

SQL

Definición de datos (DDL y DCL)

L. en C.C. Odín M. Escorza Soria

UNAM

Comando CREATE TABLE



La estructura básica es:

donde:

`<nombre>` : El nombre de la nueva tabla.

`<colspec>` : Es una especificación de columna.

`<colspec>` : Es una especificación de restricción de integridad sobre la tabla.

Al especificar los detalles de la tabla, se pueden combinar restricciones de integridad con las definiciones de las propias columnas.

Especificación de columnas



Una definición de columna debe tener al menos 2 componentes (nombre y tipo), el tercero es opcional:

`<nombre> <tipo> [<cconstr>*]`

donde:

`<nombre>` : es el nombre del atributo que estamos definiendo

`<tipo>` : es un tipo primitivo o uno definido por el usuario.

`<cconstr>` : es una restricción sobre la columna.

Ejemplos:

```
idPersona bigint primary key,  
ap1 text not null,  
sexo char(1) not null,  
idOcupacion int references ocupacion(idOcupacion)
```

Ejemplo de creación de tabla



```
create table persona (  
    idPersona bigint primary key,  
    nombre text not null,  
    ap1 text not null,  
    ap2 text not null,  
    sexo char(1) not null,  
    fnac date not null,  
    idDomicilio bigint not null  
        references domicilio(idDomicilio),  
    idOcupacion int references ocupacion(idOcupacion),  
    idEstadoCivil int references estadocivil(idestadocivil),  
    unique (nombre,ap1,ap2,sexo,fnac));
```

Tipos primitivos



Cada SDBD tiene sus propios tipos definidos, pero en la mayoría se pueden distinguir estos elementos:

- Numéricos

- ★ INT

- ★ BIGINT

- ★ FLOAT

- Sobre caracteres

- ★ CHAR

- ★ VARCHAR

- ★ TEXT

- Sobre fechas y horas

- ★ TIME

- ★ DATE

- ★ TIMESTAMP

- Lógicos (BOOLEAN)

- Auto-Incrementales

CAST



Por lo regular, los SMBD tienen “tipos fuertes”, esto es, no permiten el uso de ciertos operadores sobre tipos para los que no están definidos, por ejemplo, en PostgreSQL:

```
select 12314 || 1244
```

no se puede evaluar porque el operador de concatenación está definido sólo para el caso en el que uno de los dos argumentos es cadena.

Para forzar la evaluación podemos usar la función CAST:

```
select cast(12314 as text) || cast(1244 as text)
```

Restricciones para columnas



- **NULL:**
Especifica que la columna puede tener valores nulos (predeterminado)
- **NOT NULL:**
Evita nulidades en la columna.
- **DEFAULT <exp>:**
Establece el valor predeterminado para la columna como el resultado de evaluar <exp>
- **UNIQUE:**
Evita que existan tuplas con el mismo valor para la columna en la tabla.
- **PRIMARY KEY:**
Establece la columna como llave primaria para la tabla.
- **REFERENCES <tbl> (<col>):**
Establece el atributo como llave externa que apunta a la columna <col> de la tabla <tbl>.
- **CHECK (<exp>):**

Restricciones para tablas



- **UNIQUE** ($\langle \text{atr}_1 \rangle, \langle \text{atr}_2 \rangle, \dots$):
Evita que dos o más tuplas tengan el mismo valor en los atributos listados.
- **PRIMARY KEY** ($\langle \text{atr}_1 \rangle, \langle \text{atr}_2 \rangle, \dots$):
Establece una llave primaria compuesta.
- **FOREIGN KEY** ($\langle \text{atr}_1 \rangle, \langle \text{atr}_2 \rangle, \dots$) **REFERENCES** $\langle \text{tbl} \rangle$
($\langle \text{col}_1 \rangle, \langle \text{col}_2 \rangle, \dots$):
Establece los atributos ($\langle \text{atr}_1 \rangle, \langle \text{atr}_2 \rangle, \dots$) como llave externa apuntando a los correspondientes de la tabla $\langle \text{tbl} \rangle$.
- **CHECK** $\langle \text{exp} \rangle$
Verifica que la evaluación de $\langle \text{exp} \rangle$ sea verdadero ante cualquier actualización de datos.

Motivación



En ocasiones es necesario modificar tablas que ya están implementadas y puestas en marcha. Analizaremos las siguientes modificaciones:

- Eliminar tablas.
- Argerar columnas o restricciones a una tabla.
- Eliminar columnas o restricciones sobre una tabla.

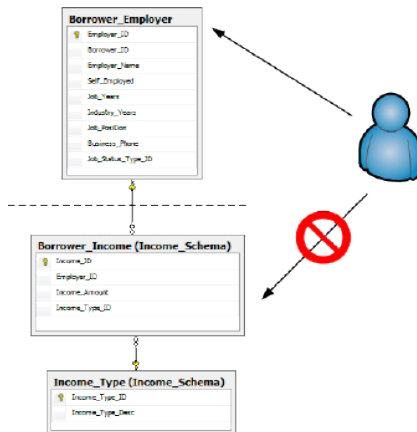
Estructura general



- `DROP TABLE <t>`
Elimina la tabla <t> de la BD.
- `ALTER TABLE <t> ADD COLUMN <ncol> <colspec>`
Agrega la columna con nombre <ncol> (cuya especificación es <colspec>) a la tabla <t>.
- `ALTER TABLE <t> ADD <constraintspec>`
Agrega la restricción especificada por <constraintspec> a la tabla <t>
- `ALTER TABLE <t> DROP COLUMN <ncol>`
Elimina la columna llamada <ncol> de la tabla <t>
- `ALTER TABLE <t> DROP CONSTRAINT <ncons>`
Elimina la restricción con nombre <ncons> de la tabla <t>

Motivación

- Los SDBD son siempre multi-usuario.
- En un sistema multiusuario, no todos tienen acceso a todo.



Gestion de usuarios



- `CREATE USER <nu>`
Crea un nuevo usuario con nombre <nu> dentro del SMBD.
- `DROP USER <ou>`
Elimina del SMBD al usuario de nombre <ou>

Componentes

Para llevar a cabo el control, se toman en cuenta 3 factores

- Sujeto (<us>): Cada uno de los usuarios registrados en el SMBD.
- Acción (<act>):

- ★ USAGE
- ★ SELECT
- ★ INSERT
- ★ DELETE

- ★ UPDATE
- ★ CREATE
- ★ EXECUTE
- ★ ALL

- Objeto (<obj>):

- ★ DATABASE
- ★ TABLE
- ★ PROCEDURE

- ★ VIEW
- ★ FUNCTION
- ★ SEQUENCE



Control de Acceso (GRANT/REVOKE)



```
GRANT <act>, ... ON <obj> TO <us>
```

```
REVOKE <act>, ... ON <obj> TO <us>
```