

Lenguaje SQL-1 : definición de tablas y actualizar

BASES DE DATOS

Profesor: Héctor Gómez Gauchía

Materiales: Héctor Gómez Gauchía, Mercedes García Merayo

Introducción SQL

► Componentes

- Lenguaje de Definición de Datos (DDL)
- Lenguaje de Manipulación de Datos (DML)
- Control de transacciones
- SQL incorporado y SQL dinámico
- Control de acceso (DAL)

Definición de Datos (DDL)

- ▶ El DDL de SQL permite la especificación de relaciones así como de la información relativa a dichas relaciones.
- ▶ Esquema de cada relación.
- ▶ Dominio de valores asociados a cada atributo.
- ▶ Restricciones de integridad.
- ▶ Conjunto de índices para cada relación...

Definición de Datos

- ▶ **CHAR(n)**: cadena de caracteres de longitud fija n.
- ▶ **VARCHAR2(n)**: cadena de caracteres de longitud máxima n.
- ▶ **LONG**: caracteres hasta 2 gigabytes.
- ▶ **NUMBER[(p,s)]**: donde **p** es el número total de dígitos (con un máximo de 38) y “**s**” indica la escala, esto es, el máximo de dígitos decimales.
 - ▶ **NUMBER(10,5)**, podrá contener cualquier número siempre y cuando el número de dígitos enteros más el número de dígitos decimales no supere 10 (y no 15). Si la escala es negativa el número se redondeará al número de posiciones a la izquierda del valor decimal.

Definición de Datos

7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	No valido, supera la precisión
7,456,123.89	NUMBER(7,-2)	7456100

Definición de Datos

- ▶ **DATE:** fecha y hora. Desde 1 de Enero del 4712 AC, hasta 31 de Diciembre 9999 DC. Almacena información del día, mes, año, hora, minuto y segundo.

El formato por defecto viene dado por el parámetro `NLS_DATE_FORMAT`.

Internamente una fecha se almacena como el número de días desde cierto punto de inicio, lo que permite que puedan ser tratadas en operaciones aritméticas:

- `'1-JAN-2001' + 10 = '11-JAN-2001'`
- `'1-JAN-2000' - 1 = '31-DEC-1999'`
- `'10-MAY-2000' - '1-MAY-2000' = 9`

Definición de Datos

- ▶ **TIMESTAMP [(fs)]:** almacena la misma información que DATE más fracciones de segundo. El parámetro fs (0-9) indica el número de dígitos de dicha fracción. Por defecto dicho valor es 6.
- ▶ Oracle dispone de más tipos de datos que pueden consultarse en la documentación <http://docs.oracle.com>

Definición de Datos -Tablas

```
create table R
(A1 D1 [propiedades],
 A2 D2 [propiedades],
 .....
 An Dn [propiedades],
< restricción integridad 1>,
 .....
< restricción integridad k>);
```

► [Propiedades]

- Valor por defecto: se asigna un valor por defecto a un atributo mediante la propiedad
default valor por defecto
- Valor obligatorio: se obliga a la asignación de un valor al atributo mediante la propiedad **not null**. Por defecto el atributo acepta valores nulos.

Definición de Datos - Tablas

- ▶ Las *restricciones* son condiciones de obligado cumplimiento para uno o más atributos de una relación.
- ▶ Es recomendable indicar un nombre de restricción. Por defecto Oracle asignará un nombre si este no se especifica.
- ▶ Recomendaciones
 - ▶ Tres letras para el nombre de la tabla.
 - ▶ Carácter de subrayado.
 - ▶ Tres letras con la columna afectada por la restricción.
 - ▶ Carácter de subrayado.
 - ▶ Dos letras con la abreviatura del tipo de restricción:
 - ▶ PK. PRIMARY KEY , UK. UNIQUE , FK. FOREIGN KEY o CK. CHECK

Definición de Datos - Tablas

► Restricciones

- **unique(Aj1,...,Ajp)**: la combinación de valores de los atributos es única.
- **primary key(Aj1,...,Ajm)**: no nulos y únicos.
- **check (<predicado>)** el predicado debe ser satisfecho por todas las tuplas.

- Cuando afectan a un único atributo pueden indicarse junto con las propiedades de dicho atributo, en otro caso, después de la descripción de todas las columnas o atributos.

Definición de Datos - Tablas

Create table *sucursal*

```
(nombre_sucursal VARCHAR2(15) CONSTRAINT suc_PK PRIMARY KEY,  
ciudad CHAR(20) NOT NULL CONSTRAINT c1_UK UNIQUE,  
activos NUMBER(12,2) default 0);
```

Create table *cliente*

```
(dni VARCHAR2(9) NOT NULL,  
nombre_cliente CHAR(35) NOT NULL,  
domicilio CHAR(50) NOT NULL,  
CONSTRAINT c1_PK PRIMARY KEY (dni));
```

Definición de Datos - Tablas

- ▶ **Clave externa:** definen un conjunto de atributos cuyos valores coinciden con los de los atributos de la *clave primaria (o única)* de la misma tabla o de otra.

foreign key(Aj1,...,Ajm) references *tabla* (Bj1,...Bjm)

- ▶ Si alguno de los atributos de la clave externa toma valor nulo el valor de los demás no es chequeado contra los valores de la clave primaria (o única) a la que hace referencia.

Definición de Datos - Tablas

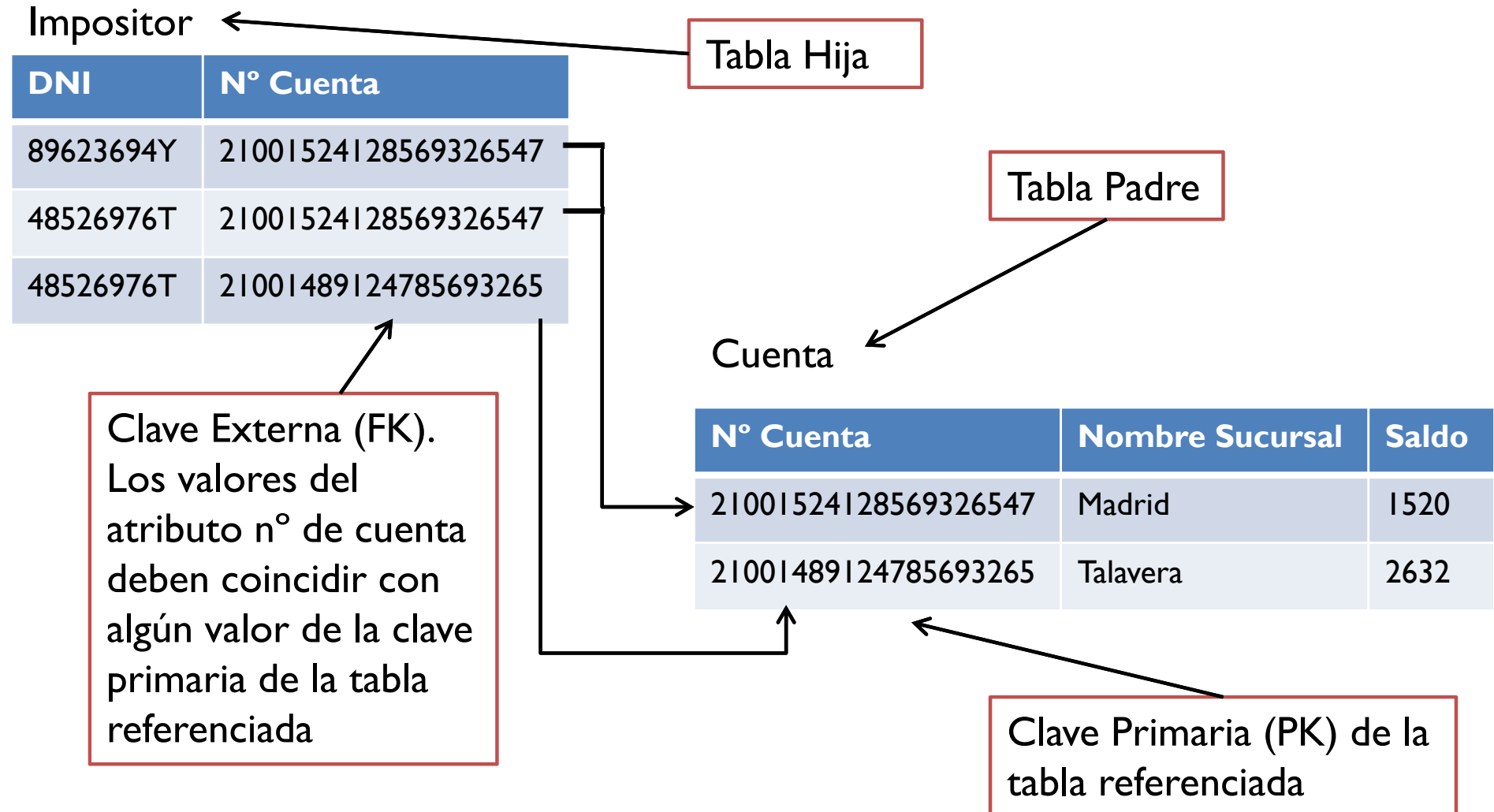
Create table *cuenta*

```
(numero_cuenta CHAR (20) PRIMARY KEY,  
 nombre_sucursal char(15) REFERENCES sucursal,  
 saldo NUMBER(12,2) default 100,  
 CONSTRAINT imp_minimo CHECK(saldo >=100))
```

Create table *impositor*

```
(dni CHAR(9) CONSTRAINT imp_dni_FK REFERENCES cliente,  
 numero_cuenta CHAR(20) NOT NULL,  
 CONSTRAINT imp_PK PRIMARY KEY (dni, numero_cuenta),  
 CONSTRAINT imp_ct_FK FOREIGN KEY (numero_cuenta)  
 REFERENCES cuenta)
```

Definición de Datos - Tablas



Definición de Datos - Tablas

- ▶ **Violación de una restricción de clave externa:**
 - ▶ Actualización de una clave primaria (o única) referenciada en otra tabla con una clave externa.
 - ▶ Borrado de una fila cuyos valores de la clave primaria (o única) son referenciados mediante una clave externa de otra tabla.
- ▶ Cuando se viola una restricción se rechaza la acción que la ha provocado (actualización o borrado).
- ▶ Existen cláusulas que permiten que el sistema realice una acción correctora en algunos casos.

Definición de Datos – Restricciones de Borrado

Situación: Si una tabla (hija) tiene FK o referencia a otra tabla (padre), qué pasa cuando se intenta borrar una fila del padre existiendo filas en la tabla hija con la FK con el valor (de la PK) del padre?

- ▶ **on delete cascade:** Cuando una fila de la tabla padre es borrada, todas las filas de las tabla hijas que referencian los valores de su clave primaria también son borradas.
- ▶ **on delete set null:** Cuando una fila de la tabla padre es borrada, los valores de los atributos de las filas en las tablas hijas, que referencian los valores de su clave primaria, se actualizan al valor NULL. Dichos atributos deben aceptar valores NULL.
- ▶ **Por defecto:** (no se pone cláusula en Oracle) **restrict**, cuando la fila del padre se intenta borrar, solo lo permite si no tiene hijos

Definición de Datos

Create table *cuenta*

```
(numero_cuenta CHAR (20) PRIMARY KEY,  
 nombre_sucursal char(15)  
 CONSTRAINT ct_FK REFERENCES sucursal on delete set null,  
 saldo NUMBER(12,2) default 100,  
 CONSTRAINT imp_minimo CHECK(saldo >=100))
```

Create table *impositor*

```
(dni CHAR(9) CONSTRAINT imp_dni_FK REFERENCES cliente on  
 delete cascade,  
 numero_cuenta CHAR(20),  
 CONSTRAINT imp_PK PRIMARY KEY (dni, numero_cuenta),  
 CONSTRAINT imp_ct_FK FOREIGN KEY (numero_cuenta)  
 REFERENCES cuenta on delete cascade)
```

Definición de Datos

- ▶ Añadir atributos a una tabla.

`alter table R add Atributo Dominio [propiedades]`

- ▶ Eliminar atributos de una tabla.

`alter table R drop COLUMN Atributo`

- ▶ *No se puede eliminar la única columna de una tabla.*

- ▶ *Si la columna interviene en una constraint dará error:*

- ▶ `alter table R drop Atributo CASCADE CONSTRAINTS`

- ▶ Modificar atributos de una tabla.

`alter table R modify (Atributo Dominio [propiedades])`

- ▶ Renombrar atributos de una tabla.

`alter table R rename column Atributo1 to Atributo2`

Definición de Datos

- ▶ Añadir restricciones a una tabla.

`alter table R add CONSTRAINT nombre Tipo (columnas)`

- ▶ Eliminar restricciones de una tabla.

`alter table R drop {PRIMARY KEY|UNIQUE(campos) |
CONSTRAINT nombre [CASCADE]}`

La opción **CASCADE** hace que se eliminen las restricciones de integridad que dependen de la eliminada.

- ▶ Desactivar restricciones.

`alter table R disable CONSTRAINT nombre [CASCADE]`

- ▶ Activar restricciones.

`alter table R enable CONSTRAINT nombre`

Definición de Datos

```
ALTER TABLE cuenta ADD comision NUMBER(4,2);  
ALTER TABLE cuenta ADD fecha_apertura DATE;  
ALTER TABLE cuenta DROP COLUMN nombre_sucursal;  
ALTER TABLE cuenta MODIFY comision DEFAULT 1.5;  
ALTER TABLE cliente MODIFY nombre_cliente NULL;  
ALTER TABLE sucursal  
    ADD CONSTRAINT cd_UK UNIQUE(ciudad);
```

Definición de Datos

- ▶ Descripción de una tabla.

describe *R*

- ▶ Eliminar una tabla.

drop table *R* [cascade constraints]

La opción CASCADE hace que se eliminen las restricciones de integridad que dependen de la tabla eliminada.

- ▶ Renombrar una tabla.

rename *R* to *S*

- ▶ Borrar contenido de una tabla.

truncate table *R*

Índices

- ▶ Los **índices** permiten que las bases de datos aceleren las operaciones de consulta y ordenación sobre los campos a los que el índice hace referencia.
- ▶ Algunos de los índices se crean de manera implícita, como consecuencia de las restricciones **PRIMARY KEY** y **UNIQUE**. (Depende del Servidor de BD)
- ▶ Se pueden crear explícitamente para aquellos campos sobre los cuales se realizarán búsquedas e instrucciones de ordenación frecuente.

```
CREATE [unique] INDEX NombreIndice  
ON NombreTabla(col1,...,colk);
```

Definición de Datos

Empleado

DNI	Nombre	Apellido	Dirección	Nº Empleado
3274623R	Antonio	López	Retama 14	RA45823633
27347234T	Marta	Sánchez	Loreto 134	PE74563669
85647456W	Antonio	López	Plaza Tornero 23	TY45236916
37562365F	María	Puente	Travesía del Reloj 1	MU45992355
34126455Y	Juan	Panero	Viviana 123	YR42326524

Varias filas pueden tomar los mismos valores en los atributos del índice

```
Create index ID_Nombre on Empleado (Apellido, Nombre);
```

```
Create unique index ID_Emp on Empleado (NºEmpleado);
```

No hay dos filas con el mismo nº de registro

Modificación de Datos

```
INSERT INTO NombreTabla [lista_atributos]  
VALUES (Lista_de_valores)
```

- ▶ La sintaxis básica de la sentencia **INSERT** permite insertar una tupla de valores en una tabla.
- ▶ La tupla de valores se escribe como una lista de valores, que deben coincidir con el tipo de los campos de la tabla en el orden que están definidos, separada por comas.

Modificación de Datos

Insertar un préstamo en la relación Prestamo

```
INSERT INTO Prestamo  
VALUES ('Navacerrada', 'Pepe Pérez', 125000)
```

También es posible obtener los datos mediante una consulta **SELECT** que actúe como proveedor de datos.

```
INSERT INTO Prestamo  
SELECT * FROM Nuevos_Prestamos
```

Modificación de Datos

DELETE FROM *NombreTabla*
WHERE *condición*

- ▶ La sintaxis básica de la sentencia **DELETE** permite borrar las tuplas de la tabla que cumplan la condición.
- ▶ La parte **WHERE** es opcional y, si no se especifica, se borran todas las tuplas de la tabla

```
DELETE FROM Prestamo  
WHERE NumPrestamo='P-260'
```

Modificación de Datos

- ▶ La cláusula **WHERE** admite consultas anidadas

Borrar todos los clientes que tengan un préstamo no registrado en la relación Préstamo.

```
DELETE
FROM Clientes
WHERE Clientes.NumPrestamo NOT IN
(SELECT NumPrestamo FROM Prestamo)
```

Modificación de Datos

```
UPDATE NombreTabla  
SET col1=valor1,..., colk = valork  
WHERE condición
```

- ▶ La sentencia **UPDATE** permite modificar los valores de los atributos de las tuplas que cumplan una determinada condición.
- ▶ La parte **WHERE** es opcional y, si no se especifica, se actualizarán todas las tuplas de la tabla.

```
UPDATE Prestamo  
SET importe=200.000  
WHERE NumPrestamo='P-170'
```

Modificación de Datos

- ▶ La cláusula **WHERE** admite consultas anidadas

Modificar todos los prestamos cuya sucursal hay sido cerrada a la sucursal 'Centro'.

```
UPDATE Prestamo
SET sucursal= 'Centro'
WHERE sucursal IN
        (SELECT sucursal
         FROM Sucursales_Cerradas)
```

Secuencias

- ▶ Una **secuencia** sirve para generar automáticamente números distintos.
- ▶ Las secuencias se almacenan independientemente de la tabla, por lo que la misma secuencia se puede utilizar para diversas tablas.

```
CREATE SEQUENCE secuencia  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n|NOMAXVALUE}]  
[{MINVALUE n|NOMINVALUE}]  
[{CYCLE|NOCYCLE}]
```

Secuencias

- ▶ Los métodos **NEXTVAL** y **CURRVAL** se utilizan para obtener el siguiente número y el valor actual de la secuencia respectivamente.
- ▶ **NEXTVAL** incrementa la secuencia y devuelve el valor actual.
- ▶ **CURRVAL** devuelve el valor de la secuencia, pero sin incrementar la misma.
- ▶ No se puede utilizar como valor para la cláusula **DEFAULT** de un campo de tabla.
- ▶ Se pueden modificar las secuencias, pero la modificación sólo puede afectar a los futuros valores de la secuencia, no a los ya utilizados.

Secuencias

```
CREATE SEQUENCE numHabitacion  
INCREMENT 100  
STARTS WITH 100  
MAXVALUE 20000;
```

```
SELECT numHabitacion.CURRVAL FROM DUAL;
```

```
INSERT INTO Habitacion(numero,tipo)  
VALUES(numHabitacion.NEXTVAL,'Suite');
```