

4.3 SQL-92

LDD: definición de tablas

- Orden **CREATE TABLE**

- Define (crea) una tabla: nombre, columnas y restricciones

- **Nombre** único dentro del esquema

- Para cada **Columna** se puede indicar...

- nombre,
 - tipo de datos (dominio)
 - restricciones de columna

- **Restricciones** de tabla...

- de **clave** candidata,
 - de **integridad de entidad**,
 - de **integridad referencial**, o
 - restricciones de **otro tipo**

```
CREATE TABLE Empleado (  
    nombre ...,  
    apellido ...,  
    nss ...,  
    dni ...,  
    año nacim ...,  
    direccion ...,  
    sexo ...,  
    salario ...,  
    nssjefe ...,  
    dep ...,  
    ...  
);
```

4.3 SQL-92

LDD: definición de tablas

- **Ordenamiento de columnas y filas**
 - **Columnas ordenadas** tal como aparecen en CREATE TABLE
 - Las **filas no están ordenadas**
- Las tablas creadas con CREATE TABLE son **tablas base**
 - El SGBD las almacena físicamente en algún fichero de la BD

4.3 SQL-92

LDD: definición de tablas

- Especificación del **tipo de datos de una columna**
 1. Especificar directamente el **tipo de datos** tras nombre de la columna

```
CREATE TABLE Empleado (  
    nombre VARCHAR(15) ... ,  
    ... );
```

O bien...

2. Definir un **dominio** y usar su nombre como tipo de datos
 - Facilita el cambio del tipo de datos usado por muchas columnas
 - Esquema más comprensible

```
CREATE DOMAIN NOMBRES VARCHAR(15);  
...  
CREATE TABLE Empleado (  
    nombre NOMBRES ... ,  
    ... );
```

4.3 SQL-92

LDD: tipos de datos

- **Numéricos**

- **Enteros y Reales**

- **INTEGER** (también **INT**), **SMALLINT**,
 - **REAL** (simple precisión), **DOUBLE PRECISION**, **FLOAT(p)**

- **Con formato**

- **NUMERIC(p,e)** o **DECIMAL(p,e)** (también **DEC(p,e)**)
p: precisión (en dígitos), e: escala; valor por omisión de la escala: **e=0**

- **Cadena de caracteres**

- Longitud **fija** **CHAR(n)** n: nº de caracteres; por omisión **n=1**
 - Longitud **variable** **VARCHAR(n)** n: máximo nº de caracteres

- **Cadena de Bits**

- Longitud **fija** **BIT(n)** n: nº de bits; por omisión **n=1**
 - Longitud **variable** **BIT VARYING(n)** n: máximo nº de bits

4.3 SQL-92

LDD: tipos de datos

- **Temporales**

- **DATE** (10 posiciones) = **YEAR, MONTH, DAY** (yyyy-mm-dd)
- **TIME** (8 posiciones) = **hour, minute, second** (hh:mi:ss)
 - Sólo permitidas **fechas y horas válidas**
- **TIMESTAMP** (marca de tiempo)
 - **DATE, TIME**, fracciones de segundo y, si se incluye **WITH TIME ZONE**, desplazamiento respecto al UTC (huso horario estándar)
- **INTERVAL**
 - Período de tiempo, para incrementar/decrementar el valor actual de una fecha, hora o marca de tiempo
 - Se califica con **YEAR/MONTH** o **DAY/TIME** para indicar su naturaleza

4.3 SQL-92

LDD: tipos de datos **en Oracle**

❑ Numéricos

- Enteros y Reales, con formato o no
 - **NUMBER(p,e)** valores por omisión: **p=38** y **e=0**

❑ Cadena de caracteres

- Longitud **fija** **CHAR(n)** n: nº de caracteres; por omisión **n=1**
- Longitud **variable** **VARCHAR2(n)**, **VARCHAR(n)** n: máximo nº de caracteres

❑ Temporales

- **DATE**, **TIMESTAMP** [WITH TIME ZONE | WITH LOCAL TIME ZONE]
- Períodos de tiempo en años y meses
 - **INTERVAL YEAR [py] TO MONTH** py: nº dígitos (0-9) en el YEAR; por omisión **py=2**
- Períodos de tiempo en días, horas, minutos y segundos
 - **INTERVAL DAY [pd] TO SECOND** pd: nº dígitos (0-9) en el DAY; por omisión **pd=2**

4.3 SQL-92

LDD: definición de dominios de datos

- **CREATE DOMAIN** <nombre dominio> <tipo de datos>
[**DEFAULT** <valor defecto>] [<lista de definición de restricciones de dominio>] ;
 - <tipo de datos>: uno de los proporcionados por el SGBD (built-in)
 - <valor defecto>: (opcional)
 - Especifica el **valor por omisión** para columnas definidas de este dominio
 - Será asignado a cada columna con dicho dominio, si no tiene ya su propia cláusula **DEFAULT**
 - <lista de definición de restricciones de dominio>:(opcional)
 - Restr. Integridad que se aplican a toda columna definida sobre el dominio
 - Cada RI puede tener un **nombre**: cláusula **CONSTRAINT** <nombre_RI>

** Ejemplo: enumeración de posibles valores componentes del dominio*

```
CREATE DOMAIN COLOR VARCHAR(8) DEFAULT 'SINCOLOR'  
CONSTRAINT color_valido  
CHECK (VALUE IN ('ROJO', 'AMARILLO', 'AZUL', 'VERDE', 'SINCOLOR'));
```

- ❑ **Oracle** no implementa dominios, por lo que esta sentencia no funciona

4.3 SQL-92

LDD: definición de tablas

- Especificación de **restricciones de columna**

- Cláusula **NULL** o **NOT NULL**

- **Opción de nulo:** indica si una columna puede contener o no NULL

- ```
CREATE TABLE Empleado (...
 nombre VARCHAR(15) NOT NULL, ...);
```

- Por **omisión**, se asume **NULL**
    - Para columnas componentes de una **clave primaria** la restricción **NOT NULL** es **obligatoria**: por omisión, se asume NOT NULL

- Cláusula **DEFAULT <valor>**

- **Valor por omisión** (o por defecto)

- ```
CREATE TABLE Empleado ( ...  
    salario DECIMAL(6,2) DEFAULT 1000 NULL,... );
```

- Si una columna no tiene DEFAULT, su valor por defecto es...
 - El de su dominio, si su tipo es un dominio que incluye DEFAULT
 - NULL en cualquier otro caso, siempre que la columna permita NULL

4.3 SQL-92

LDD: definición de tablas

- Especificación de **restricciones de tabla**
 - Cláusula **PRIMARY KEY** (<lista_columnas>)
 - Columnas que componen la **clave primaria**
 - Cláusula **UNIQUE** (<lista_columnas>)
 - Columnas que forman una **clave alternativa**
 - Cláusula **FOREIGN KEY** (<lista_columnas>)
REFERENCES <tabla>(<lista_columnas>)
 - Columnas **clave externa** (Integridad Referencial) o ajena
 - SQL-92 permite que una **clave externa** haga referencia a una **clave primaria** o bien a una **clave alternativa**
 - Cláusula **CHECK** (<expresión>)
 - **Condición** sobre los **valores de las columnas** que debe cumplir **toda fila** de la tabla

4.3 SQL-92

LDD: definición de tablas

```
CREATE TABLE Empleado (  
    nombre      VARCHAR(25)    NOT NULL,  
    apellido    VARCHAR(15)    NOT NULL,  
    nss         NUMERIC(12)    NOT NULL,  
    dni         CHAR(9)        NOT NULL,  
    añonacim    DATE           NULL,  
    direccion   VARCHAR(30)    ,  
    sexo        CHAR(1)        ,  
    salario     DECIMAL(6,2)   DEFAULT 1000 NULL,  
    dep         CHAR(3)        NOT NULL,  
    nssjefe     CHAR(12)       ,  
    PRIMARY KEY (nss),  
    UNIQUE (dni),  
    CHECK (nssjefe <> nss),  
    CHECK (sexo IN ('M', 'F')),  
    FOREIGN KEY (dep) REFERENCES Departamento(codDep),  
    FOREIGN KEY (nssjefe) REFERENCES Empleado(nss)  
);
```

4.3 SQL-92

LDD: definición de tablas

- Especificación de **restricciones de tabla** (cont.)
 - Si la restricción **afecta sólo a una columna**, puede especificarse **en la definición de dicha columna** (en la misma línea)
 - Por ejemplo, si una clave externa no es compuesta, no se necesita la cláusula FOREIGN KEY: basta con la cláusula REFERENCES...

```
CREATE TABLE Empleado (  
    nombre      VARCHAR(15) NOT NULL,  
    nss         NUMERIC(12) PRIMARY KEY,  
    dni         CHAR(9)     NOT NULL UNIQUE,  
    sexo        CHAR(1)     CHECK (sexo IN ('M', 'F')),  
    dep         CHAR(3)     NOT NULL REFERENCES Departamento(codDep),  
    nssjefe     CHAR(12)    NULL REFERENCES Empleado(nss),  
    ..., -- resto de columnas  
    CHECK (nssjefe <> nss)  
);
```

4.3 SQL-92

LDD: definición de tablas

- Especificación de **restricciones de tabla** (cont.)
 - Dar **nombre** a una restricción es opcional, pero **muy conveniente**
- CONSTRAINT** <nombre_RI> <restricción>
- El nombre de cada restricción debe ser **único** dentro del mismo **esquema**
 - Identifica una restricción, por si después debe ser eliminada o sustituida por otra

```
CREATE TABLE Empleado (  
    ...,  
    CONSTRAINT emp_pk      PRIMARY KEY (nss),  
    CONSTRAINT emp_ak      UNIQUE (dni),  
    CONSTRAINT emp_jefe_ok  CHECK (nssjefe<>nss),  
    CONSTRAINT emp_sexo_ok  CHECK (sexo IN ('M', 'F')),  
    CONSTRAINT emp_fk_dep   FOREIGN KEY (dep)  
                           REFERENCES Departamento(codDep),  
    CONSTRAINT emp_fk_emp   FOREIGN KEY (nssjefe)  
                           REFERENCES Empleado(nss)  
);
```

4.3 SQL-92

LDD: definición de tablas

- Especificación de **restricciones de tabla** (cont.):
 - **Clave externa compuesta**

```
CREATE TABLE Hotel(  
    codigo CHAR(4)          PRIMARY KEY,  
    nombre VARCHAR(30) NOT NULL,  
    ...);  
  
CREATE TABLE Salon_Hotel(  
    id          CHAR(2),  
    hotel       CHAR(4)    NOT NULL REFERENCES Hotel(codigo),  
    nombre     VARCHAR(30) NOT NULL,  
    capacidad  NUMERIC(3)  NOT NULL, ...  
    PRIMARY KEY (hotel, id),  
    ...);  
  
CREATE TABLE Reserva_Salon_Hotel(  
    numero     NUMERIC(3)   PRIMARY KEY,  
    salon      CHAR(2),  
    hotel      CHAR(4),  
    fecha     DATE          NOT NULL, ...  
    FOREIGN KEY(hotel,salon) REFERENCES Salon_Hotel(hotel,id),  
    ...);
```

4.3 SQL-92

LDD: definición de tablas

- Acciones de **mantenimiento de la integridad referencial**

Cláusulas **ON DELETE** <acción>
y **ON UPDATE** <acción>

<acción> ∈ { **NO ACTION**, **CASCADE**, **SET NULL**, **SET DEFAULT** }

```
CREATE TABLE Empleado (  
    ...,  
    CONSTRAINT emp_fk_dep FOREIGN KEY (dep)  
        REFERENCES Departamento(codDep)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE,  
    CONSTRAINT emp_fk_emp FOREIGN KEY (nssjefe)  
        REFERENCES Empleado(nss)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE,  
    ... );
```

4.3 SQL-92

LDD: definición de tablas

❑ Mantenimiento de la integridad referencial en **SQL de Oracle**

- La cláusula **ON UPDATE** **no** está implementada
 - Oracle asume **ON UPDATE NO ACTION**
- La cláusula **ON DELETE** admite las siguientes opciones
 - Por omisión (si no se indica nada): **NO ACTION**
 - Acciones que se pueden indicar explícitamente: **CASCADE** , **SET NULL**

```
CREATE TABLE Empleado ( -- SQL de Oracle
...
CONSTRAINT emp_fk_dep FOREIGN KEY (dep)
    REFERENCES Departamento(codDep)
                                -- ON DELETE NO ACTION
                                -- ON UPDATE NO ACTION
CONSTRAINT emp_fk_emp FOREIGN KEY (nssjefe)
    REFERENCES Empleado(nss)
        ON DELETE SET NULL,    -- ON UPDATE NO ACTION
... );
```

4.3 SQL-92

LDD: definición de esquemas de bases de datos

- **ESQUEMA de Base de Datos Relacional**

- Agrupa tablas y otros elementos (los de una misma aplicación o usuario)
- Una base de datos puede contener varios esquemas
 - *1^{as} versiones de SQL*: todas las tablas dentro de *un esquema único y global* a todas las aplicaciones que accedían a la BD

- **Orden CREATE SCHEMA**: definición/creación de esquemas

CREATE SCHEMA <nombre de esquema>

AUTHORIZATION <identificador de autorización>

- <identificador de autorización> usuario/cuenta propietaria del esquema

CREATE SCHEMA Empresa **AUTHORIZATION** macareno;

- A continuación se puede especificar las **definiciones de los elementos** contenidos en dicho esquema

4.3 SQL-92

LDD: Definición de esquemas de bases de datos

- **Elementos de un esquema:**

Tablas, Vistas, Dominios, Permisos o Privilegios, Asertos, etc.

- Indicación del **esquema** al que pertenece una tabla

- Esquema Explícito

```
CREATE TABLE Empresa.Empleado  
(...);
```

- Esquema Implícito en el contexto (el activo para el usuario conectado)

```
CREATE TABLE Empleado  
(...);
```

- ❑ En **SQL de Oracle** **CREATE SCHEMA AUTHORIZATION <cuenta>**

- Sólo **UN esquema por cuenta**: <cuenta> debe coincidir con el usuario conectado
- Tiene poco o nulo efecto en la BD

4.3 SQL-92

Catálogo de base de datos relacional

- **Conjunto** nombrado de **todos los esquemas** de BD existentes en un entorno SQL
 - Contiene un **esquema especial**, **INFORMATION_SCHEMA**, que almacena **datos sobre** la definición de todos los **elementos de todos los esquemas** existentes **en el catálogo**
 - ❑ En **Oracle**, el **Diccionario de Datos** (Data Dictionary) se corresponde con el **INFORMATION_SCHEMA** del estándar SQL-92
- Es posible **compartir elementos** (dominios, etc.) **entre** diferentes **esquemas del mismo catálogo**
- Sólo pueden definirse **restricciones de integridad referencial entre tablas** que existan **en esquemas dentro del mismo catálogo**
- Concepto incorporado en la versión SQL-92 del estándar

4.3 SQL-92

LDD: Modificación de la estructura (alteración) de los elementos del esquema de base de datos

- Alteración de **tablas**:

ALTER TABLE <nombre_tabla> ... ;

- Adición y Eliminación de Columnas
- Modificación de la Definición de Columnas
- Adición y Eliminación de Restricciones de Tabla

- Alteración de **dominios**:

ALTER DOMAIN <nombre_dominio> ... ;

- Eliminación y Adición de valor por defecto
- Eliminación y Adición de Restricciones de Dominio

4.3 SQL-92

LDD: alteración de tablas

- **Adición de una columna a una tabla** ya existente

ALTER TABLE <nombre_tabla> **ADD** <definición_columna> ;

- No está permitido NOT NULL en la definición de una nueva columna (si es necesaria esta restricción, podrá establecerse después)

** Añadir una columna a EMPLEADO para contener el puesto de trabajo*

ALTER TABLE Empleado **ADD** puesto **VARCHAR(12)**;

- Todas las filas de EMPLEADO tendrán puesto a NULL
- Utilizar después órdenes UPDATE para dejar el valor correcto en cada fila

- Para introducir, en cada **fila existente**, un **valor** para la nueva columna:

- Especificar la cláusula DEFAULT al añadir la columna:

ALTER TABLE Empleado

ADD puesto **VARCHAR(12)** **DEFAULT** 'aprendiz' ;

- Utilizar después órdenes UPDATE para cambiar el valor para las filas que deban contener otro puesto de trabajo

4.3 SQL-92

LDD: alteración de tablas

- **Eliminación de una columna de una tabla**

ALTER TABLE <nombre_tabla> **DROP** <nombre_columna> <opción>;

<opción> puede ser...

RESTRICT: sólo elimina la columna si ninguna vista ni restricción le referencia

CASCADE: elimina la columna y toda restricción o vista que le hace referencia

* *Eliminación de la columna direccion de la tabla EMPLEADO*

ALTER TABLE Empleado DROP direccion CASCADE;

* *Eliminación de la columna codDep de la tabla DEPARTAMENTO*

ALTER TABLE Departamento DROP codDep RESTRICT;

– No se elimina la columna 'codDep', pues existe una columna 'EMPLEADO.dep' que le hace referencia

ALTER TABLE Departamento DROP codDep CASCADE;

– Elimina la columna junto con la restricción de integridad referencial que vincula 'EMPLEADO.dep' con DEPARTAMENTO.

👁 La columna 'EMPLEADO.dep' no es eliminada, pero deja de ser clave ajena

4.3 SQL-92

LDD: alteración de tablas

❑ Eliminación de una columna de una tabla en **SQL de Oracle**

- Eliminación de **una sola columna**

ALTER TABLE <nombre_tabla>

DROP COLUMN <columna> [CASCADE CONSTRAINTS];

- Eliminación de **una o varias columnas**

ALTER TABLE <nombre_tabla>

DROP (<columna1>, <columna2>, ...) [CASCADE CONSTRAINTS];

- La opción CASCADE CONSTRAINTS elimina la columna y toda restricción o vista que le hace referencia. Si se omite esta cláusula, se asume RESTRICT

* *Eliminación de las columnas direccion y puesto de la tabla EMPLEADO*

ALTER TABLE Empleado **DROP** (direccion, puesto);

ALTER TABLE Departamento

DROP COLUMN codDep **CASCADE CONSTRAINTS**;

- Elimina la columna junto con la restricción de integridad referencial que vincula 'EMPLEADO.dep' con DEPARTAMENTO.

👁 La columna 'EMPLEADO.dep' no es eliminada, pero deja de ser clave ajena

4.3 SQL-92

LDD: alteración de tablas

- **Modificación de la definición de una columna**

ALTER TABLE <nombre_tabla> **ALTER** <nombre_columna> <acción>;

<acción> indica la **modificación** que se desea realizar...

** Eliminación de la cláusula DEFAULT existente*

ALTER TABLE Departamento **ALTER** nssdire **DROP DEFAULT**;

** Definición de un nuevo valor por omisión*

ALTER TABLE Departamento

ALTER nssdire **SET DEFAULT** '123456789012';

❑ En **SQL de Oracle** el comando es **ALTER TABLE ... MODIFY ...**

ALTER TABLE Empleado **MODIFY** salario **NOT NULL**;

ALTER TABLE Empleado **MODIFY** apellido **VARCHAR(30)**; --antes 15

ALTER TABLE Departamento **MODIFY** nombre **UNIQUE**;

ALTER TABLE Departamento

MODIFY nssdire **DEFAULT** '123456789012';

4.3 SQL-92

LDD: alteración de tablas

- **Eliminación de una restricción de tabla**

- Se debe indicar el **nombre** de la **restricción** que se desea eliminar
 - ▶ en el CREATE TABLE debería haberse asignado un nombre a la restricción

- La restricción deja de comprobarse

ALTER TABLE <nombre_tabla>

DROP CONSTRAINT <nombre_RI> <opción>;

<opción> puede ser **CASCADE** para eliminar también otras RI dependientes

** Eliminación de la restricción de clave primaria de una tabla*

ALTER TABLE Empleado

DROP CONSTRAINT emp_pk **CASCADE**;

-- elimina la restricción de clave primaria y también las restricciones
-- de clave ajena que hacen referencia a esta clave primaria (que ya no
-- lo es). No elimina columnas en ninguna tabla

4.3 SQL-92

LDD: alteración de tablas

- **Adición de una restricción de tabla**

- La restricción se comprobará a partir del momento de su definición

```
ALTER TABLE <nombre_tabla>  
    ADD CONSTRAINT <nombre_RI> <definición_RI>;
```

** Añadir una restricción que compruebe salarios mayores que 0*

```
ALTER TABLE Empleado ADD  
    CONSTRAINT emp_salario_ok CHECK (salario > 0);
```

** Añadir una restricción que compruebe que añonacim tiene valores no nulos*

```
ALTER TABLE Empleado ADD  
    CONSTRAINT emp_añonacim_ok CHECK (añonacim IS NOT NULL);
```

4.3 SQL-92

LDD: alteración de tablas

❑ **SQL de Oracle** además, permite **Desactivar** y **Activar** restricciones

– **Inhabilitación o Desactivación** de una restricción de tabla

ALTER TABLE <nombre_tabla> **DISABLE CONSTRAINT** <nombre_RI>;

** Desactivar la RI de clave ajena en EMPLEADO que referencia a DEPARTAMENTO*

```
ALTER TABLE EMPLEADO  
DISABLE CONSTRAINT emp_jefe_ok;
```

** Inhabilitar la comprobación de que el jefe de un empleado no puede ser él mismo*

```
ALTER TABLE Empleado  
DISABLE CONSTRAINT emp_fk_dep;
```

– **Habilitación o Activación** de una restricción de tabla

ALTER TABLE <nombre_tabla> **ENABLE CONSTRAINT** <nombre_RI>;

```
ALTER TABLE Empleado  
ENABLE CONSTRAINT emp_fk_dep;
```

4.3 SQL-92

LDD: alteración de dominios

- Orden **ALTER DOMAIN** <nombre_dominio> <acción>;

<acción> indica la **modificación** que se desea realizar...

- **Eliminación y Reemplazo del valor por omisión**

ALTER DOMAIN <nombre_dominio> **DROP DEFAULT**;

ALTER DOMAIN <nombre_dominio> **SET DEFAULT** <valor>;

- **Eliminación y Definición de nuevas restricciones de dominio**

ALTER DOMAIN <nombre_dominio>

DROP CONSTRAINT <nombre_RI_dominio>;

ALTER DOMAIN <nombre_dominio>

ADD [CONSTRAINT <nombre_RI_dominio>] <restricción>;

- ❑ **Oracle** no implementa dominios, por lo que estas sentencias no funcionan

4.3 SQL-92

LDD: eliminación de elementos del esquema

- Eliminación de una **tabla**

Orden **DROP TABLE**

– Destruye una tabla base, junto con su definición en el INFORMATION_SCHEMA del catálogo

DROP TABLE <nombre_tabla> <opción>;

<opción> puede ser...

RESTRICT: Destruye la tabla sólo si no se le hace referencia desde ninguna otra tabla (clave ajena), ni es tabla base de una vista

CASCADE: Elimina la tabla junto con restricciones y vistas que la referencian

- ❑ En **SQL de Oracle** la orden de destrucción de una tabla incluye estas opciones:

DROP TABLE <nombre_tabla> ; -- opción RESTRICT

DROP TABLE <nombre_tabla> **CASCADE CONSTRAINTS**; -- opción CASCADE

4.3 SQL-92

LDD: eliminación de elementos del esquema

- Eliminación de un **dominio**

Orden **DROP DOMAIN**

- Destruye un dominio de datos, junto con su definición en el INFORMATION_SCHEMA del catálogo

DROP DOMAIN <nombre_dominio> <opción> ;

(*este aspecto es ampliado en el tema “Integridad de datos”*)

- ❑ En **SQL de Oracle** no existe la orden de destrucción de un dominio

4.3 SQL-92

LDD: eliminación de elementos del esquema

- Eliminación de un **esquema**

Orden **DROP SCHEMA**

– Destruye un esquema de BD, junto con su definición en el INFORMATION_SCHEMA del catálogo

DROP SCHEMA <nombre_esquema> <opción>;

<opción> puede ser...

RESTRICT: Destruye el esquema sólo si no contiene ningún elemento

CASCADE: Elimina el esquema y además las tablas, dominios y demás elementos contenidos en dicho esquema

- ❑ En **SQL de Oracle** no existe la orden de destrucción de un esquema

4.3 SQL-92

Lenguaje de Modificación de Datos (**LMD**): Inserción

- Orden **INSERT**

- **Añade una fila completa** a una tabla
- Incluye **nombre de la tabla** y **lista de valores para las columnas**, escritos en igual orden al especificado en la orden CREATE TABLE

INSERT INTO Empleado

VALUES ('Ricardo', 'Martínez', '653298653123', '87654321R',
'30-DIC-52', 'Olmo 98, Cedros, MX', 'M', 3700, '987654321321', 'D4');

- Si se desea poner los **valores** de las columnas **en cualquier orden**, hay que especificar los **nombres de las columnas** en dicho orden (*recomendado*)

INSERT INTO Empleado (**nombre, apellido, nss, dni, dep, salario,**
nssjefe, direccion, año nacim, sexo)

VALUES ('Ricardo', 'Martínez', '653298653123', '87654321R', 'D4',
3700, '987654321321', 'Olmo 98, Cedros, MX', '30-DIC-52', 'M');

4.3 SQL-92

LMD: Inserción

- Inserción de **varias filas** en una sola orden **INSERT**

- Filas separadas por comas. Cada fila se encierra entre paréntesis

- ❑ **Oracle** NO implementa esto: sólo permite insertar las filas **una a una**

- INSERT INTO Empleado VALUES (**

- ('Simón', 'Soriano', '111222333444', '11223344S', '22-ENE-65',
'Sauce 7, Cedros, MX', 'M', 3200, '987654321321', 'D4'),

- ('Fernanda', 'Forte', '555666777888', '55667788F', '12-AGO-58',
'Pino 5, Cedros, MX', 'F', 2560, '987654321321', 'D4'));

- Especificación explícita de **algunas columnas** (y no todas)

- **Omisión de columnas cuyo valor se desconoce**

- Cada **columna no especificada** tomará el...

- **valor por omisión:** valor tomado de su cláusula **DEFAULT**, o

- **NULL:** si la columna permite nulos y no se definió cláusula **DEFAULT** para la misma

- * Inserción de un empleado del que sólo se conoce su nombre, apellidos, nss y dni

- INSERT INTO Empleado (nombre, apellido, nss, dni)**

- VALUES ('Rubén', 'Ripoll', '553298653111', '11222333R');**

4.3 SQL-92

LMD: Inserción: Restricciones de Integridad

- Comprobación automática (SGBD) de restricciones

```
INSERT INTO Empleado (nombre, apellido, dep)
VALUES ('Cirilo', 'Hum', 'D2');
```

► **Inserción rechazada:**

no se incluye valor para nss, que está definido como NOT NULL

① Supongamos que **no** existe ningún departamento con codDep='D8'

```
INSERT INTO Empleado (nombre, apellido, nss, dni, dep)
VALUES ('Cirilo', 'Hum', '980760540222', '22333444H', 'D8');
```

► **Inserción rechazada :**


no se encuentra el valor D8 en la columna codDep de DEPARTAMENTO

- se incumple la *Integridad Referencial*

4.3 SQL-92

LMD: Inserción: **filas resultado de una consulta**

- **Carga de una tabla** con “información resumen” de la BD
 - * Sea una tabla INFO_DEP vacía. En ella queremos almacenar los nombres de cada departamento, cuántos empleados tiene y la suma de los salarios de sus empleados.
 - Una vez creada la tabla INFO_DEP (nom_dep, num_emps, suma_sal) se puede rellenar así:

```
INSERT INTO Info_dep (nom_dep, num_emps, suma_sal)
  SELECT nombre, COUNT(*), SUM(salario)
  FROM Departamento JOIN Empleado ON codDep=dep
  GROUP BY nombre ;
```
 - Por supuesto, es posible hacer SELECT ... FROM Info_depto ...
 -  INFO_DEPTO puede contener **información no actualizada**
 - Si se modifica información en EMPLEADO y/o DEPARTAMENTO, **los cambios no se reflejarán en la tabla** INFO_DEPTO
 - Una **vista** sí “contiene” siempre los datos actualizados (*se verá*)

4.3 SQL-92

LMD: Inserción

- ❑ En **SQL de Oracle**, además, es posible **crear una tabla como “copia de la estructura y contenido”** de otra tabla

CREATE TABLE <tabla> **AS SELECT ... ;**

- * Crear una tabla para almacenar datos de empleados nacidos antes de 1975

```
CREATE TABLE Emp_Veterano (nss, nombre, jefe, dep)
  AS SELECT nss, nombre||apellido, nssjefe, dep
     FROM Empleado
     WHERE año nacim < 1975;
```

4.3 SQL-92

LMD: Eliminación de datos

- Orden **DELETE**

- Elimina **filas completas** de una tabla
- **Sólo una tabla** en cláusula **FROM**
- Cláusula **WHERE** para **seleccionar las filas** que eliminar
 - Si no hay WHERE, se eliminan **todas** las filas
 - ▶ La tabla permanece, pero queda **vacía**

```
DELETE FROM Empleado; -- elimina todas las filas
```

```
DELETE FROM Empleado WHERE apellido='BOJORQUEZ';
```

```
DELETE FROM Empleado WHERE nss='123456789012';
```

```
DELETE FROM Empleado
```

```
WHERE dep IN (SELECT codDep FROM Departamento  
              WHERE nombre='INVESTIGACION');
```

- **Propagación** de eliminaciones

- Según las **acciones de mantenimiento de la Integridad Referencial** (**ON DELETE...**) especificadas con LDD en los CREATE TABLE

4.3 SQL-92

LMD: Actualización de datos

- Orden **UPDATE**

- Modifica valores de columnas en una o más filas de una tabla
- Se modifican filas de una sola tabla a la vez
- Cláusula **SET** especifica columnas que modificar y nuevos valores
- Cláusula **WHERE** para seleccionar filas que actualizar
 - Si no hay WHERE, se aplica la modificación a todas las filas

* Para el departamento D4, cambiar su nombre por FORMACION y asignarle un director

UPDATE DEPARTAMENTO

SET nombre= 'FORMACION', nssdire = '980760540222'

WHERE codDep = 'D4';

- Propagación de modificaciones

- Si cambia un valor de clave candidata, este cambio se propaga a valores de clave externa de filas de otras tablas, si así se especificó en las acciones de mantenimiento de la Integridad Referencial (**ON UPDATE...**) en la definición de la tabla con CREATE TABLE

4.3 SQL-92

LMD: Actualización

- **Modificación de varias filas** a la vez con UPDATE

—Todas las seleccionadas mediante el WHERE

* Conceder a todo empleado del departamento de Investigación un 10% de aumento de su salario

```
UPDATE Empleado SET salario = salario*1.1
    WHERE dep IN (SELECT codDep
                  FROM Departamento
                  WHERE nombre='INVESTIGACION');
```

- **NULL o DEFAULT** como **nuevo valor** de una **columna**

```
UPDATE Empleado SET salario = DEFAULT;
UPDATE Empleado SET nssjefe = NULL
    WHERE ... ;
```