Fundamentos de Bases de Datos.
Práctica 11.

Profesor: M.I. Gerardo Avilés Rosas
gar@ciencias.unam.mx
Laboratorio: Luis Eduardo Castro Omaña
lalo_ castro@ciencias.unam.mx

8 de mayo de 2017

Se dan a conocer especificaciones de entrega para la práctica 11.

1. Consultas

La estructura básica de una expresión SQL consiste en tres cláusulas: SELECT, FROM y WHERE.

- La cláusula SELECT corresponde a la operación proyección del álgebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.
- La cláusula FROM corresponde a la operación producto cartesiano del álgebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión.
- La cláusula WHERE corresponde al predicado selección del álgebra relacional. Es un predicado que engloba a los atributos de las relaciones que aparecen en la cláusula FROM.

1.1. Clausula SELECT

El resultado de una consulta SQL es una relación. SQL permite duplicados en las relaciones, así como en el resultado de las expresiones SQL. En aquellos casos donde se quiera forzar la eliminación de duplicados, se insertará la palabra clave DISTINC después de SELECT.

El símbolo asterisco «*» se puede usar para denotar «todos los atributos». Así, el uso de Tabla.* indicaría que todos los atributos de Tabla serían seleccionados. Una cláusula SELECT de la forma SELECT * indica que se deben

seleccionar todos los atributos de todas las relaciones que aparecen en la cláusula FROM.

La cláusula SELECT puede contener también expresiones aritméticas que contengan los operadores, +, -, * y / operando sobre constantes o atributos de la tuplas.

1.2. Clausula WHERE

SQL usa las conectivas lógicas AND, OR y NOT (en lugar de los símbolos matemáticos $\wedge$,$\vee$ y $\neg$) en la cláusula WHERE. Los operandos de las conectivas lógicas pueden ser expresiones que contengan los operadores de comparación <, <=, >, >=, = y <>. SQL permite usar los operadores de comparación para comparar cadenas y expresiones aritméticas, así como tipos especiales, tales como el tipo fecha.

SQL incluye un operador de comparación BETWEEN para simplificar las cláusulas WHERE que especifica que un valor sea menor o igual que un valor y mayor o igual que otro valor.

1.3. Clausula FROM

La cláusula from define por sí misma un producto cartesiano de las relaciones que aparecen en la cláusula. Escribir una expresión SQL para la reunión natural es una tarea relativamente fácil, puesto que la reunión natural se define en términos de un producto cartesiano, una selección y una proyección.

1.4. Renombramiento

SQL proporciona un mecanismo para renombrar tanto relaciones como atributos. Para ello utiliza la cláusula AS, que tiene la forma siguiente: nombre-antiguo as nombre-nuevo la cláusula AS puede aparecer tanto en SELECT como en FROM.

La cláusula as es particularmente útil en la definición del concepto de variables tupla, como se hace en el cálculo relacional de tuplas. Una variable tupla en SQL se debe asociar con una relación concreta. Las variables tupla se definen en la cláusula FROM mediante el uso de la cláusula AS.

Las variables tupla son de gran utilidad para comparar dos tuplas de la misma relación. Hay que recordar que, en los casos de este tipo, se puede usar la operación renombramiento del álgebra relacional.

1.5. Operaciones sobre cadenas

La operación más usada sobre cadenas es el encaje de patrones, para el que se usa el operador LIKE. Para la descripción de patrones se utilizan los dos caracteres especiales siguientes:

- Tanto por ciento (%): El carácter % encaja con cualquier subcadena.
- Subrayado (_): El carácter _ encaja con cualquier carácter.

Los patrones son muy sensibles, esto es, los caracteres en mayúsculas no encajan con los caracteres en minúscula, o viceversa.

Para que los patrones puedan contener los caracteres especiales patrón (esto es, % y _), SQL permite la especificación de un carácter de escape. El carácter de ESCAPE se utiliza inmediatamente antes de un carácter especial patrón para indicar que ese carácter especial va a ser tratado como un carácter normal. El carácter de escape para una comparación LIKE se define utilizando la palabra clave escape.

SQL permite buscar discordancias en lugar de concordancias utilizando el operador de comparación not like.

SQL también proporciona una variedad de funciones que operan sobre cadenas de caracteres, tales como la concatenación (usando «||»), la extracción de subcadenas, el cálculo de la longitud de las cadenas, la conversión a mayúsculas y minúsculas, etc.

1.6. Orden

SQL ofrece al usuario cierto control sobre el orden en el cual se presentan las tuplas de una relación. La cláusula order by hace que las tuplas resultantes de una consulta se presenten en un cierto orden. De manera predeterminada la cláusula order by lista los elementos en orden ascendente. Para especificar el tipo de ordenación se puede incluir la cláusula desc para orden descendente o asc para orden ascendente. Además, se puede ordenar con respecto a más de un atributo.

1.7. Funciones de agregación

Las funciones de agregación son funciones que toman una colección (un conjunto o multiconjunto) de valores como entrada y producen un único valor como salida. SQL proporciona cinco funciones de agregación primitivas:

- Media: AVG
- Mínimo: MIN

- Máximo: MAX
- Total: SUM
- Cuenta: COUNT

La entrada a SUM y AVG debe ser una colección de números, pero los otros operadores pueden operar sobre colecciones de datos de tipo no numérico, tales como las cadenas.

Existen situaciones en las cuales sería deseable aplicar las funciones de agregación no sólo a un único conjunto de tuplas sino también a un grupo de conjuntos de tuplas; esto se especifica en SQL usando la cláusula GROUP BY. El atributo o atributos especificados en la cláusula GROUP BY se usan para formar grupos. Las tuplas con el mismo valor en todos los atributos especificados en la cláusula GROUP BY se colocan en un grupo.

A veces es más útil establecer una condición que se aplique a los grupos que una que se aplique a las tuplas.

Para expresar este tipo de consultas se utiliza la cláusula HAVING de SQL. Los predicados de la cláusula HAVING se aplican después de la formación de grupos, de modo que se pueden usar las funciones de agregación.

SQL no permite el uso de DISTINCT con COUNT(*). Sí se permite, sin embargo, el uso de DISTINCT con MAX y MIN, incluso cuando el resultado no cambia. Se puede usar la palabra clave all en lugar de DISTINCT para especificar la retención de duplicados, pero como ALL se especifica de manera predeterminada, no es necesario incluir dicha cláusula.

Si en una misma consulta aparece una cláusula WHERE y una cláusula HAVING, se aplica primero el predicado de la cláusula WHERE. Las tuplas que satisfagan el predicado de la cláusula WHERE se colocan en grupos según la cláusula GROUP BY. La cláusula HAVING, si existe, se aplica entonces a cada grupo; los grupos que no satisfagan el predicado de la cláusula HAVING se eliminan. La cláusula SELECT utiliza los grupos restantes para generar las tuplas resultado de la consulta.

1.8. Valores nulos

SQL permite el uso de valores nulos para indicar la ausencia de información sobre el valor de un atributo. En un predicado se puede usar la palabra clave especial IS NULL para comprobar si un valor es nulo. El predicado IS NOT NULL pregunta por la ausencia de un valor nulo.

1.9. Subconsulta anidadas

SQL proporciona un mecanismo para las subconsultas anidadas. Una subconsulta es una expresión SELECT-FROM-WHERE que se anida dentro de otra consulta. Un uso común de subconsultas es llevar a cabo comprobaciones sobre pertenencia a conjuntos, comparación de conjuntos y cardinalidad de conjuntos. Las subconsultas posibles son las siguientes:

- Pertenencia a conjuntos: SQL utiliza el cálculo relacional para las operaciones que permiten comprobar la pertenencia de una tupla a una relación. La conectiva in comprueba la pertenencia a un conjunto, donde el conjunto es la colección de valores resultado de una cláusula select. La conectiva not in comprueba la no pertenencia a un conjunto.

2. Actividad

Utilizando la base de datos que hayan poblado deberán escribir las siguientes consultas.

1. Conocer el nombre de todos los chóferes que tengan mas de 25 años.
2. Conocer el nombre y edad de todos los alumnos que hayan realizado mas de un viaje con la asociación.
3. Conocer los vehículos con mas de diez años de antigüedad.
4. Conocer los dueños de los vehículos con mas de diez años de antigüedad.
5. Todos los viajes que hayan costado mas de $100, así como el chófer que lo realizó, el dueño del automóvil y el usuario que lo hizo.
6. El promedio de edad de los chóferes que hayan ingresado a la asociación entre el 2000 y 2016.
7. Saber las personas que son dueños y chóferes al mismo tiempo.
8. Conocer el total que gasta al mes cada uno de los académicos en viajes.
9. Conocer las multas que se le hayan aplicado a los automóviles que no tengan seguro.
10. Conocer los choferes que se les haya levantado una multa en la delegación Benito Juárez, Coyoacán y Tlalpan.
11. El nombre de los choferes que su seguro no cubre daños a terceros.
12. El nombre de los usuarios que han realizado viajes con mas de 100 km de distancia después de las 6 p.m.
13. Conocer el total de multas cometidas por delegación.

14. Saber cual es la delegación en la que se comenten el mayor número de multas.
15. Nombre de los dueños que tienen mas de un chofer.
16. Conocer el número de clientes por edad.
17. Conocer el total de horas y ganancias han tenido los choferes.
18. Saber el promedio que gastan en viajes en un día alumnos y académicos.
19. Conocer el chofer con mas multas por delegación.
20. Cual es el nombre mas común entre los usuarios de la aplicación.

No es posible modificar la estructura de su base de datos para obtener el resultado de las consultas, ya que, ya se había hecho la modificación correspondiente en la práctica de álgebra relacional.

3. Entregables

Se debe entregar un script llamado consultas.sql el cual tendrá las instrucciones correspondientes para poder visualizar las consultas.

La entrega deberá ser el día martes 15 de Mayo de 2017.