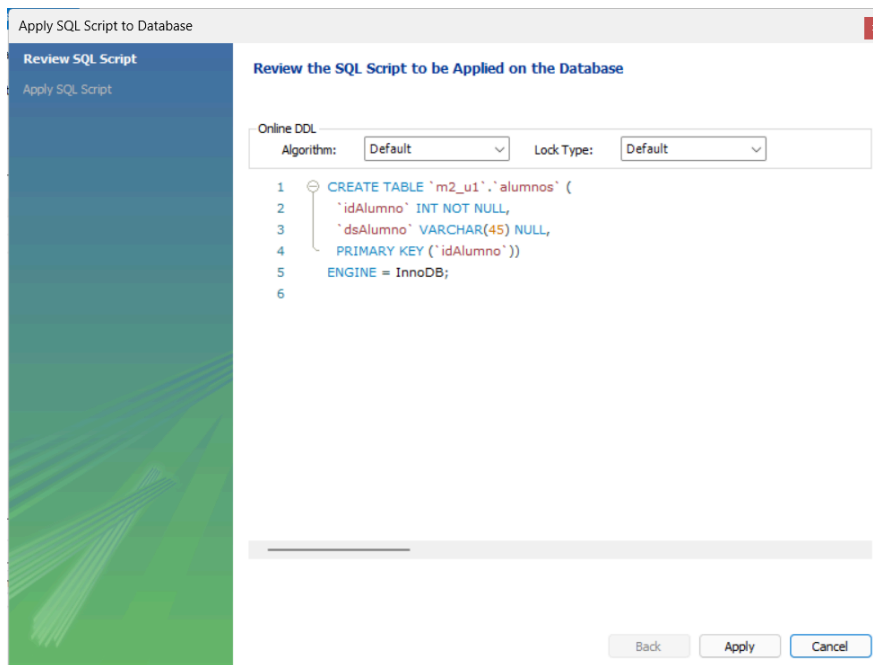
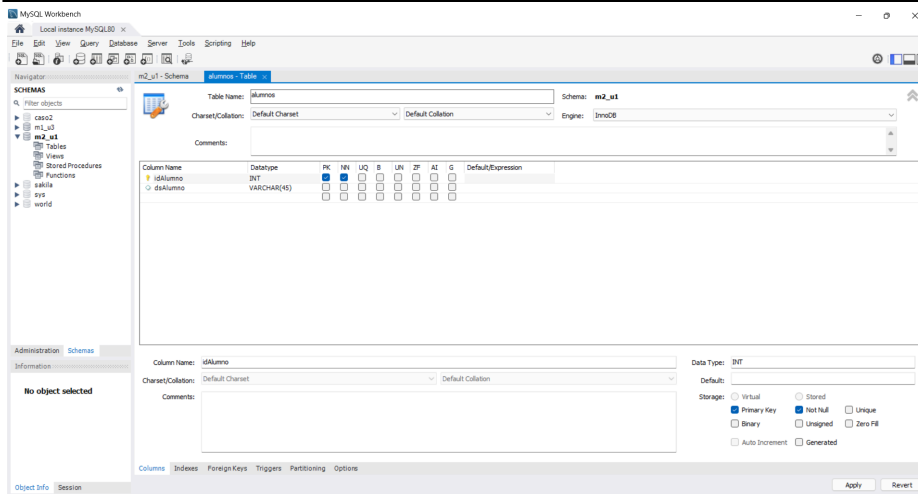


# Ejercicios Prácticos

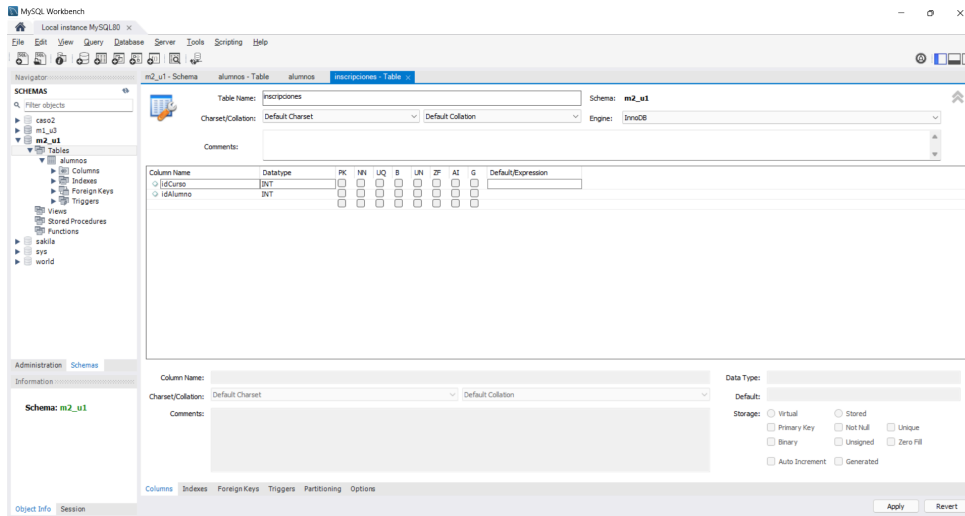
## Ejercicio Práctico 5.1

Cree una tabla Alumnos con un campo IdAlumno que sea autonumérico y otro campo DsAlumno con su nombre y apellido

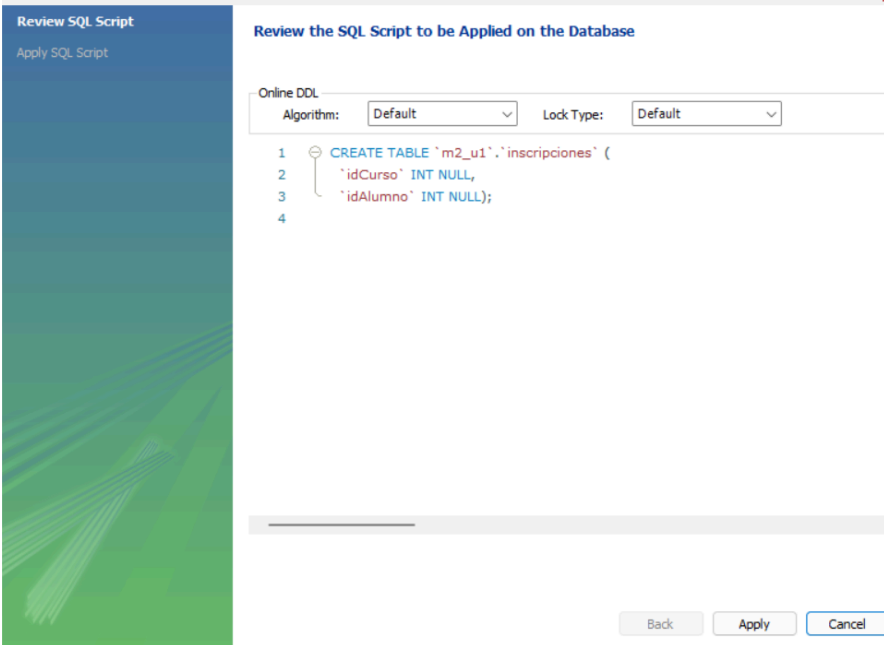


## Ejercicio Práctico 5.2

Cree una tabla que se llame Inscripciones en la cual tengamos el IdCurso y el IdAlumno

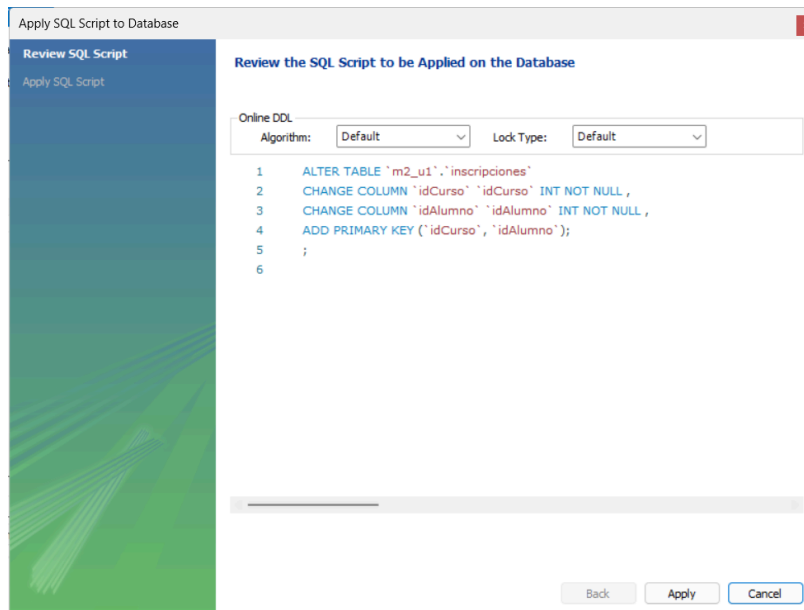
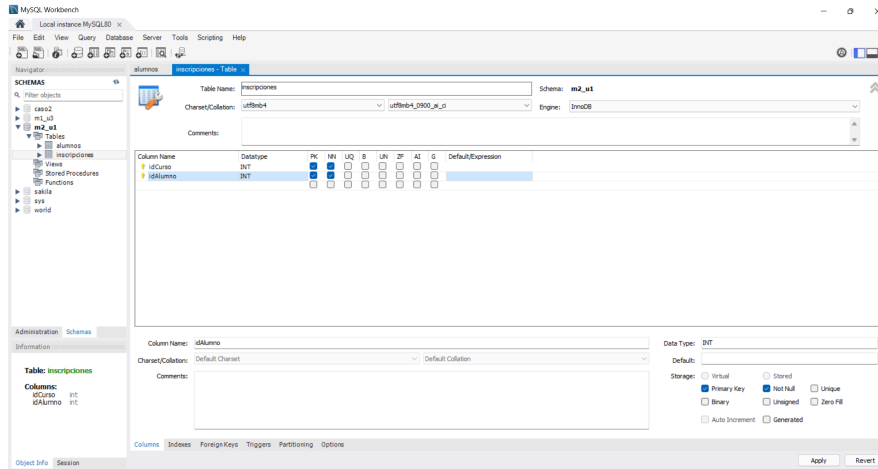


Apply SQL Script to Database



## Ejercicio Práctico 5.3

Cree una clave primaria compuesta para la tabla Inscripciones que incluya el campo IdCurso y el campo IdAlumno



## Ejercicio Práctico 5.4

Cargar tres alumnos que se llamen Mengano, Fulano y Sutano

The screenshot shows the MySQL Workbench interface. The top pane displays a query: `SELECT * FROM m2_u1.alumnos;`. The bottom pane shows the result set for the 'alumnos' table, which contains three rows:

idAlumno	dsAlumno
1	Mengano
2	Fulano
3	Sutano

Below the screenshot, a dialog box titled 'Apply SQL Script to Database' is shown. It contains a section 'Review the SQL Script to be Applied on the Database' with the following SQL script:

```
1 INSERT INTO 'm2_u1'.alumnos ('idAlumno', 'dsAlumno') VALUES ('1', 'Mengano');
2 INSERT INTO 'm2_u1'.alumnos ('idAlumno', 'dsAlumno') VALUES ('2', 'Fulano');
3 INSERT INTO 'm2_u1'.alumnos ('idAlumno', 'dsAlumno') VALUES ('3', 'Sutano');
4
```

At the bottom of the dialog, there are three buttons: 'Back', 'Apply', and 'Cancel'.

## Ejercicio Práctico 5.5

Escriba una consulta para obtener todos los alumnos y pruébela.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statement:

```
1 • SELECT * FROM m2_u1.alumnos;
```

The query has been executed, and the results are displayed in the Result Grid. The table has two columns: `idAlumno` and `dsAlumno`. The results are as follows:

idAlumno	dsAlumno
1	Mengano
2	Fulano
3	Sutano
NULL	NULL

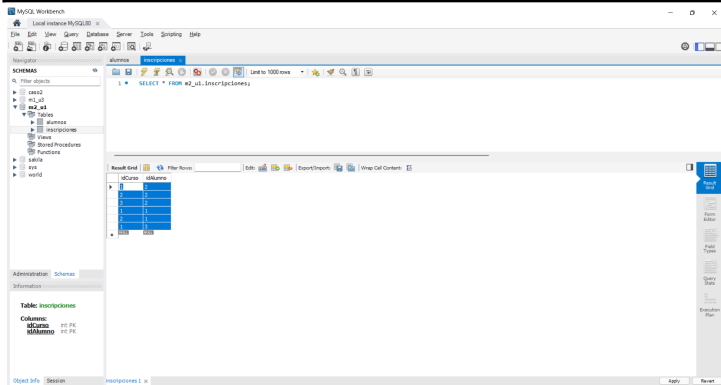
The left sidebar shows the Schemas pane with the following structure:

- caso2
  - m2\_u1
    - alumnos
    - inscripciones
    - views
    - stored procedures
    - functions
- sakila
- sys
- world

The bottom status bar shows the session information for the `alumnos 2` session.

## Ejercicio Práctico 5.6

Cargue seis registros en la tabla Inscripciones para que Fulano este tomando los tres cursos, Mengano sólo los dos primeros y Sutano sólo el primero de todos.



	idAlumno	dsAlumno
▶	1	Mengano
	2	Fulano
	3	Sutano
*	NULL	NULL

	idCurso	idAlumno
▶	1	1
	1	2
	1	3
	2	1
	2	2
	3	2
*	NULL	NULL

### Ejercicio Práctico 5.7

Cree y ejecute una consulta que indique cuántos alumnos hay en cada curso (va a tener que combinar GROUP BY y JOIN)

Se agregó la tabla cursos para una mejor utilidad de la consulta.

**TABLA cursos**

	idCurso	dsCurso
▶	1	Diplomatura en Bases de datos
	2	Diplomatura en Inteligencia de Negocios
	3	Diplomatura en Análisis de Negocios
•	NULL	NULL

**TABLA inscripciones**

	idCurso	idAlumno
▶	1	1
	2	1
	1	2
	2	2
	3	2
	1	3
•	NULL	NULL

```
SELECT dsCurso "Nombre del Curso", count(idAlumno) "Cantidad de Alumnos"  
FROM cursos  
JOIN inscripciones  
ON cursos.idCurso=inscripciones.idCurso  
GROUP BY dsCurso;
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

case2  
m1\_u3  
m2\_u1  
alumnos  
cursos  
inscripciones  
Views  
Stored Procedures  
Functions  
sakila  
sys  
world

ejercicios cursos inscripciones

Limit to 1000 rows

```
1 -- 5.7
2 • SELECT dsCurso "Nombre del Curso", count(idAlumno) "Cantidad de Alumnos"
3 FROM cursos
4 JOIN inscripciones
5 ON cursos.idCurso=inscripciones.idCurso
6 GROUP BY dsCurso;
```

Result Grid

Nombre del Curso	Cantidad de Alumnos
Diplomatura en Bases de datos	3
Diplomatura en Inteligencia de Negocios	2
Diplomatura en Análisis de Negocios	1

Administration Schemas

Information

Table: inscripciones

Columns:

- idCurso int PK
- idAlumno int PK

Object Info Session

Output

#	Time	Action	Message	Duration / Fetch
13	11:14:13	SELECT dsCurso, sum(idAlumno) FROM cursos JOIN inscripciones ON cursos.idCurso=inscripciones.idCurso ...	Error Code: 1054. Unknown column 'idAlumnos' in field list	0.000 sec
14	11:14:41	SELECT dsCurso, sum(idAlumno) FROM cursos JOIN inscripciones ON cursos.idCurso=inscripciones.idCurso G...	3 row(s) returned	0.000 sec / 0.000 sec
15	11:15:13	SELECT dsCurso "Nombre del Curso", sum(idAlumno) "Cantidad de Alumnos" FROM cursos JOIN inscripciones...	3 row(s) returned	0.000 sec / 0.000 sec
16	11:17:36	SELECT * FROM m2_u1.cursos LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
17	11:17:55	SELECT * FROM m2_u1.inscripciones LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
18	11:19:53	SELECT dsCurso "Nombre del Curso", count(idAlumno) "Cantidad de Alumnos" FROM cursos JOIN inscripciones...	3 row(s) returned	0.015 sec / 0.000 sec

SQL script saved to 'C:\Users\pablo\OneDrive\Escritorio\PABLO\CURSOS\Diplomatura\_Base\_Datos\MODULO 2\UNIDAD 1\EJERCICIOS\EJERCICIOS.sql'



### Ejercicio Práctico 5.8

Cargue valores para la tabla de inscripciones de manera que el primer alumno tenga asignados 3 cursos, el segundo 2 y el tercero 1. Si no ha cargado aún los alumnos hágalo.

Se cargaron los datos en las respectivas tablas:

- inscripciones: en el ejercicio 5.6

	idCurso	idAlumno
▶	1	1
	1	2
	1	3
	2	1
	2	2
	3	2
•	NULL	NULL

- alumnos: en el ejercicio 5.4

	idAlumno	dsAlumno
▶	1	Mengano
	2	Fulano
	3	Sutano
•	NULL	NULL

## Ejercicio Práctico 5.9

Vuelva a ejecutar el procedimiento almacenado para verificar que lista los cursos adecuadamente.

Se genera el procedimiento almacenado “**CursosxAlumno**” para ver en cuántos cursos está inscrito un alumno.

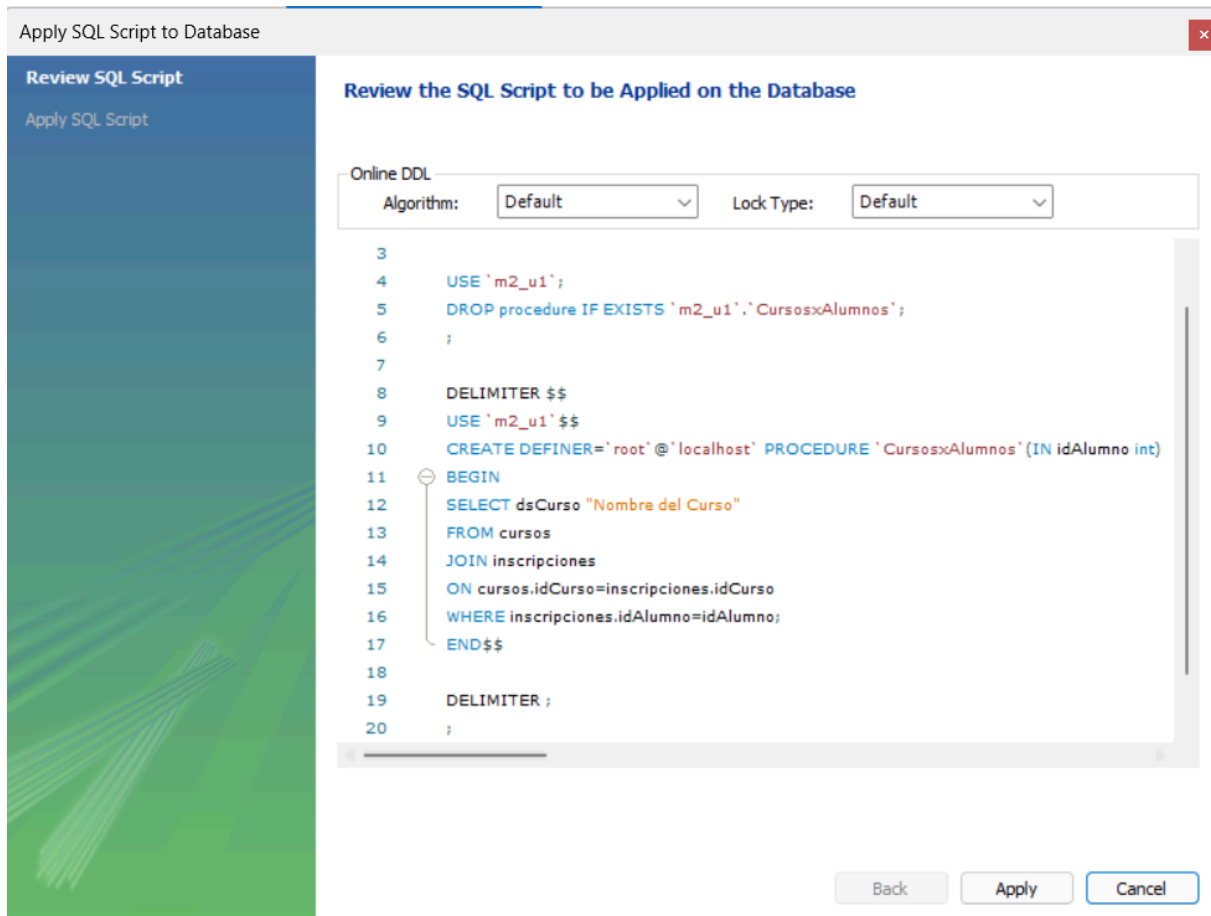
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SHEMAS' tree with 'm2\_u1' selected, showing 'Stored Procedures' and 'CursosxAlumnos'. The main editor shows the DDL for the procedure 'CursosxAlumnos'.

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `CursosxAlumnos` (IN idAlumno int)
2 BEGIN
3 SELECT dsCurso "Nombre del Curso" FROM cursos JOIN inscripciones ON cursos.idCurso=inscripciones.idCurso WHERE inscripciones.idAlumno=idAlumno;
4 END
```

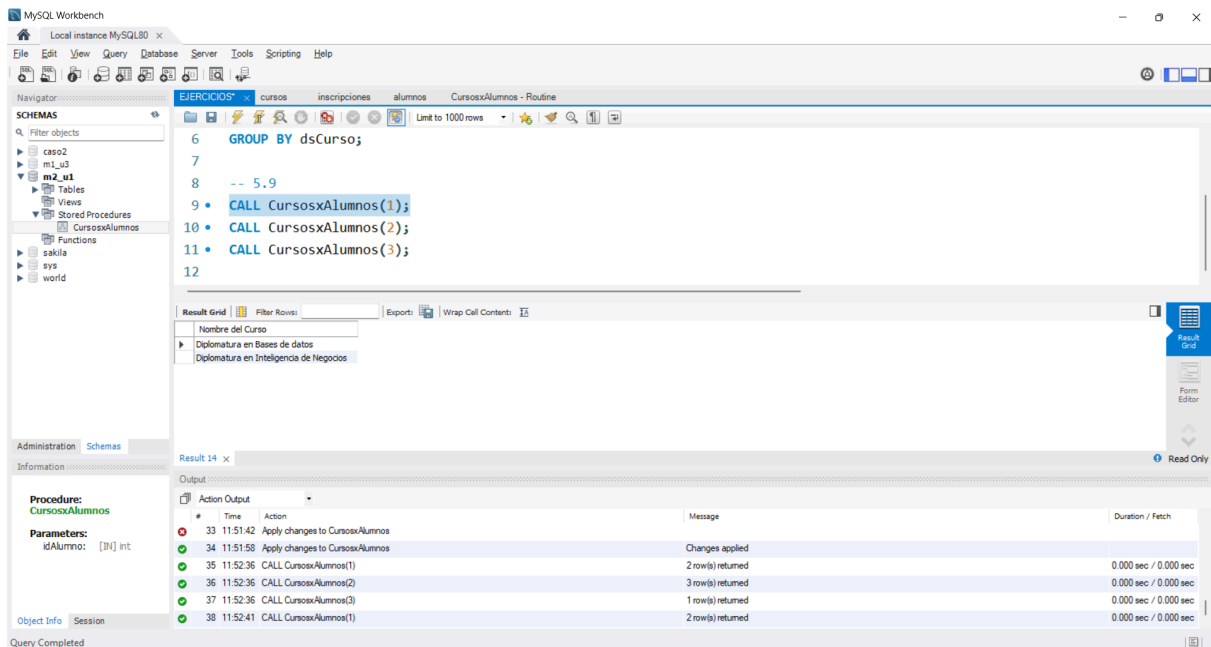
The 'Routine' tab at the bottom shows the execution output:

#	Time	Action	Message	Duration / Fetch
24	11:49:30	CALL CursosxAlumnos(1)	6 row(s) returned	0.016 sec / 0.000 sec
25	11:50:34	Apply changes to CursosxAlumnos	Changes applied	
26	11:50:42	CALL CursosxAlumnos(1)	2 row(s) returned	0.000 sec / 0.000 sec
27	11:50:49	CALL CursosxAlumnos(2)	3 row(s) returned	0.000 sec / 0.000 sec
28	11:50:59	SELECT dsCurso "Nombre del Curso", count(idAlumno) "Cantidad de Alumnos" FROM cursos JOIN inscripciones ON cursos.idCurso=inscripciones.idCurso WHERE inscripciones.idAlumno=idAlumno	3 row(s) returned	0.000 sec / 0.000 sec
29	11:50:59	CALL CursosxAlumnos(3)	1 row(s) returned	0.000 sec / 0.000 sec

The 'Parameters' section shows 'idAlumno: [IN] int'.



Se ejecuta el procedimiento almacenada para el idAlumno=1:



Nombre del Curso
Diplomatura en Bases de datos
Diplomatura en Inteligencia de Negocios

## TABLA inscripciones

	idCurso	idAlumno
▶	1	1
	2	1
	1	2
	2	2
	3	2
	1	3
	NULL	NULL

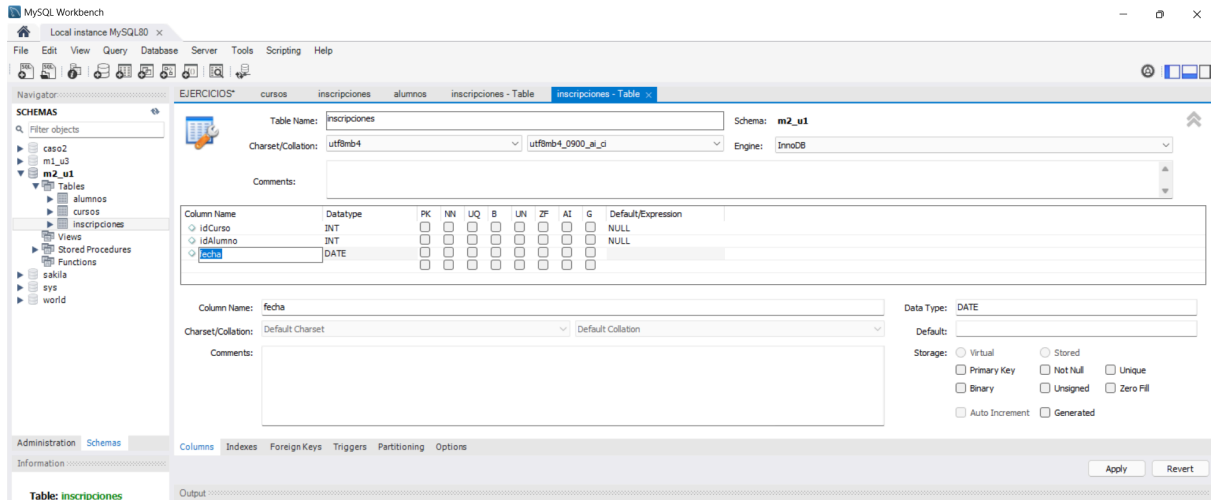
## TABLA cursos

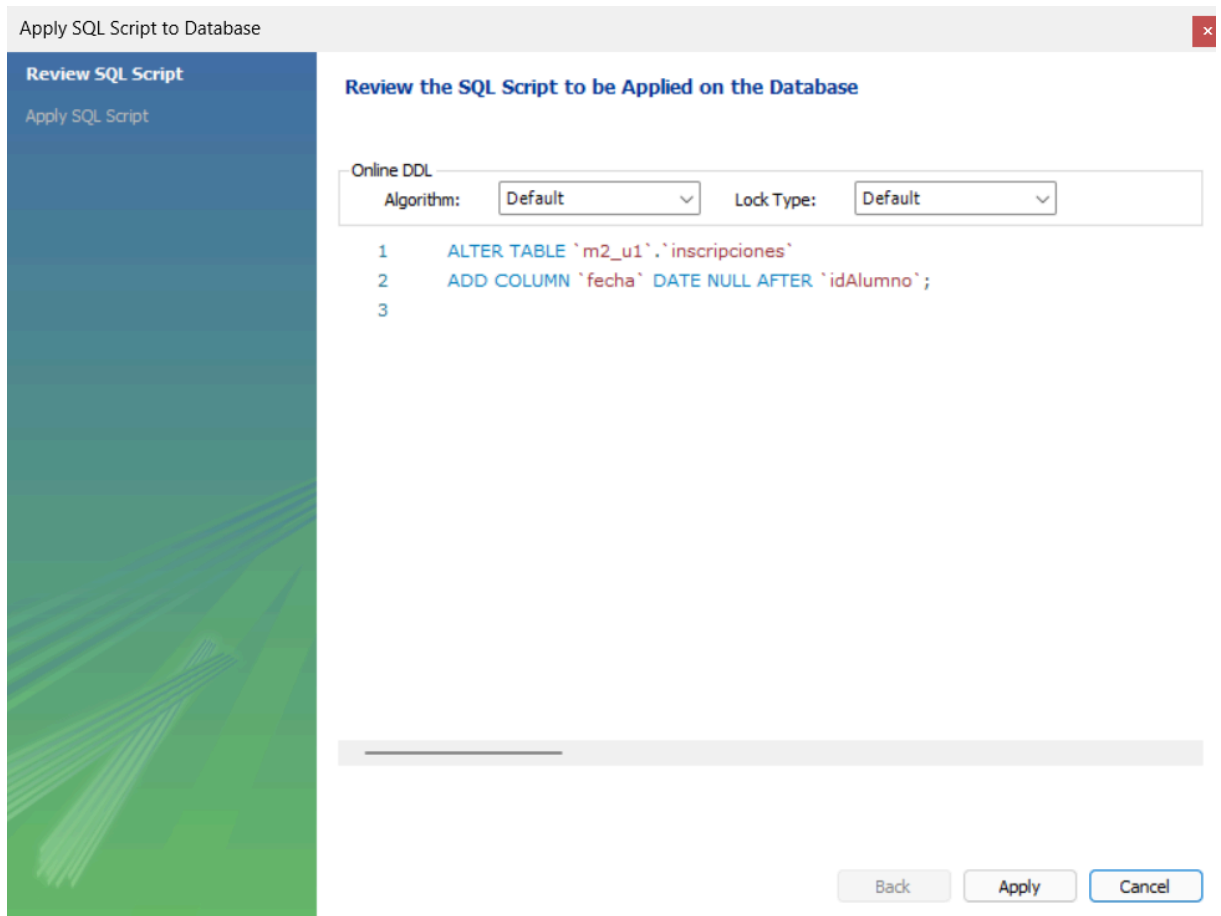
	idCurso	dsCurso
▶	1	Diplomatura en Bases de datos
	2	Diplomatura en Inteligencia de Negocios
	3	Diplomatura en Análisis de Negocios
•	NULL	NULL

## Ejercicio Práctico 5.10

Cargue las fechas de inscripción y vuelva a ejecutar la función para verificar que devuelva el resultado esperado.

Se modifica la tabla inscripciones para agregar la columna fecha, quedando de la siguiente manera:

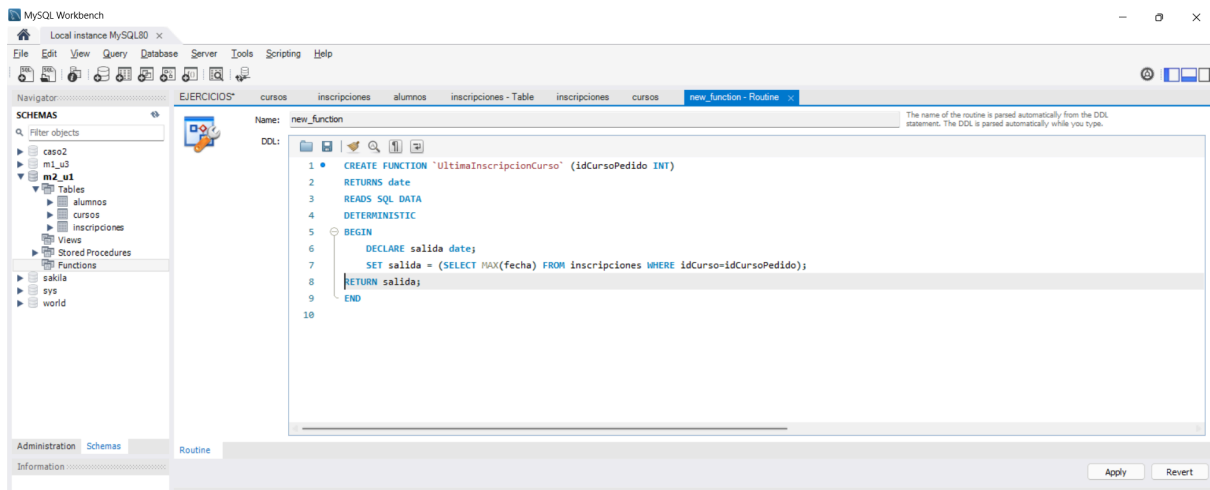




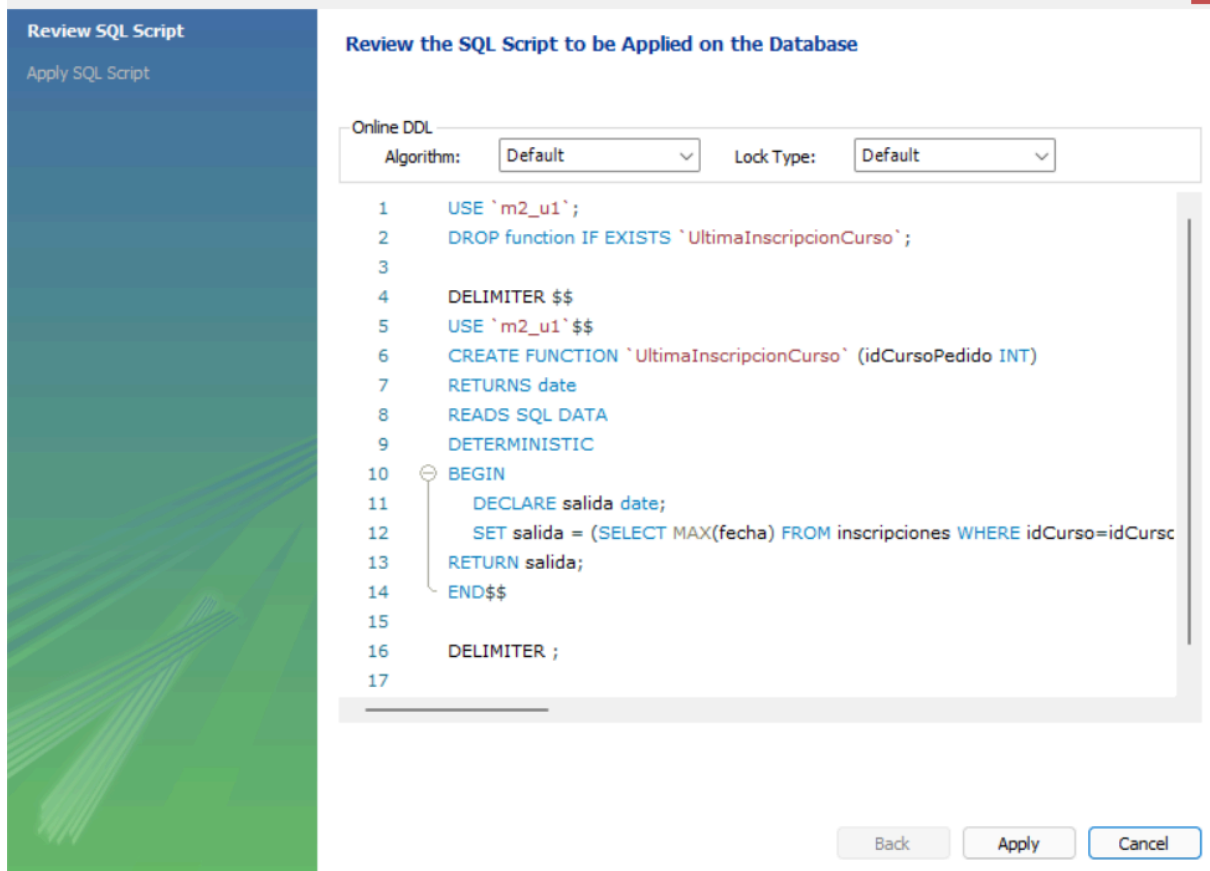
Se agregar las fechas:

	idCurso	idAlumno	fecha
▶	2	1	2020-01-05
	2	2	2023-01-10
	3	2	2025-03-05

Se genera la función:



Apply SQL Script to Database





MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

caso2

m1\_u3

m2\_u1

alumnos

inscripciones

Views

Stored Procedures

Functions

UltimaInscripcionCurso

sakila

sys

world

Administration Schemas

Information

EJERCICIOS

Limit to 1000 rows

```
13 -- 5.10
14 UPDATE inscripciones SET fecha='2020-01-05' WHERE idCurso =2 AND idAlumno=1;
15 UPDATE inscripciones SET fecha='2023-01-10' WHERE idCurso =2 AND idAlumno=2;
16 UPDATE inscripciones SET fecha='2025-03-05' WHERE idCurso =3 AND idAlumno=2;
17 SELECT UltimaInscripcionCurso(3);
18
```

Result Grid

UltimaInscripcionCurso(3)

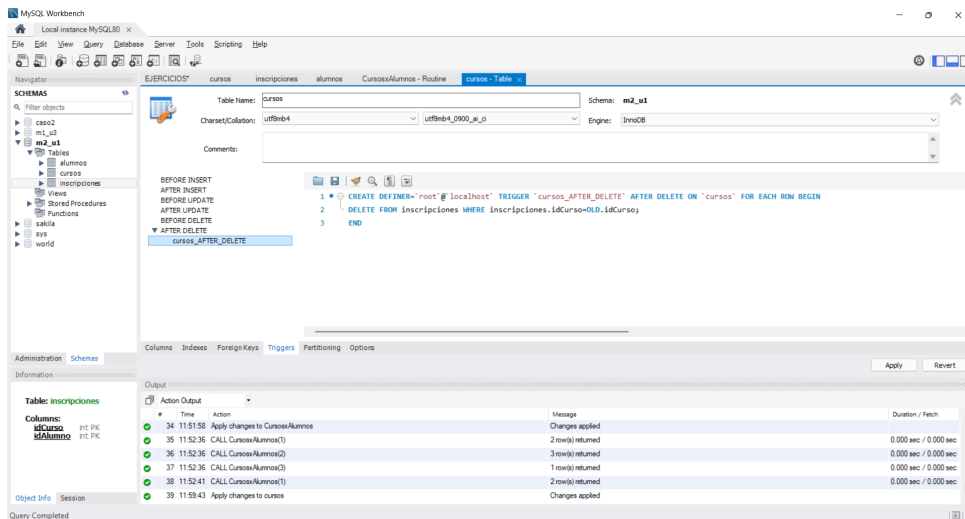
2025-03-05
------------

Result 7 x

Read Only

## EXTRA:

Se genera el TRIGGER en la tabla cursos para eliminar los registros de la tabla inscripciones después de eliminar un curso:



### Antes de eliminar el curso 1:

- `SELECT * FROM cursos;`

	idCurso	dsCurso
▶	1	Diplomatura en Bases de datos
	2	Diplomatura en Inteligencia de Negocios
	3	Diplomatura en Análisis de Negocios
✱	NULL	NULL

- `SELECT * FROM inscripciones;`

	idCurso	idAlumno
▶	1	1
	2	1
	1	2
	2	2
	3	2
	1	3
✱	NULL	NULL

### Eliminamos el curso 1:

- `DELETE FROM cursos WHERE idCurso=1;`

Después de eliminar el curso 1:

- `SELECT * FROM cursos;`

	idCurso	dsCurso
▶	2	Diplomatura en Inteligencia de Negocios
	3	Diplomatura en Análisis de Negocios
•	NULL	NULL

- `SELECT * FROM inscripciones;`

	idCurso	idAlumno
▶	2	1
	2	2
	3	2