Laboratorio IV - SQL

Restricciones de clave foránea

Select - Agrupamiento de registros

Ing. Gregorio Nicolás Tkachuk

Objetivos

- Que el estudiante logre:
  - Aplicar restricciones de clave foránea
  - Habilidad para realizar consultas utilizando funciones de agregación.

DDL: Contenidos

- Create Table
- Restricción de integridad
- Valor por defecto (DEFAULT)
- Restricción de Columna
- Restricción de clave primaria
- Restricción de clave foránea

- DML
  - Funciones de Agregación

SQL – CREATE TABLE

```
CREATE TABLE <nombre_tabla>(
<nombre_columna-1> < tipo>
      [<valor por defecto>] [<restricciones de
 columna>],
<nombre_columna-2>  <tipo>
      [<valor por defecto>] [<restricciones de
 columna>],
…
<nombre_columna-n>  <tipo>
   [<valor por defecto>] [<restricciones de columna>],
   [<restricciones de tabla>],
….
   [<restricciones de tabla>]
);
```

SQL – Restricciones de Clave Foránea

FOREIGN KEY (*<columnas>*)
REFERENCES *<nombre_tabla>*[(*<nombre_columna>*)]
[ON DELETE {RESTRICT | CASCADE | SETNULL | NOACTION }]
[ON UPDATE {RESTRICT | CASCADE | SETNULL | NOACTION }]

ON DELETE: Restricción de clave foránea que se ejecuta cuando se elimina un registro de la tabla padre.

ON UPDATE : Restricción de clave foránea que se ejecuta cuando se modifica un registro de la tabla padre.

SQL – Restricciones de Clave Foranea. (Cont.)

RESTRICT: Rechaza la operación de eliminación o actualización en la tabla padre. Opción por defecto.

SET NULL: Borra o actualiza el registro en la tabla padre y establece en NULL la o las columnas de clave foránea en la tabla hija. Esto solamente es válido si las columnas de clave foránea no han sido definidas como NOT NULL.

CASCADE: Borra o actualiza el registro en la tabla padre y automáticamente borra o actualiza los registros coincidentes en la tabla hija.

SQL – Ejemplo de relación 1 a N

Cliente

CodCli (PK)	Nombre
1	PEREZ, MARTIN
2	PAZ, MONICA
3	GARCIA, DIEGO
4	LOPEZ, MAGALI

Telefono

CodTel (PK)	Numero	CodCli (FK)
1	0385 - 4318569	**3**
2	0385 - 154096897	**4**
3	0385 - 4213333	**4**
4	0385 - 4225588	**2**

SQL – Restricciones de Clave Foránea, Ejemplo

```
CREATE TABLE cliente(
codCli   int  NOT NULL,
Nombre varchar(30)  NOT NULL,
PRIMARY KEY(codCli)      );

CREATE TABLE telefono (
codTel  int  NOT NULL,
numero  varchar(30)  NOT NULL,
codCli   int,
PRIMARY KEY(codTel),
FOREIGN KEY(codCli)  REFERENCES  cliente(codCli)
ON DELETE CASCADE );
```

Preguntas ?

Funciones de Agregación

- Las funciones de agregación permiten obtener un solo valor a partir de un conjunto de tuplas.
- A excepción de la función COUNT(*), todas las funciones de agregación ignoran los valores NULL.
- Las funciones de agregación se suelen utilizar con la cláusula GROUP BY de la instrucción SELECT.

Funciones de Agregación

Función	*Descripción*
AVG	Promedio de valores en una expresión numérica
COUNT	Número de valores en una expresión
COUNT (*)	Número de filas seleccionadas, incluye Nulos
MAX	Valor más alto en la expresión
MIN	Valor más bajo en la expresión
SUM	Valores totales en una expresión numérica
STD(expr*), STDDEV(expr)*	Desviación estadística de todos los valores

SELECT Sintaxis

```
SELECT columna, funcion_de_agregacion(columna)
    FROM tabla
    [WHERE <condicion>]
    [GROUP BY <lista_de_atributos>]
        [HAVING <condicion_de_grupo>];
```

SELECT Cláusula GROUP BY

- La cláusula **GROUP BY <lista_de_atributos>**
  permite agrupar las tuplas en grupos que tengan los mismos valores en todos los atributos de esa lista devolviendo una única tupla por grupo.
- Los atributos de la cláusula GRUOP BY deben aparecer en la lista de atributos del SELECT.

Ejemplo: Obtener la cantidad de productos vendidos.

```
SELECT productoid, SUM(cantidad)
    FROM ventas
    GROUP BY productoid;
```

SELECT Cláusula GROUP BY

```
SELECT productoid, SUM(cantidad)
    FROM ventas
    GROUP BY productoid;
```

productoid	ordenid	cantidad
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30

productoid	Sum_str
1	15
2	35
3	45

GROUP BY con cláusula HAVING

- La cláusula **HAVING <condicion_de_grupo>**

  permite establecer una condición sobre los grupos de manera que sólo se seleccionan aquellos grupos que la cumplen.

Ejemplo: Obtener los productos cuya venta sea mayor que 30.

```
SELECT productoid, SUM(cantidad)
    FROM ventas
    GROUP BY productoid
    HAVING SUM(cantidad ) > 30;
```

Ejemplo

```
USE facturacion;
SELECT productoid, ordenid
      ,cantidad
 FROM ventas;
```

```
SELECT productoid, SUM(cantidad)
   AS cantidad_total
 FROM ventas
 GROUP BY productoid
 HAVING SUM(cantidad)>30;
```

productoid	ordenid	cantidad
1	1	5
1	1	10
2	1	10
2	2	25
3	1	15
3	2	30

productid	Cantidad_total
2	35
3	45

Preguntas ?