

Bases de Datos - 1º DAM

UT 2 – Práctica 3. BBDD de temática libre (relación N:M)

En esta práctica vamos a trabajar las relaciones muchos a muchos. En esta ocasión elige tu la temática de la información que quieras almacenar.

1. Piensa en un ejemplo de relación de muchos a muchos. Por ejemplo:

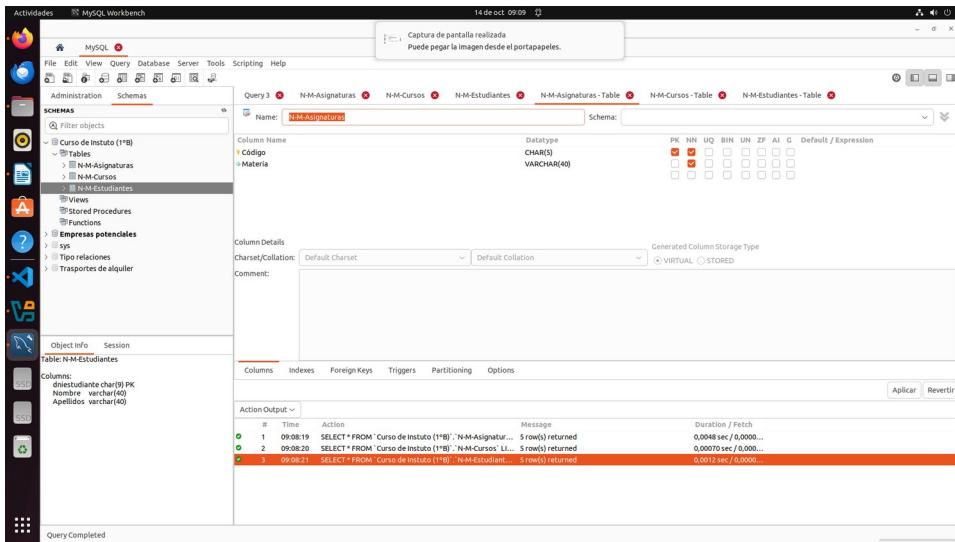
- a) Festivales de música y asistentes.
- b) Personas y aficiones.
- c) Usuarios y grupos de Whatsapp
- d) **Estudiantes y asignaturas que cursan.**
- e) Atletas y carreras en las que participan.
- f) Trabajadores y proyectos en los que trabajan.

2. Crea una nueva base de datos con un nombre apropiado a la temática elegida.

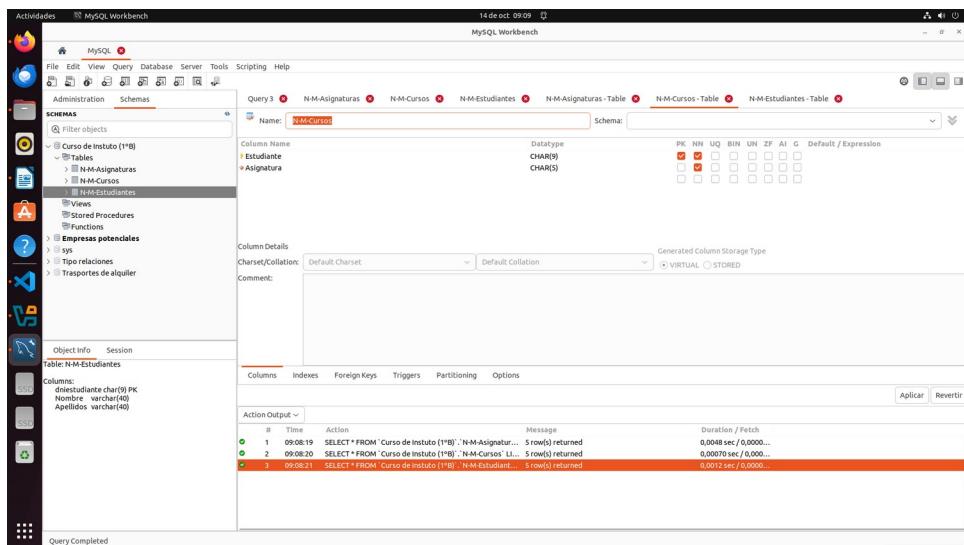
The screenshot shows the MySQL Schema Editor interface. At the top, there is a toolbar with icons for creating a database, table, view, and stored procedure. Below the toolbar, the title "Schema Editor" is displayed next to a database icon. The main area is titled "Specify the name of the schema here. You can use any combination of ANY character for names that don't require quoting. For more flexibility you can use Unicode characters in the name. You can also use the Backslash Escaped Name pane (BMP), but you will have to quote the name later when you refer to it." A "Schema Name:" label is followed by a text input field containing "Curso de Instituto (1ºB)". To the right of the input field is a "Rename References" button. Below this, a note states: "The character set and its collation selected here will be used when no other character set or collation is explicitly specified (it uses the DEFAULT value then). Setting DEFAULT here will make the schema to use the same character set and collation as the current connection." Under "Character set:", a dropdown menu is open with "Default Charset" selected. Under "Collation:", another dropdown menu is open with "Default Collation" selected.

Primeramente escogí el tema de estudiantes y asignaturas que cursan y lo nombre como **Curso de Instituto (1ºB)**.

3. Crea las tablas que necesites en la base de datos .



Cree una tabla de asignaturas donde puse las columnas de **código** (para poder identificar la materia) y **materia** (que tipo de materia o como se llama la materia), luego establecí como llave principal el código, un elemento identifiable que es **char (5)**.



La tabla **Curso**, sirve para relacionar dos tablas: la tabla de **asignaturas** y la tabla de **estudiantes** donde la clave principal es estudiante, sin embargo podía haber sido asignatura, ya que los dos son identificadores reconocibles dentro de la tabla.

MySQL Workbench Schemas

Table: N-M-Estudiantes

Columns:

- idestudiante (PK)
- Nombre
- Apellidos

Action Output:

- 1 09:08:19 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Asignatur...' 5 row(s) returned 0.0048 sec / 0.0000...
- 2 09:08:20 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Cursos' L... 5 row(s) returned 0.00070 sec / 0.000...
- 3 09:08:21 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Estudiant...' 5 row(s) returned 0.0012 sec / 0.0000...

Y por último, la tabla **estudiantes** sirve para reconocer a cada estudiante, con la clave principal **dniestudiante** y los **nombres** y **apellidos**, declarando todo como NN, obligando así a no dejar nada sin poner.

4. Relaciona las tablas mediante algún tipo de relación N:M.

MySQL Workbench Schemas

Table: N-M-Estudiantes

Foreign Key Name: Fk_Cur_Estu

Referenced Table: N-M-Estudiantes

Column: Estudiante

Referenced Column: Código

On Update: RESTRICT

On Delete: RESTRICT

Action Output:

- 1 09:08:19 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Asignatur...' 5 row(s) returned 0.0048 sec / 0.0000...
- 2 09:08:20 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Cursos' L... 5 row(s) returned 0.00070 sec / 0.000...
- 3 09:08:21 SELECT * FROM 'Curso de Instituto (1*)'. 'N-M-Estudiant...' 5 row(s) returned 0.0012 sec / 0.0000...

En la tabla de **cursos** que sirve para enlazar las otras dos tablas, relacioné la tabla **cursos** con la de asignatura con la clave principal **Código (Asignatura)**.

Query 3 - N-M-Asignaturas - N-M-Cursos - N-M-Estudantes - N-M-Asignaturas - Table - N-M-Cursos - Table - N-M-Estudantes - Table

Foreign Key Name: fk_Cur_Asig
Referenced Table: N-M-Asignaturas
Foreign Key Columns: Cur
Referenced Columns: idasignatura
On Update: RESTRICT
On Delete: RESTRICT
Foreign Key Comment:

Action Output:

#	Time	Action	Message	Duration / Fetch
1	09:08:19	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Asignatura...	5 rows(s) returned	0,0048 sec / 0,0000...
2	09:08:20	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Cursos ...	5 rows(s) returned	0,00070 sec / 0,0000...
3	09:08:21	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Estudant...	5 rows(s) returned	0,0018 sec / 0,0000...

En la tabla de **cursos** que sirve para enlazar las otras dos tablas, relacioné la tabla **cursos** con la de **Estudiantes** con la clave principal **dniestudiante** (**Estudiantes**).

5. Inserta al menos cinco filas en cada tabla.

Query 3 - N-M-Asignaturas - N-M-Cursos - N-M-Estudantes

Result Grid:

#	Código	Asignatura
1	123AA	Inglés
2	123AB	Música
3	123AC	Alemán
4	123AD	TIC
5	123AE	Desarrollo de electroencefalogramas

Action Output:

#	Time	Action	Message	Duration / Fetch
1	09:08:19	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Asignatura...	5 rows(s) returned	0,0048 sec / 0,0000...
2	09:08:20	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Cursos ...	5 rows(s) returned	0,00070 sec / 0,0000...
3	09:08:21	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Estudant...	5 rows(s) returned	0,0012 sec / 0,0000...

En la tabla de **Asignaturas**, puse 5 ejemplo con su código y materia. Y en la tabla de **Cursos** establecí los datos relacionados de **Asignatura** y **Estudiantes**.

Query 3 - N-M-Asignaturas - N-M-Cursos - N-M-Estudantes

Result Grid:

#	Estudiante	Asignatura
1	16415879C	123AA
2	62135772F	123AB
3	74599412Q	123AC
4	79829850R	123AD
5	83M	123AE

Action Output:

#	Time	Action	Message	Duration / Fetch
1	09:08:19	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Asignatura...	5 rows(s) returned	0,0048 sec / 0,0000...
2	09:08:20	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Cursos ...	5 rows(s) returned	0,00070 sec / 0,0000...
3	09:08:21	SELECT * FROM 'Curso de Instituto (1*)'; N-M-Estudant...	5 rows(s) returned	0,0012 sec / 0,0000...

Actividades MySQL Workbench

14 de oct 09:09

Captura de pantalla realizada
Puede pegar la imagen desde el portapapeles.

MySQL

File Edit View Query Database Server Tools Scripting Help

Schemas Administration Schemas

Filter objects

Curso de Instituto (1ºB)

Tables N-M-Asignaturas N-M-Cursos N-M-Estudiantes

N-M-Estudiantes

Views

Stored Procedures

Functions

Empresas potenciales

sys

Tipo relaciones

Transportes de alquiler

Object Info Session

Table: N-M-Cursos

Columns:

Estudante char(9) PK
Asignatura char(5)

Result Grid Filter Rows

Limit to 1000 rows

Query 3 N-M-Asignaturas N-M-Cursos N-M-Estudiantes

1 • SELECT * FROM `Curso de Instituto (1ºB)`.'N-M-Estudiantes';

#	dniestudiante	Nombre	Apellidos
1	06705830L	Román	Quinto
2	16415979C	Felipe	Molina
3	89000012F	Juanjo	Peral
4	74599412Q	Eustaquio	Parmesano
5	7982950R	Carlos	González

Action Output

#	Time	Action	Message	Duration / Fetch
1	09:08:19	SELECT * FROM `Curso de Instituto (1ºB)`.'N-M-Asignatur...	5 row(s) returned	0.0048 sec / 0.0000...
2	09:08:20	SELECT * FROM `Curso de Instituto (1ºB)`.'N-M-Cursos' LI...	5 row(s) returned	0.00070 sec / 0.000...
3	09:08:21	SELECT * FROM `Curso de Instituto (1ºB)`.'N-M-Estudiant...	5 row(s) returned	0.0012 sec / 0.0000...

Result Grid Filter Rows

Export/Import

Wrap Cell Content

Revert

Query Completed

En estudiante puse el **dniestudiante** como enlace a la tabla de cursos y **nombre** y **apellidos**.