Practical Class 08:

**SQL: DML and Advanced Constructs**
Insert, Update, Delete, View, Index, Procedure, Transaction, Trigger

**Martin Svoboda**
11. 4. 2017

Faculty of Electrical Engineering, Czech Technical University in Prague
Exercises

• Assume we have the following database schema

CREATE TABLE accounts (  
    ida INT PRIMARY KEY,  
    number VARCHAR(22) NOT NULL UNIQUE,  
    owner VARCHAR(100) NOT NULL,  
    city VARCHAR(50) NOT NULL,  
    balance DECIMAL(15, 2) NOT NULL DEFAULT 0  
);  

CREATE TABLE transfers (  
    idt BIGINT PRIMARY KEY,  
    datetime TIMESTAMP NOT NULL,  
    source INT REFERENCES accounts (ida) ON DELETE SET NULL,  
    target INT REFERENCES accounts (ida) ON DELETE SET NULL,  
    amount DECIMAL(15, 2) NOT NULL  
);
Insertions

- **INSERT INTO** command
Exercise 1

• **Insert two new bank accounts** into our database
  - Use only one insert statement
  - **Account A:**
    - Identifier: 501, number: 123456789/1111
    - Owner: Martin Svoboda, city: Liberec
  - **Account B:**
    - Identifier: 502, number: 101010101/1111
    - Owner: Irena Mlynkova, city: Praha
Updates

- **UPDATE** command
Exercise 2

• **Update details of a certain bank account**
  - Change attributes of an account with identifier 502
    - New owner: *Irena Holubova*
    - New city: *Praha*

• **Add interests to selected accounts**
  - Only owners from *Liberec* will be rewarded
  - Interest rate equals to *1%*
Deletions

- **DELETE** command

```sql
DELETE FROM table_name
WHERE expression
```
Exercise 3

• Remove a particular bank account
  ▪ Delete a bank account with number 101010101/1111
    – What the impact will be for the following data?

<table>
<thead>
<tr>
<th>ida</th>
<th>number</th>
<th>owner</th>
<th>city</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>501</td>
<td>123456789/1111</td>
<td>Martin Svoboda</td>
<td>Liberec</td>
<td>100000.00</td>
</tr>
<tr>
<td>502</td>
<td>101010101/1111</td>
<td>Irena Holubova</td>
<td>Praha</td>
<td>200000.00</td>
</tr>
</tbody>
</table>

• Remove all bank accounts
Views

- **CREATE VIEW** command

- **CASCADED** is the default for CHECK OPTION
Exercise 4

• Create a view on a table of accounts
  ▪ Select all accounts such that...
    – their owners are from Liberec
    – their current balance is at most 50000.00
  ▪ Include only the following columns
    – Identifier, number, owner and city
    – I.e. not balance
Exercise 5

- Attempt to insert two new bank accounts into the previous view
  
  - Account C:
    - Identifier: 503, number: 111222333/1111
    - Owner: *Jiri Helmich*, city: *Liberec*
  
  - Account D:
    - Identifier: 504, number: 444555666/1111
    - Owner: *Martin Necasky*, city: *Jicin*

- Consider different view updateability options
Query Plans

- **EXPLAIN** command
Exercise 6

• Express the following select query
  - Bank accounts of clients from Liberec with current balance below the overall average
  - Include all the original columns and calculate also the overall number of outgoing transfers for each such account

• Analyze the query evaluation plan
Indexes

- **CREATE INDEX** command
Exercise 7

• **Create an index** on a table of accounts
  ▪ Construct this index such that it helps us with the effective evaluation of the previous query

• **Analyze the query evaluation plan** once again
Stored Procedures

- **CREATE FUNCTION** command

- Arguments accessible via $1, $2, ... when not named
Exercise 8

- **Create a stored procedure for bank transfers**
  - Input arguments
    - Transfer identifier, source/target accounts, amount
  - Actions
    - Both accounts will be tested for their existence
    - The source account will be tested for sufficient balance
    - Balances of both the accounts will be updated
    - The transfer will be logged into the table of transfers
      - Current time will be used as a transfer timestamp

- **Execute this procedure** for a sample transfer
Transactions

- **BEGIN, COMMIT** and **ROLLBACK** commands
Exercise 9

- Execute the previous procedure as a transaction
  - I.e. encapsulate its call into a transaction
Triggers

- **CREATE TRIGGER** command
Exercise 10

• **Create a new trigger** that allows us to check validity of account balances
  ▪ Invoke this trigger to check impact of all INSERT and UPDATE operations
  ▪ Access old/new values via OLD/NEW records