

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.

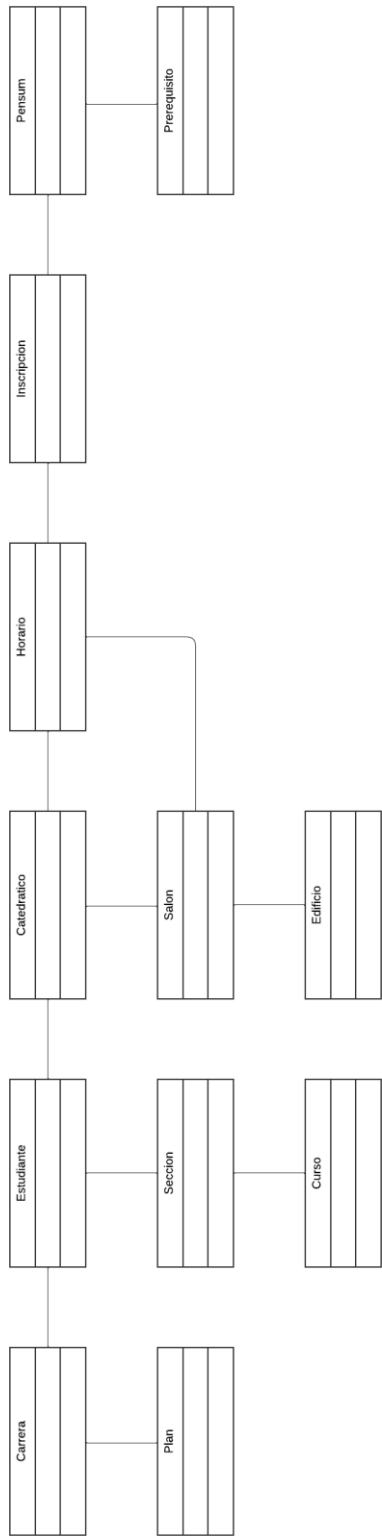
11. Prerrequisito

- Atributos:
 - Código de Curso (PK, FK)
 - Código de Curso Prerrequisito (PK, FK)

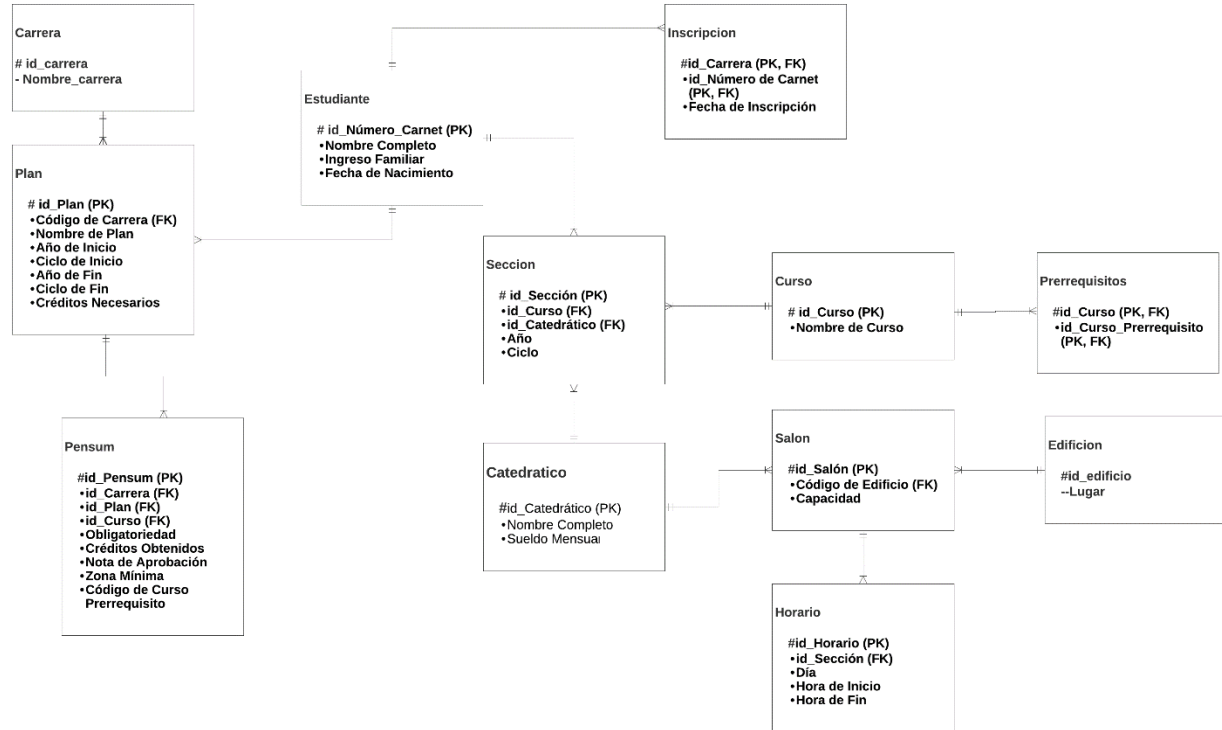
Restricciones de Integridad:

1. Para aprobar un curso es necesario tener una zona \geq zona mínima y una nota \geq 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.

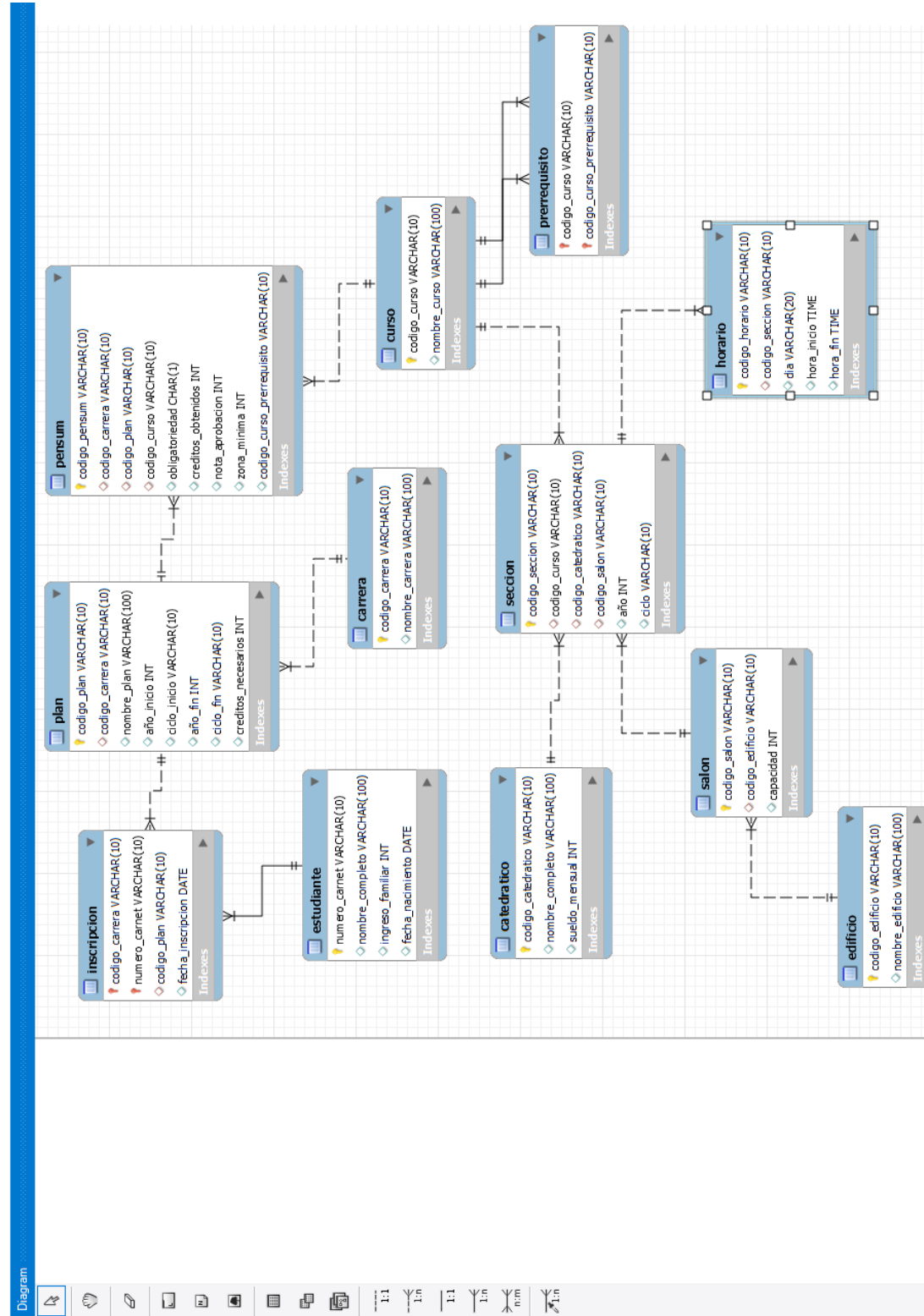
Modelo conceptual



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;
```

Output Consulta 1 ×

2 rows

	NUMERO_DE_CARNET	NOMBRE_ESTUDIANTE	PROMEDIO	NOMBRE_CARRERA
1	1005	Hermann Hesse	91	Ingenieria en Sistemas
2	1006	hans solo	91	Ingenieria Industrial

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;

Output

Consulta 2

	CODIGO_CURSO	NOMBRE_CURSO	NOMBRE_CARRERA	SECCION	HORARIO	DIA	SALON	EDIFICIO	NOMBRE_C
1		1 Algebra I	Ingenieria en Sistemas		1 08:00 - 10:00	Lunes		1 Edificio T1	Ing. Marco
2		1 Algebra I	Ingenieria en Sistemas		1 08:00 - 10:00	Martes		1 Edificio T1	Ing. Marco



Consulta 3.

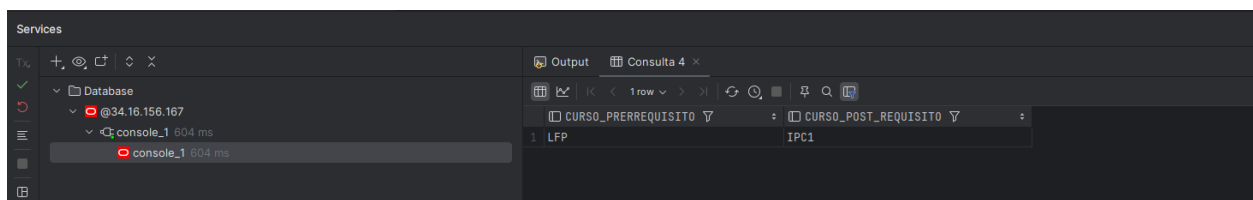
```
SELECT c.codigo_curso, c.nombre AS nombre_curso, ca.nombre AS nombre_carrera
FROM curso c
JOIN pensum p ON c.codigo_curso = p.curso_codigo_curso
JOIN carrera ca ON p.plan_codigo_plan = ca.codigo_carrera
WHERE ca.nombre='Ingeniería Industrial';
```

Output			
Result 38			
1 row			
	CODIGO_CURSO	NOMBRE_CURSO	NOMBRE_CARRERA
1	2	Quimica general	Ingeniería Industrial

Consulta 4.

```
--Consulta 4
SELECT
    post.nombre AS curso_prerrequisito,
    pre.nombre AS curso_post_requisito
FROM
    curso c
LEFT JOIN
    prerrequisito pr ON c.codigo_curso = pr.curso_codigo_curso1
LEFT JOIN
    curso post ON pr.curso_codigo_curso = post.codigo_curso
LEFT JOIN
    prerrequisito po ON c.codigo_curso = po.curso_codigo_curso
LEFT JOIN
    curso pre ON po.curso_codigo_curso1 = pre.codigo_curso
WHERE
    c.codigo_curso = 103; --cambiar al codigo de curso que se desea
```

Salida



CURSO_PRERREQUISITO	CURSO_POST_REQUISITO
LFP	IPC1

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';
```

Output Result 40 x

1 row

	NOMBRE_CURSO	AÑO	CICLO	NOMBRE_CATEDRATICO
1	Algebra I	2020-01-01	Primer semestre	Ing. Marco Pérez

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
```

Output Consulta 6 x

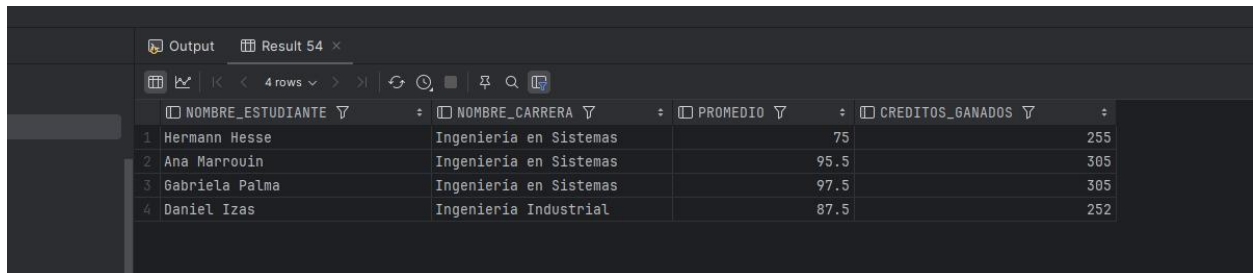
2 rows

CSV

	NOMBRE_ESTUDIANTE	NUMERO_DE_CARNET	CODIGO_CURSO	NOMBRE_CURSO	ESTADO_APROBACION
1	Ana Marrouin	1004	1	Algebra I	Aprobó
2	Gabriela Palma	1001	1	Algebra I	Aprobó

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;
```



	NOMBRE_ESTUDIANTE	NOMBRE_CARRERA	PROMEDIO	CREDITOS_GANADOS
1	Hermann Hesse	Ingenieria en Sistemas	75	255
2	Ana Marrovin	Ingenieria en Sistemas	95.5	305
3	Gabriela Palma	Ingenieria en Sistemas	97.5	305
4	Daniel Izas	Ingenieria Industrial	87.5	252

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;
```

Output nombre_estudiante:VARCHAR2(30) x

7 rows

	NOMBRE_ESTUDIANTE
1	Hermann Hesse
2	hans solo
3	Daniel Izas
4	Ana Marrouin
5	Marco Polo
6	Eduardo wolf
7	Gabriela Palma

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

```

Output		Paso 1: Identificar ...ecífico ha completado	
NOMBRE_ESTUDIANTE			
1	Ana Marrovin		