

TP 2 Parte B

Defina y ejemplifique los siguientes conceptos:

⑩ Base de Datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

⑩ Tabla de base de datos

Tabla en las bases de datos, se refiere al tipo de modelado de datos, donde se guardan y almacenan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de hoja de cálculo.

⑩ Tupla (Registro de una tabla)

Un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla.

Es cada una de las filas en que se divide la tabla. Cada registro contiene datos de los mismos tipos que los demás registros. Ejemplo: en una tabla de nombres, direcciones, etc, cada fila contendrá un nombre y una dirección.

⑩ Clave Primaria

La clave primaria se utiliza para identificar en forma única cada línea en la tabla. Puede ser parte de un registro real, o puede ser un campo artificial (uno que no tiene nada que ver con el registro real). Una clave primaria puede consistir en uno o más campos en una tabla.

Ej: DNI en una tabla de Alumnos.

⑩ Clave Foránea

La clave externa o FOREIGN KEY, es una columna o varias columnas, que sirven para señalar cual es la clave primaria de otra tabla. La columna o columnas señaladas como FOREIGN KEY, solo podrán tener valores que ya existan en la clave primaria PRIMARY KEY de la otra tabla.

Ej:

ID_USUARIO	NOMBRE
1	Nicolas
2	Gustavo
3	Pablo

TABLA USUARIOS

ID_FACTURA	DESCRIPCION	ID_USUARIO
1	COMPRA HELADERA	1
2	COMPRA COMPUTADORA	1
3	COMPRA IMPRESORA	1

TABLA FACTURAS

En el ejemplo anterior el ID_USUARIO es **FOREIGN KEY** de la tabla FACTURAS.

⑩ Tipos de Restricciones

Las **CONSTRAINTS** son restricciones que se utilizan para limitar el tipo de dato que puede recibir una columna de una tabla.

Las restricciones se puede definir cuando creamos la tabla (CREATE TABLE) o posteriormente con la sentencia ALTER TABLE.

Las posibles restricciones son:

- ⑩ NOT NULL
- ⑩ UNIQUE
- ⑩ PRIMARY KEY
- ⑩ FOREIGN KEY
- ⑩ CHECK
- ⑩ DEFAULT

⑩ Normalización de base de Datos: Primera, Segunda y Tercera Formas Normales

Existen 3 niveles de Normalización que deben respetarse para poder decir que nuestra Base de Datos, se encuentra NORMALIZADA, es decir, que cumple con los requisitos naturales para funcionar óptimamente y no perjudicar la performance por mala arquitectura. Estas 3 reglas de Normalización se las conoce como las 3 **FORMAS NORMALES**.

La Primera Forma Normal Esta primera Forma Normal, nos lleva a no repetir datos en nuestras tablas. Los famosos maestro – detalle, deben aplicarse a la estructura de la tabla. Si nuestra tabla de ventas repite una y otra vez (por cada venta) , el nombre, el domicilio y otros datos del Cliente, es que no hemos aplicado esta Normalización. Si tenemos una tabla clientes, en la tabla ventas, solo debería figurar el código del cliente, para que el resto de los datos se puedan referenciar automáticamente sin problemas y sin duplicar información. Lo mismo ocurriría en una tabla de detalle de ventas, si por cada ítem vendido colocamos el detalle del producto, con su descripción , medidas, etc...Tendríamos un desaprovechamiento de espacio y recursos muy grande. Para ello, tendremos nuestra tabla maestra de Productos y con solo grabar el código de dicho producto en nuestra tabla de ventas, será suficiente.

La Segunda Forma Normal (Si o si debe estar previamente aplicada la Primera Forma Normal) La Segunda Forma Normal nos habla de que cada columna de la tabla debe depender de la clave. Esto significa que todo un registro debe depender únicamente de la clave principal, si tuviéramos alguna columna que se repite a lo largo de todos los registros, dichos datos deberían atomizarse en una nueva tabla. Veamos un ejemplo

VentaID	ItemID	FechaVenta	ClienteVenta	Productoid	Cantidad
---------	--------	------------	--------------	------------	----------

1	1	01/12/2007	2	2334	10
1	2	01/12/2007	2	3333	2
1	3	01/12/2007	2	66643	34
1	4	01/12/2007	2	21	3
2	1	02/12/2007	5	3566	6

Ahí tenemos un claro problema !!!Acaso no se busca **NO REPETIR DATOS**? Si toda una venta tendrá el mismo numero de Cliente y la misma Fecha...Por que no crear una Tabla de **MAESTRO DE VENTAS** y que contenga esos 2 datos ?Es evidente que la columna **ClienteVenta** y **FechaVenta** se repetirán por cada venta realizada. Es por ello que proponemos el siguiente esquema

VentaID	ItemID	ProductID	Cantidad
1	1	2334	10
1	2	3333	2
1	3	66643	34
1	4	21	3
2	1	3566	6

Y ahora nuestra nueva tabla maestra

VentaID	FechaVenta	ClienteVenta
1	01/12/2007	2
2	02/12/2007	5

Entonces, nuestra 2da Forma Normal nos habla de que cada columna de una tabla debe depender de toda la clave y no constituir un dato único para cada grupo de registros.

La Tercera Forma Normal En realidad si nos guiamos en el ejemplo de esta nota, ya no quedaría normalización por aplicar y podríamos decir que nuestro ejemplo cumple con las 3 formas normales, ya que la 3ra Forma Normal nos habla de que :

1. Ninguna Columna puede depender de una columna que no tenga una clave
2. No puede haber datos derivados

En el 2do ejemplo hemos descubierto campos que dependían de la clave principal (VentaID) y que podrían incluirse en una tabla maestra. Pero supongamos un ejemplo donde ciertas columnas no dependen de la clave principal y si dependen de una columna de nuestra tabla.

VentaID	ItemID	ProductID	Cantidad	Descripcion	Medida	Proveedor
1	1	3455	12	Impresora HP LJ8000	122cm	1
1	2	2455	34	Scanner HP A3555	33cm	1
2	1	5444	21	Mouse HP Wireless	—	1

Esto es muy normal encontrar en bases mal normalizadas. Vemos que los campos DESCRIPCIÓN , MEDIDA y PROVEEDOR no dependen de VENTAID y es por ello que no deberían estar dentro de la tabla de detalle de ventas, ya que dependen de PRODUCTOID. Aquí no se trata ya de eliminar grupos repetidos de datos (1ra Forma Normal) sino que ante la inclusión de una clave perteneciente a otra tabla, cualquier campo que sea subordinado de dicha clave debe estar en otra tabla y no en nuestra tabla detalle.

⑩ Definición de SQL

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje específico del dominio utilizado en programación; y diseñado para administrar sistemas de gestión de bases de datos relacionales.

⑩ Sentencia SELECT

Una de las sentencias SQL más importantes es SELECT, ya que permite realizar consultas sobre los datos almacenados en la base de datos.

Sintaxis SQL SELECT

```
SELECT * FROM nombretabla  
SELECT columna1, columna2 FROM nombretabla
```

⑩ Sentencia INSERT

La sentencia INSERT INTO se utiliza para insertar nuevas filas en una tabla.

Es posible insertar una nueva fila en una tabla de dos formas distintas:

```
INSERT INTO nombre_tabla  
VALUES (valor1, valor2, valor3, .)  
  
INSERT INTO nombre_tabla (columna1, columna2, columna3,.)  
VALUES (valor1, valor2, valor3, .)
```

⑩ Sentencia UPDATE

*La sentencia **UPDATE** se utiliza para modificar valores en una tabla.*

La sintaxis de SQL UPDATE es:

```
UPDATE nombre_tabla  
SET columna1 = valor1, columna2 = valor2  
WHERE columna3 = valor3
```

La cláusula SET establece los nuevos valores para las columnas indicadas.

La cláusula WHERE sirve para seleccionar las filas que queremos modificar.

Ojo: Si omitimos la cláusula WHERE, por defecto, modificará los valores en todas las filas de la tabla.

⑩ Sentencia DELETE

La sentencia DELETE sirve para borrar filas de una tabla.

La sintaxis de SQL DELETE es:

```
DELETE FROM nombre_tabla  
WHERE nombre_columna = valor
```

⑩ ¿Qué es y cómo funciona la cláusula INNER JOIN de SQL?

La sentencia INNER JOIN es la sentencia JOIN por defecto, y consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición.

```
SELECT * FROM tabla1 INNER JOIN tabla2 WHERE tabla1.columna1 = tabla2.columna1
```