

# Databases



IT Learning &  
Outsourcing Center

# Foundations

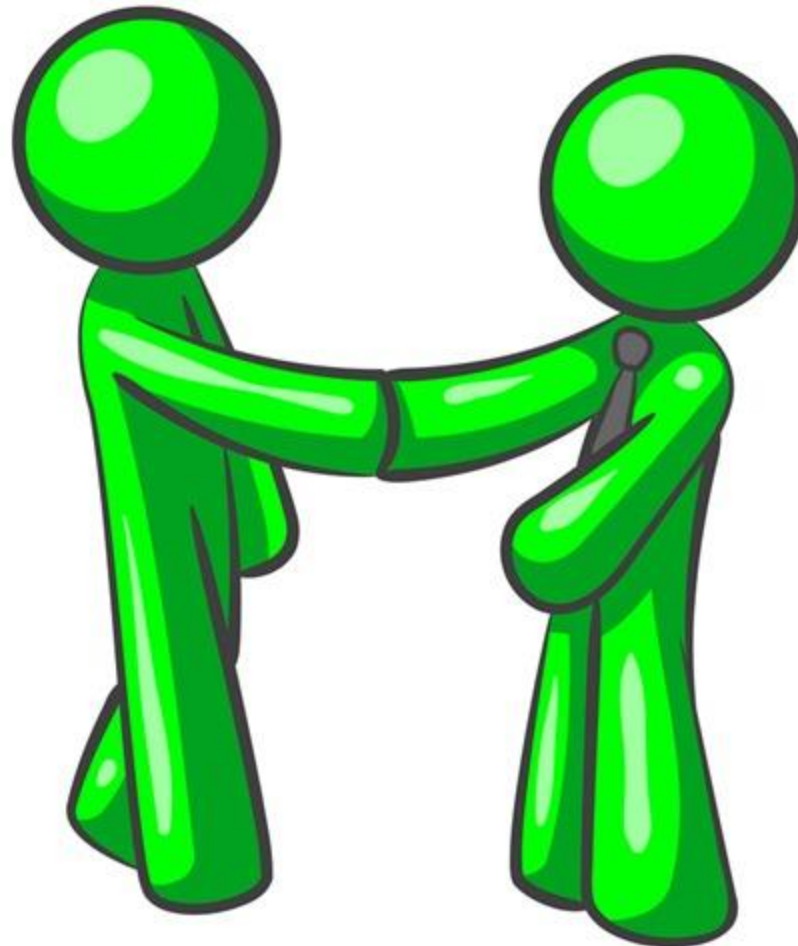
Lector: Hristo Topuzov  
Skype: h\_topuzov  
E-mail: Hristo.Topuzov@pragmatic.bg

[www.pragmatic.bg](http://www.pragmatic.bg)

2017

Copyright © Pragmatic LLC

# Introduction





# Agenda...

- Why do we need data
- Introduction to Relational Databases
- MySQL data types
- Constraints
- Data Normalization
- Other Objects

# Why do we need data



- Why is DATA important for business ?



# Database

- A database is a collection of information that is organized so that it can easily be accessed, managed and updated
- Database types:
  - Relational Database – the most popular
  - Flat Database - data is organized in a single kind of record with a fixed number of fields
  - Object Oriented Database - data is organized with similarity to object oriented programming concepts. An object consists of data and methods, while classes group objects having similar data and methods
  - Hierarchical Database - data is organized with hierarchical relationships



# Relational databases

- Store data in database tables (relations) that has a set of attributes (columns) with a type (domain) and rows (tuples) that have values for each column
- Store additional objects such as views, indexes, constraints, procedures, functions and others (most of them are covered in the course)

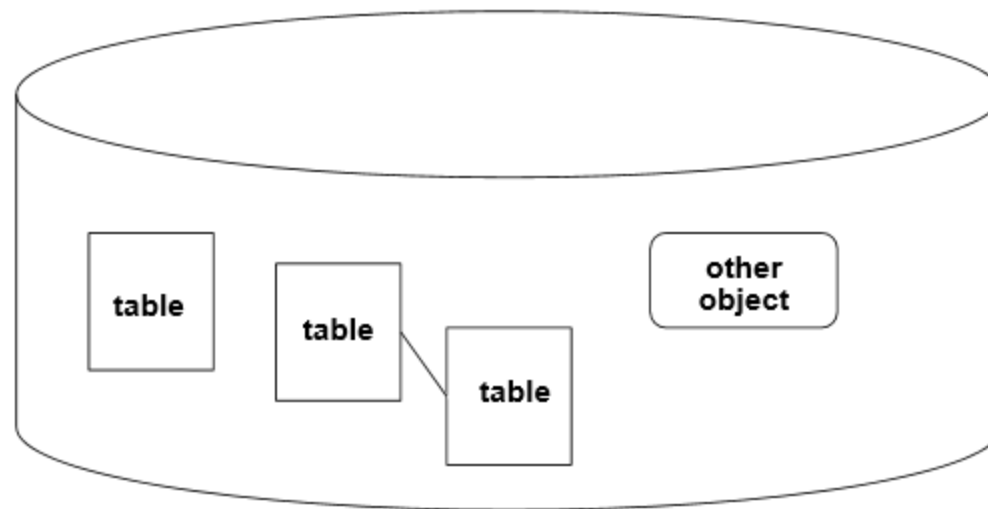
# Example relational database table



Customers

ID	FisrtName	LastName	Email	Address
1	Alexander	Karamfilov	<a href="mailto:alex.karamfilov@gmail.com">alex.karamfilov@gmail.com</a>	Sofia, bul. Bulgaria 86
2	Milen	Strahinski	<a href="mailto:m_strahinski@abv.bg">m_strahinski@abv.bg</a>	Sofia, bul. G.M. Dimitrov 47
3	Hristo	Topuzov	<a href="mailto:topuzov@gmail.com">topuzov@gmail.com</a>	Sofia, bul. Badnina 34

# Example database





# Database Management Systems



- A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.
- Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning, backup and recovery, replications and mirroring.

# Database Management Systems



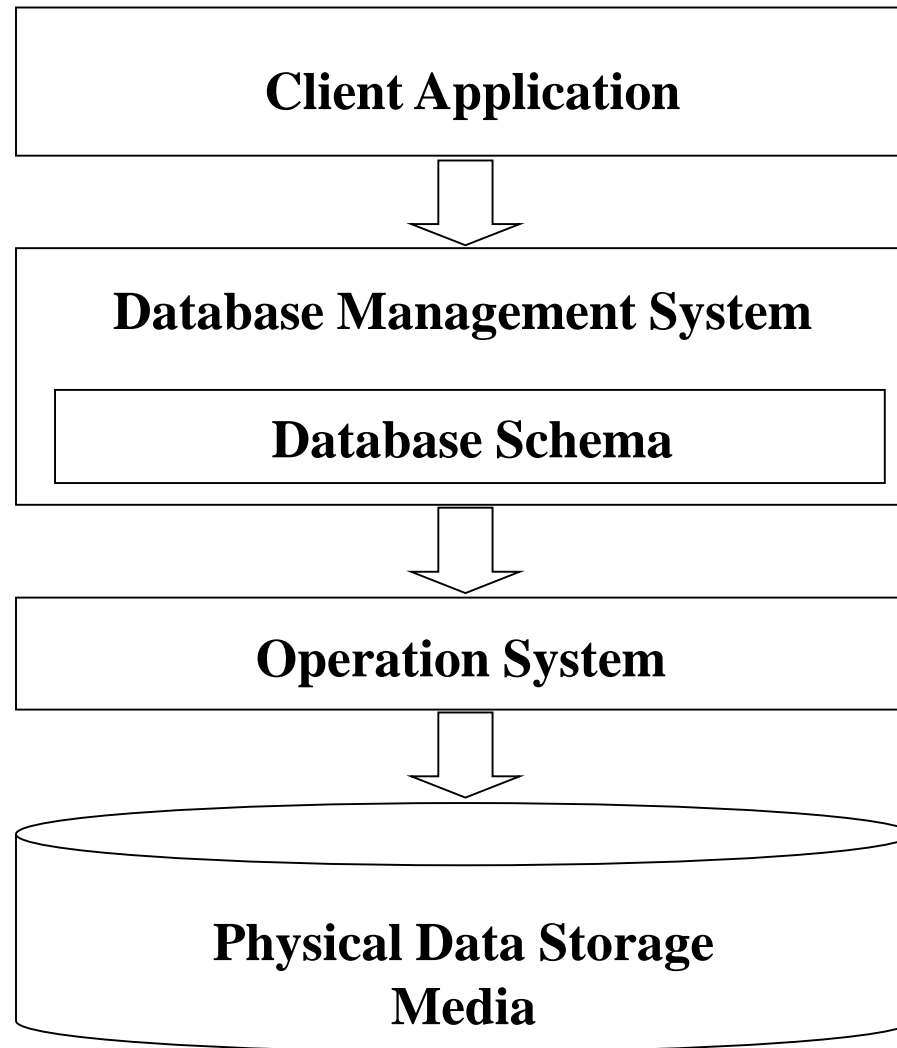
- DBMS support multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when they attempt to handle same data item, but users are always unaware of them.
- DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data.

# Database Management Systems



- DBMS provides:
  - Data abstraction and independence
  - Data security
  - A locking mechanism for concurrent access
  - The ability to swiftly recover from crashes and errors
  - Logging and auditing of activity
  - Simple access using a standard application programming interface (API)
  - Can provide an easy way to automate administrative tasks
  - Large data volumes

# Database Management Systems



# Relational database management system



- A relational database management system (RDBMS) is a program that lets you create, update, and administer a relational database
- Popular RDBMS:
  - MySQL - "popular" and open source database. It's not the most advanced, but it's very widely used
  - PostgreSQL - (probably) most advanced open source database
  - SQL Server - developed by Microsoft for the enterprise environment
  - Oracle - developed by Oracle Corporation for the enterprise environment



# MySQL data types

- Numeric data types
  - TINYINT - The signed range is  $-128$  to  $127$ . The unsigned range is  $0$  to  $255$
  - SMALLINT - The signed range is  $-32768$  to  $32767$ . The unsigned range is  $0$  to  $65535$
  - INT or INTEGER - The signed range is  $-2147483648$  to  $2147483647$ . The unsigned range is  $0$  to  $4294967295$
  - BIGINT - The signed range is  $-9223372036854775808$  to  $9223372036854775807$ . The unsigned range is  $0$  to  $18446744073709551615$
  - Decimal - A decimal number. The maximum number of digits before the decimal is  $65$  and after the decimal is  $30$



# MySQL data types

- TEXT TYPES
  - CHAR( ) - A fixed section from 0 to 255 characters long.
  - VARCHAR( ) - A variable section from 0 to 255 characters long.
  - TEXT - A string with a maximum length of 65535 characters.
- Date Types
  - DATETIME - Use when you need values containing both date and time information - YYYY-MM-DD HH:MM:SS
  - DATE - Use when you need only date information - YYYY-MM-DD
- BIT data type - true or false



# Null Value

- A value of NULL indicates that the value is unknown
- A value of NULL is different from an empty or zero value
- Comparisons between two null values, or between a NULL and any other value, return false





# Constraints

- Constraints provide restrictions on the columns of a table
- Types of constraints:
  - **Not - null** constraint - a value cannot be NULL
  - **Unique key** constraint - a set of columns that must be unique for each row; values can be NULL
  - **Check constraint** - values in a column meet some condition.  
For example: (hour>=6) AND (hour<=23)

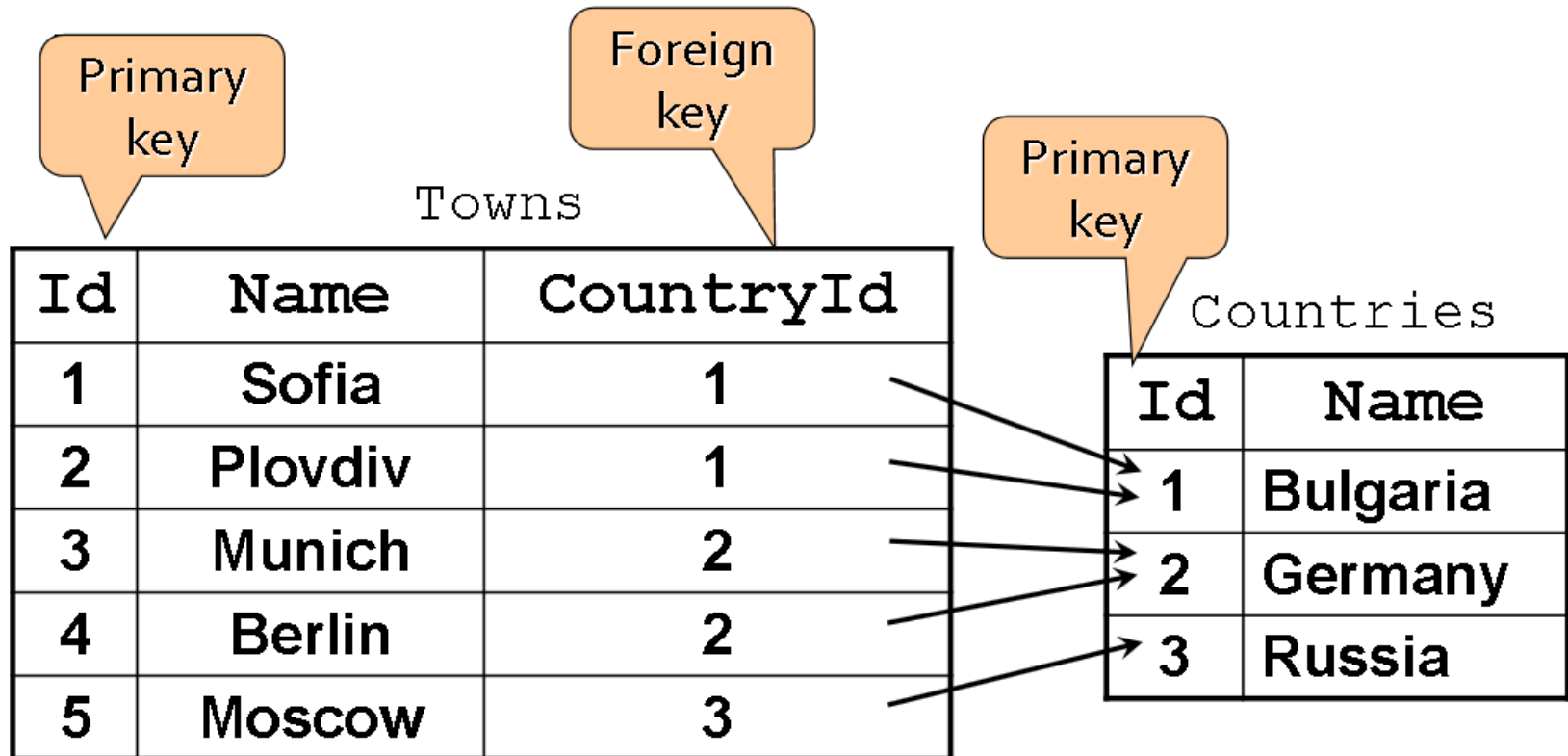


# Constraints

- Types of constraints:
  - **primary key** constraint a set of columns that identifies uniquely each row; defined over some of the columns; values cannot be NULL
  - **foreign key** constraint - a set of columns in one table references a set of columns (a unique or primary key) in another table



# Constraints



- A table has only one primary key, but can have multiple foreign keys.

# Database Relationships



- 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:
  - many x 1 (e.g. town/country)
  - 1 x many (e.g. country/town)
  - many x many (e.g. student/course)
  - 1 x 1 (human/student)

# Database Relationships



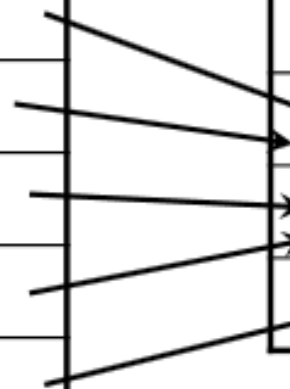
- In a relation of type **1 x many** one record in the first table has many corresponding records in the second one (**many x 1** is the opposite).

Towns

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

Countries

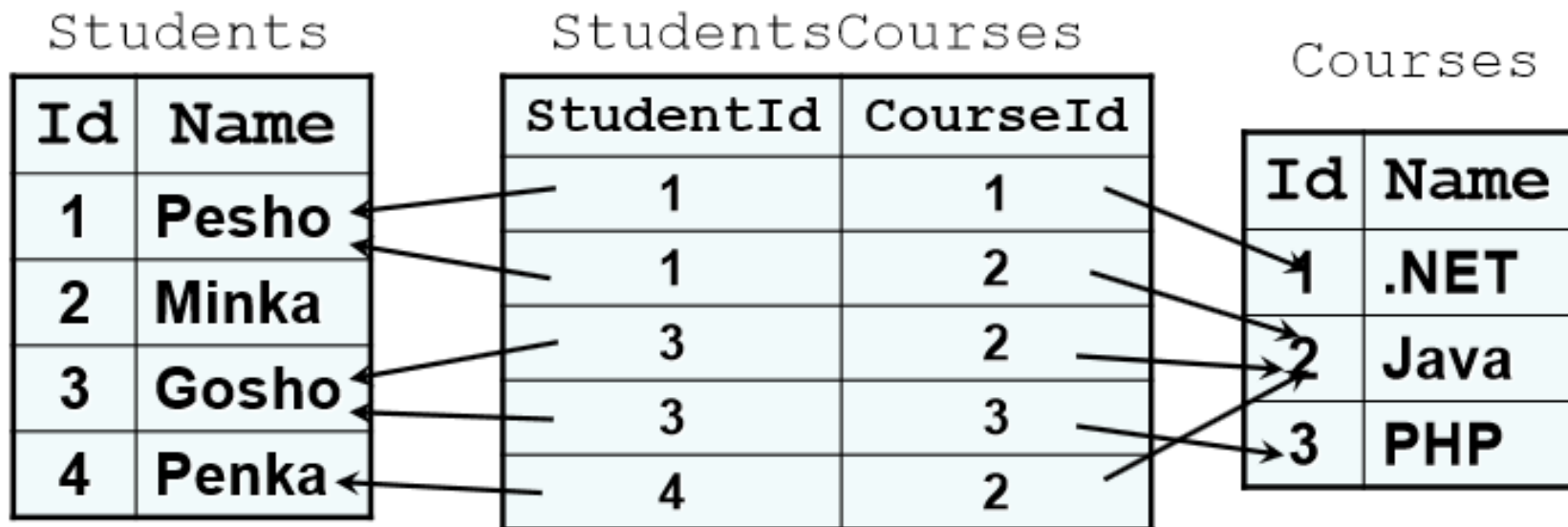
Id	Name
1	Bulgaria
2	Germany
3	Russia





# Database Relationships

- In a relation of type **many x many** one record in the first table has many corresponding records in the second one and vice versa. It is implemented using an extra table.

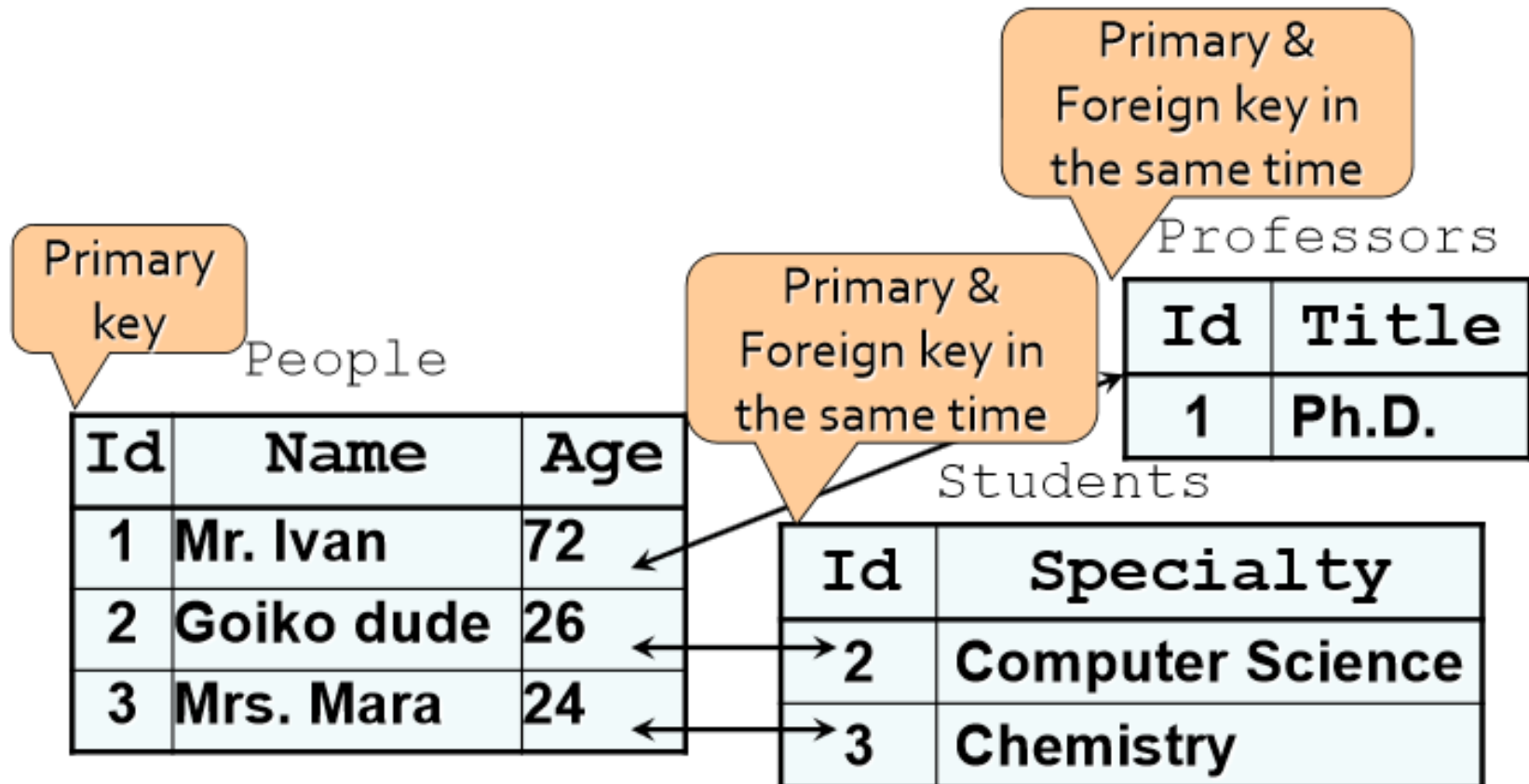


- (many x many is split into one x many and many x one so that foreign keys can be used)

# Database Relationships



- A relation of type 1 x 1 one record in a table corresponds to 1 record in the other table



# Database Relationships



- The foreign key can also point to one and the same table (this is also called an auto relation) - e.g. employees in a company have a manager, who is also an employee

Primary key	Employees	Foreign Key
Id	Name	ManagerId
1	Peter Lucas	(null)
2	John Smit	1
3	Mary Jones	1
4	Niraj Kapoor	3





# Data Normalization

- Database normalization is the process of organizing the tables of a relational database to minimize data redundancy
- Database normalization can save storage space and ensure the consistency of your data
- Separates the relations in a way that no unnecessary data is inserted, deleted or updated in the database
- Reduces the number of modifications to the database structure when new data is introduced to the schema



# First Normal Form

- There no duplicate rows (i.e. there must be a primary key) and columns in a table; the fields in the rows are atomic values (i.e. must not have more than one value)
- Student table:

Student	Age	Subject
Adam	15	Biology, Maths
Alex	14	Maths
Stuart	17	Maths



# First Normal Form

- Student table in First normal form will be:

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

# Second Normal Form



- The relation is in first normal form
- There are no columns that depend on part of the primary key
- Columns that depend only on one part of the concatenated key are extracted to a separate table

# Second Normal Form



- Student table in First normal form

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

# Second Normal Form



- New student table following Second normal form will be:

Student	Age
Adam	15
Alex	14
Stuart	17

- New subject table following Second normal form will be:

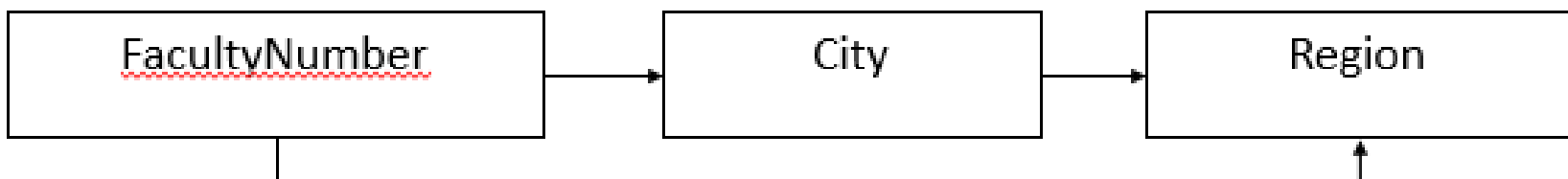
Student	Subject
Adam	Biology
Adam	Maths
Alex	Maths
Stuart	Maths



# Third Normal Form

- The relation is in second normal form
- All non-primary key columns are directly related to the primary key column

Students			
FacultyNumber	Name	City	Region
0126120	Eustace Bangs	London	Greater London
0126001	John Smith	Birmingham	West Midlands
0126012	Jane Smith	London	Greater London
0126101	John Doe	Manchester	Greater Manchester





# Third Normal Form

- New Student table following Third normal form will be:

Students		
FacultyNumber	Name	City
0126120	Eustace Bangs	London
0126001	John Smith	Birmingham
0126012	Jane Smith	London
0126101	John Doe	Manchester

- New Region table following Third normal form will be:

Regions	
City	Region
London	Greater London
Birmingham	West Midlands
London	Greater London
Manchester	Greater Manchester





# Third Normal Form

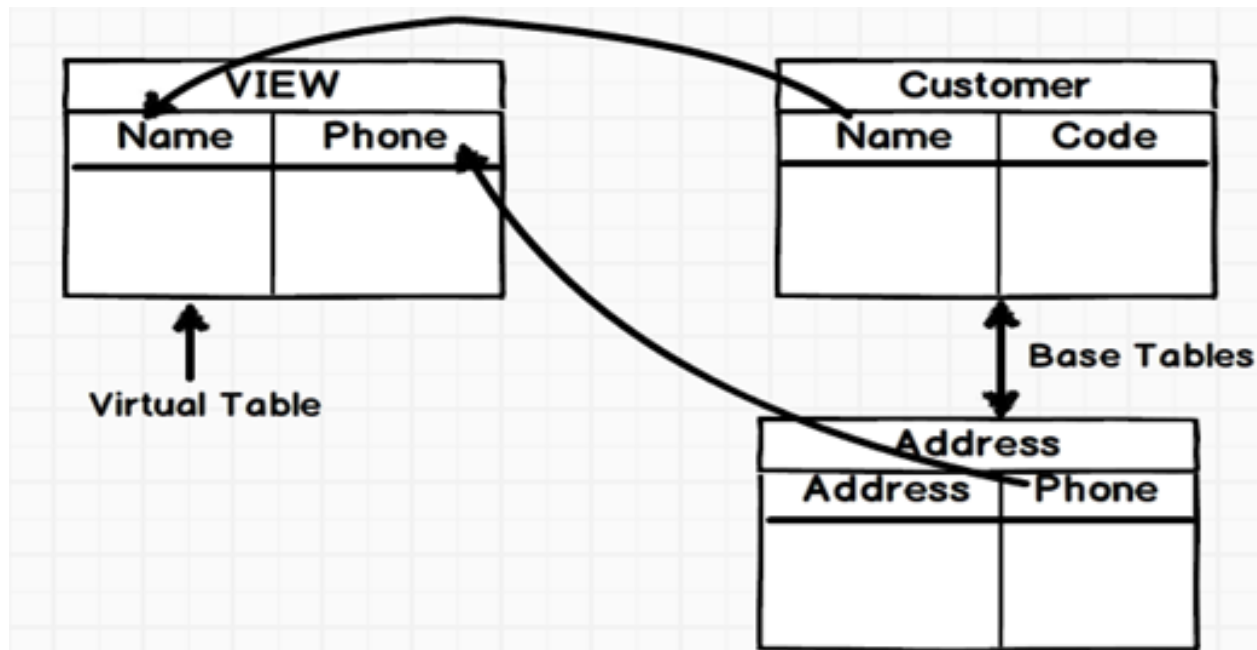
- It is not possible to register that Birmingham is in West Midlands region until a student from Birmingham is entered in the database;
- If the all Birmingham students get transferred, graduate or leave the database for some other reason the system loses the knowledge that the Birmingham is in the West Midlands region;
- If Birmingham becomes part of a different region then all **Students** need to be updated.



- Structured Query Language (SQL)
  - Declarative language for query and manipulation of relational data
  - Standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data.

# Other Objects

- View - A view is a virtual table. View does not contain any data, it is a virtual table which contains columns and data from different tables.





# Other Objects

- Store Procedure - A stored procedure is nothing more than prepared SQL code that you save so you can reuse the code over and over again
- Trigger - Special kind of stored procedure that is attached to a table and is only fired when an INSERT, UPDATE or DELETE occurs
- Indexes - Indexes are special lookup tables that the database search engine can use to speed up data retrieval



# Let's Summarize

- What RDBMS do you know?
- Which are the main functions performed by a RDBMS ?
- Explain the difference between a primary and a foreign key.
- Point out the different types of relationships between tables.
- What are the advantages of a normalized database?
- What are constraints used for in a database?
- What is SQL?

# Questions

