

SQL: FUNCIONALIDAD BÁSICA COMO DDL

Prácticas

Manuel Ramos Cabrer

Comando Create Table

- Una relación SQL se define mediante el comando **create table**:

```
create table  $r$  ( $A_1$   $D_1$   $RI_1$ ,  $A_2$   $D_2$   $RI_2$ , ...,  $A_n$   $D_n$   $RI_n$ ,  
                (restriccion-integridad1),  
                ...,  
                (restriccion-integridadk))
```

- r es el nombre de la relación
 - cada A_i es un nombre de atributo en el esquema de la relación r
 - D_i es el tipo de datos de los valores del dominio del atributo A_i
- Ejemplo:

```
create table sucursal  
              (nombre-sucursal      char(15) not null,  
              ciudad-sucursal      char(30),  
              activos               integer)
```

Restricciones de integridad en Create Table

- **not null**
- **unique**
- **primary key** (A_1, \dots, A_n)
- **check** (P), donde P es un predicado
- **foreign key** (A_1, \dots, A_n) **references** r (B_1, \dots, B_n)

Ejemplo: Declarar *nombre-sucursal* como la clave primaria de *sucursal* y asegurar que el valor de *activos* no es negativo.

```
create table sucursal
  (nombre-sucursal char(15),
   ciudad-sucursal      char(30)
   activos              integer,
   primary key (nombre-sucursal),
   check (activos >= 0))
```

Comandos Drop y Alter Table

- El comando **drop table** borra toda la información referente a la relación eliminada de la base de datos.
- El comando **alter table** se utiliza para añadir atributos a una relación existente.

alter table r add A D

donde A es el nombre del atributo a añadir a la relación r y D es el dominio de A .

- A todas las tuplas de la relación se les asigna *null* como valor del nuevo atributo.
- El comando **alter table** también se puede utilizar para eliminar atributos de una relación

alter table r drop A

donde A es el nombre de un atributo de la relación r

- Algunos SGBD no soportan la eliminación de atributos

Tipos de dominio principales en SQL

- **char(n).** Cadena de caracteres de longitud fija n indicada por el usuario.
- **varchar(n).** Cadena de caracteres de longitud variable, con una longitud máxima n indicada por el usuario.
- **int.** Entero (un subconjunto finito de enteros dependiente de la máquina).
- **smallint.** Entero corto (un subconjunto del dominio entero cuyo tamaño es dependiente de la máquina).
- **numeric(p,d).** Número en formato de punto fijo, con una precisión indicada por el usuario de p dígitos, con n dígitos a la derecha del punto decimal.
- **real, double precision.** Números en formato de punto flotante y punto flotante de doble precisión, con precisión dependiente de la máquina.
- **float(n).** Número en punto flotante, con una precisión indicada por el usuario de al menos n dígitos.
- En todos los tipos de dominios se permiten valores nulos. Si un atributo se declara **not null**, se prohíben los valores nulos para ese atributo.
- La construcción **create domain** de SQL-92 permite crear dominios definidos por el usuario:

create domain *nombre-persona* **char(20) not null**

Tipos de Fecha/Hora en SQL (Cont.)

- **date.** Fechas, conteniendo un año (4 dígitos) , mes y día.
 - P.e. **date** '2001-7-27'
- **time.** Horas, minutos y segundos.
 - P.e. **time** '09:00:30' **time** '09:00:30.75'
- **timestamp:** Fecha y hora.
 - P.e. **timestamp** '2001-7-27 09:00:30.75'
- **Interval:** periodo de tiempo
 - P.e. **Interval** '1' día
 - Restar un valor date/time/timestamp de otro nos da un valor de tipo intervalo
 - Valores de tipo intervalo se pueden sumar a valores date/time/timestamp
- Podemos extraer valores de campos de date/time/timestamp
 - P.e. **extract (year from r.fechainicio)**
- Podemos transformar tipos cadena de caracteres a date/time/timestamp
 - P.e. **cast** <expresion-tipo-string> **as date**