Contents

Introduction

- Online Course
- Assessment
  - Exam: 75%
  - Countinuous Assessment: 25%
    - Project Work
    - Online Course

Outline

- System Engineering
- Information Design

1. Introduction to Databases
2. Database Architectures
3. Database Models
4. Relational Algebra for Data Manipulation
5. Designing Databases: Functional Dependency
6. Designing Databases: The Entity Relationship Approach
7. Designing Databases: Mapping from ER to Relations

What is Data?

- *Data* is any information that you want to store and refer to again. Data can be:
  - Text
  - Numbers
  - Dates
  - Images
  - Videos
  - Files
  - Any other types of information

What is a Database?

- An organised collection of Information, or Data

  > "A database is a persistent collection of related data supporting several different applications within an organisation"

- Organised to
  - Model aspects of reality
  - In a way that supports processes that require this information
    - A collection of medican records in a Hospital
    - Finding records by a specific Doctor or Patient
  - Mostly, to make the data more useful

Metadata

- Metadata adds context to data

Metadata	data
Student Number	4366247
Name	John Patrick Smith
Account Balance	1982.00

- Metadata can include:
  - Data type
  - Name of element
  - Size
  - etc.
- Can be used at any level of aggregation

Database Management Systems (DBMS)

- Goal of a DBMS is to simplify the storage of, and access to, data
- DBMS support:
  - Definition
  - Manipulation

  - Querying

- A DBMS can manage a single, or set of, DBs
- Should provide:
  - *Efficient*, *reliable* and *secure* management of large amounts of *persistent* data.
  - Languages for **defining** the DB:
    * *Data definition language*
    * This data about data is called metadata
  - Languages for **storing**, **retrieving** and **updating** data in the DB
    * *Data manipulation languages*

Why should I care?

- Ubiquity
- Software Market
  - Roughly same size as OS market
- The majority of large corporations, web sites, scientific projects all manage both day to day operations as well as business intelligence and data mining using databases

Why use a Database?

- Pre-DB era was characterised by file processing systems
- File systems offered
  - Efficient, direct access to individual records
  - Fast sequential processing
- Choice of file organisation technique was based on the needs of the particular application
- However, if multiple applications want to share data, this can give rise to wasteful duplication
  - Patient record application and Accounting application
  - Patient names, address, visit charges, etc.
- Duplication of data
  - Wasteful of storage
  - Inefficient
  - Most importantly, leads to inconsistencies
- DB approach aims to eliminate such *redundancy*
- Data from all applications is integrated and stored once in the DB
- All applications access the same physical copy of the data

How do DB and DBMS address these problems?

Data Independence

- File-based systems are *data dependent*
    - as the way in which data is viewed by an application and the way in which it is physically stored are built into the logic of the application program
- DBMS support *logical data independence*
    - by allowing the view of the data to be changed and data added without affecting it's underlying organisation
- DBMS support *physical data independence*
    - as they *insulate* the way in which data is viewed by the applications/users from the way in which it is physically stored

Data Integrity

- Data Integrity is concerned with the *consistency* and *accurancy* of the data in the Database
- Data Redundancy is a major threat to Data Integrity
- Support for Data Integrity is a key feature of any DBMS
- Database model parts of the real world in which many rules apply
    - "A student has only one address"
    - "A student must take five courses in the final year or four courses plus a project"
- DBMS express such rules by means of "integrity constraints"
- Validation of data values being entered into the DB is another aspect of Data Integrity
- Many users/applications simultaneously updating the Database can threaten Data Integrity
    - This requires "concurrency control"

Backup and Recovery

- The only facility available to file processing systems to restore data following failure is if a back-up was scheduled/manually taken
    - Time Machine on MacOSX
    - Backup and Recovery in Windows

- Insufficient in many on-line enivonments and organisations where data is a strategic resource
- DBMS provide very sophisticated recovery mechanisms

Query Language Support

- File systems are basically tools for physical storage of data
- They make data much less accessible to users than database systems
  - If a GP wanted to examine all records for a single patient, this would be very difficult
  - Even if they were meticulous in where the stored them
  - Potenially would need an application to process and combine the data
- DBMS provide a variety of interfaces to suit the needs of a wide range of users

Metadata Management

- In applications which process data from a file system, metadata is often part of the application program
- This can lead to duplication of metadata across applications
  - Leading to integrity problems
- Imagine a patient record, and to look at the data in this record, we would need to look at an application program:

```
public class Patient {
    private int patient_ID;
    private String patient_name;
    private String patient_address;
    private int patient_phone;
    private String patient_allergy;
    ...
}
```

- With a Database approach
  - Metadata is stored centrally in the catalog
  - Database catalog entry for patient record
    * patient_record contains basic details on patient

Patient_ID	int(4)	Unique
Patient_name	varchar(255)	Firstname followed by Surname
Patient_Address	varchar(255)	Truncate if necessary
Patient_Phone	int(10)	Home phone
Patient_Allgergies	varchar(255)	Drug name of None

Advantages of Databases

- Search and Retrieval Capabilities
  - Filtered according to specific needs
- Reduced Data Redundancy
  - Ease of Update
- Greater Data Integrity
- Independence from Applications, Concurrent Access
- Improved Data Security
- Reduced Costs for Data Entry, Storage and Retrieval

Disadvantages of the DB Approach

- Training required for management and querying
- Database systems are complex and time-consuming to design
- Cost
  - Software
  - Hardware
  - Training
- Loss of autonomy brought about by centrilising control of the data
- Infelixibility due to complexity

Database Languages

- Programming languages which are used to
  - Defining a database
    - Its entities and the relationships between them

    - Manipulate its content
        * Insert new data and update or delete existing data
    - Conduct queries
        * Request information based upon defining criteria
- The Structured Query Language (SQL) is the most commonly used language for Relational Databases
    - Supported by all relational DBMS and is a standard

SQL

- SQL is split into four sets of commands which are divided based upon the tasks they are used for
    - Data Definition Language
    - Data Modification Language
    - Data Query Language
    - Data Control Language

Data Definition Language

- SQL uses a collection of imperative verbs whose effect is to modify the schema of the database
- Can use add, change, delete definitions of tables or other objects
- These statements can be freely mixed with other SQL statements
    - So the DDL is not truly a seperate language

Data Manipulation Language

- The data manipulation language comprises the SQL data change statements
    - Modifies stored data
    - Does **not** modify the schema or database objects
        * This is always the responsibility of the Data Definition Language
- Used for inserting, deleting and updating data in the tables of a database

Data Query Language

- The data query language allows users of a database formulate requests and generate reports

- There is one primary command used in SQL to query the database - the SELECT Statement
  - This statement is used to query or retrieve data from a table in the database
  - A query may retrieve information from specified columns or from all of the columns in the table
  - A query may have specified criteria that must be met in order for data to be returned

Data Control Language

- The data control language is used to control data access
- Can use grant to allow users to perform tasks
- Can use revoke to remove privileges and permissions

Transactions

- A way to group actions that must happen atomically
  - all or nothing
- Guarantees to move the DB content from one consistent state to another
- Isolates these actions from parallel execution of other actions/transactions
- Ensures the DB is recoverable in case of failure
  - e.g. the power goes out

Backup and Recovery

- Ensures that the DB can be returned to a stable state in case of errors, such as
  - Transaction failure
  - System errors
  - System crash
  - Data Corrution
  - Disk failure

Users

- DBMS implementer

  - Builds the DBMS System
- Database designer
  - Designs the Database, Establishes the Schema
- Database application developer
  - Develops programs that operate upon the DB
- Database administrator
  - Has overall responsibility for the DB including specifying access constrains, selection of appropriate backup and recovery measures, monitoring performance, etc.

Emergent Databases

- XML Databases
  - Document-Orientated
- NoSQL Databases
  - Web Scale, Non-Relational, Open Source
- In Memory Databases
  - Stores data in main memory rather than on disk
- Others
  - Massively parallel processing (MPP) databases
  - Online analytical processing (OLAP) databases