Chapter 1 : Introduction to Database Management

Database Characteristics
Features of Database Management Systems
Evolution of Database Technology
Functions of Database Management Systems
Architecture of Database Management Systems

DATABASE

"Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization."

- **Data**
  "***raw material that has not yet processed by the computer***."
  - Example: Jason, Kajang, Calculus, 12th June 1998.
- **Information**
  "***data that has been processed and can be used in making a decision.***"
  - Example: Calculus is one of the core major subject for CS students.

- ***Example of database system:***
  a) **Ticketing System**
  b) **Library Borrowing System**
  c) **Patient Information System**
  d) **Student Registration System**
  e) **Inventory System**
  f) **Cashier System**
  g) **ATM System**
  h) **Grading System**
- Logically related data comprises **entities**, **attributes** and **relationships** of an organization's information.

DATABASE TERMS

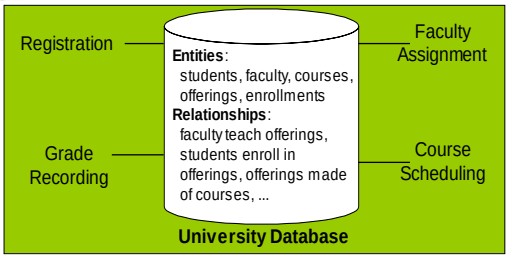
- **Entity**
  - ***unique object and can be identified in an environment.***
  - Example :
    - individual : *student, lecturer, patient, customer*
    - place : *state, district, city, country*
    - object/thing : *computer, car, book, pencil*
    - event : *sales, registration, borrowing, renting*
    - concept : *account, course, project, subject*
- **Attribute**
  - ***set of attributes for entities.***
  - Example : name, address, matricNo, gender etc.

- **Relationship**
  - All entities must be related to each other.
- **Environment**
  - ***organization where the database is build.***
  - Example : university, hospital, bank, supermarket etc.

DATABASE CHARACTERISTICS

- **Persistent**
  - Data reside on stable storage such as magnetic disk.
  - Data repetitively used will be retained on stable storage such as customers data, suppliers data etc.
- **Inter-related**
  - Data stored as separate units can be connected to provide a whole picture.
  - Example : student table relates student data with subject data in subject table = subject registration system.
- **Shared**
  - A database can have multiple uses and users.
  - Many users can access a database at the same time.

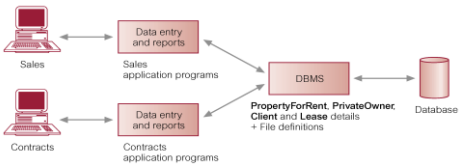
University Database



FEATURES OF DATABASE MANAGEMENT SYSTEM (DBMS)

"A software system that enables users to define, create, and maintain the database and that provides controlled access to this database."

- e.g : *Oracle, FoxPro, Microsoft Access, MySQL, Paradox, FileMaker Pro, dBase IV, Power Builder etc.*



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

Features offered by DBMS :

a) DDL (Data Definition Language)

- Permits specification of data types, structures and any data constraints.
- All specifications are stored in the database.

b) DML (Data Manipulation Language)

- allow user to insert, update, delete and retrieve data.
- general enquiry facility (query language) of the data.
- 2 types :-
  - ***procedural***
    - manipulating the database record per record and determine '***how***' the output from DML statement can be accessed.

- ***non-procedural***
  - operates on set of records and define '***what***' data to be accessed.
  - language used is SQL (Structured Query Language)

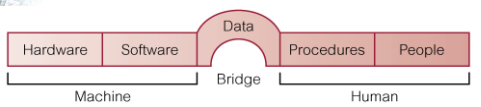
c) Controlled access to database may include:

- a security system
- an integrity system
- a concurrency control system
- a recovery control system
- a user-accessible catalog.

d) Views

- Allows each user to have his or her own view of the database.
- A view is essentially some subset of the database.
- Benefits :
  - Reduce complexity.
  - Provide a level of security.
  - Provide a mechanism to customize the appearance of the database.

COMPONENTS OF DBMS ENVIRONMENT



a) **Hardware**
- Can range from a PC to a network of computers

b) **Software**
- DBMS, operating system, network software (if necessary) and also the application programs.
- using the 3rd generation language (C,COBOL, FORTRAN, ADA, PASCAL) or 4th generation (SQL).

c) **Data**
- Used by the organization and a description of this data called the schema.

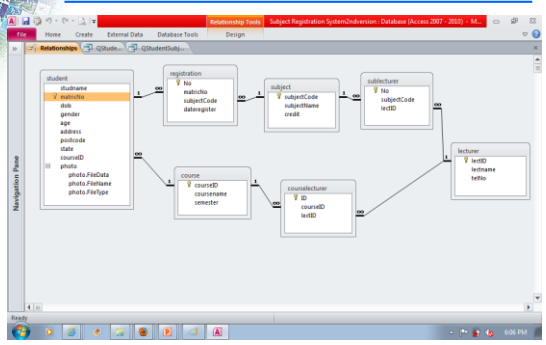
d) **Procedure**

- Instructions and rules that should be applied to the design and use of the database and DBMS.

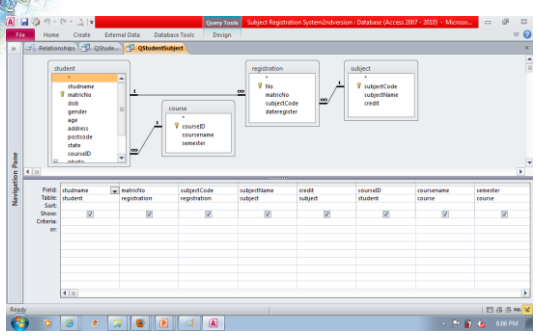
e) **People**

- 2 types :
  - i) *Database Developer* (data and database administrator, database designer, programmer)
  - ii) *Database End-user*
    - naïve user
    - sophisticated user

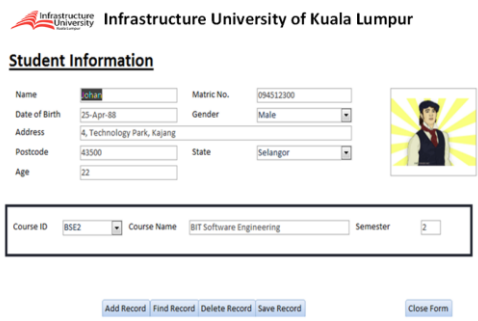
Diagram of Tables and Relationships



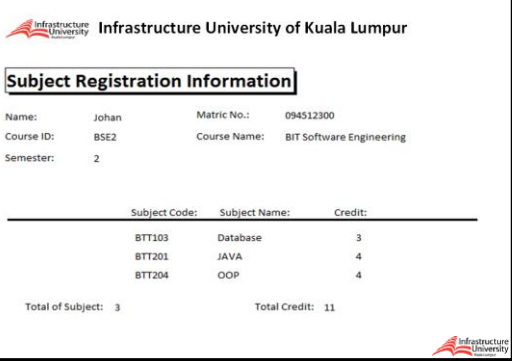
Query



Sample of Form



Sample Report



DATABASE TECHNOLOGY EVOLUTION

Era	Generation	Orientation	Major Features
1960s	1st Generation	File	File structures and proprietary program interfaces
1970s	2nd Generation	Network Navigation	Networks and hierarchies of related records, standard program interfaces
1980s	3rd Generation	Relational	Non-procedural languages, optimization, transaction processing
1990s	4th Generation	Object	Multi-media, active, distributed processing, XML enabled

ROLES in DATABASE ENVIRONMENT

a) **Data Administrator (DA)**

- responsible for the management of data resource including database planning, development and maintenance of standard, policies and procedure, and conceptual/logical database design.

Database Administrator (DBA)

- responsible for the physical realization of the database including physical database design and implementation, security and integrity control, maintenance of the operational system and ensuring satisfactory performance of the application for users.

b) **Database Designers (Logical and Physical)**

- ***Logical***
  – identifying data(entities and attributes), relationship between the data and the constraints on the data to be stored in the database.
- ***Physical***
  - decides how the logical database design is to be physically realized. This involved :-
  * mapping the logical database design into a set of tables and integrity constraints.
  * selecting specific storage structures and access methods for the data to achieve good performance.
  * designing any security measures required on the data.

c) **Application Developers**

- work from a specification produced by system analyst.
- each program contains statement that request the DBMS to perform some operation on the database such as retrieving data, inserting, updating and deleting data.
- program may be written in 3rd-generation or 4th-generation language.

d) **End-Users**

- are the clients for the database.
- 2 types :-

i) *naïve user*

- access data using icon, menu. (use a very user friendly database system)

ii) *sophisticated user*

- familiar with the database structure and facilities offered by the DBMS.
- using high-level query language such as SQL to perform the operations.
- write application program for their own use.

ADVANTAGES AND DISADVANTAGES

Advantages:-

a) ***Control of data redundancy***

- eliminate data redundancy by integrating the files so that multiple copies of the same data are not stored.
- save storage spaces, time and data consistency ensured.

b) ***Data consistency***

- if data being kept in centralized database, all data manipulation operations can be done concurrently and involve only 1 file.
- can reduce the risk of inconsistencies occurs.

c) ***More information from the same amount of data***

- if data being kept separately like file-based system, the information value is limited and only for one department.
- integration of data allows the organisation to derive additional information from the same data.

d) ***Sharing of data***

- the database belongs to the entire organisation and can be shared by all the authorised users.

e) ***Improved data integrity***

- refers to validity and consistencies of stored data.

f) ***Improved security***

- database security is the protection of the database from unauthorised users.
- integration allows DBA to define database security and the DBMS to enforce it. Can be in a form of user's name and password.

g) ***Improved backup and recovery services***

- user's responsibility or programmer to have back up in case something happen to the system or loss of data.

Disadvantages:-

a) ***Complexity***
- design, development, database management is very complex and need an expertise to develop them.
- failure to understand the system can lead to bad design decisions which can have serious consequences for an organisation.

b) ***Cost of DBMS***
- DBMS cost is inexpensive but for an organisation that needs DBMS for mainframe, the cost is very expensive.
- it is very important to do cost estimation to avoid from loss.

c) ***Additional hardware costs***
- to install DBMS, need hardware support with higher capabilities such as storage needed for back up and server to maintain user needs.

d) ***Size***
- software that has many functions such as DBMS is complex and need bigger size storage.

e) ***Cost of conversion***
- conversion cost for old software to new software include the conversion of data format, writing program usage.
- the conversion will takes time and need to train the staff.

f) ***Higher impact of a failure***
- have a high risk of failure because the data and system are centralized.
- failure of certain components can bring the operation to a halt.

FUNCTIONS OF DBMS

- several functions that a DBMS should have as listed by Codd (1982):-

a) ***data storage, retrieval, and update.***
- fundamental function of a DBMS.
- must furnish users with the ability to store, retrieve and update data in the database.

b) ***a user accessible catalog***
- must furnish a catalog in which descriptions of data items are stored and which is accessible to users.
- also called as data dictionary – a repository of information describing the data in the database.

c) ***transaction support***
- *transaction* is a series of actions carried out by a single user or application program which accesses or changes the contents of the database.
- must furnish a mechanism which will ensure either that all the updates corresponding to a given transaction are made or none of them is made.

d) ***concurrency control services***
- must furnish a mechanism to ensure that the database is updated correctly when multiple users are updating the database concurrently.
- locking method introduced to protect data being updated so that latest information can be obtained.

e) ***recovery services***
- must furnish a mechanism for recovering the database in the event that the database is damaged in any way.

f) ***authorization services***
- must furnish a mechanism to ensure that only authorized users can access the database.
- introduced the use of password.

g) ***support for data communication***
- must be capable of integrating with communication software.

h) ***integrity services***
- must furnish a means to ensure that both the data in the database and changes to the data follow certain rules.

i) ***services to promote data independence***
- must include facilities to support the independence of programs from the actual structure of the database.

j) ***utility services***
- helps the DBA to administer the database effectively.
- should provide a set of utility services:-
  i) import facilities, to load the database from the flat files and export facilities, to unload the database to flat files.
  ii) monitoring facilities, to monitor database usage and operations.
  iii) statistical analysis programs, to examine performance or usage statistics.

ARCHITECTURES OF DATABASE MANAGEMENT SYSTEMS

ANSI-SPARC Architecture

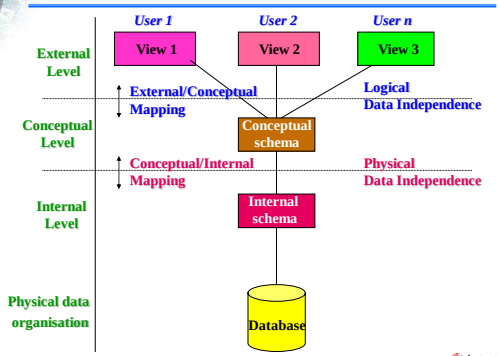
"an architecture developed to separate each user's view of the database from the way the database is physically represented."

- in 1971, 1st generation of database system was introduced by CODASYL DBTG.
- recognized the needs for a two-level approach:-
  a) schema – system view = database structure.
  b) subschema – user views.
- in 1975, ANSI-SPARC (American National Standards Institute Standards Planning & Requirements Committee) introduced three-level approach with system catalog.

- **three-level architecture** :-
  a) ***external level*** - the way users perceive the data.
  b) ***conceptual level*** - provides both the mapping and desired independence between the levels.
  c) ***internal level*** - the way the DBMS and the OS perceive the data.
- **reasons for separation**:-
  a) *all users should be able to access same data.*
  b) *users should not need to know physical database storage details.*
  c) *DBA should be able to change database storage structures without affecting the users' views.*

d) *Internal structure of database should be unaffected by changes to physical aspects of storage*
- example: the changeover to a new storage device.
e) *DBA should be able to change conceptual structure of database without affecting all users.*

ANSI-SPARC Three-Level Architecture



EXTERNAL LEVEL

"Users' view of the database. Describes that part of database that is relevant to a particular user."

- consists of several number of different external views.
- includes entities, attributes and relationship in real world.
- each view can be modeled to one subschema where each subschema will form an external schema.

CONCEPTUAL LEVEL

"Community view of the database. Describes what data is stored in database and relationships among the data."

- contains the logical structure of the entire database as seen by the Database Administrator.
- more to '*what*' data to be stored in the database and relationship between data.
- represents :-
  - all entities, attributes and relationships.
  - the constraints on the data.
  - semantic information about the data.
  - security and integrity information.
- consists only **one** conceptual schema in **one** database and the most **important** schema.

INTERNAL LEVEL

"Physical representation of the database on the computer. Describes how the data is stored in the database."

- "how" data to be stored in the database.
- covers :-
  - physical implementation of the database to achieve optimal runtime performance and storage.
  - data structure and file organizations used to store data on storage devices.
  - interfaces with the OS access method to place the data on the storage devices, build the indexes, retrieve the data and etc.
- internal schema is written in DDL.

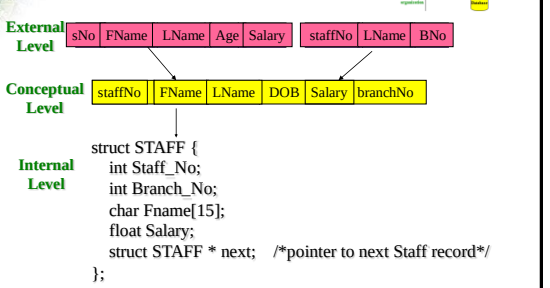
SCHEMA, MAPPINGS & INSTANCES

Schema

"overall description of the database."

- 3 different types of schema :-
  a) ***external schema*** – correspond to different views of data.
  b) ***conceptual schema*** – describes all the entities, attributes and relationships together with integrity constraints.
  c) ***internal schema*** – complete description of the internal model, containing the definition of stored records, the methods of representation, the data fields and the indexes and storage structures used.

Example :-




```
struct STAFF {
   int Staff_No;
   int Branch_No;
   char Fname[15];
   float Salary;
   struct STAFF * next;    /*pointer to next Staff record*/
};
```


Mappings

"relates between schemas."

- 2 types of mapping :-
  a) ***external/conceptual mapping***
  - enables the DBMS to map names in the user's view on to the relevant part of the conceptual schema.
  - if there is any changes on the user's view or conceptual schema, mapping has to be changed.

  b) ***conceptual/internal mapping***
  - enables the DBMS to find the actual record or combination of records in physical storage that constitute a logical record in the conceptual schema, together with any constraints to be enforced on the operations for that logical record.

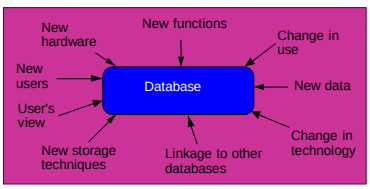
- the schema is specified during the database design process.

```
Student(studno,name,address)
Course(courseno,lecturer)
```

Schema

DATA INDEPENDENCE

"upper levels are unaffected by changes to lower levels."



- 2 kinds of data independence:-
  a) ***Logical data independence***
  b) ***Physical data independence***

LOGICAL DATA INDEPENDENCE

"refers to the immunity of the external schemas to changes in the conceptual schema."

- *changes to the conceptual schema such as the addition or removal of new entities, attributes or relationships should be possible without having to change existing external schemas or having to rewrite application programs.*

PHYSICAL DATA INDEPENDENCE

"refers to the immunity of the conceptual schema to changes in the internal schema."

- *changes to the internal schema such as using different file organizations or storage structures, using different storage devices, modifying indexes or hashing algorithms should be possible without having to change the conceptual or external schemas.*