
EVALUACION DE APRENDIZAJE



Alumno: Tomas Santiago Garcia

DNI: 42231475

Materia: Base de datos II

Email: tomas00garcia@gmail.com

¿En qué forma normal se encuentran las tablas? ¿Por qué?

Las tablas están en primera forma normal porque no presentan grupo repetidos pero no cumple con la segunda forma normal ya que tiene campos no clave que dependen de mas de un campo clave.

1) Crear la base de datos llamada Biblioteca:

A. 2 archivos de datos de 20 mb cada uno (biblio_data1 / biblio_data2)

B. 2 archivos de log de 10 mb cada uno(biblio_log1/biblio_log2)

```
CREATE DATABASE Biblioteca
ON (NAME = 'biblio_data1',
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\biblio_data1.mdf',
    SIZE = 20 MB,
    FILEGROWTH = 0),
(NAME = 'biblio_data2',
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\biblio_data2.mdf',
    SIZE = 20 MB,
    FILEGROWTH = 0)
LOG ON (NAME = 'biblio_log1',
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\biblio_log1.ldf',
    SIZE = 20 MB,
    FILEGROWTH = 0),
(NAME = 'biblio_log2',
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\biblio_log2.ldf',
    SIZE = 20 MB,
    FILEGROWTH = 0)
```

2) Crear el esquema “Biblioteca”.

```
CREATE SCHEMA Biblioteca;
```

```
CREATE TABLE Biblioteca.Cliente(
    codigoCliente INT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    CONSTRAINT PK_Cliente PRIMARY KEY CLUSTERED(codigoCliente)
);
```

```
CREATE TABLE Biblioteca.Prestamo(
    numeroPrestamo INT NOT NULL IDENTITY(1,1),
    fechaPrestamo DATE NOT NULL,
    fechaDevolucion DATE NOT NULL,
    codigoCliente INT NOT NULL,
    CONSTRAINT PK_Prestamo PRIMARY KEY CLUSTERED(numeroPrestamo),
    CONSTRAINT FK_Prestamo_Cliente FOREIGN KEY(codigoCliente)
REFERENCES Biblioteca.Cliente
);
```

```
CREATE TABLE Biblioteca.Libro(
    codigoLibro INT NOT NULL IDENTITY(1,1),
    titulo VARCHAR(200) NOT NULL,
    editorial VARCHAR(100) NOT NULL,
    CONSTRAINT PK_Libro PRIMARY KEY CLUSTERED(codigoLibro)
);
```

```
CREATE TABLE Biblioteca.LibroPrestamo(
codigoLibro INT NOT NULL,
numeroPrestamo INT NOT NULL,
CONSTRAINT PK_LibroPrestamo PRIMARY KEY CLUSTERED(codigoLibro,numeroPrestamo),
CONSTRAINT FK_LibroPrestamo_Libro FOREIGN KEY(codigoLibro) REFERENCES
Biblioteca.Libro,
CONSTRAINT FK_LibroPrestamo_Prestamo FOREIGN KEY(numeroPrestamo) REFERENCES
Biblioteca.Prestamo
);
```

```
CREATE TABLE Biblioteca.Autor(
codigoAutor INT NOT NULL IDENTITY(1,1),
nombre VARCHAR(100) NOT NULL,
apellido VARCHAR(100) NOT NULL,
CONSTRAINT PK_Autor PRIMARY KEY CLUSTERED(codigoAutor)
);
```

```
CREATE TABLE Biblioteca.LibroAutor(
codigoAutor INT NOT NULL,
codigoLibro INT NOT NULL,
CONSTRAINT PK_LibroAutor PRIMARY KEY CLUSTERED(codigoAutor,codigoLibro),
CONSTRAINT FK_LibroAutor_Autor FOREIGN KEY(codigoAutor) REFERENCES
Biblioteca.Autor,
CONSTRAINT FK_LibroAutor_Libro FOREIGN KEY(codigoLibro) REFERENCES
Biblioteca.Libro
);
```

3) Generar las filas en cada una de las tablas con la sentencia Insert. Los datos ingresados serán a efectos de prueba.

```
INSERT INTO Biblioteca.Cliente VALUES
('Pedro','Perez'),
('Jose','Lopez'),
('Martin','Rodriguez');
```

```
INSERT INTO Biblioteca.Prestamo VALUES
('01-02-2020','01-03-2020',1),
('01-03-2020','01-04-2020',2),
('01-05-2020','01-05-2020',3);
```

```
INSERT INTO Biblioteca.Libro VALUES
('Harry Potter','Iguana'),
('Harry Potter 2','Tigre'),
('Harry potter 3','Oveja');
```

```
INSERT INTO Biblioteca.LibroPrestamo VALUES
(1,1),
(1,2),
(2,2),
(3,1);
```

```
INSERT INTO Biblioteca.Autor VALUES
('Enrique','Perez'),
('Belen','Cabral'),
('Micaela','Sosa');
```

```
INSERT INTO Biblioteca.LibroAutor VALUES
(1,1),
(1,2),
(2,3),
(3,1);
```

4) Agregar el campo importe a la tabla ¿Libro?, con valor por defecto 0 (cero).

```
ALTER TABLE Biblioteca.Libro ADD importe DECIMAL NOT NULL  
CONSTRAINT DF_Libro_importe DEFAULT(0);
```

5) Agregar el campo stock en la tabla de ¿Libro?

```
ALTER TABLE Biblioteca.Libro ADD stock INT NULL;
```

6) Agregar el campo cuit en la tabla ¿Cliente? El campo debe ser único.

```
ALTER TABLE Biblioteca.Cliente ADD cuit INT NULL;  
ALTER TABLE Biblioteca.Cliente ADD CONSTRAINT UN_Cliente_cuit UNIQUE (cuit);
```

7) Listar todos los prestamos realizados. Mostrar datos del préstamo, nombre y apellido de la persona, datos del libro, datos de la editorial y datos del autor.

```
SELECT  
P.numeroPrestamo,P.fechaPrestamo,P.fechaDevolucion,C.nombre,C.apellido,L.titulo,L  
.importe,L.editorial,L.stock,A.nombre,A.apellido  
FROM Biblioteca.Prestamo AS P  
INNER JOIN Biblioteca.Cliente AS C ON P.codigoCliente=C.codigoCliente  
INNER JOIN Biblioteca.LibroPrestamo AS LP ON LP.numeroPrestamo=P.numeroPrestamo  
INNER JOIN Biblioteca.Libro AS L ON LP.codigoLibro =L.codigoLibro  
INNER JOIN Biblioteca.LibroAutor AS LA ON L.codigoLibro=LA.codigoLibro  
INNER JOIN Biblioteca.Autor AS A ON LA.codigoAutor=A.codigoAutor
```

8) Listar todos los libros que alguna vez fueron prestados

```
SELECT * FROM Biblioteca.Libro AS L  
WHERE L.codigoLibro IN(SELECT codigoLibro FROM Biblioteca.LibroPrestamo)
```

9) Listar los libros con sus respectivos autores.

```
SELECT L.codigoLibro,L.titulo,A.nombre,A.apellido FROM Biblioteca.Libro AS L  
INNER JOIN Biblioteca.LibroAutor AS LA ON L.codigoLibro=LA.codigoLibro  
INNER JOIN Biblioteca.Autor AS A ON LA.codigoAutor=A.codigoAutor  
ORDER BY L.codigoLibro ASC
```

10) Listar los libros que fueron escritos por dos o más autores.

```
SELECT * FROM Biblioteca.Libro AS A  
WHERE A.codigoLibro IN(SELECT codigoLibro FROM Biblioteca.LibroAutor GROUP BY  
codigoLibro HAVING COUNT(*)>=2)
```

11) Listar los clientes que pidieron la mayor cantidad de libros y su apellido contenga la letra 'u'.

```
SELECT c.codigoCliente,c.nombre,c.apellido,COUNT(lp.codigoLibro) AS retirados  
FROM Biblioteca.Cliente AS c  
INNER JOIN Biblioteca.Prestamo AS p  
ON p.codigoCliente=c.codigoCliente  
INNER JOIN Biblioteca.LibroPrestamo AS lp  
ON p.numeroPrestamo=lp.numeroPrestamo  
WHERE c.apellido LIKE '%u%'  
GROUP BY c.codigoCliente,c.nombre,c.apellido
```

```

HAVING COUNT(lp.codigoLibro)=(
    SELECT MAX(maximo.cantidad) FROM(
        SELECT COUNT(codigoLibro) AS cantidad FROM Biblioteca.LibroPrestamo
    AS lp
        INNER JOIN Biblioteca.Prestamo AS p
        ON p.numeroPrestamo=lp.numeroPrestamo
        INNER JOIN Biblioteca.Cliente AS c
        ON p.codigoCliente=c.codigoCliente
        GROUP BY c.codigoCliente
    ) AS maximo
);

```

12) Listar los autores cuyos libros han sido prestados por más de dos días.

```

SELECT A.nombre,A.apellido FROM Biblioteca.Autor AS A
INNER JOIN Biblioteca.LibroAutor AS LA ON A.codigoAutor=LA.codigoAutor
INNER JOIN Biblioteca.LibroPrestamo AS LP ON LA.codigoLibro=LP.codigoLibro
INNER JOIN Biblioteca.Prestamo AS P ON LP.numeroPrestamo=P.numeroPrestamo
WHERE DATEDIFF(DAY,P.fechaPrestamo,P.fechaDevolucion)>2
GROUP BY A.nombre,A.apellido

```

13) Defina brevemente la diferencia entre un índice agrupado y no agrupado. ¿Qué sucede si no defino un índice?

En un índice agrupado los registros con el mismo valor de campo se agrupan juntos, de esta forma se determina la secuencia de almacenamiento de registros de una tabla. Una tabla solo puede tener un índice agrupado.

En cambio un índice no agrupado, como dice su nombre, no esta agrupado con los registros de un campo, sino que se almacena en un lugar distinto, funciona como un índice con punteros que indican el lugar de almacenamiento de los elementos indexados en la tabla.

Al no definir índices se estaría perdiendo la posibilidad de facilitar la obtención de información de una tabla, haciendo menos eficiente las búsquedas ya que SQL Server debe recorrer secuencialmente toda la tabla para recorrer un registro.

14)Defina qué sucede cuando se crea una clave primaria. ¿Donde residen los datos?

SQL Server crea automáticamente un índice agrupado único cuando crea una clave primaria, este al ser clustered reside en la tabla donde se definió.