Database Schemas

1 Transactions and concurrency control

- We need to trust a DBMS
- We need mechanisms to ensure that the database:
  - is reliable
  - remains in a consistent state
- Especially when
  - software/hardware failures
  - multiple users access the database simultaneously
- Database Recovery - The process of restoring a database to a correct state after a failure
- Concurrency control protocols: prevent database accesses to interfere with each other
- Central notion in a DBMS:
  - Transaction: an action carried out by a single user/program which reads/updates the database
- At the end of a transaction:
  - database again in consistent state
  - valid integrity/referential constraints
- During the execution of a transaction
  - maybe in an inconsistent state
- A transaction can have two outcomes:
  - Committed - completes successfully
  - Rolled back - does not complete successfully
- Concurrency control: the process of managing simultaneous operations on the DB
- Two transactions may be:
  - both correct by themselves
  - but when executed simultaneously they may cause inconsistency in the database

2 Abstract data models

- Data definition model - specifies entities/attributes/relationships/constraints for the stored data
- However, DDL is too low level to describe the data organisation in a simple way, understandable by most users
- We need a data model - a collection of intuitive concepts describing data, their relationships and constraints

3 Types of data organisation

- Three characterisations of data:
  - Structured data
    - Data represented in strict format
    - The DBMS checks to ensure that the data follows:
      - the structures
      - the integrity and referential constraints
  - Semi structured data
    - Self describing data
    - the ”schema” information is mixed with the data values
    - How do we end up with such data?
      - sometimes ad-hoc
      - not known in advance how it will be stored/managed
    - This data may have some structure, but:
      - not all the parts of the data have the same structure
      - each data object may have different attributes that are not known in advance
  - Unstructured data
    - Very limited indication of the type/structure of the data

4 Database design

- Conceptual design
  - Construct a first, high level model of the data: ER model
  - using the users' requirements specification
  - independently of any physical considerations
  - it serves as the fundamental understanding of the system
- Logical design
  - construct the relational data model of the data
  - Using the conceptual design map entities and relationships to tables
  - use normalization techniques to eliminate data redundancy and anomalies
- Physical design
  - describe the database implementation of the logical design
  - specific storage structures/ access methods/ security protection
  - aim is optimum performance