B0B36DBS, BD6B36DBS: **Database Systems**

http://www.ksi.mff.cuni.cz/~svoboda/courses/192-B0B36DBS/

Practical Class 8

SQL: Advanced Constructs

Author: **Martin Svoboda**, martin.svoboda@fel.cvut.cz
Tutors: **Ahmad**, **Černoch**, **Kostov**, **Kouba**, **Řimnáč**, **Svoboda**, **Šír**

7. 4. 2020

Czech Technical University in Prague, Faculty of Electrical Engineering

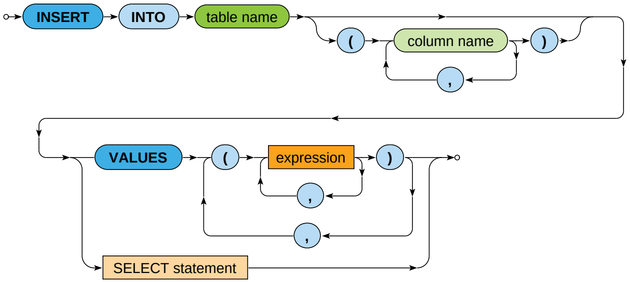
Database Schema

Assume we have the following schema of a relational database for a simple **bank information system**

```
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 statement



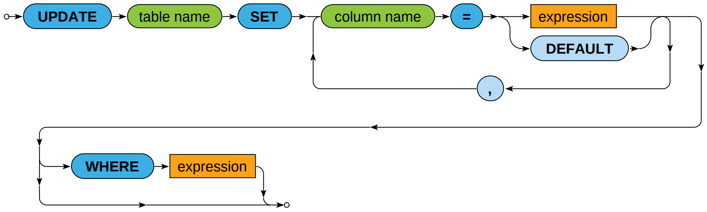
Exercise 1

Insert two new bank accounts into our database

- Account *501*
  - Number: *123456789/1111*
  - Owner: *Martin Svoboda*
  - City: *Liberec*
- Account *502*
  - Number: *101010101/1111*
  - Owner: *Irena Mlynkova*
  - City: *Praha*
- Use only one insert statement

Updates

UPDATE statement



Exercise 2

Update details of a particular bank account

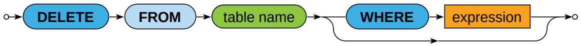
- Change attributes of an account with identifier *502*
  - New owner: *Irena Holubova*
  - New city: *Praha*

Add interests to selected accounts

- Only owners from *Liberec* will be rewarded
- Interest rate equals to *1%*

Deletions

DELETE statement



Exercise 3

Remove a particular bank account

- Delete a bank account with number *101010101/1111*
  - What will be the impact on the following snippet of data?

ida	number	owner	city	balance
501	123456789/1111	Martin Svoboda	Liberec	10000.00
502	101010101/1111	Irena Holubova	Praha	20000.00

idt	datetime	source	target	amount
1000034	2017-01-15 14:30:00	600	502	5000.00
1000035	2017-01-15 14:45:00	502	700	1000.00

Remove all bank accounts

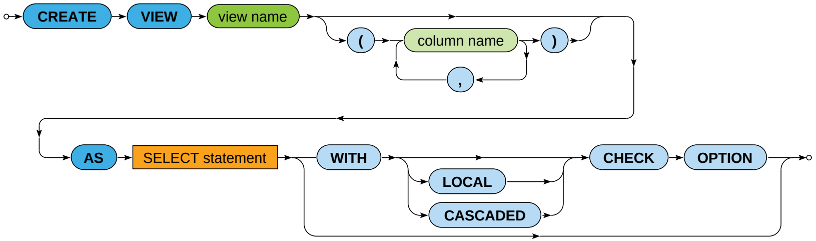
Sample Data

Assume we have the following sample data in our database

```
INSERT INTO accounts VALUES
    (501, '123456789/1111', 'Martin Svoboda', 'Liberec', 15000.00),
    (502, '101010101/1111', 'Irena Holubova', 'Praha', 20000.00),
    (503, '111222333/1111', 'Jiri Helmich', 'Liberec', 5000.00),
    (504, '444555666/1111', 'Martin Necasky', 'Jicin', 15000.00),
    (505, '777888999/1111', 'Marek Polak', 'Praha', 5000.00);
INSERT INTO transfers VALUES
    (10000034, '2017-01-15 14:30:00', 501, 502, 5000.00),
    (10000035, '2017-01-15 14:40:00', 502, 503, 1000.00),
    (10000036, '2017-01-15 14:50:00', 503, 504, 2000.00),
    (10000037, '2017-01-15 15:00:00', 503, 505, 3000.00),
    (10000038, '2017-01-15 15:10:00', 501, 502, 1000.00),
    (10000039, '2017-01-15 15:20:00', 501, 504, 5000.00);
```

Views

CREATE VIEW statement



- Options
  - **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 least *10000.00*
- Preserve all the original columns

Exercise 5

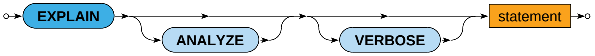
Attempt to insert two new bank accounts into the previous view

- Account *506*
  - Number: *999888777/1111*
  - Owner: *Jakub Klimek*
  - City: *Liberec*
  - Balance: *5000*
- Account *507*
  - Number: *666555444/1111*
  - Owner: *Jakub Lokoc*
  - City: *Brno*
  - Balance: *15000*

Consider different view updateability options

Evaluation Plans

EXPLAIN statement



- Options
  - **ANALYZE**
    - Executes a given statement and shows actual run times and other statistics
  - **VERBOSE**
    - Displays additional information regarding the evaluation plan

Exercise 6

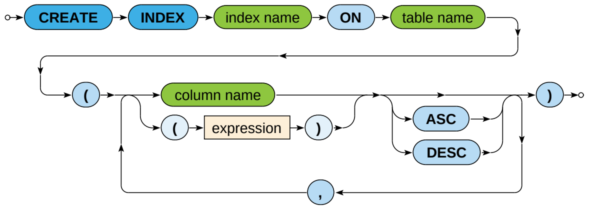
Express the following select query

- Bank accounts of clients from *Liberec* with current balance below the overall average
- Include all the original columns, calculate the overall number of outgoing transfers for each such account

Analyze the query evaluation plan

Index Structures

CREATE INDEX statement



Exercise 7

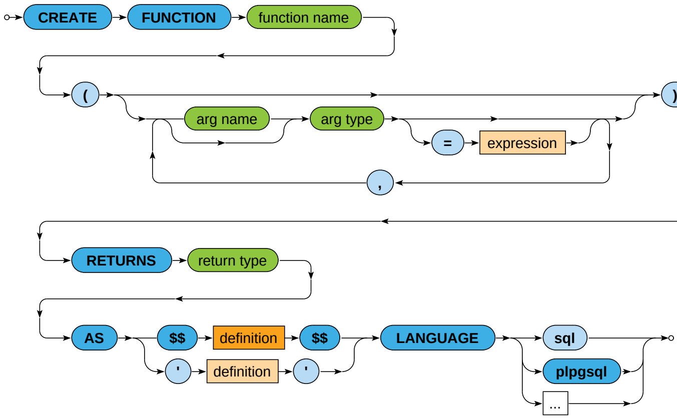
Create an index on a table of accounts

- 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 statement



- Arguments accessible via $1, $2, … when not named explicitly

Exercise 8

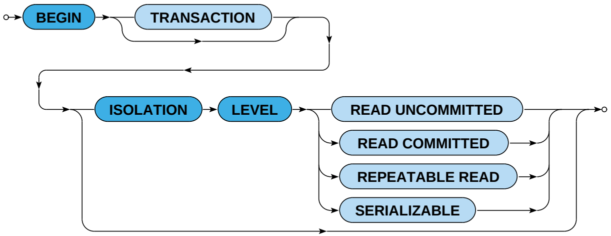
Create a stored procedure for bank transfers

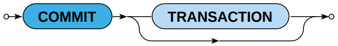
- Input
  - Transfer identifier
  - Source / target accounts
  - Amount
- Actions
  - Both accounts will be tested for their existence
  - Sufficient balance of the source account will be checked
  - Balances of both the accounts will be updated
  - The transfer will be logged into the table of transfers
  - The current time will be used as a transfer timestamp

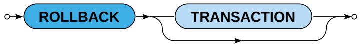
Execute this procedure for a sample transfer

Transactions

BEGIN, **COMMIT**, and **ROLLBACK** commands







- By default, individual statements are executed in *autocommit* mode unless encapsulated by an explicit transaction

Transactions

Isolation levels

- **READ UNCOMMITTED**
  - The lowest isolation level
  - Treated as READ COMMITTED in PostgreSQL
- **READ COMMITTED**
  - Only rows committed before a given statement can be seen
  - The default isolation level in PostgreSQL
- **REPEATABLE READ**
  - Only rows committed before the first statement can be seen
- **SERIALIZABLE**
  - Execution of transactions is guaranteed to be serializable
  - The highest isolation level

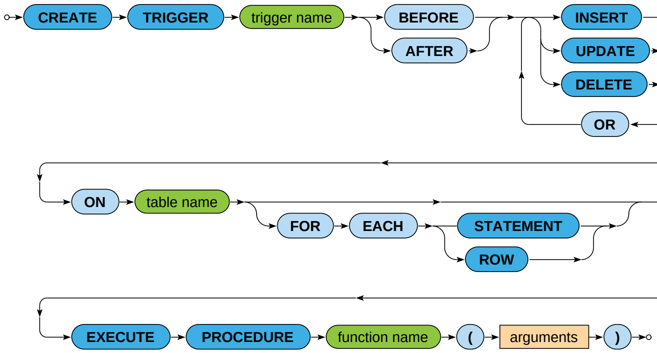
Exercise 9

Execute the previous procedure as a transaction

- I.e. encapsulate its call into a transaction
- Choose an appropriate isolation level

Triggers

CREATE TRIGGER statement



- Options
  - **FOR EACH STATEMENT** is the default mode

Exercise 10

Create a new trigger that allows us to check validity of account balances

- Invoke this trigger in a way that you will be able to check the impact of all `INSERT` and `UPDATE` operations
- Access old / new values via `OLD` / `NEW` records