

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

SISTEMAS DE BASES DE DATOS 1

CATEDRÁTICO: ING. LUIS ESPINO

TUTOR ACADÉMICO: PAULO ARGUETA



ENNER ESAÍ MENDIZABAL CASTRO

CARNÉ: 202302220

SECCIÓN: B

GUATEMALA, 19 DE SEPTIEBRE DEL 2,025

# ÍNDICE

ÍNDICE .....	1
DOCUMENTACIÓN .....	3
ANÁLISIS DE LAS ENTIDADES.....	3
EXPLICACIÓN DE LAS REGLAS DE NEGOCIO.....	4
Regla de Negocio 1: Aprobación de un Curso .....	4
Regla de Negocio 2: Cálculo de Promedios.....	4
Regla de Negocio 3: Cierre de Carrera .....	5
Regla de Negocio 4: El Mejor Estudiante da la Promoción .....	5
DESCRIPCIÓN DE CADA ENTIDAD.....	6
ESTUDIANTE.....	6
CARRERA .....	6
CURSO.....	7
CATEDRÁTICO .....	7
PLAN.....	8
SECCIÓN .....	8
PENSUM.....	9
INSCRIPCION .....	9
ASIGNACION.....	10
DIA.....	10
HORARIO .....	11
PERIODO.....	11
SALON.....	12

PRERREQUISITO.....	12
DESCRIPCIÓN DE RELACIONES ENTRE ENTIDADES.....	14
ESTUDIANTE ↔ CARRERA (MUCHOS A MUCHOS).....	14
CARRERA → PLAN (UNO A MUCHOS).....	14
PLAN ↔ CURSO (MUCHOS A MUCHOS) .....	14
CURSO → SECCION (UNO A MUCHOS).....	15
CATEDRÁTICO → SECCION (UNO A MUCHOS) .....	15
ESTUDIANTE ↔ SECCION (MUCHOS A MUCHOS) .....	15
SECCION → HORARIO (UNO A MUCHOS) .....	16
CURSO ↔ CURSO (MUCHOS A MUCHOS, REFLEXIVA).....	16
CONSULTAS EN FORMA SQL CON SUS RESPECTIVOS RESULTADOS .....	17
CONSULTA 1 .....	17
CONSULTA 2 .....	18
CONSULTA 3 .....	19
CONSULTA 4 .....	20

# DOCUMENTACIÓN

## ANÁLISIS DE LAS ENTIDADES

La Directora también le indica que los docentes de la facultad, pueden ser catedráticos de uno o más cursos, en uno o más días, en uno o más periodos de clase y en uno o más salones. Los salones pueden ser utilizados por uno o más catedráticos y que los salones se encuentran en diferentes edificios. Además, le indica que las **carreras** se identifican con un **código** y con un **nombre**, que los **docentes** se identifican con un **código de docente**, **nombre completo** y **sueldo Mensual**. Los **salones** se identifican con un **código de edificio** y **código de salón** y la **capacidad del salón**.

En un mismo horario se pueden dar uno o más cursos por uno o más catedráticos. En un salón los docentes pueden dar sus clases en diferentes horarios. Y le indica también que los **cursos** se identifican con un **código de curso** y un **nombre**. Y que un **periodo** en que se imparten los cursos se identifica por un **código de periodo**, una **hora de inicio** y una **hora de fin** y que también se debe tomar el día en que se imparte el curso y que el **día** se identifica con un **número de día** y un **nombre**.

Para que un alumno pueda asignarse los cursos debe estar inscrito previamente en una carrera y en un plan (matutino, vespertino, nocturno o mixto). También le indica que para llevar el registro de los **alumnos** se identifican con un **número de carnet**, **nombre completo**, **ingreso familiar** y **fecha de nacimiento**. Que las **inscripciones** se identifican con el **código de carrera** y el **número de carnet del estudiante** y la **fecha de inscripción**. Luego le indica que un **plan** se identifica con el **código de la carrera** y el **código del plan**, el **nombre del plan**, el **año que inicia el plan**, el **ciclo con que inicia el plan**, el **año en que finaliza el plan**, el **ciclo en que termina el plan** y los **números de créditos** necesarios para cerrar la carrera en ese plan.

Cada carrera tiene un pensum de estudios que contiene información sobre la carrera que está estudiando como por ejemplo: cursos obligatorios, optativos, número de créditos, nota de aprobación, zona mínima, cursos prerrequisito y que para el control de los **pensum por estudiante** se lleva el control de la **carrera**, el **plan** y el **código de curso**, la **obligatoriedad del curso**, los **números de créditos** que un estudiante obtiene al ganar el curso, la **nota de aprobación del curso**, la **zona mínima** que debe obtener en el curso para tener derecho a examen final y el **código de curso prerrequisito**.

Los cursos **prerrequisitos** deben pertenecer a una **carrera**, un **plan**, tener un **código** y su **código de prerrequisito** correspondiente. Que una **sección** debe tener su **código**, pertenecer a un **año** y un **ciclo específico** y debe estar asignada a un **catedrático**.

Luego le indica que la **asignación** de los cursos por los estudiantes, debe identificarse por el **número de carnet**, el **código del curso**, la **sección** a la que el estudiante se asignó, el **año**, el **ciclo**, la **zona** y la **nota del curso**.

Y finalmente le indica que cada curso se debe asignar en un **horario** específico y debe contener el **código del curso**, la **sección**, el **año**, el **ciclo**, el **periodo**, el **día**, el **edificio** y el **salón**. Y que los edificios pueden tener uno o más salones con diferentes capacidades para poder recibir a los alumnos.

- La entidades está resaltadas de **rojo**.
- Los atributos está resaltados de **verde**.
- Las llaves foráneas están resaltadas de **azul**.
- Las entidades con sus atributos están unidas por **amarillo**.
- Las entidades tipo tabla intermedia están unidas con color **rosado**.

## EXPLICACIÓN DE LAS REGLAS DE NEGOCIO

### Regla de Negocio 1: Aprobación de un Curso

Esta es la regla más importante debido a que es la que hizo que se complicaran las consultas 1, 2, y 3. Estas consultas que se mencionan, tienen que encargarse de validar la nota y la zona mínima dentro del:

*WHEN a.nota\_final >= p.nota\_aprobacion AND a.zona >= p.zona\_minima*

Dentro del *CASE*.

Además, se tienen que verificar los prerrequisitos de créditos antes de tomar el curso actual mediante esta subconsulta:

```
AND ( -- Y tambien en donde la suma de créditos de los obligatorios sea igual
      SELECT COALESCE(SUM(p2.creditos), 0) FROM ASIGNACION a2 -- Tabla de asignaciones
      JOIN PENSUM p2 ON a2.SECCION_id_curso = p2.CURSO_id_curso -- en realaciono tablas
      -- Y la tabla de inscripciones
      JOIN INSCRIPCION i2 ON a2.ESTUDIANTE_carnet = i2.ESTUDIANTE_carnet AND p2.PLAN_CARRERA_id_carrera = i2.CARRERA_id_carrera
      WHERE a2.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet -- Y también en donde la fecha sea menor
      -- Y también en donde la fecha sea menor
      AND (a2.SECCION_anio < a.SECCION_anio OR (a2.SECCION_anio = a.SECCION_anio AND a2.SECCION_ciclo < a.SECCION_ciclo))
      AND a2.nota_final >= p2.nota_aprobacion -- Y también en donde la nota sea mayor o igual a la aprobatoria
    ) >= p.creditos_prerrequisito -- Y también en donde la suma de créditos de los obligatorios sea igual
```

Por último, se tienen que verificar que los que se cumplan con los prerrequisitos, para esto se usó una doble negación con un NOT EXISTS para asegurar que todos los cursos prerrequisitos estén aprobados con esta consulta;

```
AND NOT EXISTS ( -- Y también en donde no haya algún prerrequisito
      SELECT 1 FROM PRERREQUISITO pr -- Tabla de prerrequisitos
      WHERE pr.PENSUM_CURSO_id_curso = a.SECCION_id_curso -- Y también en donde el curso sea el mismo
      AND pr.PENSUM_PLAN_id_plan = p.PLAN_id_plan -- Y también en donde el plan sea el mismo
      AND pr.PENSUM_PLAN_CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera -- Y también en donde la carrera sea la misma
      AND NOT EXISTS ( -- Y también en donde no haya algún prerrequisito aqui
            SELECT 1 FROM ASIGNACION a3 -- Tabla de asignaciones
            JOIN PENSUM p3 ON a3.SECCION_id_curso = p3.CURSO_id_curso -- en realaciono tablas
            WHERE a3.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet -- Y también en donde el carnet del estudiante sea el mismo
            AND a3.SECCION_id_curso = pr.CURSO_id_curso -- Y también en donde el curso sea el mismo
            AND a3.nota_final >= p3.nota_aprobacion -- Y también en donde la nota sea mayor o igual a la aprobatoria
          )
    )
```

### Regla de Negocio 2: Cálculo de Promedios

Tanto para la consulta 1 como para la consulta 2 se tiene que cumplir esta regla de negocio en la que se tienen que realizar los cálculos del promedio después de tener filtrados los resultados de la subconsulta de aprobaciones.

Para esto hay una subconsulta que contiene la condición:

*WHERE ca.es\_aprobado = 1*

Y se calcula el promedio con esto:

$\text{AVG}(\text{ca.nota\_final}) \text{ AS promedio\_aprobadas}$

Como el promedio es calculador únicamente sobre los registros donde es aprobado, se asegura que solamente se incluyan las notas de los cursos aprobados.

### **Regla de Negocio 3: Cierre de Carrera**

Tanto para la consulta 1 como para la consulta 2 se tuvo que implementar esta lógica, por lo tanto, para esto, se compara la suma de créditos que el estudiante ha ganado con el total requerido en la tabla [PLAN](#) y se cuenta cuantos cursos obligatorios se han aprobado por el estudiante para compararlos con lo que solicita la tabla [PENSUM](#).

### **Regla de Negocio 4: El Mejor Estudiante da la Promoción**

Debido a que ninguna de las consultas realizadas está relacionada con obtener el mejor estudiante de la promoción, esta regla de negocio no fue aplicada en ninguna parte.

## DESCRIPCIÓN DE CADA ENTIDAD

El diseño de la base de datos se estructuró para reflejar fielmente las operaciones y la organización del sistema de control académico con el objetivo de almacenar la información de manera organizada y así garantizar la integridad de los datos. Para ello, se ha dividido la información en entidades especializadas, cada una responsable de un aspecto específico del sistema.

### ESTUDIANTE

Es la entidad principal del sistema, ya que almacena toda la información personal y de identificación de los alumnos.

#### PORQUÉ DE ESTA

Se creó porque es indispensable registrar a quién pertenecen las notas, las inscripciones y el progreso académico. Sin esta entidad, los datos del sistema serían anónimos.

#### ATRIBUTOS

- **carnet (Llave Primaria):** Identificador único, numérico e irrepetible, que corresponde al código oficial con el que la universidad identifica a cada estudiante.
- **nombre, fecha\_nacimiento:** Atributos descriptivos esenciales para la identificación y caracterización del estudiante.
- **ingreso\_familiar:** Atributo de tipo socioeconómico, útil para análisis estadísticos, reportes o la administración de programas de becas.

### CARRERA

Representa la oferta académica oficial de la facultad.

#### PORQUÉ DE ESTA

Actúa como una entidad catálogo para agrupar y clasificar planes de estudio, cursos y estudiantes, permitiendo estructurar la información académica de manera coherente.

#### ATRIBUTOS

- **id\_carrera (Llave Primaria):** Código numérico que identifica de forma única cada carrera, evitando ambigüedades que podrían surgir si se usara el nombre.
- **nombre:** Atributo descriptivo esencial.

### CURSO

Funciona como un catálogo maestro de todas las materias que se pueden impartir.

#### PORQUÉ DE ESTA

La creación de esta entidad evita la redundancia. Almacenar los cursos en su propia tabla previene la repetición de su nombre y código en múltiples registros, lo cual es un principio clave de la normalización.

#### ATRIBUTOS

- **id\_curso (Llave Primaria):** Identificador numérico único para cada materia.
- **nombre:** Nombre completo y oficial del curso.

### CATEDRÁTICO

Almacena la información de todos los docentes de la facultad.



## PORQUÉ DE ESTA

Es una entidad fundamental para la asignación académica, permitiendo vincular a un profesor con los cursos que imparte. Además, centraliza la gestión de datos administrativos, como su salario.

## ATRIBUTOS

- **id\_catedratico (Llave Primaria):** Código único para cada docente.
- **nombre, sueldo\_mensual:** Atributos para la identificación del catedrático y el registro de información administrativa.

## PLAN

Define la vigencia, estructura y las reglas de un pensum específico.

## PORQUÉ DE ESTA

Es una entidad necesaria porque una misma carrera puede tener diferentes mallas curriculares a lo largo del tiempo. Esta tabla permite gestionar esas distintas versiones.

## ATRIBUTOS

- **id\_plan, CARRERA\_id\_carrera (Llave Primaria Compuesta):** Un plan de estudios solo existe en el contexto de una carrera, por lo que la combinación de ambos campos lo identifica de forma única.
- **anio\_inicial, ciclo\_inicial, anio\_final, ciclo\_final:** Definen el periodo de validez del plan.
- **creditos\_cierre:** Atributo clave que define una de las metas para completar la carrera.

## SECCIÓN

Representa la instancia específica de un curso que se imparte en un semestre o ciclo determinado.

## PORQUÉ DE ESTA

Es la entidad operativa que une un curso, un catedrático, un año y un ciclo. Sin esta tabla, no se podría saber qué profesor impartió qué materia, en qué sección y cuándo.

## ATRIBUTOS

- **anio, id\_seccion, ciclo, CURSO\_id\_curso (Llave Primaria Compuesta):** Esta combinación es única e irrepetible. Solo puede existir una sección 'A' del curso 'Bases de Datos 1' en el 'CICLO1' del año 2025.
- **CATEDRATICO\_id\_catedratico (Llave Foránea):** Vincula la sección con el docente responsable de impartirla.

## PENSUM

Es una tabla asociativa que define qué cursos pertenecen a un determinado plan de estudios.

## PORQUÉ DE ESTA

Es crucial para estructurar el camino académico de cada carrera, resolviendo la relación muchos a muchos entre PLAN y CURSO.

## ATRIBUTOS

- **PLAN\_id\_plan, PLAN\_CARRERA\_id\_carrera, CURSO\_id\_curso (Llave Primaria Compuesta):** Define de forma inequívoca que un curso específico forma parte de un plan.
- **obligatorio, creditos, nota\_aprobacion:** Contiene las reglas de negocio que definen cómo se debe cursar y aprobar esa materia dentro de ese plan.

## INSCRIPCION

Almacena el acto transaccional de un estudiante inscribiéndose en una carrera en una fecha determinada.

#### PORQUÉ DE ESTA

Resuelve la relación de muchos a muchos entre ESTUDIANTE y CARRERA, permitiendo que un estudiante pueda estar inscrito en hasta dos carreras simultáneamente, como especifican los requerimientos.

#### ATRIBUTOS

- **ESTUDIANTE\_carnet, CARRERA\_id\_carrera (Llave Primaria Compuesta y Llaves Foráneas):** Su combinación une a un estudiante con una carrera de forma única.

### ASIGNACION

Registra el acto de un estudiante asignándose una SECCION de un curso y almacena su rendimiento académico.

#### PORQUÉ DE ESTA

Es una de las tablas transaccionales más importantes del sistema. Resuelve la relación muchos a muchos entre ESTUDIANTE y SECCION y es el lugar donde se guardan las notas.

#### ATRIBUTOS

- Su llave primaria es una llave foránea compuesta que apunta a SECCION y a ESTUDIANTE, vinculando a un alumno con la instancia de un curso.
- **zona, nota\_final:** Son los atributos variables que registran el rendimiento académico del estudiante.

### DIA

Tabla catálogo que almacena los días de la semana.

## PORQUÉ DE ESTA

Se crea para estandarizar los datos y evitar el uso de cadenas de texto variables. Al usar un `id_dia`, se garantiza la consistencia en la tabla HORARIO

## ATRIBUTOS

- **id\_dia (Llave Primaria):** Identificador numérico para cada día.
- **nombre:** Nombre del día.

## HORARIO

Es una tabla asociativa que detalla cuándo y dónde se imparte cada clase de una sección.

## PORQUÉ DE ESTA

Resuelve la compleja relación entre una SECCION y su horario semanal. Una sección puede tener múltiples sesiones, y esta tabla registra cada una de ellas, vinculándola a un DIA, PERIODO y SALON.

## ATRIBUTOS

Su **llave primaria** es compuesta por todas sus columnas, las cuales a su vez son **llaves foráneas** que apuntan a **SECCION**, **DIA**, **PERIODO** y **SALON**. Esto garantiza que cada registro de horario sea una combinación única de estos cuatro elementos.

## PERIODO

Tabla catálogo que define los bloques horarios en los que se imparten las clases.

## PORQUÉ DE ESTA

Permite estandarizar los horarios de clase. En lugar de almacenar horas de inicio y fin variables en múltiples registros, se hace referencia a un `id_periodo` predefinido, lo cual simplifica la gestión de horarios.

## ATRIBUTOS

- **id\_periodeso (Llave Primaria):** Identificador numérico para cada bloque horario.
- **hora\_inicio, hora\_fin:** Definen el rango de tiempo del periodo.

## SALON

Almacena la información de las aulas y laboratorios donde se imparten las clases.

## PORQUÉ DE ESTA

Centraliza el inventario de espacios físicos. Es fundamental para la asignación de horarios y para evitar conflictos, como asignar dos secciones al mismo salón a la misma hora.

## ATRIBUTOS

- **id\_edificio, id\_salon (Llave Primaria Compuesta):** Un salón se identifica de forma única por su número y el edificio en el que se encuentra.
- **capacidad:** Atributo clave para la logística, permitiendo asignar secciones a aulas con suficiente espacio.

## PRERREQUISITO

Es una tabla asociativa que define las dependencias académicas entre los cursos de un mismo pensum.

## PORQUÉ DE ESTA

Es una entidad crucial para controlar el avance académico de los estudiantes. Establece las reglas que impiden que un alumno se asigne un curso avanzado sin haber aprobado antes las materias fundamentales requeridas, garantizando así el orden lógico del aprendizaje.

## ATRIBUTOS

- **PENSUM\_PLAN\_id\_plan,**  
**PENSUM\_PLAN\_CARRERA\_id\_carrera,**  
**PENSUM\_CURSO\_id\_curso** (Parte de la Llave Primaria y Llave Foránea a PENSUM): Identifica el curso principal que tiene el prerrequisito.
- **CURSO\_id\_curso** (Parte de la Llave Primaria y Llave Foránea a CURSO): Identifica el curso que debe ser aprobado primero (el prerrequisito).

## **DESCRIPCIÓN DE RELACIONES ENTRE ENTIDADES**

Las relaciones se establecen mediante llaves foráneas (FK), que garantizan que los datos estén conectados de forma lógica y consistente, manteniendo la integridad referencial.

### **ESTUDIANTE ↔ CARRERA (MUCHOS A MUCHOS)**

#### RAZÓN

Un estudiante puede estar en más de una carrera y una carrera tiene muchos estudiantes.

#### IMPLEMENTACIÓN

Se resuelve con la tabla intermedia INSCRIPCION. Un ESTUDIANTE puede tener varias filas en INSCRIPCION, y una CARRERA también.

### **CARRERA → PLAN (UNO A MUCHOS)**

#### RAZÓN

Una carrera puede tener varios planes de estudio a lo largo de su historia, pero un plan pertenece a una sola carrera.

#### IMPLEMENTACIÓN

La tabla PLAN tiene una llave foránea (CARRERA\_id\_carrera) que apunta a CARRERA.

### **PLAN ↔ CURSO (MUCHOS A MUCHOS)**

#### RAZÓN

Un plan de estudios consiste en muchos cursos, y un mismo curso puede ser parte de muchos planes de estudio diferentes.

#### IMPLEMENTACIÓN

Se resuelve con la tabla PENSUM, que especifica qué cursos componen cada plan y bajo qué reglas.

#### **CURSO → SECCION (UNO A MUCHOS)**

##### RAZÓN

Un curso puede abrirse en múltiples secciones cada semestre

#### IMPLEMENTACIÓN

SECCION tiene una llave foránea que apunta a CURSO.

#### **CATEDRÁTICO → SECCION (UNO A MUCHOS)**

##### RAZÓN

Un catedrático puede ser asignado para impartir muchas secciones a lo largo de su carrera docente.

#### IMPLEMENTACIÓN

SECCION tiene una llave foránea que apunta a CATEDRATICO.

#### **ESTUDIANTE ↔ SECCION (MUCHOS A MUCHOS)**

##### RAZÓN

Un estudiante se asigna muchas secciones, y una sección tiene muchos estudiantes.

#### IMPLEMENTACIÓN

Se resuelve con la tabla ASIGNACION, que es el registro formal de esta relación y donde se consignan las notas.



## **SECCION → HORARIO (UNO A MUCHOS)**

### RAZÓN

Una sección puede tener clase varios días a la semana.

### IMPLEMENTACIÓN

HORARIO tiene una llave foránea compuesta que apunta a SECCION para registrar cada uno de estos bloques.

## **CURSO ↔ CURSO (MUCHOS A MUCHOS, REFLEXIVA)**

### RAZÓN

Un curso puede tener varios cursos como prerrequisito para poder ser asignado. A su vez, ese mismo curso puede ser el prerrequisito para varios cursos futuros.

### IMPLEMENTACIÓN

Se resuelve con la tabla PRERREQUISITO. Esta tabla tiene una relación consigo misma a través de la tabla CURSO. Contiene dos llaves foráneas que apuntan a CURSO, una para identificar el curso dependiente y otra para identificar el curso prerrequisito.

# CONSULTAS EN FORMA SQL CON SUS RESPECTIVOS RESULTADOS

## CONSULTA 1

```

-- CONSULTA 1
-- ESTUDIANTES QUE HAN CERRADO SISTEMAS

SELECT
-- Tabla principal
e.nombre,
res.promedio_aprobadas,
res.creditos_ganados
FROM -- Lo necesario para obtener la tablita res
(
-- Subconsulta que resume los datos del estudiante
SELECT
ca.ESTUDIANTE_carnet,
SUM(ca.creditos) AS creditos_ganados,
AVG(ca.nota_final) AS promedio_aprobadas,
COUNT(DISTINCT CASE WHEN p.obligatorio = 1 THEN ca.SECCION_id_curso END) AS obligatorios_aprobados
FROM -- Lo necesario para obtener la tablita ca
(
-- SUBCONSULTA DE APROBACIONES
SELECT
a.ESTUDIANTE_carnet, a.SECCION_id_curso, a.SECCION_anio,
a.SECCION_ciclo, a.SECCION_id_seccion, a.nota_final, p.creditos,
i.CARRERA_id_carrera,
p.PLAN_id_plan, -- Se necesita para la subconsulta de prerequisitos
CASE
-- Selecciona solo los casos en los que la nota de aprobación y la zona cumplen
WHEN a.nota_final >= p.nota_aprobacion AND a.zona >= p.zona_minima
AND ( -- Y también en donde la suma de créditos de los obligatorios sea igual
-- Y también en donde
SELECT COALESCE(SUM(p2.creditos), 0) FROM ASIGNACION a2
JOIN PENSUM p2 ON a2.SECCION_id_curso = p2.CURSO_id_curso -- en relaciono tablas
JOIN INSCRIPCION i2 ON a2.ESTUDIANTE_carnet = i2.ESTUDIANTE_carnet AND p2.PLAN_CARRERA_id_carrera = i2.CARRERA_id_carrera
-- En donde el carnet del estudiante sea el mismo que el de la suma de todos su créditos.
WHERE a2.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet
AND (a2.SECCION_anio < a.SECCION_anio OR (a2.SECCION_anio = a.SECCION_anio AND a2.SECCION_ciclo < a.SECCION_ciclo))
AND a2.nota_final >= p2.nota_aprobacion
) >= p.creditos_prerequisito
AND NOT EXISTS ( -- Y también en donde no haya algún prerequisito
SELECT 1 FROM PREREQUISITO pr
WHERE pr.PENSUM_CURSO_id_curso = a.SECCION_id_curso
AND pr.PENSUM_PLAN_id_plan = p.PLAN_id_plan
AND pr.PENSUM_CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera
AND NOT EXISTS (
SELECT 1 FROM ASIGNACION a3
JOIN PENSUM p3 ON a3.SECCION_id_curso = p3.CURSO_id_curso
WHERE a3.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet
AND a3.SECCION_id_curso = pr.CURSO_id_curso
AND a3.nota_final >= p3.nota_aprobacion
)
)
THEN 1 ELSE 0
END AS es_aprobado
FROM ASIGNACION a
JOIN INSCRIPCION i ON a.ESTUDIANTE_carnet = i.ESTUDIANTE_carnet
JOIN PENSUM p ON a.SECCION_id_curso = p.CURSO_id_curso AND i.CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera
) ca
-- El JOIN y el GROUP BY se aplican sobre 'ca'
JOIN PENSUM p ON ca.SECCION_id_curso = p.CURSO_id_curso AND p.PLAN_CARRERA_id_carrera = 9
WHERE ca.es_aprobado = 1 AND ca.CARRERA_id_carrera = 9
GROUP BY ca.ESTUDIANTE_carnet
) res
JOIN ESTUDIANTE e ON res.ESTUDIANTE_carnet = e.carnet -- El estudiante con el carnet de la tabla previamente creada
JOIN PLAN pi ON p1.CARRERA_id_carrera = 9 -- Carrea con el ID = 9 (Sistemas)
WHERE
res.creditos_ganados >= p1.creditos_cierre -- En donde se tengan los créditos para el cierre y se tengan los cursos obligatorios aprobados
AND res.obligatorios_aprobados = (SELECT COUNT(*) FROM PENSUM WHERE PLAN_CARRERA_id_carrera = 9 AND obligatorio = 1);

```

## RESULTADO

Salida de Script x		
Tarea terminada en 0.077 segundos		
NOMBRE	FROMEDIO_APROBADAS	CREDITOS_GANADOS
ESTUDIANTE 1001	83.22	250

## (18)

```
-- CONSULTA 2
-- ESTUDIANTES QUE HAN CERRADO CUALQUIER CARRERA
-----
SELECT -- Selecciona los datos necesarios
    e.nombre AS nombre_estudiante,
    c.nombre AS nombre_carrera,
    rpc.promedio_aprobadas,
    rpc.creditos_ganados
FROM
(
    -- Subconsulta que resume datos por estudiante y carrera
    SELECT -- La tablita en si :)
        ca.ESTUDIANTE_carnet,
        ca.CARRERA_id_carrera,
        SUM(ca.creditos) AS creditos_ganados, -- Suma de créditos ganados
        AVG(ca.nota_final) AS promedio_aprobadas, -- Promedio de notas en cursos aprobados

        -- Conteo de cursos obligatorios aprobados
        COUNT(DISTINCT CASE WHEN p.obligatorio = 1 THEN ca.SECCION_id_curso END) AS obligatorios_aprobados
    FROM
    (
        -- SUBCONSULTA DE APROBACIONES COMPLETA
        SELECT
            a.ESTUDIANTE_carnet, a.SECCION_id_curso, a.SECCION_anio,
            a.SECCION_ciclo, a.SECCION_id_seccion, a.nota_final, p.creditos,
            i.CARRERA_id_carrera, p.PLAN_id_plan,
            CASE -- colocar cuando la nota sea mayor o igual a la aprobatoria asi como la minima
                -- Y tambien en donde la suma de créditos de los obligatorios sea igual
                WHEN a.nota_final >= p.nota_aprobacion AND a.zona >= p.zona_minima

            AND ( -- Y tambien en donde la suma de créditos de los obligatorios sea igual
                SELECT COALESCE(SUM(p2.creditos), 0) FROM ASIGNACION a2
                JOIN PENSUM p2 ON a2.SECCION_id_curso = p2.CURSO_id_curso
                JOIN INSCRIPCION i2 ON a2.ESTUDIANTE_carnet = i2.ESTUDIANTE_carnet AND p2.PLAN_CARRERA_id_carrera = i2.CARRERA_id_carrera
                WHERE a2.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet
                AND (a2.SECCION_anio < a.SECCION_anio OR (a2.SECCION_anio = a.SECCION_anio AND a2.SECCION_ciclo < a.SECCION_ciclo))
                AND a2.nota_final >= p2.nota_aprobacion
            ) >= p.creditos_prerequisito

            -- Y tambien en donde no haya algún prerrequisito
            AND NOT EXISTS (
                SELECT 1 FROM PRERREQUISITO pr
                WHERE pr.PENSUM_CURSO_id_curso = a.SECCION_id_curso
                AND pr.PENSUM_PLAN_id_plan = p.PLAN_id_plan -- Y tambien en donde el plan sea el mismo
                AND pr.PENSUM_PLAN_CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera -- Y tambien en donde la carrera sea la misma
                AND NOT EXISTS ( -- Y tambien en donde no haya algún prerrequisito aqui
                    SELECT 1 FROM ASIGNACION a3
                    JOIN PENSUM p3 ON a3.SECCION_id_curso = p3.CURSO_id_curso -- en relaciono tablas
                    WHERE a3.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet
                    AND a3.SECCION_id_curso = pr.CURSO_id_curso -- Y tambien en donde el curso sea el mismo
                    AND a3.nota_final >= p3.nota_aprobacion -- Y tambien en donde la nota sea mayor o igual a la aprobatoria
                )
            )
        THEN 1 ELSE 0
    END AS es_aprobado -- Y todo se guardaria como (Es aprobado)
    FROM ASIGNACION a
    JOIN INSCRIPCION i ON a.ESTUDIANTE_carnet = i.ESTUDIANTE_carnet -- Ahora los tipicos join

    -- Y tambien el join con pensum
    JOIN PENSUM p ON a.SECCION_id_curso = p.CURSO_id_curso AND i.CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera
) ca
JOIN PENSUM p ON ca.SECCION_id_curso = p.CURSO_id_curso AND p.PLAN_CARRERA_id_carrera = ca.CARRERA_id_carrera
WHERE ca.es_aprobado = 1 -- Y tambien en donde es aprobado
GROUP BY ca.ESTUDIANTE_carnet, ca.CARRERA_id_carrera -- Agrupando por carnet y carrera
) rpc
JOIN ESTUDIANTE e ON rpc.ESTUDIANTE_carnet = e.carnet -- Join con estudiante
JOIN CARRERA c ON rpc.CARRERA_id_carrera = c.id_carrera -- Join con carrera
JOIN PLAN pl ON rpc.CARRERA_id_carrera = pl.CARRERA_id_carrera -- Join con plan

-- Join con la subconsulta que cuenta los obligatorios por carrera
JOIN (SELECT PLAN_CARRERA_id_carrera, COUNT(*) AS total FROM PENSUM WHERE obligatorio = 1 GROUP BY PLAN_CARRERA_id_carrera) tob
ON rpc.CARRERA_id_carrera = tob.PLAN_CARRERA_id_carrera
WHERE -- Condiciones para saber si ha cerrado la carrera
    rpc.creditos_ganados >= pl.creditos_cierre
    AND rpc.obligatorios_aprobados = tob.total;
```

## RESULTADO

Salida de Script x	
Tarea terminada en 0.191 segundos	
HOMBRE_CARRERA	PROMEDIO_APROBADAS CREDITOS_GANADOS
INGENIERIA EN CIENCIAS Y SISTEMAS	83.22 250

## CONSULTA 3

```

CONSULTA 3
-- ESTUDIANTES QUE GANARON CURSOS CON CATEDRÁTICOS EN ESPECÍFICO

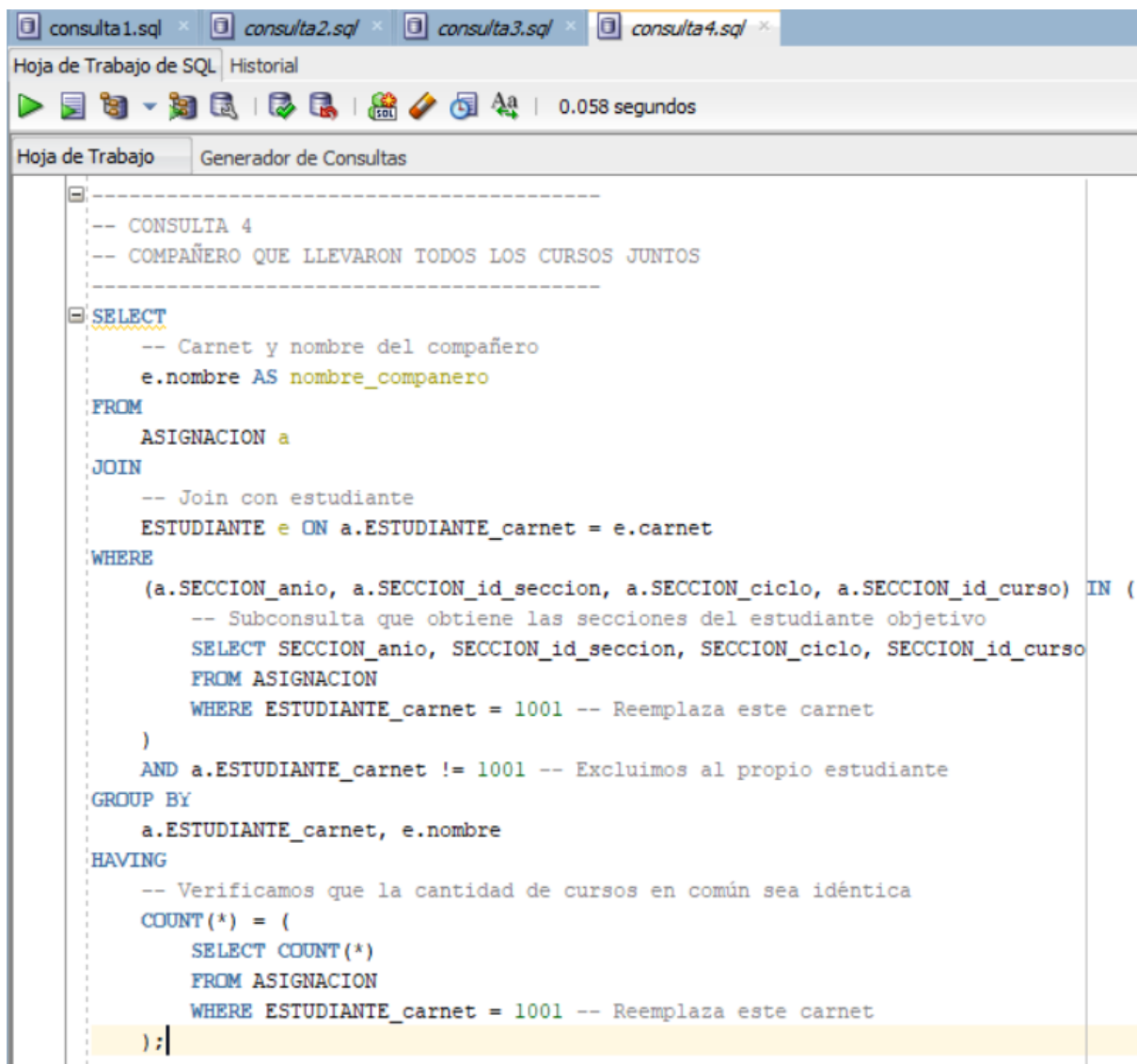
SELECT DISTINCT
e.nombre
FROM
(
-- SUBCONSULTA DE APROBACIONES COMPLETA
SELECT
-- La tablita en sí :)
a.ESTUDIANTE_carnet, a.SECCION_id_curso, a.SECCION_anio,
a.SECCION_ciclo, a.SECCION_id_seccion, a.nota_final, p.creditos,
i.CARRERA_id_carrera, p.PLAN_id_plan,
CASE -- colocar cuando la nota sea mayor o igual a la aprobatoria así como la mínima
WHEN a.nota_final >= p.nota_aprobacion AND a.zona >= p.zona_minima -- Y también en donde la suma de créditos de los obligatorios sea igual
AND ( -- Y también en donde la suma de créditos de los obligatorios sea igual
SELECT COALESCE(SUM(p2.creditos), 0) FROM ASIGNACION a2 -- Tabla de asignaciones
JOIN PENSUM p2 ON a2.SECCION_id_curso = p2.CURSO_id_curso -- en realaciono tablas
-- Y la tabla de inscripciones
JOIN INSCRIPCION i2 ON a2.ESTUDIANTE_carnet = i2.ESTUDIANTE_carnet AND p2.PLAN_CARRERA_id_carrera = i2.CARRERA_id_carrera
WHERE a2.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet -- Y también en donde la fecha sea menor
-- Y también en donde la fecha sea menor
AND (a2.SECCION_anio < a.SECCION_anio OR (a2.SECCION_anio = a.SECCION_anio AND a2.SECCION_ciclo < a.SECCION_ciclo))
AND a2.nota_final >= p2.nota_aprobacion -- Y también en donde la nota sea mayor o igual a la aprobatoria
) >= p.creditos_prerequisito -- Y también en donde la suma de créditos de los obligatorios sea igual
AND NOT EXISTS ( -- Y también en donde no haya algún prerequisito
SELECT 1 FROM PRERREQUISITO pr -- Tabla de prerequisitos
WHERE pr.PENSUM_CURSO_id_curso = a.SECCION_id_curso -- Y también en donde el curso sea el mismo
AND pr.PENSUM_PLAN_id_plan = p.PLAN_id_plan -- Y también en donde el plan sea el mismo
AND pr.PENSUM_PLAN_CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera -- Y también en donde la carrera sea la misma
AND NOT EXISTS ( -- Y también en donde no haya algún prerequisito aquí
SELECT 1 FROM ASIGNACION a3 -- Tabla de asignaciones
JOIN PENSUM p3 ON a3.SECCION_id_curso = p3.CURSO_id_curso -- en realaciono tablas
WHERE a3.ESTUDIANTE_carnet = a.ESTUDIANTE_carnet -- Y también en donde el carnet del estudiante sea el mismo
AND a3.SECCION_id_curso = pr.CURSO_id_curso -- Y también en donde el curso sea el mismo
AND a3.nota_final >= p3.nota_aprobacion -- Y también en donde la nota sea mayor o igual a la aprobatoria
)
)
THEN 1 ELSE 0 -- Y todo se guardaría como (Es aprobado)
END AS es_aprobado
FROM ASIGNACION a
JOIN INSCRIPCION i ON a.ESTUDIANTE_carnet = i.ESTUDIANTE_carnet -- Ahora los típicos join
-- Y también el join con pensum
JOIN PENSUM p ON a.SECCION_id_curso = p.CURSO_id_curso AND i.CARRERA_id_carrera = p.PLAN_CARRERA_id_carrera
) ca
JOIN ESTUDIANTE e ON ca.ESTUDIANTE_carnet = e.carnet -- Join con estudiante
JOIN SECCION s ON ca.SECCION_anio = s.anio -- Join con sección
AND ca.SECCION_id_seccion = s.id_seccion -- Y también en donde el id de sección sea el mismo
AND ca.SECCION_ciclo = s.ciclo -- Y también en donde el ciclo sea el mismo
AND ca.SECCION_id_curso = s.CURSO_id_curso -- Y también en donde el curso sea el mismo
WHERE -- Condiciones para saber si ganó el curso
ca.es_aprobado = 1 -- Y también en donde es aprobado
AND s.CATEDRATICO_id_catedrativo IN ( -- Aquí van los catedráticos específicos
SELECT DISTINCT s2.CATEDRATICO_id_catedrativo
FROM SECCION s2
JOIN PENSUM p2 ON s2.CURSO_id_curso = p2.CURSO_id_curso
WHERE p2.PLAN_CARRERA_id_carrera = 9 -- Sistemas
AND s2.anio = 2014 AND s2.ciclo = 'CICLO8' -- Asunción de "semestre pasado"
);

```

## RESULTADO

Salida de Script x	
Tarea terminada en 0.146 segundos	
NOMBRE	
ESTUDIANTE 1001	

## CONSULTA 4

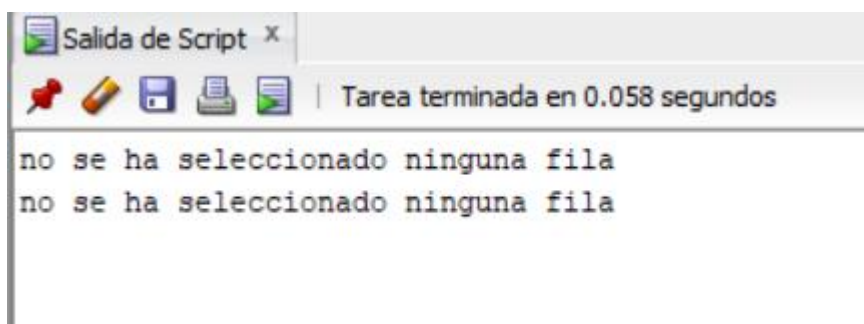


The screenshot shows the SQL Developer interface with a query editor. The query is for 'CONSULTA 4' and aims to find companions who took all the same courses as a specific student (carnet 1001). The query uses a complex nested subquery and a HAVING clause to compare the count of courses.

```
-- CONSULTA 4
-- COMPAÑERO QUE LLEVARON TODOS LOS CURSOS JUNTOS

SELECT
  -- Carnet y nombre del compañero
  e.nombre AS nombre_companero
FROM
  ASIGNACION a
JOIN
  -- Join con estudiante
  ESTUDIANTE e ON a.ESTUDIANTE_carnet = e.carnet
WHERE
  (a.SECCION_anio, a.SECCION_id_seccion, a.SECCION_ciclo, a.SECCION_id_curso) IN (
    -- Subconsulta que obtiene las secciones del estudiante objetivo
    SELECT SECCION_anio, SECCION_id_seccion, SECCION_ciclo, SECCION_id_curso
    FROM ASIGNACION
    WHERE ESTUDIANTE_carnet = 1001 -- Reemplaza este carnet
  )
  AND a.ESTUDIANTE_carnet != 1001 -- Excluimos al propio estudiante
GROUP BY
  a.ESTUDIANTE_carnet, e.nombre
HAVING
  -- Verificamos que la cantidad de cursos en común sea idéntica
  COUNT(*) = (
    SELECT COUNT(*)
    FROM ASIGNACION
    WHERE ESTUDIANTE_carnet = 1001 -- Reemplaza este carnet
  );
```

## RESULTADO



The screenshot shows the 'Salida de Script' (Script Output) window. It displays the message 'no se ha seleccionado ninguna fila' (no rows selected) twice, indicating that the query did not return any results.

```
no se ha seleccionado ninguna fila
no se ha seleccionado ninguna fila
```

Claramente no muestra nada porque en la base solo hay un estudiante asignado a los cursos (que es el mismo que ganó)