Relational Databases

Fundamental concepts

Contents

1. Database models
2. Relational database models
3. RDBMS systems
4. Tables, relations, relations multiplicity , E/R diagrams
5. Normalization
6. Constraints
7. Indexes

Contents

8. The SQL language
9. Stored procedures
10. Views
11. Triggers

Relational databases

- Database models
  - Hierarchical (tree)
  - Network (graph-oriented)
  - Relational (table)
  - Object-oriented (document-oriented)

Relational databases

They represent a bunch of tables together with the relations between them.

They maintain the *integrity* of the data.

They have a strong mathematical foundation: relational algebra.

Relational Database Management System (RDBMS)

- Provide program means for:
  - Creating / altering / deleting tables and relations between them
  - Adding, changing, deleting, searching and retrieving of data from the tables
  - SQL support
  - Transaction management (optional)

Relational Database Management System (RDBMS)

- RDBMS systems are also known as:
  - Database management servers
  - or simpler still "Database servers"
- Famous RDBMS servers:
  - Oracle Database
  - Microsoft SQL Server
  - IBM DB2
  - PostgreSQL
  - MySQL
  - Borland Interbase

Tables

- Tables are compilations of values, ordered in rows and columns. For example (table `People`):

Id	Name	Family	Employer
1	Krasimir	Stoev	ITTalents
2	Rozaliya	Dimitrova	Imperia Online
3	Rositsa	Tsvetanova	ITTalents

- All rows have the same structure
- Columns have a name and a type (number, string, data or other)

Table Scheme

- The scheme of a table is an ordered sequence of descriptions of columns (name and type)
- For example the `People` table has the following scheme:

```
PEOPLE (
   id: number,
   name: string,
   family: string,
   employer: string
)
```

Primary Key

- Primary key is a column of the table, that can be used to uniquely identify one of its rows

Primary key

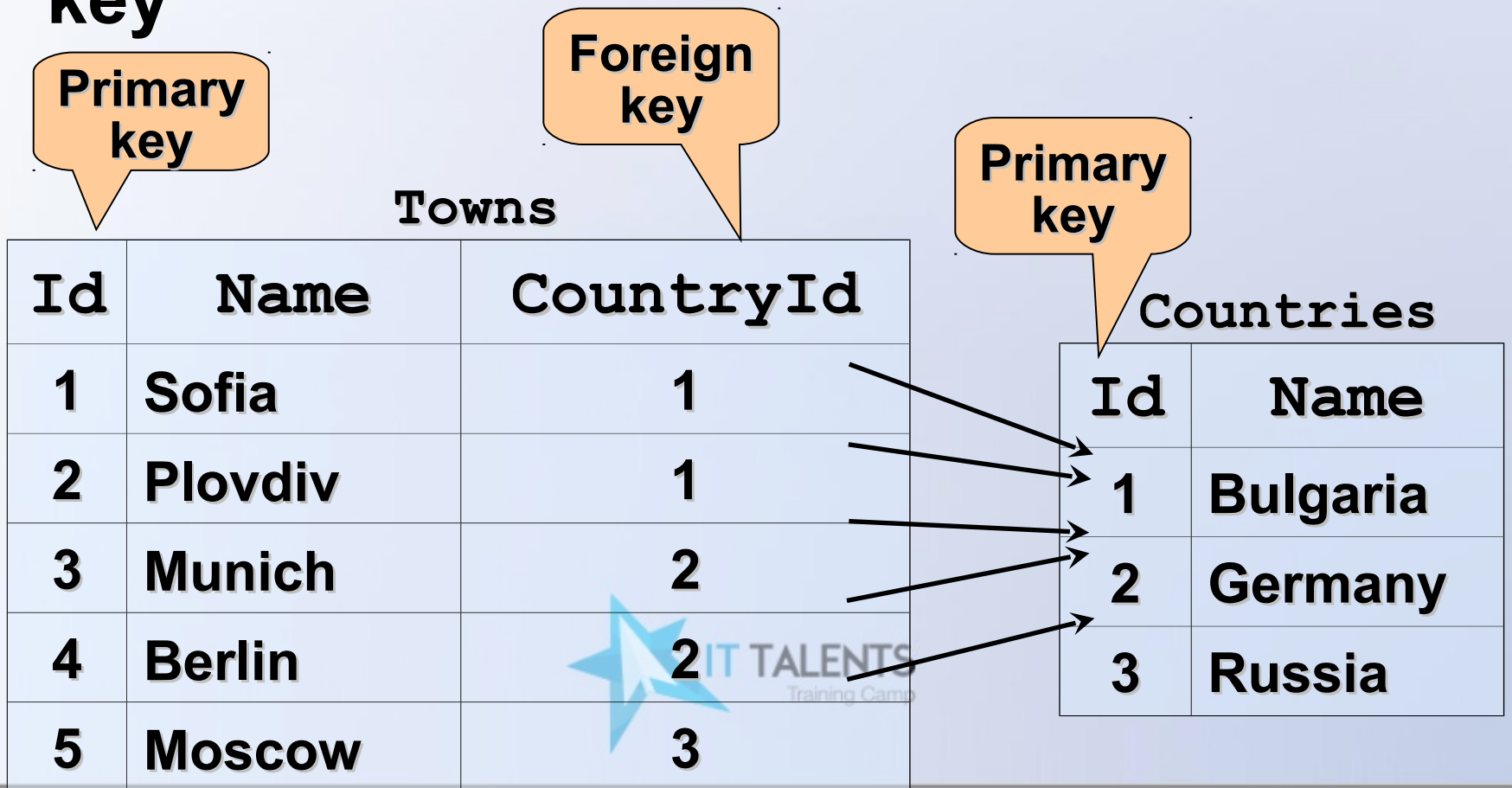
Id	Name	Family	Employer
1	Nikolay	Tomitov	ITTalents
2	Lyubomir	Slavilov	Imperia Online
3	Rositsa	Tsvetanova	ITTalents

Primary Key

- Two records (rows) are different only when their primary keys are different
- The primary key can be composed from several columns (composite primary key)
- Thus, primery key is always non-null and unique

Relations

- Relations between tables are based on interconnections primary key / foreign key



Relations

- The foreign key (most of the time) is a number of a record (primary key) in another table
- By using relations we avoid repeating information in our database
  - In the example the name of the country is not repeated for every town
- Relations have multiplicity :
  - 1 x many – country / towns
  - many x many – student / course
  - 1 x 1 – for example human / student

Relations' multiplicity

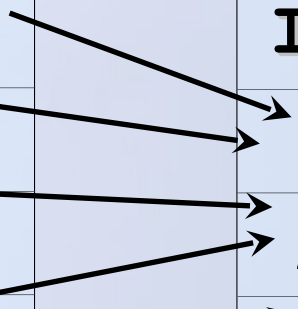
- A relation 1 x many (or many x 1)
  - 1 record in the first table has many corresponding records in the second one. Used very often

Towns

Id	Name	CountryId
1	Sofia	1
2	Plovdiv	1
3	Munich	2
4	Berlin	2
5	Moscow	3

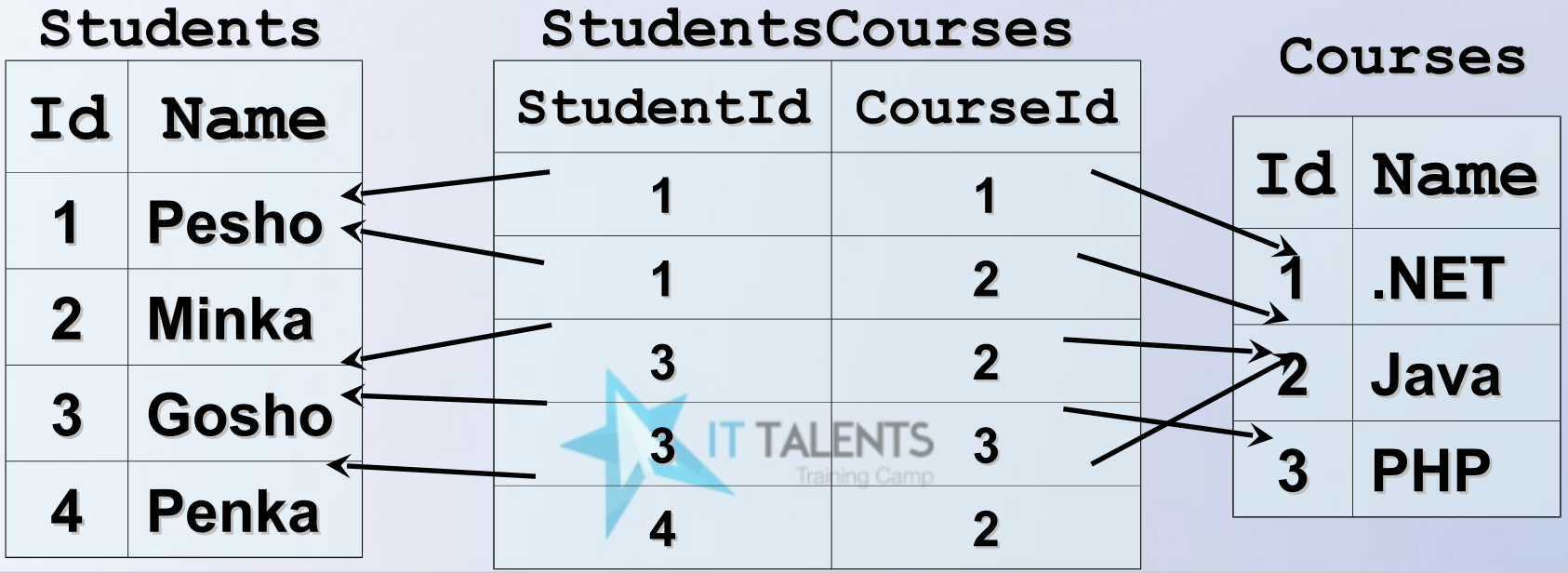
Countries

Id	Name
1	Bulgaria
2	Germany
3	Russia



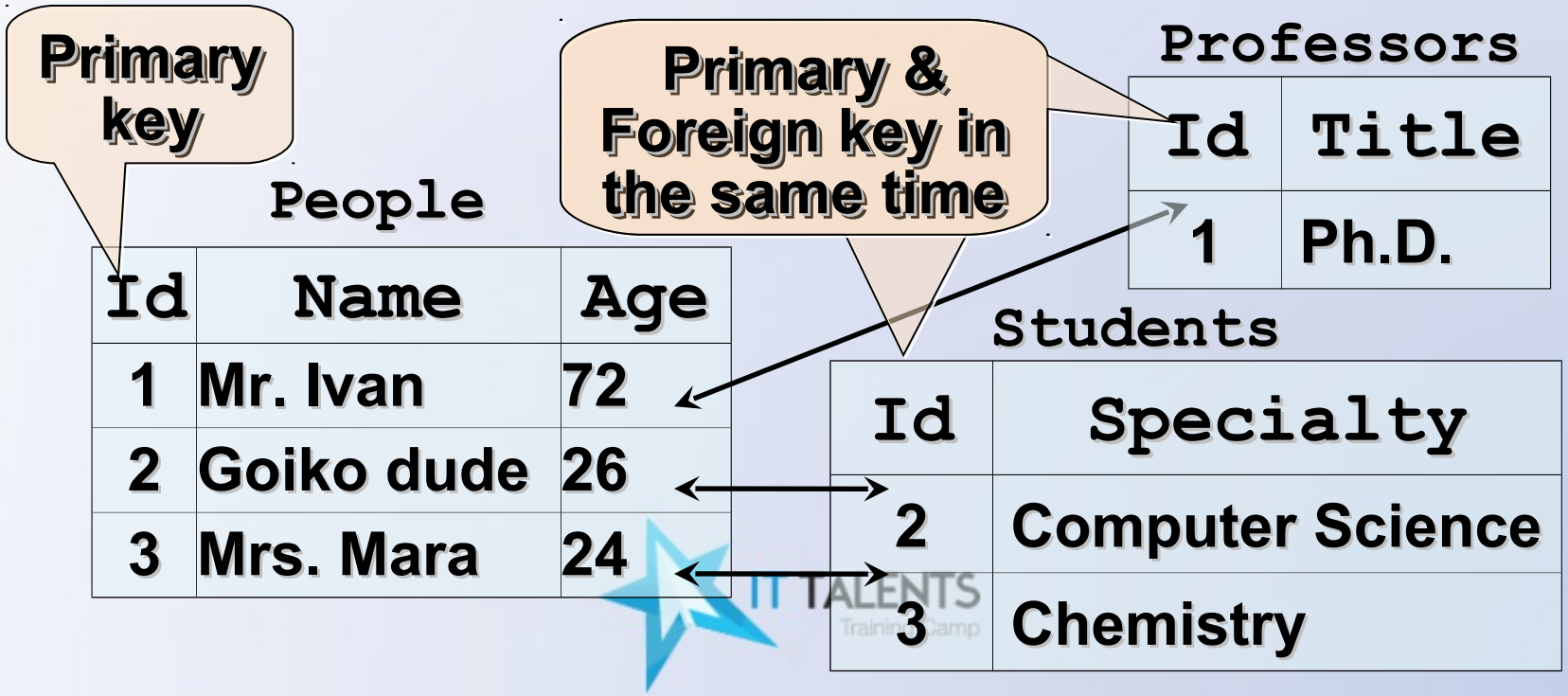
Relations' multiplicity

- Relationship many x many
  - 1 record in the first table has many corresponding records in the second one and vice versa
  - Implemented through an extra table



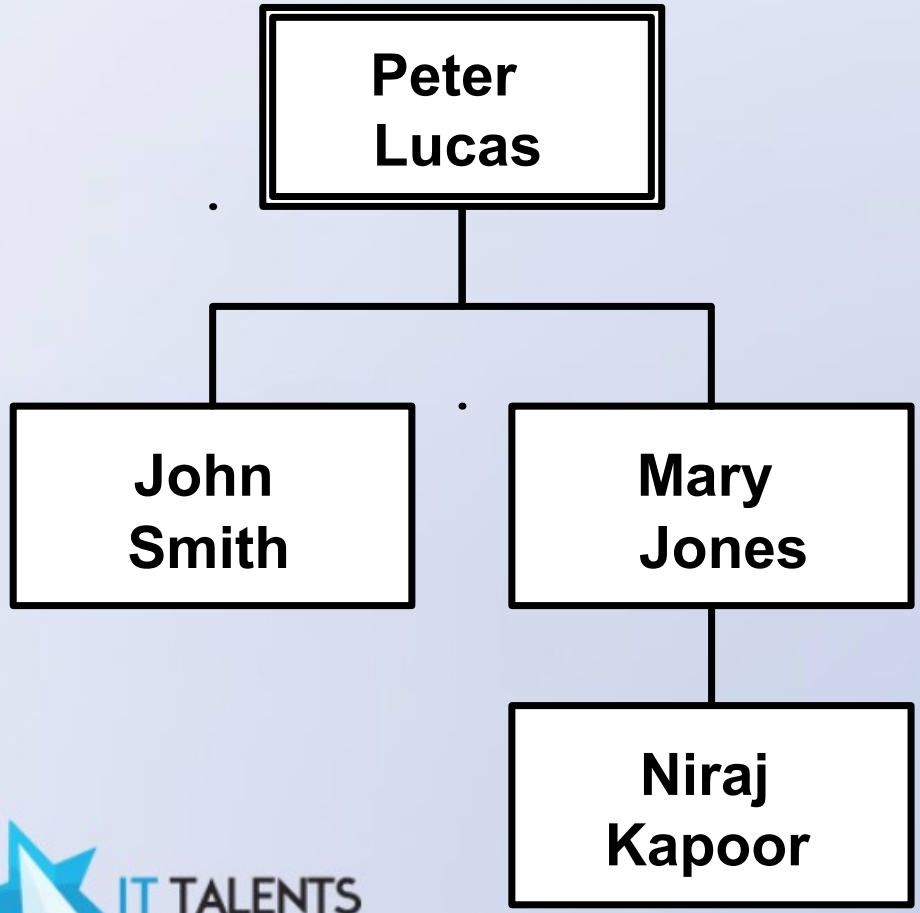
Relations' multiplicity

- Relation 1 x 1
  - 1 record in a table corresponds to 1 record in the other table
  - Models table inheritance



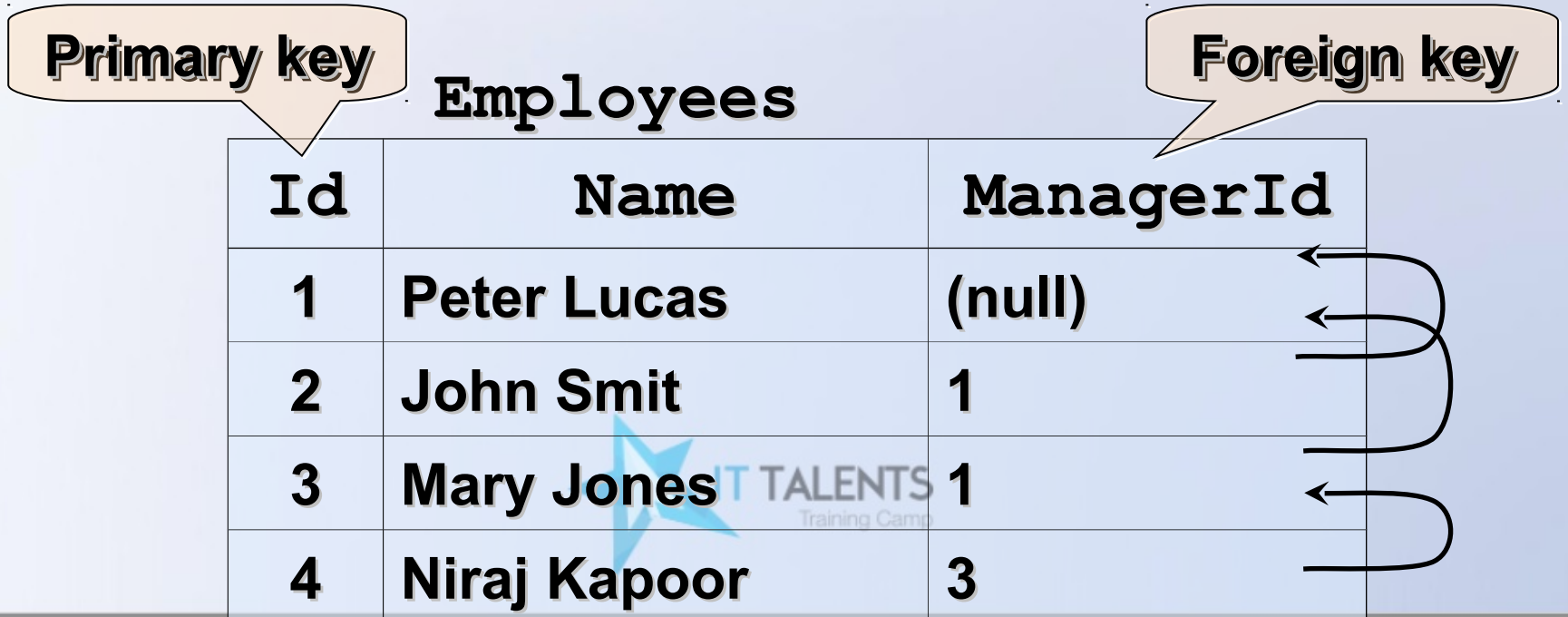
Representation of data organized as trees

- How do we represent trees and graphs?



Self-relationships

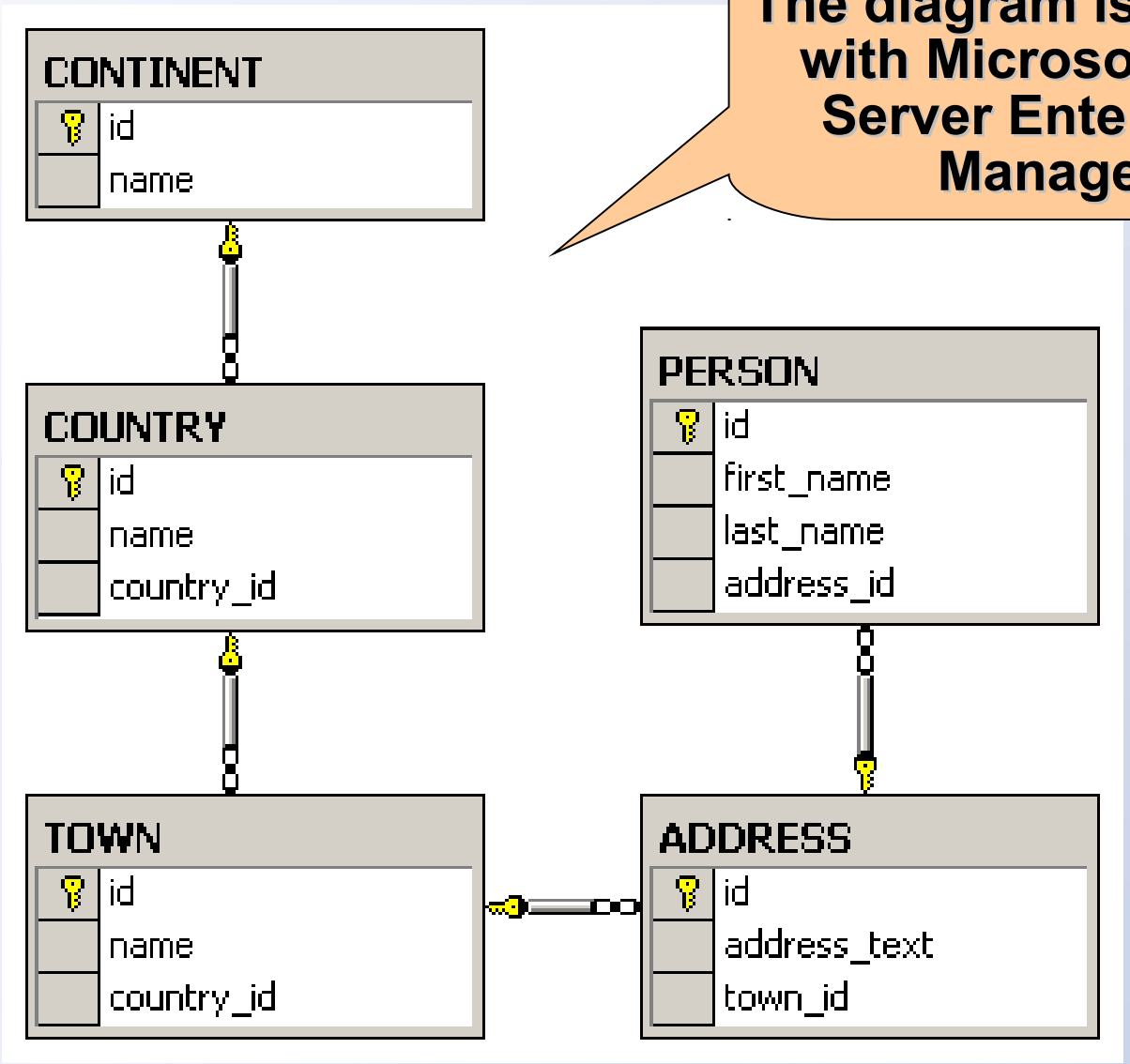
- The primary/foreign key relations can point to one and the same table (this is also called an auto relation)
  - Example: employees in a company have a manager, who is also an employee



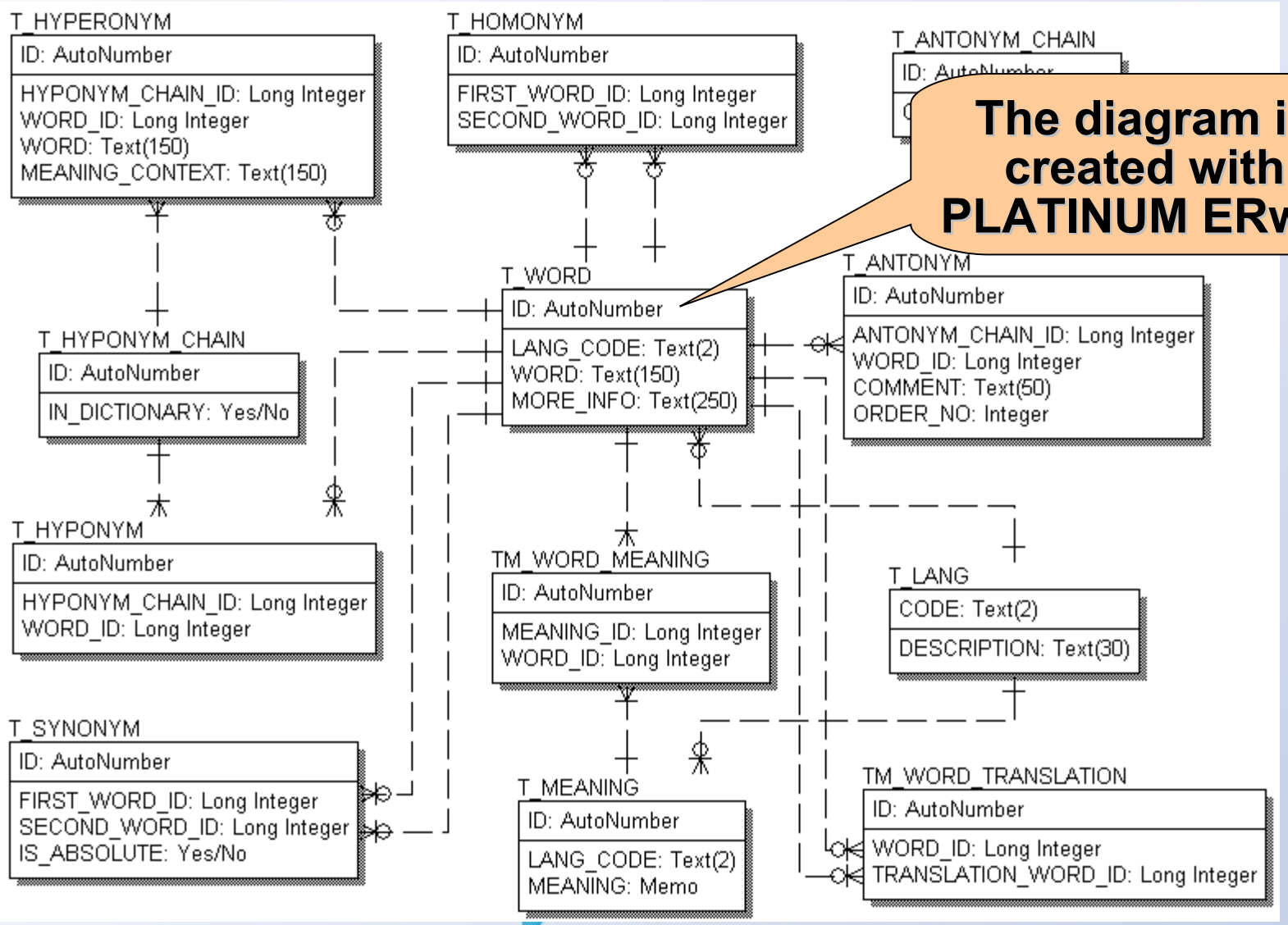
Relational scheme

- A *relational scheme* of a DB is the collection of:
  - All the schemes of all the tables
  - The relations between the tables
- The relational scheme describes the structure of a database
  - Doesn't contain data, but *metadata*
- Relational schemes are graphically displayed through Entity/Relationship diagrams (E/R Diagrams)

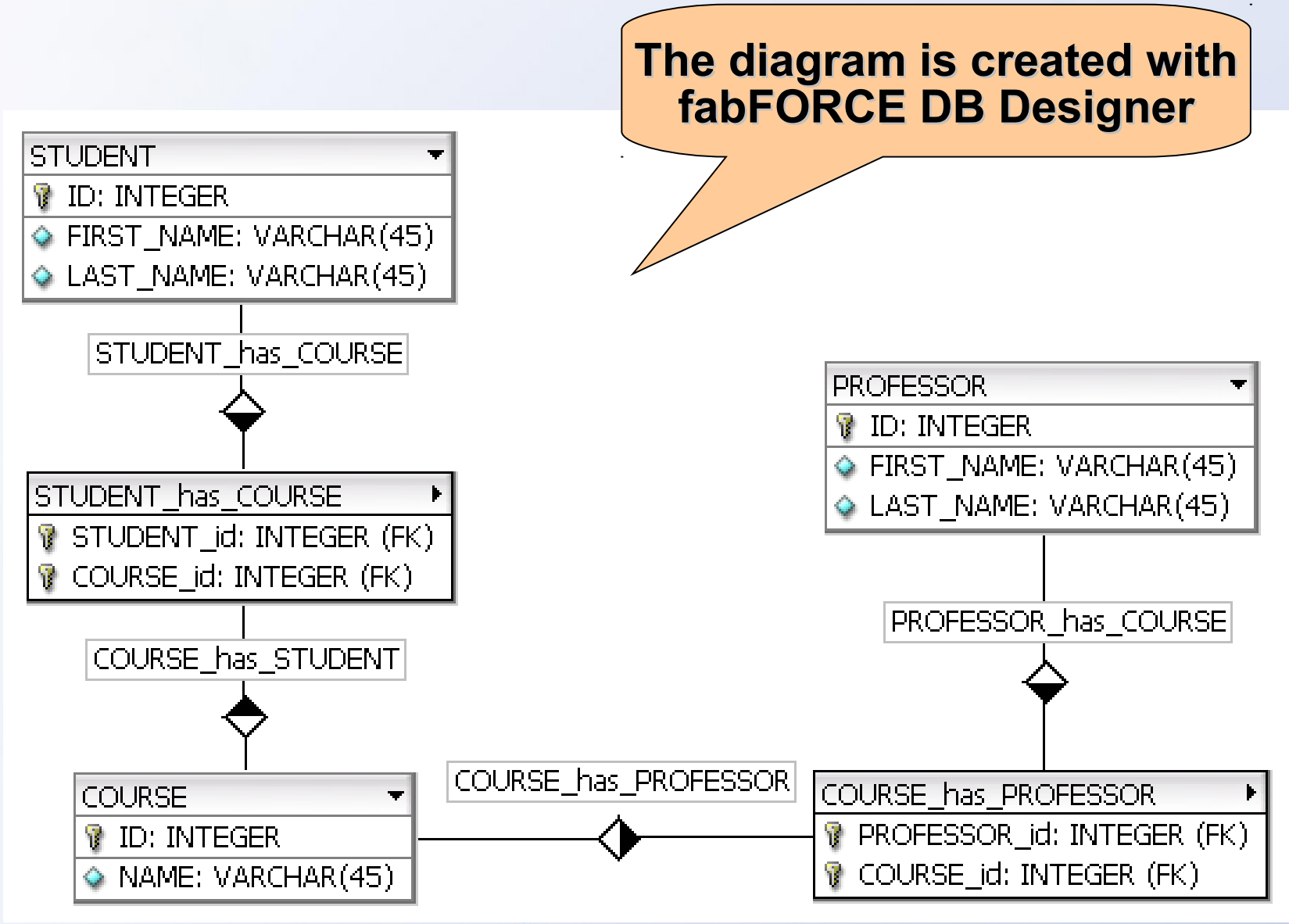
E/R Diagrams – example



E/R Diagrams – example

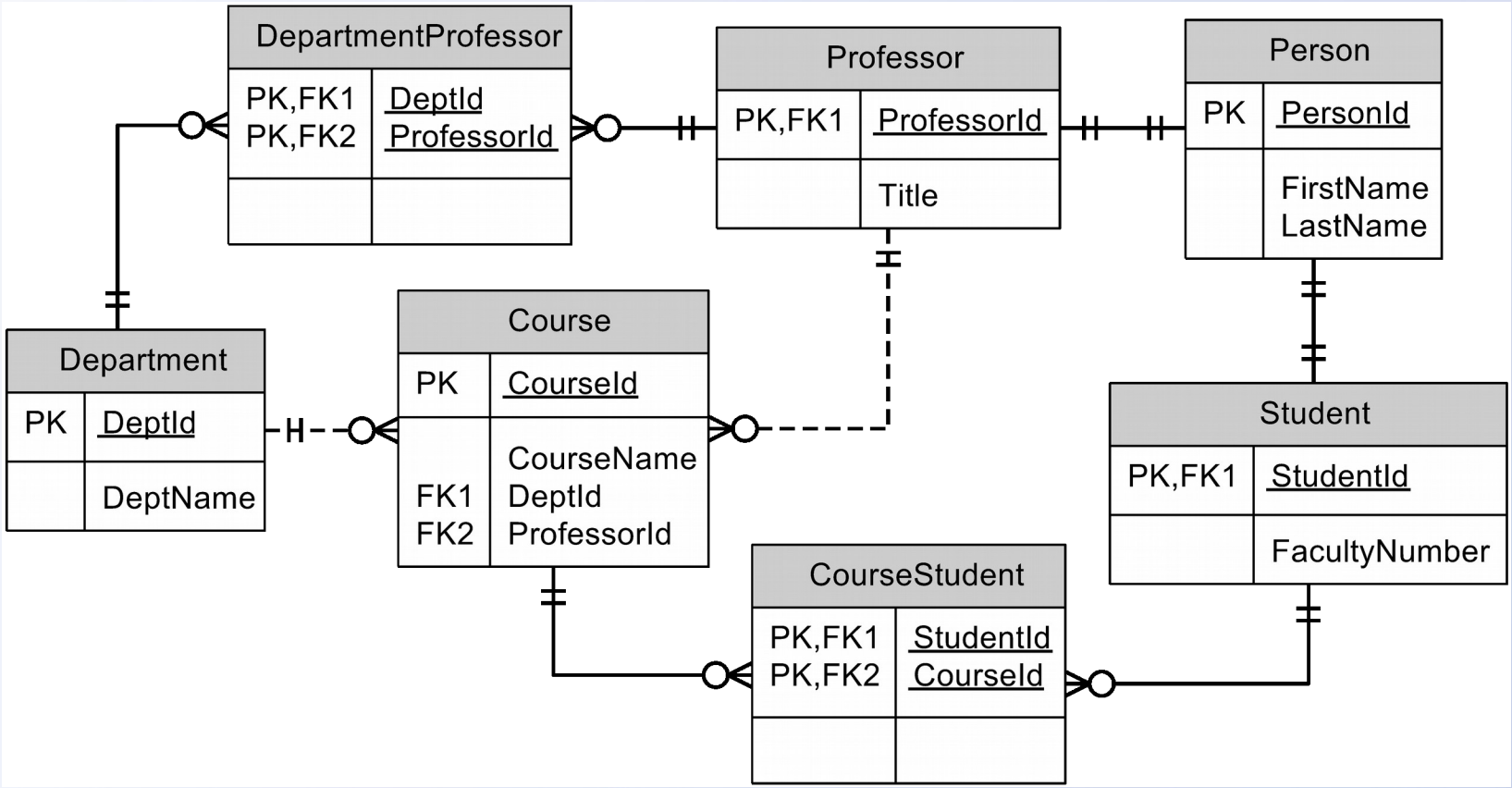


E/R Diagrams – example



E/R Diagrams – example

The diagram is created with MS Visio



Tools for E/R Design

- E/R diagrams are created with Data Modeling Tools:
  - SQL Server Enterprise Manager Oracle Designer
  - Microsoft Visio
  - Computer Associates ERwin
  - CASE Studio
  - IBM Rational Rose
  - theKompany Data Architect

Normalization

- Normalization of the relational scheme removes repeating data
- Non-normalized data contain many repetitions. For example:

What`s wrong here ?

Product	Producer	Price	Category	Shop	Town
Yoghurt	Mlexis LTD	0.67	food	store "Mente"	Sofia
bread "Dobrudja"	Bakery "Smoky"	0.85	food	store "Mente"	Sofia
beer "Zagorka"	Zagorka CO	0.68	soft drinks	stall "non-stop"	Varna
beer "Tuborg"	Shumen Drinks CO	0.87	soft drinks	stall "non-stop"	Varna

Normalization

- 1-st Normal Form
  - The data has table appearance
  - The fields in the rows are atomic (inseparable) values
  - There are no repetitions within a single row
  - A primary key is defined for each table

Book	ISBN (PK)	Author	AuthorEmail
.NET Framework	3847028437	Mr. Kiro	bai-kiro@abv.bg
Beginning SQL	7234534450	Santa	dedo@mraz.org

Normalization

- 2-nd Normal Form
  - Retains all requirements of 1-st Normal Form
  - There are no columns that depend on part of the primary key (if it consists of several columns)

The price depends on the book

E-mail depends on the author

book (PK)	Author (PK)	Price	AuthorEma
Introduction to PHP	Mr. Kiro	37.25	bai-kiro@abv.bg
Beginning SQL	Santa	19.95	dedo@mraz.org

Normalization

- 3-rd Normal Form
  - Retains all requirements of 2-nd Normal Form
  - The only dependencies between columns are "a column depends on the PK"

Id	Product	ProducerId	Price	CategoryId	Shop Id	Town Id
1	Yoghourt	2	0.67	2	4	1
2	bread "Dobrudja"	3	0.85	2	4	1
3	rakiya "Peshtera"	6	6.38	5	2	1
4	beer "Tuborg"	4	0.87	4	1	3

Normalization

- 4-th Normal Form
  - Retains all requirements of 3-rd Normal Form
  - There is one column at most in each table that can have many possible values for a single key (multi-valued attribute)

One author can have many books

One author can have many articles

AuthorId	Book	Article
2	.NET Programming	Regular Expressions in .NET
4	Mastering J2EE	Best Practices in J2EE

Normalization

- Example for a normalized scheme (in 4-th Normal Form):

Products

Id	Product	ProducerId	Price	CategoryId	ShopId	TownId
1	Youghurt	2	0.67	2	4	1
2	bread "Dobrudja"	3	0.55	2	4	1
3	rackia "Pe6tera"	6	4.38	5	2	1
4	beer "Tuborg"	4	0.67	4	1	3

Vendors

Id	Name
2	"Mlex" ООД
4	"Zagorka" АД

Categories

Id	Name
4	beer
2	food

Stores

Id	Name
1	Billa
4	METRO

Towns

Id	Name
1	София
3	Варна

Constraints

- Constraints set data rules which cannot be broken
- Primary key constraint
  - The primary key is unique for each record
  - The primary key can not be null
- Unique key constraint
  - Values in a column (or a group of columns) are unique
  - The value can be null

Constraints

- Foreign key constraint
  - The value in a given column is a key from another table
- Check constraint
  - Values in a certain column meet some condition
  - For example:
    - `(hour>=0) AND (hour<=24)`
    - `name = upper(name)`

Indexes

- Indexes speed up searching of values in a certain column or group of columns
- Used in big tables
- Usually implemented as B-trees or hash tables
- They can be outer indexes (outside the table) or built-in
- Adding and deleting of records in indexed tables is slower

The SQL language

- SQL (Structured Query Language)
  - Standardized declarative language for manipulation of relational databases
- SQL supports:
  - Creating, altering, deleting tables and other objects in the database
  - Searching for, retrieving, inserting, changing and deleting records

The SQL language

- SQL consists of:
  - DDL – Data Definition Language
    - CREATE, ALTER, DROP commands
  - DML – Data Manipulation Language
    - SELECT, INSERT, UPDATE, DELETE commands
- Example for an SQL SELECT query:

```
SELECT Towns.Name, Countries.Name
FROM Towns, Countries
WHERE Towns.CountryId = Countries.Id
```

Stored Procedures

- Procedures at database level (stored procedures)
  - Code executed on the very database server
  - Much faster than an outer code
  - Data is locally accessible
  - Can accept parameters
  - Can return result
    - single value
    - record set

Views

- Views are named SQL SELECT queries which are used as tables
- Facilitate writing of complex SQL queries
- Also used to do fine security adjustments:
  - A certain user isn't given permissions on any of the tables
  - He is given permissions on some of the views only (a subset of data)

Views – Example

V_Companies

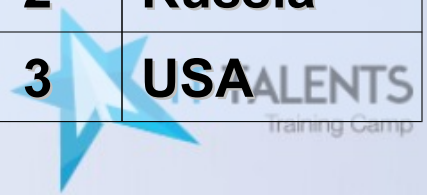
Id	Company	TownId
1	Mente LTD	1
2	BulkSoft Inc.	2
3	HardSoft CO	4
4	Sputnick CO	3

V_Towns

Id	Town	CountryId
1	Sofia	1
2	New York	3
3	Moscow	2
4	Plovdiv	1

V_Countries

Id	Country
1	Bulgaria
2	Russia
3	USA



Triggers

- Triggers are database level procedures that activate when some event occurs, for instance:
  - When inserting a record
  - When changing a record
  - When deleting a record
- Triggers can perform additional data processing
  - To change the newly added data
  - Support for logs and history

Triggers – Example

- We have a table with company names :

```
CREATE TABLE Companies(
   Id number NOT NULL,
   Name varchar(50) NOT NULL)
```

- A trigger that appends an "Ltd." at the end of the name of a new company:

```
CREATE OR REPLACE TRIGGER trg_Companies_INSERT
   BEFORE INSERT ON Company
   FOR EACH ROW
BEGIN
   :NEW.Name := :NEW.Name || '  Ltd.';
END;
```

Questions?

Exercises

1. What database models do you know?
2. Which are the main functions performed by a Relational Database Management System (RDBMS)?
3. Define "table" in database terms.
4. Explain the difference between a primary and a foreign key.
5. Point out the different types of relationships between tables.

Exercises (2)

6. When is a certain database normalized? What are the advantages of a normalized database?
7. What are constraints used for in a database?
8. Point out the pros and cons of using indexes in a database.
9. What's the main purpose of the SQL language?
10. What are the pros and cons of using views ?