





Clase 19

BASE DE DATOS 2era parte



SENTENCIA SELECT





La instrucción **SELECT** nos permite traer datos de una tabla.

select * from products;

product_id	product_name	brand_id	category_id	model_year	list_price
1	Trek 820 - 2016	9	6	2016	379.99
2	Ritchey Timberwolf Frameset - 2016	5	6	2016	749.99
3	Surly Wednesday Frameset - 2016	8	6	2016	999.99
4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99
5	Heller Shagamaw Frame - 2016	3	6	2016	1320.99





SELECT WHERE



SENTENCIA SELECT WHERE

La instrucción WHERE nos permite traer datos que cumplan una condición.

SELECT * FROM products WHERE model_year = '2017';

product_id	product_name	brand_id	category_id	model_year	list_price
27	Surly Big Dummy Frameset - 2017	8	6	2017	999.99
28	Surly Karate Monkey 27.5+ Frameset - 2017	8	6	2017	2499.99
29	Trek X-Caliber 8 - 2017	9	6	2017	999.99
30	Surly Ice Cream Truck Frameset - 2017	8	6	2017	999.99
The court	The Property of the Control of the C				The same and the same and







SHOW TABLES

Mediante el uso de la sentencia SHOW TABLES podremos listar las tablas de una base de datos.

1	SHOW TABLES FROM product	tion
⇔.	Consultas favoritas 🗸	Histo
Table	es_in_production	
bran	nds	
cate	gories	
proc	ducts	
stoc	:ks	



INSERT INTO







- La sentencia "INSERT INTO" es utilizada para crear nuevos registros en una tabla.
- Existen dos formas de crear registros dentro de una tabla
- En la siguiente forma, se especifica las columnas sobre las cuales queremos agregar datos:

```
INSERT INTO brands (brand_name) VALUES ('Audi')
```

- Otra forma de crear registros es indicando valores para todas las columnas de la tabla:

INSERT INTO brands VALUES (11, 'BMW');







DELETE



DELETE







- La sentencia "DELETE" es utilizada para eliminar registros existentes en una tabla.
- Al momento de ejecutar dicha sentencia, debemos prestar mucha atención con los registros que queremos eliminar, ya que si no utilizamos ningún filtro, se eliminarán todos los registros de la tabla. Es decir:
 - DELETE FROM brands, eliminará todos los registros de dicha tabla
 - DELETE FROM brands WHERE brand_id = 1, eliminará el registro cuyo id es igual 1.













• La sentencia "UPDATE" es utilizada para actualizar registros existentes en una tabla.

- Al igual que sucedía con la sentencia DELETE, debemos prestar mucha atención con los registros que queremos actualizar, ya que si no utilizamos ningún filtro, se actualizarán todos los registros de la tabla.

Sintaxis

UPDATE [nombre_tabla]
SET nombre_columna = valor WHERE condición;







- Actualizando un campo, a partir de su campo identidad

UPDATE usuario
SET estado = 2 WHERE usuario_id = 1;







- Actualizando múltiples campos, a partir de su campo identidad.

UPDATE usuario

SET estado = 1, usuario = 'juanca' WHERE usuario_id = 1;







PRIMARY KEYS



CLAVE PRIMARIA COMO ID





La instrucción PRIMARY KEY define la clave primaria de la tabla.

```
CREATE TABLE Personas (
   id int PRIMARY KEY,
   apellido varchar(255) NOT NULL,
   nombre varchar(255),
   edad int
);
```



AUTOINCREMENTAL





La instrucción **AUTO_INCREMENT** produce que el identificador se incremente cada vez que se inserte un nuevo registro.

```
CREATE TABLE Personas (
   id int PRIMARY KEY AUTO_INCREMENT,
   apellido varchar(255) NOT NULL,
   nombre varchar(255),
   edad int
);
```



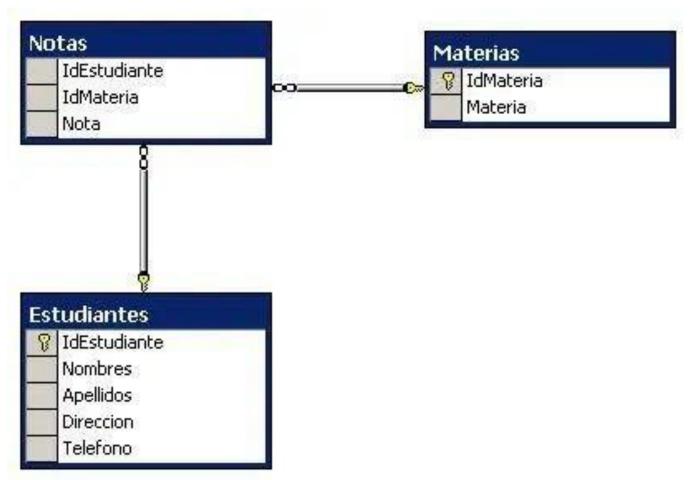




FOREIGN KEYS



RELACIONAR MÚLTIPLES TABLAS









FOREIGN KEY





```
CREATE TABLE Estudiantes (
   idEstudiante int PRIMARY KEY AUTO_INCREMENT,
   apellido varchar(255) NOT NULL,
   nombre varchar(255) NOT NULL,
   direccion varchar(255),
   telefono varchar(255)
);
```

```
CREATE TABLE Materias (
   idMateria int PRIMARY KEY AUTO_INCREMENT,
   materia varchar(255)
);
```



FOREIGN KEY





La instrucción FOREIGN KEY la utilizamos para unir lógicamente dos tablas.

```
CREATE TABLE Notas (
   idEstudiante int PRIMARY KEY,
   idMateria int,
   nota int,
   FOREIGN KEY (idEstudiante) REFERENCES Estudiantes(idEstudiante),
   FOREIGN KEY (idMateria) REFERENCES Materias(idMateria)
);
```







JOINS



JOIN





Con un JOIN puedo traer todas las Notas de un estudiante y también su información.

```
SELECT * FROM Notas

JOIN Estudiantes ON Estudiantes.idEstudiante = Notas.idEstudiante
WHERE isEstudiante = 1;
```



JOIN

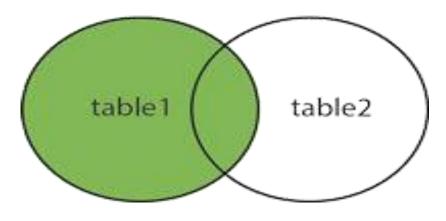




Con un **LEFT JOIN** puedo traer todas las Notas de un estudiante y también su información incluso aunque el estudiante no tenga ninguna nota registrada.

SELECT * FROM Estudiantes

LEFT JOIN Notas ON Estudiantes.idEstudiante = Notas.idEstudiante
WHERE isEstudiante = 1; LEFT JOIN









JOINS & GROUP BY



JOINS & GROUP BY





La instrucción JOIN en conjunto con GROUP BY nos permite agrupar registros de múltiples tablas.

```
SELECT store_name, count(*)
FROM sales.staffs st
JOIN sales.stores s ON s.store_id = st.store_id
GROUP BY s.store_name;
```

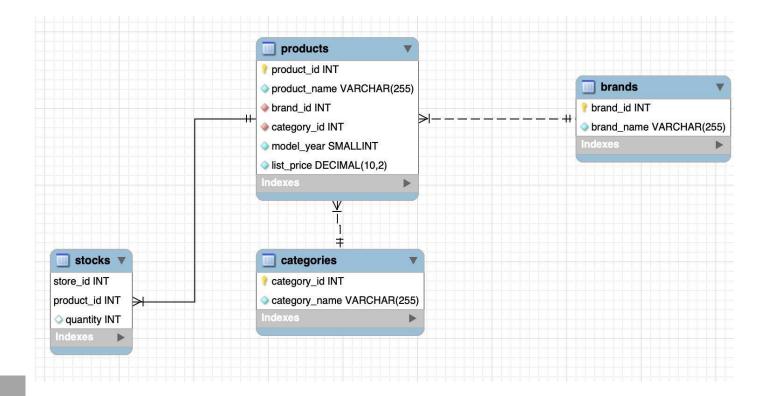


DB SCHEMA









Un SCHEMA de BD es un conjunto de tablas.

Para acceder a una de las tablas se utiliza esquema.tabla

Por ejemplo, si mi esquema se llama concesionario la tabla de autos se accede como:

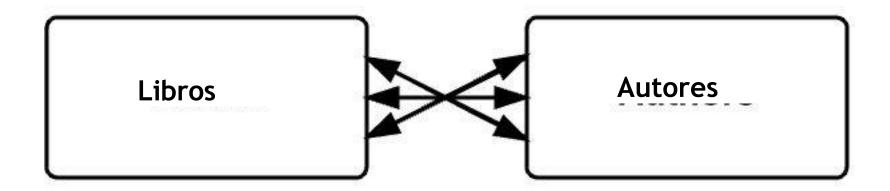
concesionario.auto



Libros <-> Autores









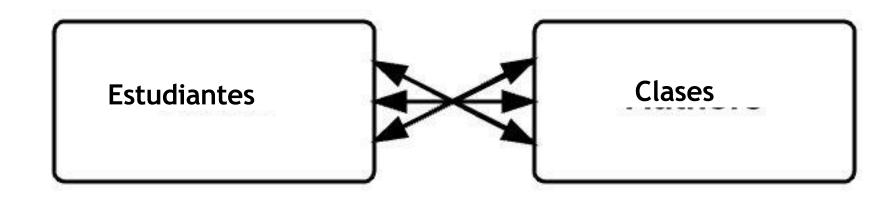




Relación muchos a muchos



Relación muchos a muchos









Relación muchos a muchos

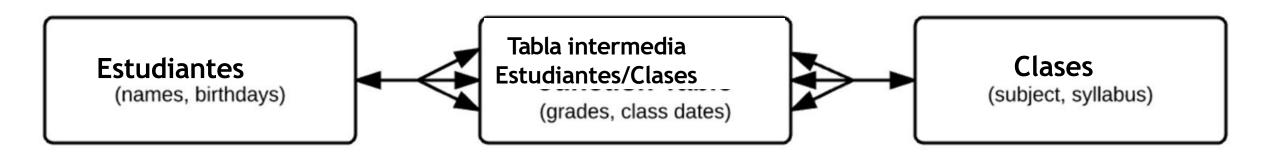








TABLA INTERMEDIA





La tabla intermedia almacena la relación entre los estudiantes y las clases.

```
CREATE TABLE estudiante_clase (
   id_estudiante INT,
   id_clase INT,
   FOREIGN KEY (id_estudiante) REFERENCES estudiante(id_estudiante),
   FOREIGN KEY (id_clase) REFERENCES clase(id_clase)
);
```



TABLA INTERMEDIA





La tabla intermedia puede contener información extra sobre la relación.

```
CREATE TABLE estudiante_clase (
   id_estudiante INT,
   id_clase INT,
   nota INT,
   FOREIGN KEY (id_estudiante) REFERENCES estudiante(id_estudiante),
   FOREIGN KEY (id_clase) REFERENCES clase(id_clase)
);
```



FUNCIONES QUE AGREGAN DATOS

Las funciones que agregan datos se utilizan en conjunto con los GROUP BY

AVG	Utilizada para calcular el promedio de los valores de un campo determinado	
COUNT	Utilizada para devolver el número de registros de la selección	
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado	
MAX	Utilizada para devolver el valor más alto de un campo especificado	
MIN	Utilizada para devolver el valor más bajo de un campo especificado	





