Introduction to Databases

Lecture #10

Views

Ljubov Jaanuska
(ljubov.jaanuska@ut.ee)
Lecture 9. Summary

- Tools for graphical models
  - DBDesigner.net

- Project requirements
Lecture 10. What will you learn

- Home assignment: task 10
- Views
- Revision on queries
Task 10

- Who has the same amount of losses and the wins? Show the first name and the surname.
There are two types of tables in an SQL-database:

- **base tables** – (the tables we have discussed thus far): the rows are physically stored on the disks of the database, and they obtain their data through the insert, update and delete operations.

- **derived tables - views** – tables are not stored anywhere but exist only as specifications and used in queries like base tables.
Views

A view is like a “virtual” table

• defined by a query, which describes how to compute the view contents on the fly

• DBMS stores the view definition query instead of view contents

• Can be used in queries just like a regular table
Syntax

**Syntax**

**Format**: `create view view_name [(column_list)] as query`

**Example**:

```
CREATE VIEW v_club54 AS
SELECT * FROM Player WHERE club_id=54

CREATE VIEW v_club54 (p_name, p_surname) AS
SELECT p_name, surname FROM Player WHERE c_id=54
```

To drop a view:

```
DROP VIEW view_name;
```
v_player

Merge two tables: Club and Player

CREATE VIEW v_player (club_name, club_id, player_name, player_id) AS SELECT Club.c_name, Club.c_id, Player.p_name || ' ' || Player.surname, Player.p_id FROM Player JOIN Club ON Player.club_id = Club.c_id
Using views in queries

Show players’ ids, names and club names using view v_player.

```sql
SELECT player_id, player_name, club_name FROM v_player
ORDER BY 1 ASC
```
Creating views using base tables and views

Show players’ names and club names in the context of the game. To simplify the solution, use view v_player.

CREATE VIEW v_game (g_id, t_id, begins, w_name, w_club, w_points, b_name, b_club, b_points) AS

SELECT g.game_id, g.tournament_id, g.begins, w.player_name, w.club_name, g.white_result/ 2.0, b.player_name, b.club_name, g.black_result/ 2.0
FROM Game as g, v_player as w, v_player as b
WHERE g.white=w.player_id AND g.black=b.player_id;
Why views?

The main reasons for the use of views are

• To hide data from users
• To hide complexity from users
• Data independence
  • it is possible to alter the structure of the base tables quite radically without changing the queries and without changing the programs (used in application)
• better security control
  • no insert, update, delete statements on views made of multiple tables

• Real database applications use tons of views
Revision on queries (self-study/group work)

1. Which player has won the most games?
2. Which club is the most successful?
3. What is an average number of points scored by each club?
4. Which club scored the most points?
5. How many clubs competed in each tournaments?
6. Which district provides the most players?
7. How many people are there with the same first names?
8. How many players are in each club?
9. Which name, surname beginning letter is the most common?
10. Show the ID's of the players when the result of the game was 2:1.
Home Tasks

1. Create a view v_tournament_game (t_id, g_id, g_beginning, g_end) where t_id is the tournament id, g_id is the game id, g_beginning and g_end are the date and time when the game began and ended.
2. Create a view v_club_game (club_name, count_games), where count_games is the number of games played by a club member. If both players of the game are from the same club, then it counts as two games.
3. Create a view v_point (g_id, t_id, p_id, color, points):

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41</td>
<td>73</td>
<td>W</td>
<td>0.50</td>
</tr>
<tr>
<td>1</td>
<td>41</td>
<td>92</td>
<td>B</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>71</td>
<td>B</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>77</td>
<td>W</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>77</td>
<td>B</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**NB**: the result in the picture is not correct!

**NB!** Define color as B if the player played with black chess pieces and W if the player played with white chess pieces.

**NB!** Use **UNION!**
Home Tasks

4. Create a view v_rating (p_name, t_id, points):
   p_name is a name of the player obtained from v_player. player_name;
   t_id is the id of the tournament obtained from v_point.t_id;
   points are summed for each player and each tournament separately.

<table>
<thead>
<tr>
<th>Name</th>
<th>Points</th>
<th>Total Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jelena Pirm</td>
<td>42</td>
<td>21.00</td>
</tr>
<tr>
<td>Artur Muld</td>
<td>42</td>
<td>19.50</td>
</tr>
<tr>
<td>Maria Murakas</td>
<td>42</td>
<td>18.00</td>
</tr>
<tr>
<td>Toomas Umnik</td>
<td>42</td>
<td>16.50</td>
</tr>
<tr>
<td>Mihkel Maakamar</td>
<td>42</td>
<td>16.50</td>
</tr>
</tbody>
</table>

NB: the points in the picture are not correct!

5. Save the query as a view:
Which name, surname beginning letter is the most common?
   Show letter @, where @ is the most common capital letter among all
   names and surnames.

PLEASE SUBMIT via Moodle by 18.04 23:55
• .db file and .log file
Anonymous feedback

Please fill in an anonymous mid-term feedback by 16th of April:
https://goo.gl/forms/1TXenNUTKgKoNZwh2

Each answer is important!