

CONCEPTOS BÁSICOS DE BASES DE DATOS

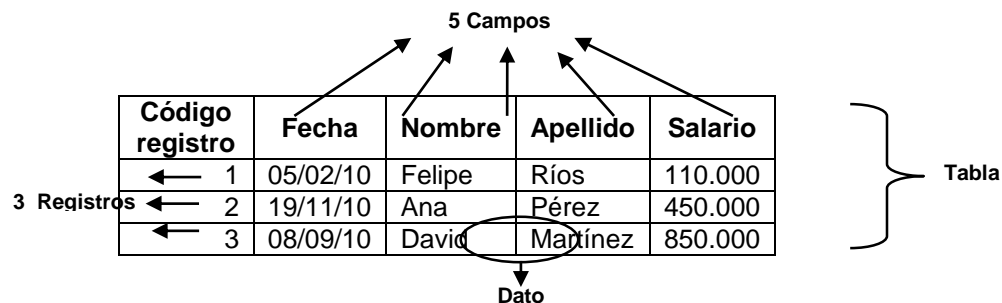
Dato: Son hechos, observaciones o sucesos del mundo real que pueden tomar la forma de números, letras, símbolos o señales. Ejemplos: 4, Medellín, David, 300.000, 05/06/2005, sí, verdadero.

Información: Es el resultado de procesar los datos.

Campo: Es una porción de memoria. También se llama Columna o Atributo. Es donde se guardan los datos. Los campos tienen determinado tipo: Numérico, alfabético, fecha, entre otros.

Registro: Es un conjunto de campos. También se llama Fila o Tupla. Son varios datos

Tabla: Es una colección de datos relacionados con un tema específico. Es un conjunto de registros, todos del mismo tipo que guardan alguna relación entre sí.



Base de Datos: Es un conjunto de tablas relacionadas entre sí.

Clave Principal o Primaria: Uno o mas campos que hacen único un registro

SGDB: (DataBase Management System – Sistema administrador de bases de datos). Son un tipo de software muy específico, dedicado a servir de interfaz entre La base de datos, El usuario y las aplicaciones que la utilizan.

Access es un DBMS. Otros DMBS utilizados son: SQL server, Oracle, MySQL, entre otros.

NORMALIZACIÓN

Como hemos visto, los datos se almacenan en registros que a su vez forman tablas. Para determinar cómo están conformados los registros de cada tabla es necesario hacer un estudio de los datos y cómo podemos agruparlos en diferentes tablas para que su almacenamiento sea óptimo, este procedimiento se denomina **normalización**.

La normalización se puede definir como el examen a los almacenamientos de datos y a su reagrupación de tal forma que permitan de una manera eficiente el diseño y administración del sistema de información. Con este proceso se obtiene menor redundancia en los datos (evitar que un mismo dato se almacena varias veces innecesariamente) y una mayor integridad de los mismos (que un dato en particular no contenga diferentes valores para un mismo objeto). Para lograr una estructura normalizada (en tercera forma normal) se debe cumplir el siguiente proceso:

Para normalizar una estructura de almacenamiento, lo primero que se debe hacer es definir cuál es el objeto a almacenar (recibo, factura, remisión, cita médica, historia clínica, etc.) y definir su nombre, luego identificar que datos (campos) se deben almacenar de este objeto.

Antes de empezar el proceso de normalización, se sugiere tener varias entrevistas con la persona responsable de entregar la información y solicitar formatos (impresos o digitales) de la forma como se obtienen los datos.

Por ejemplo, si se quisiera almacenar facturas de venta, estaríamos definiendo inicialmente el objeto factura de la siguiente forma:

PAPELERIA LA NUEVA ERA				
NUMERO DE FACTURA				
FECHA:			CODIGO DEL CLIENTE:	
DESCRIPCION DEL CLIENTE:			DIRECCION DEL CLIENTE:	
FORMA DE PAGO:			FECHA DE VENCIMIENTO:	
CODIGO DEL PRODUCTO	DESCRIPCION	PRECIO UNITARIO	UNIDADES VENDIDAS	TOTAL POR PRODUCTO
OBSERVACIONES			BRUTO PAGAR	A
			DESCUENTO	
			IVA	
			NETO A PAGAR	

Inicialmente la información se recolecta de la siguiente manera, dando un nombre genérico a una tabla, con base en la información que se observa en el formato (impreso o digital). Para el ejemplo como el formato es de una factura se sugiere FACTURA aunque después se pueda cambiar. Los nombres de los campos se separan con el signo + (más) y estos a su vez se encierran entre llaves {}

FACTURA = {Número de la factura + Fecha + Código del cliente + Descripción del cliente + Dirección del cliente + Forma de pago + Fecha de vencimiento + {Código del producto + Descripción + Precio unitario + Unidades vendidas + Total por producto} + Bruto a pagar + Descuento + IVA + Neto pagar + Observaciones}

Observe que el grupo repetitivo, es decir, los campos que hacen que se repita la información, se encierran entre llaves (color rojo)

Luego se procede a aplicar los siguientes pasos:

PRIMERA FORMA NORMAL

Se identifica la clave primaria y se define en forma única. Esta clave primaria consta de uno o más datos que identifican unívocamente la ocurrencia (registro, tupla o relación), esta clave se identifica subrayándola. Esto quiere decir que no debe existir ninguna otra ocurrencia con la misma clave o identificación. Como se puede observar en el ejemplo, el número de la factura siempre y cuando se lleven los respectivos controles, será la clave primaria por ser único.

Adicionalmente se remueven (pasar a otra tabla) todos los grupos repetitivos de campos, es decir, los campos que hacen que se repita la información y probablemente se conforman con ellos otras

tablas. En el ejemplo, vemos que los campos: {Código del producto + Descripción + Precio unitario + Unidades + Total por producto} para una factura pueden tomar varios datos diferentes

Por lo tanto, el almacenamiento original quedaría de la siguiente forma:

FACTURA = {Número de la factura + Fecha + Código del cliente + Descripción del cliente + Dirección del cliente + Forma de pago + Fecha de vencimiento + Bruto a pagar + Descuento + IVA + Neto a pagar + observaciones}

LINEA_FACTURA = {Número de la factura + Código del producto + Descripción + Precio unitario + Unidades + Total por producto}

Cuando se crean nuevas tablas al aplicar la primera forma normal, a estas se les debe adicionar el campo clave de la tabla origen, y generalmente, en la nueva tabla, también será clave junto con otro u otros campos de la nueva tabla. En este ejemplo es el número de la factura. Esta clave la leeríamos de la siguiente manera: Un número de factura no debe tener relacionado, dos o más veces, el mismo código del producto. Tenga en cuenta que la tabla LINEA_FACTURA puede tener otros nombres, tales como: ARTICULOSXFACTURA o ITEMS_FACTURA. Todo depende del contexto o la forma como nos estén dando la información.

SEGUNDA FORMA NORMAL

Luego de haber obtenido la primera forma normal se logra la segunda forma normal cuando **todos los campos dependen funcionalmente (son determinados) de la clave primaria**, esto es que todos los campos sean una propiedad directa del objeto de almacenamiento, los que no cumplan esta condición se deben remover (pasar a otra tabla) y probablemente con ellos formar nuevas tablas.

En el ejemplo, la descripción del cliente y su dirección no dependen del número de factura (clave principal), sino del código del cliente de la factura, mientras el código del cliente puede ser importante en este almacenamiento para reconocer el responsable de la factura, con estos campos, que no dependen funcionalmente de la clave primaria, se crea un nuevo almacenamiento. Se puede afirmar algo similar de la descripción del producto y su precio cuando es único, en la tabla LINEA_FACTURA. Las estructuras quedarían así:

FACTURA = {Número de la factura + Fecha + Código del cliente + Forma de pago + Fecha de vencimiento + Bruto a pagar + Descuento + IVA + Neto a pagar + observaciones}

LINEA_FACTURA = {Número de la factura + Código del producto + Unidades + Total por producto}

CLIENTE = {Código del cliente + Descripción del cliente + Dirección del cliente}

PRODUCTO = {Código del producto + Descripción del producto + Precio unitario}

Nótese que en el almacenamiento original se deja el campo que será clave en el nuevo almacenamiento (código cliente en el almacenamiento FACTURA y código del producto en el almacenamiento LINEA_FACTURA)

TERCERA FORMA NORMAL

Una vez lograda la segunda forma normal se obtiene la tercera eliminando toda dependencia transitiva de los campos respecto de la clave primaria. Esto es, **todo campo que pueda ser calculado a partir de otros datos almacenados en la base de datos se debe eliminar para evitar problemas de integridad**. En el ejemplo, el Bruto a pagar (es la suma de los totales por producto), el IVA (suponiendo que todos los productos tienen el mismo porcentaje) y el neto a pagar del almacenamiento FACTURAS, así como el total por producto (Precio Unitario X Unidades) de LINEA_FACTURA, pueden ser calculados y por lo tanto eliminados del almacenamiento así:

FACTURA = {Número de la factura + Fecha + Código del cliente + Forma de pago + Fecha de vencimiento + Descuento + observaciones}
LINEA_FACTURA = {Número de la factura + Código del producto + Unidades}
CLIENTE = {Código del cliente + Descripción del cliente + Dirección del cliente}
PRODUCTO = {Código del producto + Descripción del producto + Precio unitario}

Es importante anotar que un buen funcionamiento de un sistema depende fundamentalmente de la estructura de datos, más que de cualquier otro factor.

APLICACIÓN PRÁCTICA DE NORMALIZACIÓN (EXTRACLASE)

1. Para una concesionaria de automóviles, se determina que se deben almacenar los siguientes datos por cada venta que se realiza:
Número de la venta (NumVen), fecha de la venta (FecVen), placa del vehículo vendido (pueden ser varios) (PlaVeh), marca del vehículo vendido (MarVeh), color del vehículo vendido (ColVeh), tipo del vehículo vendido (TipVeh), modelo del vehículo vendido (ModVeh), valor del vehículo vendido (ValVeh), código del vendedor (CodVen), nombre del vendedor (NomVen), dirección del vendedor (DirVen), teléfono del vendedor (TelVen), valor bruto de la venta (ValBru), valor del IVA de la venta (Vallva), valor neto de la venta que es el bruto + el Iva (ValNet).
2. En un centro médico se desea almacenar la información que se genera en cada una de las consultas que realiza, para ello se define que los datos son:
Número de la consulta, fecha de la consulta, código del médico que realiza la consulta, nombre del médico que realiza la consulta, dirección del médico que realiza la consulta, teléfono del médico que realiza la consulta, especialidad del médico que realiza la consulta, cédula del paciente, nombre del paciente, dirección del paciente, teléfono del paciente, código de la EPS a la que pertenece el paciente, nombre de la EPS a la que pertenece el paciente, dirección de la oficina de la EPS a la que pertenece el paciente, teléfono de la EPS a la que pertenece el paciente, diagnóstico que hace el médico en la consulta, consultorio en que se hace la consulta, valor de la consulta
3. En un centro de estudios se desea almacenar la información de las matrículas de sus alumnos así:
Número de la matrícula, fecha de la matrícula, cédula del estudiante, nombre del estudiante, dirección del estudiante, teléfono del estudiante, estrato del estudiante, código de las materias matriculadas, nombre de las materias matriculadas, nivel de las materias matriculadas, valor de cada una de las materias matriculadas, grupo de las materias matriculadas, cédula del profesor de las materias matriculadas, nombre del profesor de las materias matriculadas, dirección del profesor de las materias matriculadas, teléfono del profesor de las materias matriculadas, Fecha de los pagos que debe realizar el alumno, valor de cada uno de los pagos que debe realizar el alumno

4. En una empresa de transporte urbano de pajareros, se desea almacenar la información de cada uno de los viajes que realizan los buses, los datos a almacenar son:
placa del bus, marca del bus, capacidad del bus (número de pasajeros), número del viaje, fecha del viaje, hora del viaje, ruta del viaje (A, B, o C), valor del pasaje (depende de la ruta), número de pasajeros transportados, valor de los gastos del viaje, total producido, valor neto producido, cédula del conductor (un bus solo tiene un conductor asignado), nombre del conductor, dirección del conductor, teléfono del conductor, cédula del propietario (un bus solo tiene un propietario), nombre del propietario, dirección del propietario, teléfono del propietario
5. El departamento de personal de una empresa desea crear una base de datos con la información de sus empleados, los datos a almacenar por cada empleado son:
Cédula del empleado, nombre del empleado, sexo del empleado, fecha nacimiento del empleado, fecha de ingreso a la empresa del empleado, salario básico del empleado, cédula de la esposa (o) del empleado, nombre de la esposa (o) del empleado, fecha de nacimiento de la esposa (o) del empleado, número de identificación de los hijos del empleado, nombre de los hijos del empleado, fecha de nacimiento de los hijos del empleado, sexo de los hijos del empleado.

MODELO RELACIONAL

Una base de datos relacional permite la utilización simultánea de datos procedentes de más de una tabla.

Al hacer uso de las relaciones, se evita la duplicidad de datos, ahorrando memoria y espacio en el disco, aumentando la velocidad de ejecución y facilitando al usuario el trabajo con tablas.

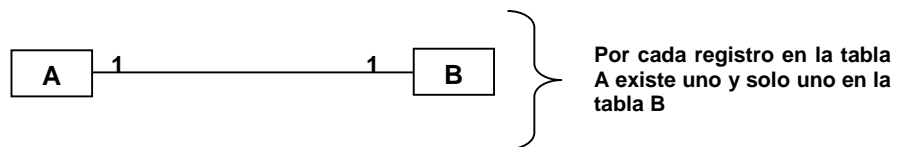
Para conseguir una correcta base de datos relacional es imprescindible realizar un estudio previo del diseño de la base de datos.

Para poder relacionar tablas entre sí, se deberá especificar un **campo en común que contenga el mismo valor en las dos tablas y dicho campo será clave principal en una de ellas**. Las tablas se relacionan de dos a dos, donde una de ellas será la tabla principal de la que parte la relación y la otra será la tabla secundaria o destino de la relación.

TIPOS DE RELACIONES

Relación Uno a Uno

Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa.



Por ejemplo: Tenemos dos tablas una con los datos de Ciudades y otra con una lista de Alcaldes, una Ciudad sólo puede tener un alcalde, y un alcalde lo será únicamente de una Ciudad.

Relación Uno a Varios

Cuando un registro de una tabla (tabla principal) puede tener más de

