

►►► Structured Query Lenguaje

Introducción

Conceptos
de BBDD
Relacionales

Restricciones
de integridad

Conversión
del modelo
Conceptual

Tipos de
relaciones

Ejercicios



01

Introducción

Introducción

- Structured Query Language (SQL)
- Lenguaje **declarativo** de acceso a bases de datos que combina construcciones del álgebra relacional y el cálculo relacional.
- Originalmente desarrollado en los '70 por IBM en su Research Laboratory de San José a partir del cálculo de predicados creado por Codd.
- Lenguaje estándar *de facto* en los SGBD comerciales
- Estándares:

SEQUEL(Structured English QUery Language), IBM 1976

SQL-86 (ANSI SQL)

SQL-89 (SQL1)

SQL-92 (SQL2), gran revisión del estándar

SQL:1999 (SQL3), Añade disparadores, algo de OO, ...

SQL:2003. Añade XML, secuencias y columnas autonuméricas.



Introducción

- SQL no forma parte del modelo relacional
- Query-By-Example (QBE), otros lenguajes de consulta pueden aplicarse también al modelo
- SQL aceptado por Codd, con matices
- Sirve como lenguaje completo: de definición (DDL) y de manipulación (DML) de datos según el modelo relacional
- Tiene una estructura “pseudo inglesa”
- Se utiliza como “*lingua franca*” entre sistemas



02

Características de SQL

Características

- El Lenguaje de Definición de Datos (LDD)
 - Proporciona comandos para la creación, borrado y modificación de esquemas relacionales
- El Lenguaje de Manipulación de Datos (LMD)
 - Basado en el álgebra relacional y el cálculo relacional permite realizar consultas y adicionalmente insertar, borrar y actualizar de tuplas
 - Ejecutado en una consola interactiva
 - Embebido dentro de un lenguaje de programación de propósito general
- Definición de vistas
- Autorización
 - Definición de usuarios y privilegios
- Integridad de datos
- Control de Transacciones



SQL

SQL - LDD

SQL - LMD

- CREATE
- ALTER
- DROP

- Selección:
 - SELECT
- Modificación:
 - INSERT
 - UPDATE
 - DELETE





03

LDD

Creación de tablas

La creación de tablas se lleva a cabo con la sentencia **CREATE TABLE**.

Ejemplo: creación del siguiente esquema de BD.

CLIENTES (DNI, NOMBRE, DIR)
SUCURSALES (NSUC, CIUDAD)
CUENTAS (COD, DNI, NSUCURS, SALDO)

1.-CREATE TABLE CLIENTES (
DNI VARCHAR(9) **NOT NULL**,
NOMBRE VARCHAR(20),
DIR VARCHAR(30),
PRIMARY KEY (DNI)
);

2.-CREATE TABLE SUCURSALES (
NSUC VARCHAR(4) **NOT NULL**,
CIUDAD VARCHAR(30),
PRIMARY KEY (NSUC)
);



Creación de tablas

El siguiente paso es crear la tabla CUENTAS, con las claves externas:

```
CREATE TABLE CUENTAS (  
  COD VARCHAR(4) NOT NULL,  
  DNI VARCHAR(9) NOT NULL,  
  NSUCURS VARCHAR(4) NOT NULL,  
  SALDO INT DEFAULT 0,  
  PRIMARY KEY (COD, DNI, NSUCURS),  
  FOREIGN KEY (DNI) REFERENCES CLIENTES (DNI),  
  FOREIGN KEY (NSUCURS) REFERENCES SUCURSALES (NSUC) );
```



Modificación de Tablas

- **Modificación de tablas: sentencia ALTER TABLE.**
Es posible añadir, modificar y eliminar campos. Ejemplos:
Adición del campo PAIS a la tabla CLIENTES
ALTER TABLE CLIENTES ADD PAIS VARCHAR(10);
Modificación del tipo del campo PAIS
ALTER TABLE CLIENTES MODIFY PAIS VARCHAR(20);
Eliminación del campo PAIS de la tabla CLIENTES
ALTER TABLE CLIENTES DROP PAIS;
También es posible añadir nuevas restricciones a la tabla
(claves externas, restricciones check).
- **Eliminación de tablas: sentencia DROP TABLE.**
DROP TABLE CUENTAS;
-- Las tablas a las que referencia deben haber sido eliminadas antes.



04

LMD



Estructura básica de la sentencia **SELECT**

Consta de tres cláusulas:

SELECT

La lista de los atributos que se incluirán en el resultado de una consulta.

FROM

Especifica las relaciones que se van a usar como origen en el proceso de la consulta.

WHERE

Especifica la condición de filtro sobre las tuplas en términos de los atributos de las relaciones de la cláusula **FROM**.



Estructura de la sentencia **SELECT**

SELECT A1, ..., An

-Describe la salida deseada con:

- Nombres de columnas
- Expresiones aritméticas
- Literales
- Funciones escalares
- Funciones de columna

FROM T1, ..., Tn
WHERE P

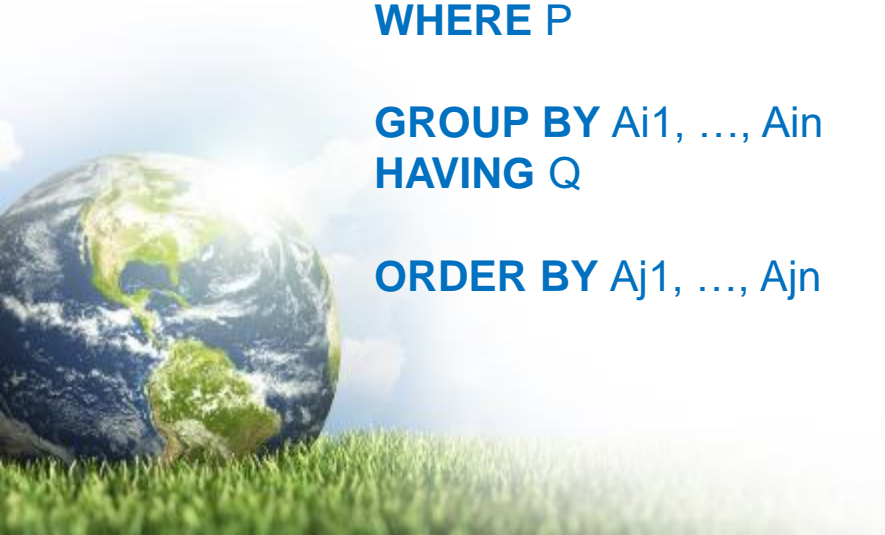
- Nombres de las tablas / vistas
- Condiciones de selección de filas

GROUP BY Ai1, ..., Ain
HAVING Q

- Nombre de las columnas
- Condiciones de selección de grupo

ORDER BY Aj1, ..., Ajn

- Nombres de columnas



Eliminación de filas duplicadas


SQL permite duplicados en el resultado

Para eliminar las tuplas repetidas se utiliza la cláusula **DISTINCT**.

También es posible pedir explícitamente la inclusión de filas repetidas mediante el uso de la cláusula **ALL**.

```
SELECT ADMRDEPT  
FROM DEPARTMENT
```

ADMRDEPT



A00
A00
A00
A00
D01
D01
A00
E01

```
SELECT ALL ADMRDEPT  
FROM DEPARTMENT
```



```
SELECT DISTINCT ADMRDEPT  
FROM DEPARTMENT
```

ADMRDEPT

A00
D01
E01

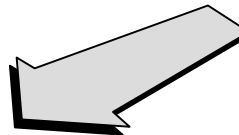


Proyección de todos los atributos

Se puede pedir la proyección de todos los atributos de la consulta mediante utilizando el símbolo '*'

La tabla resultante contendrá todos los atributos de las tablas que aparecen en la cláusula **FROM**.

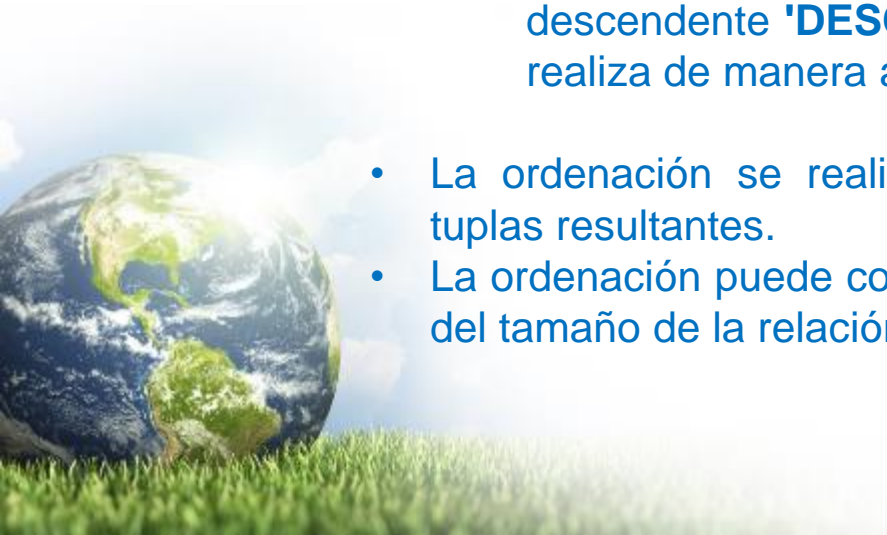
```
SELECT * FROM DEPARTMENT
```



DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	
B01	PLANNING	000020	A00	
C01	INFORMATION CENTER	000030	A00	
D01	DEVELOPMENTCENTER	- - - - -	A00	
D11	MANUFACTURING SYSTEMS	000060	D01	
D21	ADMINISTRATION SYSTEMS	000070	D01	
E01	SUPPORT SERVICES	000050	A00	
E11	OPERATIONS	000090	E01	
E21	SOFTWARE SUPPORT	000100	E01	

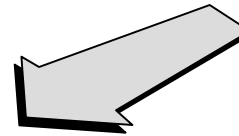
Salida ordenada

- SQL permite controlar el orden en el que se presentan las tuplas de una relación mediante la cláusula **ORDER BY**.
- La cláusula ORDER BY tiene la forma
 - **ORDER BY A1 <DIRECCION>, ..., An <DIRECCION>**
 - A1, ..., An son atributos de la relación resultante de la consulta
 - Ai <DIRECCION> controla si la ordenación es Ascendente '**ASC**' o descendente '**DESC**' por el campo Ai. Por defecto la ordenación se realiza de manera ascendente.
- La ordenación se realiza tras haber ejecutado la consulta sobre las tuplas resultantes.
- La ordenación puede convertirse en una operación costosa dependiendo del tamaño de la relación resultante.



Salida ordenada

```
SELECT DEPTNO, DEPTNAME, ADMRDEPT  
FROM DEPARTMENT  
ORDER BY ADMRDEPT ASC
```



DEPTNO	DEPTNAME	ADMRDEPT
A00	SPIFFY COMPUTER SERVICE DIV.	A00
C01	INFORMATION CENTER	A00
B01	PLANNING	A00
E01	SUPPORTSERVICES	A00
D01	DEVELOPMENTCENTER	A00
D11	MANUFACTURING SYSTEMS	D01
D21	ADMINISTRATION SYSTEMS	D01
E21	SOFTWARE SUPPORT	E01
E11	OPERATIONS	E01



Selección de filas

- La cláusula **WHERE** permite filtrar las filas de la relación resultante.
 - La condición de filtrado se especifica como un predicado.
- El predicado de la cláusula WHERE puede ser simple o complejo
 - Se utilizan los conectores lógicos AND (conjunción), OR (disyunción) y NOT (negación)
- Las expresiones pueden contener
 - Predicados de comparación
 - BETWEEN / NOT BETWEEN
 - IN / NOT IN (con y sin subconsultas)
 - LIKE / NOT LIKE
 - IS NULL / IS NOT NULL
 - ALL, SOME/ANY (subconsultas)
 - EXISTS (subconsultas)



Selección de filas

- Predicados de comparación
 - Operadores: =, <> (es el ≠), <, <=, >=, >
- **BETWEEN Op1 AND Op2**
 - Es el operador de comparación para intervalos de valores o fechas.
- **IN** es el operador que permite comprobar si un valor se encuentra en un conjunto.
 - Puede especificarse un conjunto de valores (**Val1, Val2, ...**)
 - Puede utilizarse el resultado de otra consulta **SELECT**.



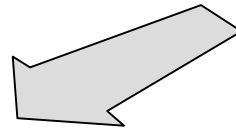
Selección de filas

- **LIKE** es el operador de comparación de cadenas de caracteres.
 - SQL distingue entre mayúsculas y minúsculas
 - Las cadenas de caracteres se incluyen entre comillas simples
 - SQL permite definir **patrones** a través de los siguientes caracteres:
 - '%', que es equivalente a "cualquier subcadena de caracteres"
 - '_', que es equivalente a "cualquier carácter"
- **IS NULL** es el operador de comparación de valores nulos.



Ejemplo selección de filas

```
SELECT DEPTNO, ADMRDEPT  
FROM DEPARTMENT  
WHERE ADMRDEPT='A00'
```



DEPTNO	ADMRDEPT
A00	A00
B01	A00
C01	A00
D01	A00
E01	A00



Múltiples condiciones AND/OR

- Necesito el número de empleado, el trabajo y el nivel de formación de los analistas con un nivel de educación 16

```
SELECT EMPNO, JOB, EDLEVEL  
FROM EMPLOYEE  
WHERE JOB='ANALYST' AND EDLEVEL=16
```



EMPNO	JOB	EDLEVEL
000130	ANALYST	16

```
SELECT EMPNO, JOB, EDLEVEL  
FROM EMPLOYEE  
WHERE (JOB='ANALYST' AND EDLEVEL=16)  
OR EDLEVEL=18  
ORDER BY JOB, EDLEVEL
```



EMPNO	JOB	EDLEVEL
000130	ANALYST	16
000140	ANALYST	18
000220	DESIGNER	18
000020	MANAGER	18
000010	PRES	18

SELECT con BETWEEN

Obtener el número de empleado y el nivel de todos los empleados con un nivel entre 12 y 15



```
SELECT EMPNO, EDLEVEL  
FROM EMPLOYEE  
WHERE EDLEVEL BETWEEN 12 AND 15  
ORDER BY EDLEVEL
```

EMPNO	EDLEVEL
000310	12
000290	12
000300	14
000330	14
000100	14
000230	14
000120	14
000270	15
000250	15



SELECT con IN

Listar los apellidos y nivel de formación de todos los empleados de nivel 14, 19 o 20



```
SELECT LASTNAME, EDLEVEL  
FROM EMPLOYEE  
WHERE EDLEVEL IN (14, 19, 20)  
ORDER BY EDLEVEL, LASTNAME
```

LASTNAME	EDLEVEL
JEFFERSON	14
LEE	14
O'CONNELL	14
SMITH	14
SPENSER	14
LUCCHESI	19
KWAN	20




Búsqueda parcial - LIKE

Obtener el apellido de todos los empleados cuyo apellido empiece por G

```
SELECT LASTNAME  
FROM EMPLOYEE  
WHERE LASTNAME LIKE 'G%';
```

```
SELECT LASTNAME  
FROM EMPLOYEE  
WHERE LASTNAME LIKE '%SON';
```



THOMPSON
HENDERSON
ADAMSON
JEFFERSON
JOHNSON

Búsqueda parcial - LIKE

¿Qué empleados tienen una C como segunda letra de su apellido?

```
SELECT LASTNAME  
FROM EMPLOYEE  
WHERE LASTNAME LIKE '_C%';
```



Búsqueda parcial – NOT LIKE

Necesito todos los departamentos excepto aquellos cuyo número NO empiece por 'D'



```
SELECT DEPTNO, DEPTNAME  
FROM DEPARTMENT  
WHERE DEPTNO NOT LIKE 'D%';
```

DEPTNO	DEPTNAME
A00	SPIFFY COMPUTER SERVICE DIV.
B01	PLANNING
C01	INFORMATION CENTER
E01	SUPPORT SERVICES
E11	OPERATIONS
E21	SOFTWARE SUPPORT



Renombramiento de tablas

Es posible obtener “copias” de una tabla situando etiquetas junto al nombre de las tablas.

Ejemplo: en el siguiente esquema de base de datos, queremos obtener el nombre de los empleados con al menos dos hijos.

EMP (DNI, NOM)

HIJOS (DNI, NOMH)

```
SELECT NOM  
FROM EMP, HIJOS H1, HIJOS H2  
WHERE EMP.DNI = H1.DNI AND EMP.DNI = H2.DNI AND  
H1.NOMH <> H2.NOMH;
```

Realiza el producto cartesiano de las tres tablas y da como resultado aquellas tuplas con igual DNI en las tres y con distinto nombre en los hijos.



05

Tipos y
Expresiones
SQL

Tipos SQL y valores literales

Tipo	Ejemplo
BIGINT	8589934592
INTEGER	186282
SMALLINT	186
NUMERIC(8,2)	999999.99 (precisión, escala)
DECIMAL(8,2)	999999.99 (precisión, escala)
REAL	6.02257E23
DOUBLE PRECISION	3.141592653589
FLOAT	6.02257E23
CHARACTER(<i>max</i>)	'GREECE ' (15 caracteres)
VARCHAR(<i>n</i>)	'hola'
DATE	date 'YYYY-MM-DD'
TIME	time 'hh:mm:ss.ccc'
TIMESTAMP	timestamp 'YYYY-MM-DD hh:mm:ss.ccc'

Expresiones

- Aunque SQL no es un lenguaje de programación de propósito general, permite definir expresiones calculadas.
- Estas expresiones pueden contener
 - Referencias a columnas
 - Valores literales
 - Operadores aritméticos
 - Llamadas a funciones
- Los operadores aritméticos son los habituales: +, -, * y /
 - Estos operadores sólo funcionan con valores numéricos.
 - Los operadores '+' y '-' habitualmente funcionan para fechas.
- Aunque las normas SQL definen un conjunto mínimo de funciones, los SGBD proporcionan una gran variedad.
 - Es necesario consultar el manual del SGBD particular.

Funciones matemáticas comunes

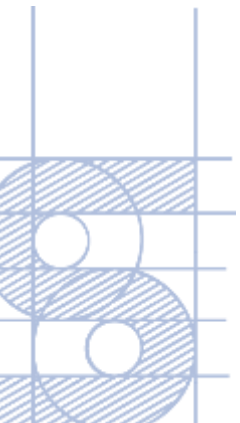
Descripción	IBM DB2	SQL Server	Oracle	MySQL
Valor absoluto	ABSs	ABS	ABS	ABS
Menor entero \geq valor	CEIL	CEILING	CEIL	CEILING
Menor entero \leq valor	FLOOR	FLOOR	FLOOR	FLOOR
Potencia	POWER	POWER	POWER	POWER
Redondeo a un número de cifras decimales	ROUND	ROUND	ROUND	ROUND
Módulo	MOD.	%	MOD.	%

Funciones de cadena

Descripción	IBM DB2	SQL Server	Oracle	MySQL
Convierte todos los caracteres a minúsculas	LOWER	LOWER	LOWER	LOWER
Convierte todos los caracteres a mayúsculas	UPPER	UPPER	UPPER	UPPER
Elimina los blancos del final de la cadena	RTRIM	RTRIM	RTRIM	RTRIM
Elimina los blancos del comienzo de la cadena	LTRIM	LTRIM	LTRIM	LTRIM
Devuelve una subcadena	SUBSTR	SUBSTRING	SUBSTR	SUBSTRING
Concatena dos cadenas	CONCAT	+	CONCAT	CONCAT

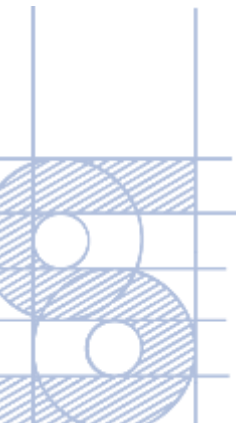
Operadores de conjunto

- SQL incluye las operaciones:
 - **UNION**
 - **INTERSECT**
 - **EXCEPT (MINUS** en Oracle)
- Por definición los operadores de conjunto eliminan las tuplas duplicadas.
 - Para retener duplicados se debe utilizar **<Operador> ALL**



UNION

- Cada SELECT debe tener el mismo número de columnas
- Las columnas correspondientes deben tener tipos de datos compatibles
- UNION elimina duplicados
- Si se indica, el ORDER BY debe ser la última cláusula de la sentencia



Consultar más de una tabla: JOIN

EMPLOYEE

EMPNO	LASTNAME	WORKDEPT...
000010	HAAS	A00
000020	THOMPSON	C01
000030	KWAN	C01
000040	PULASKI	D21
:	:	

DEPARTMENT

DEPTNO	DEPTNAME	...
A00	SPIFFY COMPUTER SERVICE DIV.	
C01	INFORMATION CENTER	
D01	DEVELOPMENT CENTER	
D21	ADMINISTRATION SYSTEMS	
:	:	



Funciones de columna

- Las funciones de columna o funciones de agregación son funciones que toman una colección (conjunto o multiconjunto) de valores de entrada y devuelve un solo valor.
- Las funciones de columna disponibles son: AVG, MIN, MAX, SUM, COUNT.
- Los datos de entrada para SUM y AVG deben ser una colección de números, pero el resto de operadores pueden operar sobre colecciones de datos de tipo no numérico

Funciones de columna

- Por defecto las funciones se aplican a todas las tuplas resultantes de la consulta.
- Podemos agrupar las tuplas resultantes para poder aplicar las funciones de columna a grupos específicos utilizando la cláusula GROUP BY.
- En la cláusula SELECT de consultas que utilizan funciones de columna solamente pueden aparecer funciones de columna.
- En caso de utilizar GROUP BY, también pueden aparecer columnas utilizadas en la agrupación.
- Adicionalmente se pueden aplicar condiciones sobre los grupos utilizando la cláusula HAVING.

Funciones de columna

- Cálculo del total : SUM (expresión)
- Cálculo de la media : AVG (expresión)
- Obtener el valor mínimo : MIN (expresión)
- Obtener el valor máximo : MAX (expresión)
- Contar el número de filas que satisfacen la condición de búsqueda : COUNT(*)
 - Los valores NULL SI se cuentan.
- Contar el número de valores distintos en una columna : COUNT (DISTINCT nombre-columna)
 - Los valores NULL NO se cuenta.

GROUP BY

Necesito conocer los salarios de todos los empleados de los departamentos A00, B01, y C01. Además, para estos departamentos quiero conocer su masa salarial.



```
SELECT  WORKDEPT, SALARY
FROM    EMPLOYEE
WHERE   WORKDEPT IN ('A00', 'B01', 'C01')
ORDER BY WORKDEPT
```

WORKDEPT	SALARY
A00	52750.00
A00	46500.00
A00	29250.00
B01	41250.00
C01	38250.00
C01	23800.00
C01	28420.00

```
SELECT  WORKDEPT, SUM(SALARY) AS SUM
FROM    EMPLOYEE
WHERE   WORKDEPT IN ('A00', 'B01', 'C01')
GROUP BY WORKDEPT
ORDER BY WORKDEPT
```

WORKDEPT	SUM
A00	128500.00
B01	41250.00
C01	90470.00

Ejecución de consultas **SELECT**

El orden de ejecución de una consulta es el siguiente:

1. Se aplica el predicado **WHERE** a las tuplas del producto cartesiano/join/vista que hay en el **FROM**.
2. Las tuplas que satisfacen el predicado de **WHERE** son colocadas en grupos siguiendo el patrón **GROUP BY**.
3. Se ejecutan la cláusula **HAVING** para cada grupo de tuplas anterior.
4. Los grupos obtenidos tras aplicar **HAVING** son los que serán procesados por **SELECT**, que calculará, en los casos que se incluyan, las funciones de agregación que le acompañan.
5. A las tuplas resultantes de los pasos anteriores se le aplica la ordenación descrita en la cláusula **ORDER BY**.

Subconsulta con IN

¿Qué departamentos no tienen proyectos asignados?



Tabla DEPARTMENT

<u>DEPTNO</u>	<u>DEPTNAME</u>
A00	SPIFFY COMPUTER SERVICE
B01	PLANNING
C01	INFORMATION CENTER
...	...

```
SELECT DEPTNO, DEPTNAME
FROM DEPARTMENT
WHERE DEPTNO NOT IN (SELECT DEPTNO
                     FROM PROJECT)
```

Resultado final

DEPTNO	DEPTNAME
A00	SPIFFY COMPUTER SERVICE

Resultado subconsulta

B01
C01
D01
D11
D21
E01
E11
E21

06

Modificación
de la BBDD



Modificación de la BBDD

Las instrucciones SQL que permiten modificar el estado de la BBDD son:

INSERT : Añade filas a una tabla

UPDATE : Actualiza filas de una tabla

DELETE : Elimina filas de una tabla

La instrucción **INSERT**

La **inserción** de tuplas se realiza con la sentencia **INSERT**,

Es posible insertar directamente valores.

O bien insertar el conjunto de resultados de una consulta.

En cualquier caso, los valores que se insertan deben pertenecer al dominio de cada uno de los atributos de la relación.

Ejemplos: CLIENTES (DNI, NOMBRE, DIR)

La inserción

```
INSERT INTO CLIENTES VALUES (1111, 'Mario',  
'C/. Mayor, 3');
```

Es equivalente a las siguientes sentencias

```
INSERT INTO CLIENTES (NOMBRE, DIR, DNI)  
VALUES ('Mario', 'C/. Mayor, 3', 1111);  
INSERT INTO CLIENTES (DNI, DIR, NOMBRE)  
VALUES (1111, 'C/. Mayor, 3', 'Mario');
```

La instrucción UPDATE

La **modificación** de tuplas se realiza con la sentencia **UPDATE**,

Es posible elegir el conjunto de tuplas que se van a actualizar usando la clausula **WHERE**.

Ejemplos: CUENTAS (COD, DNI, NSUCURS, SALDO)

Suma del 5% de interés a los saldos de todas las cuentas.

```
UPDATE CUENTAS SET SALDO = SALDO * 1.05;
```

Suma del 1% de bonificación a aquellas cuentas cuyo saldo sea superior a 100.000 €.

```
UPDATE CUENTAS SET SALDO = SALDO * 1.01  
WHERE SALDO > 100000;
```

Modificación de DNI y saldo simultáneamente para el código 898.

```
UPDATE CUENTAS SET DNI='555', SALDO=10000  
WHERE COD LIKE '898';
```

La instrucción DELETE

La **eliminación** de tuplas se realiza con la sentencia **DELETE**:

DELETE FROM R WHERE P; -- WHERE es opcional

Elimina tuplas completas, no columnas. Puede incluir subconsultas.

Ejemplos: para la BD de CLIENTES, CUENTAS, SUCURSALES.

Eliminar todas cuentas con código entre 1000 y 1100.

DELETE FROM CUENTAS WHERE COD BETWEEN 1000 AND 1100;

Eliminar todas las cuentas del cliente “Jose María García”.

**DELETE FROM CUENTAS WHERE DNI IN
(SELECT DNI FROM CLIENTES
WHERE NOMBRE LIKE 'Jose María García');**

Eliminar todas las cuentas de sucursales situadas en "Chinchón".

**DELETE FROM CUENTAS WHERE NSUCURS IN
(SELECT NSUC FROM SUCURSALES
WHERE CIUDAD LIKE 'Chinchón');**