

Introducción de Base de Datos y al Lenguaje de Manipulación de Datos

CIMAT MTY

¿Qué es una base de datos?

Es una serie de **datos** relacionados que forman una **estructura lógica**.

Una estructura reconocible desde un programa informático. La cual no solo contiene los datos en sí, sino la forma en la que se relacionan .

Dato. Cualquier elemento informativo que tenga relevancia para un usuario.

Componentes de un sistema de bases de datos

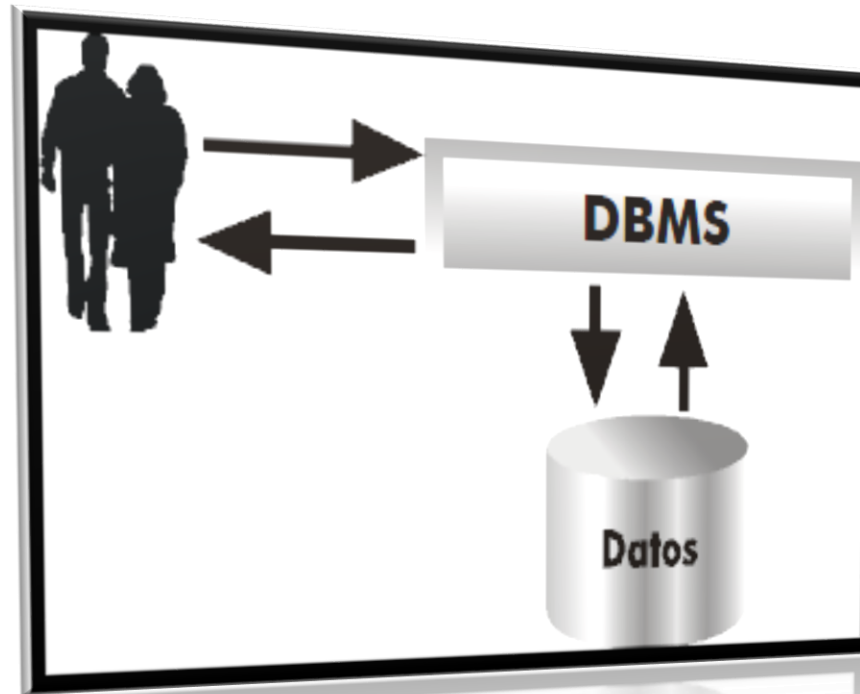
- **Datos.** Información relevante que almacena y gestiona la base de datos.
- **Hardware.** Equipamiento físico que se utiliza para gestionar los datos.
- **Software.** Aplicaciones que permiten el funcionamiento adecuado de la base de datos.
- **Recursos humanos.** Personal que maneja la base de datos.

El Objetivos del Desarrollo de Base de Datos

- Disminuir la redundancia e inconsistencia de los datos.
- Evitar dificultad en el acceso a los datos.
- Evitar el aislamiento de datos.
- Evitar los problemas de Integridad.
- Evitar el problema de atomicidad.
- Evitar anomalías en el acceso concurrente.
- Evitar los problemas de seguridad

Gestores de bases de datos DBMS (*Data Base Management System*)

Es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

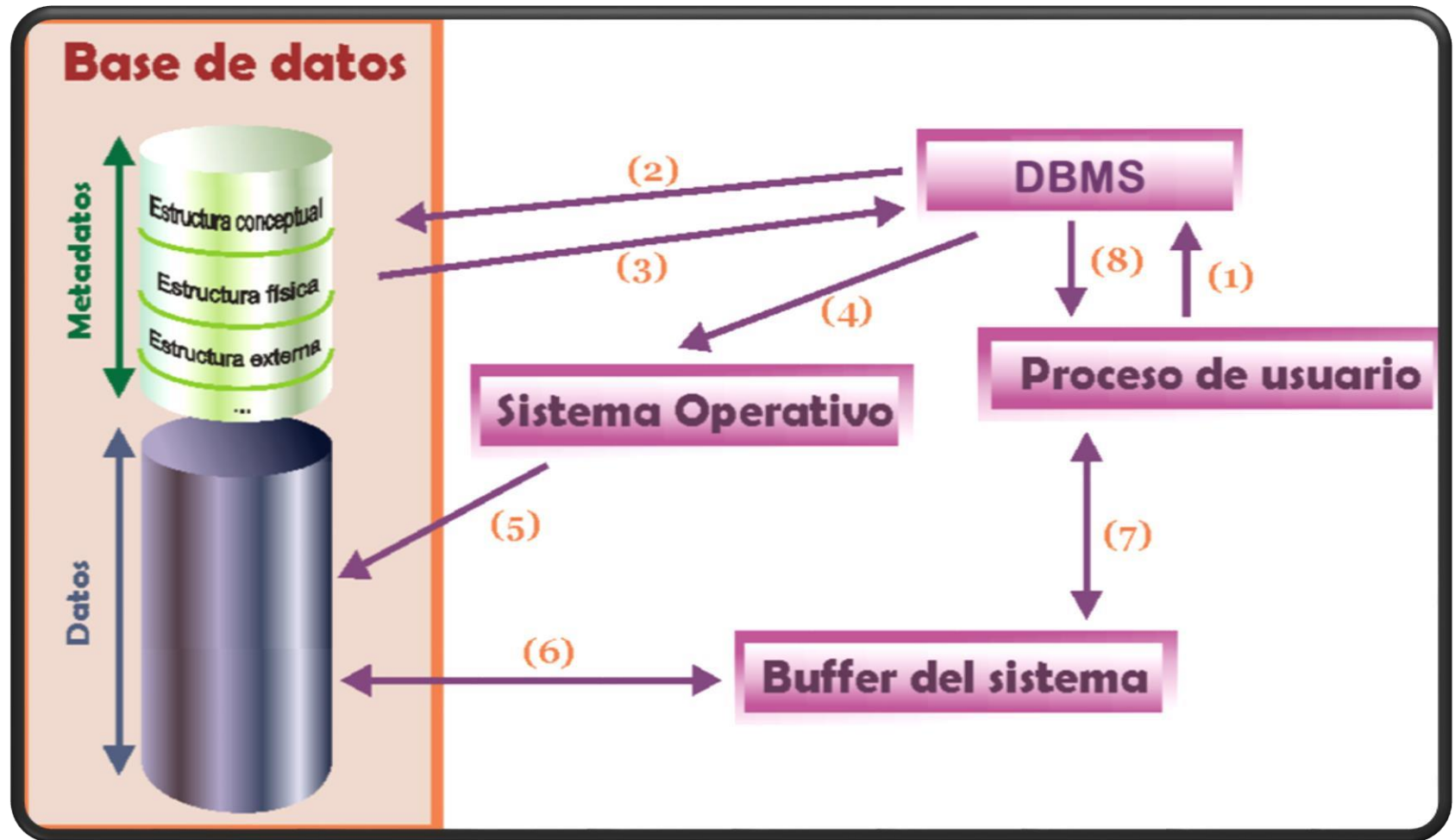


Arquitectura de los DBMS

La arquitectura de tres niveles para los Sistemas de Gestión de Base de Datos, cuyo objetivo principal era el de separar los programas de aplicación de la Base de datos física.

- **Nivel interno o físico:** Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etc.
- **Nivel externo o de visión:** Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.
- **Nivel conceptual:** Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento.

Funcionamiento del DBMS



El éxito del DBMS reside en mantener la seguridad e integridad de los datos.

Entre las herramientas DBMS:

- Creación y especificación de los datos.
- administrar y crear la estructura física.
- manipulación de los datos.
- recuperación.
- creación de copias de seguridad.
- creación de aplicaciones.
- instalación.
- exportación e importación.

Lenguajes de los DBMS

- **Función de control.** Mediante esta función los administradores poseen mecanismos para determinar las visiones de los datos permitidas a cada usuario, además de proporcionar elementos de creación y modificación de esos usuarios.

Se suelen incluir aquí las tareas de copia de seguridad, carga de ficheros, auditoria, protección ante ataques externos, configuración del sistema,...

El lenguaje que implementa esta función es el **lenguaje de control de datos o DCL.**

Lenguajes de los DBMS

- **Función de descripción o definición.** Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos.
- Esta función se realiza mediante el **lenguaje de descripción de datos o DDL.**

Mediante ese lenguaje se definen:

- Estructuras de datos
- Relaciones entre los datos
- Reglas que han de cumplir los datos

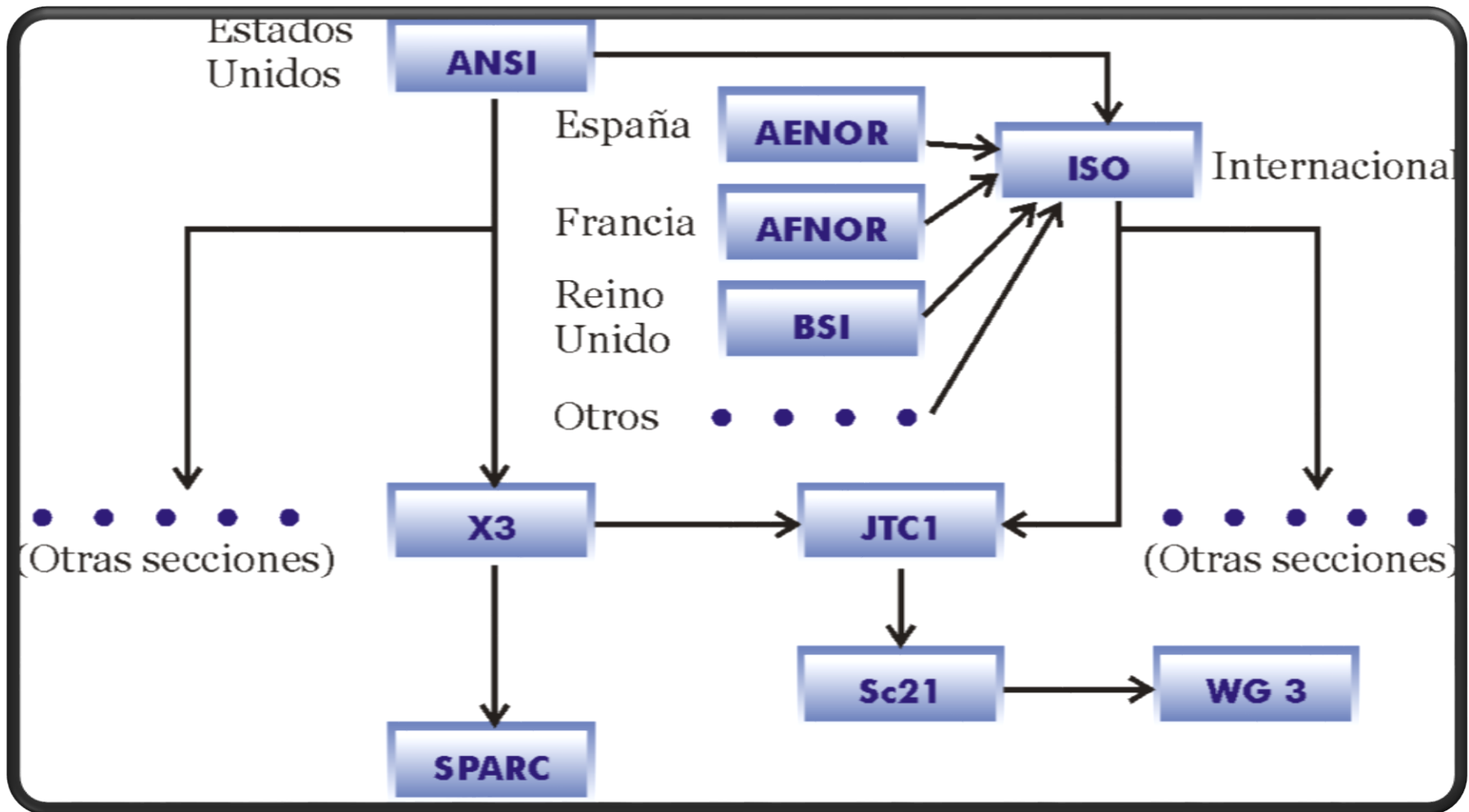
Lenguajes de los DBMS

- **Función de manipulación.** Permite modificar y utilizar los datos de la base de datos. Se realiza mediante el **lenguaje de modificación de datos** o **DML**. Mediante ese lenguaje se puede:
 - Añadir datos
 - Eliminar datos
 - Modificar datos
 - Buscar datos (**lenguaje de consulta de datos** o **DQL**)

Arquitectura de los DBMS estándares

- Organismos de estandarización
 - ISO (*International Organization for Standardization*)
 - IEC (*International Electrotechnical Commission*)
 - JTC 1 (*Joint Technical Committee*)
- Organismos de divulgación
 - ANSI(*American National Science Institute*)
 - SPARC (*System Planning and Repairments Committee*)
 - X3
 - ODASYL
 - ODMG

Arquitectura de los DBMS estándares



Modelos BD

Modelo Jerárquico

Es un modelo de datos en el cual los datos son organizados en una estructura parecida a un árbol. La estructura permite a la información que se repite y usa relaciones padre/Hijo: **cada padre puede tener muchos hijos pero cada hijo sólo tiene un padre.**

Modelo Red

El modelo de red expande la estructura jerárquica, permitiendo relaciones N:N en una estructura tipo árbol que permite múltiples padres, el modelo en red era el más popular para las bases de datos antes de relacional.

Modelo relacional

Es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud;

Diseño conceptual de bases de datos

el modelo Entidad - Relación

- Componentes del modelo
 - Entidad
 - Tipos de entidades
 - Regulares
 - Débiles
 - Relaciones
 - Tipos de relaciones
 - Binarias, Ternarias, n -arias, dobles y dobles.

Ejemplo Tabla 1

Invoice	InvoiceDATE	CustomerId	Name	LastName	InvoiceLineId	TrackId	SongName	PRICE
1	01/01/2009 00:00	2	Leonie	Köhler	1	2	Balls to the Wall	0.99
1	01/01/2009 00:00	2	Leonie	Köhler	2	4	Restless and Wild	0.99
2	02/01/2009 00:00	4	Bjørn	Hansen	3	6	Put The Finger On You	0.99
2	02/01/2009 00:00	4	Bjørn	Hansen	4	8	Inject The Venom	0.99
2	02/01/2009 00:00	4	Bjørn	Hansen	5	10	Evil Walks	0.99
2	02/01/2009 00:00	4	Bjørn	Hansen	6	12	Breaking The Rules	0.99
3	03/01/2009 00:00	8	Daan	Peeters	7	16	Dog Eat Dog	0.99
3	03/01/2009 00:00	8	Daan	Peeters	8	20	Overdose	0.99
3	03/01/2009 00:00	8	Daan	Peeters	9	24	Love In An Elevator	0.99
3	03/01/2009 00:00	8	Daan	Peeters	10	28	Janie's Got A Gun	0.99
3	03/01/2009 00:00	8	Daan	Peeters	11	32	Deuces Are Wild	0.99
3	03/01/2009 00:00	8	Daan	Peeters	12	36	Angel	0.99
4	06/01/2009 00:00	14	Mark	Philips	13	42	Right Through You	0.99
4	06/01/2009 00:00	14	Mark	Philips	14	48	Not The Doctor	0.99
4	06/01/2009 00:00	14	Mark	Philips	15	54	Bleed The Freak	0.99
4	06/01/2009 00:00	14	Mark	Philips	16	60	Confusion	0.99
4	06/01/2009 00:00	14	Mark	Philips	17	66	Por Causa De Você	0.99
4	06/01/2009 00:00	14	Mark	Philips	18	72	Angela	0.99
4	06/01/2009 00:00	14	Mark	Philips	19	78	Master Of Puppets	0.99
4	06/01/2009 00:00	14	Mark	Philips	20	84	Welcome Home (Sanitarium)	0.99
4	06/01/2009 00:00	14	Mark	Philips	21	90	Set It Off	0.99
5	11/01/2009 00:00	23	John	Gordon	22	99	Your Time Has Come	0.99

¿Qué problemas hay, si se hacen cambios?

- Supongamos que se quiere cambiar el precio de una canción...
- Repetición de elementos
 - Modificación de precios
 - Modificación de Nombres de Canciones

Normalización

- Primera forma normal
 - Dividir la tabla en tablas más pequeñas con elementos que no se repitan, las cuales esán asociadas.
 - Esta normalización aún no es suficiente para que sea una forma aprovechable.

Invoice	InvoiceDATE	CustomerId	Name	LastName	Invoice	InvoiceLineId	Track	SongName	PRICE
1	01/01/2009 00:00	2	Leonie	Köhler	1	1	2	Balls to the Wall	0.99
2	02/01/2009 00:00	4	Bjørn	Hansen	1	2	4	Restless and Wild	0.99
3	03/01/2009 00:00	8	Daan	Peeters				Put The Finger On	
4	06/01/2009 00:00	14	Mark	Philips	2	3	6	You	0.99
5	11/01/2009 00:00	23	John	Gordon	2	4	8	Inject The Venom	0.99
					2	5	10	Evil Walks	0.99
					2	6	12	Breaking The Rules	0.99
					3	7	16	Dog Eat Dog	0.99
					3	8	20	Overdose	0.99
					3	9	24	Love In An Elevator	0.99
					3	10	28	Janie's Got A Gun	0.99
					3	11	32	Deuces Are Wild	0.99
					3	12	36	Angel	0.99
					4	13	42	Right Through You	0.99
					4	14	48	Not The Doctor	0.99
					4	15	54	Bleed The Freak	0.99

Normalización

- Segunda forma normal
 - ¿Podemos añadir una nueva canción a la tabla dividida anterior?
 - Cuando una tabla está definida de tal manera que un elemento se encuentra definido por una clave única, hablamos de la segunda forma normal.

TrackId	SongName	PRICE
2	Balls to the Wall	0.99
4	Restless and Wild	0.99
6	Put The Finger On You	0.99
8	Inject The Venom	0.99
10	Evil Walks	0.99
12	Breaking The Rules	0.99
16	Dog Eat Dog	0.99
20	Overdose	0.99
24	Love In An Elevator	0.99
28	Janie's Got A Gun	0.99
32	Deuces Are Wild	0.99
36	Angel	0.99
	Through You	0.99
	e Doctor	0.99

Normalización

- Segunda forma normal
 - ¿Podemos añadir una nueva canción a la tabla dividida anterior?
 - Cuando una tabla está definida de tal manera que un elemento se encuentra definido por una clave única, hablamos de la segunda forma normal.

TrackId	SongName	PRICE
2	Balls to the Wall	0.99
4	Restless and Wild	0.99
6	Put The Finger On You	0.99
8	Inject The Venom	0.99
10	Evil Walks	0.99
12	Breaking The Rules	0.99
16	Dog Eat Dog	0.99
20	Overdose	0.99
24	Love In An Elevator	0.99
28	Janie's Got A Gun	0.99
32	Deuces Are Wild	0.99
36	Angel	0.99
	Through You	0.99
	e Doctor	0.99

Normalización

- Tercera forma normal
 - ¿Cómo agregamos una nueva venta a la tabla anterior de ventas y clientes?

Invoice	InvoiceDATE	CustomerId	Name	LastName
1	01/01/2009 00:00	2	Leonie	Köhler
2	02/01/2009 00:00	4	Bjørn	Hansen
3	03/01/2009 00:00	8	Daan	Peeters
4	06/01/2009 00:00	14	Mark	Philips
5	11/01/2009 00:00	23	John	Gordon

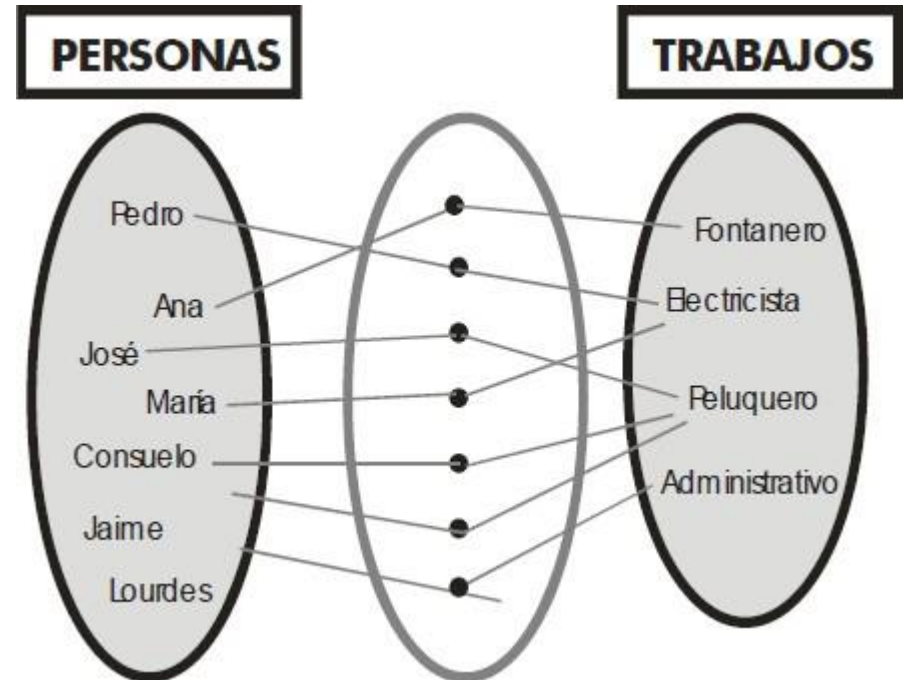
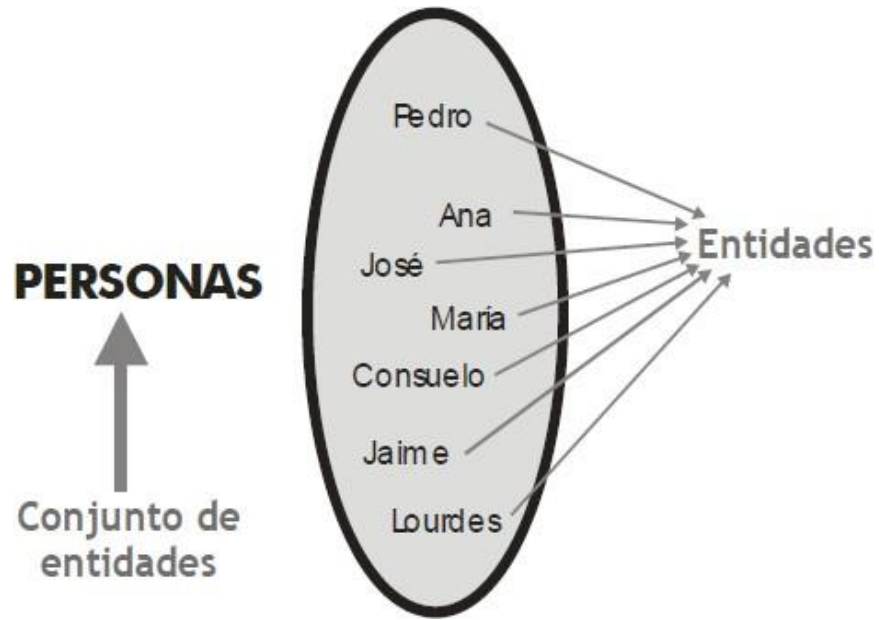
Normalización

- Tercera forma normal
 - Cada columna en una tabla en la tercera forma normal debe de estar formado por una clave y/o determinado de manera única.

Invoice	InvoiceDATE	CustomerId
1	01/01/2009 00:00	2
2	02/01/2009 00:00	4
3	03/01/2009 00:00	8
4	06/01/2009 00:00	14
5	11/01/2009 00:00	23

CustomerId	Name	LastName
2	Leonie	Köhler
4	Bjørn	Hansen
8	Daan	Peeters
14	Mark	Philips
23	John	Gordon

Entidad



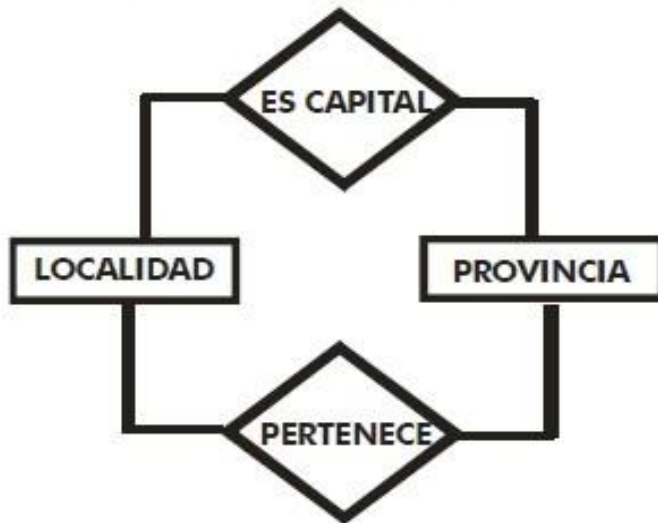
Relaciones



Relación binaria



Relación ternaria



Relación doble

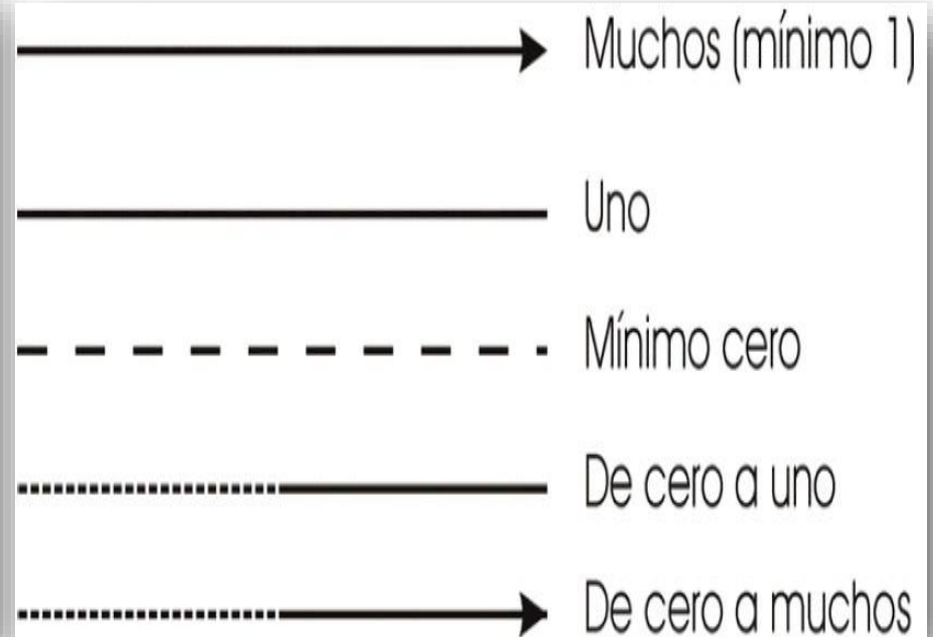
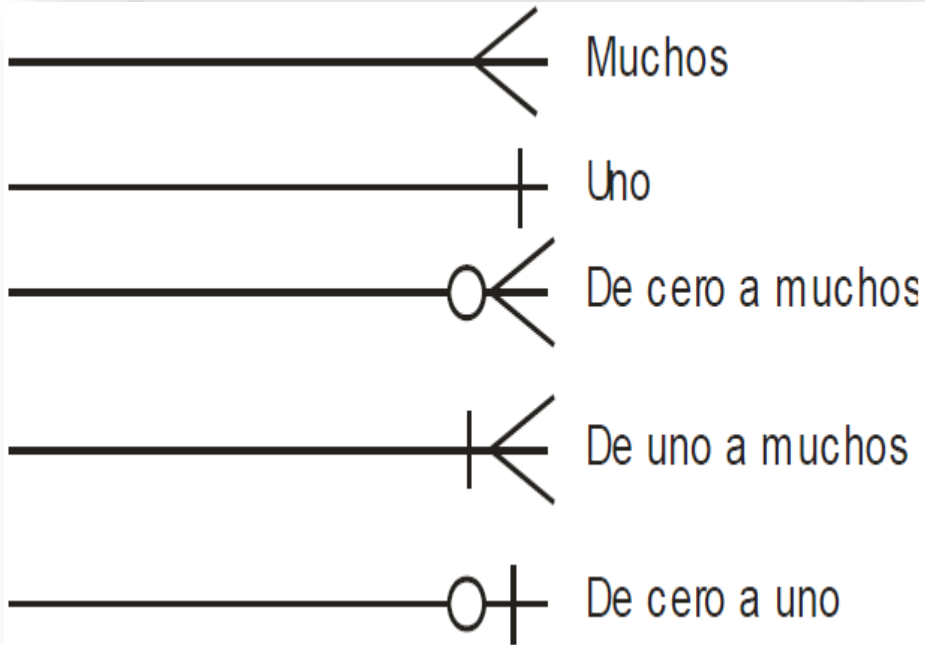


Relación reflexiva

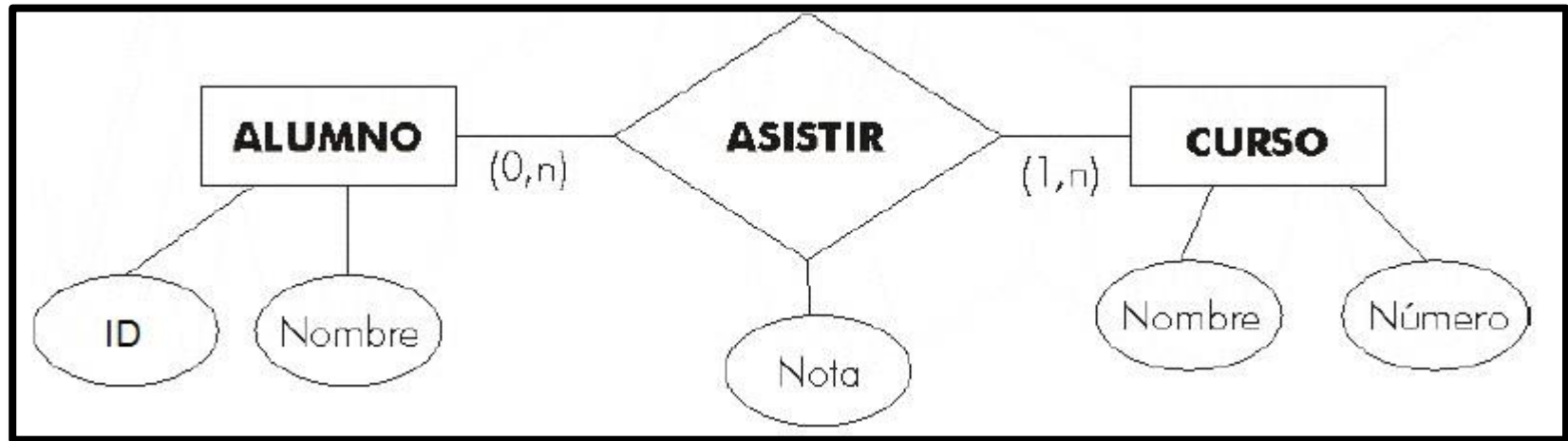
Diseño conceptual de bases de datos el modelo entidad - relación

- Cardinalidad
 - Cardinalidad mínima
 - Cardinalidad máxima
- roles
- Atributos
 - Compuesto
 - Múltiples
 - Opcionales

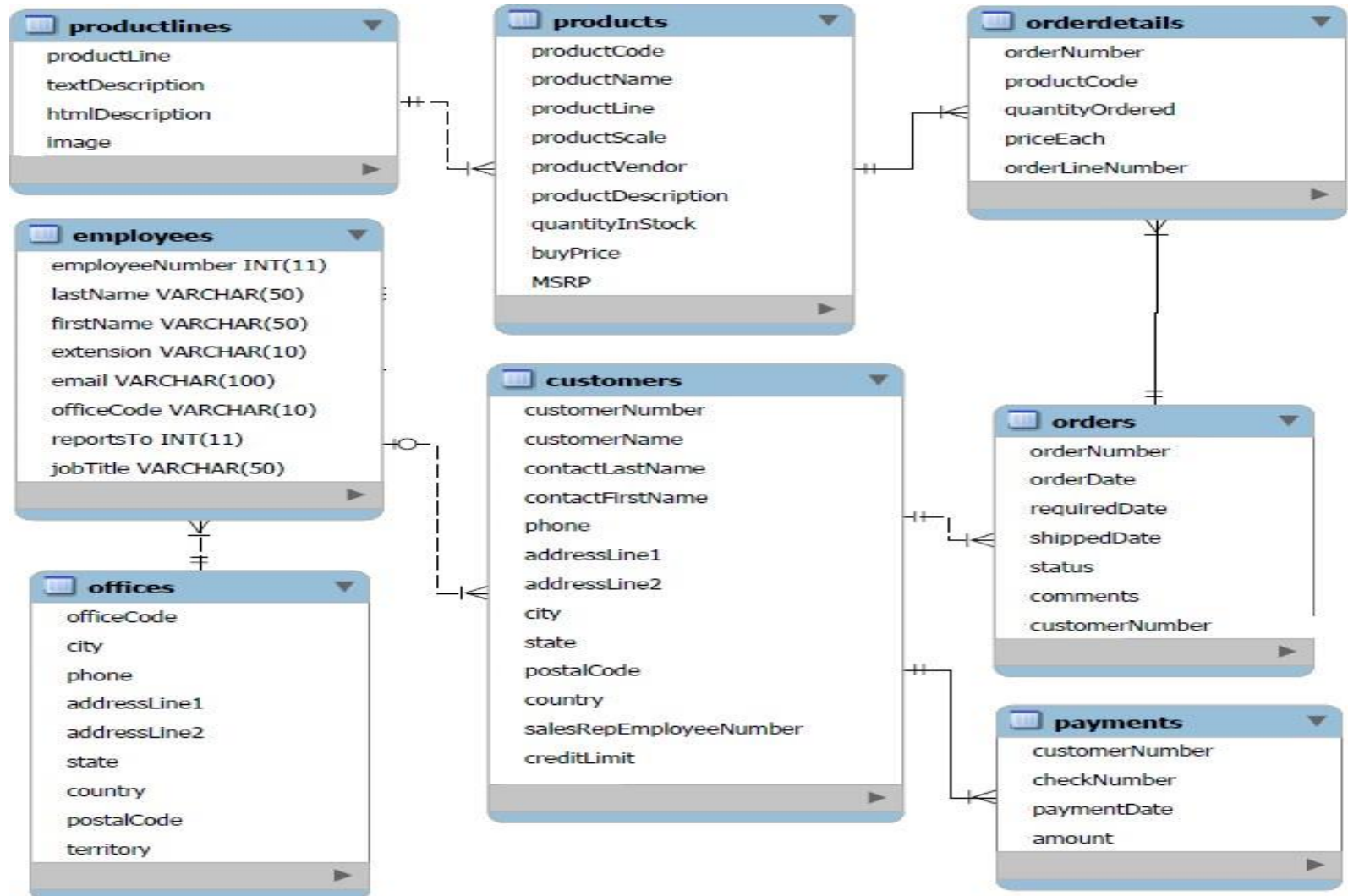
Cardinalidad



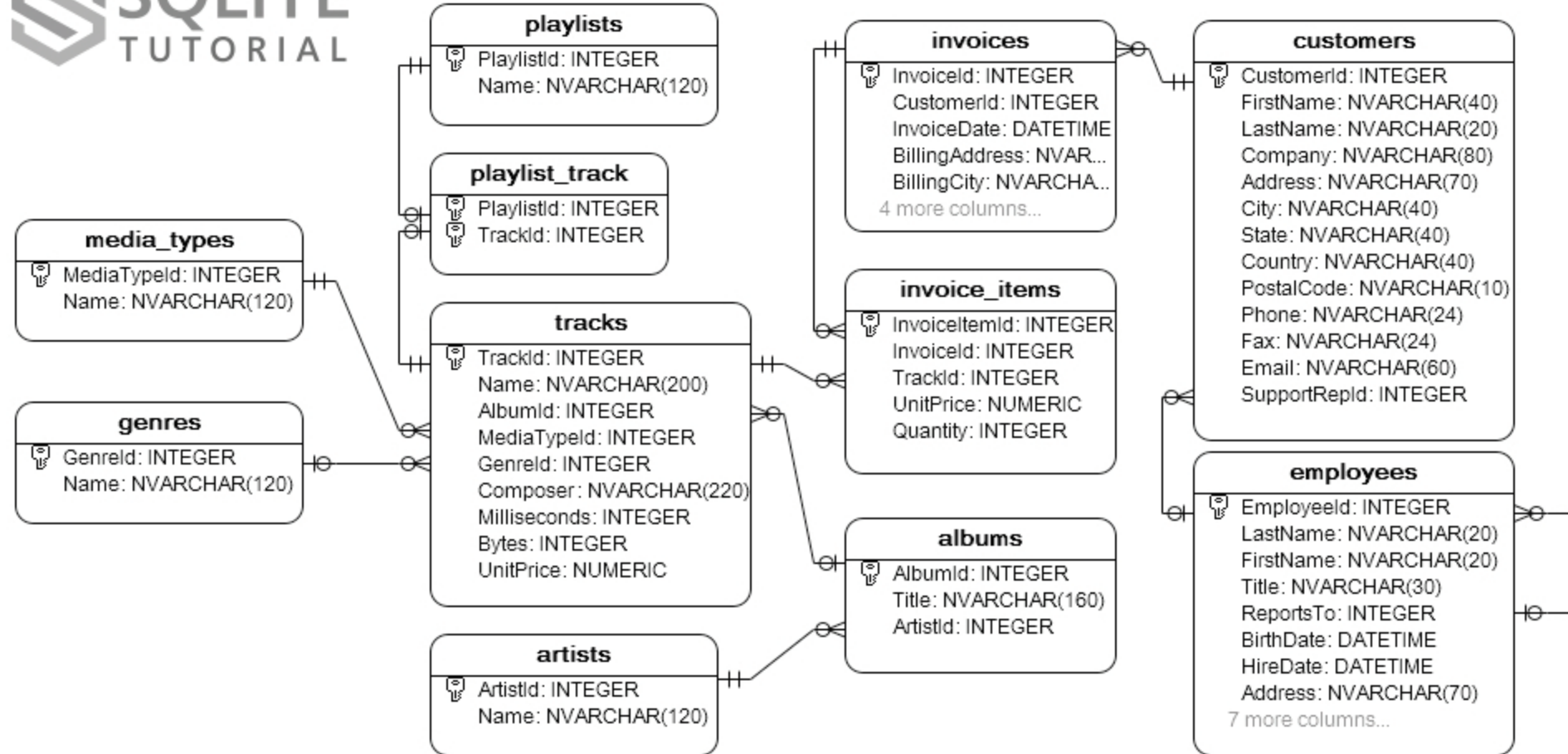
Atributos



Ejemplo1



Ejemplo2



LENGUAJE SQL

- SQL
 - Lenguaje declarativo
 - Lenguaje de cuarta generación
- Componentes de un entorno de ejecución SQL
 - **agente SQL**
 - **implementación SQL**
 - **cliente SQL**
 - **servidor SQL**

- **Agentes SQL.**

- Ejecución directa. SQL interactivo
- Ejecución incrustada o embebida
- Ejecución a través de clientes gráficos
- Ejecución dinámica

Proceso de las instrucciones SQL

El proceso de una instrucción SQL es el siguiente:

- (1) Se analiza la instrucción. Para comprobar la sintaxis de la misma.
- (2) Si es correcta se valora si los metadatos de la misma son correctos. Se comprueba esto en el diccionario de datos.
- (3) Si es correcta, se optimiza, a fin de consumir los mínimos recursos posibles.
- (4) Se ejecuta la sentencia y se muestra el resultado al emisor de la misma.

Elementos del lenguaje SQL

- código SQL
 - **Comandos**
 - DQL (*Data Query Language*)
 - DML (*Data Manipulation Language*)
 - DDL, (*Data Definition Language*)
 - DCL, (*Data Control Language*)
 - Instrucciones de control de transacciones (DTL).
 - **Cláusulas**
 - **Operadores**
 - **Funciones**
 - **Literales**
 - **Metadatos**

- **Acceso Base Datos**

Server

Cliente <https://dev.mysql.com/downloads/mysql/>

Driver C, php, R, .NET, python...

`mysql -h[host] -u [usuario] -p`

- **Descriptivos BD**

`SHOW databases;`

`USE mysql;`

`DESCRIBE mysql.user;`

- **Usuarios**

- Creación y eliminación

`CREATE USER user IDENTIFIED BY 'pw';`

`DROP USER user@%;`

- Privilegios

`GRANT ALL PRIVILEGES ON * . * TO 'user'@'%';`

`REVOKE ALL ON * . * FROM 'user'@'%';`

`FLUSH PRIVILEGES;`

`SHOW GRANTS FOR user;`

- **Copias seguridad BD**

- Respaldar

`mysqldump.exe -h host -u user -p BB > archivo.sql`

- restaurar

`mysql.exe -h host -u user -p BB < archivo.sql`

- **Creación**

- Creación de bases de datos

`CREATE DATABASE prueba;`

- creación de tablas

`CREATE TABLE [esquema.]nombreDeTabla(nombreDeLaColumna1 tipoDeDatos [, ...]);`

- Tipos Datos

- **Modificación**

- Modificación de tablas y Datos

`ALTER TABLE nombreViejo RENAME TO nombreNuevo;`

`ALTER TABLE nombreDeTabla ADD (nombreDeLaColumna DATE);`

`ALTER TABLE nombreDeTabla MODIFY COLUMN `nombreDeLaColumna` VARCHAR(3);`

`ALTER TABLE nombreDeTabla CHANGE `nombreDeLaColumna` nombreDeLaColumnaNuevo DATE;`

- **Eliminación**

`DROP DATABASE esquema;`

`DROP TABLE nombreDeTabla;`

`ALTER TABLE nombreDeTabla REMOVE(nombreDeLaColumna DATE);`

- **Restricciones**

`CREATE TABLE nombreDeTabla(nombreDeLaColumna VARCHAR2(9) NOT NULL);`

- **Datos**

Valores únicos

Clave primaria

Clave secundaria o foránea

Descripción	Tipos Estándar SQL	Descripción	Tipos Estándar SQL
Texto		Fechas	
Texto de anchura fija	CHARACTER(<i>n</i>) CHAR(<i>n</i>)	Fechas	DATE
Texto de anchura variable	CHARACTER VARYING(<i>n</i>) VARCHAR (<i>n</i>)	Fecha y hora	TIMESTAMP
Texto de anchura fija para caracteres nacionales	NATIONAL CHARACTER(<i>n</i>) NATIONAL CHAR(<i>n</i>) NCHAR(<i>n</i>)	Intervalos	INTERVAL
Texto de anchura variable para caracteres nacionales	NATIONAL CHARACTER VARYING(<i>n</i>) NATIONAL CHAR VARYING(<i>n</i>) NCHAR VARYING(<i>n</i>)	Booleanos y binarios	
		Lógicos	BOOLEAN BOOL
		Binarios	BIT BIT VARYING(<i>n</i>) VARBIT(<i>n</i>)
Números			
Enteros pequeños (2 bytes)	SMALLINT		
Enteros normales (4 bytes)	INTEGER INT		
Enteros largos (8 bytes)	BIGINT (en realidad no es estándar, pero es muy utilizado en muchas bases de datos)		
Enteros precisión decimal			
Decimal de coma variable	FLOAT DOUBLE DOUBLE PRECISION REAL		
Decimal de coma fija	NUMERIC(<i>m,d</i>) DECIMAL(<i>m,d</i>)		

- **Inserción de datos**

- **INSERT INTO** tabla [(listaDeCampos)]**VALUES** (valor1 [,valor2 ...])

- **Actualización de registros**

- **UPDATE** tabla **SET** columna1=valor1 [,columna2=valor2...] [**WHERE** condición]

- **Borrado de registros**

- **DELETE** [**FROM**] tabla[**WHERE** condición]

- **Obtener datos de ciertas columnas de una tabla (proyección)**

`SELECT Campo1, Campo2,... FROM nombretabla;`

- **Obtener registros (filas) de una tabla de acuerdo con ciertos criterios (selección)**

`SELECT Campo1, Campo2,... FROM nombretabla WHERE condición;`

- **Mezclar datos de tablas diferentes (asociación)**

`SELECT Campo1, Campo2,... FROM tabla1 JOINS Tabla2 ON Tabla1.Campo = Tabla2.Campo;`

- **Agrupar datos**

`SELECT Campo1, Campo2,... FROM nombretabla Group by Campo;`

Sintaxis comando SQL

- **Cálculos**

- Aritméticos
- Concatenación de textos

- **Condiciones**

- Operadores de comparación
- Valores lógicos

- **BETWEEN, IN, LIKE, IS NULL**

Precedencia de operadores

Orden de precedencia	Operador
1	*(Multiplicar) / (dividir)
2	+ (Suma) - (Resta)
3	(Concatenación)
4	Comparaciones (>, <, !=, ...)
5	IS [NOT] NULL, [NOT]LIKE, IN
6	NOT
7	AND
8	OR

Operadores de comparación y Valores lógicos

Operador	Significado
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
=	Igual
<>	Distinto
!=	Distinto
AND	Devuelve verdadero si las expresiones a su izquierda y derecha son ambas verdaderas
OR	Devuelve verdadero si cualquiera de las dos expresiones a izquierda y derecha del OR, son verdaderas
NOT	Invierte la lógica de la expresión que está a su derecha. Si era verdadera, mediante NOT pasa a ser falso.

Sintaxis comando SQL

- **Ordenación**
- **Funciones**
 - **Funciones numéricas**
 - Redondeos
 - Matemáticas
 - **Funciones de caracteres**
 - Conversión del texto a mayúsculas y minúsculas
 - Funciones de transformación
 - Funciones de caracteres
 - **Funciones de fecha**

Redondeos

Función	Descripción
ROUND (<i>n,decimales</i>)	Redondea el número al siguiente número con el número de decimales indicado más cercano. ROUND (8.239,2) devuelve 8.3
TRUNC (<i>n,decimales</i>)	Los decimales del número se cortan para que sólo aparezca el número de decimales indicado

Matemáticas

Función	Descripción
MOD (<i>n1,n2</i>)	Devuelve el resto resultado de dividir <i>n1</i> entre <i>n2</i>
POWER (<i>valor,exponente</i>)	Eleva el valor al exponente indicado
SQRT (<i>n</i>)	Calcula la raíz cuadrada de <i>n</i>
SIGN (<i>n</i>)	Devuelve 1 si <i>n</i> es positivo, cero si vale cero y -1 si es negativo
ABS (<i>n</i>)	Calcula el valor absoluto de <i>n</i>
EXP (<i>n</i>)	Calcula <i>eⁿ</i> , es decir el exponente en base <i>e</i> del número <i>n</i>
LN (<i>n</i>)	Logaritmo neperiano de <i>n</i>
LOG (<i>n</i>)	Logaritmo en base 10 de <i>n</i>
SIN (<i>n</i>)	Calcula el seno de <i>n</i> (<i>n</i> tiene que estar en radianes)
COS (<i>n</i>)	Calcula el coseno de <i>n</i> (<i>n</i> tiene que estar en radianes)
TAN (<i>n</i>)	Calcula la tangente de <i>n</i> (<i>n</i> tiene que estar en radianes)
ACOS (<i>n</i>)	Devuelve en radianes el arco coseno de <i>n</i>
ASIN (<i>n</i>)	Devuelve en radianes el arco seno de <i>n</i>
ATAN (<i>n</i>)	Devuelve en radianes el arco tangente de <i>n</i>
SINH (<i>n</i>)	Devuelve el seno hiperbólico de <i>n</i>
COSH (<i>n</i>)	Devuelve el coseno hiperbólico de <i>n</i>
TANH (<i>n</i>)	Devuelve la tangente hiperbólica de <i>n</i>

Conversión del texto a mayúsculas y minúsculas

Funciones de transformación

Función	Descripción
<code>LOWER(texto)</code>	Convierte el texto a minúsculas (funciona con los caracteres españoles)
<code>UPPER(texto)</code>	Convierte el texto a mayúsculas
<code>INITCAP(texto)</code>	Coloca la primera letra de cada palabra en mayúsculas

Función	Descripción
<code>RTRIM(texto)</code>	Elimina los espacios a la derecha del texto
<code>LTRIM(texto)</code>	Elimina los espacios a la izquierda que posea el texto
<code>TRIM(texto)</code>	Elimina los espacios en blanco a la izquierda y la derecha del texto y los espacios dobles del interior.
<code>TRIM(caracteres FROM texto)</code>	Elimina del texto los caracteres indicados. Por ejemplo <code>TRIM('h' FROM nombre)</code> elimina las haches de la columna <i>nombre</i> que estén a la izquierda y a la derecha
<code>SUBSTR(texto,n[,m])</code>	Obtiene los <i>m</i> siguientes caracteres del texto a partir de la posición <i>n</i> (si <i>m</i> no se indica se cogen desde <i>n</i> hasta el final).
<code>LENGTH(texto)</code>	Obtiene el tamaño del texto
<code>INSTR(texto, textoBuscado [,posInicial [, nAparición]])</code>	<p>Obtiene la posición en la que se encuentra el texto buscado en el texto inicial. Se puede empezar a buscar a partir de una posición inicial concreta e incluso indicar el número de aparición del texto buscado.</p> <p>Ejemplo, si buscamos la letra <i>a</i> y ponemos 2 en <i>nAparición</i>, devuelve la posición de la segunda letra <i>a</i> del texto).</p> <p>Si no lo encuentra devuelve 0</p>

Función	Descripción
REPLACE (<i>texto</i> , <i>textoABuscar</i> , [<i>textoReemplazo</i>])	<p>Buscar el texto a buscar en un determinado texto y lo cambia por el indicado como texto de reemplazo.</p> <p>Si no se indica texto de reemplazo, entonces está función elimina el texto a buscar</p>
TRANSLATE (<i>texto</i> , <i>caracteresACambiar</i> , <i>caracteresSustitutivos</i>)	<p>Potentísima función que permite transformar caracteres. Los <i>caracteresACambiar</i> son los caracteres que se van a cambiar, los <i>caracteresSustitutivos</i> son los caracteres que reemplazan a los anteriores. De tal modo que el primer carácter a cambiar se cambia por el primer carácter sustitutivo, el segundo por el segundo y así sucesivamente. Ejemplo:</p> <pre>SELECT TRANSLATE('prueba','ue','wx') FROM DUAL;</pre> <p>El resultado sería el texto <i>prwxba</i>, de tal forma que la <i>u</i> se cambia por la <i>w</i> y la <i>e</i> por la <i>x</i>.</p> <p>Si la segunda cadena es más corta, los caracteres de la primera que no encuentran sustituto, se eliminan. Ejemplo:</p> <pre>SELECT TRANSLATE('prueba','ue','w') FROM DUAL;</pre> <p>Da como resultado <i>prwba</i></p>

Función	Descripción
LPAD (<i>texto</i> , <i>anchuraMáxima</i> , [<i>caracterDeRelleno</i>]) RPAD (<i>texto</i> , <i>anchuraMáxima</i> , [<i>caracterDeRelleno</i>])	<p>Rellena el texto a la izquierda (LPAD) o a la derecha (RPAD) con el carácter indicado para ocupar la anchura indicada.</p> <p>Si el texto es más grande que la anchura indicada, el texto se recorta.</p> <p>Si no se indica carácter de relleno se rellenará el espacio marcado con espacios en blanco.</p> <p>Ejemplo:</p> <pre>LPAD('Hola',10,'-')</pre> <p>da como resultado</p> <p>-----<i>Hola</i></p>
REVERSE (<i>texto</i>)	<p>Invierte el texto (le da la vuelta)</p>

Funciones de caracteres

Función	Descripción
<code>ASCII(carácter)</code>	Devuelve el código ASCII del carácter indicado
<code>CHR(número)</code>	Devuelve el carácter correspondiente al código ASCII indicado
<code>SOUNDEX(texto)</code>	<p>Devuelve el valor fonético del texto. Es una función muy interesante para buscar textos de los que se no se sabe con exactitud su escritura. Por ejemplo:</p> <pre>SELECT * FROM personas WHERE SOUNDEX(apellido1)=SOUNDEX('Smith')</pre> <p>En el ejemplo se busca a las personas cuyo primer apellido suena como <i>Smith</i></p>

Obtener la
fecha y hora
actual

Calcular
fechas

Función	Descripción
<code>SYSDATE</code>	Obtiene la fecha y hora actuales
<code>SYSTIMESTAMP</code>	Obtiene la fecha y hora actuales en formato <code>TIMESTAMP</code>

Función	Descripción
<code>ADDMONTHS(fecha,n)</code>	Añade a la fecha el número de meses indicado por <i>n</i>
<code>MONTHS_BETWEEN(fecha1, fecha2)</code>	Obtiene la diferencia en meses entre las dos fechas (puede ser decimal)
<code>NEXT_DAY(fecha,día)</code>	Indica cual es el día que corresponde a añadir a la fecha el día indicado. El día puede ser el texto ' <i>Lunes</i> ', ' <i>Martes</i> ', ' <i>Miércoles</i> ',... (si la configuración está en español) o el número de día de la semana (1=lunes, 2=martes,...)
<code>LAST_DAY(fecha)</code>	Obtiene el último día del mes al que pertenece la fecha. Devuelve un valor <code>DATE</code>
<code>EXTRACT(valor FROM fecha)</code>	Extrae un valor de una fecha concreta. El valor puede ser <i>day</i> (día), <i>month</i> (mes), <i>year</i> (año), etc.
<code>GREATEST(fecha1, fecha2,...)</code>	Devuelve la fecha más moderna la lista
<code>LEAST(fecha1, fecha2,...)</code>	Devuelve la fecha más antigua la lista
<code>ROUND(fecha [, 'formato'])</code>	Redondea la fecha al valor de aplicar el formato a la fecha. El formato puede ser: ' <i>YEAR</i> ' Hace que la fecha refleje el año completo ' <i>MONTH</i> ' Hace que la fecha refleje el mes completo más cercano a la fecha ' <i>HH24</i> ' Redondea la hora a las 00:00 más cercanas ' <i>DAY</i> ' Redondea al día más cercano
<code>TRUNC(fecha [formato])</code>	Igual que el anterior pero trunca la fecha en lugar de redondearla.

Sintaxis comando SQL

- **Agrupaciones**

- Funciones de cálculo con grupos
- Condiciones HAVING

- **Obtener datos de múltiples tablas**

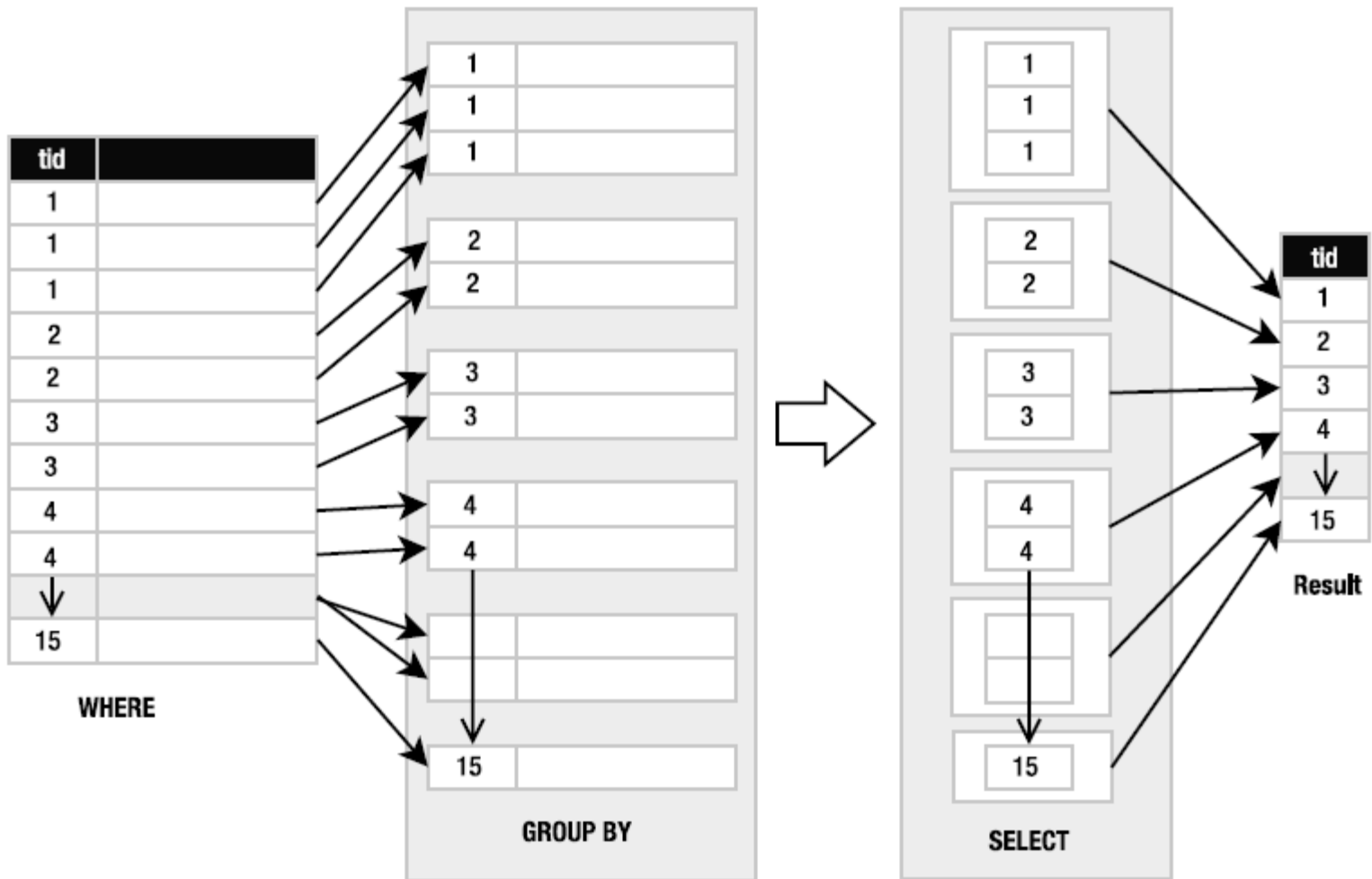
- Producto cruzado o cartesiano de tablas
- Asociando tablas

- **Subconsultas**

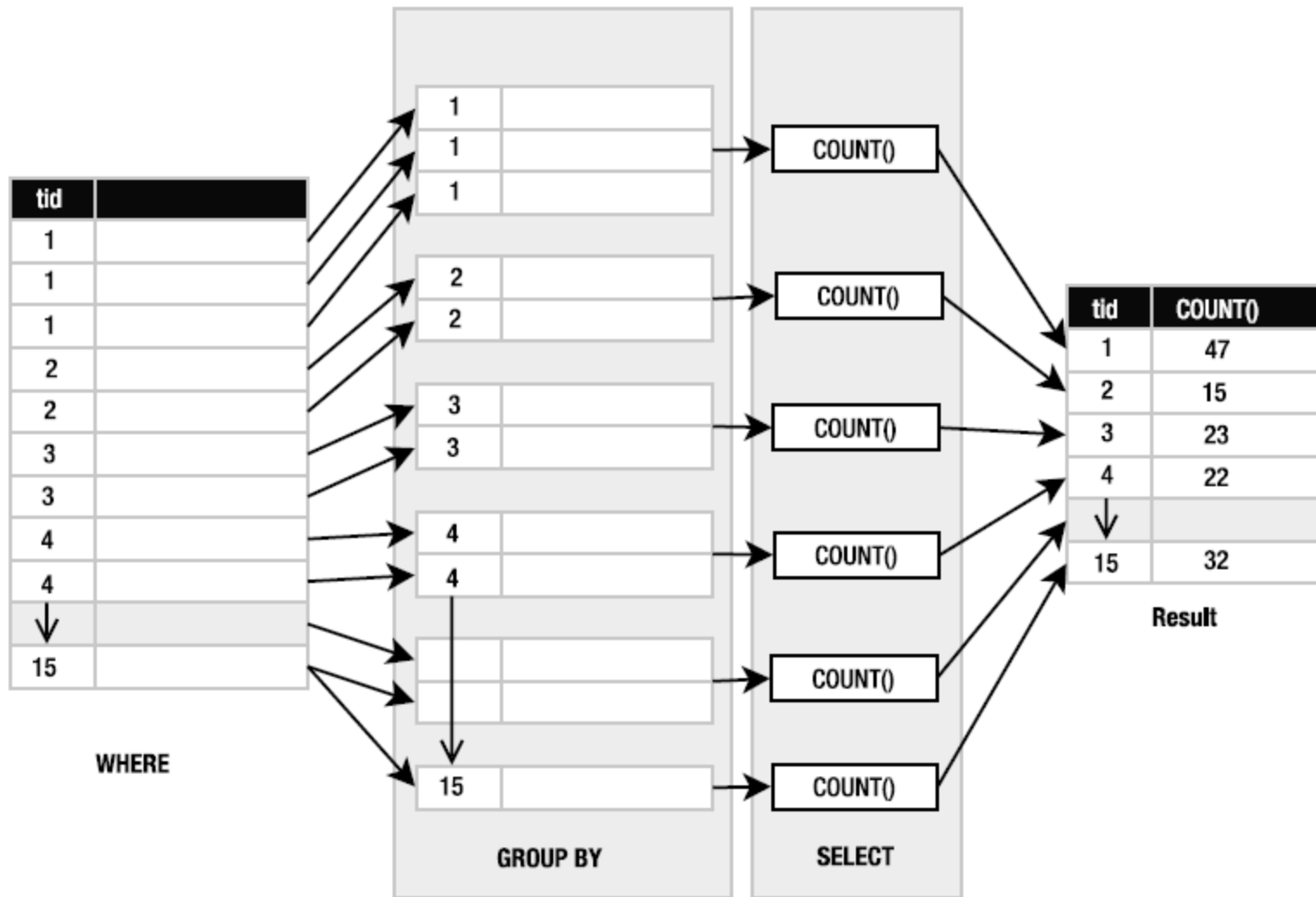
Funciones de cálculo con grupos

Función	Significado
COUNT(*)	Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna
SUM(<i>expresión</i>)	Suma los valores de la expresión
AVG(<i>expresión</i>)	Calcula la media aritmética sobre la expresión indicada
MIN(<i>expresión</i>)	Mínimo valor que toma la expresión indicada
MAX(<i>expresión</i>)	Máximo valor que toma la expresión indicada
STDDEV(<i>expresión</i>)	Calcula la desviación estándar
VARIANCE(<i>expresión</i>)	Calcula la varianza

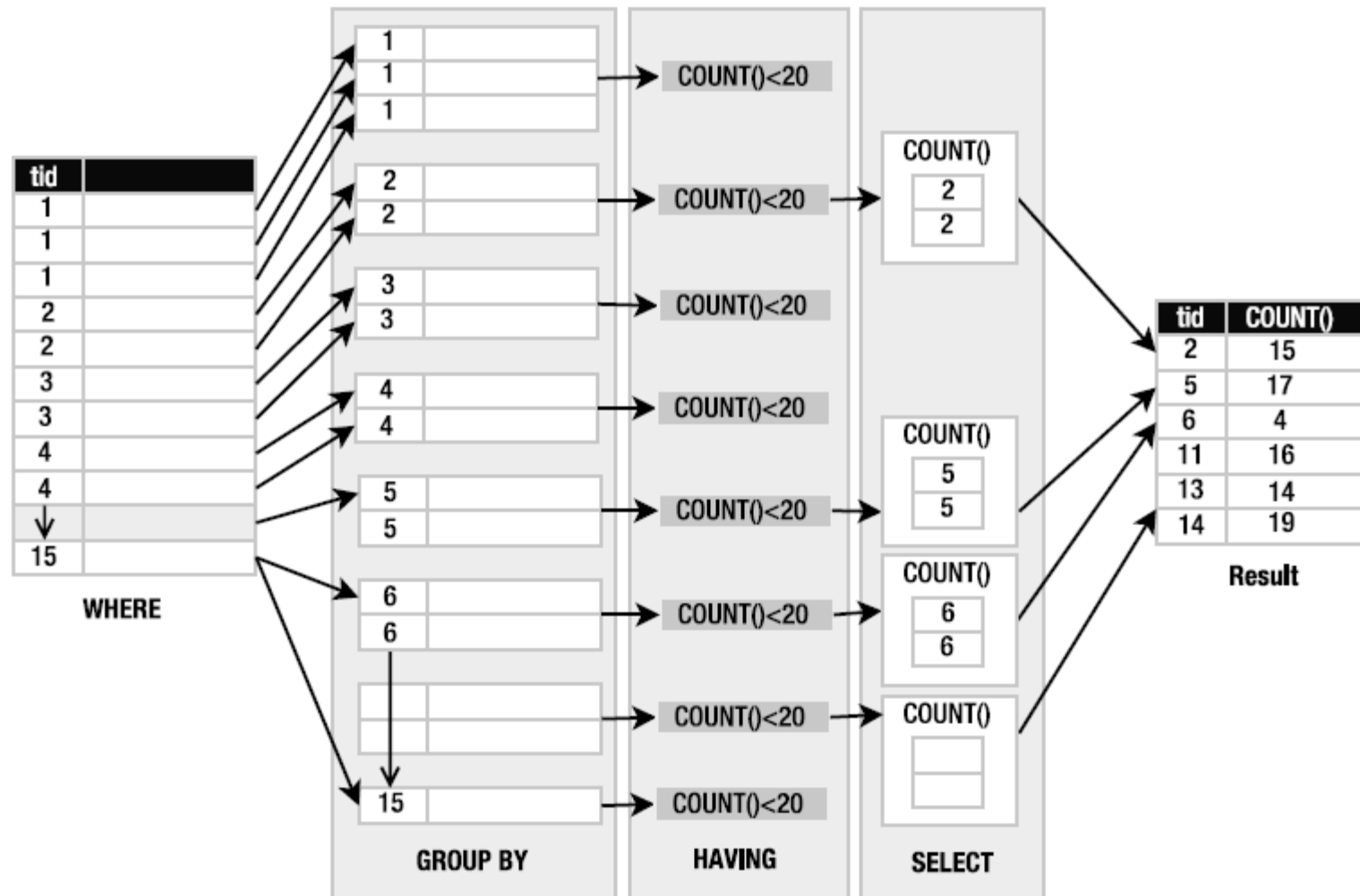
Funcionamiento- GROUP BY



Funcionamiento- GROUP BY



Funcionamiento- GROUP BY



Null-Lógica de Tres Estados

x	y	x AND y	x OR y
True	True	True	True
True	False	False	True
True	NULL	NULL	True
False	False	False	False
False	NULL	False	NULL
NULL	NULL	NULL	NULL