

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SISTEMAS DE BASES DE DATOS 1
ING. ÁLVARO GIOVANNI LONGO MORALES
TUTOR: JOSÉ VALERIO CHOC MIJANGOS

Proyecto 1

Grupo 9

Nombre	Carnet
Daniel Eduardo Mellado Ayala	201020232
Piter Angel Esaú Valiente de León	201902301
Eduardo Josué González Cifuentes	201900647

Fase de Análisis

1. Carrera

- Atributos: código de carrera (pk), nombre de carrera, código plan (fk).
- Relaciones:
 - Carrera-Plan: Un plan puede tener una o muchas carreras.
 - Carrera-Inscripción: Una carrera puede tener muchos inscritos.

2. Docente

- Atributos: código_docente (pk), nombre_completo, sueldo mensual.
- Relaciones:
 - Docente-Sección: Un docente puede tener una o muchas secciones.

3. Salón

- Atributos: edificio, código_salón (pk), capacidad, código_horario(fk).
- Relaciones:
 - Salón-Horario: Pueden haber varios salones en un horario.

4. Curso

- Atributos: código_curso (pk), nombre, código_pensum(fk), código_prerequisito(fk)).
- Relaciones:
 - Curso-Pensum: Un pensum puede tener uno o muchos cursos.
 - Curso-Prerrequisito: Un curso puede tener uno o muchos prerrequisitos.
 - Curso-Prerrequisito: Varios cursos pueden tener un mismo prerrequisito.

5. Periodo

- Atributos: código_periodo (pk), hora de inicio, hora de fin.
- Relaciones:
 - Periodo-Horario: Varios períodos pueden estar en un horario.

6. Estudiante

- Atributos: número de carnet (pk), nombre completo, ingreso familiar, fecha de nacimiento.
- Relaciones:
 - Estudiante-Inscripción: Un estudiante puede tener una o muchas inscripciones.
 - Estudiante-Asignación: Un estudiante puede tener una o muchas asignaciones.

7. Plan

- Atributos: código_plan (pk), carrera, nombre_plan, año_inicio, año_fin, ciclo_inicio, ciclo_fin, créditos_necesarios, código_pensum(fk)..
- Relaciones:
 - Plan-Carrera: Un plan puede tener una o muchas carreras.

8. Sección

- Atributos: código_sección (pk), año, ciclo, sección, código_docente (fk), código_horario(fk), código_curso (fk).
- Relaciones:
 - Sección-Curso: Muchas secciones pueden estar relacionadas a un curso.
 - Sección-Docente: Una sección puede estar relacionada con un docente.
 - Sección-horario: Una sección puede tener uno o más horarios.
 - Sección-Asignación: En una sección pueden haber varias asignaciones.

9. Pensum

- Atributos: Código_pensum (pk), plan, obligatoriedad, número_créditos, nota_aprobada, zona_mínima, crédito_prerrequisito.
- Relaciones:

- Pensum-Curso: Un pensum incluye uno o más cursos.
- Pensum-plan: Un pensum puede tener un plan.

10. Prerequisito

- Atributos: codigo_prerrequisito (pk), carrera, plan, curso_prerrequisito.
- Relaciones:
 - Curso-Prerrequisito: Un curso puede tener uno o muchos prerrequisitos
 - Curso-Prerrequisito: Varios cursos pueden tener un mismo prerrequisito.

11. Horario

- Atributos: código_dia_horario (pk), dia.
- Relaciones:
 - Horario-Período: En un horario pueden haber varios periodos.
 - Horario-Salon: En un horario pueden haber varios salones.
 - Horario-Sección: Varios horarios puede tener una sección.

12. Asignación

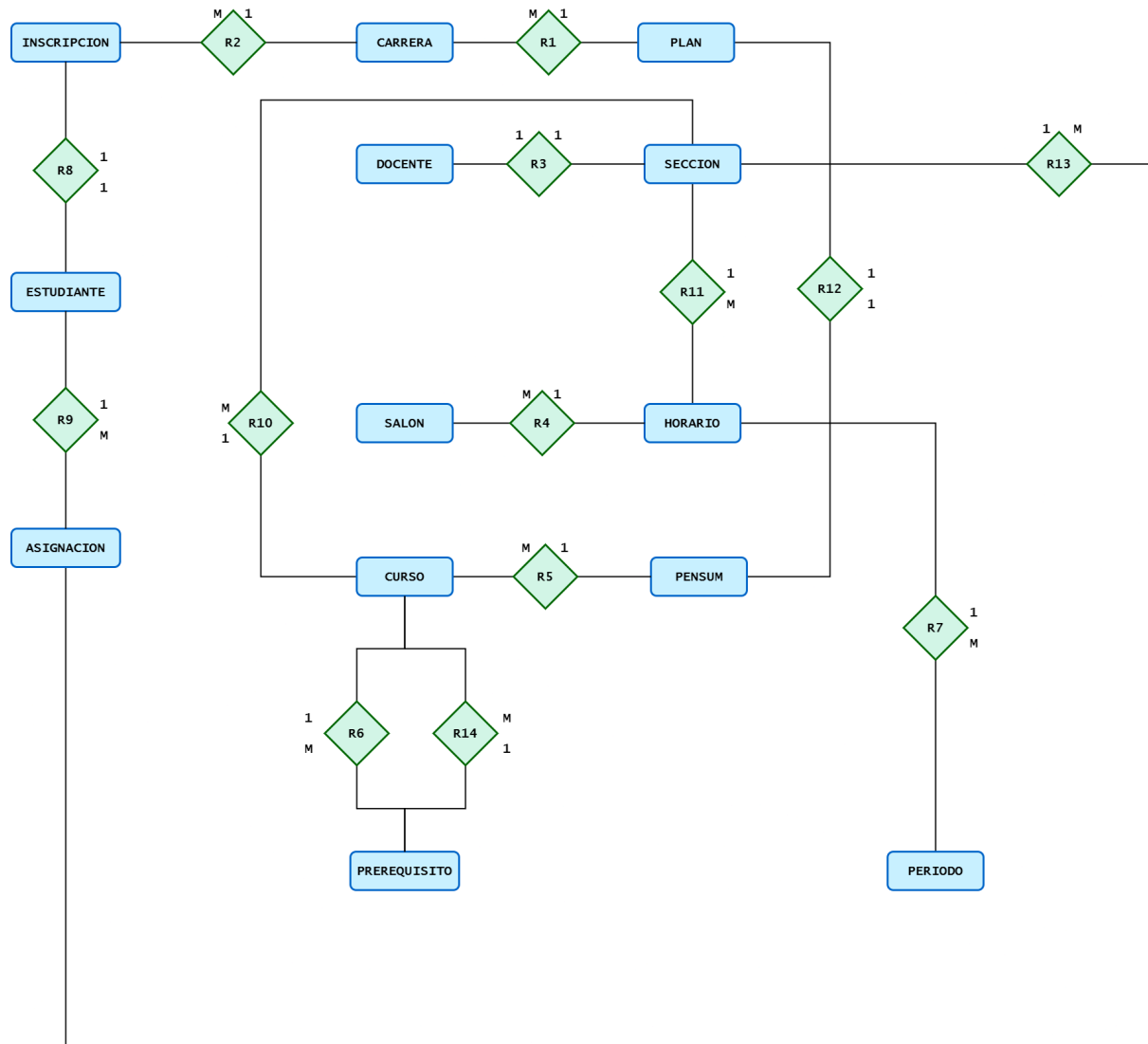
- Atributos: codigo_asignacion (pk), fecha_asignacion, estudiante(fk), zona, nota codigo_seccion (fk).
- Relaciones:
 - Asignacion-Estudiante: Un estudiante puede estar relacionado a una o muchas asignaciones.
 - Asignación-Sección: Hay varias asignaciones en una sección.

13. Inscripcion

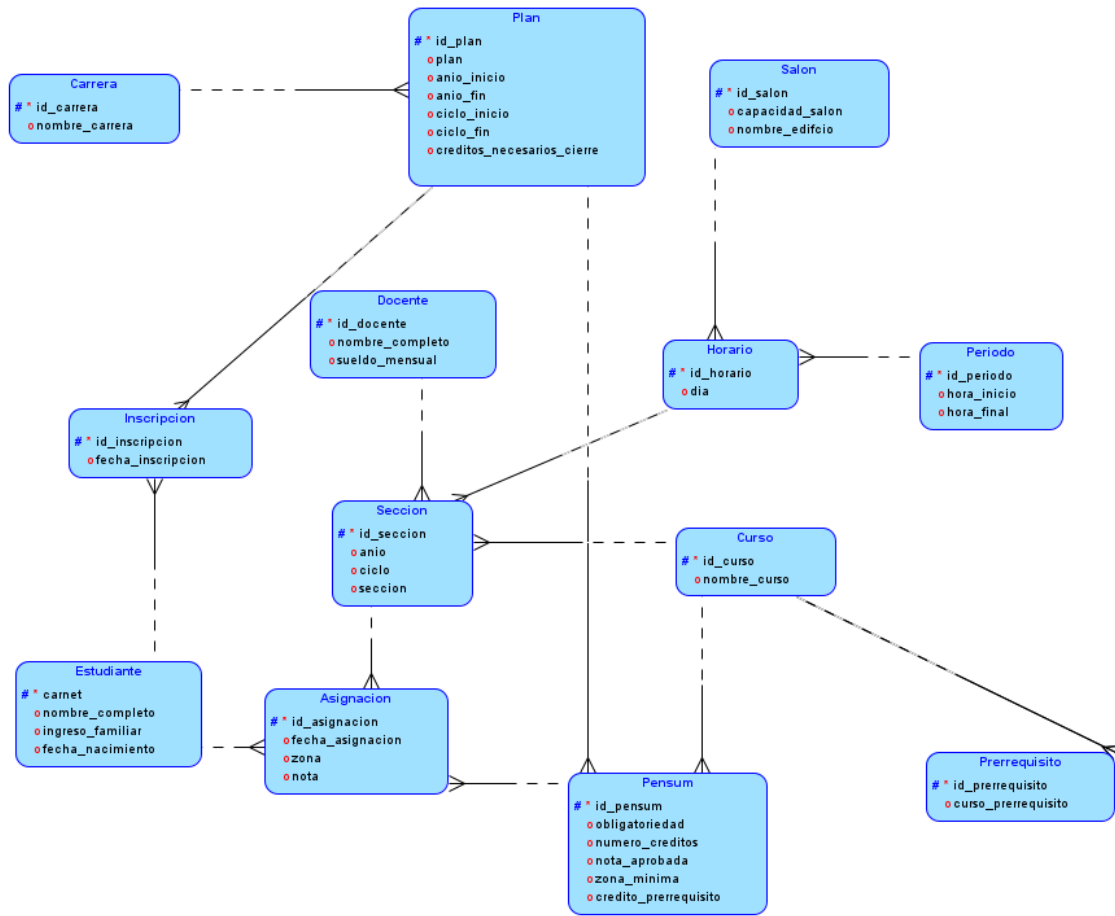
- Atributos: codigo_inscripcion (pk), fecha_inscripcion, codigo_carrera (fk), estudiante(fk)
- Relaciones:
 - Un estudiante puede tener una o más inscripciones.

- Una carrera puede estar relacionada a una o muchas inscripciones.

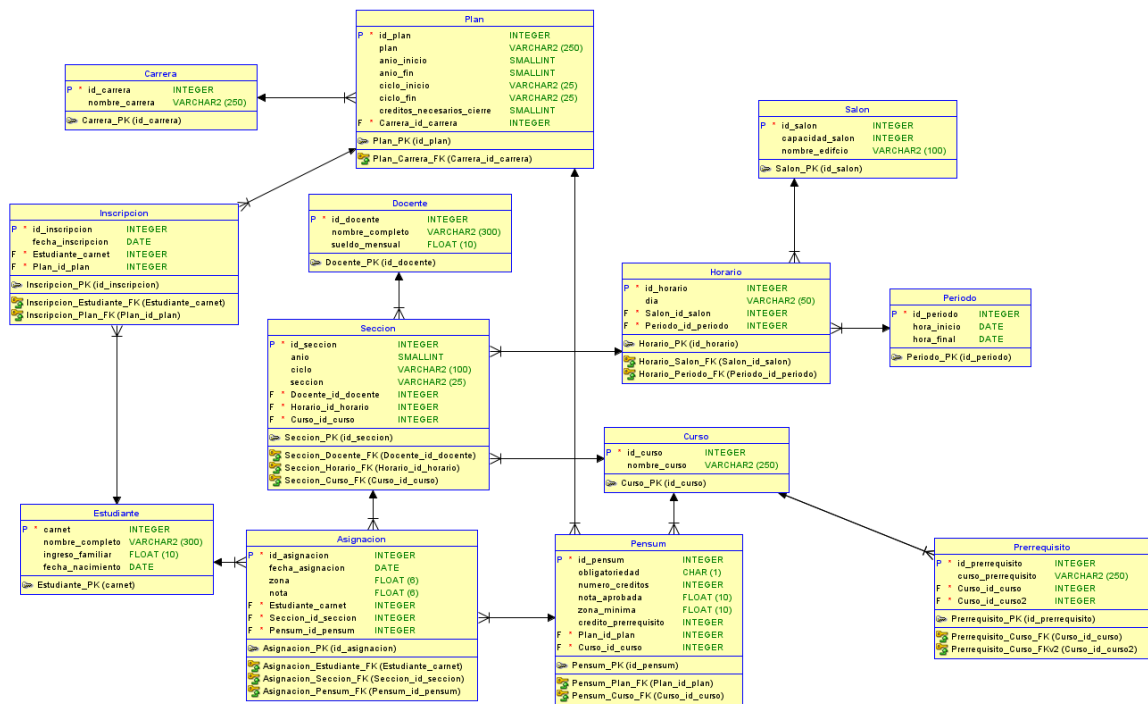
Modelo Conceptual



Modelo Lógico



Modelo Físico



Consultas

1. Consulta No.1

```

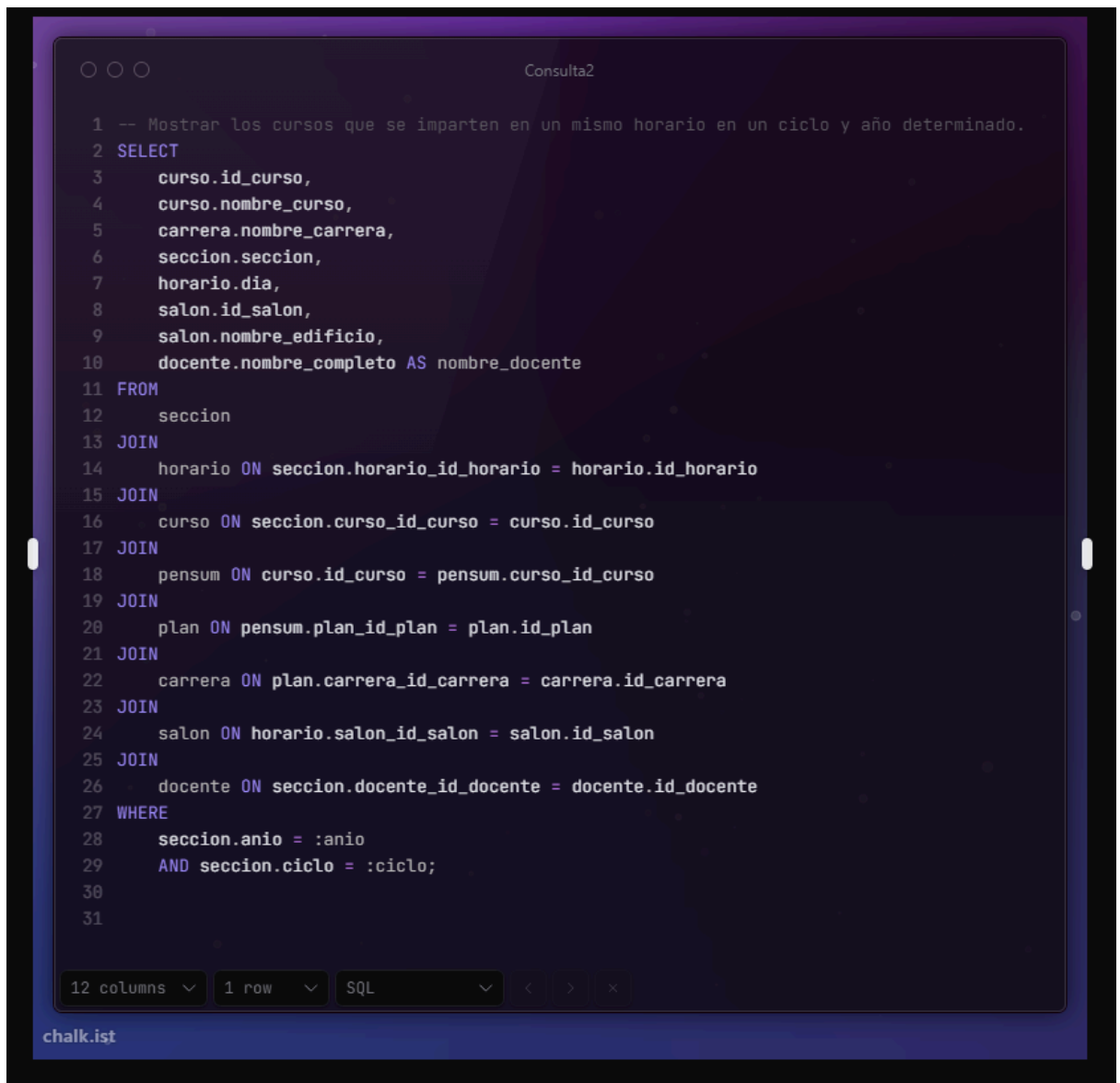
Consulta1

1 -- Mostrar el estudiante con mejor promedio por carrera el cual haya empezado a estudiar antes del 2022.
2 SELECT carrera.nombre_carrera, estudiante.nombre_completo AS mejor_estudiante,
3      MAX(calcular_promedio(estudiante.carnet, inscripcion.plan_id_plan)) AS mejor_promedio
4 FROM estudiante
5 JOIN inscripcion ON estudiante.carnet = inscripcion.estudiante_carnet
6 JOIN plan ON inscripcion.plan_id_plan = plan.id_plan
7 JOIN carrera ON plan.carrera_id_carrera = carrera.id_carrera
8 WHERE plan.anio_inicio < 2025
9 GROUP BY carrera.nombre_carrera, estudiante.nombre_completo;

12 columns 1 row SQL

```

2. Consulta No.2

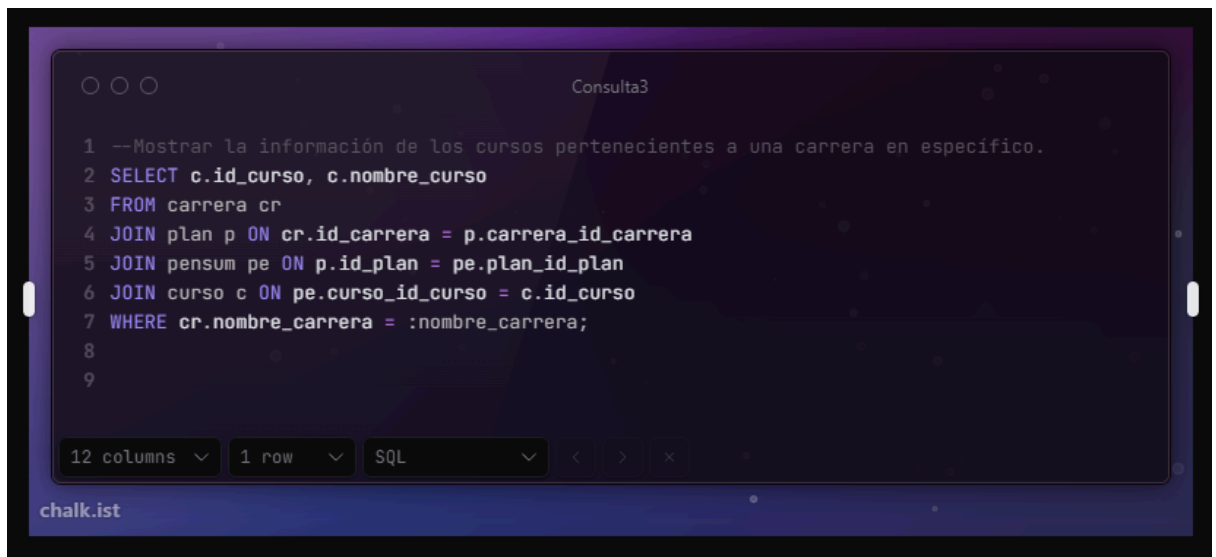


The image shows a screenshot of a SQL query editor window titled "Consulta2". The editor contains a SQL query that joins several tables to find courses taught in a specific year and cycle. The query is as follows:

```
1 -- Mostrar los cursos que se imparten en un mismo horario en un ciclo y año determinado.
2 SELECT
3     curso.id_curso,
4     curso.nombre_curso,
5     carrera.nombre_carrera,
6     seccion.seccion,
7     horario.dia,
8     salon.id_salon,
9     salon.nombre_edificio,
10    docente.nombre_completo AS nombre_docente
11 FROM
12     seccion
13 JOIN
14     horario ON seccion.horario_id_horario = horario.id_horario
15 JOIN
16     curso ON seccion.curso_id_curso = curso.id_curso
17 JOIN
18     pensum ON curso.id_curso = pensum.curso_id_curso
19 JOIN
20     plan ON pensum.plan_id_plan = plan.id_plan
21 JOIN
22     carrera ON plan.carrera_id_carrera = carrera.id_carrera
23 JOIN
24     salon ON horario.salon_id_salon = salon.id_salon
25 JOIN
26     docente ON seccion.docente_id_docente = docente.id_docente
27 WHERE
28     seccion.anio = :anio
29     AND seccion.ciclo = :ciclo;
30
31
```

At the bottom of the editor, there is a status bar showing "12 columns", "1 row", and "SQL". The editor is styled with a dark background and light-colored text.

3. Consulta No.3



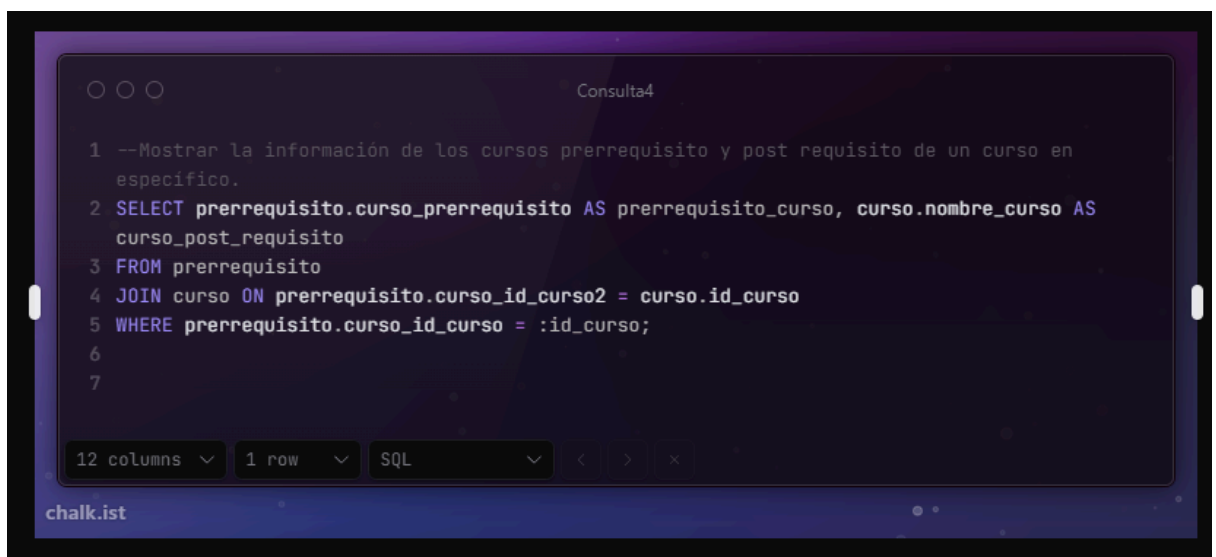
A screenshot of a SQL query editor window titled 'Consulta3'. The editor has a dark theme and shows a SQL query with 9 lines. The query selects course information for a specific career. The interface includes a status bar at the bottom showing '12 columns', '1 row', and 'SQL'.

```
1 --Mostrar la información de los cursos pertenecientes a una carrera en específico.
2 SELECT c.id_curso, c.nombre_curso
3 FROM carrera cr
4 JOIN plan p ON cr.id_carrera = p.carrera_id_carrera
5 JOIN pensum pe ON p.id_plan = pe.plan_id_plan
6 JOIN curso c ON pe.curso_id_curso = c.id_curso
7 WHERE cr.nombre_carrera = :nombre_carrera;
8
9
```

12 columns 1 row SQL

chalk.ist

4. Consulta No.4



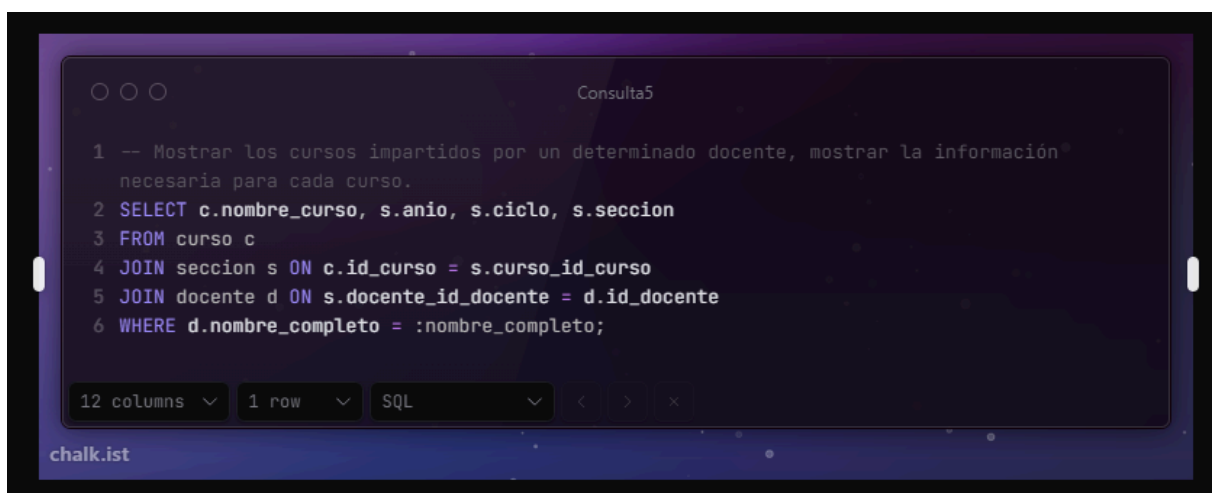
A screenshot of a SQL query editor window titled 'Consulta4'. The editor shows a SQL query with 7 lines. The query selects prerequisite and post-requisite course information for a specific course. The interface includes a status bar at the bottom showing '12 columns', '1 row', and 'SQL'.

```
1 --Mostrar la información de los cursos prerequisito y post requisito de un curso en
  específico.
2 SELECT prerequisito.curso_prerequisito AS prerequisito_curso, curso.nombre_curso AS
  curso_post_requisito
3 FROM prerequisito
4 JOIN curso ON prerequisito.curso_id_curso2 = curso.id_curso
5 WHERE prerequisito.curso_id_curso = :id_curso;
6
7
```

12 columns 1 row SQL

chalk.ist

5. Consulta No.5



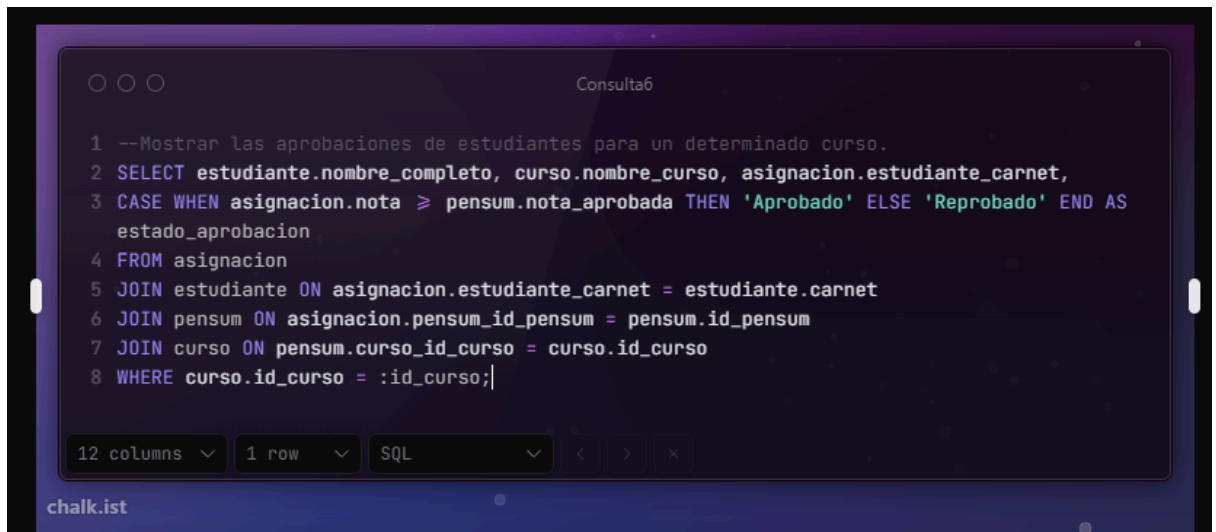
A screenshot of a SQL query editor window titled 'Consulta5'. The editor shows a SQL query with 6 lines. The query displays courses taught by a specific teacher, including course name, year, cycle, and section. The interface includes a status bar at the bottom showing '12 columns', '1 row', and 'SQL'.

```
1 -- Mostrar los cursos impartidos por un determinado docente, mostrar la información
  necesaria para cada curso.
2 SELECT c.nombre_curso, s.anio, s.ciclo, s.seccion
3 FROM curso c
4 JOIN seccion s ON c.id_curso = s.curso_id_curso
5 JOIN docente d ON s.docente_id_docente = d.id_docente
6 WHERE d.nombre_completo = :nombre_completo;
```

12 columns 1 row SQL

chalk.ist

6. Consulta No.6



The screenshot shows a SQL query editor window titled "Consulta6". The query is as follows:

```
1 --Mostrar las aprobaciones de estudiantes para un determinado curso.
2 SELECT estudiante.nombre_completo, curso.nombre_curso, asignacion.estudiante_carnet,
3 CASE WHEN asignacion.nota >= pensum.nota_aprobada THEN 'Aprobado' ELSE 'Reprobado' END AS
   estado_aprobacion
4 FROM asignacion
5 JOIN estudiante ON asignacion.estudiante_carnet = estudiante.carnet
6 JOIN pensum ON asignacion.pensum_id_pensum = pensum.id_pensum
7 JOIN curso ON pensum.curso_id_curso = curso.id_curso
8 WHERE curso.id_curso = :id_curso;
```

Below the query, the execution results are displayed:

12 columns	1 row	SQL

The interface includes a "chalk.ist" logo at the bottom left and navigation buttons (back, forward, close) on the right side of the results area.

Script

```
CREATE TABLE asignacion (  
    id_asignacion    INTEGER NOT NULL,  
    fecha_asignacion DATE,  
    zona             FLOAT(6),  
    nota             FLOAT(6),  
    estudiante_carnet INTEGER NOT NULL,  
    seccion_id_seccion INTEGER NOT NULL,  
    pensum_id_pensum  INTEGER NOT NULL  
);
```

```
ALTER TABLE asignacion ADD CONSTRAINT asignacion_pk PRIMARY KEY (  
id_asignacion );
```

```
CREATE TABLE carrera (  
    id_carrera    INTEGER NOT NULL,  
    nombre_carrera VARCHAR2(250)  
);
```

```
ALTER TABLE carrera ADD CONSTRAINT carrera_pk PRIMARY KEY (  
id_carrera );
```

```
CREATE TABLE curso (  
    id_curso    INTEGER NOT NULL,  
    nombre_curso VARCHAR2(250)  
);
```

```
ALTER TABLE curso ADD CONSTRAINT curso_pk PRIMARY KEY ( id_curso  
);
```

```
CREATE TABLE docente (  
    id_docente    INTEGER NOT NULL,  
    nombre_completo VARCHAR2(300),  
    sueldo_mensual FLOAT(10)  
);
```

```
ALTER TABLE docente ADD CONSTRAINT docente_pk PRIMARY KEY (  
id_docente );
```

```
CREATE TABLE estudiante (  
    carnet    INTEGER NOT NULL,  
    nombre_completo VARCHAR2(300),  
    ingreso_familiar FLOAT(10),
```

```
    fecha_nacimiento DATE
);
```

```
ALTER TABLE estudiante ADD CONSTRAINT estudiante_pk PRIMARY KEY (
carnet );
```

```
CREATE TABLE horario (
    id_horario      INTEGER NOT NULL,
    dia             VARCHAR2(50),
    salon_id_salon  INTEGER NOT NULL,
    periodo_id_periodo INTEGER NOT NULL
);
```

```
ALTER TABLE horario ADD CONSTRAINT horario_pk PRIMARY KEY (
id_horario );
```

```
CREATE TABLE inscripcion (
    id_inscripcion  INTEGER NOT NULL,
    fecha_inscripcion DATE,
    estudiante_carnet INTEGER NOT NULL,
    plan_id_plan    INTEGER NOT NULL
);
```

```
ALTER TABLE inscripcion ADD CONSTRAINT inscripcion_pk PRIMARY KEY
( id_inscripcion );
```

```
CREATE TABLE pensum (
    id_pensum        INTEGER NOT NULL,
    obligatoriedad   CHAR(1),
    numero_creditos  INTEGER,
    nota_aprobada    FLOAT(10),
    zona_minima      FLOAT(10),
    credito_prerrequisito INTEGER,
    plan_id_plan     INTEGER NOT NULL,
    curso_id_curso   INTEGER NOT NULL
);
```

```
ALTER TABLE pensum ADD CONSTRAINT pensum_pk PRIMARY KEY (
id_pensum );
```

```
CREATE TABLE periodo (
    id_periodo  INTEGER NOT NULL,
    hora_inicio DATE,
    hora_final  DATE
);
```

```
ALTER TABLE periodo ADD CONSTRAINT periodo_pk PRIMARY KEY (
id_periodo );
```

```
CREATE TABLE plan (
  id_plan          INTEGER NOT NULL,
  plan             VARCHAR2(250),
  anio_inicio      SMALLINT,
  anio_fin         SMALLINT,
  ciclo_inicio     VARCHAR2(25),
  ciclo_fin        VARCHAR2(25),
  credits_necesarios_cierre SMALLINT,
  carrera_id_carrera INTEGER NOT NULL
);
```

```
ALTER TABLE plan ADD CONSTRAINT plan_pk PRIMARY KEY ( id_plan );
```

```
CREATE TABLE prerrequisito (
  id_prerrequisito INTEGER NOT NULL,
  curso_prerrequisito VARCHAR2(250),
  curso_id_curso    INTEGER NOT NULL,
  curso_id_curso2    INTEGER NOT NULL
);
```

```
ALTER TABLE prerrequisito ADD CONSTRAINT prerrequisito_pk PRIMARY
KEY ( id_prerrequisito );
```

```
CREATE TABLE salon (
  id_salon      INTEGER NOT NULL,
  capacidad_salon INTEGER,
  nombre_edificio VARCHAR2(100)
);
```

```
ALTER TABLE salon ADD CONSTRAINT salon_pk PRIMARY KEY ( id_salon );
```

```
CREATE TABLE seccion (
  id_seccion      INTEGER NOT NULL,
  anio            SMALLINT,
  ciclo          VARCHAR2(100),
  seccion         VARCHAR2(25),
  docente_id_docente INTEGER NOT NULL,
  horario_id_horario INTEGER NOT NULL,
  curso_id_curso  INTEGER NOT NULL
);
```

```
ALTER TABLE seccion ADD CONSTRAINT seccion_pk PRIMARY KEY (
id_seccion );
```

```
ALTER TABLE asignacion
  ADD CONSTRAINT asignacion_estudiante_fk FOREIGN KEY (
estudiante_carnet )
  REFERENCES estudiante ( carnet );
```

```
ALTER TABLE asignacion
  ADD CONSTRAINT asignacion_pensum_fk FOREIGN KEY (
pensum_id_pensum )
  REFERENCES pensum ( id_pensum );
```

```
ALTER TABLE asignacion
  ADD CONSTRAINT asignacion_seccion_fk FOREIGN KEY (
seccion_id_seccion )
  REFERENCES seccion ( id_seccion );
```

```
ALTER TABLE horario
  ADD CONSTRAINT horario_periodo_fk FOREIGN KEY (
periodo_id_periodo )
  REFERENCES periodo ( id_periodo );
```

```
ALTER TABLE horario
  ADD CONSTRAINT horario_salon_fk FOREIGN KEY ( salon_id_salon )
  REFERENCES salon ( id_salon );
```

```
ALTER TABLE inscripcion
  ADD CONSTRAINT inscripcion_estudiante_fk FOREIGN KEY (
estudiante_carnet )
  REFERENCES estudiante ( carnet );
```

```
ALTER TABLE inscripcion
  ADD CONSTRAINT inscripcion_plan_fk FOREIGN KEY ( plan_id_plan )
  REFERENCES plan ( id_plan );
```

```
ALTER TABLE pensum
  ADD CONSTRAINT pensum_curso_fk FOREIGN KEY ( curso_id_curso )
  REFERENCES curso ( id_curso );
```

```
ALTER TABLE pensum
  ADD CONSTRAINT pensum_plan_fk FOREIGN KEY ( plan_id_plan )
  REFERENCES plan ( id_plan );
```

```
ALTER TABLE plan
```

```
ADD CONSTRAINT plan_carrera_fk FOREIGN KEY ( carrera_id_carrera )  
REFERENCES carrera ( id_carrera );
```

```
ALTER TABLE prerrequisito  
ADD CONSTRAINT prerrequisito_curso_fk FOREIGN KEY (  
curso_id_curso )  
REFERENCES curso ( id_curso );
```

```
ALTER TABLE prerrequisito  
ADD CONSTRAINT prerrequisito_curso_fkv2 FOREIGN KEY (  
curso_id_curso2 )  
REFERENCES curso ( id_curso );
```

```
ALTER TABLE seccion  
ADD CONSTRAINT seccion_curso_fk FOREIGN KEY ( curso_id_curso )  
REFERENCES curso ( id_curso );
```

```
ALTER TABLE seccion  
ADD CONSTRAINT seccion_docente_fk FOREIGN KEY (  
docente_id_docente )  
REFERENCES docente ( id_docente );
```

```
ALTER TABLE seccion  
ADD CONSTRAINT seccion_horario_fk FOREIGN KEY (  
horario_id_horario )  
REFERENCES horario ( id_horario );
```