



# ACCESO A BASES DE DATOS

José Carlos Ciria Cosculluela

Eladio Domínguez Murillo

Inés Escario Jover

Jorge Lloret Gazo

Imprime: Servicio de Publicaciones  
Universidad de Zaragoza

Depósito Legal: Z – 2787 – 2006

# Acceso a Bases de Datos

J.C. Ciria Cosculluela, E. Domínguez Murillo, I. Escario Jover, J. Lloret  
Gazo

## ÍNDICE

### Nociones Básicas

1. El contexto .....	1
2. Nociones Generales.....	7
3. Tipos de SGBD .....	21

### SQL: Lenguaje de Consultas

1. Sobre el lenguaje SQL .....	29
2. Tablas Ejemplo .....	39
3. Consultas Simples .....	41
4. Consultas de Expresión.....	53
5. Consultas de Función .....	59
6. Consultas Anidadas.....	63
7. Consultas de Grupo.....	71
8. Consultas Conjuntistas.....	77
9. Consultas Multitabla .....	87
10. LMD.....	109
11. LDD.....	113
12. Restricciones de Integridad.....	133
13. Privilegios .....	143

### Apéndices

A. Resumen de la sintaxis SQL .....	151
B Referencia de ORACLE.....	165

Bibliografía.....	171
-------------------	-----





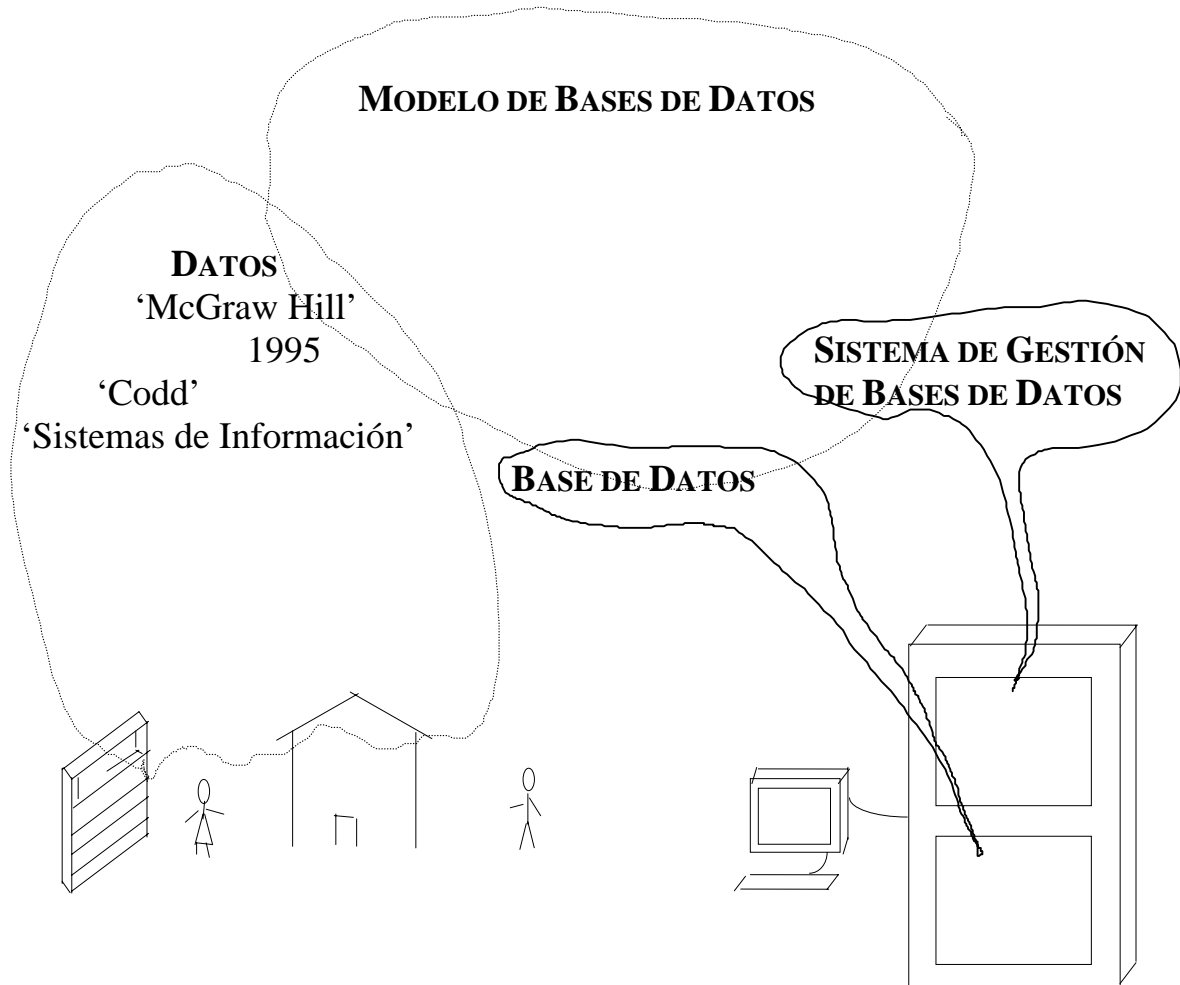
# NOCIONES BÁSICAS

## 1



# 1.- El contexto

Nuestro universo del discurso



- En los sistemas se **perciben datos**. Los sistemas se **representan mediante datos**.
- Un **modelo de base datos** nos dice cómo estructurar los datos.
- Un **sistema de gestión de bases de datos** es la herramienta soft que nos permite **gestionar automáticamente los datos** basándose en las ideas dadas en un modelo de bases de datos.

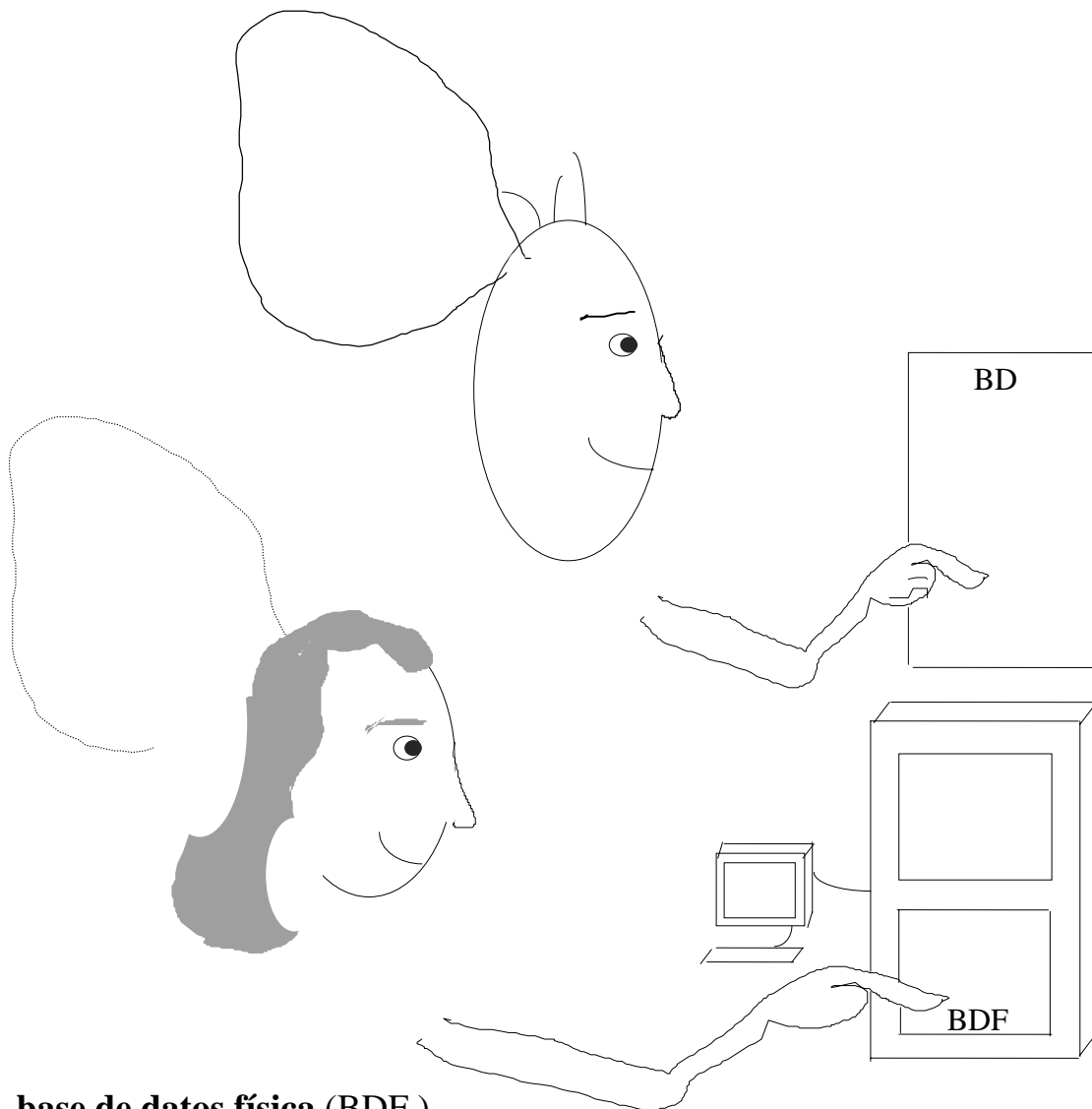
**base de datos (BD)**

\*sistema de datos sobre soporte permanente;  
es decir,

**familia de datos** relacionados entre sí y presentados sobre soporte permanente\*

Toda comunicación sobre el contenido de una base de datos es información.

Una base de datos contiene información sobre un sistema percibido y se debe percibir como representación del sistema percibido.

**base de datos física (BDF )**

\*base de datos almacenada en memoria secundaria de uno o más ordenadores\*



**usuario**

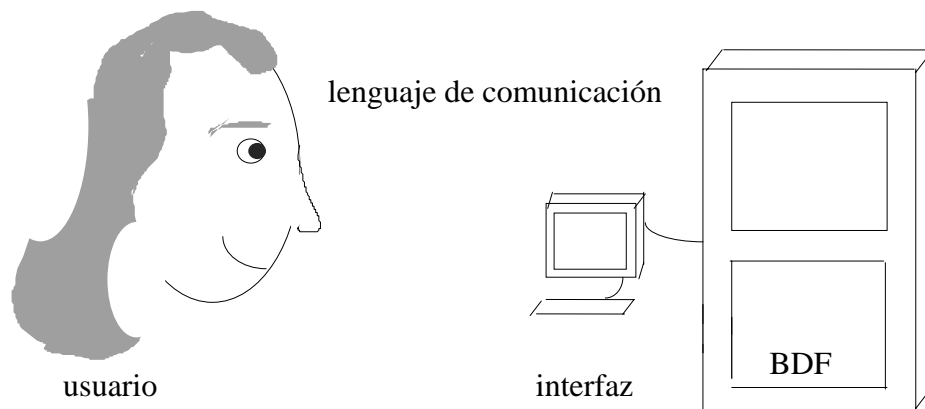
\***persona** que, potencialmente, está **interesada en la información** almacenada en una BDF\*

**interfaz**

\***sistema intermedio entre el usuario y la BDF** a través del cual se realiza la comunicación\*

**lenguaje de comunicación**

\***lenguaje** que puede ser utilizado por el usuario **para acceder a la BDF, a través de una interfaz**\*

**Tipos de usuarios****usuario final**

\***usuario que tiene interés en conocer la información** almacenada en una BDF\*

**gestor de la información**

\***usuario** que es **responsable de la información** que está almacenada en una BDF y de la información que puede ser comunicada a un usuario final\*

**administrador de la base de datos (ABD o, en inglés, DBA)**

\***usuario responsable de un sistema de gestión de bases de datos (SGBD)**\*

**sistema de gestión de bases de datos (SGBD)**

o sistema gestor de bases de datos

o gestor de base de datos (GBD)

**\*herramienta de software** que proporciona los medios necesarios **para crear y modificar bases de datos físicas**, así como **para la extracción de la información** contenida en dichas bases de datos, **proporcionando lenguajes de comunicación** al usuario\*

**No confundir** SGBD con Sistema de Bases de Datos (SBD).

Un SBD es una familia de bases de datos posiblemente relacionadas.

**No confundir** SGBD con el administrador de la base de datos (DBA).

**modelo de bases de datos (MBD)**

Familia de **nociones y métodos que proporcionan una forma** concreta **de estructurar los datos** en el contexto de las bases de datos.

**El modelo relacional es un modelo de bases de datos.**

Formas de presentar el modelo relacional: Muy variadas.

Completamente formales: en términos puramente matemáticos.

...

Completamente informales: a través de casos, en términos puramente aplicativos.

**Nuestro método pretende ser formal distinguiendo** simultáneamente **dos formas:**

Una basada en el lenguaje natural

para expresar los significados de usuario

y otra basada en el lenguaje simbólico

para expresar los significados de ordenador.

## **Modelos de bases de datos:**

- **Modelo relacional**

- Modelo jerárquico, Modelo en red
- Modelo entidad/asociación (entidad/relación),...
- ...

## **Sistemas de Gestión de bases de datos (SGBD):**

- **Oracle**

- Ingress, DB2, Informix,...
- FoxPro, Access,...
- ...

**Todo SGBD está basado sobre un modelo de base de datos.**

**Los SGBD se clasifican** en tipos **según el** tipo de **modelo de BD** sobre el que basan.

Los SGBD relacionales están basados en el modelo relacional.

**Oracle es un SGBD relacional.**

Los SGBD de un mismo tipo no recogen uniformemente los aspectos del modelo de BD en el que se basan.

- Cada SGBD tiene sus propias particularidades.
- Realmente se podría hablar de que no existe un modelo relacional, existen diferentes formas de entender lo que es el modelo relacional.

**Es necesaria una normalización.**

# RESUMEN

Nociones que es necesario **recordar**

- Base de Datos
- Modelo de Bases de Datos
- Sistema de Gestión de Bases de Datos
- Usuario
  - usuario final
  - gestor de la información
  - gestor del sistema (SGBD)
- interfaz

**No confundir** entre

- Administrador del Sistema, o DBA, o ABD
- gestor de la información
- gestor de bases de datos

Propuesta de **normalización**

Expresiones **adecuadas**

la|una interfaz

las|unas interfaces

Expresiones **no adecuadas**

el|un interfaz

los|unos interfaces

el|un interface

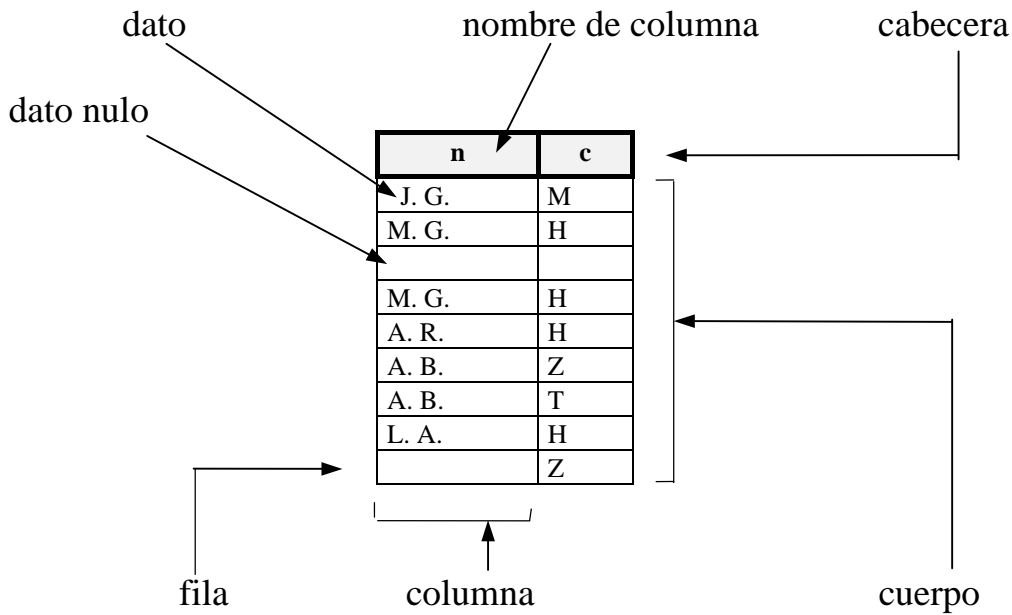
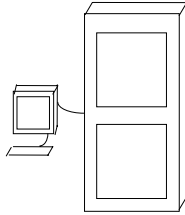
la|una|el|un interfase

## 2.- Nociones Generales

### Bajo la perspectiva de la máquina

#### tabla

\***matriz de cadenas** de un lenguaje predeterminado en la que se distinguen las partes que se indican a continuación\*



#### Principio

Toda tabla tiene cabecera.

#### Principio

Los **datos de cabecera** son todos **no nulos**.

#### Definición

Dos **tablas** que difieren sólo en el orden de las columnas o en el orden de las filas (del cuerpo) se consideran **como iguales**

**Observación**

Pueden existir datos repetidos.

n	n
n.	n
n.	n

**dato nulo** *null*

\*cadena vacía\*

\*ausencia de información\*

**tabla vacía**

\*tabla sin cuerpo\*

n	c

**Intención de uso de una tabla vacía**

Es el **estado inicial** de las tablas, es el estado en el momento en que son creadas en la máquina, o bien, estado en el que se queda la tabla cuando se han borrado todas las filas.

**Intención de uso de un nombre de columna**

**Referir a la columna** que encabeza; referir a la columna de la que forma parte como elemento.

**fila nula**

\*fila con todos los datos nulos\*

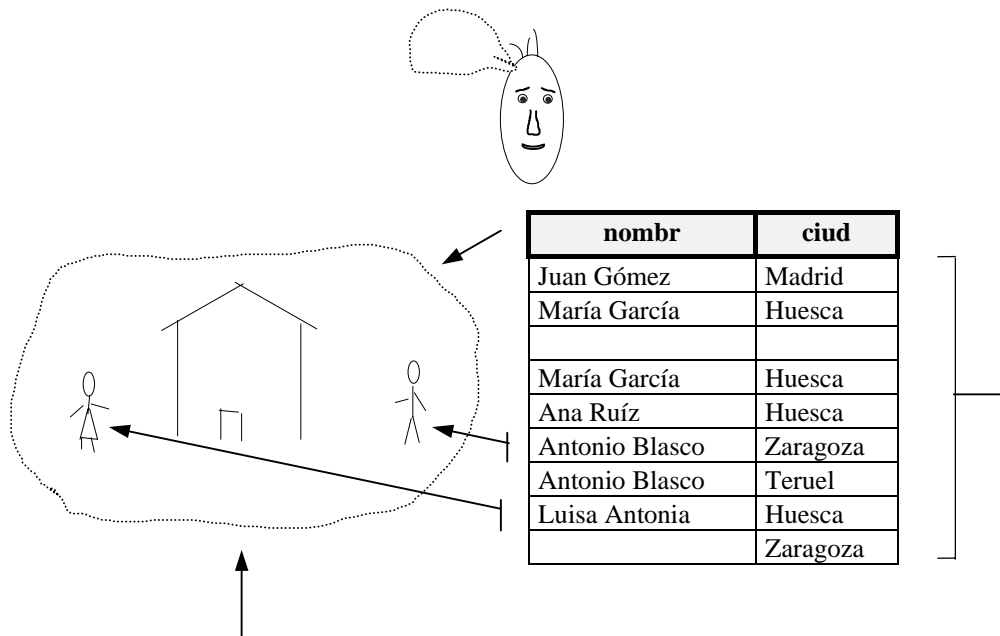
**tabla nula**

\*tabla con todas las filas nulas\*

n	c

## Bajo la perspectiva del usuario

Contiene la perspectiva explicada de la máquina y además lo que se indica a continuación.



Intenciones de uso

### Tabla

**refiere**, implícita o explícitamente, **a una familia de individuos.**

### Nombre de columna

**refiere genéricamente a un tipo de propiedad** de la familia de individuos.

### Dato del cuerpo

**refiere a la propiedad concreta**, de un individuo de la familia referida por la tabla, expresada genéricamente por su nombre de columna.

### Dato nulo

**Ausencia** en la tabla **de la propiedad concreta**, de un individuo de la familia referida por la tabla, expresada genéricamente por su nombre de columna.

### Fila

Las propiedades concretas, o sus ausencias, referidas por **los datos de una misma fila** tratan sobre **un mismo individuo.**

## Tablas nominadas

bajo cualquier perspectiva  
salvo que se exprese lo contrario

### tabla nominada

\*tabla con un nombre de la tabla\*

### Intención de uso del nombre de una tabla

Refiere a la tabla.

### Intención de uso del nombre de una tabla bajo la perspectiva del usuario

Refiere **genéricamente** a un individuo de la familia.

## Carácter de representación

### tabla base

\*tabla nominada\*

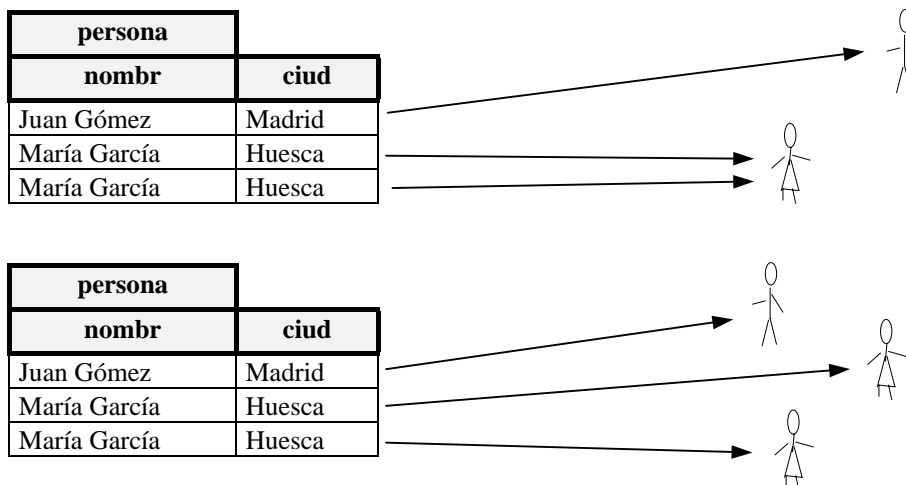
### Intención de uso de una tabla base

Cada fila representa fielmente al individuo que refiere.

### Principio

Una tabla base no debe contener filas repetidas.

### Tablas base **inadecuadas**





# Bases de Datos

tabulares

## base de datos

\***conjunto finito de tablas** base de modo que los nombres de las tablas son distintos dos a dos\*

## Intención de uso de una base de datos

Cada una de sus tablas representa fielmente una familia de individuos de un sistema.

## Notas

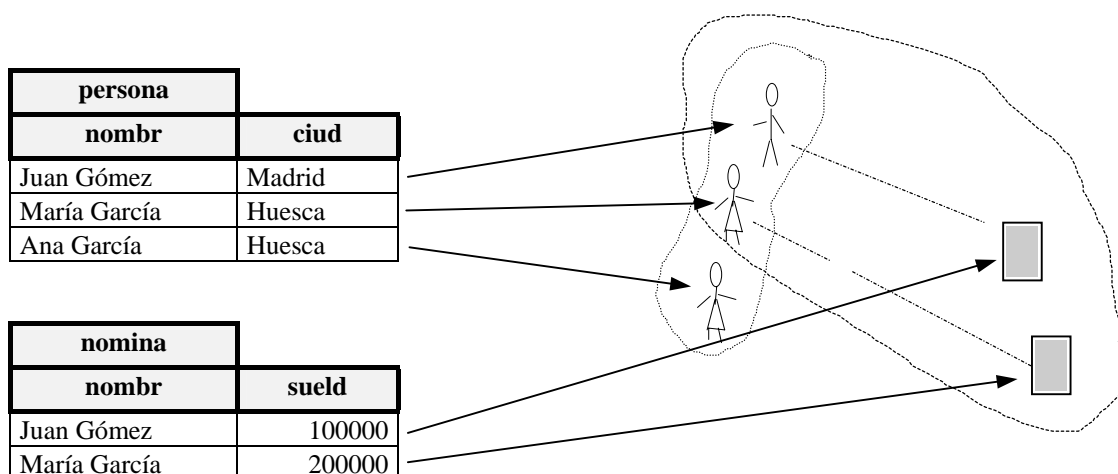
Debemos conocer explícitamente las tablas que constituyen la base de datos.

Los nombre de las tablas representan fielmente al conjunto de tablas de una base de datos (por ser distintos dos a dos).

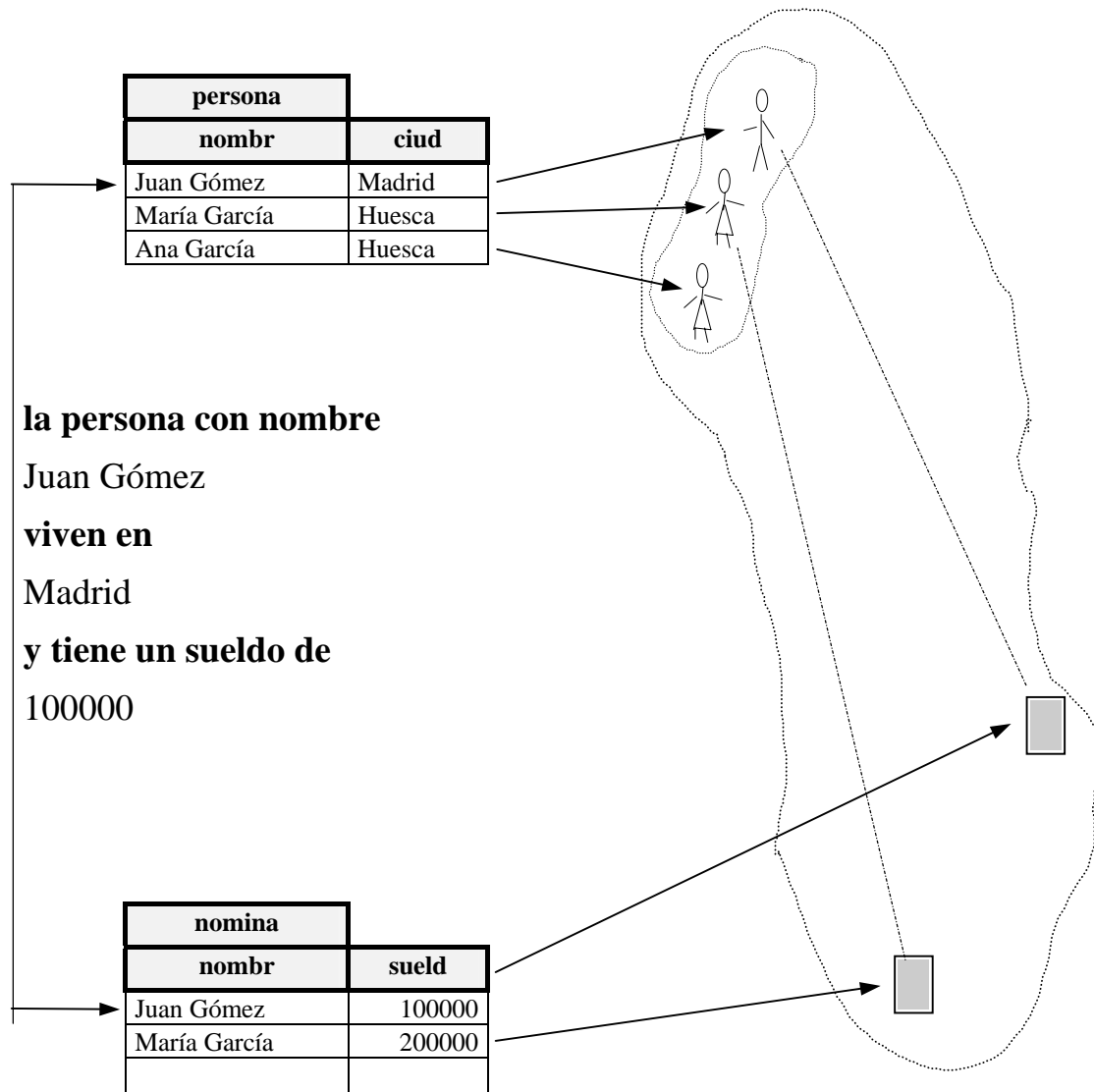
Las familias de individuos representadas por las tablas deben ser distinguibles dos a dos.

Toda base de datos debe encontrarse sobre soporte permanente.

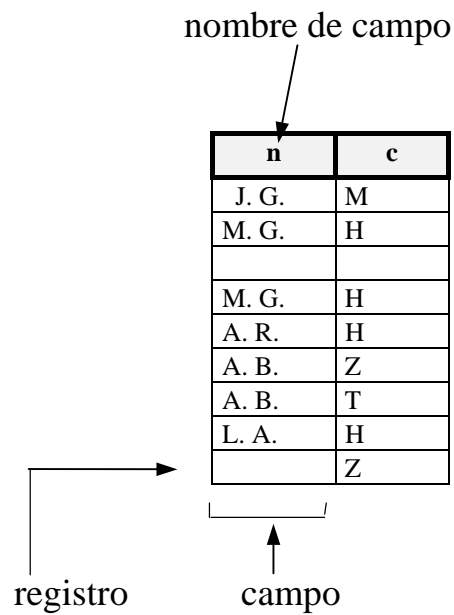
Una base de datos será base de datos física cuando el soporte sea la memoria secundaria de uno o más ordenadores.



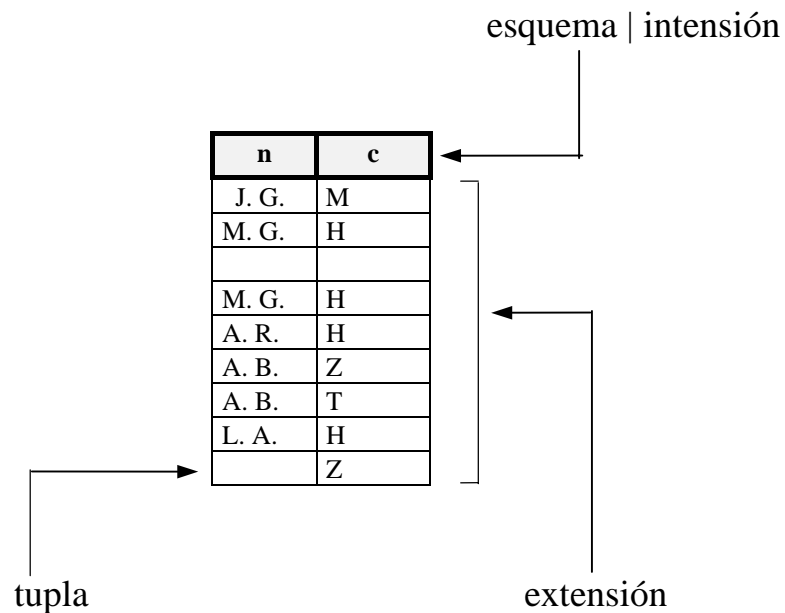
Algunas **relaciones** entre los **individuos** de distintas familias **están implícitamente representadas por relación** entre los valores de las **columnas**



## Términos inadecuados



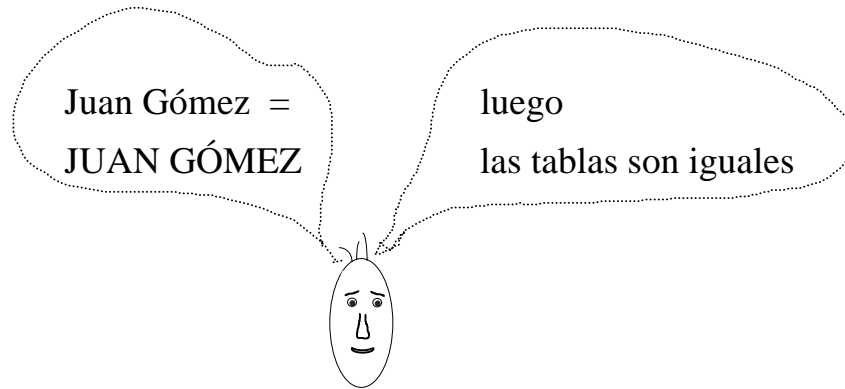
## Otra terminología posible



No es adecuado: Sustituir el nombre *tabla* por *relación*

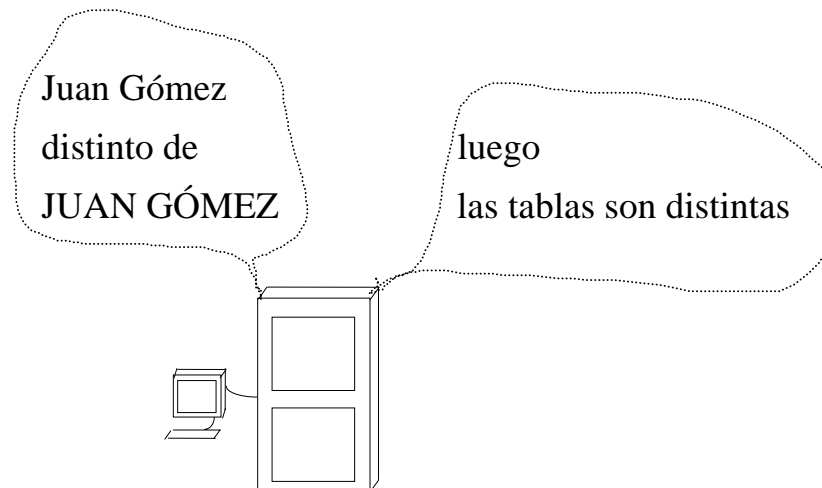
## Sobre la igualdad

¡cuidado!

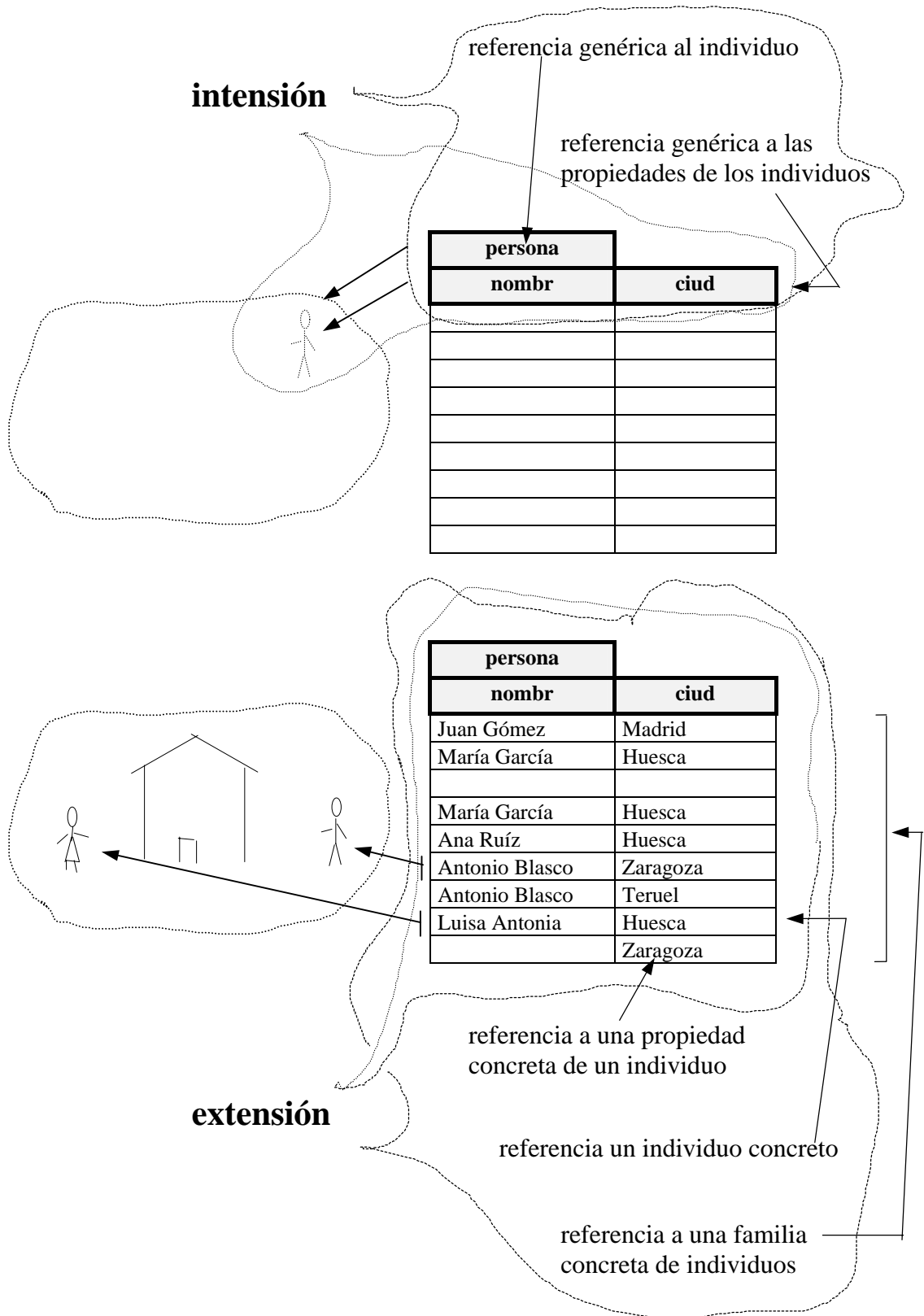


nombr	ciud
Juan Gómez	Madrid
María García	Huesca
María García	Huesca
Ana Ruíz	Huesca
Antonio Blasco	Zaragoza
Antonio Blasco	Teruel
Luisa Antonia	Huesca
	Zaragoza

nombr	ciud
JUAN GÓMEZ	Madrid
María García	Huesca
María García	Huesca
Ana Ruíz	Huesca
Antonio Blasco	Zaragoza
Antonio Blasco	Teruel
Luisa Antonia	Huesca
	Zaragoza



Normalización del lenguaje



## Normas de escritura

### 1.- Sobre los nombres de las tablas

Si tienen varios términos, separarlos mediante guión bajo; p.e. ejemplar\_perdido.

No utilizar signos de puntuación.

- Si el nombre refiere a una cosa:

Todas las letras en minúscula y, en general, en singular; p.e. monografía.

- Si el nombre refiere a una relación entre cosas representadas por tablas:

Los términos que refieren a las cosas, en minúsculas y abreviados con la regla que se describe más adelante.

El vínculo lingüístico que refiere a la relación, en mayúsculas.

P.e. ejempl\_DE\_monogr, refiriéndose a ejemplar de monografía.

### 2.- Sobre los nombres de las columnas

Si tienen varios términos, separarlos mediante guión bajo; p.e. ejemplar\_perdido.

No utilizar signos de puntuación.

Abreviar los términos con la regla que se describe más adelante.

Si un término refiere a una pluralidad, abreviar y añadir una s al final separando con guión bajo; p.e. autor\_s para autores.

No utilizar preposiciones; p.e. nombr para nombre de persona.

**Excepción:** Cuando el nombre de la columna refiere a un identificador, este último término se abrevia por id; p.e. id\_empl para identificación de empleado.

## Abreviaturas

Sustituir una palabra por

- la primera sílaba y primeras consonantes de la segunda sílaba, si con ello se consiguen cuatro o más caracteres
- las dos primeras sílabas y primeras consonantes de la segunda sílaba, en caso contrario

**estudiante**      **estud**                      **autor**                      **autor**

- Si la palabra refiere a una pluralidad, a la abreviatura se le añade al final una s

**estudiantes**      **estud\_s**                      **autores**                      **autor\_s**

**tijeras**                      **tijer** si refiere a una  
**tijer\_s** si refiere a dos o más

- En todos los casos se omiten los acentos.

## Comunicación de las tablas

Formas semigráficas

**de cabecera**

<b>persona</b>	
<b>nombr</b>	<b>ciud</b>

<b>nombr</b>	<b>ciud</b>
--------------	-------------

**de tabla**

<b>persona</b>	
<b>nombr</b>	<b>ciud</b>
Juan Gómez	Madrid
María García	Huesca
María García	Huesca

**de tabla vacía**

persona	
nombr	ciud

Formas textuales

**persona(nombr, ciud)**

## Unas nociones

**esquema de tabla**

es información sobre las partes de la cabecera y otros aspectos necesarios para la gestión de los datos (como son los dominios y las restricciones de integridad).

**esquema de BD**

es información sobre los esquemas de las tablas que van a formar parte de la base de datos.

**especificar una BD**

\*representar un esquema de BD mediante un lenguaje específico\*

**lenguaje de especificación**

\*lenguaje orientado a la especificación\*

**diseñar una BD**

\*proceso que conduce a la especificación de un esquema de BD\*



# RESUMEN

Nociones que es necesario **recordar**

- tabla
- tabla vacía
- dato nulo
- tabla base
- base de datos
- cabecera, esquema, intensión
- cuerpo, extensión
- esquema de tabla, esquema de base de datos
- especificar, diseñar

Intenciones de uso

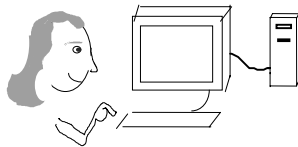
- tabla
- nombre de columna
- dato del cuerpo (siempre del cuerpo)
- dato nulo
- fila
- nombre de tabla
- tabla base
- base de datos



## 3.- Tipos de SGBD

- Según el modelo de base de datos que utilizan
  - relacional
  - jerárquico
  - red
  - orientado a objeto
  - .....
  
- Según el tipo de red que lo soporta

### **SGBD personales**

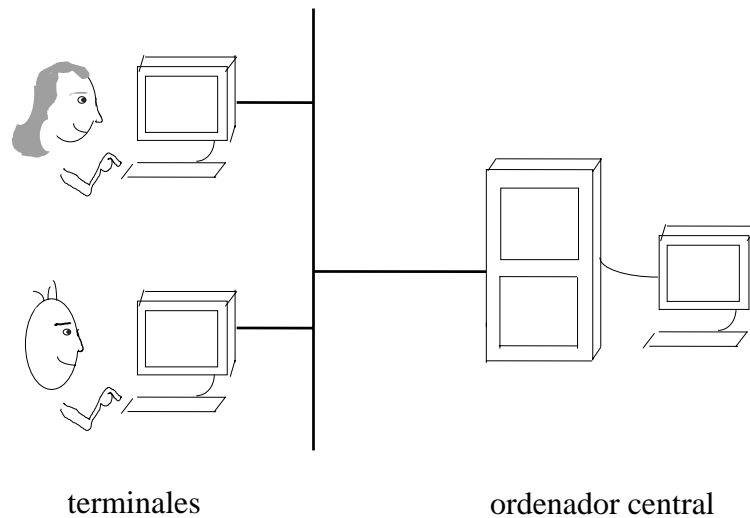


Soportado por un ordenador personal

Acceso directo

Sólo un usuario puede interactuar a la vez

## SGBD centralizados



### Soportado por una red

LAN, de área local

WAN, de área amplia

Todos los recursos del SGBD y la BD están almacenados en un ordenador central

Acceso remoto a través de una terminal

Cada terminal emula el trabajo con el ordenador central

El único software que utiliza la terminal (¡terminal tonta!) es el necesario para establecer la conexión con el ordenador central

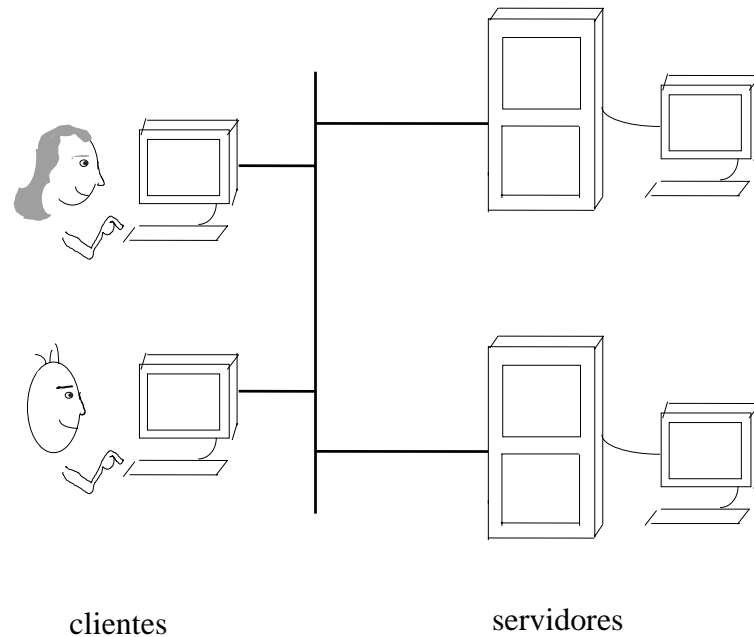
Se trabaja con la potencialidad del ordenador central

Varios usuarios pueden interactuar a la vez

Problemas de concurrencia

Problemas de integridad de datos

## SGBD distribuidos



Todos los recursos del SGBD y la BD están distribuidos en varios ordenadores llamados servidores

Las consultas se distribuyen entre los servidores

Varios usuarios pueden interactuar a la vez

Problemas de concurrencia

Problemas de integridad de datos

Las partes del SGBD soportadas por los servidores pueden ser

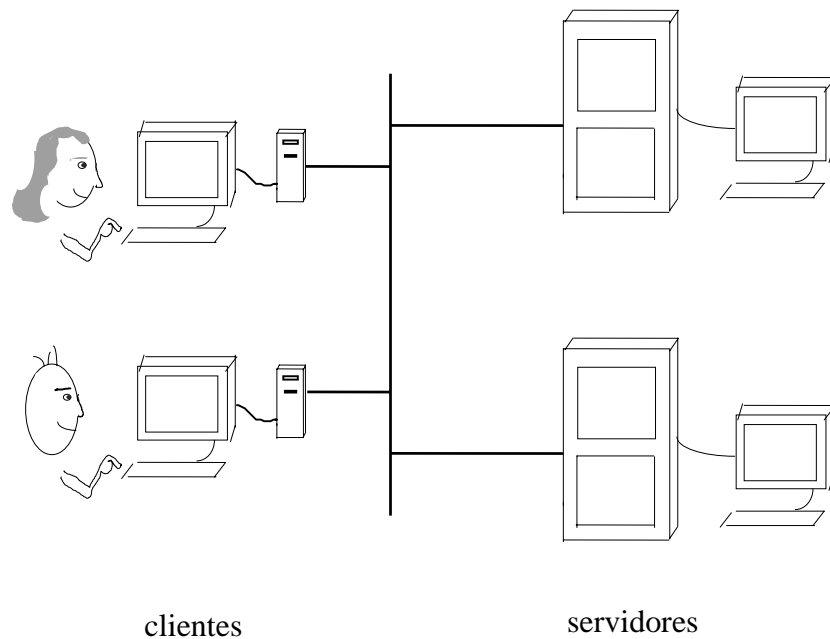
homogéneas

heterogéneas

En sistemas heterogéneos

Problema de la interoperabilidad (ODBC es una interfaz de aplicación propuesto por Microsoft en 1991 para la construcción de aplicaciones heterogéneas)

## SGBD cliente/servidor



Es un tipo de sistema distribuido, aunque en la mayoría de los casos, el SGBD está localizado en un solo servidor

El cliente tiene sus propios recursos informáticos (¡terminal inteligente!).

Es, en cierto sentido, autónomo

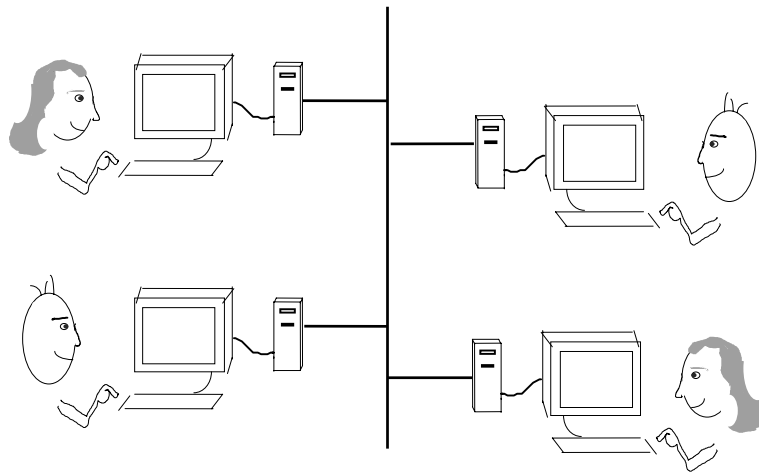
Está **dedicado al usuario final**

Genera automáticamente peticiones de servicios a los servidores

El servidor ofrece servicios

El servidor es reactivo; sólo opera como resultado de una petición por parte de algún cliente

## SGBD federado



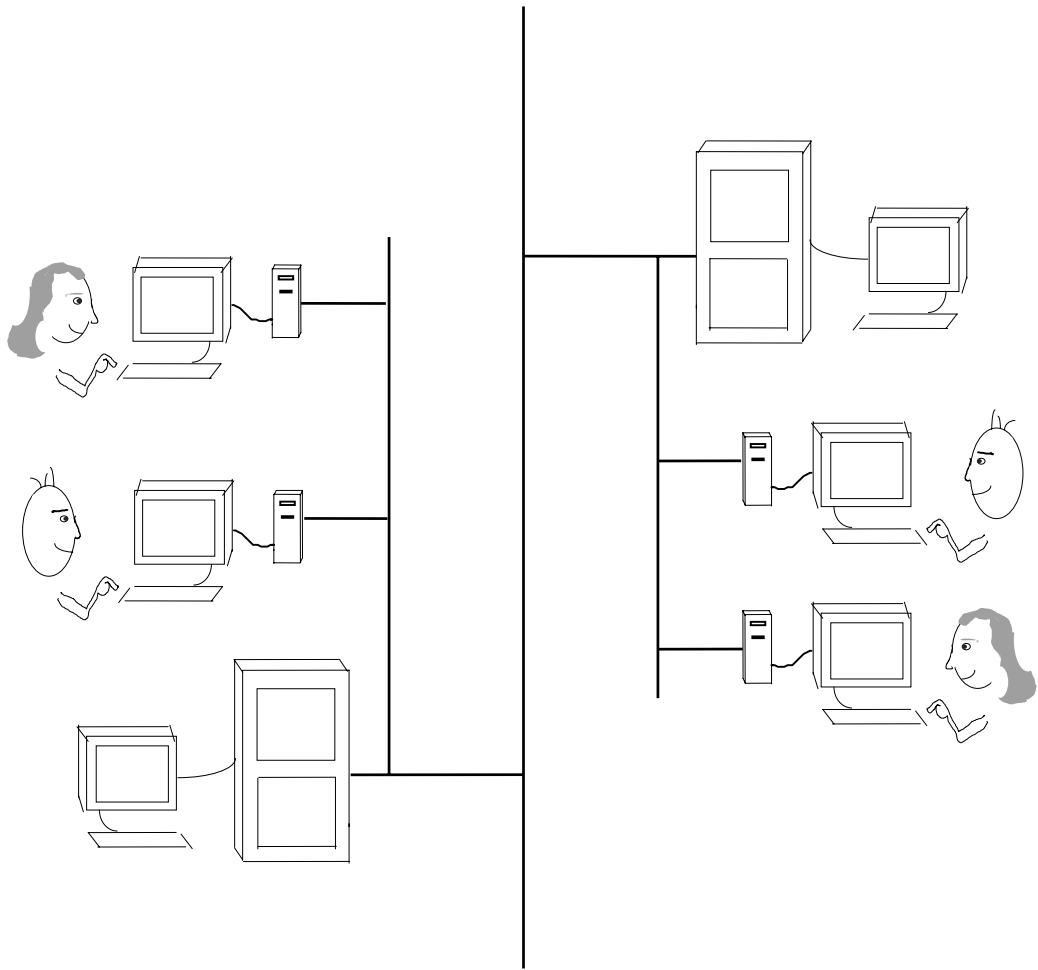
SGBD's

Es un sistema de SGBD autónomos que pueden solicitarse información

Cada SGBD tiene información privada e información pública

En general la información pública tendrá que administrarse cooperativamente entre los miembros del sistema. En ocasiones será necesario realizar una administración cooperativa de la información pública.

### Un ejemplo de arquitectura federada



SGBD's





**SQL:**

**LENGUAJE DE CONSULTAS**

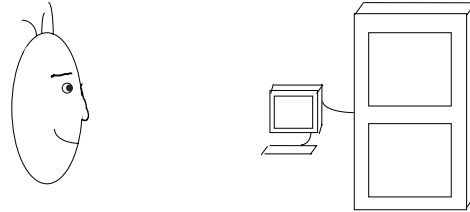
**LMD**

**LDD**



# 1. Sobre el lenguaje SQL

## Comunicación con la máquina



### **lenguaje de definición de datos (LDD)**

**lenguaje** que es **utilizado para pedir** a un SGBD la creación de **una BD y operaciones de modificación sobre el esquema** de una BD previamente implementada.

**Para la creación** de una BD se **comunica su esquema**.

Una vez creada, **se puede pedir** también **la modificación de sus partes o su supresión (anulación, el borrado)**.

### **lenguaje de modificación de datos (LMD)**

**\*lenguaje** que es utilizado **para pedir** a un SGBD **operaciones** de mantenimiento **sobre los datos** propios de una BD previamente implementada\*

Una vez creada una BD, en general, a continuación se comunican los datos que deben formar parte del cuerpo.

Una vez introducidos los datos, **se puede pedir** también **la modificación de los datos** (del cuerpo) **o borrado de los mismos**.

### **lenguaje de consultas (LC)**

**\*lenguaje** que es utilizado **para pedir** a un SGBD **la información que se desea** sobre una BD previamente implementada\*

### **lenguaje de acceso a bases de datos**

**lenguaje que tiene las características de un LDD, un LMD y un LC**

## **CUIDADO**

Se suelen intercambiar indebidamente los términos LA y LC

# SQL

## (Structured Query Language)

**Lenguaje de acceso** a bases de datos (LC+LDD+LMD+....)

**Existe una norma** actual (SQL99).

Es el lenguaje **adoptado por** la casi totalidad de **los Sistemas** de Gestión de Bases de Datos **Relacionales**

Pueden existir algunas variaciones entre las distintas herramientas comerciales. Especialmente en las utilidades que incluyen y aún no están normalizadas.

### Un poco de historia

1974	SEQUEL	Se presenta el lenguaje por primera vez bajo las siglas anteriores, correspondientes a 'Structured English Query Language'.  El primer SGBD que lo incorpora es 'System R' de IBM.
1979	SQL	Oracle lo incorpora en su SGBD, bajo las siglas anteriores, correspondientes a 'Structured Query Language'.
1986	SQL/ANS	ANSI (American National Standards Institute) presenta una norma de dicho lenguaje.
1987	SQL	ISO (International Standardization Organization) presenta una norma de dicho lenguaje.
1989	ISO-89 SQL-Addendum	Norma que mejora la integridad referencial.
1989	SQL-Embebido	Norma ANSI.
1992	SQL-2 SQL-92	Norma ISO en la que se incrementa la capacidad semántica.
1999	SQL-3 SQL-99	

Es un **lenguaje declarativo**. Es un lenguaje de cuarta generación (**4GL**).

Se **expresa lo que se quiere**.

No es procedural.

**NO** se **expresa qué se debe hacer** para conseguir lo que se quiere.

.....

El lenguaje SQL proporciona

- un conjunto finito de palabras reservadas con un significado preciso

*NULL*

\*cadena vacía\*

*table*

\*tabla\*

*SELECT*

\*cláusula de tipo *SELECT* o sentencia de tipo *SELECT*\*

Sólo pueden ser utilizadas en el contexto que se determine y con el significado que se dé.

- unas reglas sintácticas para determinar las expresiones admisibles por el lenguaje

*SELECT \*/<nombr\_column>*

\*selecciona todo o selecciona la columna nominada\*

- un significado preciso para cada una de las expresiones admisibles

*SELECT \**

\*selecciona todo \*

*SELECT <nombr\_column>*

\* selecciona la columna nominada\*

El lenguaje SQL proporciona

- la posibilidad de usar términos de usuario, con el significado propio de usuario, restringidos a ciertas reglas sintácticas muy generales.

Refieren a **metadatos** (nombres genéricos) como nombres de tablas, de columnas,...

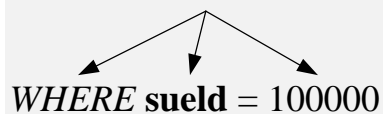
**SELECT sueld**  
\* selecciona sueldo\*

Refieren a **datos** de columnas,...

**WHERE sueld = 100000**  
\*filas en las que el dato en la columna **sueld** es 100000\*

**WHERE nombr\_pers = 'Juan García'**  
\*filas en las que el dato en la columna **nombr\_pers** es Juan García\*

Los tres tipos de niveles semánticos en las expresiones SQL



## El lenguaje SQL proporciona

- dominios para que puedan ser asignados a las columnas

Su función es delimitar sintácticamente las cadenas que pueden ser utilizadas en una columna

### *CHAR* (4)

\*cadena alfanumérica de longitud exactamente igual a 4, posiblemente vacía\*

\*Sintaxis: '<cadena>' para las cadenas no vacías  
*null* para la cadena vacía\*

### *VARCHAR* (4)

\*cadena alfanumérica de longitud máxima 4, posiblemente vacía\*

\*Sintaxis: '<cadena>' para las cadenas no vacías  
*null* para la cadena vacía\*

### *INTEGER*

\*cadena numérica\*

\*Sintaxis: <entero> para las cadenas no vacías  
*null* para la cadena vacía\*

### *NUMBER* / *NUMBER* (*ndigitos*, *precision*)

\*cadena numérica\*

\*Sintaxis: <real> para las cadenas no vacías  
*null* para la cadena vacía\*

### *DATE*

\*cadena numérica\*

\*Sintaxis: <hora, minuto, segundo, día, mes, año> para las cadenas no vacías  
*null* para la cadena vacía\*

- restricciones de integridad

Su función es delimitar los datos que pueden ser introducidos en las tablas

*not null*

\*declarada sobre una columna **impide que se introduzca un dato nulo en dicha columna\***



Ejemplo de  
especificación de una BD  
con dominios y restricciones de integridad

→ **persona**(**id\_pers** *char(3) not null*, **nombr\_pers** *varchar(40) not null*,  
**ciud** *varchar(20)*, **sueld** *number(38,2) not null*)  
\*persona con los datos de su identificación, nombre, ciudad en la  
que reside y sueldo que tiene asignado\*

→ **empresa**(**id\_empr** *char(5) not null*, **nombr\_empr** *varchar(40) not  
null*, **ciud** *varchar(20)*)  
\*empresa con los datos de su identificación, nombre y ciudad en  
la que está ubicada\*

→ **contrato**(**id\_contr** *char(6) not null*, **id\_pers** *varchar(3) not null*,  
**id\_empr** *char(5) not null*, **fecha\_contr** *date*)  
\*contrato entre una persona y una empresa con los datos del  
contrato, de la persona y de la empresa, así como la fecha en la  
que se firmó el contrato\*

esquema de tabla

esquema de base de datos

## Tipos de expresiones con sentido en SQL

### sentencia (SQL)

**\*expresión permitida por el lenguaje para la comunicación de una acción que se demanda\***

\*expresa una acción que se demanda ejecutar al SGBD en relación al acceso a las bases de datos que gestiona: crear una tabla, borrar una columna de una tabla, introducir una fila en una tabla,...\*

### cláusula (SQL)

**\*expresión, que forma parte de una sentencia, orientada a especificar algo necesario para la ejecución de la acción que se demanda\***

\*tipos de datos que se piden, condición que deben cumplir las filas sobre las que se quiere información,...\*

### Principio

Una sentencia está constituida por un conjunto finito ordenado de cláusulas.

### Principio

Toda sentencia y toda cláusula comienza con una palabra reservada que refiere a lo que expresa.

### cláusula *SELECT*

Especifica **lo que se desea**  
los tipos de datos que se desean

### sentencia *SELECT*

Especifica **una consulta a una BD.**  
Siempre comienza con las dos cláusulas siguientes

*SELECT ...*  
*FROM....*

....

conservando el orden expuesto

## Parte LC del SQL

(lenguaje de consultas)

Está constituida por un sólo tipo de sentencia

la sentencia *SELECT*

que admite formas muy variadas.

Forma elemental de la sintaxis

*SELECT* <qué se pide>

*FROM* <dónde está>

[*WHERE* <bajo qué condición>]

opcional

cláusula *SELECT*

Especifica **lo qué se desea**

los tipos de datos que se desean

cláusula *FROM*

Especifica **dónde se encuentra**

las tablas en las que se encuentra la información que se desea

cláusula *WHERE*

Especifica **bajo qué condición**

la condición que deben cumplir las filas sobre las que se pide información

# RESUMEN

Nociones que es necesario **recordar**

- lenguaje de definición de datos (LDD)
- lenguaje de modificación de datos (LMD)
- lenguaje de consultas (LC)
- lenguaje de acceso (LA)
- lenguaje SQL

## 2. Tablas Ejemplo

### monografia

```
(  
id_monogr  VARCHAR(7)      NOT NULL,  
autor_s   VARCHAR(50),  
titul     VARCHAR(100)   NOT NULL,  
edit      VARCHAR (20)    NOT NULL,  
año_publ  CHAR(4)      NOT NULL,  
volum     VARCHAR (8)  
)
```

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum

\*monografías de las que existe un ejemplar en la biblioteca, con los datos de su identificación, lista de autores, título, siglas de la editorial que la ha publicado, año de publicación e identificación del volumen en el caso en que forme parte de una obra\*

### ejempl\_DE\_monogr

```
(  
numer_registr  VARCHAR (5)      NOT NULL,  
id_monogr      VARCHAR (7)      NOT NULL,  
cant_pagad    INTEGER,  
fech_pago     DATE  
)
```

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago

\*ejemplares existentes en la biblioteca de monografías con los datos del número de registro del ejemplar, identificación de la monografía correspondiente, cantidad pagada por el ejemplar y la fecha de pago\*

monografía					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de Bases de Datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

### 3. Consultas Simples

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

#### Datos sobre las monografías

*SELECT* \*

*FROM* **monografia**

devuelve la tabla **monografia**, como **tabla FROM**

---

*SELECT*  
selecciona todas las columnas de la **tabla FROM**

id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

## Años de publicación de las monografías existentes

*SELECT* año\_publ

*FROM* monografia

devuelve la tabla **monografia**, como **tabla FROM**

---

*SELECT*

devuelve como tabla respuesta la columna **año\_publ** de la **tabla FROM**

año_publ
1993
1993
1995
1982
1990
1982
1992



Consulta

### Años de publicación de las monografías existentes

*SELECT DISTINCT* año\_publ

*FROM* monografia

devuelve la tabla **monografia**, como **tabla FROM**

*SELECT*

devuelve como tabla respuesta la columna **año\_publ** de la **tabla FROM** eliminando repeticiones en las filas de la tabla respuesta

año_publ
1993
1995
1982
1990
1992

Consulta

### Listado ordenado de los años de publicación de las monografías existentes

*SELECT DISTINCT* año\_publ

*FROM* monografia

*ORDER BY* año\_publ

devuelve la tabla **monografia**, como **tabla FROM**

ordena las filas de la tabla FROM según el orden ascendente de la columna **año\_publ**

*SELECT*

devuelve como tabla respuesta la columna **año\_publ** de la **tabla FROM** eliminando repeticiones en las filas de la tabla respuesta

año_publ
1982
1990
1992
1993
1995

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Datos bibliográficos de las monografías, presentados en el orden: título, autores, datos de edición**

```
SELECT  titul,
        volum,
        autor_s,
        edit,
        año_publ
```

*FROM* monografia

devuelve la tabla **monografia**, como **tabla FROM**

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM** descritas en *SELECT*, ordenadas (las columnas) según el orden en qué están especificadas

títul	volum	autor_s	edit	año_publ
Introducción a los Sistemas de bases de datos	I	Date, C.J.	AWI	1993
Concepción y diseño de bases de datos		De Miguel, A.; Piattini, M.	R	1993
Puesta a punto de Oracle		Corey, M.; Abbey, M.; Dechichio, D.	O/MH	1995
The Handbook of Artificial Intelligence	II	Barr, A.; Feigenbaum, E.A.	AW	1982
RDW1985-1989		Date, C.J.	AW	1990
The Handbook of Artificial Intelligence	III	Cohen, P.R.; Feigenbaum, E.A.	AW	1982
RDW1989-1991		Date, C.J.	AW	1992

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías publicadas por la editorial AWI

*SELECT* \*

*FROM* **monografia**

*WHERE* **edit = 'AWI'**

devuelve la tabla **monografia**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **edit** es AWI

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM** (modificada)

id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Monografías publicadas por la editorial AWI o bien publicadas por AW pero un año posterior o igual a 1990**

```

SELECT *
FROM monografia
WHERE edit = 'AWI'
OR (
    edit = 'AW'
    AND año_publ >= '1990'
)

```

devuelve la tabla **monografia**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **edit** es AWI, o bien, es AW y además en **año\_publ** el valor es 1990 o posterior

**SELECT**  
devuelve como tabla respuesta las columnas de la **tabla FROM** (modificada)

id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías publicadas por Date

*SELECT* \*

*FROM* **monografia**

*WHERE* **autor\_s** *LIKE* '%Date%'

devuelve la tabla **monografia**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **autor\_s** contiene como subcadena Date

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM** (modificada)

id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías publicadas en los noventa

*SELECT* \*

*FROM* monografia

*WHERE* año\_publ *LIKE* '199\_'

devuelve la tabla **monografia**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **año\_publ** tiene cuatro caracteres y los tres primeros son 199

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM** (modificada)

id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías que forman parte de una obra

*SELECT* \*

*FROM* **monografia**

*WHERE* **volum IS NOT NULL**

devuelve la tabla **monografia**, como **tabla FROM**

-----  
 modifica la **tabla FROM** seleccionando las filas para las que existe un valor en la columna **volum**

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM** (modificada)

id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

### Consulta

**Títulos y autores, en orden descendente por año de publicación y, subordinadamente, por orden de autores, de las monografías que no forman parte de una obra**

<i>SELECT</i>	<b>títul,</b>	
	<b>autor_s</b>	
<i>FROM</i>	<b>monografia</b>	devuelve la tabla <b>monografia</b> , como <b>tabla FROM</b>
<i>WHERE</i>	<b>volum IS NULL</b>	modifica la <b>tabla FROM</b> seleccionando las filas para las que no existe un valor en la columna <b>volum</b>
<i>ORDER BY</i>	<b>año_publ DESC,</b>	ordena las filas de la tabla FROM por orden descendente de <b>año_publ</b> y, en su caso, por orden ascendente de la columna <b>autor_s</b>
	<b>autor_s</b>	
<i>SELECT</i>		devuelve como tabla respuesta las columnas <b>títul</b> y <b>autor_s</b> de la <b>tabla FROM</b>

títul	autor_s
RDW1985-1989	Date, C.J.
RDW1989-1991	Date, C.J.
Concepción y diseño de bases de datos	De Miguel, A.; Piattini, M.
Puesta a punto de Oracle	Corey, M.; Abbey, M.; Dechichio, D.



monografia					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías publicadas en el año 1982, en el 90 o en el 92

*SELECT*

\*

*FROM*

**monografia**

*WHERE*

**año\_publ IN ('1982','1990','1992')**

devuelve la tabla **monografia**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que **año\_publ** es 1982, 1990 o 1992

*SELECT*

devuelve como tabla respuesta las columnas de la **tabla FROM**

id_monogr	autor_s	título	edit	año_publ	volum
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

*SELECT*

\*

*FROM*

**monografia**

*WHERE*

**año\_publ = '1982'  
OR año\_publ = '1990'  
OR año\_publ = '1992'**



## 4. Consultas de expresión

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Las cantidades pagadas aumentadas en un 10 por ciento de los libros comprados en 1999**

*SELECT* cant\_pagad + (cant\_pagad\*10/100)

*FROM* ejempl\_DE\_monogr

*WHERE* fech\_pago LIKE '%1999'

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

-----  
 modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **fech\_compr** es 1999

*SELECT*

devuelve una tabla con una sola columna con nombre la expresión especificada en el *SELECT*, con filas en correspondencia con las filas de la **tabla FROM**, de modo que el dato en cada fila devuelta por *SELECT* es el resultado de calcular la expresión

**cant\_pagad+ cant\_pagad\*10/100**,  
 sustituyendo **cant\_pagad** por el valor que tiene esa columna en la fila

cant_pagad + cant_pagad*10/100
7480
6160
6215

¿A quién se refiere?

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

## Consulta

**Las cantidades pagadas aumentadas en un 10 por ciento de los libros comprados en 1999**

```

SELECT  cant_pagad + cant_pagad*10/100 ,
        numer_registr,
        id_monogr
FROM    ejempl_DE_monogr
WHERE   fech_pago LIKE '%1999'

```

cant_pagad + cant_pagad*10/100	numer_registr	id_monogr
7480	06543	DATE90a
6160	00190	DATE93a
6215	02000	DATE93a

## ALIAS DE COLUMNA

Consulta

**Las cantidades pagadas aumentadas en un 10 por ciento de los libros comprados en 1999**

```
SELECT  cant_pagad + cant_pagad*10/100 "precio real",  
          numer_registr ejemplar,  
          id_monogr "monografía"  
FROM    ejempl_DE_monogr  
WHERE   fech_pago like '%99'
```

precio real	EJEMPLAR	monografía
7480	06543	DATE90a
6160	00190	DATE93a
6215	02000	DATE93a

monografía					
id_monogr	autor_s	título	edit	año_public	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Listado de la información sobre las monografías existentes con el formato de salida siguiente**

**<autor\_s>||-||título||,-||edit||,-||año\_public||.**

**ordenado por autores**

**SELECT autor\_s||' ||título||', '||edit||', '||año\_public||'.**

**FROM monografía**

**ORDER BY autor\_s**

devuelve la tabla **monografía**, como tabla **FROM**

ordena las filas de **tabla FROM** por la columna de los autores

**SELECT**

devuelve una tabla con una sola columna con nombre la expresión especificada en el **SELECT** con filas en correspondencia con las filas de la **tabla FROM**, de modo que el dato en cada fila devuelta por **SELECT** es el resultado de concatenar, según la expresión del **SELECT**, los datos de las correspondientes filas de la **tabla FROM**

<b>autor_s  '   título  ', '  edit  ', '  año_public  '.</b>
Barr, A.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, AW, 1982.
Cohen, P.R.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, AW, 1982.
Corey, M.; Abbey, M.; Dechichio, D. Puesta a punto de Oracle, O/MH, 1995.
Date, C.J. Introducción a los Sistemas de bases de datos, AWI, 1993.
Date, C.J. RDW1989-1991, AW, 1992.
Date, C.J. RDW1985-1989, AW, 1990.
De Miguel, A.; Piattini, M. Concepción y diseño de bases de datos, R, 1993.

monografía					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Listado de la información sobre las monografías existentes con el formato de salida siguiente**

**<autor\_s>||-||títul||,-||edit||,-||año\_publ||.**

**ordenado por autores**

```

SELECT    autor_s||' '||títul||', '||edit||', '||año_publ||'. ' "monografías"
FROM      monografía
order by  autor_s

```

monografías
Barr, A.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, AW, 1982.
Cohen, P.R.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, AW, 1982.
Corey, M.; Abbey, M.; Dechichio, D. Puesta a punto de Oracle, O/MH, 1995.
Date, C.J. Introducción a los Sistemas de bases de datos, AWI, 1993.
Date, C.J. RDW1989-1991, AW, 1992.
Date, C.J. RDW1985-1989, AW, 1990.
De Miguel, A.; Piattini, M. Concepción y diseño de bases de datos, R, 1993.





## 5. Consultas de función

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Cantidad media pagada por un libro.**

```
SELECT  AVG(cant_pagad)
FROM    ejempl_DE_monogr
```

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

---

*SELECT*

devuelve una tabla con una sola fila y una sola columna que contiene como dato el valor de la función *AVG* (valor medio) calculado sobre la columna **cant\_pagad** de la **tabla FROM**

<i>AVG(cant_pagad)</i>
7112,5

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Número de ejemplares físicos de monografías.**

```
SELECT COUNT(numer_registr)
FROM   ejempl_DE_monogr
```

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

**SELECT**

devuelve una tabla con una sola fila y una sola columna que contiene como dato el valor de la función *COUNT* (número de datos de columna) calculado sobre **numer\_registr** de la **tabla FROM**

<b>COUNT(numer_registr)</b>
12

```
SELECT COUNT(DISTINCT numer_registr)
FROM   ejempl_DE_monogr
```

**¡asegurarse!**

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

**SELECT**

devuelve una tabla con una sola fila y una sola columna que contiene como dato el valor de la función *COUNT* (número de datos de columna) calculado sobre los **datos no repetidos** de la columna **numer\_registr** en la **tabla FROM**

```
SELECT COUNT(*)
FROM ejempl_DE_monogr
```

**¡cuidado!**

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

---

**SELECT**  
devuelve una tabla con una sola fila y una sola columna que contiene como dato el **número de filas de la tabla FROM**

```
SELECT COUNT (DISTINCT * )
FROM ejempl_DE_monogr
```

**ERROR**

```
SELECT DISTINCT COUNT(*)
FROM ejempl_DE_monogr
```

```
SELECT COUNT(*)
FROM ejempl_DE_monogr
```

---

**SELECT DISTINCT**  
**SELECT**  
tienen los mismos efectos sobre expresiones funcionales

.....

Consulta

**Número de ejemplares físicos de monografías, cantidad media, máxima cantidad y mínima cantidad pagada por un libro, así como la suma total de las cantidades registradas.**

```
SELECT COUNT (DISTINCT numer_registr) "número de ejemplares",
AVG (cant_pagad) "precio medio",
MAX (cant_pagad) "precio máximo",
MIN(cant_pagad) "precio mínimo",
SUM (cant_pagad) total
FROM ejempl_DE_monogr
```

número de ejemplares	precio medio	precio máximo	precio mínimo	total
12	7112,5	10000	4200	85350



## 6. Consultas anidadas sobre una tabla

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1996
03215	DATE92a	7400	29-MAR-1996
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1996
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Mínima cantidad pagada por un libro.**

```
SELECT MIN(cant_pagad)
FROM   ejempl_DE_monogr
```

<i>min(cant_pagad)</i>
4200

¿De qué libro?

```
SELECT MIN(cant_pagad),
       numer_registr,
       id_monogr
FROM   ejempl_DE_monogr
```

*ERROR*

La función *min(cant\_pagad)* se calcula sobre la tabla y devuelve un número

La columna **numer\_registr**, como función, se calcula sobre la tabla y devuelve una lista de datos (todos los de la columna nominada)

ejempl_DE_monogr
------------------

numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1991
03215	DATE92a	7400	29-MAR-1991
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1991
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Mínima cantidad pagada por un libro, junto con las identificaciones del libro.**

```

SELECT cant_pagad "cantidad mínima",
       numer_registr,
       id_monogr
FROM   ejempl_DE_monogr
WHERE  cant_pagad = (SELECT MIN (cant_pagad) "cantidad mínima"
                    FROM ejempl_DE_monogr)

```

devuelve la tabla **ejempl\_DE\_monogr**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **cant\_pagad** es igual al que devuelve la consulta anidada

**SELECT**  
devuelve una tabla con las columnas de la **tabla FROM** modificada que se han especificado en **SELECT**

"cantidad mínima"	numer_registr	id_monogr
4200	07878	BaFe82a

```

SELECT cant_pagad "cantidad mínima",
       numer_registr,
       id_monogr
FROM   ejempl_DE_monogr
WHERE  cant_pagad =

```

<b>cantidad mínima</b>
------------------------

4200
------

monografía					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

### Monografías disponibles publicadas el año 1999

```

SELECT      *
FROM        monografía
WHERE       año_publ = '1999'

```

id_monogr	autor_s	títul	edit	año_publ	volum

monografía					
id_monogr	autor_s	título	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Monografías disponibles publicadas en el año más reciente sobre el que se tiene una monografía.**

```

SELECT *
FROM monografía
WHERE año_publ = (
                    SELECT MAX(año_publ)
                    FROM monografía
                )

```

```

SELECT *
FROM monografía
WHERE año_publ = max(año_publ)
                  1995

```

id_monogr	autor_s	título	edit	año_publ	volum
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	



monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1991
03215	DATE92a	7400	29-MAR-1991
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1991
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

### Monografías de las que no se tienen ejemplares

```

SELECT*
FROM monografia
WHERE id_monogr NOT IN
(
    SELECT DISTINCT id_monogr
    FROM ejempl_DE_monogr
)

```

id_monogr	autor_s	Títul	edit	año_publ	volum

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1991
03215	DATE92a	7400	29-MAR-1991
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1991
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Ejemplares para los que el precio de compra es mayor o igual que alguna de las cantidades pagadas el 22 de febrero de 1997 o el 23 de marzo del mismo año.**

```

SELECT *
FROM   ejempl_DE_monogr
WHERE  cant_pagad >= ANY
      (
        SELECT cant_pagad
        FROM   ejempl_DE_monogr
        WHERE  fech_pago IN
              ('22-FEB-1997','23-MAR-1997')
      )

```

-----  
 modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **cant\_pagad** es mayor o igual que alguno de los datos devueltos por la consulta anidada

numer_registr	id_monogr	cant_pagad	fech_pago
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
09863	CoFe82a	8500	25-JUN-1991

ejempl_DE_monogr			
numer_registr	id_monogr	cant_pagad	fech_pago
00178	DATE93a	5400	01-ENE-1998
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
10056	MiPi93a	7900	14-DIC-1998
09514	CoAD95a	10000	20-FEB-1998
06543	DATE90a	6800	15-ABR-1999
09863	CoFe82a	8500	25-JUN-1991
03215	DATE92a	7400	29-MAR-1991
00190	DATE93a	5600	12-MAY-1999
07878	BaFe82a	4200	05-NOV-1991
08907	MiPi93a	7500	14-DIC-1998
02000	DATE93a	5650	12-OCT-1999

Consulta

**Ejemplares para los que el precio de compra es mayor o igual que todas las cantidades pagadas el 22 de febrero de 1997 o el 23 de marzo del mismo año.**

```

SELECT *
FROM   ejempl_DE_monogr
WHERE  cant_pagad >= ALL
      (
        SELECT cant_pagad
        FROM   ejempl_DE_monogr
        WHERE  fech_pago IN
              ('22-FEB-1997','23-MAR-1997')
      )

```

-----  
 modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **cant\_pagad** es mayor o igual que todos los datos devueltos por la consulta anidada

numer_registr	id_monogr	cant_pagad	fech_pago
08908	MiPi93a	7900	22-FEB-1997
09843	MiPi93a	8500	23-MAR-1997
09514	CoAD95a	10000	20-FEB-1998
09863	CoFe82a	8500	25-JUN-1991



## 7. Consultas de Grupo

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Encontrar el número de monografías disponibles, sin contar los ejemplares repetidos, por cada editorial**

*SELECT* **edit,**

*COUNT* (\*)

*FROM* **monografia**

*GROUP BY* **edit**

devuelve la tabla **monografia**, como **tabla FROM**

-----  
agrupa las filas; cada grupo es el conjunto de filas que tienen el mismo valor en **edit**

*SELECT*

devuelve una tabla con las columnas **edit** y *COUNT* (\*)

devuelve una fila por cada grupo en **edit** la fila tiene el valor común del grupo en esa columna

en cada función de grupo, como *COUNT* (\*), la fila toma el valor de la función calculado sobre el grupo

id_monogr	autor_s	títul	edit	año_publ	volum
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	

edit	COUNT (*)
AW	4
AWI	1
O/MH	1
R	1

```

SELECT  edit,
        COUNT (id_monogr)
FROM    monografia
GROUP BY edit

```

```

SELECT  edit,
        COUNT ( DISTINCT id_monogr )
FROM    monografia
GROUP BY edit

```

monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Encontrar el número de monografías disponibles, sin contar los ejemplares repetidos, por cada editorial pero sólo de aquellas editoriales de las que se dispone de dos o más monografías**

```

SELECT  edit,
        COUNT (*)
FROM    monografia
GROUP BY edit
HAVING  COUNT (*) >=2

```

devuelve la tabla **monografia**, como **tabla FROM**

agrupa las filas; cada grupo es el conjunto de filas que tienen el mismo valor en **edit**

selecciona los grupos que cumplen la condición  $COUNT (*) \geq 2$

**SELECT**

devuelve una tabla con las columnas **edit** y **COUNT (\*)**

devuelve una fila por cada grupo en **edit** la fila tiene el valor común del grupo en esa columna

en cada función de grupo, como **COUNT (\*)**, la fila toma el valor de la función calculado sobre el grupo

id_monogr	autor_s	títul	edit	año_publ	volum
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	

edit	COUNT (*)
AW	4



monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

## Consulta

**Encontrar el número de monografías disponibles por cada editorial, sin contar los ejemplares repetidos, publicadas en los noventa pero sólo de aquellas editoriales de las que se dispone de dos o más monografías. Presentar la información ordenada según el número de monografías.**

*SELECT* edit,

*COUNT* (\*)

*FROM* monografía

*WHERE* año\_publ *LIKE* '\_\_99'

*GROUP BY* edit

*HAVING* *COUNT* (\*) >=2

*ORDER BY* *COUNT* (\*)

devuelve la tabla **monografía**, como **tabla FROM**

modifica la **tabla FROM** seleccionando las filas para las que el valor en la columna **fech\_compr** es una cadena terminada en 99

agrupa las filas; cada grupo es el conjunto de filas que tienen el mismo valor en **edit**

selecciona los grupos que cumplen la condición *COUNT* (\*) >=2

ordena los grupos por orden ascendente de *COUNT* (\*)

*SELECT*

devuelve una tabla con las columnas edit y *COUNT* (\*)

devuelve una fila por cada grupo en **edit** la fila tiene el valor común del grupo en esa columna

en cada función de grupo, como *COUNT* (\*), la fila toma el valor de la función calculado sobre el grupo



## 8. Consultas conjuntistas

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Autores que han publicado en 1992 y en 1993**

```

SELECT autor_s autores
FROM monografia
WHERE año_publ = '1992'

INTERSECT

SELECT autor_s
FROM monografia
WHERE año_publ = '1993'

```

autores
Date, C.J.

*INTERSECT*

autor_s
Date, C.J.
De Miguel, A.; Piattini, M.

Autores
Date, C.J.

monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Autores que han publicado en 1992 o en 1993**

```

SELECT      autor_s autores_92_93
FROM        monografía
WHERE       año_publ = '1992'

UNION

SELECT      autor_s
FROM        monografía
WHERE       año_publ = '1993'
    
```

autores_92_93
Date, C.J.

UNION

autor_s
Date, C.J.
De Miguel, A.; Piattini, M.

autores_92_93
Date, C.J.
De Miguel, A.; Piattini, M.

```

SELECT autor_s autores_92_93
FROM    monografía
WHERE   año_publ IN ('1992', '1993')
    
```

monografía					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Autores que han publicado en 1992 y no han publicado en 1993**

```
SELECT autor_s autores
FROM monografía
WHERE año_publ = '1992'
```

MINUS

```
SELECT autor_s
FROM monografía
WHERE año_publ = '1993'
```

autores
Date, C.J.

MINUS

autor_s
Date, C.J. De Miguel, A.; Piattini, M.

autores

monografía					
id_monogr	autor_s	títul	edit	año_publ	volum

Consulta

**Monografías disponibles publicadas en los dos últimos años contados desde el año más reciente sobre el que se tiene una monografía.**

```

SELECT      *
FROM        monografía
WHERE       año_publ IN
            (
              SELECT      MAX (año_publ) "dos últimos años"
              FROM        monografía
              UNION
              SELECT      año_publ "penúltimo año"
              FROM        monografía
              WHERE       año_publ >= ALL
            (
              SELECT      año_publ "años anteriores"
              FROM        monografía
              MINUS
              SELECT      MAX (año_publ) "último año"
              FROM        monografía
            )
            )
    
```

```

SELECT      *
FROM        monografía
WHERE       año_publ IN (
              SELECT      MAX (año_publ) "dos últimos años"
              FROM        monografía
              UNION
              SELECT      año_publ "penúltimo año"
              FROM        monografía
              WHERE       año_publ >= ALL (
    
```

años anteriores
1993
1995
1982
1990
1992

MINUS

último año
1995

)  
)

```

SELECT*
FROM monografia
WHERE año_publ IN
(
    SELECT max(año_publ) "dos últimos años"
    FROM monografia
    UNION
    SELECT año_publ "penúltimo año"
    FROM monografia
    WHERE año_publ >= (
        años anteriores
        1993
        1982
        1990
        1992
    )
)
    
```

```

SELECT*
FROM monografia
WHERE año_publ IN (
    dos últimos años
    1995
    UNION
    penúltimo año
    1993
)
    
```

```

SELECT*
FROM monografia
WHERE año_publ IN (
    dos últimos años
    1995
    1993
)
    
```

monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	

ejempl_DE_monogr			
numer_registr	id_monogr	prec_compr	fech_pago
08908	MiPi93a	7900	22-FEB-97
09843	MiPi93a	8500	23-MAR-97
10056	MiPi93a	7900	14-DIC-98

Consulta

**Número de registro de ejemplar, identificación de la monografía correspondiente y precio pagado por el ejemplar atendiendo a lo siguiente: En el caso de que existan monografías de las que no se dispone de ningún ejemplar que aparezcan las columnas que correspondan con valor nulo.**

```

SELECT numer_registr,
       id_monogr,
       cant_pagad
FROM   ejempl_DE_monogr
UNION
SELECT NULL,
       id_monogr,
       NULL
FROM   monografia
WHERE  id_monogr NOT IN
      (
        SELECT id_monogr
        FROM   ejempl_DE_monogr
      )
    
```

numer_registr	id_monogr	prec_compr
08908	MiPi93a	7900
09843	MiPi93a	8500
10056	MiPi93a	7900
	Date93a	



monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Listado de la información sobre las monografías existentes con el formato de salida siguiente**

**<autor\_s>||-||título||,-vol.-||volum||,-||edit||,-||año\_publ||.**

**ordenado por autores**

**SELECT autor\_s||' ||título||', vol. ||volum||', ||edit||', ||año\_publ||'.**

**FROM monografía**

**ORDER BY autor\_s**

devuelve la tabla **monografía**, como **tabla FROM**

ordena las filas de **tabla FROM** por la columna de los autores

**SELECT**

devuelve una tabla con una sola columna con nombre la expresión especificada en el **SELECT** con filas en correspondencia con las filas de la **tabla FROM**, de modo que el dato en cada fila devuelta por **SELECT** es el resultado de concatenar, según la expresión del **SELECT**, los datos de las correspondientes filas de la **tabla FROM**

<b>autor_s  '   título  ', vol.   volum  ',   edit  ',   año_publ  '.</b>
Barr, A.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, vol. II, AW, 1982.
Cohen, P.R.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, vol. III, AW, 1982.
Corey, M.; Abbey, M.; Dechichio, D. Puesta a punto de Oracle, <b>vol. ,</b> O/MH, 1995.
Date, C.J. Introducción a los Sistemas de bases de datos, vol. I, AWI, 1993.
Date, C.J. RDW1989-1991, <b>vol. ,</b> AW, 1992.
Date, C.J. RDW1985-1989, <b>vol. ,</b> AW, 1990.
De Miguel, A.; Piattini, M. Concepción y diseño de bases de datos, <b>vol. ,</b> R, 1993.

monografía					
id_monogr	autor_s	título	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

Consulta

**Listado de la información sobre las monografías existentes con el formato de salida siguiente**

**<autor\_s>||-||título||,-vol.-||volum||,-||edit||,-||año\_publ||.**

**ordenado por autores**

```

SELECT autor_s||' '||título||', vol. '||volum||', '||edit||', '||año_publ||'.'
FROM monografía
WHERE volum IS NOT NULL
UNION
SELECT autor_s||' '||título||', '||edit||', '||año_publ||'.'
FROM monografía
WHERE volum IS NULL
ORDER BY 1
    
```

<b>autor_s  ' '  título  ', vol. '  volum  ', '  edit  ', '  año_publ  '.'</b>
Barr, A.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, vol. II, AW, 1982.
Cohen, P.R.; Feigenbaum, E.A. The Handbook of Artificial Intelligence, vol. III, AW, 1982.
Corey, M.; Abbey, M.; Dechichio, D. Puesta a punto de Oracle, O/MH, 1995.
Date, C.J. Introducción a los Sistemas de bases de datos, vol. I, AWI, 1993.
Date, C.J. RDW1989-1991, AW, 1992.
Date, C.J. RDW1985-1989, AW, 1990.
De Miguel, A.; Piattini, M. Concepción y diseño de bases de datos, R, 1993.

En las consultas conjuntistas...

... las consultas sobre las que se apliquen han de tener el mismo número de columnas, y de igual tipo.

... se eliminan las filas duplicadas

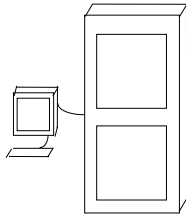
... en pantalla salen los nombres y alias de primera consulta.

... sólo puede haber, como mucho, una única cláusula 'Order by', sólo al final, y valen alias de primera consulta.



## 9. Consultas Multitabla

### Tabla producto



p		
a	b	c
a1	b1	c1
a2	b2	c2

q		
d	e	f
d1	e1	f1
d2	e2	f2
d3	e3	f3

a1	b1	c1	d1	e1	f1
a1	b1	c1	d2	e2	f2
a1	b1	c1	d3	e3	f3

a2	b2	c2	d1	e1	f1
a2	b2	c2	d2	e2	f2
a2	b2	c2	d3	e3	f3

a	b	c	d	e	f
a1	b1	c1	d1	e1	f1
a1	b1	c1	d2	e2	f2
a1	b1	c1	d3	e3	f3
a2	b2	c2	d1	e1	f1
a2	b2	c2	d2	e2	f2
a2	b2	c2	d3	e3	f3

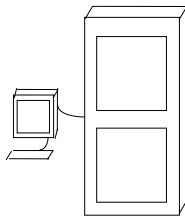
Nota

Si la tabla p tiene m filas y la tabla q tiene n filas, **el número de filas de la tabla producto es el producto de m por n.**

## Enlaces

### Definición

Se llama enlace a una **comparación intensional entre dos columnas** derivadas de las tablas especificadas en la cláusula *FROM*; es decir, a una expresión que describe una comparación genérica entre dos columnas (propias o derivadas de una tabla).



p		
a	b	c
a1	b1	c1
a2	b2	c2

q		
d	e	f
d1	e1	f1
d2	e2	f2
d3	e3	f3

### Ejemplos de enlaces

 $c = d$ 
 $c \leq d$ 
 $c \diamond d * e$ 

p		
a	b	c
a1	b1	c1
a2	b2	c2

p		
a	b	c
a1	b1	c1
a2	b2	c2

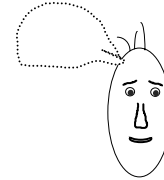
### Ejemplos de enlaces

 $c = b$ 
 $c \leq b$ 
 $c \diamond a * b$ 

### Regla

Si en la cláusula *FROM* se especifican dos o más tablas, en general, **será necesario describir enlaces** en la cláusula *WHERE* de modo que toda tabla esté enlazada.

## Perspectiva del usuario



### Intención de uso del enlace

Describir genéricamente relaciones entre los individuos representados por tablas a través de la especificación de relaciones de comparación entre columnas.

persona	
nombr	ciud
García, J.	Madrid
Rodríguez, T.	Zaragoza

empresa	
nombr	ciud
Laminados	Madrid
Cartonajes	Madrid
El Ebro	Zaragoza

### Consulta

**Personas que viven en una ciudad en la que existe una empresa.**

```
SELECT  persona.nombr, persona.ciud,
        empresa.nombr, empresa.ciud
```

```
FROM    persona, empresa
```

```
WHERE   persona.ciud =
        empresa.ciud
```

devuelve como tabla *FROM* la tabla producto de las especificadas en *FROM*  
 .....  
 selecciona las filas que cumplen la condición  
 establece el **enlace entre las tablas**

---

```
SELECT
```

### Nota

Si una columna pertenece a dos o más tablas del *FROM*, **es necesario distinguir** la tabla sobre la que se especifica **la columna anteponiendo**, al nombre de la columna, **el nombre de la tabla** separado por un punto.

nombr	ciud	nombr	ciud
García, J.	Madrid	Laminados	Madrid
García, J.	Madrid	Cartonajes	Madrid
Rodríguez, T.	Zaragoza	El Ebro	Zaragoza

## Consulta

**Personas que viven en una ciudad en la que existe una empresa.**

```
SELECT  persona.*, empresa.nombr
FROM    persona, empresa
WHERE   persona.ciud = empresa.ciud
```

devuelve como tabla *FROM* la tabla producto de las especificadas en *FROM*...  
selecciona las filas que cumplen la condición  
establece el **enlace entre las tablas**  

---

*SELECT*

## Sin repetir información

nombr	ciud	nombr
García, J.	Madrid	Laminados
García, J.	Madrid	Cartonajes
Rodríguez, T.	Zaragoza	El Ebro

```
SELECT  persona.nombr persona,  
         persona.ciud ciudad,  
         empresa.nombr empresa  
FROM    persona,  
         empresa  
WHERE   persona.ciud = empresa.ciud
```

persona	ciudad	empresa
García, J.	Madrid	Laminados
García, J.	Madrid	Cartonajes
Rodríguez, T.	Zaragoza	El Ebro



## Alias de tabla

### Definición

**Un alias** de tabla **es una designación simbólica de una tabla del FROM** que sólo tiene efecto en el contexto de la sentencia *SELECT* en la que se especifica.

```
SELECT p.nombr, p.ciud, e.nombr
FROM persona p, empresa e
WHERE p.ciud = e.ciud
```

### Intención de uso del alias de tabla

- Para simplificar la tarea de distinguir columnas de distintas tablas que tienen el mismo nombre.
- Para distinguir tablas repetidas en la cláusula *FROM*.

---

persona	
nombr	ciud
García, J.	Madrid
Rodríguez, T.	Zaragoza

### Consulta

**Pares de personas que viven en la misma ciudad.**

```
SELECT p1.*, p2.nombr
FROM persona p1, persona p2
WHERE p1.ciud = p2.ciud
```

nombr	ciud	nombr

## Casos

persona		
id_pers	nombr	ciud
p12	García, J.	Madrid
p22	Rodríguez, T.	Zaragoza

\*persona con los datos de su identificación, nombre y ciudad en la que reside\*

empresa		
id_empr	nombr	ciud
e209	Laminados	Madrid
e100	Cartonajes	Madrid
e098	El Ebro	Zaragoza

\*empresa con su identificación, nombre y ciudad en la que se encuentra\*

contrato	
id_pers	id_empr
p12	e209
p22	e100

\*relación de contrato entre una persona y una empresa con los datos de las identificaciones de ambos entes\*

relación semántica	enlace
persona contratada	<b>persona.id_pers = contrato.id_pers</b>
empresa que tiene persona contratada	<b>empresa.id_empr = contrato.id_empr</b>
persona contratada y empresa que la tiene contratada	<b>persona.id_pers = contrato.id_pers AND empresa.id_empr = contrato.id_empr</b>

## Tablas ejemplo

Las siguientes tablas son parte de una base de datos que contiene información sobre personas relacionadas con una empresa. Algunas son empleados y otras accionistas. Es posible que un accionista sea, a su vez, empleado.

### empleado

```
(  
    id_pers    VARCHAR (15)    NOT NULL,  
    NSS        CHAR (10)    NOT NULL,  
    salar      NUMBER (38,2) NOT NULL  
)
```

\*empleado de la empresa, con los datos de su identificador, número de la Seguridad Social y salario\*

### accionista

```
(  
    id_pers    VARCHAR (15)    NOT NULL,  
    fech_alta  DATE             NOT NULL,  
    numer_accion INTEGER        NOT NULL  
)
```

\*accionista de la empresa, con los datos de su identificador, fecha de alta como accionista y número de acciones que posee \*

empleado		
id_pers	NSS	salar

accionista		
id_pers	fech_alta	numer_accion

## Uniones Externas

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre los empleados. Para aquellos empleados que sean a su vez accionistas, deben mostrarse su fecha de alta como tales y el número de acciones que poseen.**

```

SELECT    e.*,
          fech_alta,
          numer_accion
FROM      empleado e, accionista a
WHERE     e.id_pers = a.id_pers (+)
/

```

id_pers	NSS	salar	fecha_alta	numer_accion
JoseGarc01	123	1500	01/01/2000	500
JuCaRodr01	456	1300		

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre los accionistas. Para aquellos accionistas que sean a su vez empleados, deben mostrarse su número de la seguridad social y salario**

```

SELECT      a.*,
            NSS,
            salar
FROM        empleado e, accionista a
WHERE      e.id_pers (+) = a.id_pers
/

```

id_pers	fecha_alta	numer_accion	NSS	salar
JoseGarc01	01/01/2000	500	123	1500
FelipRomer01	01/02/2000	200		

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre todas las personas de la empresa. Para los que sean a la vez accionistas y empleados, deben mostrarse todos sus datos.**

```
SELECT      a.*,
            NSS,
            salar
FROM        empleado e, accionista a
WHERE      e.id_pers (+) = a.id_pers (+)
/
```

¡No soportado por ORACLE!

A partir de la versión ORACLE 9i, el SQL implementado en ORACLE enriquece las posibilidades de cruzar tablas, adaptándose al estándar ANSI-SQL.

```
SELECT      *  
FROM        <tabla_unión>
```

```
<tabla_unión> ::=  
    <nombre_tabla1> CROSS JOIN <nombre_tabla2>  
    | <nombre_tabla1> NATURAL JOIN <nombre_tabla2>  
    | <nombre_tabla1> JOIN <nombre_tabla2> USING (<nombre_columna>)  
    | <nombre_tabla1> JOIN <nombre_tabla2> ON ( <condición>)  
    | <nombre_tabla1> <tipo_outer> OUTER JOIN <nombre_tabla2>  
    ON ( <condición>)
```

```
<tipo_outer> ::=  
    RIGHT | LEFT | FULL
```

## Cross Join

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

**Mostrar cada empleado junto a cada accionista**

```
SELECT *
FROM empleado CROSS JOIN accionista
/
```

id_pers	NSS	salar	id_pers	fecha_alta	numer_accion
JoseGarc01	123	1500	JoseGarc01	01/01/2000	500
JuCaRodr01	456	1300	FelipRomer01	01/02/2000	200
JoseGarc01	123	1500	FelipRomer01	01/02/2000	200
JuCaRodr01	456	1300	JoseGarc01	01/01/2000	500



## Unión Natural

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	feh_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

**Mostrar toda la información de los empleados que también son accionistas**

```
SELECT *
FROM empleado NATURAL JOIN accionista
/
```

id_pers	NSS	salar	feh_alta	numer_accion
JoseGarc01	123	1500	01/01/2000	500

Comentarios

- 1) El NATURAL JOIN es un CROSS JOIN del que se seleccionan las filas basándose en la igualdad en **todas** las columnas comunes que tengan el mismo nombre en ambas tablas
- 2) Si se hace un SELECT \*, cada una de las columnas comunes aparece una vez en el resultado
- 3) Si no hay columnas comunes, entonces el NATURAL JOIN equivale a un CROSS JOIN

## Join Con Cláusula USING

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	feh_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

**Mostrar toda la información de los empleados que también son accionistas**

```
SELECT *
FROM empleado JOIN accionista USING(id_pers)
/
```

id_pers	NSS	salar	feh_alta	numer_accion
JoseGarc01	123	1500	01/01/2000	500

Comentarios

- 1) Sean c1, c2,...cn la lista de columnas que aparecen en la cláusula USING. Entonces cada una de esas columnas debe estar no cualificada y debe identificar a una columna de cada una de las tablas sobre las que se hace el join
- 2) Se seleccionan las filas basándose en la igualdad en las columnas de ambas tablas con el mismo nombre indicadas en la cláusula USING
- 3) Si se usa un SELECT \*, entonces
  - cada una de las columnas comunes aparece una vez en el resultado
  - las columnas comunes aparecen al principio (es decir, a la izquierda) en el orden especificado en la cláusula USING, luego las columnas de la primera tabla (en el orden en que aparecen en ella) y luego las de la segunda (en el orden en que aparecen en ella)

## Union Natural vs Join Con Cláusula USING

Considera las siguientes tablas:

artículo						
id_artic	nombr	peso	color	prec_compr	prec_vent	id_prov
impr01	impresora	150	rojo	90.00	140.00	Sanj01
calc01	calculadora	150	negro	32.90	55.00	CaEl01
cale01	calendario	100	blanco	2.50	6.00	Sanj01
lamp01	lámpara	550	rojo	12.60	20.00	Elec01
lamp02	lámpara	550	blanco	12.60	20.00	Elec01
lamp03	lámpara	550	azul	12.60	20.00	Elec01
lamp04	lámpara	550	verde	12.60	20.00	Elec01
pesa01	pesacartas			14.60	24.00	MePr01
pesa02	pesacartas			18.00	30.00	MePr01
boli01	bolígrafo	20	rojo	0.15	0.25	EsRe01
boli02	bolígrafo	20	azul	0.15	0.25	EsRe01
bolu01	bolígrafo lujo	20	rojo	2.60	6.50	EsRe01
bolu02	bolígrafo lujo	20	verde	2.60	6.50	EsRe01

proveedor	
id_prov	nombr
CaEl01	Catio Electronic
EsRe01	Estilográficas Reunidas
MePr01	Mecánica de Precisión
Sanj01	Sanjita
Elec01	Electrolamp

¿Cuál es el resultado de ejecutar la siguiente consulta?

```
SELECT *
FROM artículo NATURAL JOIN proveedor;
```

Compárala con el de

```
SELECT *
FROM artículo JOIN proveedor USING (id_prov);
```

## Join Basado en Condición

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información de los empleados que también son accionistas**

```

SELECT    e.*,
          fech_alta,
          numer_accion
FROM      empleado e JOIN accionista a
          ON (e.id_pers = a.id_pers)
  
```

id_pers	NSS	salar	fecha_alta	numer_accion
JoseGarc01	1234567890	1500	01/01/2000	500

Observa que esta consulta es equivalente a

```

SELECT    e.*,
          fech_alta,
          numer_accion
FROM      empleado e JOIN accionista a USING (id_pers)
  
```

Comentarios

1) Si se usa un **SELECT \***, entonces

- las columnas comunes aparece repetidas en el resultado
- en primer lugar aparecen las columnas de la primera tabla (en el orden en que aparecen en ella) y luego las de la segunda (en el orden en que aparecen en ella)

## Otro ejemplo de Join Basado en Condición

Consulta. Clasificar por peso los artículos de la tabla artículo de acuerdo con la tabla clasificacion\_peso

articulo						
id_artic	nombr	peso	color	prec_compr	prec_vent	id_prov
impr01	impresora	150	rojo	90.00	140.00	Sanj01
calc01	calculadora	150	negro	32.90	55.00	CaEl01
cale01	calendario	100	blanco	2.50	6.00	Sanj01
lamp01	lámpara	550	rojo	12.60	20.00	Elec01
lamp02	lámpara	550	blanco	12.60	20.00	Elec01
lamp03	lámpara	550	azul	12.60	20.00	Elec01
lamp04	lámpara	550	verde	12.60	20.00	Elec01
pesa01	pesacartas			14.60	24.00	MePr01
pesa02	pesacartas			18.00	30.00	MePr01
boli01	bolígrafo	20	rojo	0.15	0.25	EsRe01
boli02	bolígrafo	20	azul	0.15	0.25	EsRe01
bolu01	bolígrafo lujo	20	rojo	2.60	6.50	EsRe01
bolu02	bolígrafo lujo	20	verde	2.60	6.50	EsRe01

clasificacion_peso		
nombr_peso	peso_minim	peso_maxim
leve	0	100
ligero	101	500
medio	501	2500
pesado	2501	9999

```

SELECT    id_artic,
          nombr,
          color,
          nombr_peso
FROM      articulo JOIN clasificacion_peso
          ON (peso BETWEEN peso_minim AND peso_maxim)

```

## Unión Externa Izquierda

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre los empleados. Para aquellos empleados que sean a su vez accionistas, deben mostrarse su fecha de alta como tales y el número de acciones que poseen.**

```

SELECT      e.*,
            fecha_alta,
            numer_accion
FROM        empleado e LEFT OUTER JOIN accionista a
            ON (e.id_pers = a.id_pers)
/

```

es equivalente a

```

SELECT      e.*,
            fecha_alta,
            numer_accion
FROM        empleado e, accionista a
            e.id_pers = a.id_pers (+)
/

```

id_pers	NSS	salar	fecha_alta	numer_accion
JoseGarc01	123	1500	01/01/2000	500
JuCaRodr01	456	1300		

## Unión Externa Derecha

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre los accionistas. Para aquellos accionistas que sean a su vez empleados, deben mostrarse su número de la seguridad social y salario**

```

SELECT      a.*,
            NSS,
            salar
FROM        empleado e RIGHT OUTER JOIN accionista a
ON         (e.id_pers = a.id_pers )
/

```

Es equivalente a

```

SELECT      a.*,
            NSS,
            salar
FROM        empleado e, accionista a
WHERE      e.id_pers (+) = a.id_pers
/

```

id_pers	fecha_alta	numer_accion	NSS	salar
JoseGarc01	01/01/2000	500	123	1500
FelipRomer01	01/02/2000	200		

## Unión Externa Total

empleado		
id_pers	NSS	salar
JoseGarc01	123	1500
JuCaRodr01	456	1300

accionista		
id_pers	fecha_alta	numer_accion
JoseGarc01	01/01/2000	500
FelipRomer01	01/02/2000	200

Consulta

**Mostrar toda la información disponible sobre todas las personas de la empresa. Para los que sean a la vez accionistas y empleados, deben mostrarse todos sus datos.**

```

SELECT      *
FROM        empleado e FULL OUTER JOIN accionista a
ON         ( e.id_pers = a.id_pers)
/

```

id_pers	NSS	salar	id_pers	fecha_alta	numer_accion
JoseGarc01	123	1500	JoseGarc01	01/01/2000	500
JuCaRodr01	456	1300			
			FelipRomer01	01/02/2000	200







## 10.- Parte LMD del SQL

(lenguaje de modificación de datos)

### 1) **Introducción de filas** en una tabla previamente creada.

1a) De una fila.

```
INSERT INTO <nombre_tabla>  
VALUES ( <dato> {,<dato>})
```

Significado (para la máquina y el usuario)

Se añade como fila, en la tabla nominada, la lista de datos.

Dar de alta a un ejemplar de la monografía 'Date92a', con número de registro 10000. Costó 7900 pesetas el 21 de septiembre de 1999.

```
INSERT INTO      ejempl_monogr  
VALUES          ('10000', 'Date92a', 7900, '21-11-99')
```

1b) De una fila

```
INSERT INTO <nombre_tabla> (<columna> {, <columna>})
VALUES (<dato> {,<dato>})
```

Principio (para la máquina y el usuario)

El número de datos debe coincidir con el número de columnas especificadas y éstas deben ser columnas de la tabla nominada. No es necesario que el conjunto de columnas listado sea el total de las columnas de la tabla, pero es **obligado** que las columnas que faltan no estén declaradas *not null*.

Significado (para la máquina y el usuario)

Se añade una fila, en la tabla nominada, considerando cada dato como dato de la columna nominada listada con su mismo número de orden y tomando dato *null* en el resto de columnas (las no nominadas).

monogr					
id_monogr	autor_s	titul	edit	año_publ	volum

```
INSERT INTO monogr (autor_s, titul, edit, año_publ, id_monogr)
VALUES ('Loney, K.', 'Oracle. Manual del administrador',
'MGH', '1994', 'Lone94a')
```

monogr					
id_monogr	autor_s	titul	edit	año_publ	volum
Lone94a	Loney, K.	Oracle. Manual del administrador	MGH	1994	

1c) Del resultado de una consulta

```
INSERT INTO <nombre_tabla> [ ( <columna> {, <columna>} ) ]
<consulta>
```

Principio (para la máquina y el usuario)

En el caso en que se especifique una lista de columnas, el número de columnas de respuesta de la sentencia *select* debe coincidir con el número de las columnas especificadas y éstas deben ser columnas de la tabla nominada. No es necesario que el conjunto de columnas listado sea el total de las columnas de la tabla, pero es **obligado** que las columnas que faltan no estén declaradas *not null*.

Significado (para la máquina y el usuario)

Se añaden como filas, en la tabla nominada, las filas respuesta de la sentencia *select*.

monogr					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I

monogr_no_obra				
id_monogr	autor_s	títul	edit	año_publ
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995

```
INSERT INTO monogr (id_monogr, autor_s, títul, edit,
año_publ)
SELECT *
FROM monogr_no_obra
```

monogr					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	

## 2) Borrar filas

```
DELETE [FROM] <nombr_tabl>
WHERE <cond_fila>
```

Significado (para la máquina y el usuario)  
Suprimir las filas que cuplen la condición.

```
DELETE monogr
WHERE volum IS NOT NULL
```

## 3) Modificar filas

```
UPDATE <nombr_tabl>
SET <columna> = <expresión> {, <columna> = <expresión>}
WHERE <cond_fila>
```

Significado (para la máquina y el usuario)

En la tabla nominada, sobre las filas que cumplen la condición expresada en el *WHERE* se toman en las columnas nombradas en set como nuevos datos los nuevos valores expresados.

monogr					
id_monogr	autor_s	titul	edit	año_publ	volum
Lone94a	Loney, K.	Oracle. Manual del administrador	MGH	1994	II

```
UPDATE monogr
SET volum = NULL
WHERE id_monogr = 'Lone94a'
```

monogr					
id_monogr	autor_s	titul	edit	año_publ	volum
Lone94a	Loney, K.	Oracle. Manual del administrador	MGH	1994	

# 11.- Parte LDD del SQL

(lenguaje de definición de datos)

## 1) Crear una tabla.

1a)

```
CREATE TABLE <nombr_tabl>
(
    <nombr_column> <dominio> [NULL / NOT NULL]
    {, <nombr_column> <dominio> [NULL / NOT NULL]}
)
```

Significado (para la máquina y el usuario)

Crear una tabla con el nombre y columnas que se describen. Se crea como tabla vacía.

```
CREATE TABLE monografia
(
    id_monogr char(7) NOT NULL,
    autor_s char(50),
    titul char(100) NOT NULL,
    edit char(20) NOT NULL,
    año_publ char(4) NOT NULL,
    volum char(8)
)
```

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum

1b)

```
CREATE TABLE <nombr_tabl>
[ (
    <nombr_column> [NULL | NOT NULL]
    {, <nombr_column> [NULL | NOT NULL] }
)
]
AS <sentencia_SELECT>
```

Significado (para la máquina y el usuario)

Crear una tabla con nombre el especificado y con filas las que se obtienen a través de la sentencia <sentencia\_SELECT>.

- Si no se ha descrito el listado de nombres de columna, los nombres de la tabla que se crea son los de salida de la sentencia <sentencia\_SELECT>.
- En caso contrario, los nombres de la tabla que se crea son los descritos en la lista.



monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

```

CREATE TABLE monografia_sin_obra
    (ident, autor_s, títul, edit, año_publ)
AS
SELECT id_monogr, autor_s, títul, edit, año_publ
FROM monografia
WHERE volum IS NULL

```

monografia_sin_obra				
ident	autor_s	títul	edit	año_publ
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995
Date90a	Date, C.J.	RDW1985-1989	AW	1990
Date92a	Date, C.J.	RDW1989-1991	AW	1992

```

CREATE TABLE monografia_con_obra
AS
SELECT *
FROM monografia
WHERE volum IS NOT NULL

```

monografia_con_obra					
id_monogr	autor_s	títul	edit	año_publ	volum
Date93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III

## 2) Borrar una tabla

*DROP TABLE* <nombr\_tabl> [*CASCADE CONSTRAINTS*]

*DROP TABLE* **monografia**

## 3) Modificar una tabla

### 3.1) Renombrar una tabla

*RENAME* <nombr\_tabl\_antig> *TO* <nombr\_tabla\_nuev>

Significado (para la máquina y el usuario)

Considerar <nombr\_tabla\_nuev> como nuevo nombre de la tabla <nombr\_tabl\_antig>.

*RENAME* **monografia\_sin\_obra** *TO* **monografia\_no\_obra**

## 3.2) Añadir columnas

```
ALTER TABLE <nombr_tabl>
  ADD ( <nombr_column1> <domin> [NULL|NOT NULL]
        { , (<nombr_column1> <domin> [NULL|NOT NULL] }
        )
```

Significado (para la máquina y el usuario)

Si la tabla es vacía

se añade la columna especificada

Si la tabla no es vacía

**se añade** la columna especificada **SÓLO si no** está declarada **NOT NULL**, con dato **NULL** en todas las filas.

monografia_no_obra				
ident	autor_s	títul	edit	año_publ
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995
Date90a	Date, C.J.	RDW1985-1989	AW	1990
Date92a	Date, C.J.	RDW1989-1991	AW	1992

```
ALTER TABLE monografia_no_obra
  ADD (ISBN INTEGER)
```

monografia_no_obra					
ident	autor_s	títul	edit	año_publ	ISBN
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
Date90a	Date, C.J.	RDW1985-1989	AW	1990	
Date92a	Date, C.J.	RDW1989-1991	AW	1992	

### 3.3) Modificar columnas

```
ALTER TABLE <nombr_tabl>
  MODIFY ( <nombr_column1> [<domin>] [NULL|NOT NULL]
    {, <nombr_column1> [<domin>] [NULL|NOT NULL]} )
```

Significado (para la máquina y el usuario)

Sobre las columnas nombradas se considera como nueva restricción la del dominio o declaración *NULL|NOT NULL* especificada a su derecha, teniendo en cuenta las siguientes condiciones:

Se puede **disminuir la extensión** de un tipo *char* **sólo si**

- la tabla es vacía, o bien
- todos los datos de la correspondiente columna tienen una longitud menor o igual que la nueva

Se puede cambiar a *NOT NULL* **sólo si**

- la tabla es vacía, o bien
- todos los datos de la correspondiente columna **NO SON NULL**

monografia					
id_monogr	autor_s	títul	edit	año_publ	volum
DATE93a	Date, C.J.	Introducción a los Sistemas de bases de datos	AWI	1993	I
MiPi93a	De Miguel, A.; Piattini, M.	Concepción y diseño de bases de datos	R	1993	
CoAD95a	Corey, M.; Abbey, M.; Dechichio, D.	Puesta a punto de Oracle	O/MH	1995	
BaFe82a	Barr, A.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	II
DATE90a	Date, C.J.	RDW1985-1989	AW	1990	
CoFe82a	Cohen, P.R.; Feigenbaum, E.A.	The Handbook of Artificial Intelligence	AW	1982	III
DATE92a	Date, C.J.	RDW1989-1991	AW	1992	

```
ALTER TABLE monogr
  MODIFY (volum NOT NULL)
```

¡ERROR!

```
ALTER TABLE monografia
  MODIFY (autor_s char(60))
```

¡SE ACEPTA!

### 3.4) Borrar una columna c en un tabla t

```
ALTER TABLE <nombr_tabl>
  DROP ( <nombr_column>
        {, <nombr_column1> } )
```

#### Tarea: Borrar la columna fech\_pago

ejempl_DE_monogr			
numer_registr	id_monogr	prec_compr	fech_pago

```
ALTER TABLE   ejempl_DE_monogr
                DROP ( fech_pago )
```

ejempl_DE_monogr		
numer_registr	id_monogr	prec_compr

5) **Renombrar una columna** *c* en un tabla *t*, tomando *c'* como nuevo nombre de columna.

- Crear una tabla con nombre *aux* como resultado de realizar una consulta sobre la tabla *t* pidiendo todas sus columnas pero obligando a que la columna *c* se cree con el nombre *c'*.
- Borrar la tabla *t*.
- Renombrar la tabla *aux* considerando *t* como nuevo nombre.

**Tarea: Renombrar** la columna *id\_monogr* por *ident*

ejempl_DE_monogr			
numer_registr	id_monogr	prec_compr	fech_pago

```
CREATE TABLE aux
AS SELECT numer_registr,
          id_monogr ident,
          cant_pagad
FROM      ejempl_DE_monogr
```

ejempl_DE_monogr			
numer_registr	id_monogr	prec_compr	fech_pago
aux			
numer_registr	Ident	prec_compr	fech_pago

```
DROP TABLE ejempl_DE_monogr
```

aux			
numer_registr	Ident	prec_compr	fech_pago

```
RENAME aux to ejempl_DE_monogr
```

ejmpl_monogr			
numer_registr	Ident	prec_compr	fech_pago

**Otro ejemplo de tarea**

monogr					
id_monogr	autor_s	titul	edit	año_publ	volum

```
CREATE TABLE monografia_obra
AS
SELECT id_monogr, volum
FROM monografia
WHERE volum IS NOT NULL
```

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum
monografia_obra					
id_monogr	volum				

```
CREATE TABLE monografia_no_obra
AS
SELECT id_monogr
FROM monogr
WHERE volum IS NULL
```

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum
monografia_no_obra					
id_monogr	volum				
monografia_no_obra					
id_monogr					

```

CREATE TABLE aux
AS
SELECT id_monogr, autor_s, titul, edit, año_publ
FROM monografia

```

monografia					
id_monogr	autor_s	titul	edit	año_publ	volum
monografia_obra					
id_monogr	volum				
monografia_no_obra					
id_monogr					
aux					
id_monogr	autor_s	titul	edit	año_publ	

```

DROP TABLE monografia

```

monografia_obra					
id_monogr	volum				
monografia_no_obra					
id_monogr					
aux					
id_monogr	autor_s	titul	edit	año_publ	



**RENAME aux TO monografia**

monografia_obra				
id_monogr	volum			
monografia_no_obra				
id_monogr				
monografia				
id_monogr	autor_s	titul	edit	año_publ

**ALTER TABLE monografia  
ADD (ISBN INTEGER)**

monografia_obra					
id_monogr	volum				
monografia_no_obra					
id_monogr					
monogr					
id_monogr	autor_s	titul	edit	año_publ	ISBN

# Vistas

## **vista**

matriz de cadenas que tiene todas las características de una tabla nominada salvo en lo siguiente:

1. la especificación de sus columnas es la de respuesta a una consulta, a la que quizás se le han cambiado los nombres de las columnas.
2. no contiene filas salvo cuando se la invoca mediante alguna consulta.

En relación a las consultas, en general, una vista puede ser tratada como una tabla

En relación a las sentencias tipo LDD y tipo LMD tienen fuertes restricciones.

## **Creación de una vista**

```
CREATE OR REPLACE VIEW <nombr_vist> [(<nombr_column1>,.....)]  
AS <consulta_SQL>  
[WITH READ ONLY]  
[WITH CHECK OPTION]
```

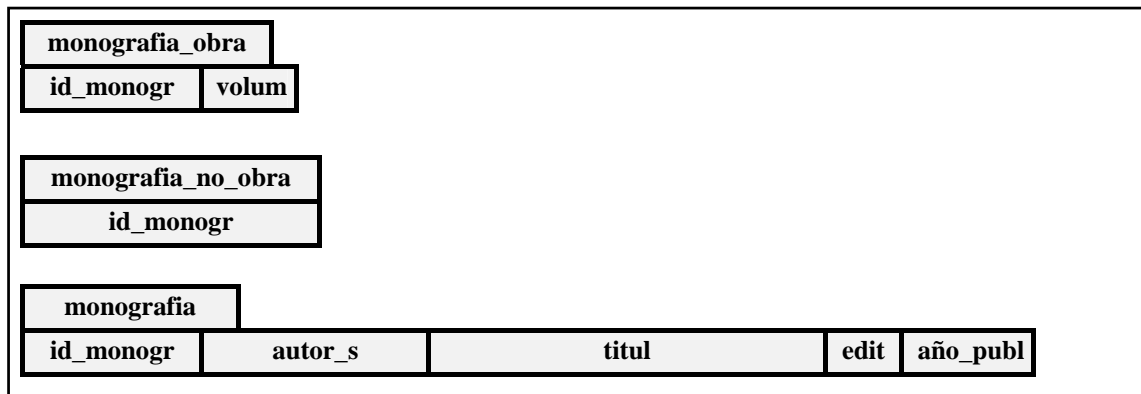
## **Principio**

El nombre de una vista no puede coincidir con el de una tabla o una vista previamente creadas.

Los nombres de columna en <consulta\_SQL> deben ser distintos dos a dos.

Es necesario definir alias para las columnas resultantes de aplicar expresiones, funciones de agregación...

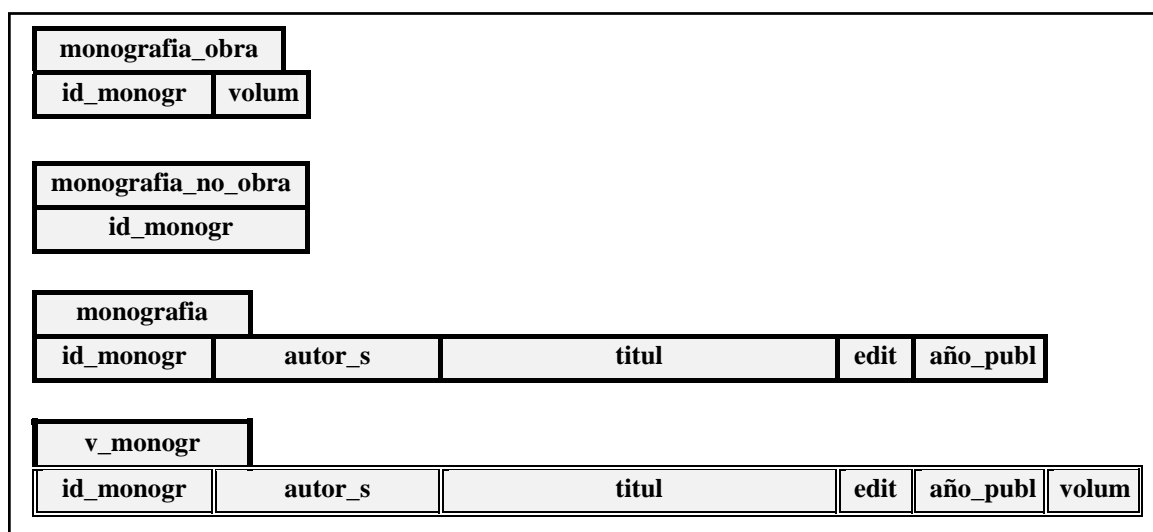
Para la creación de una vista se pueden utilizar otras vistas en la consulta que la define.



```

CREATE OR REPLACE VIEW v_monogr
AS
    SELECT m.id_monogr, autor_s, titul, edit,
           año_publ, volum
    FROM monografia m,
         monografia_obra mo
    WHERE m.id_monogr = mo.id_monogr
    UNION
    SELECT m.id_monogr, autor_s, titul, edit,
           año_publ, NULL
    FROM monogr m,
         monografia_no_obra mno
    WHERE m.id_monogr = mno.id_monogr

```



Para visualizar la especificación de una vista se puede utilizar el comando *describe*

```
describe      monografia

                monogr
                (id_monogr char(7) not null,
                autor_s char(50),
                titul char(100) not null,
                edit char(20) not null,
                año_publ char(4) not null,
                )
```

Para visualizar la sentencia *SELECT* utilizada para la creación de una vista se puede consultar la tabla *user\_views* del catálogo

```
SELECT  *
FROM   user_views
```

```
CREATE OR REPLACE VIEW  v_monogr
AS
    SELECT  m.id_monogr, autor_s, titul, edit,
           año_publ, volum
    FROM    monografia m,
           monografia_obra mo
    WHERE   m.id_monogr = mo.id_monogr
    UNION
    SELECT  m.id_monogr, autor_s, titul, edit,
           año_publ, NULL
    FROM    monografia m,
           monografia_no_obra mno
    WHERE   m.id_monogr = mno.id_monogr
```

```
describe      v_monogr

                v_monogr
                (id_monogr char(7) not null,
                autor_s char(50),
                titul char(100) not null,
                edit char(20) not null,
                año_publ char(4) not null,
                volum char(8))
```

## Semántica de las vistas ante las consultas

```
SELECT *  
FROM v_monogr
```

Cuando se ejecuta el comando

- El gestor realiza la revisión sintáctica.
  - Si no hay problema, en la tabla *tab* del catálogo, observa si **v\_monogr** es una tabla o una vista.
  - En el caso en que es una vista, busca en el catálogo la forma en cómo está definida y ejecuta la sentencia *SELECT* que la define.
  - La respuesta encontrada es tomada, por el sistema, como tabla **v\_monogr** en el contexto de la consulta que se ha planteado.
- 

## Borrado de una vista

```
DROP VIEW <nombr_vist>
```

## Descripción de una vista

```
describe <nombr_vist>
```

## Renombrar una vista

```
RENAME <nombr_vist_antig> TO <nombr_vist_nuev>
```

Una vista refleja automáticamente los cambios hechos sobre la(s) tabla(s) de que procede.

Es posible realizar operaciones LMD sobre una vista cuando:

- i) Se basa en una única tabla.
- ii) Contiene todas las columnas NOT NULL de la tabla.
- iii) Cada fila y columna de la vista corresponden exactamente a una fila y columna de la tabla.

Ejemplo:

```
CREATE VIEW v_monografia AS
SELECT id_monogr,
       titul,
       edit,
       año_publ
FROM monografia
/
```

```
INSERT INTO v_monografia
VALUES ('PRICE05', 'Oracle Database 10g SQL', 'Oracle Press', '2005')
/
```

```
CREATE OR REPLACE VIEW      v_monografia AS
SELECT      id_monogr,
            titul,
            edit
FROM        monografia
/
```

```
INSERT INTO v_monografia
VALUES ('RKD05', 'Oracle JDeveloper 10g HandBook', 'Oracle Press')
/
```

ERROR en línea 1:

ORA-01400: no se puede realizar una inserción NULL en  
("GRUPO01"."MONOGRAFIA"."AÑO\_PUBL")

```
CREATE OR REPLACE VIEW      v_monografia AS
SELECT      id_monogr,
            titul,
            edit,
            año_publ,
            COUNT (*) numer_volum_s
FROM        monografia
GROUP BY   id_monogr, titul, edit, año_publ
/
```

```
INSERT INTO v_monografia
VALUES ('RKD05', 'Oracle JDeveloper 10g HandBook', 'Oracle Press',
'2005', 2)
/
```

ERROR en línea 1:

ORA-01732: operación de manipulación de datos no válida en esta vista

Las vistas sirven para:

- facilitar consultas que se realizan con frecuencia

```
CREATE OR REPLACE VIEW      v_monogr
AS
    SELECT m.id_monogr, autor_s, titul, edit,
           año_publ, volum
    FROM   monogr m,
           monografia_obra mo
    WHERE  m.id_monogr = mo.id_monogr
    UNION
    SELECT  m.id_monogr, autor_s, titul, edit,
           año_publ, null
    FROM   monogr m,
           monografia_no_obra mno
    WHERE  m.id_monogr = mno.id_monogr
```

Son equivalentes

```
SELECT
    titul,
    edit
FROM v_monogr
```

```
SELECT
    titul,
    edit

FROM   monogr m, monografia_obra mo
WHERE  m.id_monogr = mo.id_monogr
UNION
SELECT  m.id_monogr, autor_s, titul, edit,
        año_publ, null
FROM   monogr m, monografia_no_obra
        mno
WHERE  m.id_monogr = mno.id_monogr
```



- restringir el acceso a determinada información

#### Vista para el proveedor 4

<b>tienda</b>						
<b>id_tiend</b>	<b>ciud</b>	<b>nombr_gerent</b>				
<b>cliente</b>						
<b>id_client</b>	<b>apell1</b>	<b>nombr_client</b>	<b>nacion</b>	<b>ciud</b>		
<b>articulo</b>						
<b>id_artic</b>	<b>nombr_artic</b>	<b>peso</b>	<b>color</b>	<b>prec_compr</b>	<b>prec_vent</b>	<b>id_prov</b>
<b>venta</b>						
<b>id_vent</b>	<b>id_client</b>	<b>id_tiend</b>	<b>id_artic</b>	<b>cant_artic_s</b>	<b>prec_total</b>	
<b>fech_vent</b>						

```

CREATE OR REPLACE VIEW vent_4
AS
    SELECT id_vent, id_client, id_tiend,
           id_artic, cant_artic_s, prec_total,
           fech_vent
    FROM   venta
    WHERE  id_prov = 4
  
```

En ventas, al proveedor 4 sólo se le permite acceso a la información de las ventas de sus artículos.

- dotar de cierto nivel de independencia de datos

Sin que afecte a una vista, se pueden modificar las características de los metadatos de las tablas (o vistas) que fueron utilizadas para su creación, siempre y cuando esas características no aparezcan explícitamente en la sentencia *SELECT*.

Incluso se puede cambiar la forma en como se crea la vista, sin que sea necesario cambiar su descripción.



## 12.- Restricciones de Integridad

Sobre la base de datos 'bd\_empr'...

### **Dudas**

¿Es posible que la empresa ignore el salario de alguno de sus empleados?

¿Es posible que la empresa tenga dos departamentos o proyectos con el mismo nombre?

¿Es posible que un mismo empleado esté asignado dos veces o más a un mismo proyecto?

¿Es posible que algún empleado reciba órdenes de alguna persona ajena a la empresa?

¿Es posible que alguno de nuestros empleados trabaje en un departamento que no existe, o que trabaje en un proyecto inexistente?

Una vez que un empleado se va de la empresa, ¿interesa mantener la información de sus familiares?

Cuando un empleado se va de la empresa, ¿qué ocurre con sus subordinados?

¿Es posible que un empleado cobre un salario inferior al mínimo interprofesional?

¿Es posible que un empleado cobre un salario inferior al mínimo interprofesional?

¿Qué valores son posibles en la columna 'sexo'?

¿Hasta qué grado de parentesco nos interesan los familiares de los empleados?

¿Es posible que un empleado trabaje más de 40 horas semanales?

¿Es posible que un empleado gane más que su jefe, o que el jefe de su departamento?

## *integridad de los datos*

adecuación de los datos a aquello que pretenden representar

Como ayuda, la tecnología de las bases de datos proporciona

*restricción de integridad* (perspectiva de la máquina y del usuario)

condición que deben cumplir los datos de una base de datos con la intención de que reflejen la realidad del modo más fiel posible

Declarando una restricción de integridad sobre una base de datos, el sistema se encarga de que el estado de la base de datos siempre sea conforme con su cumplimiento.

## Algunas restricciones

### **not null**

\*se declara sobre una columna\*  
no se pueden introducir datos nulos

#### Intención de uso

Evitar la ausencia de un tipo determinado de información.

#### Sintaxis:

1) La restricción puede imponerse en el momento de la creación de la tabla:

```
CREATE TABLE <tabla>  
(  
  <nombre_columna> <tipo_dato> [NOT NULL]  
  ...  
)
```

2) Posteriormente a la creación de las tablas:

```
ALTER TABLE <nombre_tabla>  
MODIFY (  
  <nombre_columna> NOT NULL  
  {, <nombre_columna> NOT NULL }  
)
```

## unique

\*se declara sobre una o varias columnas\*

**no se pueden repetir filas de datos** en las columnas referidas; es decir,

dos filas no pueden tener el mismo dato en todas y cada una de las columnas referidas

### Intención de uso

**Evitar que un tipo determinado de información se repita** en la misma tabla.

### Nota

Si en una tabla existe una declaración unique, entonces en la tabla no pueden existir filas repetidas.

### Sintaxis:

1) La restricción puede imponerse en el momento de crear la tabla ...

```
CREATE TABLE <tabla>
```

```
(
    <nombre_columna> <tipo_dato> UNIQUE
    ...
)
```

```
CREATE TABLE <tabla>
```

```
(
    <nombre_columna> <tipo_dato>,
    { <nombre_columna> <tipo_dato> , }

    [CONSTRAINT <nombre_de_restricción>]
    UNIQUE ( <nombre_de_columna> {, <nombre_de_columna>}
)
/
```

2) ... o posteriormente

```
ALTER TABLE <nombre_tabla>
```

```
ADD [CONSTRAINT <nombre_de_restricción>]
```

```
    UNIQUE ( <nombre_de_columna> {, <nombre_de_columna>} )
/
```

## primary key

\*se declara sobre una o varias columnas\*

**not null**, para cada una de las columnas referidas, y **unique** para la totalidad de éstas

### Intenciones de uso

- Determinar biunívocamente los individuos que representa la tabla.
- Que constituya la **clave de acceso** a los datos de un individuo representado.

### Principio

**En una tabla sólo** se puede declarar **una clave primaria** como máximo.

No se puede declarar simultáneamente un conjunto de columnas como **unique** y como **clave primaria**.

### Sintaxis:

#### 1) La restricción puede imponerse en el momento de crear la tabla ...

```
CREATE TABLE <tabla>
(
    <nombre_columna> <tipo_dato> PRIMARY KEY
    ...
)
```

```
CREATE TABLE <tabla>
(
    <nombre_columna> <tipo_dato>,
    { <nombre_columna> <tipo_dato> , }

    [CONSTRAINT <nombre_de_restricción>]
    PRIMARY KEY ( <nombre_de_columna> {, <nombre_de_columna>}
)
/
```

#### 2)... o posteriormente

```
ALTER TABLE <nombre_tabla>
ADD [CONSTRAINT <nombre_de_restricción>]
    PRIMARY KEY ( <nombre_de_columna> {, <nombre_de_columna>} )
/
```

## foreign key

\*es la llamada integridad referencial\*

\*se declara sobre una o varias columnas de una tabla **t1**; a la tabla **t1** se le llama **tabla referente**; al conjunto de columnas declaradas se le llama **clave foránea**; a cada una de sus columnas se le llama **columna referente**\*

\*la **clave foránea** refiere a una **clave primaria o conjunto de columnas con restricción unique** de una tabla **t2** cuyo número de columnas tiene que ser el mismo que el de las columnas referentes; a la tabla **t2** se le llama **tabla referida**; a cada una de sus columnas se le llama **columna referida**; se dice que la familia de las columnas referentes es **clave foránea de t2 en t1**\*

**para toda fila f1 en t1 debe existir una fila f2 en t2** de modo que los datos de f2 en las columnas referidas sean los mismos que los datos de f1 en las correspondientes columnas referentes

Intenciones de uso

Dada la información sobre un individuo representado, obligar a la existencia de información sobre un individuo.

Sintaxis:

1) La restricción puede imponerse en el momento de crear la tabla:

```
CREATE TABLE <tabla>
(
    <columna> <tipo_dato> REFERENCES ( <tabla>.<columna>)
    ...
)
```

```
CREATE TABLE <tabla>
(
    <nombre_columna> <tipo_dato>,
    { <nombre_columna> <tipo_dato> , }

    [CONSTRAINT <nombre_de_restricción>]
    FOREIGN KEY ( <nombre_de_columna> {, <nombre_de_columna>})
    REFERENCES <tabla> ( <columna> {, <columna>})
    [ON DELETE CASCADE| ON DELETE SET NULL]
    [DEFERRABLE INITIALLY DEFERRED|NOT DEFERRED ]
```

/



2)... o posteriormente

```
ALTER TABLE <nombre_tabla>
ADD
[CONSTRAINT <nombre_de_restricción>]
FOREIGN KEY ( <nombre_de_columna> {, <nombre_de_columna>})
REFERENCES <tabla> ( <columna> {, <columna>})
[ON DELETE CASCADE| ON DELETE SET NULL]
[DEFERRABLE INITIALLY DEFERRED|NOT DEFERRED ]
/
```

### Principio

Si las columnas referidas tienen nombre distinto a las referentes, es obligado señalarlas en la declaración.

## check

\*declara una **condición** sobre una o varias columnas de una tabla\*  
obliga al cumplimiento de la condición expresada

### Intenciones de uso

Establecer una condición de obligado cumplimiento con objeto de potenciar la coherencia con la realidad de la información almacenada  
La condición afecta a una o varias columnas de una misma fila.  
No es posible definir una restricción de tipo check que involucre a varias filas distintas.

### Sintaxis:

#### 1) La restricción puede imponerse en el momento de crear la tabla ...

```
CREATE TABLE <tabla>
(
    <nombre_columna> <tipo_dato> CHECK ( <condición>)
    ...
)
```

```
CREATE TABLE <tabla>
(
    <nombre_columna> <tipo_dato>,
    { <nombre_columna> <tipo_dato> , }
    [CONSTRAINT <nombre_de_restricción>]
    CHECK ( <condición>)
)
/
```

#### 2) ... o posteriormente

```
ALTER TABLE <nombre_tabla>
ADD [CONSTRAINT <nombre_de_restricción>]
    CHECK ( <condición>)
/
```

Activar, desactivar y borrar una restricción:

```
ALTER TABLE <nombre_tabla>  
ENABLE <nombre_restricción>  
/
```

```
ALTER TABLE <nombre_tabla>  
DISABLE <nombre_restricción>  
/
```

```
ALTER TABLE <nombre_tabla>  
DROP CONSTRAINT <nombre_restricción>  
/
```



# 13.- Privilegios

Una de las tareas del administrador de una base de datos es garantizar la seguridad de la información almacenada. Ello requiere, entre otras cosas, planificar una política de gestión de privilegios: a cada usuario conviene darle el **mínimo privilegio** estrictamente necesario para realizar su labor.

La sintaxis de asignación de privilegios es:

*GRANT* <privilegio> *TO* <usuario>;

Es posible también retirarle privilegios a un usuario:

*REVOKE* <privilegio> *FROM* <usuario>;

Algunos de los privilegios que permite gestionar ORACLE son:

*CREATE SESSION* permite al usuario conectarse a su cuenta.

*CREATE SYNONYM* permite al usuario crear sinónimos.

*SELECT ON* <nombre\_tabla> permite al usuario consultar una tabla de otro usuario.

*INSERT ON* <nombre\_tabla> permite al usuario hacer inserciones en una tabla.

*UPDATE [ ( <columna> {, <columna>} )] ON* <nombre\_tabla> permite al usuario modificar valores de todas o algunas columnas de una tabla.

Planteamos el siguiente problema: la base de datos `bd_comerc` almacena toda la información relativa a una cadena de tiendas. Entre otras, tiene la tabla 'venta', donde se registran todas las ventas que hace cada tienda a cada cliente.

venta					
id_client	id_tiend	id_artic	cant_artic_s	prec_total	fech
AnLI01	RaBa01	lamp01	1	20.00	06/06/2001
PaGo01	EsPa01	boli01	1	0.25	06/06/2001
PaGo01	EsPa01	boli02	2	0.50	06/06/2001
PaGo01	EsPa01	bolu03	3	18.75	08/06/2001
GeCo01	LMLy01	calc01	1	55.00	09/06/2001
MaSo01	VePa01	cale01	2	12.00	09/06/2001
DiCo01	BrMa01	cale01	1	6.00	10/02/2001
MaBo01	GVMa01	calc01	1	55.00	10/02/2001
MaBo01	GVMa01	bolu01	1	6.50	10/02/2001
MaBo01	GVMa01	bolu02	10	60.00	10/03/2001
MiDu01	LMLy01	impr01	8	1100.00	11/03/2001

Consideraremos los siguientes usuarios:

- *DBA*: administrador de la base de datos.
- *Gestor*: responsable de la gestión de las tiendas. Es el propietario de todas las tablas de la base de datos.
- *GVMa01\_gerente*: es el gerente de la tienda GVMa01. Tiene control total de la información relativa a su tienda: puede insertar filas, borrar filas y modificar datos, pero sólo relativos a ventas efectuadas en su tienda.
- *GVMa01\_dependiente*: Puede hacer consultas sobre las ventas de su tienda, y corregir errores relativos a fechas de venta, cantidad de artículos y precio pagado. No puede insertar ni borrar filas.
- *auditor*: es un auditor externo, que puede hacer todas las consultas relativas a la tienda, pero no puede alterar ninguno de los datos.

El Gestor empieza por crear una vista:

```
CREATE OR REPLACE VIEW venta_GVMa01
AS SELECT *
   FROM   venta
   WHERE  id_tienda = 'GVMa01'
WITH CHECK OPTION
/
```

La cláusula *WITH CHECK OPTION* garantiza que sobre la vista no se pueden realizar inserciones ni modificaciones que no cumplan la condición de la cláusula *WHERE*.

Auditor:

El **DBA** crea una cuenta para él y le otorga los privilegios que necesita para su trabajo:

```
CREATE USER auditor IDENTIFIED BY auditor
/
GRANT CREATE SESSION TO auditor
/
GRANT CREATE SYNONYM TO auditor
/
```

El **gestor** le permite consultar los datos de la tienda GVMa01:

```
GRANT SELECT ON venta_GVMa01 TO auditor
/
```

El **auditor** define un sinónimo para referirse a la vista gestor.venta\_GVMa01:

```
CREATE SYNONYM venta FOR gestor.venta_GVMa01
/
```

Dependiente:

El **DBA** crea una cuenta para él y le otorga los privilegios que necesita para su trabajo:

```
CREATE USER dependiente IDENTIFIED BY dependiente  
/  
GRANT CREATE SESSION TO dependiente  
/  
GRANT CREATE SYNONYM TO dependiente  
/
```

El **gestor** le permite consultar los datos de la tienda GVMa01, y hacer modificaciones en algunas de sus columnas:

```
GRANT SELECT, UPDATE (cant_artic_s, prec_total, fech) ON  
venta_GVMa01 TO dependiente  
/
```

El **dependiente** define un sinónimo para referirse a la vista gestor.venta\_GVMa01:

```
CREATE SYNONYM venta FOR gestor. venta_GVMa01  
/
```



Gerente:

El **DBA** crea una cuenta para él y le otorga los privilegios que necesita para su trabajo:

```
CREATE USER gerente IDENTIFIED BY gerente  
/  
GRANT CREATE SESSION TO gerente  
/  
GRANT CREATE SYNONYM TO gerente  
/
```

El **gestor** le da control total sobre los datos de la tienda GVMa01:

```
GRANT SELECT, UPDATE, INSERT, DELTE ON venta_GVMa01  
TO gerente  
/
```

El **gerente** define un sinónimo para referirse a la vista gestor.venta\_GVMa01:

```
CREATE SYNONYM venta FOR gestor. venta_GVMa01  
/
```





**ACCESO A BASES DE  
DATOS**

**APÉNDICES**



# Apéndice A: Resumen de la sintaxis SQL

## Símbolos utilizados

**::=** se define como  
| o  
**<>** nombre de categoría  
**[ ]** opcional (0 ó 1)  
**{ }** repetir 0 ó más veces

## Expresiones

**<expresión> ::=**  
| <expresión\_constante>  
| <expresión\_de\_columna>  
| <expresión\_numérica>  
| <expresión\_de\_carácter>  
| <expresión\_funcional\_escalar>  
| <expresión\_funcional\_de\_grupo>  
| <expresión\_CASE>  
| (<consulta\_anidada>)  
| (<expresión>)

**<expresión\_constante> ::=**  
NULL | '<cadena>' | <número>

```
SELECT *  
FROM articulo  
WHERE peso IS NULL ;
```

```
SELECT *  
FROM articulo  
WHERE nombr = 'bolígrafo' ;
```

```
SELECT *  
FROM articulo  
WHERE peso > 100 ;
```

**<expresión\_de\_columna> ::=**  
[<nombre\_tabla>.|<alias\_tabla>.]<nombre\_columna>

```
SELECT articulo.nombr  
FROM articulo;
```

```
SELECT a.nombr  
FROM articulo a;
```

```
SELECT nombr  
FROM articulo;
```

**<expresión\_numérica>**

```
SELECT prec_vent - prec_compr
FROM articulo;
```

```
SELECT (prec_vent-prec_compr)/prec_compr
FROM articulo;
```

```
SELECT (prec_vent - prec_compr)*166.386
FROM articulo;
```

**<expresión\_carácter>**

```
SELECT apell_s || ', ' || nombr
FROM cliente
```

**<expresión\_funcional\_escalár > ::=**

```
<símbolo_función> ( <expresión> {,<expresión>} )
```

```
SELECT SQRT (prec_vent)
FROM articulo;
```

```
SELECT INITCAP (nombr)
FROM articulo;
```

**<expresión\_funcional\_de\_grupo > ::=****<función\_de\_agregación>**

```
( [DISTINCT] <expresión_constante>
```

```
| [DISTINCT] <expresión_columna>
```

```
| [DISTINCT] <expresión_numérica>
```

```
| [DISTINCT] <expresión_carácter>
```

```
| [DISTINCT] <expresión_funcional_escalár>
```

```
)
```

```
| COUNT (*)
```

```
SELECT MAX (prec_vent)
FROM articulo;
```

```
SELECT SUM (prec_total)
FROM venta;
```

```
SELECT COUNT (DISTINCT id_client)
FROM venta;
```

**<función\_de\_agregación> ::=**

```
COUNT | MAX | MIN | SUM | AVG
```

```
<expresión_CASE> ::=
CASE <expresión>
  WHEN <expresión_1> THEN <expresión_1>
  {WHEN <expresión_i> THEN <expresión_i>}
  ELSE <expresión_else>
END
|
CASE
  WHEN <condición_1> THEN <expresión_1>
  {WHEN <condición_i> THEN <expresión_i>}
  ELSE <expresión_else>
END
```

```
SELECT      ciud,
           CASE ciud
             WHEN 'París' THEN 'Francia'
             WHEN 'Lyon' THEN 'Francia'
             ELSE 'España'
           END país
FROM tienda
/
```

```
SELECT      id_artic,
           peso,
           CASE
             WHEN (peso BETWEEN 0 AND 100) THEN 'leve'
             WHEN (peso BETWEEN 101 AND 500) THEN 'ligero'
             WHEN (peso BETWEEN 501 AND 2500) THEN 'medio'
             WHEN (peso > 2500) THEN 'pesado'
             ELSE 'indefinido'
           END "Tipo de peso"
FROM articulo
/
```

## Condiciones

```
<condición_fila> ::=
    <condición>
```

```
<condición_grupo> ::=
    <condición>
```

```
<condición> ::=
    <condición_simple>
    | <condición_combinada>
    | (<condición>)
```

```
<condición_simple> ::=
    <_valor_nulo>
    | <_rango>
    | <_semejante>
    | <_comparación>
    | <_miembro>
    | <_existencia>
```

```
<condición_simple_valor_nulo > ::=
    <expresión> IS [NOT] NULL
SELECT *
FROM articulo
WHERE peso IS NULL ;
```

```
<condición_simple_rango> ::=
    <expresión> BETWEEN <expresión> AND <expresión>
SELECT *
FROM articulo
WHERE peso BETWEEN 100 AND 150 ;
```

```
<condición_simple_semejante> ::=
    <expresión> [NOT] LIKE '<cadena>'
SELECT *
FROM cliente
WHERE apell_s LIKE 'Borrás' ;

SELECT *
FROM cliente
WHERE apell_s LIKE 'D%' ;

SELECT *
FROM cliente
WHERE apell_s LIKE 'D_____ ' ;
```



**<condición\_simple\_comparación> ::=**

```

  <expresión> <operador_comparación> <expresión>
| <expresión> <operador_comparación> <cuantificador> (<expresión> {, <expresión>} )
| <expresión> <operador_comparación> <cuantificador> <consulta>

```

```

SELECT *
FROM articulo
WHERE prec_compr > 12 ;

```

```

SELECT *
FROM articulo
WHERE prec_compr < > ALL (12.6, 18);

```

```

SELECT *
FROM articulo
WHERE peso = ( SELECT peso
                FROM articulo
                WHERE id_artic = 'boli01');

```

**<operador\_comparación> ::=**

```

= | <> | != | < | <= | > | >=

```

**<cuantificador> ::=**

```

ALL | SOME | ANY

```

**<condición\_simple\_miembro> ::=**

```

  <expresión> [NOT] IN (<expresión> {, <expresión>})
| <expresión> [NOT] IN <consulta>

```

```

SELECT *
FROM cliente
WHERE ciud IN ('Madrid', 'Barcelona');

```

```

SELECT *
FROM venta
WHERE id_artic IN (
    SELECT id_artic
    FROM articulo
    WHERE id_prov = 'Sanj01'
);

```

**<condición\_de\_existencia> ::=**  
 [NOT] EXISTS <consulta>

```

SELECT id_artic, nombr
FROM articulo a
WHERE NOT EXISTS (SELECT id_tiend
                  FROM venta
                  WHERE id_client = 'PaGo01'
                  MINUS
                  SELECT id_tiend
                  FROM venta
                  WHERE id_artic = a.id_artic
                  );

```

```
<condición_combinada> ::=  
  [NOT] <condición>  
  [<operador_lógico> [NOT] <condición>]
```

```
SELECT *  
FROM articulo  
WHERE ( color = 'rojo'  
        AND peso > 100)  
        OR color = 'verde';  
  
SELECT *  
FROM articulo  
WHERE NOT (color = 'rojo'  
           AND peso > 100)  
;
```

```
<operador_lógico> ::=  
  AND | OR
```

## Consulta de datos (LC)

```
<consulta_sql> ::=
  <consulta_no_conjuntista>
  | <consulta_conjuntista>
```

```
<consulta_no_conjuntista> ::=
  SELECT [ALL|DISTINCT] <lista_select>
  FROM <tabla_from>
  [WHERE <condición_fila>]
  [GROUP BY
    <expresión_columna> {,<expresión_columna>}
    [HAVING <condición_grupo>]]
  ]
  [ORDER BY [<número_de_columna>|<alias_de_columna>|<expresión_de_columna>][ASC|DESC]
    {, [<número_de_columna>|<alias_de_columna>|<expresión_de_columna>][ASC|DESC]}
  ]
```

```
SELECT p.id_prov identificador,
       p.nombr Proveedor,
       count(*) "Artículos"
FROM   proveedor p,
       articulo a
WHERE  p.id_prov = a.id_prov
GROUP BY p.id_prov, p.nombr
HAVING AVG (prec_compr) > 12
ORDER BY count(*) ASC;
```

```
<lista_select> ::=
  *
  | <expresión> [alias] {, <expresión> [alias]} )
```

```
SELECT *
FROM articulo;
```

```
SELECT p.id_prov identificador,
       p.nombr Proveedor
FROM articulo;
```

```
SELECT nombr||' '||color,
       prec_vent-prec_compr
FROM articulo;
```

```
<tabla_from> ::=
  <nombre_tabla> [<alias_tabla>] {, <nombre_tabla> [<alias_tabla>] }
  | <tabla_unión>
```

```
SELECT *
FROM articulo a,
     proveedor p
WHERE a.id_prov = p.id_prov;
```

```

<tabla_unión> ::=
    <nombre_tabla1> CROSS JOIN <nombre_tabla2>
  | <nombre_tabla1> NATURAL JOIN <nombre_tabla2>
  | <nombre_tabla1> JOIN <nombre_tabla2> USING (<nombre_columna>)
  | <nombre_tabla1> JOIN <nombre_tabla2> ON (<condición>)
  | <nombre_tabla1> <tipo_outer> OUTER JOIN <nombre_tabla2> ON (<condición>)

```

```

SELECT *
FROM articulo NATURAL JOIN proveedor;

SELECT *
FROM articulo JOIN proveedor ON id_prov;

SELECT *
FROM articulo LEFT OUTER JOIN proveedor ON id_prov;

```

```

<tipo_outer> ::=
RIGHT | LEFT | FULL

```

```

<consulta_conjuntista> ::=
    <consulta_no_conjuntista_sin_order_by>
  UNION | INTERSECT | MINUS
  <consulta_sin_order_by>
  { UNION | INTERSECT | MINUS
    <consulta_no_conjuntista_sin_order_by>
  }
  [ORDER BY
    [<número_de_columna>|<alias_de_columna>|<expresión_de_columna>][ASC|DESC]
    {, [<número_de_columna>|<alias_de_columna>|<expresión_de_columna>][ASC|DESC] }
  ]

```

```

SELECT apell_s||', '|nombr Nombre,
       ciud Ciudad,
       'Cliente' Tipo
FROM cliente
UNION
SELECT nombr_gerent,
       ciud,
       'Gerente'
FROM tienda
ORDER BY ciudad ASC, 1 ASC ;

```

## Modificación de datos (LMD)

### Inserción de filas

```
INSERT INTO <nombre_tabla> [ ( <nombre_columna> {, <nombre_columna>} ) ]  
VALUES (<dato> {, <dato>} )  
|<consulta_SQL>
```

```
INSERT INTO cliente  
VALUES ('JuRo01', 'Rodríguez', 'Juan', 'ESP', NULL);
```

```
INSERT INTO cliente (id_client, nombr, apell_s)  
VALUES ('JuRo02', 'Julio', 'Romero');
```

```
INSERT INTO cliente_historico  
SELECT * FROM cliente;
```

### Borrado de filas

```
DELETE [FROM] <nombre_tabla>  
[WHERE <condición_de_fila>]
```

```
DELETE cliente;
```

```
DELETE cliente  
WHERE ciud = 'Madrid';
```

### Modificación de datos

```
UPDATE <nombre_tabla>  
SET <nombre_columna> = <expresión_escalar>  
{, <nombre_columna> = <expresión_escalar>}  
[WHERE <condición_de_fila>]
```

```
UPDATE articulo  
SET prec_compr= prec_compr * 1.05,  
prec_vent= prec_vent * 1.07;
```

```
UPDATE articulo  
SET prec_vent= prec_vent * 1.07  
WHERE prec_vent > 18;
```

## Definición de datos (LDD)

### Creación de tabla

```
CREATE TABLE <nombre_tabla>
(
    <nombre_columna> <tipo_dato> [NULL|NOT NULL] [<restricción_columna>]
    {,<nombre_columna> <tipo_dato> [NULL|NOT NULL] [<restricción_columna>]}
    [ <restricción_tabla> {, >restricción_tabla}> ]
)
```

```
CREATE TABLE cliente_historico
(
    id_client    VARCHAR2(10)    NOT NULL CONSTRAINT Pkclient PRIMARY KEY,
    apell_s     VARCHAR2(25)    NOT NULL,
    nombr       VARCHAR2(20)    NOT NULL,
    sexo        CHAR,
    nacion      VARCHAR2(3),
    ciud        VARCHAR2(20),
    CONSTRAINT Csexo CHECK(sexo IN ('V','M'))
);
```

```
CREATE TABLE <nombre_tabla> [ ( <nombre_columna> [NULL | NOT NULL]
                                {, <nombre_columna> [NULL | NOT NULL]} ) ]
AS <consulta_SQL>
CREATE TABLE cliente_historico
AS SELECT * FROM cliente;
```

### Renombrar una tabla

```
RENAME <nombre_tabla> TO <nombre_tabla>
RENAME cliente_historico TO client_hist;
```

### Borrar una tabla

```
DROP TABLE <nombre_tabla> [CASCADE CONSTRAINTS]
DROP TABLE client_hist;
```

## Modificación de tabla (de su definición)

```

ALTER TABLE <nombre_tabla>
  ADD ( <nombre_columna> <tipo_dato> [NULL | NOT NULL]
        {, <nombre_columna> <tipo_dato> [NULL | NOT NULL]}
        )
  | ADD (<restricción> {, <restricción>} )
  | MODIFY ({<nombre_columna> [<tipo_dato>] [NULL | NOT NULL],...})
  | DROP CONSTRAINT <restricción_tabla>

```

```

ALTER TABLE cliente
ADD (tarj_socio VARCHAR2 (10));

```

```

ALTER TABLE cliente
MODIFY (tarj_socio VARCHAR2 (14) NOT NULL);

```

## Creación de vista

```

CREATE [OR REPLACE] VIEW
<nombre_vista> [( {<nombre_columna>, ... } )]
AS
<consulta_SQL>
[WITH READ ONLY]
[WITH CHECK OPTION]

```

```

CREATE OR REPLACE VIEW
v_cliente_Madrid
AS SELECT *
      FROM cliente
      WHERE ciud='Madrid';

```

```

CREATE OR REPLACE VIEW
v_cliente_Madrid (id_client, nombr)
AS SELECT id_client, apell_s ||
      ', ' || nombr
      FROM cliente
      WHERE ciud='Madrid';

```

## Borrado de vista

```

DROP VIEW <nombre_vista>

```

```

DROP VIEW v_cliente_Madrid ;

```

## Restricciones de integridad

```
<restricción > ::=
    <restricción_de_columna >
  | <restricción_de_tabla
```

```
<restricción_de_columna > ::=
    [CONSTRAINT <nombre_restricción>]
      NOT NULL
    | UNIQUE
    | PRIMARY KEY
    | REFERENCES <nombre_tabla> [( <nombre_columna> {, <nombre_columna>} )]
      [ON DELETE CASCADE|ON DELETE SET NULL]
      [DEFERRABLE INITIALLY [NOT] DEFERRED]
    | CHECK (<condición> )
```

```
CREATE TABLE persona
(
  id_pers VARCHAR2(10) NOT NULL
)
/
```

```
CREATE TABLE persona
(
  id_pers      VARCHAR2(10)      CONSTRAINT PKpersona PRIMARY KEY,
  salar        NUMBER (20,2)    CONSTRAINT Csalar_minimo CHECK(salar > 600),
  id_depart    VARCHAR2 (30)    CONSTRAINT Fkpers_id_depart
                                REFERENCES departamento (id_depart)
                                ON DELETE CASCADE
)
/
```



```

<restricción_de_tabla=
  [CONSTRAINT <nombre_restricción>]
  NOT NULL
  | UNIQUE ( <nombre_columna> {, <nombre_columna>} )
  | PRIMARY KEY ( <nombre_columna> {, <nombre_columna>} )
  | FOREIGN KEY ( <nombre_columna> {, <nombre_columna>} )
  REFERENCES <nombre_tabla> [( <nombre_columna> {, <nombre_columna>} )]
  [ON DELETE CASCADE|SET NULL]
  [DEFERRABLE INITIALLY [NOT] DEFERRED]
  | CHECK (<condición> )

```

```

CREATE TABLE persona
(
  id_pers    VARCHAR2(10),
  salar      NUMBER (20,2),
  id_depart  VARCHAR2 (30),
  CONSTRAINT PKpersona PRIMARY KEY(id_pers),
  CONSTRAINT Csalar_minimo
  CHECK(salar > 600),
  CONSTRAINT Fkpers_id_depart
  FOREIGN KEY (id_depart)
  REFERENCES departamento (id_depart)
  ON DELETE CASCADE
)
/

```

```

ALTER TABLE persona
ADD CONSTRAINT Fkpers_id_depart
  FOREIGN KEY (id_depart)
  REFERENCES departamento (id_depart)
  ON DELETE CASCADE
  DEFERRABLE INITIALLY DEFERRED
/

```

#### Activar una restricción

```

ALTER TABLE    <nombre_tabla>
ENABLE        <nombre_restricción>
/

```

#### Deshabilitar una restricción

```

ALTER TABLE    <nombre_tabla>
DISABLE       <nombre_restricción>
/

```

#### Borrar una restricción

```

ALTER TABLE    <nombre_tabla>
DROP CONSTRAINT    <nombre_restricción>
/

```



# Apéndice B: Referencia ORACLE

## Funciones

### Tratamiento de nulos

**NVL** ( <expresión1>, <expresión2>)

Equivale a:

```
IF (<expresión1> IS NULL) THEN
  <expresión2>
ELSE
  <expresión1>
```

```
SELECT      id_artic,
            nombr,
            NVL(color, 'indefinido')
FROM articulo;
```

**NVL2** ( <expresión1>, <expresión2>, <expresión3>)

Equivale a:

```
IF (<expresión1> IS NULL) THEN
  <expresión3>
ELSE
  <expresión2>
```

```
SELECT      id_artic,
            nombr,
            NVL2(color, 'definido', 'indefinido')
FROM articulo;
```

## Tratamiento de cadenas de caracteres

### **INITCAP (<cadena>)**

Pasa a mayúsculas las iniciales de todas las palabras de cadena. Todos los demás caracteres pasan a minúsculas

```
SELECT      INITCAP (nombr)
FROM        articulo;
```

### **LENGTH ( <cadena> )**

Devuelve la longitud en caracteres de cadena.

```
SELECT      LENGTH (nombr)
FROM        empleado;
```

### **LOWER (<cadena>)**

Devuelve la cadena formada al pasar a minúsculas todos los caracteres de texto.

### **LPAD (<cadena>, <longitud>, [<carácter>{<carácter>}] )**

Construye una cadena tomando longitud caracteres de cadena. En su caso, rellena a la izquierda con los caracteres especificados entre [] hasta llegar a longitud.

```
SELECT      LPAD (nombr, 20, '.')
FROM        articulo;
```

### **LTRIM ( <cadena>, '<carácter>{<carácter>}' )**

Elimina todos los caracteres especificados del inicio de la cadena.

```
SELECT      LTRIM (nombr, 'lbo')
FROM        articulo;
```

### **RPAD (<cadena>, <longitud>, [<carácter>{<carácter>}] )**

Construye una cadena tomando longitud caracteres de cadena. En su caso, rellena a la derecha con los caracteres especificados entre [] hasta llegar a longitud.

```
SELECT      RPAD (nombr, 20, '.')
FROM        articulo;
```

### **RTRIM ( <cadena>, '<carácter>{<carácter>}' )**

Elimina todos los caracteres especificados del fin) de <cadena>.

```
SELECT      RTRIM (nombr, 'lbo')
```

```
FROM      articulo;
```

### **SUBSTR ( <cadena>, <entero1>, <entero2> )**

Devuelve <entero2> caracteres de cadena, empezando en <entero1>

```
SELECT    SUBSTR (id_empl, 7,3)
FROM      empleado;
```

### **TRANSLATE ( <cadena>, <cadena\_vieja>, <cadena\_nueva> )**

Devuelve la cadena formada al sustituir en texto cada uno de los caracteres de <cadena\_vieja> por su correspondiente en <cadena\_nueva>.

```
SELECT    TRANSLATE ( apell_s, 'áéíóú', 'aeiou')
FROM      empleado;
```

### **UPPER ( <cadena> )**

Devuelve la cadena formada al pasar a mayúsculas todos los caracteres de texto.

## Tratamiento de cadenas de fechas

### **TO\_CHAR** ( <fecha>, <formato>)

La función TO\_CHAR transforma una fecha en una cadena de caracteres con el formato especificado.

Algunos posibles formatos son:

**MM**: número de mes.

**MONTH**: nombre del mes (9 caracteres)

**DD**: día del mes

**YY**: año (2 dígitos)

**YYYY**: año (4 dígitos)

**HH**: hora del día (de 1 a 12)

**HH24**: hora del día (de 0 a 23)

**MI**: minutos

**SS**: segundos

```
SELECT    id_monogr,
          TO_CHAR (fech_pago, 'DD-MONTH-YYYY')
FROM      ejempl_DE_monogr;
```

```
SELECT    TO_CHAR (fech_pago, 'MONTH') "Mes",
          COUNT (*) "Ejemplares comprados"
FROM      ejempl_DE_monogr
GROUP BY  TO_CHAR (fech_pago, 'MONTH');
```

```
SELECT    TO_CHAR (fech_pago, 'MONTH') "Mes",
          COUNT (*) "Ejemplares comprados"
FROM      ejempl_DE_monogr
GROUP BY  TO_CHAR (fech_pago, 'MONTH'),
          TO_CHAR (fech_pago, 'MM')
ORDER BY  TO_CHAR (fech_pago, 'MM');
```

### **TO\_DATE** ( <cadena\_caracteres>, <formato>)

La función TO\_DATE transforma una cadena de caracteres, que representa una fecha en el formato especificado, en una fecha.

```
INSERT INTO ejempl_DE_monogr
VALUES ( 00179, DATE93a, 100 , TO_DATE ('01-01-2005', 'DD-MM-YYYY') );
```

## Comandos

```
ALTER SESSION SET <nombre_parámetro> = <valor_parámetro>  
                    {<nombre_parámetro> = <valor_parámetro>}
```

El comando ALTER SESSION permite determinar los parámetros de sesión (idioma, formato estándar de fechas...) para la sesión en que se ejecuta. Su efecto se mantiene mientras dura la sesión.

```
ALTER SESSION SET NLS_LANGUAGE='SPANISH'
```

```
ALTER SESSION SET NLS_LANGUAGE='ENGLISH' NLS_DATE_FORMAT='DD-MM-yyyy'
```





# Bibliografía

## Teoría: El Modelo Relacional

- [Cod90] **E.F.Codd**, *The Relational Model for Database Management*, Addison-Wesley 1990
- [Dat01] **C.J.Date**, *Introducción a los Sistemas de Bases de Datos (7ª Edición)*, Pearson 2001

## Teoría General sobre Bases de Datos

- [Con04] **Th. Connolly, C. Begg**, *Database Systems (4ª edición)*, Addison-Wesley 2004
- [Elm02] **Elmasri, R.; Navathe, S.B.** *Sistemas de Bases de Datos: Conceptos fundamentales (3ª edición)*, Addison-Wesley, 2002.

## SQL

- [Cel97] **J. Celko**, *SQL Puzzles and Answers*, Morgan-Kaufmann 2004
- [Gen04] **J. Gennik**, *Oracle SQL \*Plus: The Definitive Guide, (2ª edición)*, O'Reilly 2004
- [Lon02] **Loney K., Koch G.** *Oracle 9i: The Complete Reference*, McGraw-Hill, 2002.
- [Mar92] **Marée C., Ledant G., L.**, *SQL: Iniciación, programación y prácticas avanzadas*, Masson 1992.
- [Mis04] **S. Mishra, A. Beaulieu**, *Mastering Oracle SQL (2ª edición)* O'Reilly 2004
- [Pri05] **J.Price**, *Oracle Database 10g SQL* Oracle Press McGraw-Hill, 2005

