

# Desarrollo completo de una aplicación utilizando Base de Datos MySql.

## Parte 1: Diseñar el Modelo de Datos

En esta primera fase del proyecto se debe **diseñar el modelo entidad-relación** que representará la estructura lógica de la base de datos.

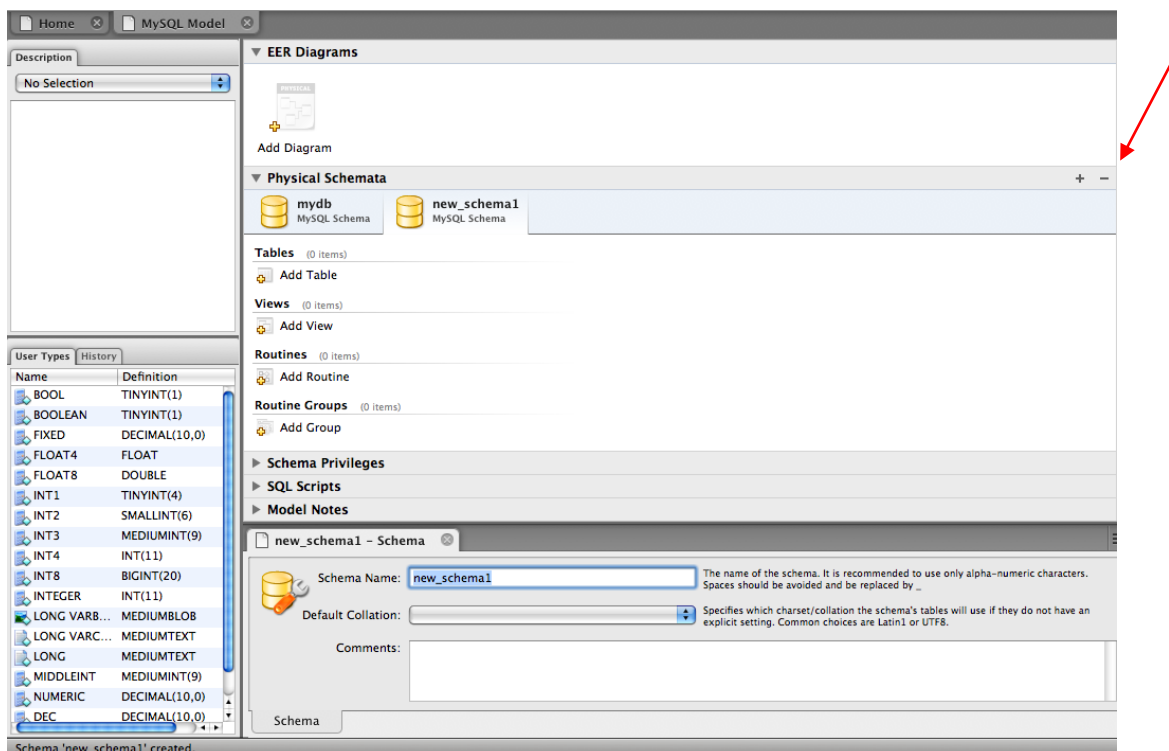
El desarrollo del modelo se realizará utilizando **MySQL 8**, disponible en: <https://dev.mysql.com/downloads/workbench/>

Alternativamente, se podrá utilizar el motor **PostgreSQL**, según la disponibilidad y preferencia de cada uno.

El **modelo de datos** se elaborará mediante la herramienta **MySQL Workbench**, la cual permite definir entidades, atributos y relaciones, así como aplicar procesos de **Ingeniería Directa (Forward Engineering)** para generar automáticamente el esquema físico de la base de datos y **sincronizarlo** con un servidor MySQL.

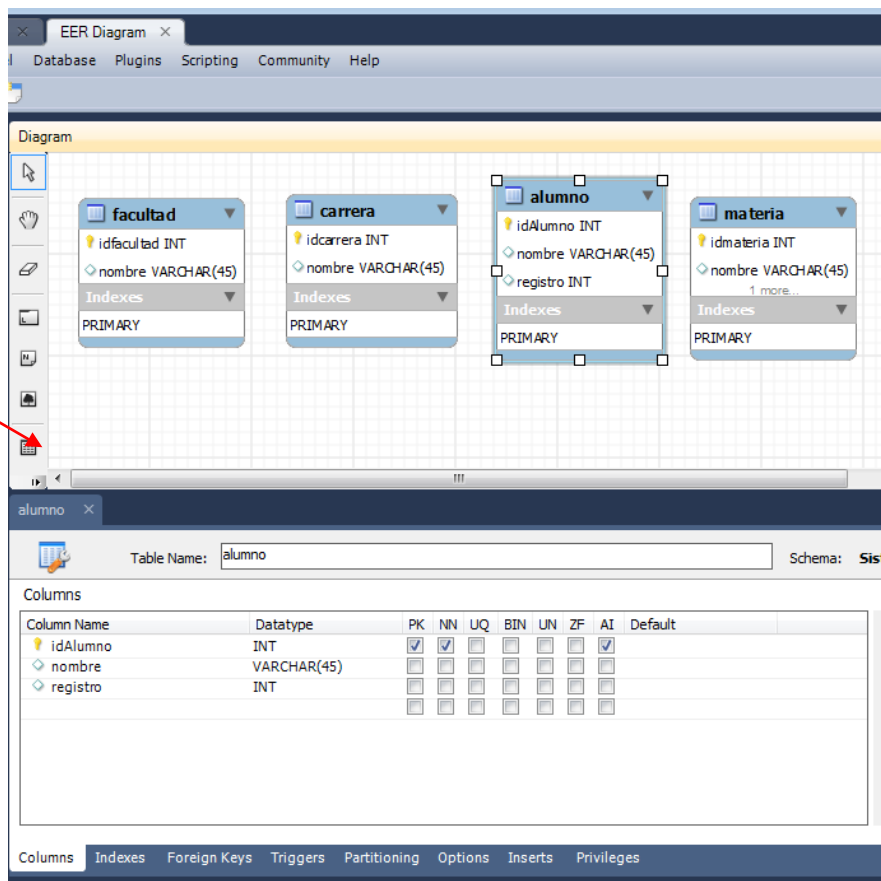
### 1. Crear las tablas:

En primera instancia debemos crear un nuevo esquema, en este caso **bdalumnos**




Luego se debe crear cada tabla del sistema del tipo **InnoDB** por que soportan claves foráneas. Por cada entidad se especifica su nombre y sus atributos con los valores que corresponda, por ej:

Columna	Tipo	PK (Clave Primaria)	NN (No Nulo)	UN (Sin signo)	AI (Autoincremento)
id	INT	✓	✓	✓	✓

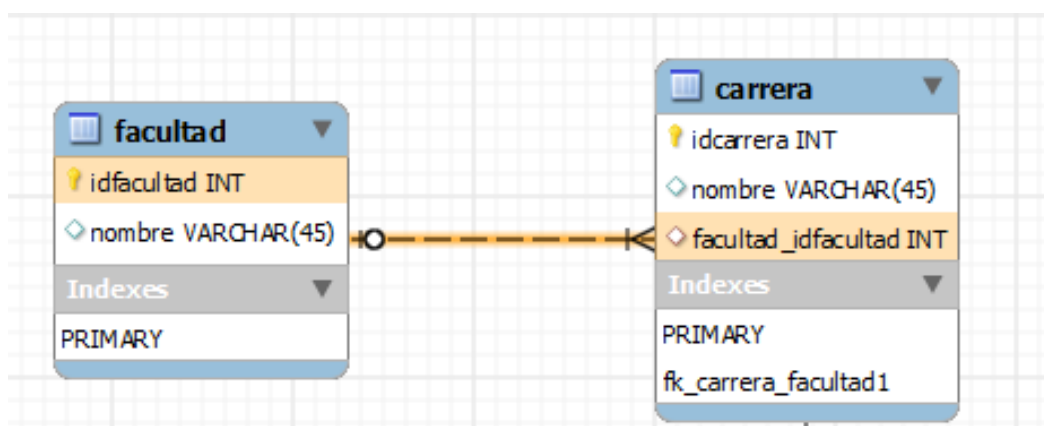


## 2. Establecer y crear los vínculos entre tablas:

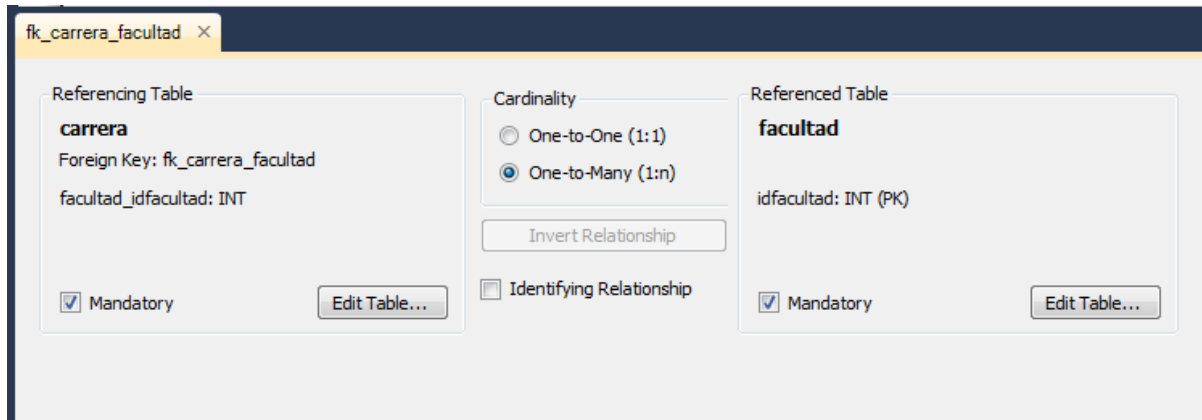
- ✓ Si analizamos las tablas **facultad** y **carreras** se deduce un vínculo de uno-muchos:
  - Una facultad tiene varias carreras y
  - Una carrera pertenece solo a una facultad.

El botón  (1:n Non-Identifying Relationship // El atributo de la clave primaria del padre no se convierte en atributo de clave primaria del hijo.) permite crear este vínculo: Se hace click sobre este botón y luego primero sobre la tabla que contendrá la clave foránea (carrera) que referencia a la segunda tabla (facultad).

La clave foránea se agrega como una nueva columna en la tabla.




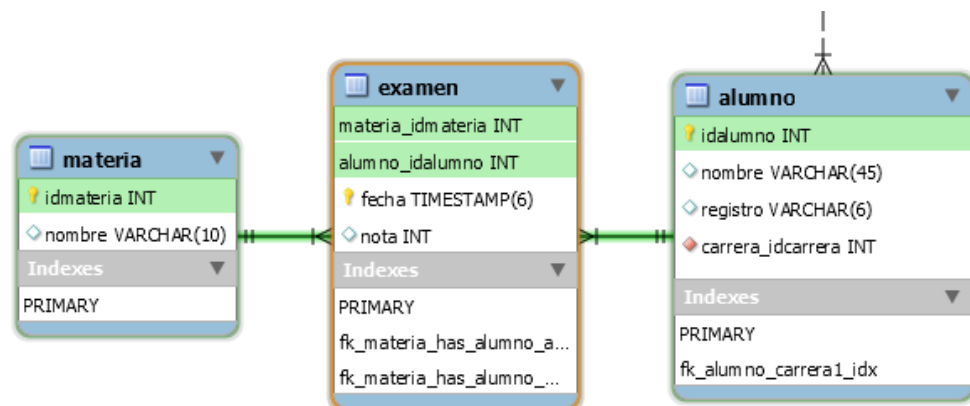
Puede editarse esta relación haciendo doble click sobre la relación:



Aquí puede modificarse la relación teniendo en cuenta que:

- **cardinalidad:** indica si la relación entre dos entidades es uno-uno o uno-muchos.
  - **ordinalidad:** si la referencia entre entidades debe existir para mantener la integridad del modelo. (Tildar el checkbox Mandatory en ambos lados)
  - **tipo:** (*identifying* o *non-identifying*). Una relación *non-identifying*, refiere a que los hijos (carreras) se pueden identificar independientes del padre (facultad). Una relación *identifying* significa que los hijos no pueden ser unívocamente identificados sin el padre y en este caso facultad\_idfacultad se definiría también como clave primaria.
- ✓ Si analizamos las entidades **alumno** y **materia**, donde se quiere representar la relación de materias rendidas, se deduce entonces una relación mucho-mucho:
- Un alumno puede tener rendidas varias materias y
  - Una materia puede haber sido rendida por varios alumnos.

Se crea eligiendo el botón  (n:m Identifying Relationship) y luego se hace click en las entidades.



Para implementar la relación mucho-mucho en la base de datos, es necesario crear una tercera tabla (alumno\_has\_materia renombrada como **examen**) que contenga las claves primarias de las dos tablas originales.

Dado que un alumno puede rendir una materia varias veces es necesario agregar el atributo **fecha** como parte de la clave primaria (de tipo [TIMESTAMP](#)). **Agregar además el atributo *nota*.**

alumno\_has\_materia

Table Name:  Schema:

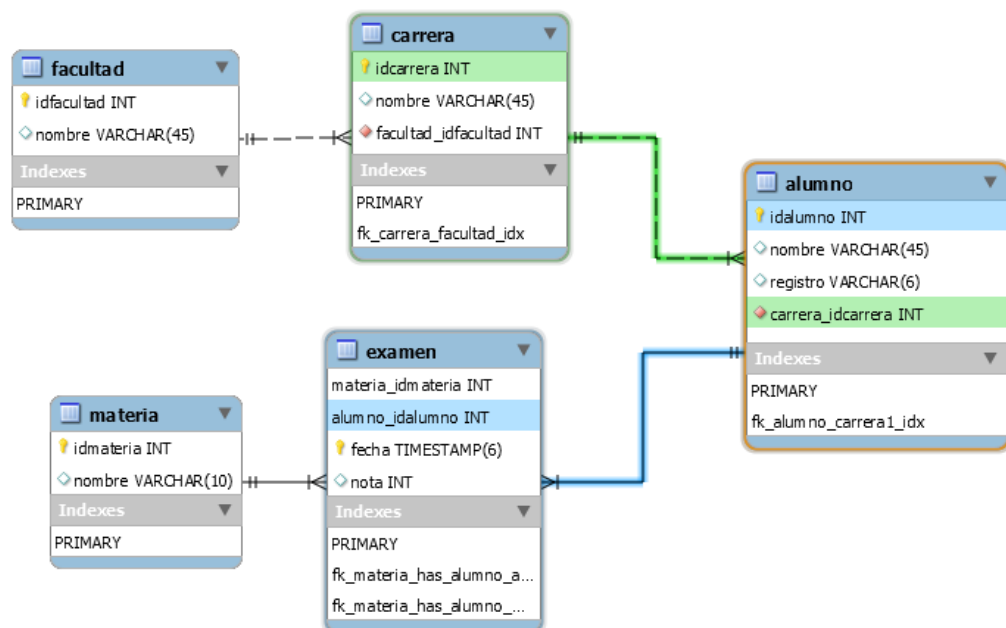
Columns

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
alumno_idAlumno	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
materia_idmateria	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha	TIMESTAMP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

- ✓ Las entidades **carrera** y **alumno** se definen como una relación de uno-muchos, para simplificar el modelo.

De esta forma queda completo el diagrama entidad-relación ERD. Este diagrama representa el modelo de datos del **Sistema Alumnos**.

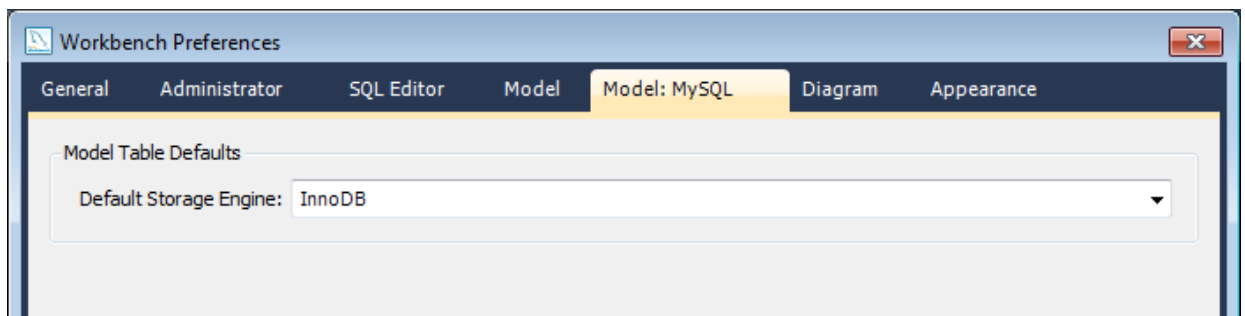
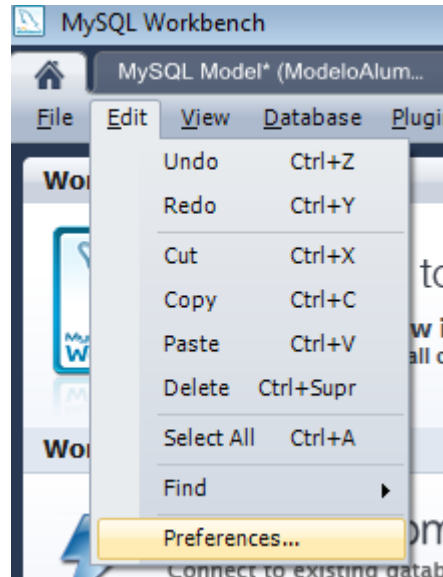


## Forward-Engineering hacia la Base de Datos

Para generar el esquema en MySQL y las tablas correspondientes al modelo definido anteriormente, se aplica **forward-engineer** usando el asistente de Workbench.

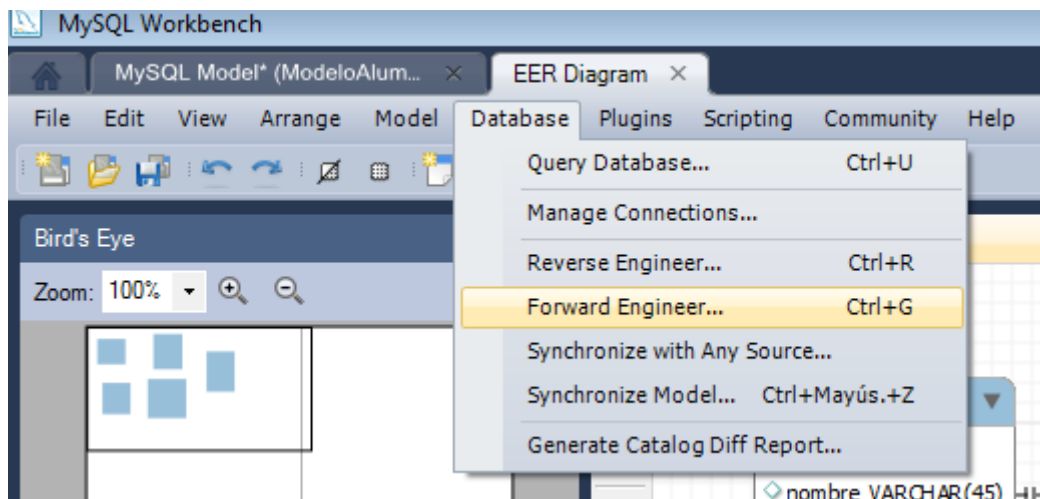
### Importante:

- El servidor de MySQL debe estar corriendo.
- Configurar el **default storage engine** usado en Workbench como **InnoDB**.

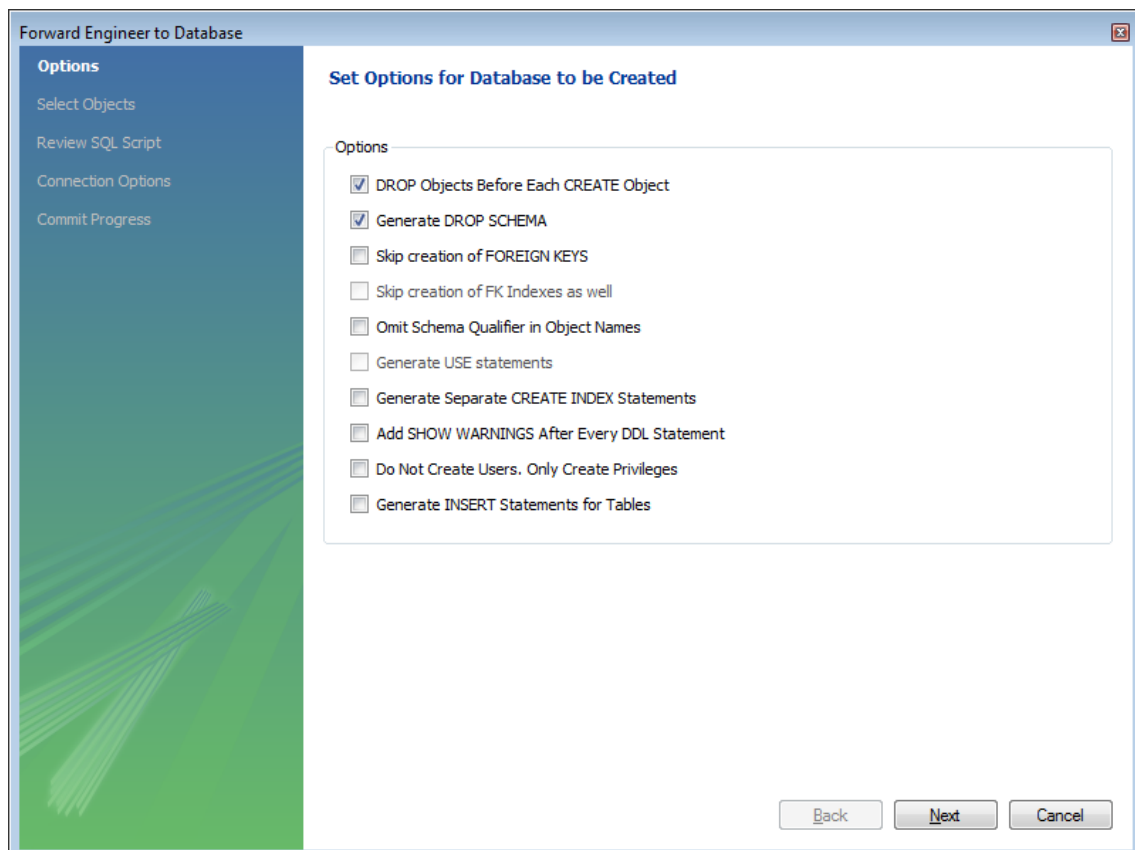


Una vez configurada la herramienta se inicia el proceso:

1)



2) En esta ventana tildar las opciones para borrar el esquema si existiera, para evitar que el servidor de errores.

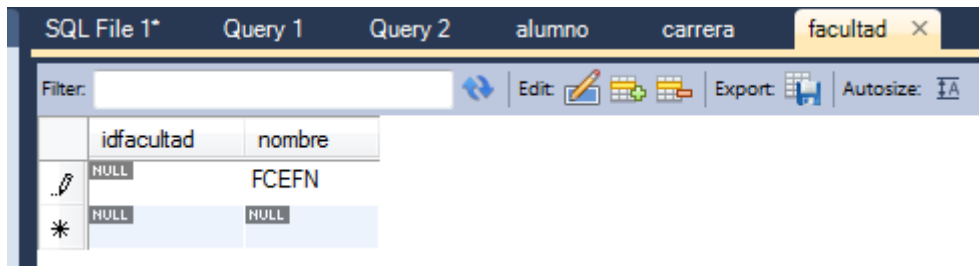


- 3) En esta ventana aparece, por defecto, tildada la opción de exportar todas las tablas.
- 4) En esta ventana se muestra el script SQL que se ha generado en base al modelo de datos. Aparece la opción de grabar el script.
- 5) Aquí se deben configurar los parámetros para conectarse al servidor de MySQL.

**Hostname:** localhost (o 127.0.0.1)  
**Port:** 3306  
**Username:** root  
**Password:** ...

6) Click **Execute**.

7) Cerrar el asistente y luego abrir la base de datos para agregar algunos datos (a **Facultad**, **Carrera** y **Alumno**), por ej:



	idfacultad	nombre
	NULL	FCEFN
*	NULL	NULL

Ayuda sobre sintaxis el lenguaje SQL:

<https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html>

```
INSERT INTO `bdalumnos`.`materia_has_alumno` (`materia_idmateria`,  
`alumno_idalumno`, `fecha`) VALUES (1, 2, 'CURRENT_TIMESTAMP());
```

```
INSERT INTO table_name(today)  
VALUES (STR_TO_DATE('07-25-2012', '%m-%d-%y'));
```

```
SELECT DATE_FORMAT(today, '%m-%d-%y') from table_name;
```

## Parte 2: Generar el Proyecto Web

---

### 1) Revisión teórica:

Revisar los conceptos teóricos relacionados con la estructura de un proyecto web Java EE con persistencia y seguir los pasos recomendados.

### 2) Configuración de la conexión a la base de datos:

Para establecer la conexión con MySQL, es necesario descargar e incorporar el controlador JDBC: mysql-connector-java-8.0.19.jar.

Atención: Si aparece una "falta en la conexión" revisar el JDBC URL:  
jdbc:mysql://localhost:3306/comercio?zeroDateTimeBehavior=CONVERT\_TO\_NULL&serverTimezone=UTC

### 3) Proyecto Web:

Generar un nuevo proyecto **AppAlumno**, que integrará el modelo entidad-relación previamente implementado en MySQL. Adaptar el archivo index.jsp del ejemplo visto en clase.

### 4) Creación de las clases de Entidad (modelo OO):

Generar las clases Entidades que representen las tablas del modelo de datos.

### 5) Gestión de entidades principales:

Implementar las vistas y formularios necesarios para registrar y consultar datos de las entidades principales del sistema. Esto incluye dotar de funcionalidad a las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) de Alumno, Materia y Carrera.

**IMPORTANTE:** Dejar este ejercicio funcionando y bien probado, porque en la próxima clase vamos a continuar trabajando a partir de él. Si algo queda incompleto o con errores, te va a resultar más difícil seguir el desarrollo. Asegúrate de guardar todos los archivos y verificar que el programa se ejecute correctamente.