

BASE DE DATOS

UNED



Contenido

Introducción	2
Creación Base de Datos	3
Creación de tablas	3
Alumnos	4
País	5
Carrera.....	5
Matrícula.....	6
Resultado tablas sin inserción	7
Inserción en tablas.....	8
Inserción tabla alumnos	8
Inserción tabla país.....	9
Inserción tabla carrera	10
Inserción tabla matrícula.....	11
Consultas SQL	12
¿Qué carrera estudia x alumno?	12
¿Cuántos alumnos estudian x carrera?	13
¿Cuántos alumnos son de x país?	14
¿Cuántos alumnos de x país estudian x carrera?	15

Introducción

Esta es una base de datos creada para una asignatura de un curso de la UNED (Universidad Nacional de Educación a Distancia) con el fin de poder realizar una prueba de como unos cuantos registros pueden ser de gran ayuda y como sacar provecho gracias a las bases de datos y las consultas que se hacen sobre las mismas, mostrando así la cantidad de resultados que puedes obtener debido a su simpleza y eficiencia.

En esta base de datos la cual se llama escuela repasaremos paso a paso todo el proceso realizado, desde la creación de la base de datos, pasando por la creación de tablas e inserción de los datos hasta las consultas realizadas sobre las tablas para poder obtener los resultados deseados. La creación de la base datos, tablas etc. será realizada en la herramienta DB-FIDDLE la cual se puede acceder de manera gratuita y sin instalación ninguna a través del navegador web en el siguiente enlace: <https://www.db-fiddle.com>

En resumen, la base de datos consta de cuatro tablas que se mencionarán más adelante con más detalle, cada una de ellas con campos personalizados, para así poder probar de manera muy intuitiva la conexión que existen entre todas, además de haber realizado la inserción de 87 personas, la mayoría ficticias con sus respectivos datos para realizar consultas de todo tipo.

Creación Base de Datos

Antes de todo si estás usando DB-FIDDLE debes saltar al siguiente paso que se corresponde a la creación de tablas, por el contrario si estás usando otra herramienta como MySQL Workbench este paso es obligatorio.

Para crear la base de datos para así poder empezar con la creación de las tablas, la inserción y demás. Para crear la base de datos lo que vamos a ingresar es lo siguiente: **DROP DATABASE IF EXISTS ESCUELA;** Esto significa que si ya existe la base de datos llamada escuela la borre por completo, como apunte importante toda consulta SQL tiene que acabar con punto y coma.

Ahora viene la creación de la base de datos para esto escribiremos lo siguiente: **CREATE DATABASE ESCUELA;** Como se puede observar lo que realizará la línea de código es crear la base de datos que se va a llamar escuela.

Para terminar la secuencia de la creación de la base de datos, esta vez escribiremos lo siguiente: **USE ESCUELA;** Con esto especificamos la base de datos que se va a utilizar para las operaciones que se ejecuten posteriormente. En otras palabras, te permite cambiar la base de datos activa en tu sesión de trabajo actual.

```
DROP DATABASE IF EXISTS ESCUELA;  
CREATE DATABASE ESCUELA;  
USE ESCUELA;
```

Creación de tablas

La base de datos escuela como se mencionó previamente consta de cuatro tablas, estas tablas son Alumnos, País, Carrera y Matricula cada una de ellas con sus respectivos datos y variables que se compartirán y explicarán a continuación.

Alumnos

La tabla alumnos como bien indica su nombre es donde se ingresarán los datos de los alumnos, para eso en esta tabla las columnas de la tabla alumnos serán id, apellidos, nombre, género, correo, ciudad y por último id_pais. Para explicar las variables de los campos iremos una por una explicando que significa cada una.

Primero tenemos id que hemos puesto int, que significa que tendrá el valor de un número entero, luego primary key, que viene siendo clave primaria, es decir que es un campo que se identifica de manera única en cada fila de esa tabla y por último auto_increment, esto se ejecuta para que el id por cada valor ingresado el id incremente de valor de uno en uno. Para id_pais tenemos también el valor int de número entero para que en vez de poner el nombre del país de cada uno asignemos un número en específico a ese país para luego conectarlo a la tabla PAÍS.

En segundo lugar, para generalizar tendremos los campos apellidos, nombre, correo y ciudad estos cuatros campos tienen el valor varchar, este es un tipo de datos que almacena cadenas de caracteres de longitud variable, la longitud máxima se especifica entre los paréntesis. También se aprecia como esto se acompaña de la variable not null, esto con el fin de que no haya valores nulos en los campos especificados.

Por último, tenemos el campo género el cual tiene la variable char, que es muy similar a varchar, solo que en vez de ser una cadena de caracteres variable ahora es fija, es decir si tiene menos caracteres de los especificados los que faltan se rellenan con espacios en blanco. Además, también tenemos la variable check para especificar que este campo solo pueda tener los valores M para masculino y F para femenino.

```
-- CREACION TABLA ALUMNOS
CREATE TABLE ALUMNOS(
ID INT PRIMARY KEY AUTO_INCREMENT,
APELLIDOS VARCHAR(50) NOT NULL,
NOMBRE VARCHAR(50) NOT NULL,
GENERO CHAR(1) CHECK (GENERO IN ('M', 'F')),
CORREO VARCHAR(70) NOT NULL,
CIUDAD VARCHAR(50) NOT NULL,
id_pais int
);
```

País

La tabla PAIS es muy sencilla ya que solo consta de tres columnas con el fin de mostrar la primera conexión entre tablas, estos campos son id, PAIS y cod_pais.

A partir de ahora el primer campo id va a ser igual para todas las tablas con las mismas variables de antes y significando exactamente lo mismo, todo esto para que tenga un orden cada valor y pueda existir la conexión entre tablas.

La columna País y cod_pais ambas tendrán la variable varchar una especificando el nombre del país y la otra su código, es decir para España sería ESP.

```
-- CREACION TABLA PAIS --  
CREATE TABLE PAIS(  
id int primary key auto_increment,  
PAIS varchar(50) not null,  
cod_pais varchar(5) not null  
);
```

Carrera

Para la tabla carrera se especificará todas las carreras de todas las personas ingresadas en la tabla alumnos, como en la tabla alumnos no se dicta la carrera de nadie esta tabla está para eso ya que se conectará con el id de cada alumno para especificar su carrera estudiada. Esta tabla cuenta con varios campos como id_carrera, nombre_carrera, duración y descripción.

Una vez más el primer campo que corresponde a id será un número entero, clave primaria y auto_increment. Para campos como duración que también es un valor de número entero se especifica de cuantos años es la carrera.

Luego tendremos campos como nombre_carrera que será varchar, es decir una cadena de caracteres y not null ya que no podrá tener un valor nulo.

Por último, tenemos el campo descripción el cual tiene la variable longtext, esto significa que este campo puede almacenar texto de longitud variable pero esta vez de una gran cantidad de caracteres por lo tanto no escribiremos ningún valor numérico.

```
-- CREACION TABLA CARRERA --
```

```
CREATE TABLE CARRERA (
```

```
  ID_CARRERA INT PRIMARY KEY AUTO_INCREMENT,
```

```
  NOMBRE_CARRERA VARCHAR(100) NOT NULL,
```

```
  DURACION INT NOT NULL,
```

```
  DESCRIPCION LONGTEXT NOT NULL
```

```
);
```

Matrícula

Para la última tabla que es matrícula de nuestra base de datos hemos ingresado varios campos muy interesantes a explicar, esta tabla servirá de enlace para todas las demás. Esta tabla cuenta con campos como id_matricula, id_alumno, id_carrera y fecha_matricula.

Para la primera columna que es id_matricula se especifica los mismos valores que anteriormente se han puesto a todas las demás ids. Por otro lado, para id_alumno e id_carrera se atribuye el valor int de número entero y en las últimas dos líneas se especifica que ambas son foreign key, es decir, que establecen una relación referencial con una clave primaria de otra tabla y así pueda existir un vínculo o conexión entre ambas y el valor references se utiliza para especificar la tabla y columna a la que se está haciendo referencia. Por último, tenemos fecha_matricula que como bien su nombre indica el valor será el de una fecha cualquiera.

```
-- CREACION TABLA MATRICULA --
```

```
CREATE TABLE MATRICULA (
```

```
  ID_MATRICULA INT PRIMARY KEY AUTO_INCREMENT,
```

```
  ID_ALUMNO INT,
```

```
  ID_CARRERA INT,
```

```
  FECHA_MATRICULA DATE NOT NULL,
```

```
  FOREIGN KEY (ID_ALUMNO) REFERENCES ALUMNOS(ID),
```

```
  FOREIGN KEY (ID_CARRERA) REFERENCES CARRERA(ID_CARRERA)
```

```
);
```

Resultado tablas sin inserción

Ahora en este apartado mostraremos brevemente como hacer una consulta de las tablas antes de insertar los datos en cada una. Para ir a una tabla tendremos que escribir siempre la misma estructura que es: **SELECT * FROM nombre_tabla;**

Tabla alumnos

```
1 • SELECT * FROM a.alumnos;
```

ID	APELLIDOS	NOMBRE	GENERO	CORREO	CIUDAD	id_pais
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabla carrera

```
1 • SELECT * FROM a.carrera;
```

ID_CARRERA	NOMBRE_CARRERA	DURACION	DESCRIPCION
NULL	NULL	NULL	NULL

Tabla matricula

```
1 • SELECT * FROM a.matricula;
```

ID_MATRICULA	ID_ALUMNO	ID_CARRERA	FECHA_MATRICULA
NULL	NULL	NULL	NULL

Tabla país

```
1 • SELECT * FROM a.pais;
```

id	PAIS	cod_pais
NULL	NULL	NULL

Inserción en tablas

Para poder pasar al siguiente paso que es insertar datos en una tabla es necesario saber la estructura para hacerlo ya que es muy importante. La estructura de inserción es la siguiente: **INSERT INTO nombre_tabla(campos_tabla) VALUES ('datos a insertar');**

Además de ya conocer la estructura es importante saber datos adicionales por ejemplo en nombre_tabla se pondrá el nombre de la tabla en la que queramos insertar los datos, en campos_tabla se tiene que poner todos los campos de la tabla correspondiente en el orden en el que has puesto los campos a la hora de crear la tabla y separados mediante comas y nunca debes poner la clave primaria esa nunca se escribe.

Por último, en datos a insertar son los datos que van escritos es cada columna de la tabla, cada paréntesis es una fila, cada dato va separado mediante comas y si es cadena de caracteres va entre comillas y si es un valor numérico no hay que ponerlas.

En el siguiente paso se mostrarán las capturas de las primeras 5 filas de cada inserción de cada una de las tablas de la base de datos.

Inserción tabla alumnos

```
INSERT INTO ALUMNOS(APELLIDOS,NOMBRE,GENERO,CORREO,CIUDAD,id_pais) VALUES
('TOCINO FERNANDEZ','MARIA SALUD','F','MTOCINO FERNANDEZ@GMAIL.COM','Cadiz',1),
('PARDO PRADO','SARA','F','SPARDOPRADO@HOTMAIL.COM','A Coruña',1),
('TOBALINA PULIDO','LETICIA','F','LTOTALINAPULIDO@HOTMAIL.COM','Bilbao',1),
('KELLY HOPFENBLATT','TATIANA','F','TKELLYHOPFENBLATT@GMAIL.COM','Berna',2),
('MUÑOZ GARCIA','PATRICIA','F','PMUÑOZGARCIA@OUTLOOK.COM','Cadiz',1);
```

13 Select * from ALUMNOS

test to ODL

Results

Query #1 Execution time: 1ms

ID	APELLIDOS	NOMBRE	GENERO	CORREO	CIUDAD	id_pais
1	TOCINO FERNANDEZ	MARIA SALUD	F	MTOCINO FERNANDEZ@GMAIL.COM	Cadiz	1
2	PARDO PRADO	SARA	F	SPARDOPRADO@HOTMAIL.COM	A Coruña	1
3	TOBALINA PULIDO	LETICIA	F	LTOTALINAPULIDO@HOTMAIL.COM	Bilbao	1
4	KELLY HOPFENBLATT	TATIANA	F	TKELLYHOPFENBLATT@GMAIL.COM	Berna	2
5	MUÑOZ GARCIA	PATRICIA	F	PMUÑOZGARCIA@OUTLOOK.COM	Cadiz	1

Copy as Markdown

Inserción tabla país

```
-- INSERT INTO TABLA PAIS --
```

```
INSERT INTO PAIS (PAIS, cod_pais) VALUES
```

```
('España', 'ES'),
```

```
('Suiza', 'CH'),
```

```
('Inglaterra', 'GB'),
```

```
('Francia', 'FR'),
```

```
('Italia', 'IT'),
```

```
('Alemania', 'DE');
```

```
--
49 -- INSERT INTO TABLA PAIS --
50 INSERT INTO PAIS (PAIS, cod_pais) VALUES
51 ('España', 'ES'),
52 ('Suiza', 'CH'),
53 ('Inglaterra', 'GB'),
54 ('Francia', 'FR'),
55 ('Italia', 'IT'),
56 ('Alemania', 'DE');
```

Text to DDL

```
14 Select * from PAIS;
```

Results

Query #1 Execution time: 0ms

id	PAIS	cod_pais
1	España	ES
2	Suiza	CH
3	Inglaterra	GB
4	Francia	FR
5	Italia	IT
6	Alemania	DE

Inserción tabla carrera

```
-- INSERT INTO TABLA CARRERA --
```

```
INSERT INTO CARRERA (NOMBRE_CARRERA, DURACION, DESCRIPCION) VALUES
```

```
('Filología clasica y doctora en latin', 5, 'Estudio avanzado de la lengua y cultura clásica, con especial
énfasis en la literatura y el latín.'),
```

```
('Lenguas Modernas', 4, 'Estudio de varias lenguas modernas, incluyendo literatura, cultura y
traducción.'),
```

```
('Arqueología', 5, 'Investigación y excavación de restos materiales de culturas antiguas para comprender
la historia y la evolución humana.'),
```

```
('Museología', 3, 'Gestión y conservación de museos, con enfoque en la preservación y presentación de
artefactos y colecciones.'),
```

```
('Filología Hispanica', 4, 'Estudio profundo de la lengua y literatura españolas, abarcando diferentes
periodos históricos.'),
```

```
60 -- INSERT INTO TABLA CARRERA --
61 INSERT INTO CARRERA (NOMBRE_CARRERA, DURACION, DESCRIPCION) VALUES
62 ('Filología clasica y doctora en latin', 5, 'Estudio avanzado de la lengua y cultura clásica, con especial
énfasis en la literatura y el latín.'),
63 ('Lenguas Modernas', 4, 'Estudio de varias lenguas modernas, incluyendo literatura, cultura y traducción.'),
64 ('Arqueología', 5, 'Investigación y excavación de restos materiales de culturas antiguas para comprender la
historia y la evolución humana.'),
65 ('Museología', 3, 'Gestión y conservación de museos, con enfoque en la preservación y presentación de
artefactos y colecciones.'),
66 ('Filología Hispanica', 4, 'Estudio profundo de la lengua y literatura españolas, abarcando diferentes
periodos históricos.'),
```

Text to DDL

```
11 Select * from CARRERA;
```

Results

Query #1 Execution time: 1ms

ID_CARRERA	NOMBRE_CARRERA	DURACION	DESCRIPCION
1	Filología clasica y doctora en latin	5	Estudio avanzado de la lengua y cultura clásica, con especial énfasis en la literatura y el latín.
2	Lenguas Modernas	4	Estudio de varias lenguas modernas, incluyendo literatura, cultura y traducción.
3	Arqueología	5	Investigación y excavación de restos materiales de culturas antiguas para comprender la historia y la evolución humana.
4	Museología	3	Gestión y conservación de museos, con enfoque en la preservación y presentación de artefactos y colecciones.
5	Filología Hispanica	4	Estudio profundo de la lengua y literatura españolas, abarcando diferentes periodos históricos.

Inserción tabla matrícula

```
-- INSERT INTO TABLA MATRICULAS --
```

```
INSERT INTO MATRICULA (ID_ALUMNO, ID_CARRERA, FECHA_MATRICULA) VALUES
```

```
(1, 1, '2023-09-01'),
```

```
(2, 2, '2023-09-02'),
```

```
(3, 3, '2023-09-03'),
```

```
(4, 4, '2023-09-04'),
```

```
(5, 5, '2023-09-05');
```

```
72 -- INSERT INTO TABLA MATRICULAS --
73 INSERT INTO MATRICULA (ID_ALUMNO, ID_CARRERA, FECHA_MATRICULA) VALUES
74 (1, 1, '2023-09-01'),
75 (2, 2, '2023-09-02'),
76 (3, 3, '2023-09-03'),
77 (4, 4, '2023-09-04'),
78 (5, 5, '2023-09-05');
79
80
```

Text to DDL

```
13 Select * from MATRICULA;
```

Results

Query #1 Execution time: 0ms

ID_MATRICULA	ID_ALUMNO	ID_CARRERA	FECHA_MATRICULA
1	1	1	2023-09-01
2	2	2	2023-09-02
3	3	3	2023-09-03
4	4	4	2023-09-04
5	5	5	2023-09-05

Consultas SQL

Las consultas o queries en las bases de datos y SQL se refieren a comandos o instrucciones que se utilizan con el fin de poder interactuar con la base de datos. Básicamente son operaciones para realizar acciones como la recuperación de datos, inserción de datos, actualización de datos o eliminación de datos entre otros. Cuando previamente hemos realizado los SELECT o INSERT INTO en las tablas ya estábamos realizando consultas, pero esta vez daremos un paso más allá con recolección de datos importantes.

¿Qué carrera estudia x alumno?

Por ejemplo, queremos saber que carrera estudia el alumno número 27 o si sabemos el nombre que se llama Javier Martínez Serrano queremos saber que ha estudiado, para poder saberlo tendremos que realizar una consulta.

```
SELECT CARRERA.NOMBRE_CARRERA FROM ALUMNOS
JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
JOIN CARRERA ON MATRICULA.ID_CARRERA = CARRERA.ID_CARRERA
WHERE ALUMNOS.APELLIDOS = 'MARTÍNEZ SERRANO' AND ALUMNOS.NOMBRE = 'JAVIER';
```

Lo que hemos realizado en la consulta es un **select** para que nos muestre el campo que queremos **from alumnos** que se refiere a la tabla alumnos seguido de un **join**, esto se utiliza para combinar filas de dos o más tablas diferentes basándose en la relación que hay entre ellas, lo que está realizando es una combinación entre la tabla alumnos y la tabla matrícula en donde dice que el campo id de la tabla alumnos es igual al campo id_alumno de la tabla matrícula, de nuevo ejecuta otro join haciendo referencia a la tabla carrera y termina con un **where**, que se utiliza para filtrar resultados con una condición específica que en este caso son los apellidos y nombre del alumno.

Como resultado MySQL Workbench nos dice que la carrera de Javier Martínez Serrano es Filología Hispánica.

Query SQL ●

```
1 SELECT CARRERA.NOMBRE_CARRERA
2 FROM ALUMNOS
3 JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
4 JOIN CARRERA ON MATRICULA.ID_CARRERA = CARRERA.ID_CARRERA
5 WHERE ALUMNOS.APELLIDOS = 'MARTÍNEZ SERRANO' AND ALUMNOS.NOMBRE = 'JAVIER';
```

NOMBRE_CARRERA

Filología Hispánica

¿Cuántos alumnos estudian x carrera?

Otro ejemplo puede ser que necesitáramos saber cuantos alumnos han estudiado cualquier carrera para saber esto necesitaremos ejecutar la siguiente consulta.

```
SELECT COUNT (*) AS Cantidad_Alumnos FROM ALUMNOS
JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
WHERE MATRICULA.ID_CARRERA = 2;
```

En esta consulta empezamos de nuevo con **select**, pero usamos la cláusula **COUNT (*)** para indicar que queremos contar el número de filas que cumplen la condición indicada posteriormente. **From alumnos** ya que se indica que es de la tabla alumnos y de nuevo se realiza otro **join** para combinar la tabla alumnos con matricula en donde el campo id de alumnos es igual al campo id_alumno de matrícula en donde se cumpla la condición con la cláusula **where** de que la carrera sea la número 2 que es Lenguas Modernas.

Como resultado MySQL Workbench nos dice que hay 10 alumnos que estudian Lenguas Modernas.

Query SQL ●

```
1 SELECT COUNT(*) AS Cantidad_Alumnos
2 FROM ALUMNOS
3 JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
4 WHERE MATRICULA.ID_CARRERA = 2;
```

Cantidad_Alumnos

10

¿Cuántos alumnos son de x país?

Si quisiésemos saber cuantos alumnos son de algún país en específico las consultas también son de gran ayuda, para poder hacerlo deberíamos ejecutar la siguiente querie.

```
SELECT COUNT (*) AS Cantidad_Alumnos_Francia FROM ALUMNOS  
JOIN PAIS ON ALUMNOS.id_pais = PAIS.id  
WHERE PAIS.PAIS = 'Francia';
```

En este ejemplo queremos saber cuantas personas de Francia hay en nuestra base de datos, para esto haremos de nuevo un **select COUNT (*) FROM ALUMNOS** y luego un **join** para combinar la tabla alumnos a la tabla país en donde el campo **id_pais** de la tabla alumnos es igual al campo **id** de la tabla país y ejecutando nuevamente otra condición usando **where** para decir que el país es Francia.

Como resultado MySQL Workbench nos dice que hay 13 alumnos de Francia en nuestra base de datos.

Query SQL ●

```
1 SELECT COUNT(*) AS Cantidad_Alumnos_Francia  
2 FROM ALUMNOS  
3 JOIN PAIS ON ALUMNOS.id_pais = PAIS.id  
4 WHERE PAIS.PAIS = 'Francia';
```

Cantidad_Alumnos_Francia

13

¿Cuántos alumnos de x país estudian x carrera?

Por último, utilizaremos algo parecido a las 2 últimas consultas, si quisiéramos saber cuantos alumnos de España estudian Derecho en nuestra base de datos tendríamos que hacer la siguiente consulta.

```
SELECT COUNT (*) AS Cantidad_Alumnos_Espana_Derecho FROM ALUMNOS
JOIN PAIS ON ALUMNOS.id_pais = PAIS.id
JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
JOIN CARRERA ON MATRICULA.ID_CARRERA = CARRERA.ID_CARRERA
WHERE PAIS.PAIS = 'España' AND CARRERA.NOMBRE_CARRERA = 'Derecho';
```

Lo curioso de esta consulta es que se realiza un triple join para combinar la tabla alumnos con la tabla país, matrícula y carrera para acabar con la cláusula where para especificar que el alumno sea de España y su carrera Derecho.

Como resultado final MySQL Workbench nos dice que en nuestra base de datos hay 6 alumnos de España que estudian Derecho.

Query SQL ●

```
1 SELECT COUNT(*) AS Cantidad_Alumnos_Espana_Derecho
2 FROM ALUMNOS
3 JOIN PAIS ON ALUMNOS.id_pais = PAIS.id
4 JOIN MATRICULA ON ALUMNOS.ID = MATRICULA.ID_ALUMNO
5 JOIN CARRERA ON MATRICULA.ID_CARRERA = CARRERA.ID_CARRERA
6 WHERE PAIS.PAIS = 'España' AND CARRERA.NOMBRE_CARRERA = 'Derecho';
```

Cantidad_Alumnos_Espana_Derecho

6