

Tarea HIT02

Universitario: Samuel poma Velasco

Materia: Base de datos I

11.Crear el diseño para una UNIVERSIDAD.

UNIVERSIDAD

Id(integer, Llave primaria y unica)

nombre(varchar = string)

Direccion(varchar = string)

Carreras(varchar = int)

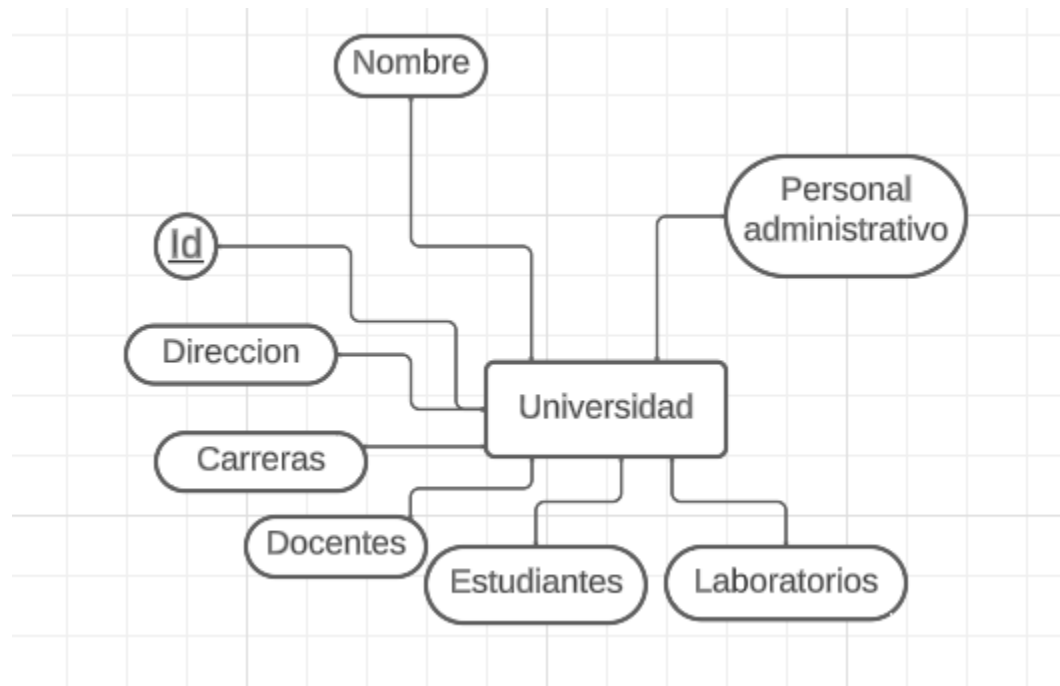
Docentes(varchar = int)

Estudiantes(varchar = int)

Laboratorios(varchar = int)

Personal_administrativo(varchar = int)

12. Crear el diagrama Entidad Relación E-R para el ejercicio anterior



13. Crear la tabla universidad en base al diseño anterior

```
CREATE TABLE UNIVERSIDAD (  
    Id INT PRIMARY KEY,  
    nombre VARCHAR(255),  
    Direccion VARCHAR(255),  
    Carreras INT,  
    Docentes INT,  
    Estudiantes INT,  
    Laboratorios INT,  
    Personal_administrativo INT  
);
```

14. Agregar registros a la tabla creada anteriormente.

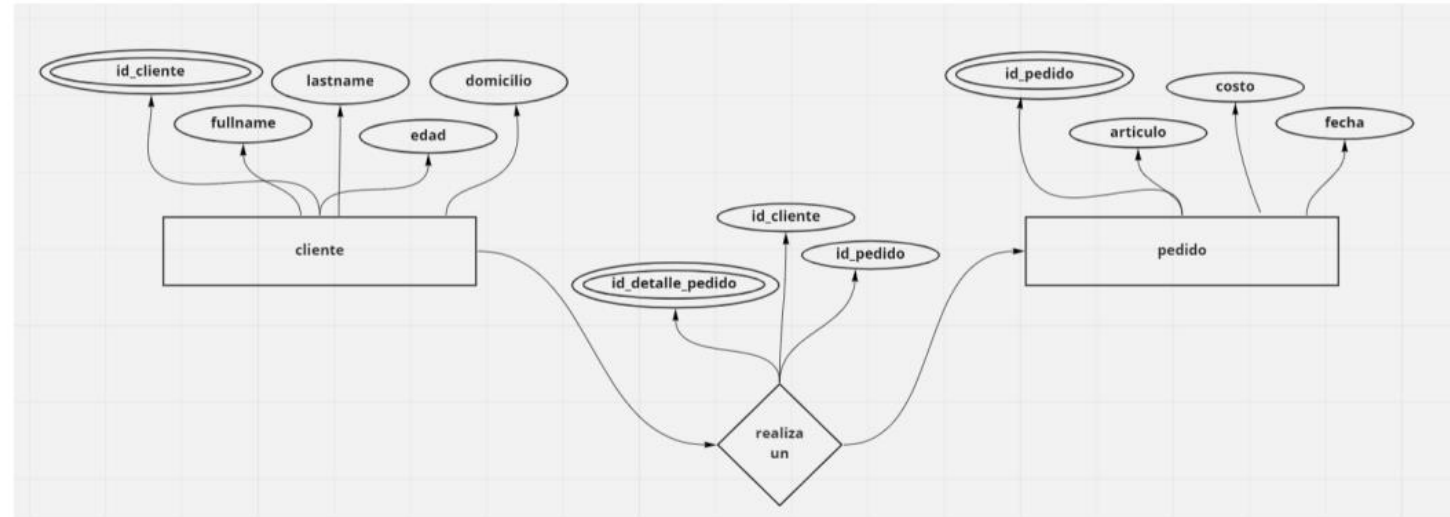
```
INSERT INTO UNIVERSIDAD (Id, nombre, Direccion, Carreras, Docentes, Estudiantes, Laboratorios, Personal_administrativo)
VALUES
(1, 'UPEA', 'El alto', 15, 30, 500, 10, 50);
```

```
INSERT INTO UNIVERSIDAD (Id, nombre, Direccion, Carreras, Docentes, Estudiantes, Laboratorios, Personal_administrativo)
VALUES
(2, 'UMSA', 'La paz', 20, 45, 700, 15, 60);
```

```
INSERT INTO UNIVERSIDAD (Id, nombre, Direccion, Carreras, Docentes, Estudiantes, Laboratorios, Personal_administrativo)
VALUES
(3, 'UNIFRANZ', 'La paz', 12, 25, 400, 8, 40);
```

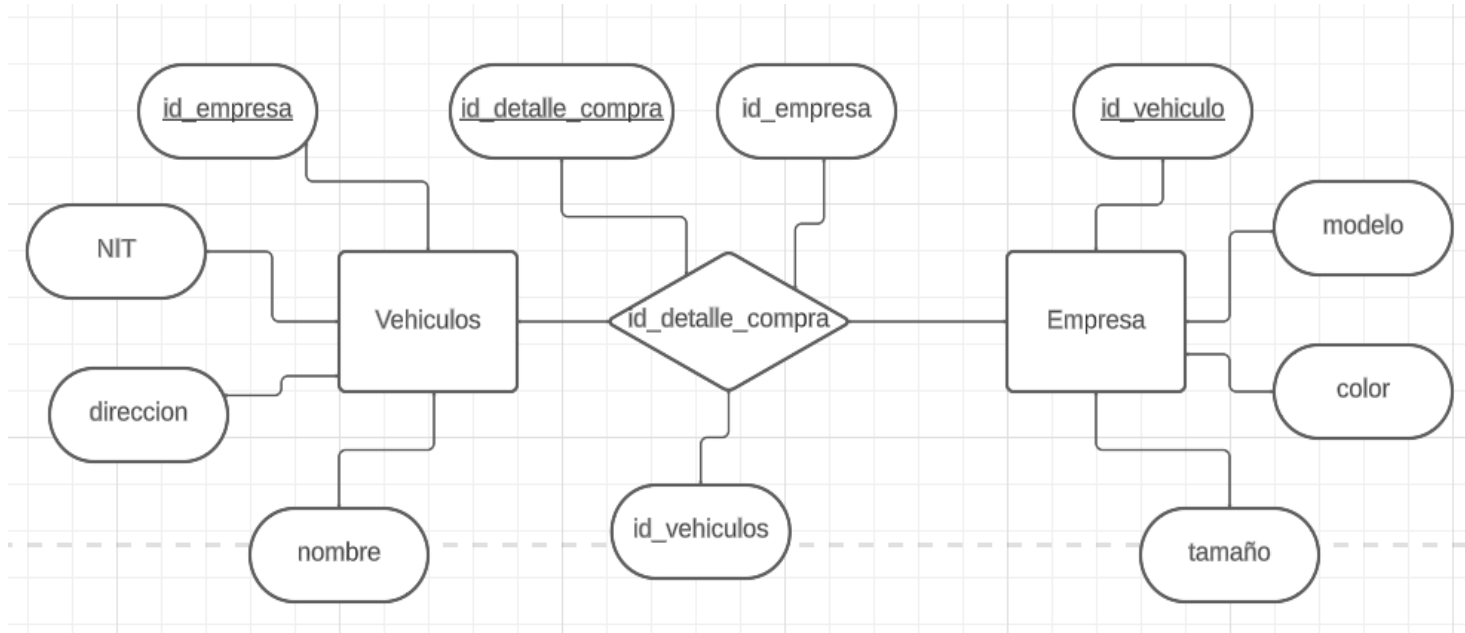
15. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER

```
1 CREATE TABLE cliente (  
2   id_cliente INT PRIMARY KEY,  
3   fullname VARCHAR(255),  
4   lastname VARCHAR(255),  
5   edad INT,  
6   domicilio VARCHAR(255)  
7 );  
8 CREATE TABLE pedido (  
9   id_pedido INT PRIMARY KEY,  
10  articulo VARCHAR(255),  
11  costo DECIMAL(10, 2),  
12  fecha DATE  
13 );  
14 CREATE TABLE detalle_pedido (  
15   id_detalle_pedido INT PRIMARY KEY,  
16   id_cliente INT,  
17   id_pedido INT,  
18   FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),  
19   FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido)  
20 );
```



```
23 INSERT INTO cliente (id_cliente, fullname, lastname, edad, domicilio)
24 VALUES (1, 'Juan', 'Pérez', 30, 'Calle 123, Ciudad A');
25
26 INSERT INTO pedido (id_pedido, articulo, costo, fecha)
27 VALUES (1, 'Producto A', 100.00, '2023-09-13');
28
29 INSERT INTO detalle_pedido (id_detalle_pedido, id_cliente, id_pedido)
30 VALUES (1, 1, 1); -- Asociar el cliente con id_cliente 1 al pedido con id_pedido 1
31
32 INSERT INTO cliente (id_cliente, fullname, lastname, edad, domicilio)
33 VALUES (2, 'María', 'López', 28, 'Avenida 456, Ciudad B');
34
35 INSERT INTO pedido (id_pedido, articulo, costo, fecha)
36 VALUES (2, 'Producto B', 150.00, '2023-09-14');
37
38 INSERT INTO detalle_pedido (id_detalle_pedido, id_cliente, id_pedido)
39 VALUES (2, 2, 2); -- Asociar el cliente con id_cliente 2 al pedido con
```

16. Crear el modelo entidad relación ER y su código SQL.




```
1 CREATE TABLE empresa (  
2     id_empresa INT PRIMARY KEY,  
3     nit VARCHAR(255),  
4     direccion VARCHAR(255),  
5     nombre VARCHAR(255)  
6 );
```

```
7  
8 CREATE TABLE vehiculos (  
9     id_vehiculo INT PRIMARY KEY,  
10    modelo VARCHAR(255),  
11    color VARCHAR(255),  
12    tamaño VARCHAR(255)  
13 );
```

```
14  
15 CREATE TABLE detalle_compra (  
16     id_detalle_compra INT PRIMARY KEY,  
17     id_empresa INT,  
18     id_vehiculo INT,  
19     FOREIGN KEY (id_empresa) REFERENCES empresa(id_empresa),  
20     FOREIGN KEY (id_vehiculo) REFERENCES vehiculos(id_vehiculo)  
21 );
```

```
22
```

```
24 INSERT INTO empresa (id_empresa, nit, direccion, nombre)  
25 VALUES (1, '123456-7', 'Calle 123, Ciudad A', 'Empresa A');  
26 INSERT INTO vehiculos (id_vehiculo, modelo, color, tamaño)  
27 VALUES (1, 'Sedán', 'Rojo', 'Mediano');  
28 INSERT INTO detalle_compra (id_detalle_compra, id_empresa, id_vehiculo)  
29 VALUES (1, 1, 1); -- Asociar la empresa con id_empresa 1 al vehículo con id_vehiculo 1  
30  
31  
32 INSERT INTO empresa (id_empresa, nit, direccion, nombre)  
33 VALUES (2, '987654-3', 'Avenida 456, Ciudad B', 'Empresa B');  
34 INSERT INTO vehiculos (id_vehiculo, modelo, color, tamaño)  
35 VALUES (2, 'Camioneta', 'Azul', 'Grande');  
36 INSERT INTO detalle_compra (id_detalle_compra, id_empresa, id_vehiculo)  
37 VALUES (2, 2, 2); -- Asociar la segunda empresa con id_empresa 2 al segundo vehículo con id_vehiculo 2  
38 |
```

cuestionario

1. ¿Qué son las bases de datos?

R. Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático.

2. ¿A que se refiere cuando se habla de bases de datos relacionales?

R. Cuando se habla de bases de datos relacionales, se refiere a un tipo de sistema de gestión de bases de datos que organiza la información en tablas estructuradas que están relacionadas entre sí. En este tipo de bases de datos, los datos se almacenan en una serie de tablas, con filas que representan registros individuales y columnas que representan atributos o características de esos registros.

3. ¿Qué es el modelo entidad relación y/o diagrama entidad relación?

R. El modelo entidad-relación y/o diagrama entidad-relación es una herramienta gráfica utilizada en la modelación de bases de datos que representa las entidades, atributos y relaciones entre entidades en un sistema de información.

4. ¿Cuáles son las figuras que representan a un diagrama entidad relación? Explique cada una de ellas.

R. En un diagrama entidad-relación, se utilizan diferentes figuras para representar las entidades, atributos y relaciones entre entidades. A continuación se explican cada una de ellas:

1. Entidad: representada por un rectángulo, es la estructura principal del diagrama y define un objeto o concepto del mundo real, como una persona, un producto o una orden. Por ejemplo, si se modela una base de datos de una tienda, una entidad podría ser “Cliente” o “Producto”.

2. Atributo: representado por un óvalo, son las características o propiedades de una entidad. Cada entidad puede tener uno o varios atributos asociados que describen las características específicas de esa entidad. Por ejemplo, en el caso de la entidad “Producto”, los atributos podrían ser el nombre, el precio y la descripción.

3.Relación: representada por una línea que conecta dos entidades, indica la asociación o interacción entre dos o más entidades. Existen diferentes tipos de relaciones que se pueden representar, como una relación uno a uno, uno a muchos o muchos a muchos. Por ejemplo, en una base de datos de una tienda, se puede tener una relación entre las entidades “Cliente” y “Orden de Compra”, donde un cliente puede realizar varias órdenes de compra.

4.Cardinalidad: es una notación utilizada para indicar la cantidad de instancias que pueden participar en una relación. Se utiliza un símbolo para representar la cardinalidad en cada extremo de la línea de relación. Por ejemplo, el símbolo (1, N) indica que una entidad puede tener una relación de uno a muchos con otra entidad.

5.Restricciones: se utilizan símbolos adicionales para representar restricciones o reglas específicas en una relación. Por ejemplo, el símbolo de doble línea indica una restricción de participación total, lo que significa que todas las instancias de una entidad deben estar relacionadas con al menos una instancia de la otra entidad.

Estas figuras y notaciones se utilizan en conjunto para construir un diagrama entidad-relación que represente de manera clara y precisa la estructura y relaciones de una base de datos.

5. ¿Qué es SQL Server y qué es SQL Server Management Studio?

R. SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Es utilizado para almacenar, administrar y recuperar datos de manera eficiente y segura. SQL Server utiliza el lenguaje de consultas estructurado (SQL) para interactuar con la base de datos.

SQL Server Management Studio (SSMS) es una interfaz gráfica de usuario utilizada para administrar y configurar instancias de SQL Server. Proporciona herramientas y características avanzadas para administrar bases de datos, diseñar y ejecutar consultas, realizar copias de seguridad y restaurar bases de datos, programar tareas, monitorear el rendimiento y mucho más. SSMS es una herramienta clave para desarrolladores, administradores de bases de datos y otros profesionales de TI que trabajan con SQL Server.

6.¿Cómo se crea una base de datos?

R. Para crear una base de datos, se pueden seguir los siguientes pasos:

- 1.Definir los requisitos: Determinar cuál será el propósito de la base de datos, qué tipo de información se almacenará, qué relaciones existen entre los datos, etc.
- 2.Elegir un sistema de gestión de bases de datos (DBMS): Hay diferentes opciones disponibles, como MySQL, Oracle, Microsoft SQL Server, PostgreSQL, entre otros.
- 3.Instalar y configurar el DBMS: Descargar e instalar el software del DBMS elegido, y configurarlo según las necesidades del proyecto.
- 4.Diseñar el esquema de la base de datos: Crear un diseño conceptual de la estructura de la base de datos, definiendo las tablas, campos, relaciones y restricciones necesarias.
- 5.Crear las tablas: Utilizando el lenguaje de definición de datos (DDL) del DBMS, crear las tablas necesarias para almacenar la información.
- 6.Definir los campos y tipos de datos: Para cada tabla, especificar los campos y los tipos de datos que se utilizarán para almacenar la información.
- 7.Establecer relaciones entre tablas: Si hay relaciones entre las tablas, como claves primarias y foráneas, definir esas relaciones utilizando el DDL.
- 8.Crear índices: Si es necesario, crear índices para acelerar las consultas en la base de datos.
- 9.Establecer restricciones de integridad: Definir las reglas que aseguren la integridad de los datos, como restricciones de clave primaria, claves foráneas, reglas de validación, etc.
- 10.Crear usuarios y asignar permisos: Si se requiere, crear usuarios en la base de datos y asignarles los permisos necesarios para acceder y manipular los datos.

Una vez completados estos pasos, la base de datos estará lista para ser utilizada y se podrán realizar consultas, inserciones, actualizaciones y eliminaciones de datos según sea necesario.

7. ¿Para qué sirve el comando USE?

R. El comando USE se utiliza en lenguajes de consulta, como SQL, para cambiar la base de datos en la que se está trabajando. Al usar el comando USE seguido del nombre de la base de datos deseada, el sistema cambiará el contexto de trabajo actual a la base de datos seleccionada. Esto permite realizar consultas y manipulaciones de datos en la base de datos deseada. En resumen, el comando USE sirve para seleccionar la base de datos en la que se quiere trabajar.

8. Crear una tabla cualquiera con 3 columnas y su primary key.

R.

```
```SQL
```

```
CREATE TABLE nombre_tabla (
```

```
 id INT PRIMARY KEY,
```

```
 columna1 VARCHAR(255),
```

```
 columna2 INT,
```

```
 columna3 DECIMAL(10,2)
```

```
);
```

```
```
```

9. Insertar 3 registros a la tabla creada anteriormente.

R.

```
```SQL
```

```
INSERT INTO nombre_tabla (id, columna1, columna2, columna3) VALUES (1, 'valor1', 10, 3.5);
```

```
INSERT INTO nombre_tabla (id, columna1, columna2, columna3) VALUES (2, 'valor2', 20, 5.2);
```

```
INSERT INTO nombre_tabla (id, columna1, columna2, columna3) VALUES (3, 'valor3', 30, 8.9);
```

```
```
```

10.¿Cómo se elimina una tabla?

R. Para eliminar una tabla en un sistema de gestión de bases de datos, se debe utilizar la sentencia SQL DROP TABLE.

La sintaxis para eliminar una tabla en SQL es:

```
DROP TABLE nombre_tabla;
```

Por ejemplo, si se quiere eliminar una tabla llamada “clientes”, se debe ejecutar la siguiente sentencia:

```
DROP TABLE clientes;
```

Es importante tener en cuenta que esta operación es irreversible y elimina por completo la tabla y todos sus datos. Por lo tanto, se recomienda realizar una copia de seguridad de la información antes de eliminar una tabla.

...