Introduction to Database

Linnéuniversitetet
Kalmar Växjö

Agenda

- What a database is, what it does and why database design is important in society
- Examples of database applications
- What a DBMS is, what it does.
- Types of databases
- Types of database users
- What is GDPR regulations
- When not to use databases

What database is and what it does

- **Database** is a collection of related data.
- **Data** are known facts that can be recorded and have an implicit meaning. (e.g., people names, mobile phones, addresses)
- **What is does**:
  - Store data
  - Provide an organizational structure for data
  - Provide a mechanism to interact with data (e.g., querying, creating, modifying, and deleting data, or CRUD (create, read, update, delete))
  - Can store data and relationships that are more complicated than a single table can

Problems with Single Table

- **Redundancy**: in a simple two-dimensional table with rows and columns, each row is ***irrelevant*** to other rows. As a result, the same information may be entered several times.

For exmaple: if a person is working at 10 projects, his/her information would appear in a simple table 10 times"

- **Multiple Themes**: in a simple table, each row may contain data on more than one ***entity*** (e.g., employee, project). As a result, certain information might appear in table only if information about other entities is also present.

For example: a table of projects may include project manager information (such as name, ID) and project information (name, ID,startDate) ***<u>in the same row</u>***. Because we have a single/one table.

These cause issues with modification of data in single tabel (updating, inserting, deleting)

Database vs. Single Table

Database

- Stores information in tables, where each buisness concept (e.g., Student, Department, Teacher) is stored in its own table. For example, all assosiated information with student will be stored in student table.
- Tables are joined together using matched pairs of unique data values. This matching allows us to associate one row in one table with one or more rows in another table.

Single Table

- Stores all information in a single table

A Database Properties

- A database represents some aspect of the real world (e.g., University, Shop, Hospital)
- A database is a logically coherent collection of data with some inherent meaning (**relationships** between data)
- A database is **designed**, **build**, and **populated** with data for a specific purpose and for target **users**, and **applications**
- A database can be of any **size** and **complexity**

Why database design is important in society

- The key is in "**relationships**" between buisness concepts (entities)
- Relational databases are designed to address many of the information complexity issues that arise in buisness and society.
- Allows us to model **complex natural relationships** in real-world between different buisness concepts
- Provides **backup** and **recovery** services
- Potential for enforsing **standards**
- Availability of current information (extremly important for shopping, airline, hotel, rental cars companies, etc.)
- Provides **efficient search** and complex queries

Database Applications

Traditional database applications

- Store textual or numeric related data
- Supports relationships in data
- Examples: MySQL (relational database) , Oracle, PostgreSql

Non-traditional database applications

- Store non-traditional data (images, tweets, files, videos, etc.)
- Not necessarily support relationships in data
- Examples: NoSQL (non-relationships data), Graph databases, Cloud storage (Big Data technologies for storing and managing big data)

DBMS

- A **database management system** (DBMS) is a software program that enables users to create and maintain a database.
- What it does:
  - **Defining** or creating a database involves specifying the tables, data types, structures, constrains of the data to be stored in a database.
  - **Constructing** the database is the process of storing the data on some storage (file system) that is controlled by the DBMS.
  - **Manipulating** a database includes CRUD operations such as querying data (instering, updating, deleting, reading).
  - **Sharing** a database allows multiuple users and applications to access the database simultaneously.
  - **Protecting** a database includes system protection against hardware or software crashes, and security protection against unauthorized access.
  - **Maintaining** a database includes software and hardware updates, changin requirements over time

Database Users

- **Database Administrators:** are administrating the content, the DBMS and realted software. The responisibilities are providing the authoriable access tot eh database, coordinating and monitoring its use, and acquiring software and hardware resources as needed, security breaches, and system response time.
- **Database Designers**: are responsible for identifying the data to be stored in the database according to user requirements (data abstraction, data modeling, indetifying entities and their relationships).
- **End Users**: are the people whose jobs requires access to the database for querying, updating, and generating reports.
- **Software Engineers**: are developers which implements requirements defined by database designers or on the database design phase. Then they, test, debug, and maintain these implemneted queries/interactions to database.

Whats is GDPR?

- The **General Data Protection Regulation (GDPR**) is a set of privacy and data protection rules that apply to organizations that process personal data of people residing in the European Union and European Economic Area (whether the organizations are located in the EU or not). The goal of GDPR is to give individuals more control over their personal data and at the same time simplify the regulatory environment for international businesses.

- The rules of GDPR ensure that personal data is collected under strict conditions and oblige the entities that collect the data to protect it from misuse and exploitation.
- An organization must:
  - place limits on the type of data it holds, and how long it stores it;
  - data must be legally obtained and accurate;
  - it must be stored securely in a ***GDPR compliant database*** and accessed only by approved individuals;
  - and it must be organized and easily accessible.

GDPR Database Requirements

1. **Data subject rights**: Organizations must ensure that individuals have the right to access, rectify, delete and restrict the use of their personal data.
2. **Data accuracy**: Organizations must take steps to ensure that the personal data they collect is accurate and kept up-to-date.
3. **Data minimization**: Organizations must limit the collection of personal data to what is necessary for the purpose for which it is being collected and processed.
4. **Security**: Organizations must take appropriate technical and organizational measures to ensure a level of security appropriate to the risk of the data being processed.
5. **Privacy by design**: Organizations must implement technical and organizational measures to ensure that data privacy is built into the design of the data processing system.
6. **Data breach notification**: Organizations must notify the relevant supervisory authority and affected individuals without undue delay upon becoming aware of a data breach.
7. **Data protection impact assessment**: Organizations must conduct a risk assessment of the data processing activities and document the results.

The GDRP Compliance Database

The **GDPR Compliance Database** provides a centralized location for documenting and managing the relevant aspects of the metainformation pertaining to the personal information your organization collects both internally (e.g., from employees) and externally (e.g., from customers).

How to make your database GDPR compliant:

- Get the individual's consent. This must be active, affirmative action, such as having to tick a 'yes' box – and not via a passive acceptance for example through pre-ticked boxes or opt-outs.
- Record how and when an individual gave consent.

When not to use databases

- A DBMS is a complex system which requires some resources (cost) that may be overhead.
- Thus, DBMS should not be used when
  - the data is relatively simple (or unstructured) and can be stored in files,
  - the data is not expected to grow significantly over time,
  - the data is not expected to be accessed by multiple users,
  - the data is not security-sensitive
  - the application has some real-time reqirements which can not be met by DBMS
  - you need some special operations which are not supported by DBMS (e.g., distributed operations)