

DISEÑO DE UNA BASE DE DATOS

PPRUEBA TECNICA

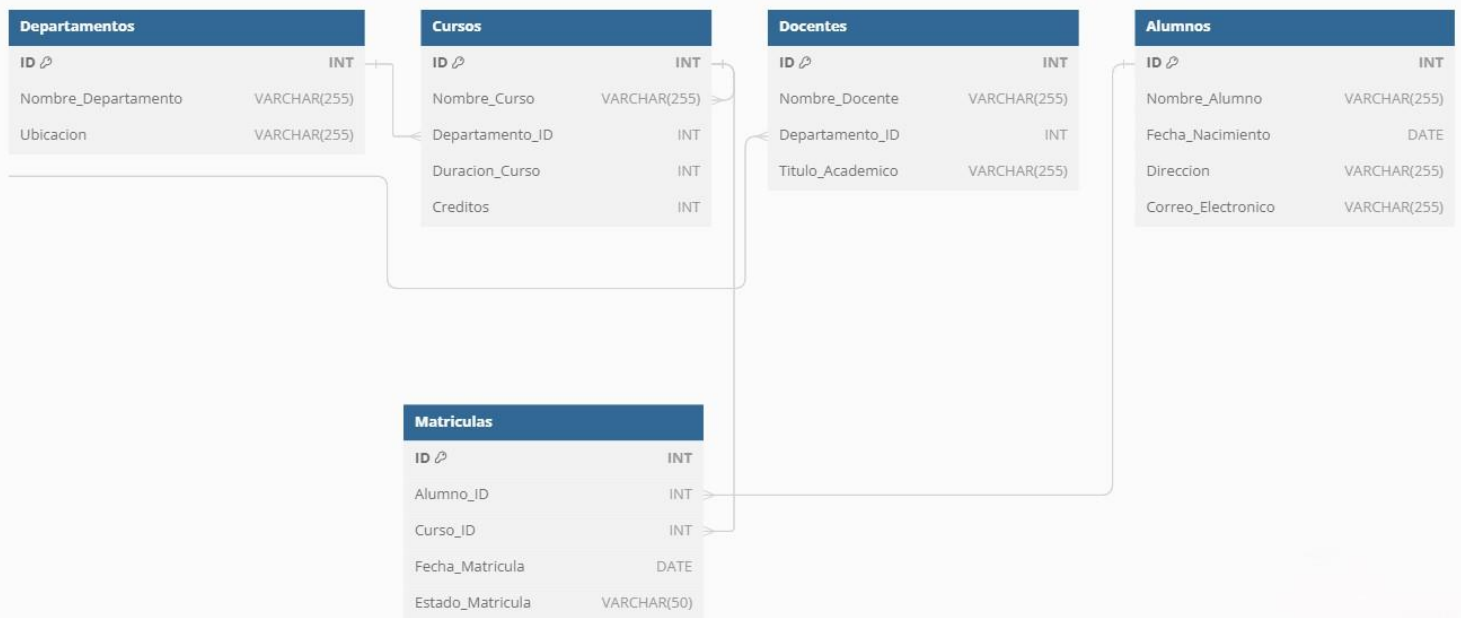
Repositorio en GitHub con los pasos para desarrollar esta prueba usando DOCKER y el archivo sql para usarlo de manera local en MySQL Workbench V.8.0.29

https://github.com/Cayo322/DB_Centro_Educativo.git

Paso 1:

Diagrama Entidad relación y seleccionar el gestor de base de datos adecuado para este proyecto:

En vista que el proyecto es pequeño y no necesita características empresariales avanzadas, MySQL puede ser más que suficiente.



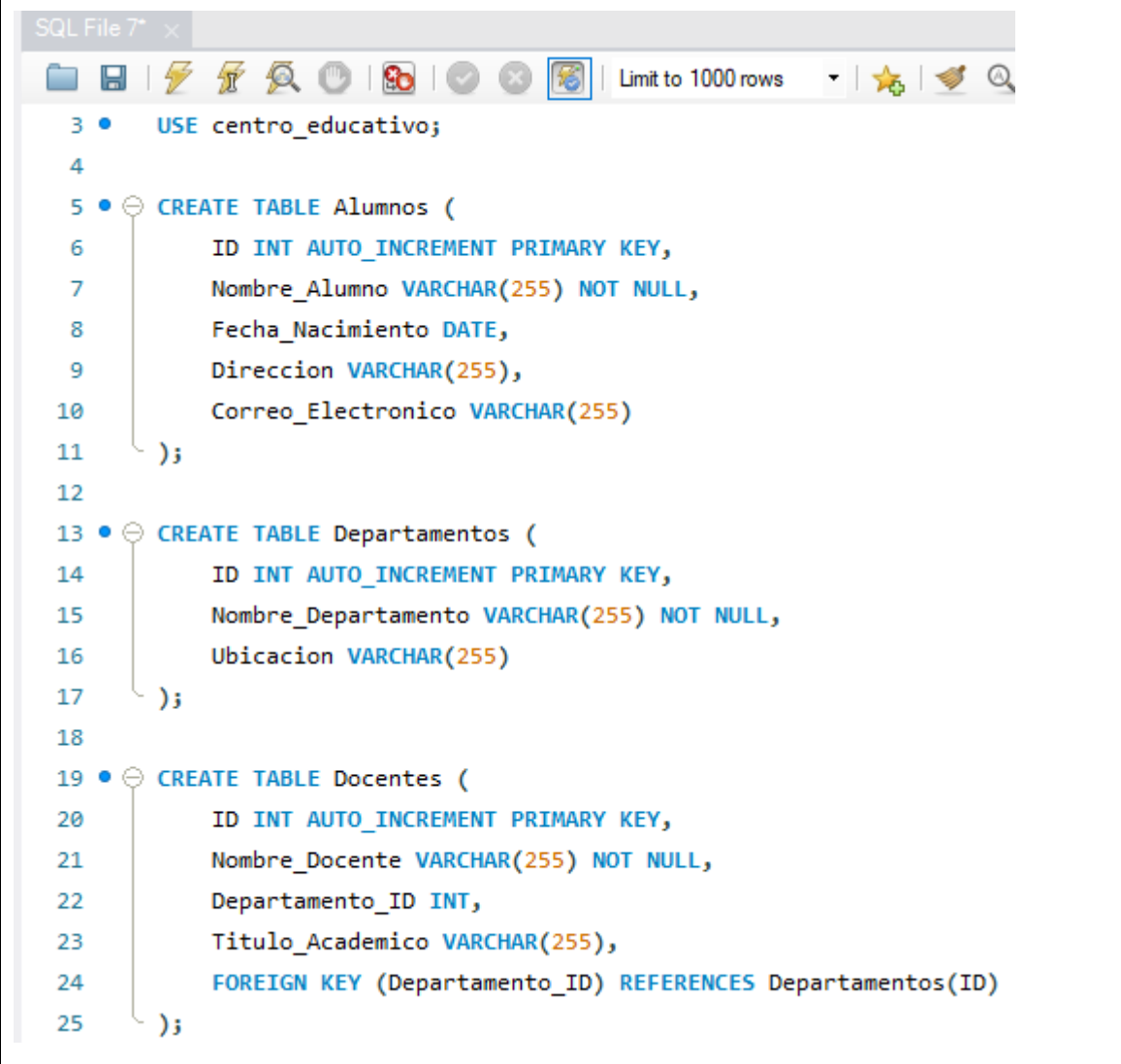
Paso 2:

Creación de la base de datos y las tablas con sus relaciones respectivas

Primero tenemos que crear la Base de datos con un nombre que sea muy referencial al proyecto y su funcionamiento.



Ahora que ya tenemos la base de datos lista para su uso, procedemos a crear las tablas.

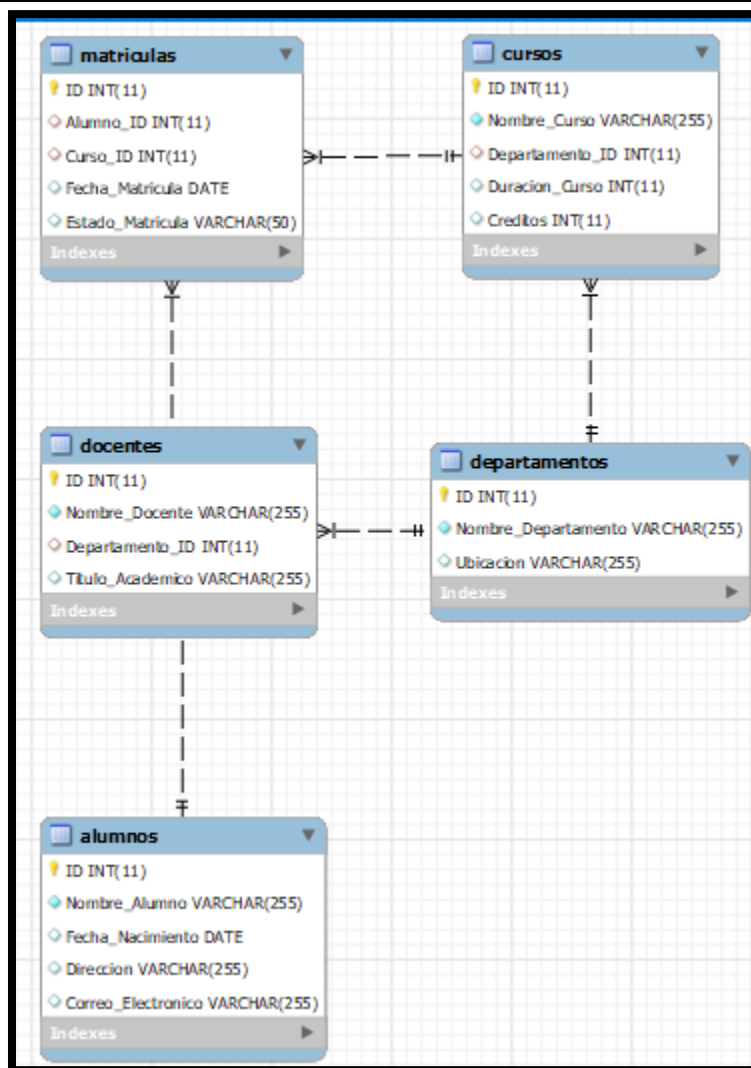


```

26
27 ● ○ CREATE TABLE Cursos (
28     ID INT AUTO_INCREMENT PRIMARY KEY,
29     Nombre_Curso VARCHAR(255) NOT NULL,
30     Departamento_ID INT,
31     Duracion_Curso INT,
32     Creditos INT,
33     FOREIGN KEY (Departamento_ID) REFERENCES Departamentos(ID)
34 );
35
36 ● ○ CREATE TABLE Matriculas (
37     ID INT AUTO_INCREMENT PRIMARY KEY,
38     Alumno_ID INT,
39     Curso_ID INT,
40     Fecha_Matricula DATE,
41     Estado_Matricula VARCHAR(50),
42     FOREIGN KEY (Alumno_ID) REFERENCES Alumnos(ID),
43     FOREIGN KEY (Curso_ID) REFERENCES Cursos(ID)
44 );

```

Verificando diagrama Entidad Relación



Insertando datos tabla:

Alumnos

```

INSERT INTO
  Alumnos (
    Nombre_Alumno,
    Fecha_Nacimiento,
    Direccion,
    Correo_Electronico
  )
VALUES (
  'Cayo Phocco',
  '2000-11-21',
  'Calle rosales 123',
  'cayo.phocco@gmail.com'
),
  
```

```
(
    'Alberto Machacca',
    '2001-05-21',
    'Calle Lujan 123',
    'alberto.machacca@gmail.com'
),
(
    'Alan García',
    '1961-05-21',
    'Mar de Bolivia 123',
    'alan.garcia@gmail.com'
);
```

Departamentos

```
INSERT INTO
    Departamentos (
        Nombre_Departamento,
        Ubicacion
    )
VALUES (
    'Ciencias de la Computación',
    'Edificio A'
),
(
    'Estudios Generales',
    'Edificio B'
),
('Mecanica', 'Edificio C');
```

Docentes

```
INSERT INTO
    Docentes (
        Nombre_Docente,
        Departamento_ID,
        Titulo_Academico
    )
VALUES (
    'Cesar Bobadilla',
    1,
    'Ingeniero en Computación'
),
(
    'Hugo Sanchez',
```

```

        2,
        'Economista'
    ),
    (
        'Julio Belarde',
        3,
        'Ingeniero Mecánico'
    );

```

Cursos

```

INSERT INTO
    Cursos (
        Nombre_Curso,
        Departamento_ID,
        Duracion_Curso,
        Creditos
    )
VALUES (
    'Introducción a la Programación',
    1,
    60,
    4
),
(
    'Fundamentos de la economía',
    2,
    60,
    4
),
(
    'Mecánica de fluidos',
    3,
    60,
    4
);

```

Matrículas

```

INSERT INTO
    Matriculas (
        Alumno_ID,
        Curso_ID,
        Fecha_Matricula,
        Estado_Matricula
    )

```

```

)
VALUES (1, 1, '2024-08-11', 'Activa'),
       (2, 2, '2024-08-11', 'Activa'),
       (3, 3, '2024-08-11', 'Activa');

```

Consultas simples a las tablas Alumnos, Docentes y Cursos

Alumnos

SQL File 7 SQL File 8 SQL File 9 SQL File 10 alumnos alumnos

1 • `SELECT * FROM centro_educativo.alumnos;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ID	Nombre_Alumno	Fecha_Nacimiento	Direccion	Correo_Electronico
▶	1	Cayo Phocco	2000-11-21	Calle rosales 123	cayo.phocco@gmail.com
	2	Alberto Machacca	2001-05-21	Calle Lujan 123	alberto.machacca@gmail.com
	3	Alan García	1961-05-21	Mar de Bolivia 123	alan.garcia@gmail.com
	4	Peter Castle	1951-05-21	Cajamarca chota 123	peter.castle@gmail.com
	5	Kekis Fujimori	1971-05-21	Palacio de gobierno 123	kekis.fuji@gmail.com
	6	Martin Vizcarra	1965-05-24	Av fiu fiu Moquegua 123	martin.vizcarra@gmail.com
*	NULL	NULL	NULL	NULL	NULL

Docentes

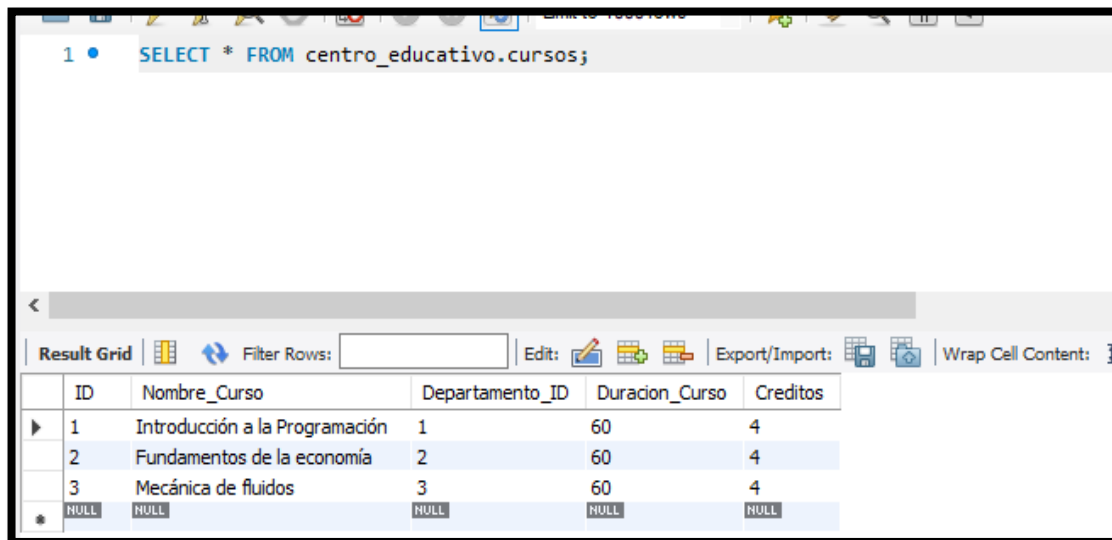
SQL File 7* SQL File 8* SQL File 9* SQL File 10* alumnos alumnos docentes

1 • `SELECT * FROM centro_educativo.docentes;`

Result Grid Filter Rows: Edit: Export/Import: Wrap C

	ID	Nombre_Docente	Departamento_ID	Titulo_Academico
▶	1	Cesar Bobadilla	1	Ingeniero en Computación
	2	Hugo Sanchez	2	Economista
	3	Julio Belarde	3	Ingeniero Mecánico
	4	Abel Ccopa	1	Ingeniero de Sistemas
	5	Carlos Tevez	2	Desarrollo Personal
	6	Isaac Newton	3	Físico
*	NULL	NULL	NULL	NULL

Cursos

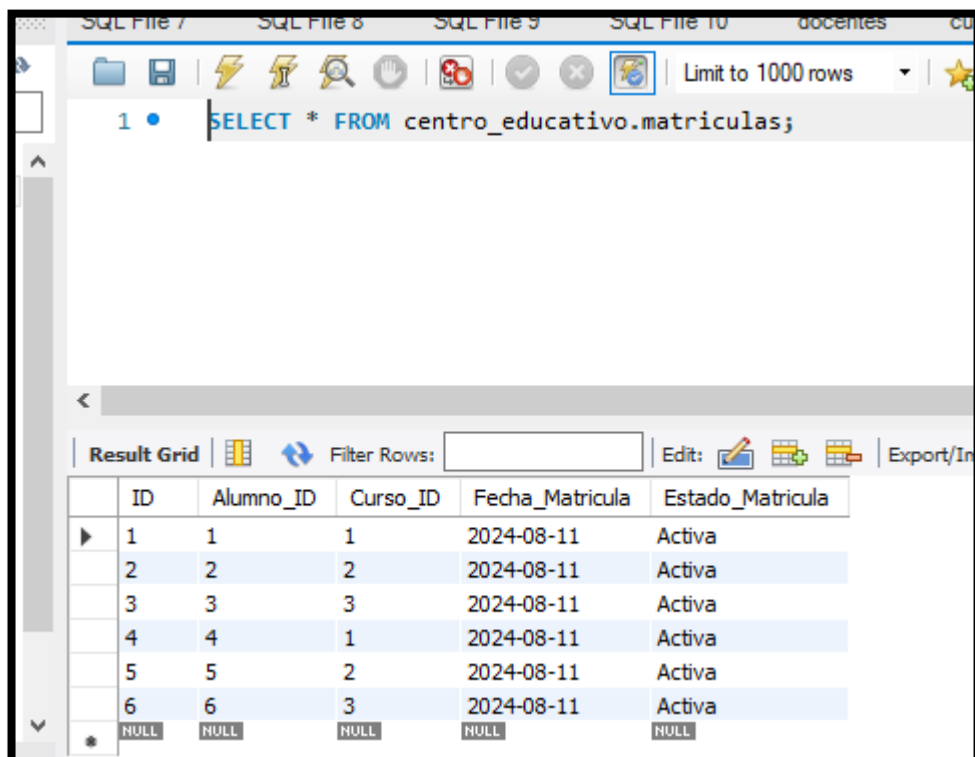


1 • `SELECT * FROM centro_educativo.cursos;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	ID	Nombre_Curso	Departamento_ID	Duracion_Curso	Creditos
▶	1	Introducción a la Programación	1	60	4
	2	Fundamentos de la economía	2	60	4
	3	Mecánica de fluidos	3	60	4
*	NULL	NULL	NULL	NULL	NULL

Matriculas



SQL File 7 | SQL File 8 | SQL File 9 | SQL File 10 | docentes | cu

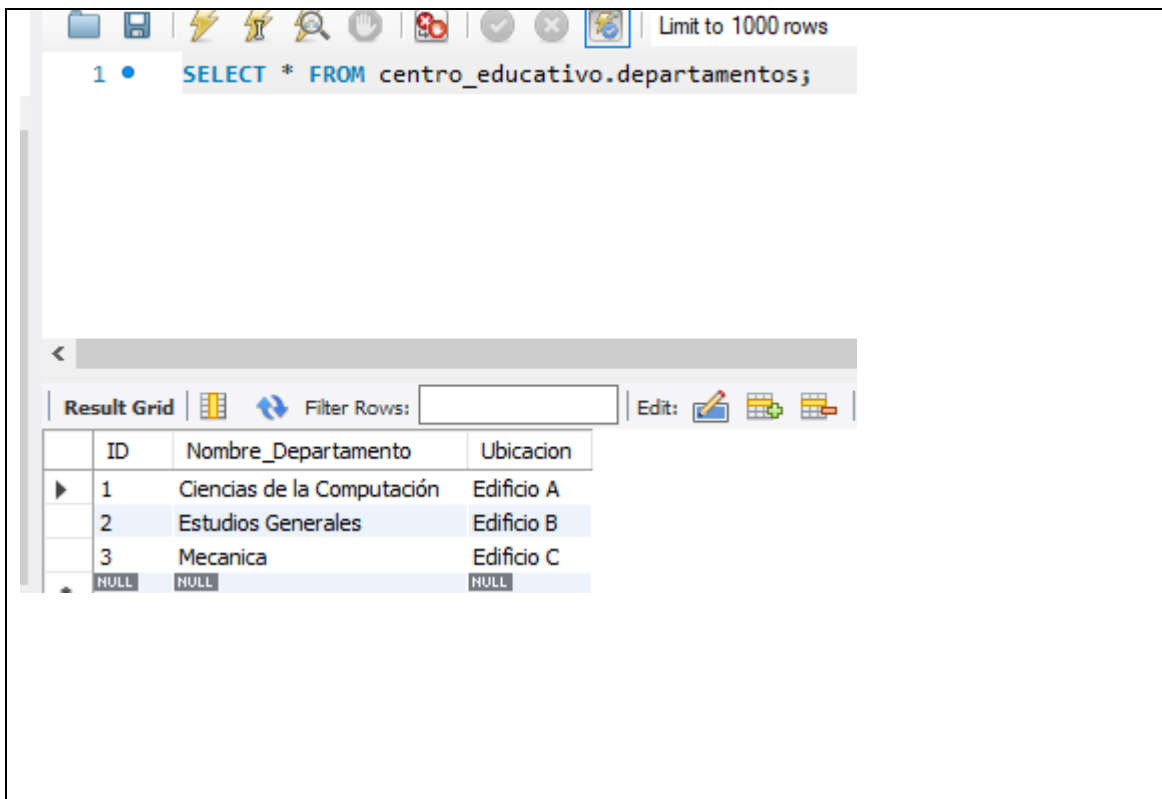
1 • `SELECT * FROM centro_educativo.matriculas;`

Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Export/Import: |

	ID	Alumno_ID	Curso_ID	Fecha_Matricula	Estado_Matricula
▶	1	1	1	2024-08-11	Activa
	2	2	2	2024-08-11	Activa
	3	3	3	2024-08-11	Activa
	4	4	1	2024-08-11	Activa
	5	5	2	2024-08-11	Activa
	6	6	3	2024-08-11	Activa
*	NULL	NULL	NULL	NULL	NULL

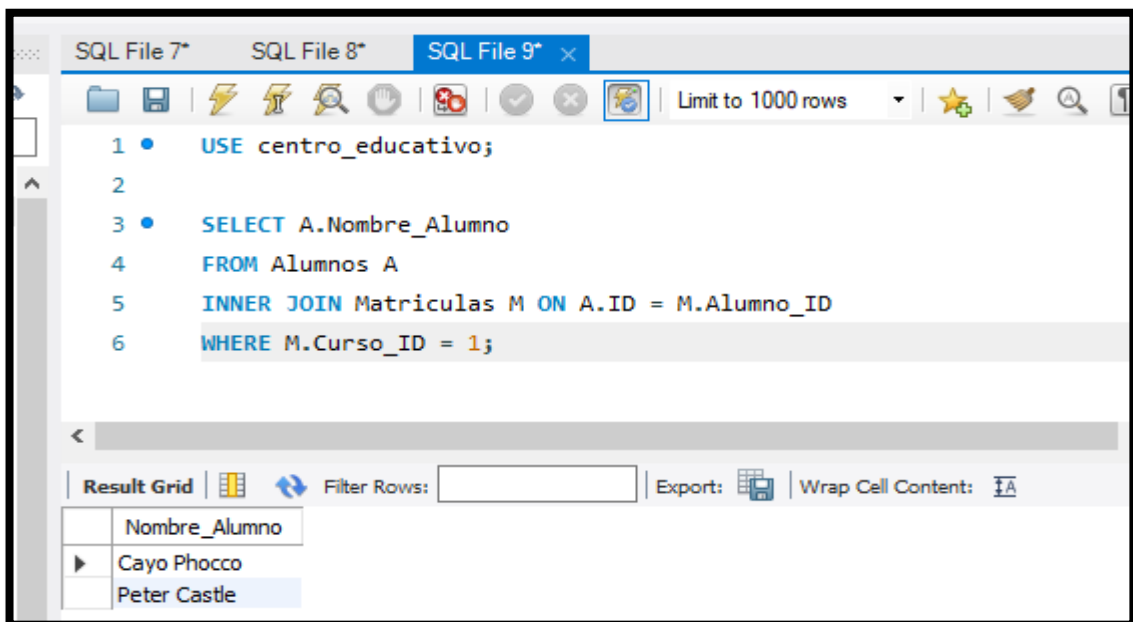
Departamentos

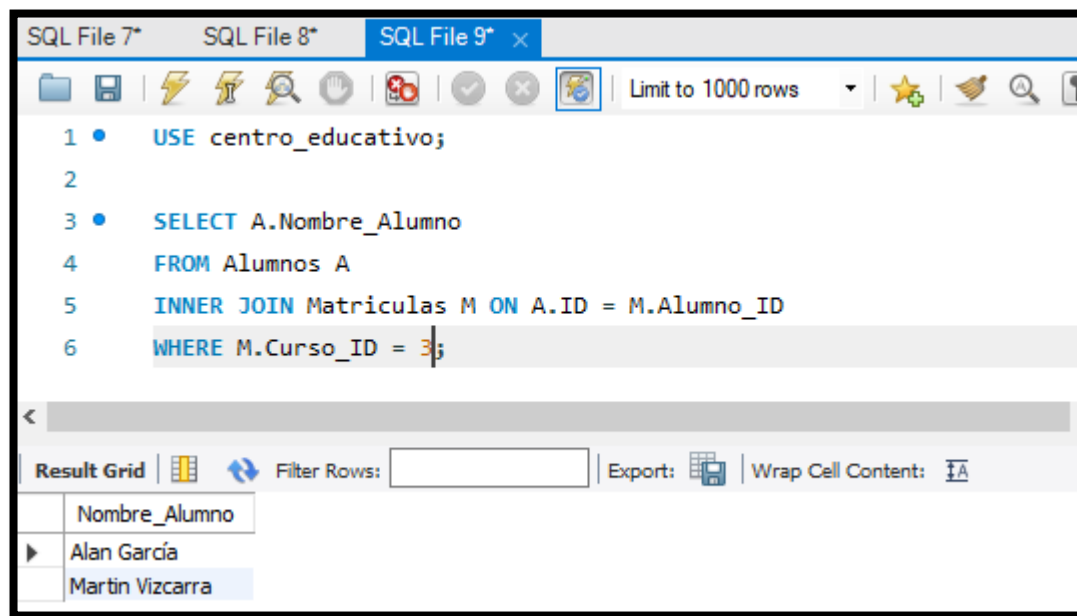


Paso 3

Consultas SQL

Escribe una consulta SQL que muestre todos los alumnos matriculados en un curso específico.





□ SELECT A.Nombre_Alumno:

- SELECT: Esta cláusula se utiliza para especificar las columnas que deseas obtener como resultado de la consulta.
- A.Nombre_Alumno: Hace referencia al nombre del alumno desde la tabla Alumnos. A es un alias para la tabla Alumnos, lo que significa que puedes usar A en lugar de escribir Alumnos cada vez.

□ FROM Alumnos A:

- FROM: Esta cláusula especifica la tabla de la que se obtendrán los datos.
- Alumnos A: Indica que los datos se obtienen de la tabla Alumnos. A es un alias para simplificar el uso de la tabla en la consulta.

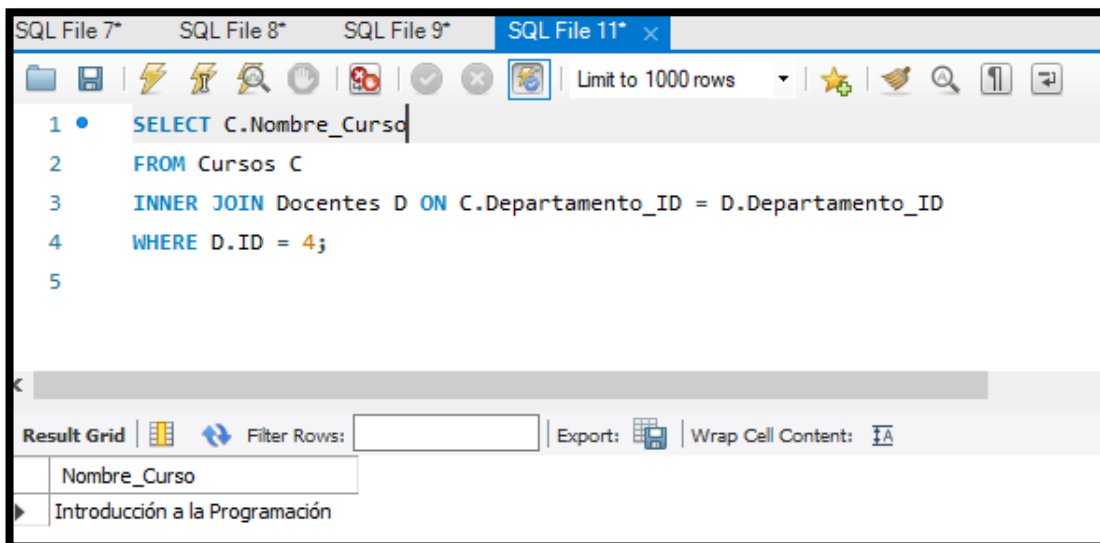
□ INNER JOIN Matriculas M ON A.ID = M.Alumno_ID:

- INNER JOIN: Esta cláusula combina filas de dos tablas basándose en una condición que especificas. Solo devuelve filas donde hay una coincidencia en ambas tablas.
- Matriculas M: Especifica que se unirá la tabla Matriculas. M es un alias para esta tabla.
- ON A.ID = M.Alumno_ID: Especifica la condición de unión. Aquí, la consulta une las tablas Alumnos y Matriculas donde el ID del alumno en la tabla Alumnos coincide con el campo Alumno_ID en la tabla Matriculas.

□ WHERE M.Curso_ID = <id_curso>:

- WHERE: Filtra las filas devueltas por la consulta basándose en una condición.
- M.Curso_ID = <id_curso>: Esta condición especifica que solo se deben devolver las filas donde el Curso_ID en la tabla Matriculas coincide con el ID del curso específico que se quiere consultar (indicado por <id_curso>).

Escribe una consulta SQL que muestre todos los cursos en los que un docente en específico está asignado como instructor.

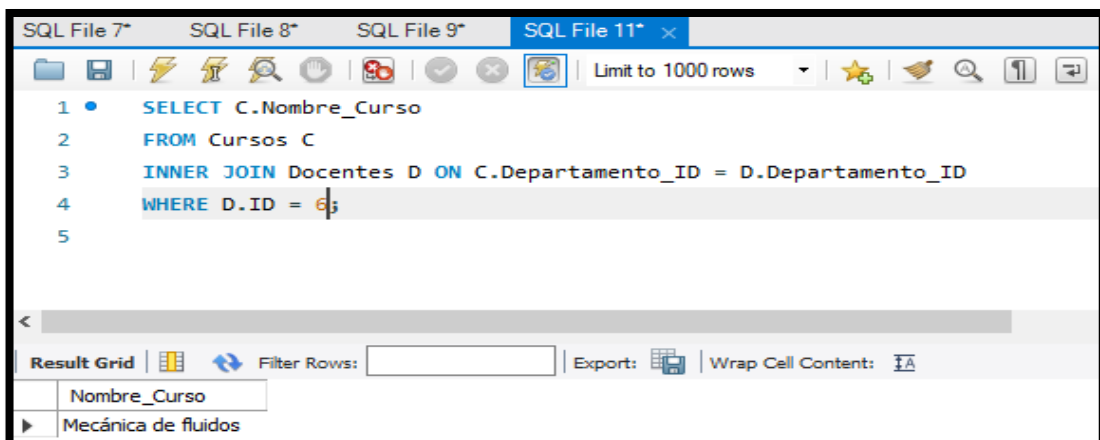


The screenshot shows a SQL editor window titled 'SQL File 11'. The query is as follows:

```
1 • SELECT C.Nombre_Curso
2 FROM Cursos C
3 INNER JOIN Docentes D ON C.Departamento_ID = D.Departamento_ID
4 WHERE D.ID = 4;
5
```

Below the query, the 'Result Grid' is visible, showing a table with the following data:

Nombre_Curso
Introducción a la Programación



The screenshot shows the same SQL editor window with the query modified to filter for instructor ID 6:

```
1 • SELECT C.Nombre_Curso
2 FROM Cursos C
3 INNER JOIN Docentes D ON C.Departamento_ID = D.Departamento_ID
4 WHERE D.ID = 6;
5
```

The 'Result Grid' now shows a different result:

Nombre_Curso
Mecánica de fluidos

- **SELECT C.Nombre_Curso:**
 - **SELECT:** Indica las columnas que deseas obtener.
 - **C.Nombre_Curso:** Especifica que quieres obtener el nombre de los cursos desde la tabla Cursos. C es un alias para la tabla Cursos.
- **FROM Cursos C:**
 - **FROM:** Especifica la tabla de la cual se obtendrán los datos.
 - **Cursos C:** Indica que los datos se obtienen de la tabla Cursos. C es un alias para la tabla Cursos.
- **INNER JOIN Docentes D ON C.Departamento_ID = D.Departamento_ID:**
 - **INNER JOIN:** Combina filas de dos tablas basándose en una condición.
 - **Docentes D:** Especifica que se unirá la tabla Docentes. D es un alias para esta tabla.
 - **ON C.Departamento_ID = D.Departamento_ID:** Especifica la condición de unión. Aquí, la consulta une las tablas Cursos y Docentes donde el Departamento_ID en la tabla Cursos coincide con el Departamento_ID en la tabla Docentes.
- **WHERE D.ID = <id_docente>:**
 - **WHERE:** Filtra las filas devueltas basándose en una condición.
 - **D.ID = <id_docente>:** Especifica que solo se deben devolver las filas donde el ID del docente en la tabla Docentes coincide con el ID del docente específico que se quiere consultar (indicado por <id_docente>).