Aulas práticas - Ficha 8

Bases de Dados (CC2005), Dep. Ciência de Computadores, FCUP

Eduardo R. B. Marques, DCC/FCUP

Objectivos: resolução de exercícios versando vistas SQL e "persistent stored modules" no servidor Mooshak.

Referências:

- Vistas SQL
- SQL PSM ("Persistent Stored Modules")

1

Abaixo é fornecido o código da "stored function" **getChargeValue** discutida nos slides 6 a 11 das aulas téoricas, apenas **modificada** para suportar um parâmetro extra **customer_id**.

```
DROP FUNCTION IF EXISTS getChargeValue;
DELIMITER $

CREATE FUNCTION
getChargeValue(stream_time DATETIME, movie_id INT, customer_id INT)
RETURNS DECIMAL(4,2)
BEGIN
  DECLARE c DECIMAL(4,2);
  DECLARE movie_duration INT;
  DECLARE country_name VARCHAR(128);
  DECLARE region_name VARCHAR(128);

  SELECT Duration INTO movie_duration
  FROM MOVIE WHERE MovieId = movie_id;

  SET c = 0.5 + 0.01 * movie_duration;

  IF HOUR(stream_time) >= 21 THEN
    SET c = c + 0.75;
    IF WEEKDAY(stream_time) >= 4 THEN
      SET c = c + 0.75;
    END IF;
  END IF;

  RETURN c;
END $
```

Estenda a lógica de cálculo do valor a cobrar pelo "stream" considerando que a este, além das condições já tidas em conta, acresce **1** euro se o cliente for originário dos Estados Unidos (**COUNTRY.Name = 'United States'**) ou da Europa (**REGION.Name = 'Europe'**). Note que no código acima já se encontram definidas variáveis **country_name** e **region_name** para guardar o nome do país e o nome da região de um cliente.

Caso defina o código correctamente os resultados da seguinte consulta:

```
SELECT
   CustomerId, Name, Country,
   getChargeValue('2019-04-11 20:59:59', MovieId, CustomerId) AS V1,
   getChargeValue('2019-04-11 21:00:00', MovieId, CustomerId) AS V2,
   getChargeValue('2019-04-12 21:00:00', MovieId, CustomerId) AS V3
FROM MOVIE, CUSTOMER
WHERE Title='Pulp Fiction'
AND
(
   CustomerId = (SELECT CustomerId FROM CUSTOMER WHERE Country='Brazil' ORDER BY Name LIMIT
OR
   CustomerId = (SELECT CustomerId FROM CUSTOMER WHERE Country='Germany' ORDER BY Name LIMI
OR
   CustomerId = (SELECT CustomerId FROM CUSTOMER WHERE Country='United States' ORDER BY Name
)
ORDER BY Country;
```

deverão ser:

```
+------------+---------------+---------------+------+------+------+
| CustomerId | Name          | Country       | V1   | V2   | V3   |
+------------+---------------+---------------+------+------+------+
|        398 | Antonio Meek  | Brazil        | 2.04 | 2.79 | 3.54 |
|        196 | Alma Austin   | Germany       | 3.04 | 3.79 | 4.54 |
|         51 | Alice Stewart | United States | 3.04 | 3.79 | 4.54 |
+------------+---------------+---------------+------+------+------+
```

Nota: Caso tenha um erro do tipo:

```
ERROR 1418 (HY000): This function has none of DETERMINISTIC, NO SQL, or READS SQL DATA in it
declaration and binary logging is enabled (you *might*  want to use the less safe log_bin_tr
```

configure a opção `log_bin_trust_function_creators` usando o seguinte comando na consola mysql:

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

2

Crie uma vista ("view") chamada **STREAM_COUNT_BY_COUNTRY** com campos **Country** e **StreamCount** para obter os nomes de países de clientes e o correspondente nº de streams feitos por clientes para cada país, incluindo países cujos clientes não tenha feito qualquer "stream". Os dados da vista não precisam de estar ordenados por nenhum critério.

Para criar a vista use código com a seguinte forma:

```
DROP VIEW STREAM_COUNT_BY_COUNTRY;

CREATE VIEW STREAM_COUNT_BY_COUNTRY(Country,StreamCount)
AS
SELECT ... ;
```

Após criada a vista, para a consulta (que é usada pelo avaliador automático):

```
SELECT * FROM STREAM_COUNT_BY_COUNTRY ORDER BY StreamCount DESC, Country;
```

deverá obter os resultados ilustrados pelo seguinte fragmento:

```
+---------------------------------+-------------+
| Country                         | StreamCount |
+---------------------------------+-------------+
| China                           |        1016 |
| India                           |         985 |
| Nigeria                         |         514 |
| Brazil                          |         511 |
| Japan                           |         478 |
| United States                   |         470 |
...
| Virgin Islands                  |           1 |
| Zambia                          |           1 |
| French Guiana                   |           0 |
| Greenland                       |           0 |
| Nepal                           |           0 |
| Turkmenistan                    |           0 |
+---------------------------------+-------------+
```

Nota: Irá obter como resposta **Compile Time Error** se os nomes dos campos da vista não forem os esperados (**Country** e **StreamCount**).

3

Crie uma vista ("view") chamada **DEPARTMENT_STAFF** com campos **DName**, **SName**, e **Job** para obter os dados do nome de um departamento (**DName** de **DEPARTMENT.Name**), nome de funcionário (**SName** de **STAFF.Name**) e cargo do funcionário (**Job** de **STAFF.job**) tal que o funcionário em causa seja ou o gerente do departamento (**DEPARTMENT.Manager**) ou supervisionado (directamente) pelo gerente do departamento. Os dados da vista não precisam de estar ordenados por nenhum critério.

Para criar a vista use código com a seguinte forma:

```
DROP VIEW DEPARTMENT_STAFF;

CREATE VIEW DEPARTMENT_STAFF(DName,SName,Job)
AS
SELECT ... ;
```

Após criada a vista, para a consulta (que é usada pelo avaliador automático)

```
SELECT * FROM DEPARTMENT_STAFF ORDER BY DName, Job, SName;
```

deverá obter

```
+------------------+-------------------+---------------------------+
| DName            | SName             | Job                       |
+------------------+-------------------+---------------------------+
| Customer         | António Mota      | Customer Manager Director |
| Customer         | Felícia Antunes   | Region Manager            |
| Customer         | Gabriela Silva    | Region Manager            |
| Customer         | Gastão Pinto      | Region Manager            |
| Finance          | Alexandra Romeu   | Accountant                |
| Finance          | Fábio Cruz        | Accountant                |
| Finance          | Augusto Sousa     | Finance Manager           |
| IT               | Filipa Mendes     | Database Administrator    |
| IT               | Xavier Semedo     | IT Manager                |
| IT               | Eva Mendes        | Lead Programmer           |
| Public Relations | Filipa Magalhães  | Customer PR               |
| Public Relations | João Santorini    | Industry PR               |
| Public Relations | José Santos       | PR Manager                |
+------------------+-------------------+---------------------------+
13 rows in set (0.00 sec)
```

Nota: Irá obter como resposta **Compile Time Error** se os nomes dos campos da vista não forem os esperados (**DName**,**SName**, **Job**).

4

Escreva uma "stored function" **getDepartment** com o esqueleto dado abaixo, tal que para um dado id **staff_id** de um funcionário seja obtido o identificador do seu departamento, determinado pelo facto de o funcionário ser o gestor do departamento (**DEPARTMENT.Manager**) ou por ser supervisionado directamente pelo gestor do departamento. Caso o funcionário não esteja associado a um departamento por este critério, a "stored function" deve retornar **NULL**.

```
DROP FUNCTION IF EXISTS getDepartment;
DELIMITER $
CREATE FUNCTION getDepartment(staff_id INT)
RETURNS INT
BEGIN
  DECLARE dep_id INT;
  SET dep_id = NULL;
  ...
  RETURN dep_id;
END $
DELIMITER ;
```

Depois de criada a "stored function" **getDepartment** o resultado esperado da seguinte consulta (que é usada pelo avaliador automático):

```
SELECT StaffId, getDepartment(StaffId) AS DepId
FROM STAFF
ORDER BY DepId, StaffId;
```

deverá ser:

```
+---------+-------+
| StaffId | DepId |
+---------+-------+
|       1 |  NULL |
|       5 |  NULL |
|       6 |  NULL |
|      10 |  NULL |
|       2 |     1 |
|       3 |     1 |
|       4 |     1 |
|       7 |     2 |
|       8 |     2 |
|       9 |     2 |
|      11 |     3 |
|      12 |     3 |
|      13 |     3 |
|      14 |     4 |
|      15 |     4 |
|      16 |     4 |
|      17 |     4 |
+---------+-------+
17 rows in set (0.00 sec)
```