DESIGN OF DATABASE SYSTEMS
LEVELS OF DATA ABSTRACTION

- The main purpose of a database system is to provide users with an abstract view of the system.
- Database systems are made-up of complex data structures.
- To make it easy for the user to interaction with database, the internal irrelevant details are hidden from the users.
- This process of hiding irrelevant details from user is called data abstraction

There are three levels of abstraction:

i) View level/ External level

- This is the highest level of data abstraction.
- It describes the user interaction with database system i.e describes only part of the database for a particular group of users.
- Can be many different views of a database.

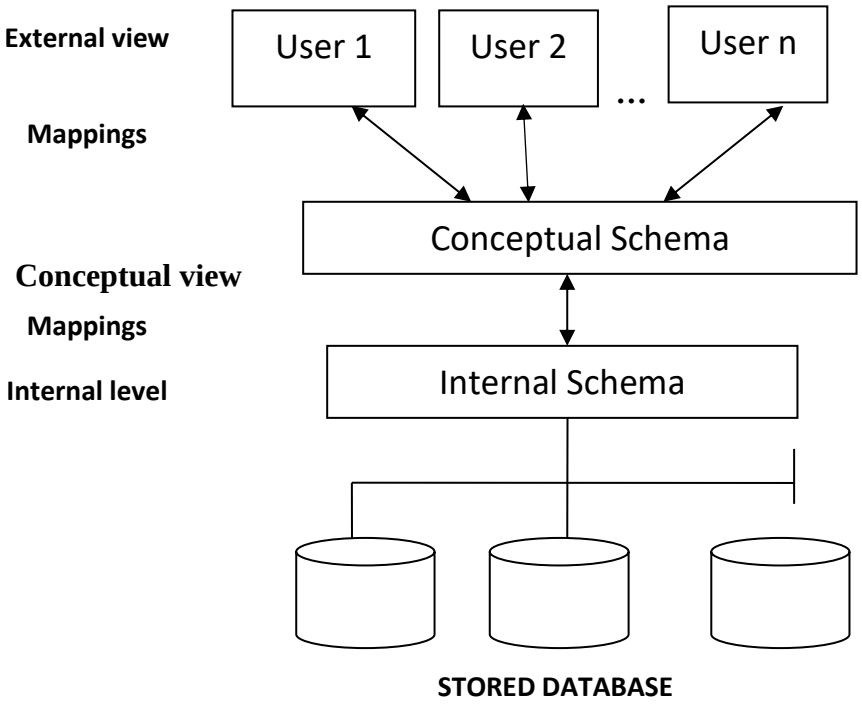
ii) Conceptual level

- This is the middle level of 3-level data abstraction architecture.
- It describes what data is stored in database what relationships exit among those data.

iii) Internal level

- This is the lowest level of data abstraction.
- It is the physical representation of the database on the computer
- It describes how data is actually stored in database.

Given an example that information about staff is to be stored in Staff table, at **external level**, a user just interact with system with the help of GUI and enter the details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them. At the **conceptual level** the records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented. The programmers generally work at this level because they are aware of such things about database systems. At **internal level,** the records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory.



DATABASE SCHEMAS

- A database schema is the overall description of the database.
- There are 3 types of schemas in the database and are defined according to the level of abstraction. These are:

i) At the highest level are the multiple external schemas also called sub-schemas that corresponds to different views of data.
ii) At the conceptual level we have the conceptual schemas
iii) The lowest level is the internal schema which is a complete description of the internal model.

Goal of 3 level/schema of Database

- Every user should be able to access the same data but able to see a customized view of the data.
- The user need not to deal directly with physical database storage detail.
- The DBA should be able to change the database storage structure without disturbing the user's views
- The internal structure of the database should remain unaffected when changes made to the physical aspects of storage.

Advantages Database Schema

- You can manage data independent of the physical storage
- Faster Migration to new graphical environments
- DBMS Architecture allows you to make changes on the presentation level without affecting the other two layers
- It is more secure as the client doesn't have direct access to the database business logic

- In case of the failure of the one-tier no data loss as you are always secure by accessing the other tier

Disadvantages Database Schema

- Complete DB Schema is a complex structure which is difficult to understand for every one
- Difficult to set up and maintain

Mapping

- The DBMS is responsible of schemas, mapping between the 3 levels of schema.
- That is it must check that each external schema is derived from the conceptual schema and it must use the information in the conceptual schema and the internal schema.
- The conceptual schema is related the internal/physical schema through conceptual/internal mapping. This enables the DBMS to find the actual record or combination of records in a physical storage that constitute a logical record in the conceptual schema. It enables the DBMS to identify the constraints to be enforced on the operational data for that logical record.
- It allows any difference in entity names, attributes orders, data type etc. to be resolved.
- The external schema is related to the conceptual schema by the external/conceptual mapping. This enables the DBMS to map names in the users view on to relevant part of the conceptual schema.
- The database schema is specified during the design process and is not expected to change frequently.

Data Independence.

The main objectives for the 3 level architecture is to provide data dependence, which means that upper level are unaffected by changes to lower level. There are of 2 types:

i) Logical data independence

- This refers to the immunity of the external schemas to the changes in the conceptual schema.
- Changes to the conceptual schema such as the addition or removal of new entities, attributes or relationship should be possible without having to change the existing external schemas or having to rewrite application programs.

ii) Physical data independence

- This is the immunity of the conceptual schema to change in the internal schema.
- Changes to the internal schema such as using different file organization or storage structures using different storage devices, modifying indexes should be possible without having to change the conceptual or external schema.

DATABASE SYSTEM DEVELOPMENT LIFECYCLE

1. **Database planning.**

- It involves activities that allow other stages of the database system development lifecycle to be realized efficiently and effectively.

Example:

- Evaluating the current systems to determine existing strengths and weaknesses;
- Identifying enterprise plans and goals with subsequent determination of information system's needs.

2. **System definition.**

- At this stage the scope and boundaries of the database application and the major user views are described.

3. **Requirements collection and analysis.**

- This involves collecting and analysing information about the part of the organization that is to be supported by the database system, and using this information to identify the requirements for the new system.
- Some of the techniques used for gathering this information include:

i) Use of a questionnaire
ii) Interview schedule
iii) Observation e.t.c

4. **Database design.**

- It involves creation of a design that will support the enterprise's mission statement and mission objectives for the required database system.
- Describes desired features and operations in detail, including screen layouts, business rules, process diagrams and other documentation.
- It helps in defining overall system architecture.

Phases of Database Design

Database design is made up of three main phases, namely conceptual, logical, and physical design.

i) Conceptual database design

- The process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- It is the first phase of database design whose purpose is to build a conceptual data model of the data requirements of the enterprise.
- A conceptual data model comprises entity types, relationship types, attributes and attribute domains, primary keys and integrity constraints.
- The conceptual data model is supported by documentation, including ER diagrams and a data dictionary.

ii) **Logical database design**

- The process of constructing a model of the data used in an enterprise based on a specific data model, but independent of a particular DBMS and other physical considerations.
- It is the second phase of database design which results in the creation of a logical data model.
- The objective is to translate the conceptual data model into a logical data model and then to validate this model to check that it is structurally correct and able to support the required transactions.
- Whereas a conceptual data model is independent of all physical considerations, a logical model is derived knowing the underlying data model of the target DBMS. In other words, we know that the DBMS is, for example, relational, network, hierarchical, or object oriented.
- However, we ignore any other aspects of the chosen DBMS and, in particular, any physical details, such as storage structures.
- The steps involved in logical database design include deriving relations for logical data model, validating relations using normalization, validating relations against user transactions, checking integrity constraints, reviewing logical data model with user and checking for future growth.

iii) **Physical database design**

- Physical database design is the third and final phase of the database design process, during which the designer decides how the database is to be implemented.
- Decide how the logical structure is to be physically implemented (as base relations) in the target Database Management System (DBMS).
- The steps involved include translating logical data model for target DBMS, designing file organizations and indexes, designing user views, designing security mechanisms i.e system security and data security.

5. DBMS selection

- The selection of an appropriate DBMS to support the database system.
- A DBMS is selected between the conceptual and logical database design phases.

Approach to 'best' DBMS selection

i) *Define Terms of Reference of study.* The Terms of Reference for the DBMS selection is established, stating the objectives and scope of the study, and the tasks that need to be undertaken.

ii) *Shortlist two or three products.* Decision to include a DBMS product depends on the budget available, level of vendor support, compatibility with other software, and whether the product runs on particular hardware.

iii) *Evaluate products.* There are various features that can be used to evaluate a DBMS product. Weight features and/or groups of features with respect to their importance to the organization, and to obtain an overall weighted value that can be used to compare products.

iv) *Recommend selection and produce report. D*ocument the process and to provide a statement of the findings and recommendations for a particular DBMS product.

6. Application design.

- The design of the user interface and the application programs that use and process the database.
- There are two aspects of application design, namely transaction design and user interface design.

Transaction An action, or series of actions, carried out by a single user or application program, which accesses or changes the content of the database.

User Interface Design Guidelines

- Meaningful title
- Comprehensible instructions
- Logical grouping and sequencing of fields
- Visually appealing layout of the form/report
- Familiar field labels
- Consistent terminology and abbreviations
- Consistent use of colour
- Visible space and boundaries for data-entry fields
- Convenient cursor movement
- Error correction for individual characters and entire fields
- Error messages for unacceptable values
- Optional fields marked clearly
- Explanatory messages for fields
- Completion signal

7. **Prototyping**

- Building a working model of a database system.

8. **Implementation**

- The physical realization of the database and application designs.
- Security and integrity controls for the system are also implemented.

9. **Data conversion and loading.**

- Transferring any existing data into the new database and converting any existing applications to run on the new database.
- This stage is required only when a new database system is replacing an old system.

10. **Testing**

- It is the process of running the database system with the intent of finding errors.
- Testing cannot show the absence of faults; it can show only that software faults are present.
- If testing is conducted successfully, it will uncover errors with the application programs and possibly the database structure.
- Testing demonstrates that the database and the application programs appear to be working according to their specification and that performance requirements appear to be satisfied.

11. Operational maintenance.

- It is the process of monitoring and maintaining the database system following installation.
- It involves:
  - Monitoring the performance of the system. If the performance falls below an acceptable level, tuning or reorganization of the database may be required.
  - Maintaining and upgrading the database system (when required). New requirements are incorporated into the database system through the preceding stages of the lifecycle.

ENTITY-RELATIONSHIP MODELLING

- It is based on a perception over a real world, which consists of a collection of basic objects called entities and relationships among this objects.
- An entity is an object that is distinguished from other objects via a specific set of attributes.

E-R Model Basic Concepts

The model employs the following components:

i) Entity sets
ii) Relationship sets
iii) Attributes

1. Entity sets

- An entity is a thing or object in the real world that is distinguishable from all other objects.
- It may be concrete e.g. a person or a book or it may be abstract e.g. a loan, holiday a concept etc.
- An entity set is a set of entities of the same type that share the same properties e.g. a set of all persons who are customers of a bank.
- An entity is represented in ER diagrams as a rectangular box enclosing the entity type name.

2. Relationship sets

An association between two or more entities is called a relationship.

3. Attributes

They are descriptive properties or characteristics possessed by each member of an entity set.

Characteristics of Attributes

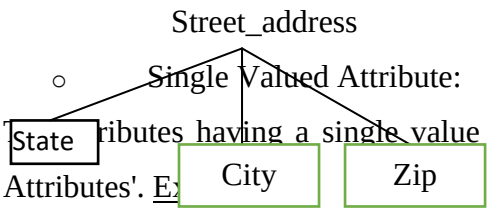
- Simple attributes

The attributes that are not divisible are called as **'simple or atomic attributes'**. Example: first-name, accountNo etc..

- Composite attributes

The attributes that can be divided into smaller subparts, which represent more basic attributes with independent meaning. Example:

Street_address can be divided into 3 simple attributes as State, City and Zip code.



- Single Valued Attribute:

Attributes having a single value for a particular entity are called as 'Single Valued Attributes'. Ex

'Age' is a single valued attribute of 'Person'.

- Muti-Valued Attribute:

The attributes, which are having a set of values for the same entity, are called as 'Multi Valued Attributes'. Example:

A 'College Degree' attribute for a person i.e, one person may not have a college degree, another person may have one and a third person may have 2 or more degrees.

A multi-valued attribute may have lower and upper bounds on the number of values allowed for each individual entity.

- Derived Attributes:

An attribute which is derived from another attribute is called as a ‘derived attribute.

Example: ‘Age’ attribute is derived from another attribute ‘Date’.

- Stored Attribute:

An attribute which is not derived from another attribute is called as a ‘stored attribute.

Example: In the above example,’ Date’ is a stored attribute.

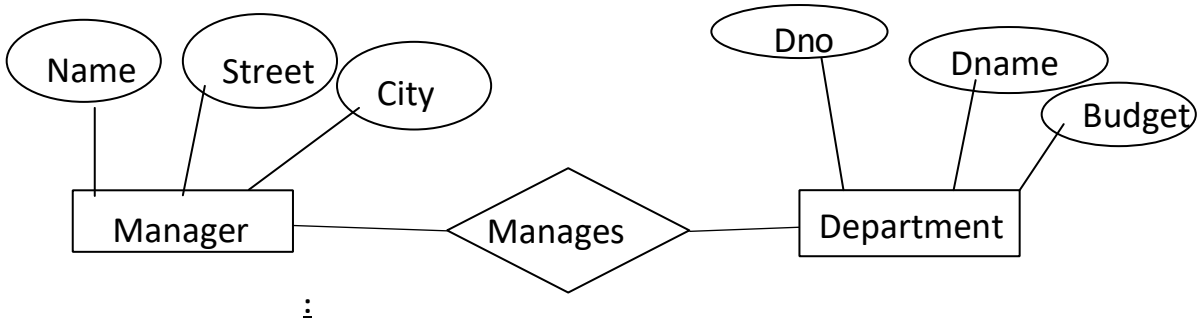
Relationship Sets

A relationship is an association among entities while relationship set is a collection of relationships.

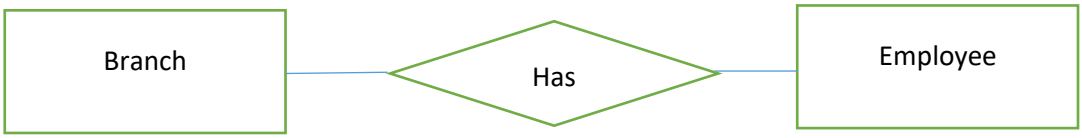
Types of Relationships

i. One to one relationship (1:1) - An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.

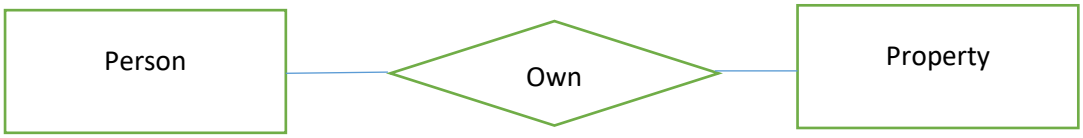
Example: an employee managing a department



ii. One to Many relationship (1:M) - An entity in A is associated with any number of entities in B while an entity in B can be associated with at most one entity in A .



iii. Many to many (M:N) - An entity in A is associated with any number of entities in B and an entity in B can be associated with a number of entities in A.



Components of E-R diagram

(i) Rectangles: - They represent entity sets.

(ii) Ellipses: - represent attributes

(iii) Diamond: - represents relationship sets

(iv) Lines - Link attributes to entities and entity sets to relationship sets

(v) Double ellipses: - represent multi-value attributes

(vi) Dashed ellipses: - denote derived attributes

Exercise

Given that a company that specializes in IT training. The Company has 20 instructors and can handle up to 80 trainees per training session. The Company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.

(a) Identify the main entity types for the company.

(b) Identify the main relationship types and specify the mulplicity for each relationship. State any assumptions you make about the data.

(c) Using your answers for (a) and (b), draw a single ER diagram to represent the data requirements for the company.