**TEMA 2 – BASES DE DATOS RELACIONALES**

1. **MODELO DE DATOS**

**Modelo**, el esquema teórico, generalmente en forma matemática, de un sistema o de una relaidad compleja.

**Modelo de Datos**, un conjunto de métodos y reglas que indican cómo se ha de almacenar la información y cómo se han de manipular los datos.

Se implementa con un lenguaje utilizado para la descripción de una Bda, con este vamos a poder describir las estructuras de los datos, las restricciones de integridad y las operaciones de manipulación de los datos.

Este lenguaje presnta 2 subleguajes:

* **Data Defenition Language (DDL)**, su función es describir, de forma abstracta, las estructuras de datos y las restricciones de integridad.
* **Data Manipulation Language (DML)**, sirve para describir las operaciones de manipulación de los datos.

Existen 3 **fases de modelado** en el diseño de una Bda en el nivel de abstracción:

**Modelo de Datos Conceptual**

* Utilizado en el diseño conceptual.
* Representación de la realidad no comprometida con ningún entorno informático.
* Describen las estructuras de datos y restricciones de integridad.
* Se utilizan durante la etapa de análisis de un problema dado, y están orientados a representadar los elementos que intervienen y sus relaciones.

**Modelo Lógico**

* Utilizado en el diseño lógico
* Determinan unos críterios de almacenamiento y de operaciones de manipulación de datos dentro de un tipo de entorno informático.Los SGBD comerciales se basan en un modelo lógicos concreto.

**Modelo Físico**.

* Se utilizan en el diseño físico.
* Es la implementación físisca del modelo anterior.
* Son estructuras de datos a bajo nivel, implementadas dentro de un sistema gestor de Bda comercial.

1. **TERMINOLOGÍA DEL MODELO RELACIONAL**

Modelo lógico que establece una estructura sobre los datos, independientemente del modo en que luego los almacenemos.

* 1. **Relación o Tabla – Tuplas – Dominios**

Veremos una relación como una **Tabla** con **filas** y **columnas**.

* Podemos asociar atributos a columnas .
* Podemos asociar tuplas a filas.
* **Atributos**, el nombre de cada dato que se almacena en la relación (Tabla).

El nombre del atributo debe describir el significado de la info que representa, a veces es necesario añadir una pequeña descripción.

* **Dominio**, conjunto de valores que puede tomar un atributo.

Son **atómicos**, los valores contenidos no se pueden separar en valores de dominios más simples.

* **Tuplas**, referido a cada elemento de la relación o tabla.

Cada una de las filas de la tabla, se corresponde con la idea de registro y tiene que cumplir que:

* + Cada tupla se debe corresponder con un elemento del mundo real.
  + No puede haber dos tuplas iguales.
  1. **Grado – Cardinalidad**

**Grado**, el tamaño de una tabla en base a su número de atributos (columnas).

**Cardinalidad**, el número de tuplas (filas) de una relación (tabla).

* 1. **Sinónimos**

Los términos tienen distintos sinónio según la nomeclatura utilizada:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelo relacional** | RELACION | TUPLA | ATRIBUTO | GRADO | CARDINALIDAD |
| **Tablas** | TABLA | FILA | COLUMNAS | Nº COLUMNAS | Nº FILAS |
| **Términos de registros** | FICHEROS | REGISTROS | CAMPOS | Nº CAMPOS | Nº REGISTROS |

1. **RELACIONES. CARACTERÍSTICAS DE UNA RELACIÓN (TABLA)**

* Cada tabla tiene un nombre distinto.
* Cada atributo(columna) de la tabla toma un solo valor en cada tupla(fila).
* Cada atributo(columna) tiene un nombre distinto en cada tabla.
* No puede haber dos tuplas(filas) completamente iguales.
* El orden de las tuplas(filas) no importa.
* El orden de los atributos(columnas) no importa.
* Todos los datos de un atributo(columna) deben ser del mismo dominio.
  1. **Tipos de Relaciones o Tablas**

Se pueden clasificar en:

* **Persistentes**, solo pueden ser borradas por los usuarios.
  + **Base**, independientes, se crean indicando su estructura y sus ejemplares.
  + **Vistas**, sólo almacenan una definición de consulta, resultado de la cual se obtiene datos que proceden de otras tablas base o de otras vistas e instantáneas.

Si los datos de las tablas base cambian, los de la vista que utilizan esos datos también cambiarán ya que se obtienen a partir de ellas.

* + **Instantáneas**, son vistas pero sí almacenan los datos que muestran, además de la consulta que la creó.

Solo modifican su resultado cuando el sistema se refresca cada cierto tiempo, es como una foto de la relación, que sólo es válida durante un cierto periodo de tiempo.

* **Temporales**, son eliminadas automáticamente por el sistema.

1. **TIPOS DE DATOS**

Al crear la relación o tabla decimos qué conjunto de datos deberá ser almacenado en los atributos de las finas, hay que asignar un tipo de dato a cada atributo.

Con la asignación de tipos de datos, tabmien se selecciona un domino para cada atributo

Cada campo:

* Debe poseer un **nombre**.
* Debe tener un **tipo de dato** asociado que determinrará qué valores puedo tomar y qué operaciones se puede realizar con ellos.

Tipos de datos más comunes:

* **Text. - Autonumeric.**
* **Numeric. - Memo.**
* **Date and Time. - Moneda.**
* **Y/N. - Object Linking and Embedding (OLE).**
* **Autonumeric.**

Para determinar qué tipo de dato se asocia a cada atributo se tiene en cuenta el conjunto de valores que puede tomar y las operaciones que hay que realizar con él.

1. **CLAVES**

**Superclaves**, atributo o conjunto de ellos que identifican de modo único las tuplas o filas de una relación o tabla.

Toda la fila como conjunto es una superclave.

En el modelo relacional se trabaja con los siguientes tipos de claves:

* 1. **Clave Candidata**

Aquel conjunto de atributos que identifican de manera única cada tupla o fila de la relación o tabla, es decir, las columnas cuyos valores no se repiten en ninguna otra fila de la tabla.

Debe haber por lo tanto al menos una clave candidata, puede haber varias.

**Clave compuesta**, cuando esta compuesta de más de un atributo, estos deben identificar de forma única a la fila.

Requisitos de una clave candidata:

* **Unicidad**, no puede haber dos tuplas o filas con los mismos valores para esos atributos.
* **Irreductibilidad**, si se elimina alguno de los atributos deja de ser única.
  1. **Clave Primaria**

Aquella clave candidata que se escoge para identificar sus tuplas de modo único.

Ya que una relación no tiene tuplas duplicadas, siempre hay una clave candidata y, por lo tanto, la relación siempre tiene clave primaria.

* 1. **Clave Alternativa**

Son las claves candidatas que no son escogidas como clave primaria.

* 1. **Clave Externa, Ajena o Secundaria**

Una clave ajena es un atributo o conjunto de ellos de una relación cuyos valores coinciden con los valores de la clave primaria de alguna otra relación.

**Representan relaciones entre datos**, son los datos de atributos de una tabla cuyos valores están relacionados con atributos de otra tabla.

Pueden repetirse en la tabla.

Tienen por objetivo establecer una conexión con la clave primaria que referencian, por lo tanto, **los valores de una clave ajena deben estar presentes como clave primaria en la tabla a la que hacen referencia**, o bien ser valores nulos. En caso contrario, la clave ajena representaría una referencia o conexiónincorrecta, lo que supondría que la información almacenada es inconsistente.

1. **ÍNDICES – CARACTERÍSTICAS**

Un **índice** es una estructura de datos que premite acceder a diferentes filas de una misma talba a través de un campo o campos, lo que permite un acceso más rápido a los datos.

Son utiles al realizar consultas frecuentes a un rango de filas o a una fila de una tabla.

Los cambios en los datos de las tablas son incorporados automáticamente a los índices con transparencia total.

Son independientes lógica y físicamente de los datos, pueden ser creados y eliminados en cualquier momento sin afectar a las tablas ni a otros índices.

Al crear índices, las operaciones de modificar o agregar datos se ralentizan, ya que al realizarlas es necesario actualizar tanto la tabla como el índice.

Si se elimina un índice, el acceso a datos puede ser más lento a partir de ese momento.

1. **EL VALOR NULL. OPERACIONES**

Valor especial que designa la ausencia de dato, independientemente del dominio al que pertenezca un campo.

Cuando se trabaja con claves secundarias el valor nulo indica que la tupla o fila no está relacionada con ninguna otra, este valor NULL es común a cualquier domino.

Operadores lógicos más comunes usando el valor Null:

* TRUE AND NULL, result NULL
* FALSE AND NULL, result FALSE
* TRUE OR NULL, result TRUE
* FALSE OR NULL, result NULL
* NOT NULL, result NULL

En todas las bases de datos relacionales se utiliza un operador llamado IS NULL que devuelve TRUE si el valor con el que se compara el NULL.

1. **VISTAS**

Una **vista** es una tabla virtual cuyas filas y columnas se obtienen a partir de una o varias tablas que constituyen un modelo.

Lo que almacena no es la tabla en sí, sino su definicion.

Actúa como filtro de las tablas en las que hace referencia en ella.

La consulta que define la vista puede provenir de una o varias tablas, o bien de otras vistas de la dba ctual u otras dba.

Se pueden dar 2 razones por las que crear vistas:

* **Seguridad**, para que los users tengan acceso a una parte de la info que hay en la tabla, pero no a toda ella.
* **Comodidad**, las vistas no tienen la complejidad como las sentencias de un lenguaje de dba.

Si se actualizan los datos de una vista, estamos actualizando realmente la tabla, y si actualizamos la tabla los cambios serán visibles desde la vista.

No siempre se pueden actualizar los datos de una vista, depende de su complejidad y del gestor de Bda.

1. **USUARIOS – ROLES – PRIVILEGIOS**

Un **user** es un conjunto de permisos que se aplican a una conexión de Bda.

Otras características:

* Es el propietario de ciertos objetos.
* Realiza los backups.
* Tiene asignada una cuota de almacenamiento.
* Tiene asignado un tablespace por defecto para los objets en Oracle.

Un **privilegio** es un permiso dado a un user para que realice ciertas operaciones, pueden ser de 2 tipos:

* **Op. de sistema**, necesita el permiso de sistema correspondiente.
* **Op. Sobre objeto**, necesita el permiso sobre el objeto en cuestión.

Un **rol** es una agrupación de permisos de sistema y objeto.

1. **SQL – Structured Query Language**

Es un lenguaje de domino específico utilizado en programcación, diseñado para administrar y recuperar información de sistemas de gestion de Dba relacionales, es el lenguaje fundamental de los SGBD relacionales.

Posee 2 caracteristicas muy apreciadas: **potencia** y **versatilidad**, ademas es un lenguaje muy natural.

Tiene otras **capacidades** además de las consultas de la Dba:

* La definición de la propia estructura de los datos – **DDL**.
* Su manipulación – **DML**.
* La especificación de conexiones seguras – **DCL**.

**Formas de trabajar** con SQL:

* **SQL embebido**, las sentencias se escriben dentro de un programa escrito en otro lenguaje.
* **SQL interpretado**, se utiliza un entrono gráfico para escribir y ejecutar sentencias o bien desde SQL\*Plus.
  1. **Elementos del Lenguaje – Normas de Escritura**
* **COMMANDS**, las instrucciones que se pueden crear en SQL:
  + **DDL**, **Data Definition Language**.
  + **DML**, **Data Manipulation Language**.
  + **DCL**, **Data Control Languaje**.
* **CLAUSES**, condiciones o crtiterios, palabras especiales que permiten modificar el funcionamiento de los users.
* **OPERATORS**, permiten crear expresiones complejas, son aritméticos o lógicos.
* **LITERALS**, para conseguir valores complejos.
* **FUNCTIONS**, constantes, valores concretos.

**Normas** a seguir:

* Todas las instrucciones terminan con un signo de punto y coma.
* No distingue entre mayúsculas y minúsculas.
* Cualquier comando puede ser partido con saltos de línea o espacios para facilitar su lectura y comprensión.
* Los comentarios comienzan por /\* y terminan con \*/
  1. **Data Definition Languaje – DDL**

La primera fase del trabajo con Dba comienza con sentencias DDL.

Conocer DDL es imprescindible para crear, modificar y eliminar objetos de la Dba.

Las instrucciones DDL generan acciones que no se pueden deshacer, por eso es conveniente usarlas con precaución y tener backups cuando manipulamos la Dba.

* 1. **Creación de DBA. Objetos de la DBA.**

**Esquemas** o **users**, espacio de nombres separado para cada conjunto de tablas.

Sentencia SQL para crear una DBA:

***CREATE DATABASE*** *NombreDBA;*

Una DBA es un conjunto de objetos que servirán para gestionar datos, estos objetos estan contenidos en esquemas y éstos suelen estar asociados a un user.

* 1. **Data Control Languaje -DCL**

Información previa a la creación de una tabla:

* Nombre de la tabla.
* Nombre de cada columna.
* Tipo y tamaño de datos almacenados en cada columna.
* Restricciones sobre los datos.
* Otras inforamaciones adicionales que necesitemos.

Reglas que deben cumplir los nombres de las tablas:

* No puede haber nombres de tablas duplicados en un mismo esquema.
* Deben comenzar por un cáracter alfabético.
* Longitud máxima 30 caracteres.
* No letras del alfabeto ingés, dígitos o guión bajo.
* No pueden ser palabras reservadas de SQL.
* No distingue entre mayúsculas y minúsculas.
* Si el nombre tiene espacios en blanco o caracteres nacionales, entonces se suele entrecomillar con comillas dobles.
  1. **Restricciones.**

Condiciones que una o varias columnas deben cumplir obligatoriamente.

Cada restricción creada llevará un nombre. Es conveniente ponerle de algún modo el nombre de la tabla, campos involucrados y el tipo de restricción en el nombre de la misma.

* Restricción NOT NULL

Con esta restricción se obliga a que una columna tenga un valor, prohibe valores nulos para una columna en una determinada tabla.

Se puede añadir cuando se crea o modifica un campo añadiendo NOT NULL después de poner el tipo de dato.

* Restricción UNIQUE

Obliga a que no se puedan repetir valores en la columna.

* Restricción PRIMARY KEY

En el modelo relacional las tablas deben tener una clave primaria.

Solo puede haber 1 clave primaria por tabla, pero puede estar formada por varios campo, tambien podrá ser referenciada como clave agena en otras talbas.

La clave primari hace que los campos que la forman sean NOT NULL y que sus valores sean de tipo UNIQUE.

* Restricción REFERENCES. FOREING KEY

Las claves ajenas, secundarias o foráneas son campos de una tabla que se relacionablan con la clave primaria de otra tabla.

Cuando se crea una tabla hay que indicar de alguna forma quiés es clave ajena, haciendo referencia a la tabla y campos de donde procede.

* Restricción DEFAULT y VALIDACIÓN

DEFAULT asigna un valor por defecto a un campo de una tabla.

En sus especificaciones se puede añadir distintas expresiones:

* Constantes, funciones SQL y variables.

Para incluir en un campo la fecha actual se utiliza SYSDATE como valor.

Con CHECK se comprueba que los valores introducidos son adecuados para ese campo, comprueba que se cumpla una condución determinada al rellenar una columna.

* 1. **Eliminación de Tablas**

**DROP TABLE**, elimina una tabla incluidos sus datos.

**CASCADE CONSTRAINTS**, se añade si alguna columna es clave ajena en otra tabla borrando así las restricciones donde es clave ajena.

* 1. **Modificacion de Tablas**
* **RENAME**;
* **ALTER TABLE** table\_name **ADD**;
* **ALTER TABLE** table\_name **DROP COLUMN**;
* **ALTER TABLE** table\_name **MODIFY**;
* **ALTER TABLE** table\_name **RENAME COLUM** old\_name **TO** new\_name;
* **ALTER TABLE** table\_name **DROP CONSTRAINT** constraint\_name;
* **ALTER TABLE** table\_name **RENAME CONSTRAINT** constraint\_name;
* **ALTER TABLE** table\_name **DISABLE CONSTRAINT** constraint\_name;
* **ALTER TABLE** table\_name **ENABLE CONSTRAINT** constraint\_name;
  1. **Creación y Eliminación de Índices**
* **CREATE INDEX** index\_name **ON** table\_name;
* **DROP INDEX** index\_name;

1. **DATA CONTROL LANGUAJE – DCL**

* **CREATE USER**, crea un name de user identificado por el sistema.
* **IDENTIFIED BY**, permite dar una clave al user creado.
* **DEFAULT TABLESPACE**, asigna a un usuario el Tablespace por defecto para almacenar los objetos que cree, si no se asigna ninguna será **SYSTEM**.
* **TEMPORARY TABLESPACE**, especifica el name del Tablespace para trabajos temporles, por defecto sera SYSTEM.
* **QUOTA**, asigna un espacio en Mb o Kb en el Tablespace asignado, si no se especifica el user no tendrá espacio y no podrá crear objetos.
* **PROFILE**, asigna un perfil al user, si no se especifica sera por defecto.
  1. **Permisos**

Permiten llevar a cabo operaciones a los users.

Se pueden agrupar formando **roles**, lo que simplifica la administración.

Los roles pueden activarse, desactivarse o protegerse con una clave.

Mediante los roles podemos gestionar los comandos que pueden utilizar los users.

Un permiso se puede asignar a un user o a un rol.

Un privilegio o permiso se concede con el comando **GRANT.**

* **ON**, especifica el objeto sobre el que se conceden los privilegios.
* **TO**, señala a los users o roles a los que se les conceden privilegios.
* **ALL**, concede todos los privilegios sobre el objeto especificado.
* **WITH GRANT OPTION**, permite que el receptor del privilegio se lo pueda conceder a otros.
* **PUBLIC**, hace que un privilegio esté disponible para todos los users.

Los **privilegios de sistema** dan derecho a ejecutar los comandos SQL o acciones sobre objetos de un tipo especificado.

Existen gran variedad de privilegios distintos.

**Retirar permisos o privilegios:**

**REVOKE** -> Sobre objetos -> Del sistema o roles a users

**ELEMENTOS DEL LENGUAJE SQL**

* **COMANDOS**
  + **Comandos DDL**
    - **CREATE**
    - **DROP**
    - **ALTER**
  + **Comandos DML**
    - **SELECT**
    - **INSERT**
    - **UPDATE**
    - **DELETE**
  + **Comandos DCL**
    - **GRANT**
    - **REVOKE**
* **CLAUSULAS**
  + - **FROM**
    - **WHERE**
    - **GROUP BY**
    - **HAVING**
    - **ORDER BY**
* **OPERADORES**
  + **Lógicos**
    - **AND – OR - NOT**
  + **De Comparacion**
    - **<** ; **>** ; **<=** ; **>=** ; **=**
    - **BETWEEN**
    - **LIKE**
    - **IN**
* **FUNCIONES**
  + **De Agregado**
    - **AVG**
    - **COUNT**
    - **SUM**
    - **MAX**
    - **MIN**
* **LITERALES**
  + - **31/12/99** -> Literal fecha.
    - **Maria** -> Literal caracteres.
    - **5** -> Literal número.