**Database Systems Overview Exercises**

# What database models do you know?

Common logical data models for databases include:

* Hierarchical database model
* [Network model](http://en.wikipedia.org/wiki/Network_model)
* [Relational model](http://en.wikipedia.org/wiki/Relational_model)
* [Entity–relationship model](http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)
* [Enhanced entity–relationship model](http://en.wikipedia.org/wiki/Enhanced_entity%E2%80%93relationship_model)
* [Object model](http://en.wikipedia.org/wiki/Object_database)
* [Document model](http://en.wikipedia.org/wiki/Document-oriented_database)
* [Entity–attribute–value model](http://en.wikipedia.org/wiki/Entity%E2%80%93attribute%E2%80%93value_model)
* [Star schema](http://en.wikipedia.org/wiki/Star_schema)

# Which are the main functions performed by a Relational Database Management System (RDBMS)?

Relational Database Management Systems (RDBMS) manage data stored in tables:

* Creating / altering / deleting tables and relationships between them (database schema)
* Adding, changing, deleting, searching and retrieving of data stored in the tables
* Support for the SQL language
* Transaction management (optional)

# Define what is "table" in database terms

Database tables consist of data, arranged in rows and columns. All rows have the same structure, Columns have name and type (number, string, date, image, or other).

# Explain the difference between a primary and a foreign key.

* Primary key is a column of the table that uniquely identifies its rows (usually it is a number), also the primary key can be composed by several columns (composite primary key)
* The foreign key is an identifier of a record located in another table (usually its primary key)

# Explain the different kinds of relationships between tables in relational databases.

Relationships between tables are based on interconnections: primary key / foreign key. By using them we avoid repeating data in the database.

Relationships have multiplicity:

* One-to-many – country / towns
* Many-to-many – student / course
* One-to-one – human / student

# When is a certain database schema normalized? What are the advantages of normalized databases?

Normalization of the relational schema removes repeating data.

# What are database integrity constraints and when are they used?

Integrity constraints ensure data integrity in the database tables - enforce data rules which cannot be violated.

Primary key constraint - ensures that the primary key of a table has unique value for each table row;

Unique key constraint - ensures that all values in a certain column (or a group of columns) are unique;

Foreign key constraint - ensures that the value in given column is a key from another table;

Check constraint - ensures that values in a certain column meet some predefined condition.

# Point out the pros and cons of using indexes in a database.

Indices speed up searching of values in a certain column or group of columns - usually implemented as B-trees. Indices can be built-in the table (clustered) or stored externally (non-clustered).

Adding and deleting records in indexed tables is slower! Indices should be used for big tables only (e.g. 50 000 rows).

# What's the main purpose of the SQL language?

SQL (Structured Query Language) - standardized declarative language for manipulation of relational databases.

SQL language supports:

* Creating, altering, deleting tables and other objects in the database
* Searching, retrieving, inserting, modifying and deleting table data (rows)

# What are transactions used for? Give an example.

Transactions are a sequence of database operations which are executed as a single unit:

* Either all of them execute successfully
* Or none of them is executed at all

Example:

A bank transfer from one account into another (withdrawal + deposit). If either the withdrawal or the deposit fails the entire operation should be cancelled.

# What is a NoSQL database?

NoSQL is non-relational database.

* Data is stored as documents
* Single entity (document) is a single record
* Documents do not have a fixed structure

# Explain the classical non-relational data models.

Document model - set of documents, e.g. JSON strings;

Key-value model - set of key-value pairs;

Hierarchical key-value - hierarchy of key-value pairs;

Wide-column model - key-value model with schema;

Object model - set of OOP-style objects.

# Give few examples of NoSQL databases and their pros and cons.

Redis - ultra-fast in-memory data structures server;

MongoDB - mature and powerful JSON-document database;

CouchDB - JSON-based document database with REST API;

Cassandra - distributed wide-column database