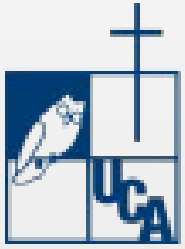
Artilugios Avanzados I

Administración de Bases de Datos



Departamento de electrónica e informática

Carlos Juan Martín Pérez

Artilugios avanzados I

Tipos de datos personalizados

Además de los tipos de datos "base" en los SGBD se pueden definir tipos de datos adaptados a necesidades específicas. Los tipos de datos creados se asocian al esquema indicado (public por defecto) y pertenecen al usuario que los crea, que deberá hacer GRANT USAGE a otros usuarios si así lo desea.

Una primera acción es crear dominios sobre tipos de datos base, sus dimensiones y restricciones particulares de fila (CHECK, NOT NULL) con el fin de no tener que describirlos cada vez que se usen. CREATE DOMAIN. Más información: link.

Además se pueden crear tipos completamente nuevos. Más información: link

Tipos de datos compuestos: sirve para crear una estructura / registro. Link.

Tipos de datos enumerados: permite crear una lista estática y ordenada de valores. Link.

Tipos de datos array: crea matrices multidimensionales de cualquier tipo (base o creado). Link.

También rangos (subrangos de tipos base) o incluso tipos base nuevos.

Herencia

Herencia: concepto de orientación a objetos que es utilizado en algunos SGBD (llamados objeto-relacionales por tal motivo), como PostgreSQL. Esto se efectúa para formar una jerarquía de tablas (útil por tanto en traducción de especializaciones) mediante el uso de la palabra INHERITS. (más info: link)

- **SÍ se heredan todas las columnas de la/s tabla/s padre, así como las restricciones de columna CHECK y NOT NULL, salvo que se exprese lo contrario en cláusulas NO INHERIT.**
- **NO se heredan rest. UNIQUE, claves primarias, ni foráneas.**
- **ALTER TABLE sí propaga los cambios a las tablas hijas.**
- **NO se puede eliminar una tabla padre mientras exista una hija, salvo que se use CASCADE.**
- **SÍ se puede heredar de varias tablas padre, resultando una unión de columnas: si hubiera varias con el mismo nombre se espera que tengan el mismo tipo de datos y restricciones o dará un error.**
- **SELECT (y UPDATE/DELETE) de una tabla padre también consulta en las hijas automáticamente (salvo que se use ONLY).**
- **Tristemente, INSERT solo inserta en la tabla expresada: puede producir datos duplicados.**

Artilugios avanzados I

Particionamiento

Particionar una tabla significa dividirla en pedazos sin perder la capacidad de consulta total y sin preocuparnos por tener que hacer múltiples JOIN). Esto tiene sentido cuando tenemos tablas masivas que rara vez se consultan por completo: p.e. el tamaño de los índices/datos es menor y así pueden caber en RAM, o considerando la posibilidad de ubicación de particiones en diferentes Tablespaces, se podrína ubicar los datos menos consultados en disco más barato. También es posible acelerar inserción/borrado en masa agregando/eliminando particiones. (más información: link). PARTITION BY { RANGE | LIST | HASH }

Particionamiento por rango: La tabla es dividida en función de el valor de una columna (o columnas) de tal modo que no haya solapamiento en dicho valor. Por ejemplo: rangos de fechas.

Particionamiento por lista: La tabla se divide listando explícitamente los valores que estarán en cada partición.

Particionamiento por hash: La tabla se particiona especificando un módulo (número de tablas). Cada partición tendrá las filas para las que el valor hash de la clave de partición, dividido por el módulo definido, produzca el resto que le corresponda.