INSTITUTO KHIPU

Semestre IV

Sesión 17

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS

Tema

LENGUAJE ESTRUCTURADO SQL

LENGUAJE ESTRUCTURADO SQL

1. Definición de la base de datos: SQL DDL

Los comandos más importantes del lenguaje de definición de datos (DDL) SQL son los siguientes:

CREATE TABLE

CREATE INDEX

ALTER TABLE

RENAME TABLE

DROP TABLE

DROP INDEX

Estos enunciados se usan para crear, cambiar y destruir las estructuras lógicas que constituyen el modelo lógico. Estos comandos se pueden usar en cualquier momento para realizar cambios a la estructura de la base de datos. Existen comandos adicionales disponibles para especificar detalles físicos de almacenamiento, pero no se les discutirá aquí, pues son específicos al sistema. Se aplicarán estos comandos al siguiente ejemplo, que se usó en capítulos anteriores:

Student (stuId, lastName, firstName, major, credits)

Faculty (facId, name, department, rank)

Class (classNumber, facId, schedule, room)

Enroll (classNumber, stuId, grade)

1.1. Create Table (crear tabla)

Este comando se usa para crear las tablas base que forman el corazón de una base de datos relacional. Dado que se puede usar en cualquier momento durante el ciclo de vida del sistema, el desarrollador de la base de datos puede comenzar con un pequeño número de tablas y agregarlas conforme se planeen y desarrollen

aplicaciones adicionales. Una tabla base está bastante cercana a la noción abstracta de una tabla relacional. Consiste de uno o más encabezados (headings) de columna, que proporcionan el nombre de columna y tipo de datos, y cero o más filas de datos, que contienen un valor de datos del tipo de datos especificado para cada una de las columnas.

CREATE TABLE nombre tabla base (nombre_col tipo_dato [restricciones columna - NULL/NOT NULL, DEFAULT . . . , UNIQUE, CHECK . . . , PRIMARY KEY . . .]] [,nombre_col tipo_dato [restricciones columna] . . . [restricciones tabla - PRIMARY KEY . . . , FOREIGN KEY . . . , UNIQUE . . . , CHECK . . .] [especificaciones almacenamiento]);

Aquí, nombre tabla base es un nombre proporcionado por el usuario para la tabla. No se pueden usar palabras clave SQL, y el nombre de la tabla debe ser único dentro de la base de datos. Para cada columna, el usuario debe especificar un nombre que sea único dentro de la tabla, y un tipo de datos. La sección opcional de especificaciones de almacenamiento del comando CREATE TABLE.

1.1.1. Tipos de datos

Los tipos de datos disponibles incluyen varios tipos numéricos, cadenas de caracteres de longitud fija y de longitud variable, cadenas de bits y tipos definidos por el usuario. Los tipos de datos disponibles varían de DBMS a DBMS. Por ejemplo, los tipos más comunes en Oracle son CHAR(N), VARCHAR2(N), NUMBER(N,D), DATE y BLOB (gran objeto binario). En DB2, los tipos incluyen SMALLINT, INTEGER, BIGINT, DECIMAL/NUMERIC, REAL, DOUBLE, CHAR(N), VARCHAR(N), LONG VARCHAR, CLOB, GRAPHIC, DBCLOB, BLOB, DATE, TIME y TIMESTAMP. Microsoft SQL Server incluye NUMERIC, BINARY, CHAR, VARCHAR, DATE-TIME, MONEY, IMAGE y otros. Microsoft Access soporta varios tipos de NUMBER, así como TEXT, MEMO, DATE/TIME, CURRENCY, YES/NO y otros. Además, algunos sistemas como Oracle permiten a los usuarios crear nuevos dominios, y construir sobre los tipos de datos existentes. En lugar de usar uno de

los tipos de datos disponibles, los usuarios pueden especificar dominios por adelantado y pueden incluir una condición de verificación para el dominio. SQL:1999 permite la creación de nuevos tipos de datos distintos usando uno de los tipos anteriormente definidos como el tipo fuente. Por ejemplo, se podría escribir:

CREATE DOMAIN creditValues INTEGER DEFAULT 0 CHECK (VALUE >=0 AND VALUE =0 AND credits < 150).

1.1.2. Restricciones (constraints) de columna y tabla

El sistema de gestión de base de datos tiene facilidades para reforzar la exactitud de los datos, las que debe usar el ABD cuando cree tablas. Recuerde que el modelo relacional usa restricciones de integridad para proteger la exactitud de la base de datos y permitir la creación sólo de instancias legales. Estas restricciones protegen el sistema de errores en entrada de datos que crearían datos inconsistentes. Aunque el nombre de la tabla, los nombres de columna y los tipos de datos son las únicas partes requeridas en el comando CREATE TABLE, se pueden y deben agregar restricciones opcionales, tanto a nivel columna como a nivel tabla. Las restricciones de columna incluyen opciones para especificar NULL/NOT NULL, UNIQUE, PRIMARY KEY, CHECK y DEFAULT para cualquier columna, inmediatamente después de la especificación del nombre de columna y el tipo de datos. Si no se especifica NOT NULL, el sistema permitirá que la columna tenga valores nulos, lo que significa que el usuario puede insertar registros que no tengan valores para dichos campos.

```
CREATE TABLE Student (
stuId CHAR(6),
lastName CHAR(20) NOT NULL,
firstName CHAR(20) NOT NULL,
major CHAR(10),
credits SMALLINT DEFAULT 0,
CONSTRAINT Student_stuId_pk PRIMARY KEY (stuId),
CONSTRAINT Student_credits_cc CHECK (CREDITS>=0 AND credits < 150);
```

```
CREATE TABLE Faculty (
facId CHAR(6),
name CHAR(20) NOT NULL,
department CHAR(20) NOT NULL,
rank CHAR(10),
CONSTRAINT Faculty_facId_pk PRIMARY KEY (facId));
```

```
CREATE TABLE Class (
classNumber CHAR(8),
facId CHAR(6) NOT NULL,
schedule CHAR(8),
room CHAR(6),
CONSTRAINT Class_classNumber_pk PRIMARY KEY (classNumber),
CONSTRAINT Class_facId_fk FOREIGN KEY (facId) REFERENCES Faculty (facId));
```

```
CREATE TABLE Enroll (
stuId CHAR(6),
classNumber CHAR(8),
grade CHAR(2),
CONSTRAINT Enroll_classNumber_stuId_pk PRIMARY KEY (classNumber, stuId),
CONSTRAINT Enroll_classNumber_fk FOREIGN KEY (classNumber) REFERENCES Class (classNumber),
CONSTRAINT Enroll_stuId_fk FOREIGN KEY (stuId) REFERENCES Student (stuId) ON DELETE CASCADE);
```

1.1.3. ALTER TABLE, RENAME TABLE

Una vez creada una tabla, los usuarios pueden encontrarla más útil si contiene un ítem de datos adicional, no tiene una columna particular o tiene diferentes restricciones. Aquí, la naturaleza dinámica de una estructura de base de datos relacional hace posible cambiar las tablas base existentes. Por ejemplo, para

agregar una nueva columna a la derecha de la tabla, use un comando de la forma:

ALTER TABLE nombre tabla base ADD nombre_col tipo_dato;

Note que no se puede usar la especificación NULL para la columna. Un comando ALTER TABLE ...ADD (alterar tabla ...agregar) hace que el nuevo campo se agregue a todos los registros ya almacenados en la tabla, y los valores nulos se asignen a dicho campo en todos los registros existentes. ALTER TABLE nombre tabla base DROP COLUMN nombre columna; Para eliminar la columna cTitle y regresar a la estructura original para la tabla Class, se escribiría: ALTER TABLE Class DROP COLUMN cTitle; Si quiere agregar, eliminar o cambiar una restricción, puede usar el mismo comando ALTER TABLE. Por ejemplo, si creó la tabla Class y despreció hacer facId una clave externa en Class, podría agregar la restricción en cualquier momento al escribir: ALTER TABLE Class ADD CONSTRAINT Class_facId_fk FOREIGN KEY (facId) REFERENCES Faculty (facId)); Podría eliminar una restricción nombrada existente al usar el comando ALTER TABLE. Por ejemplo, para eliminar la condición check (verificar) en el atributo credits de Student que se creó antes, podría escribir: ALTER TABLE Student DROP CONSTRAINT Student_credits_cc; Puede cambiar fácilmente el nombre de una tabla existente mediante el comando: RENAME TABLE nombre tabla anterior TO nombre tabla nueva;

2. Manipulación de la base de datos: DML SQL

El lenguaje de consulta de SQL es declarativo, también llamado no procedural, lo que significa que permite especificar cuáles datos se recuperan sin dar los procedimientos para recuperarlos. Se puede usar como un lenguaje interactivo para consultas, incrustado en un lenguaje de programación huésped, o como un lenguaje completo en sí para cálculos con el uso de SQL/PSM (Persistent Stored Modules = módulos de almacenamiento persistentes).

Los enunciados DML SQL son: SELECT UPDATE INSERT DELETE

2.1. Introducción al enunciado SELECT

El enunciado SELECT se usa para recuperar datos. Es un comando poderoso, que realiza el equivalente de SELECT, PROJECT y JOIN del álgebra relacional, así como otras funciones, en un solo enunciado simple. La forma general de SELECT es SELECT [DISTINCT] nombre col [AS nombre nuevo], [,nombre col..] . . . FROM nombre tabla [alias] [,nombre tabla] . . . [WHERE predicado] [GROUP BY nombre col [,nombre col]

. . . [HAVING predicado] o [ORDER BY nombre col [,nombre col] . . .]; El resultado es una tabla que puede tener filas duplicadas. Dado que los duplicados se permiten en tal tabla, no es una relación en el sentido estricto, pero se le conoce como multiconjunto o bolsa. Como se indica por la ausencia de corchetes se requieren las cláusulas SELECT y FROM, pero no WHERE u otras cláusulas.

- **Ejemplo 1. Recuperación simple con condición**

Pregunta: Obtener nombres, identificaciones y número de créditos de todos los que tienen especialidad Math.

Solución: La información solicitada aparece en la tabla Student. Para dicha tabla se seleccionan sólo las filas que tienen un valor de 'Math' para major. Para dichas filas, sólo se despliegan las columnas lastName, firstName, stuId y credits. Note que se hace el equivalente SELECT (al encontrar las filas) y PROJECT (al desplegar sólo ciertas columnas) del álgebra relacional. También se reordenan las columnas.

Note que el resultado de la consulta es una tabla o un multiconjunto.

- Ejemplo 2. Uso de notación asterisco para "todas las columnas"

Pregunta: Obtener toda la información acerca de Faculty CSC.

Solución: Se quiere todo el registro Faculty de cualquier miembro del personal docente cuyo departamento sea 'CSC'. Dado que muchas recuperaciones SQL requieren todas las columnas de una sola tabla, existe una forma corta de expresar.

SELECT * FROM Nombre de la Tabla

Bibliografía

- Bases de Datos- Catherine M. Ricardo

Bibliografía:

- https://openwebinars.net/blog/como-realizar-la-normalizacion-de-bases-de-datos-y-por-que/
- https://www.grch.com.ar/docs/bd/materia/11077/Normalizacion_ejercicios.pdf

INSTITUTO

KHIPU