Código de Edificio (FK)
     + Capacidad
7. **Sección**
   * Atributos:
     + Código de Sección (PK)
     + Código de Curso (FK)
     + Código de Catedrático (FK)
     + Año
     + Ciclo
   * Relaciones: Una Sección está asociada a un Curso y tiene asignado un Catedrático.
8. **Horario**
   * Atributos:
     + Código de Horario (PK)
     + Código de Sección (FK)
     + Día
     + Hora de Inicio
     + Hora de Fin
9. **Inscripción**
   * Atributos:
     + Código de Carrera (PK, FK)
     + Número de Carnet (PK, FK)
     + Fecha de Inscripción
   * Relaciones: Un Estudiante se inscribe en uno o más Planes.
10. **Pensum**
    * Atributos:
      + Código de Pensum (PK)
      + Código de Carrera (FK)
      + Código de Plan (FK)
      + Código de Curso (FK)
      + Obligatoriedad
      + Créditos Obtenidos
      + Nota de Aprobación
      + Zona Mínima
      + Código de Curso Prerrequisito
    * Relaciones: Un Plan está compuesto por un Pensum.
11. **Prerrequisito**
    * Atributos:
      + Código de Curso (PK, FK)
      + Código de Curso Prerrequisito (PK, FK)

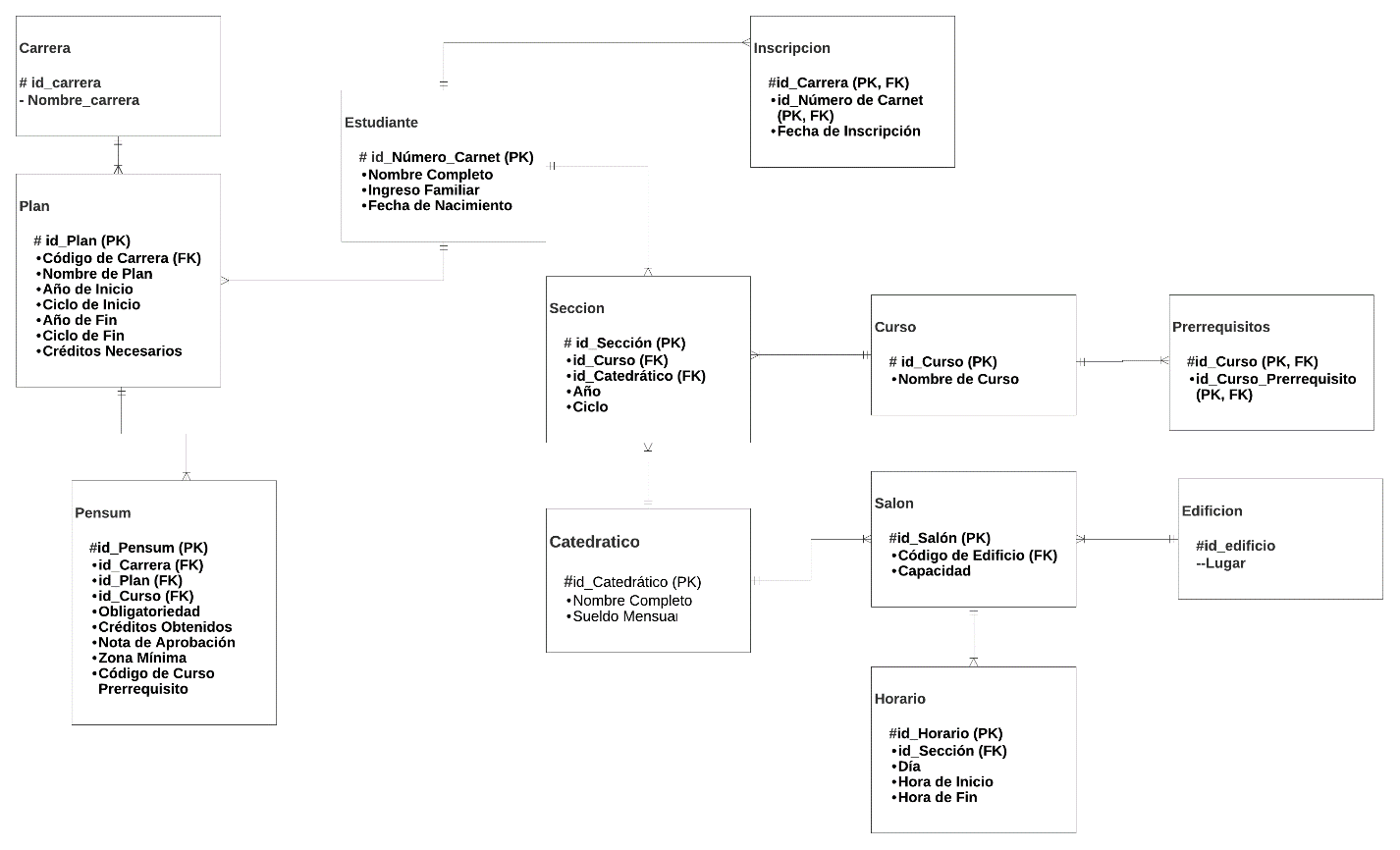
**Restricciones de Integridad:**

1. Para aprobar un curso es necesario tener una zona >= zona mínima y una nota >= nota de aprobación, que corresponda al plan (pensum) vigente en la asignación.
2. Los promedios se calculan solamente sobre notas aprobadas (dependiendo del plan o pensum en que se asignó el curso).
3. Para cerrar en una carrera es necesario aprobar todos los cursos obligatorios de la carrera antes de que finalice el período de vigencia del plan (pensum) en que se está cerrando. Es posible incluir cursos que se aprobaron en planes anteriores. Además, es necesario tener al menos la cantidad de créditos necesarios para cierre.
4. Para que un estudiante sea considerado el mejor estudiante de su promoción, debe tener el mejor promedio y no haber perdido ningún curso.

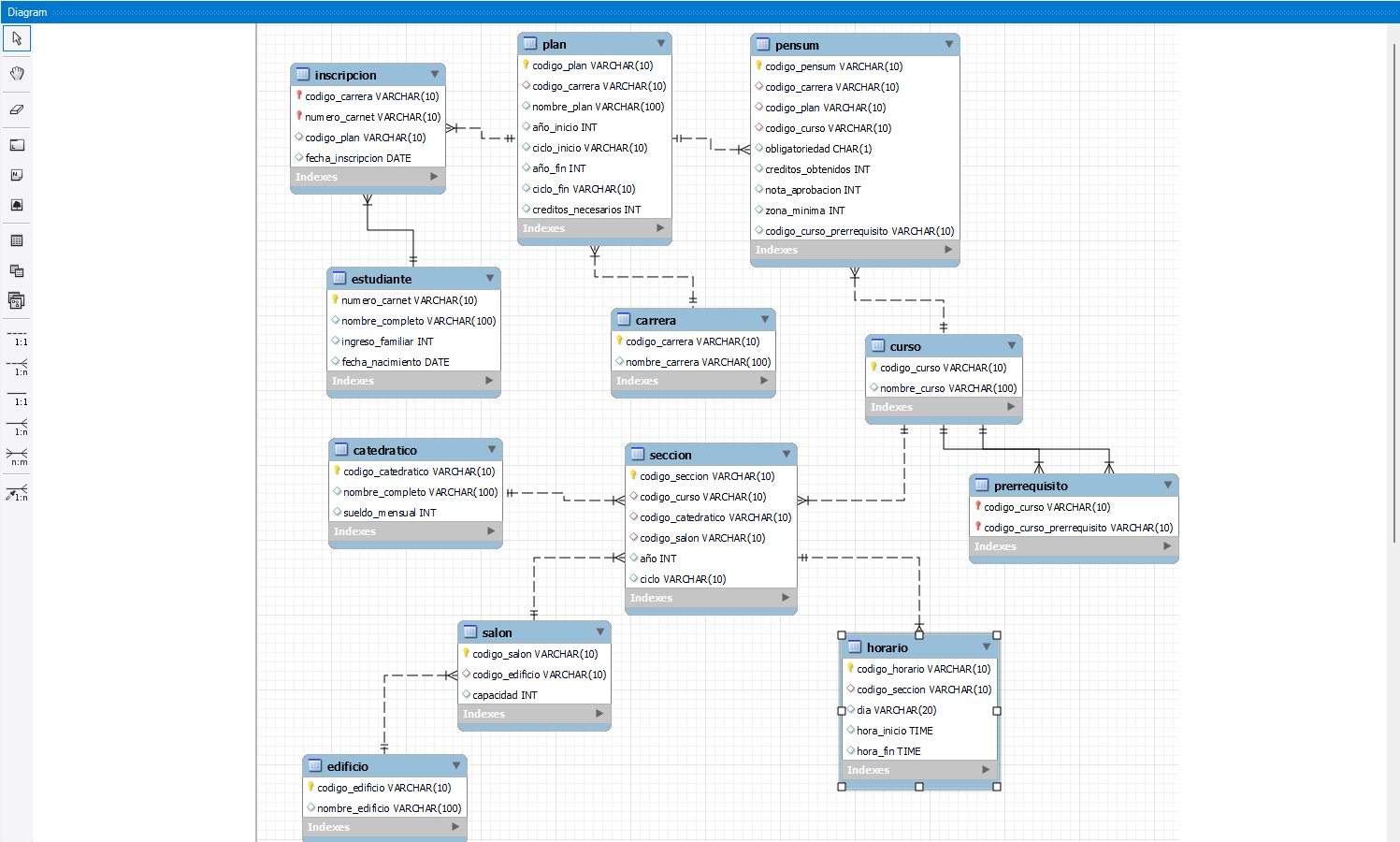
Modelo conceptual



Modelo lógico

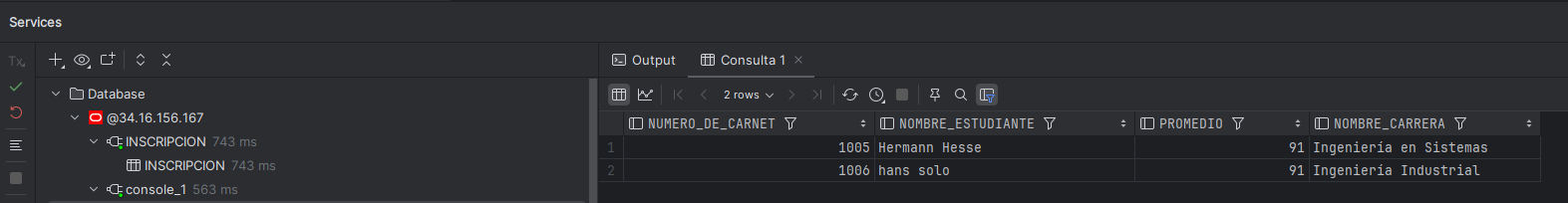


ER



Consultas en forma SQL y su resultado.

-- Consulta 1  
WITH estudiantes\_promedio AS (  
 SELECT  
 e.numero\_de\_carnet,  
 e.nombre AS nombre\_estudiante,  
 *AVG*(a.nota) AS promedio,  
 p.carrera\_codigo\_carrera  
 FROM  
 estudiante e  
 INNER JOIN  
 inscripcion i ON e.numero\_de\_carnet = i.estudiante\_numero\_de\_carnet  
 INNER JOIN  
 pensum pe ON i.plan\_codigo\_plan = pe.plan\_codigo\_plan  
 INNER JOIN  
 asignacion a ON i.estudiante\_numero\_de\_carnet = a.estudiante\_numero\_de\_carnet  
 INNER JOIN  
 seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion  
 INNER JOIN  
 curso c ON s.curso\_codigo\_curso = c.codigo\_curso  
 INNER JOIN  
 plan p ON pe.plan\_codigo\_plan = p.codigo\_plan  
 WHERE  
 *TO\_CHAR*(i.fecha\_inscripcion, 'YYYY') < '2022'  
 GROUP BY  
 e.numero\_de\_carnet,  
 e.nombre,  
 p.carrera\_codigo\_carrera  
),  
mejor\_promedio\_por\_carrera AS (  
 SELECT  
 p.carrera\_codigo\_carrera,  
 *MAX*(ep.promedio) AS mejor\_promedio  
 FROM  
 estudiantes\_promedio ep  
 INNER JOIN  
 plan p ON ep.carrera\_codigo\_carrera = p.codigo\_plan  
 GROUP BY  
 p.carrera\_codigo\_carrera  
)  
SELECT  
 ep.numero\_de\_carnet,  
 ep.nombre\_estudiante,  
 ep.promedio,  
 c.nombre AS nombre\_carrera  
FROM  
 estudiantes\_promedio ep  
INNER JOIN  
 mejor\_promedio\_por\_carrera mp ON ep.carrera\_codigo\_carrera = mp.carrera\_codigo\_carrera  
INNER JOIN  
 carrera c ON ep.carrera\_codigo\_carrera = c.codigo\_carrera  
WHERE  
 ep.promedio = mp.mejor\_promedio;



Consulta 2.  
-- Consulta 2

SELECT

    c.codigo\_curso,

    c.nombre AS nombre\_curso,

    carr.nombre AS nombre\_carrera,

    s.codigo\_seccion AS seccion,

    p.horario\_inicio || ' - ' || p.horario\_final AS horario,

    d.dia,

    sa.codigo\_salon AS salon,

    e.nombre\_edificio AS edificio,

    cat.nombre AS nombre\_catedratico

FROM

    curso c

INNER JOIN

    pensum pe ON c.codigo\_curso = pe.curso\_codigo\_curso

INNER JOIN

    plan pl ON pe.plan\_codigo\_plan = pl.codigo\_plan

INNER JOIN

    carrera carr ON pl.carrera\_codigo\_carrera = carr.codigo\_carrera

INNER JOIN

    seccion s ON c.codigo\_curso = s.curso\_codigo\_curso

INNER JOIN

    horario h ON s.codigo\_seccion = h.seccion\_codigo\_seccion

INNER JOIN

    dia d ON h.dia\_codigo\_dia = d.codigo\_dia

INNER JOIN

    periodo p ON h.periodo\_codigo\_periodo = p.codigo\_periodo

INNER JOIN

    salon sa ON h.salon\_codigo\_salon = sa.codigo\_salon

INNER JOIN

    edificio e ON sa.edificio\_codigo\_edificio = e.codigo\_edificio

INNER JOIN

    catedratico cat ON s.catedratico\_codigo\_catedratico = cat.codigo\_catedratico

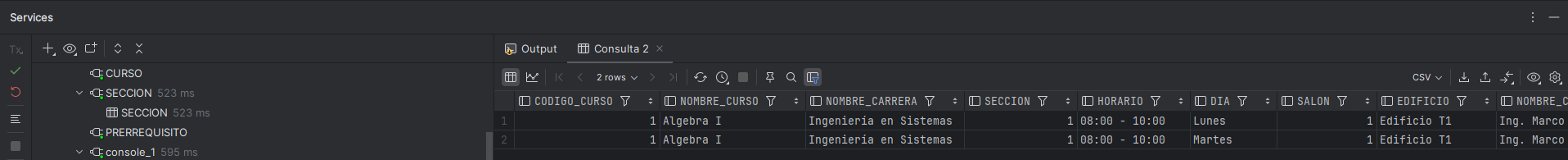
WHERE

    s.ciclo = '1' -- Reemplazar con el ciclo específico

    AND EXTRACT(YEAR FROM s.año) = 2020 -- Extraer el año de la columna s.año y comparar

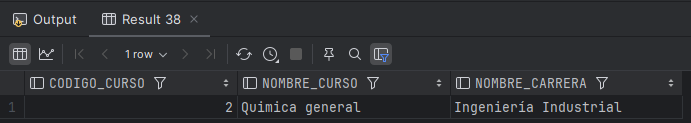
ORDER BY

    p.horario\_inicio,d.dia;

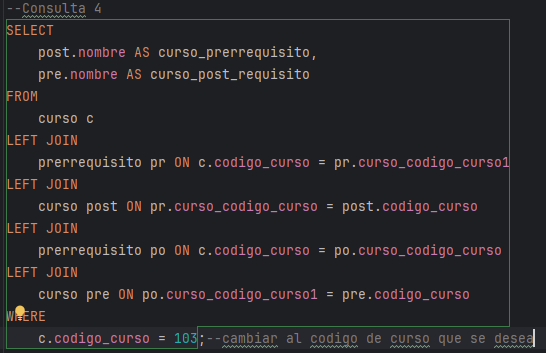


Consulta 3.

SELECT c.codigo\_curso, c.nombre AS nombre\_curso, ca.nombre AS nombre\_carrera  
FROM curso c  
JOIN pensum p ON c.codigo\_curso = p.curso\_codigo\_curso  
JOIN carrera ca ON p.plan\_codigo\_plan = ca.codigo\_carrera  
WHERE ca.nombre='Ingeniería Industrial';



Consulta 4.



Salida

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Consulta 5

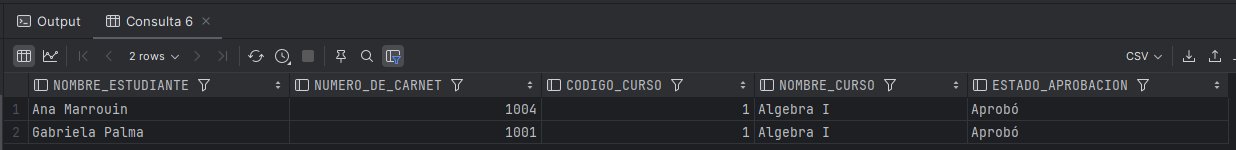
SELECT  
 c.nombre AS nombre\_curso,  
 s.año,  
 s.ciclo,  
 ct.nombre AS nombre\_catedratico  
FROM  
 curso c  
JOIN  
 seccion s ON c.codigo\_curso = s.curso\_codigo\_curso  
JOIN  
 catedratico ct ON s.catedratico\_codigo\_catedratico = ct.codigo\_catedratico  
WHERE  
 ct.nombre = 'Ing. Marco Pérez';

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Consulta 6

--Consulta 6  
SELECT  
 e.nombre AS nombre\_estudiante,  
 e.numero\_de\_carnet,  
 c.codigo\_curso,  
 c.nombre AS nombre\_curso,  
 CASE  
 WHEN a.nota >= 61 THEN 'Aprobó'  
 ELSE 'No Aprobó'  
 END AS estado\_aprobacion  
FROM  
 asignacion a  
JOIN  
 estudiante e ON a.estudiante\_numero\_de\_carnet = e.numero\_de\_carnet  
JOIN  
 seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion  
JOIN  
 curso c ON s.curso\_codigo\_curso = c.codigo\_curso  
JOIN  
 pensum p ON c.codigo\_curso = p.curso\_codigo\_curso  
WHERE  
 c.codigo\_curso = 1; -- Reemplaza # con el código del curso deseado



Consulta 8

SELECT

    e.nombre AS nombre\_estudiante,

    c.nombre AS nombre\_carrera,

    AVG(a.nota) AS promedio,

    SUM(p.creditos\_obtenidos) AS creditos\_ganados

FROM

    estudiante e

JOIN

    asignacion a ON e.numero\_de\_carnet = a.estudiante\_numero\_de\_carnet

JOIN

    seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion

JOIN

    pensum p ON s.curso\_codigo\_curso = p.curso\_codigo\_curso

JOIN

    plan pl ON p.plan\_codigo\_plan = pl.codigo\_plan

JOIN

    carrera c ON pl.carrera\_codigo\_carrera = c.codigo\_carrera

GROUP BY

    e.nombre,

    c.nombre,

    pl.creditos\_necesarios

HAVING

    SUM(p.creditos\_obtenidos) >= pl.creditos\_necesarios;

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Consulta 9

WITH catedraticos\_sistemas AS (

    SELECT DISTINCT

        s.catedratico\_codigo\_catedratico

    FROM

        seccion s

    JOIN

        curso c ON s.curso\_codigo\_curso = c.codigo\_curso

    JOIN

        pensum p ON c.codigo\_curso = p.curso\_codigo\_curso

    JOIN

        plan pl ON p.plan\_codigo\_plan = pl.codigo\_plan

    JOIN

        carrera ca ON pl.carrera\_codigo\_carrera = ca.codigo\_carrera

    WHERE

        ca.nombre = 'Ingeniería en Sistemas'

        AND s.año BETWEEN DATE '2021-01-01' AND DATE '2021-06-30'

)

SELECT DISTINCT

    e.nombre AS nombre\_estudiante

FROM

    estudiante e

JOIN

    asignacion a ON e.numero\_de\_carnet = a.estudiante\_numero\_de\_carnet

JOIN

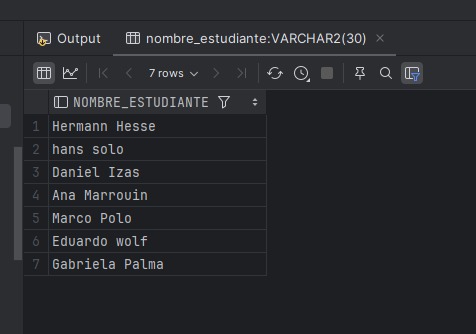
    seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion

JOIN

    catedraticos\_sistemas cs ON s.catedratico\_codigo\_catedratico = cs.catedratico\_codigo\_catedratico

WHERE

    a.nota >= 61;



Consulta 10

-- Paso 1: Identificar los cursos que el estudiante específico ha completado

WITH cursos\_del\_estudiante AS (

    SELECT

        a.seccion\_codigo\_seccion,

        s.curso\_codigo\_curso

    FROM

        asignacion a

    JOIN

        seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion

    WHERE

        a.estudiante\_numero\_de\_carnet = 1001

        AND a.nota >= 61  -- Suponiendo que 60 es la nota mínima para aprobar el curso

),

-- Paso 2: Obtener la lista de estudiantes que hayan completado cada uno de los cursos del estudiante específico

estudiantes\_que\_completaron\_cursos AS (

    SELECT

        a.estudiante\_numero\_de\_carnet,

        a.seccion\_codigo\_seccion,

        s.curso\_codigo\_curso

    FROM

        asignacion a

    JOIN

        seccion s ON a.seccion\_codigo\_seccion = s.codigo\_seccion

    JOIN

        cursos\_del\_estudiante cde ON s.curso\_codigo\_curso = cde.curso\_codigo\_curso

    WHERE

        a.nota >= 61

)

-- Paso 3: Contar cuántos cursos en común tienen los estudiantes con el estudiante específico

SELECT

    e.nombre AS nombre\_estudiante

FROM

    estudiante e

JOIN

    (SELECT

        e2.estudiante\_numero\_de\_carnet,

        COUNT(e2.curso\_codigo\_curso) AS cursos\_en\_comun

     FROM

        estudiantes\_que\_completaron\_cursos e2

     GROUP BY

        e2.estudiante\_numero\_de\_carnet

     HAVING

        COUNT(e2.curso\_codigo\_curso) = (SELECT COUNT(\*) FROM cursos\_del\_estudiante)

    ) ec ON e.numero\_de\_carnet = ec.estudiante\_numero\_de\_carnet

WHERE

    e.numero\_de\_carnet <> 1001;  -- Excluir al estudiante específico

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente