Database Management System

Aravinda Kumar V

Session Objectives

- Basic Concepts
- Data Models
- Advantages of Database Approach
- Overview of E-R Modeling
- Data Modeling
- Entity
- Attributes
- Relationships
- Key Concepts
- Normalization
- Basic Normal Forms (restricted up to 3 NF)
- Overview of SQL

Database Management System

- Imagine if you turned on a computer, started the word processor, but could not save a document
- Imagine if you opened a music player but there was no music to play
- Imagine opening a web browser but there were no web pages

Without data, hardware and software are not very useful! Data is the third component of an information system.

Introduction

The goal of many information systems is to transform data into information in order to generate knowledge that can be used for decision making. In order to do this, the system must be able to take data, put the data into context, and provide tools for aggregation and analysis. A database is designed for just such a purpose

Data

Data is raw fact or figures or entity

Data are the raw bits and pieces of information with no Context

DATA: It is a collection of Raw facts.

Example: 101 Raju 2000
102 john 3000

In above example, there no meaningful data such data is known as Raw facts.

Data can be quantitative or qualitative Information

By itself, data is not that useful. To be useful, it needs to be given context, Processed data is called information

The purpose of data processing is to generate the meaningful information

INFORMATION:

It is a collection of meaningful data or processed data.

Example:	EmpID	Ename	Salary
	101	Raju	2000
	102	John	3000

In the above example, there is meaningful data which is in table format which consist of three different fields.

DATA STORE: It is a place where we can store data or information.

1) Books & Papers
2) Flat files
3) Database

FLAT FILES: This is a traditional mechanism which is used to store data or information in individual unrelated files. These files are also called as Flat Files.

Drawbacks of Flat files:
1) Data Retrieval
2) Data Redundancy
3) Data Integrity
4) Data Security
5) Data Indexing

1) Data Retrieval: If we want to retrieve data from flat files then we must develop application program in high level languages, where as if we want to retrieve data from databases then we are using Sequel Language.
SEQUEL (Structured English Query Language)

2) **Data Redundancy**: Sometimes we are maintaining multiple copies of the same data in different locations this data is also called as Duplicate data or Redundant data. In Flat files mechanism when we are modifying data in one location it is not effected in another location. This is called INCONSISTENCY.

Cont…

In databases, every transaction internally having 4 properties. These properties are known as
ACID properties. ACID Properties:
A mean Atomicity (ROLLBACK)
C mean Consistency
I mean Isolation
D mean Durability (COMMIT)
These properties only automatically maintains consistent data in databases

3) **Data Integrity**: Integrity means to maintain proper data. If we want to maintain proper data then we are defining set of rules, these rules are also called as " Business rules". In databases, we are maintaining proper data using 'constraints', 'triggers'. If we want to maintain proper data in flat files we must develop application programs in high level languages like COBOL, JAVA, ETC….

4) **Data Security**: Data stored in flat files cannot be secured because flat files doesn't provides security mechanism. Whereas databases provides "ROLE based security".
5) **Data Indexing**: If we want to retrieve data very quickly from database then we are using indexing mechanism. Whereas flat files doesn't provide indexing mechanism.

To overcome all the above problems, a new software used by all organization to store data or information in secondary storage devices. This is called DBMS software.

DATABASE:

It is an organized collection of interrelated data used by application program in an organization. Once data stored in database it can be shared by number of users simultaneously and also this data can be integrated.

Database

Database

A database is an organized collection of related information. It is an organized collection, because in a database, all data is described and associated with other data. All information in a database should be related as well

A database that contains information about students should not also hold information about company stock prices

DBMS (DATABASE MANAGEMENT SYSTEM):

It is a collection of programs (S/W) written to manage database.
Example: ORACLE, FOXPRO, DB2, TERADATA, SQLSERVER, SYBASE, MYSQL, INGRESS, INFORMIX, SQLLITE…… etc;
In file based approach, every application program in the organization maintain its own file separate from other application program.

Database Management System

Database Management System (DBMS) is a collection of programs that enable users to create, maintain database and control all the access to the database. The primary goal of the DBMS is to provide an environment that is both convenient and efficient for user to retrieve and store information.

Benefits of DBMS

Data Redundancy

In the Database approach, ideally each data item is stored in only one place in the database
However, in some case redundancy is still exists to improving system performance, but such redundancy is controlled and kept to minimum

Data Sharing

The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data

Data Independence

The system data descriptions are separated from the application programs. Changes to the data structure is handled by the DBMS and not embedded in the program.

Benefits of DBMS

Providing multiple views of data

A view may be a subset of the database. Various users may have different views of the database itself.
Users may not need to be aware of how and where the data they refer to is stored

Restricting Unauthorised Access

Not all users of the system have the same accessing privileges.
DBMS provides a security subsystem to create and control the user accounts

Providing backup and recovery facilities

If the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the program started executing.

Functions of DBMS

- Data Definition
- Data Manipulation Security &
- Data Integrity Recovery and
- Data Concurrency
- Data Dictionary Maintenance

Architecture of DBMS

A commonly used views of data approach is the three-level architecture suggested by ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee)

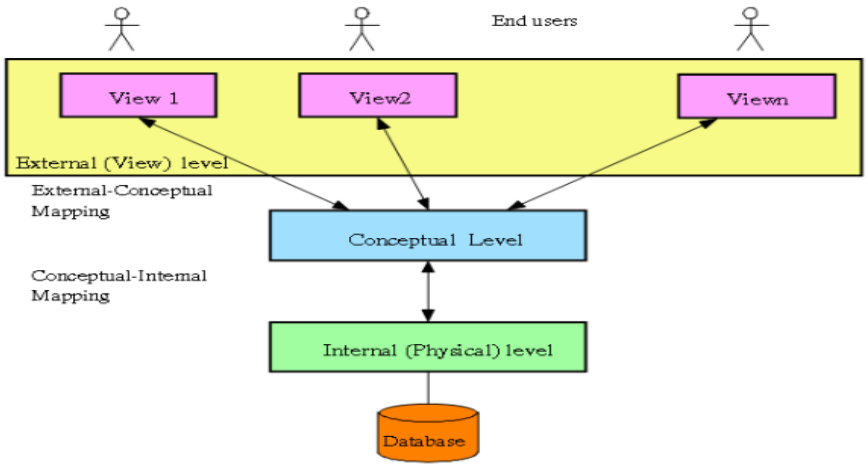
Three – level architecture for database system is proposed to archive the characteristics of the database approach. The goal of this architecture is separate the applications and the physical database so the actual details of how data is organized are hided from the users

Architecture of DBMS

The three levels of the DBMS architecture are:

- **External Level - individual user view**
- **Conceptual/Logical level - community user view**
- **Internal Level- physical or storage view**

Architecture of DBMS



External - individual user view

- In this highest level, there exists a number of views which of is defined a part of the actual database
- Each view is provided for a user or a group of users so that it helps in simplified the interaction between the user and system

Conceptual - community user view

This level describes the logical structure of the whole database.

The entire database is described using simple logical concepts such as objects, their properties or relationships. Thus the complexity of the implementation detail of the data will not be available for users.

Internal - physical or storage view

This level describes how the data are actually stored, how to access the data?

The internal level involves how the database is physically represented on the computer system. It describes how the data is actually stored in the database and on the computer hardware

Schema

A schema is an overall description of a database, and it is usually represented by the entity relationship diagram (ERD).

External schemas: multiple and multiple subschemas also, these display multiple external views of the data

Conceptual schema: there is only one. This schema includes data items, relationships and constraints, all represented in an ERD.

Physical schema: there is only one

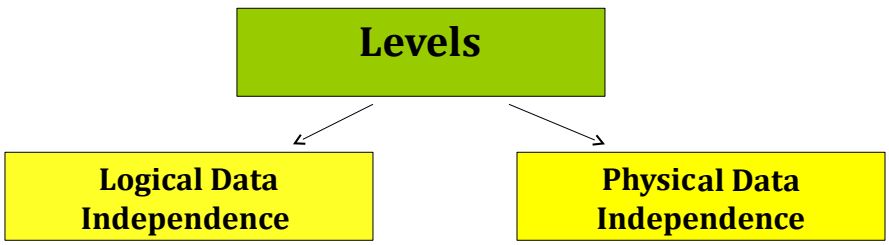
DBMS architecture provides "DATA INDEPENDENCE".

Data Independence: Upper levels are unaffected by changes in the lower levels is called as "Data Independence".

DBMS architecture have two types of Data Independences:
1) Logical Data Independence
2) Physical Data Independence

Data Independence

Data Independence is the ability to modify the schema in one level without affecting the schema in the higher level



DATA INDEPENDENCE

1) **Logical Data Independence:** Changes to the conceptual level do not required to change to the external level this is called Logical Data Independence.
Example: Adding a new entity in conceptual level does not effect in external level.
2) **Physical Data Independence:** Changes to the internal level do not required to changes in conceptual level. This is called Physical Data Independence.
Example: Adding an index to the internal level it is not affected in conceptual level.

Data Independence:

Logical Data Independence

Logical data independence is the ability to make change in the conceptual schema without causing a modify in the user views or application program

For example, the addition or removal of new entities, attributes or relationships to this conceptual schema should be possible without having to change existing external schemas or rewrite existing application programs.

Changes to the logical schema (e.g., alterations to the structure of the database like adding a column or other tables) should not affect the function of the application (external views)

Data Independence:

Physical Data Independence

Physical data independence is the ability to make change in the internal schema without causing a modify in the conceptual schema or application program

Physical data independence seem to be easier to achieve since the way the data is organized in the memory affect only the performance of the system

The logical schema stays unchanged even though changes are made to file organization or storage structures, storage devices or indexing strategy

Data Model

Data Model is a collection of concepts that can be used to describe the structure of database. Structure of database means data types, relationships and constraints

Data Modeling

Data modeling is the first step in the process of database design. This step is sometimes considered to be a high-level and abstract design phase, also referred to as conceptual design. The aim of this phase is to describe:

- The data contained in the database (e.g., entities: students, lecturers, courses, subjects)
- The relationships between data items (e.g., students are supervised by lecturers; lecturers teach courses)
- The constraints on data (e.g., student number has exactly eight digits; a subject has four or six units of credit only)

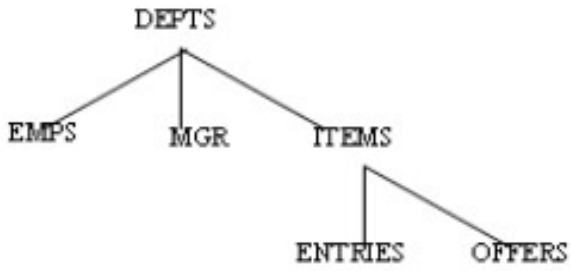
Data Models

The most common data models are:

- **Relational Model**
- **Network Model**
- **Hierarchical Model**

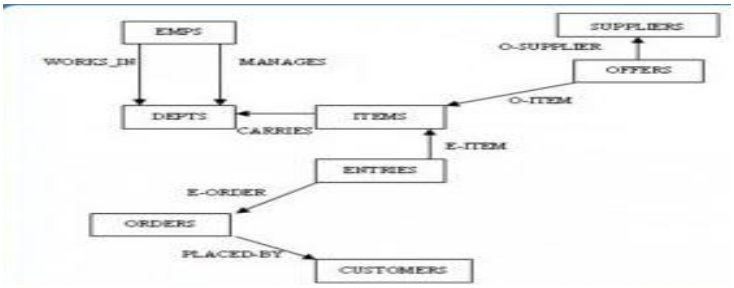
Hierarchical Model

The hierarchical database stores data in a series of records. These records have a set of field values attached to them. All instances of a specific record are collected together as a "record type". It uses "Parent Child Relationships" to create links between record types. It does this by using trees



Network Model

The network model represents data as record types. The relationships between data are represented by links. The network database model is very similar to the hierarchical model. However, instead of using a single-parent tree hierarchy, the network model uses set theory to provide a tree-like hierarchy with the exception that the child tables were allowed to have more than one parent



Relational Model

The relational model represents data as relations, or tables. Relationships between data items are expressed by means of tables. It is a database that maintains a set of separate, related files (tables), but combines data elements from the files for queries and reports when required

Example:
Relational database comprising of two tables
Customer –Table
Account –Table

Relational Model

Customer –Table

Customer Name	Customer ID	Address	City	Account Number
Amit Kumar	345678	Sector 8	Gurugram	K-100
Mohit Dutta	345679	Dwarka	Delhi	K-302
Pawan Singh	345122	Salt Lake	Kolkata	K-113
Pooja Garg	345103	First Lane	Agra	K-201
Amy	345221	3/114 C	Chandigarh	K-415

Account –Table

Account Number	Balance
K-100	5000.00
K-302	12000.00
K-113	3500.00
K-201	250000.00
K-415	12000.00

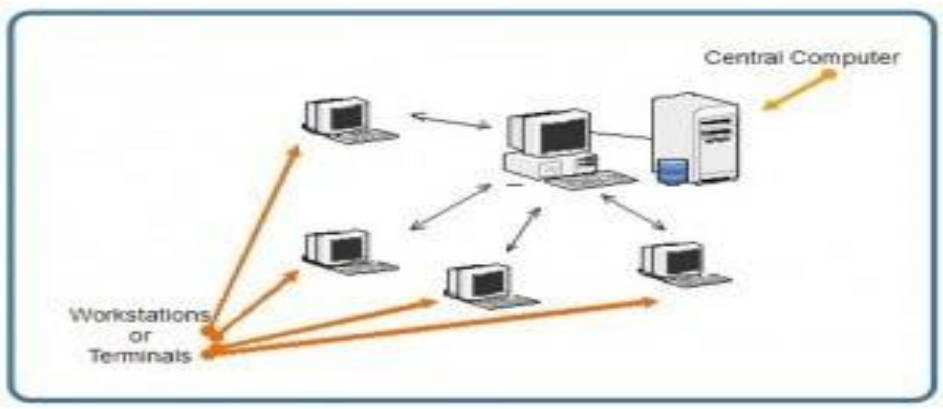
Database Management Systems: Classification

- **Classification Based on Data Model**
  - Relational Model
  - Network Model
  - Hierarchical Model
- **Classification Based on User Numbers**
  - Single-user database system
  - Multi user database system
- **Classification Based on Database Distribution**
  - Centralized systems
  - Distributed database system
    - Homogeneous
      use the same DBMS software from multiple sites
    - Heterogeneous
      different sites might use different DBMS software

Database Management Systems:
Classification Based on Database Distribution

Centralized systems

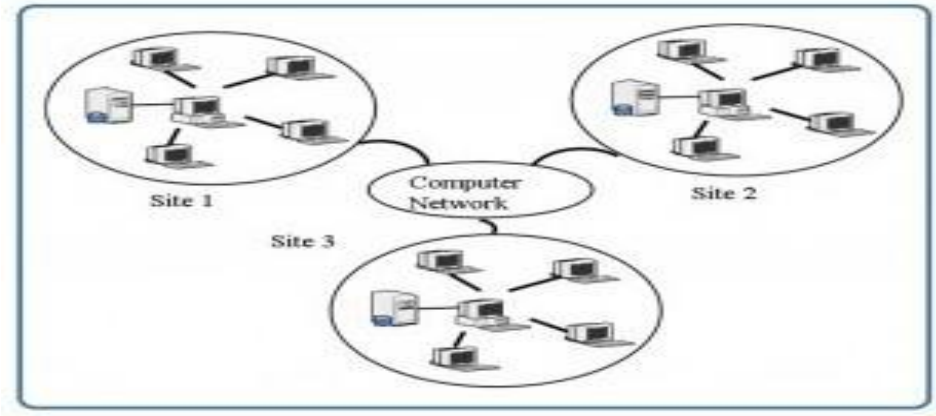
DBMS and database are stored at a single site that is used by several other systems too



Database Management Systems:
Classification Based on Database Distribution

Distributed database system

The actual database and the DBMS software are distributed from various sites that are connected by a computer network



Overview of ER Modeling (The Entity Relationship Data Model)

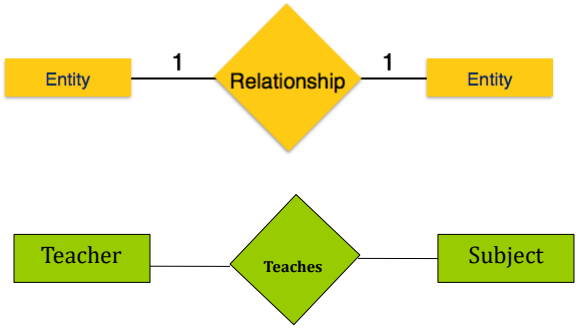
ER models, also called an ER schema, are represented by ER diagrams

ER modeling is based on two concepts:

- Entities, defined as tables that hold specific information (data)
- Relationships, defined as the associations or interactions between entities

Overview of ER Modeling (The Entity Relationship Data Model)

- Entity
- Attributes
- Relationship



Entity

An entity is an object in the real world with an independent existence that can be differentiated from other objects. An entity might be

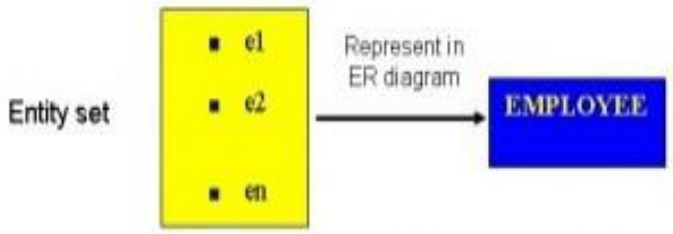
- ***An object with physical existence (e.g., a lecturer, a student, a car)***
- ***An object with conceptual existence (e.g., a course, a job, a position)***

An entity is something of importance to a business or organization. When the entity is something an organization or business wants to store, it becomes a database table

Entity Set

- An entity set is a collection of entities of an entity type at a particular point of time

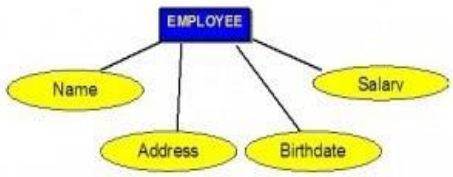
For example, an Employee set may contain all the employees of an organisation



Attributes

- Each entity is described by a set of attributes (e.g., Employee = (Name, Address, Birthdate, Salary).
- Each attribute has a name, and is associated with an entity and a domain of legal values

For example, a employee's name cannot be a numeric value. It has to be alphabetic. An employee's age cannot be negative, etc.



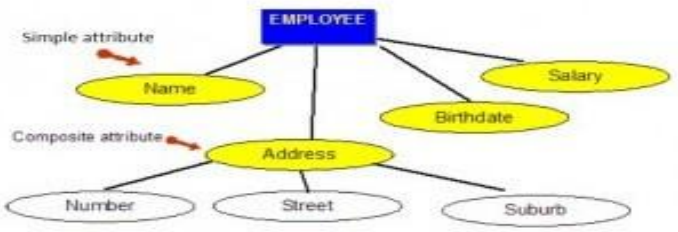
Types of Attributes

- **Simple attribute** – Simple attributes are those drawn from the atomic value(cannot be divided further) domains; they are also called single-valued attributes.

  For example, an Employee's phone number is an atomic value of 10 digits.

- **Composite attribute** – Composite attributes are made of more than one simple attribute.

  For example, Address may consist of House Number, Street and Area



Cont…

Field, Record & File

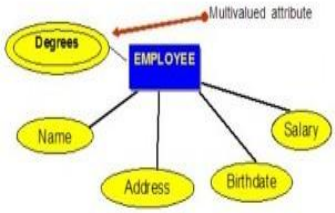
In a database table, a field is a data structure for a single piece of data. A combination of fields makes up a table, which contain all the information within the table relevant to a specific entity. A record is one instance of a set of fields in a table. Multiple data records make up a table or data file. Files store data information

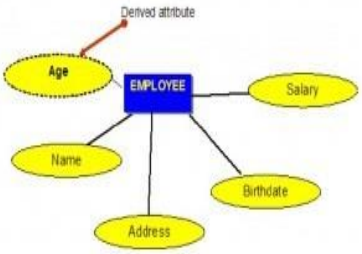
For example, in a table called Student, the Enrollment number would likely be a field in a row that would also contain other fields such as Student Name, Class, Address and Pin Code. The records make up the table rows and the fields make up the columns

Types of Attributes

- **Derived attribute** – Derived attributes are attributes that contain values calculated from other attributes. These don't exist in the physical database
  For example, Age can be derived from the attribute Birthdate

- **Multi-value attribute** – Multivalued attributes are attributes that have a set of values for each entity.
  For example, an employee can have more than one Degree, phone number, email address, etc.





Relationship

- Association between entities
- The glue that holds the tables together
- Used to connect related information between tables.

For example, a teacher works at a department, a student enrolls in a course. Here, Works at and Enrolls are relationships.

Relationship

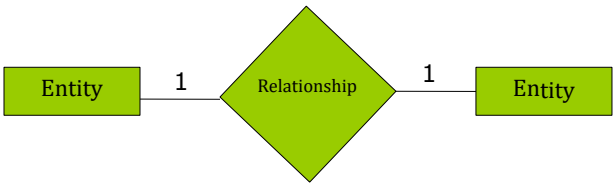
Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (represented as rectangles) participating in a relationship, are connected to it by a line.

Relationship

There are three different relationships between entities

- **One-to-one** – This exists when one entity is related to only one other entity of a particular type.

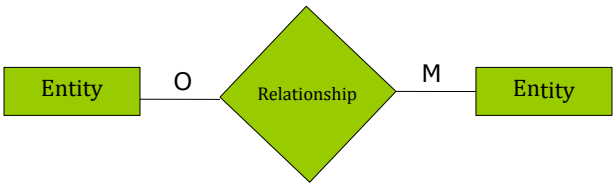
For example, a business has multiple stores, each store location has a single manager, the relationship between store and manager is 1:1



Relationship

- **One-to-many –** These are the most common relationships and exist when one entity can be related to more than one other entity

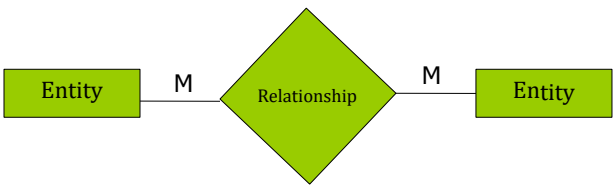
For example, if a supplier supplies more than one product to the company, the relationship between the supplier and the products is O:M



Relationship

- **Many-to-Many** – This relationship exists when one entity can be related to multiple entities of the same type of the original entity

For example, if an order can contain multiple products and one product can appear on many orders, the relationship between orders and products is M:M



Keys

An important constraint on an entity is the key. The key is an attribute or a group of attributes whose values can be used to uniquely identify an individual entity in an entity set.

Types of Keys

- **Candidate**
- **Composite**
- **Primary**
- **Foreign**
- **Alternate**

Types of Keys

Candidate Key

A candidate key is a simple or composite key that is unique. It is unique because no two rows in a table may have the same value at any time

Composite Key

It consists of more than one attribute to uniquely identify rows (tuples) in a table is called composite key

Keys

- **Primary Key**
  The primary key is a candidate key that is selected by the database designer to be used as an identifying mechanism for the whole entity set. It must uniquely identify tuples in a table. A table cannot have more than one primary key

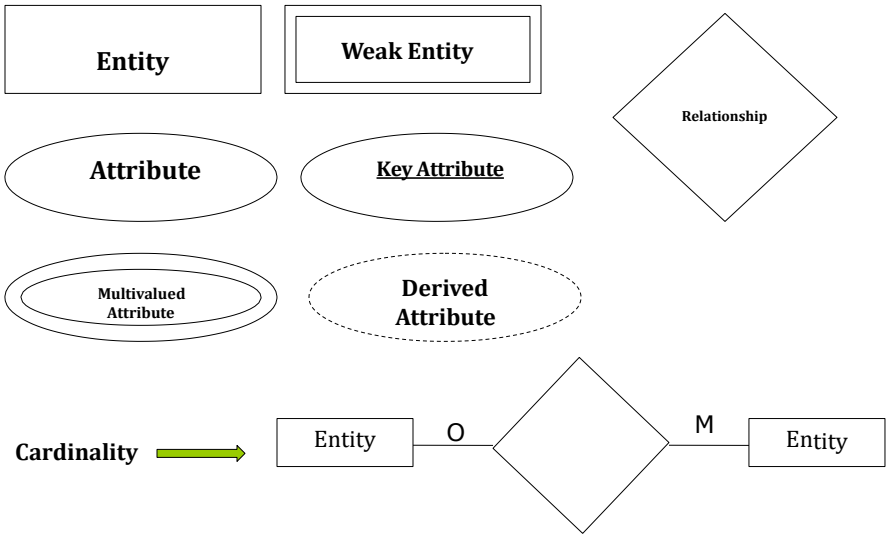
- **Foreign Key**
  A foreign key (FK) is an attribute in a table that references the primary key in another table. They act as a cross-reference between tables

- **Alternate Key**
  Alternate keys are all candidate keys not chosen as the primary key.

ER Modeling Steps

- Identify the entities
- Find relationships
- Identify the key attributes for every entity
- Identify other relevant attributes
- Complete E-R diagram
- Review

ER Diagram Notations



Normalization

- An entity relation diagram (ERD) provides the big picture, or macro view, of an organization's data requirements and operations. This is created through an iterative process that involves identifying relevant entities, their attributes and their relationships

- **Normalization procedure focuses on characteristics of specific entities and represents the micro view of entities within the ERD**

Normalization

Normalization is basically to design a database schema such that duplicate and redundant data is avoided. If some piece of data is duplicated several places in the database, there is the risk that it is updated in one place but not the other, leading to data corruption.

Normalization levels:

1NF

2NF

3NF

Each level describes how to get rid of some specific problem, usually related to redundancy.

Normalization

Normalization is a method or database design technique which organizes tables in a perfect way that reduces redundancy and dependency of data. It divides larger tables to smaller tables and links them using relationships.

Database with normalization errors means the risk of getting invalid or corrupt data into the database

Normalization is a step by step process for transforming a table into simpler tables, by removing a certain type of dependency at each step. Normalization is usually performed when a table is being designed

Normalization

Imperfect database design may contain following anomalies:

- **Update anomalies**

  anomalies that cause data inconsistencies

- **Deletion anomalies**

  unintended loss of data

- **Insert anomalies**

  inability to add data

Normalization prevents data anomalies and bring the database to a consistent state.

Normalization: Example

Video Library

Customer Name	Address	Movies Rented	Salutation	Category
Amit Jain	Sector 4	DDLJ, Don	Mr.	Romantic, Action
Yash Dev	Model Town	Karz, Bazigar	Mr.	Drama, Thriller
Yash Dev	5th Avenue	Yarana	Mr.	Drama

Movies Rented Column has multiple values

Normalization Levels

In the first normal form, only single values are permitted at the intersection of each row and column; hence, there are no repeating groups. Each record needs to be unique

Table: Customer

Customer Name	Address	Movies Rented	Salutation	Category
Amit Jain	Sector 4	DDLJ	Mr.	Romantic
Amit Jain	Sector 4	Don	Mr.	Action
Yash Dev	Model Town	Karz	Mr.	Drama
Yash Dev	Model Town	Bazigar	Mr.	Thriller
Yash Dev	5th Avenue	Yarana	Dr.	Drama

Normalization Levels

For the *second normal form*, the relation must first be in 1NF and there should be single column Primary Key.

Membership ID	Customer Name	Address	Salutation
1	Amit Jain	Sector 4	Mr.
2	Yash Dev	Model Town	Mr.
3	Yash Dev	5th Avenue	Dr.

Membership ID	Movies Rented
1	DDLJ
1	Don
2	Karz
2	Bazigar
3	Yarana

Normalization Levels

Membership ID	Customer Name	Address	Salutation
1	Amit Jain	Sector 4	Mr.
2	Yash Dev	Model Town	Mr.
3	Yash Dev	5th Avenue	Dr.

Normalization Levels

To be in *third normal form*, the relation must be in second normal form. Also all transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute.

Membership ID	Customer Name	Address	Salutation ID
1	Amit Jain	Sector 4	1
2	Yash Dev	Model Town	1
3	Yash Dev	5th Avenue	2

Membership ID	Movies Rented
1	DDLJ
1	Don
2	Karz
2	Bazigar
3	Yarana

Salutation ID	Salutation
1	Mr.
2	Dr.

Overview of SQL

- SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems
- SQL is a descriptive, entity-oriented query language for data manipulation
- Today SQL is used either as a stand-alone programming language or within other languages like C, C++, Java, ADA, COBOL, FORTRAN, PL/1, PASCAL etc

Overview of SQL

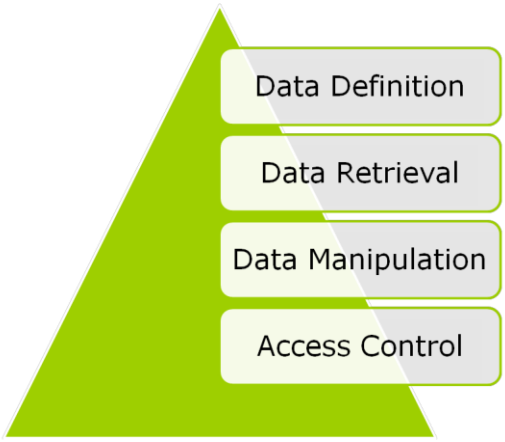
- Structured Query Language (SQL) is a database language designed for managing data held in a relational database management system
- SQL was initially developed by IBM in the early 1970s and the initial version was SEQUEL(Structured English Query Language)
- In the late 1970s, Relational Software Inc., which is now Oracle Corporation, introduced the first commercially available implementation of SQL

Overview of SQL

Some common relational database management systems that use SQL

Overview of SQL

SQL statements are used to perform following tasks



Overview of SQL

SQL language is divided into four types of primary language
statements: **DML, DDL, DCL** and **TCL**

These statements are used
to:

- Create the database and table structures
- Perform basic data management chores (add, modify) delete and
- Perform complex queries to transform raw data into useful
- information

  Control which user can read/write data or manage transactions

Overview of SQL

- The four main categories of SQL statements are as follows:

- **DML (Data Manipulation Language)**
- **DDL (Data Definition Language)**
- **DCL (Data Control Language)**
- **TCL (Transaction Control Language)**

SQL Commands

- **DML statements include**

SELECT – select records from a table
INSERT – insert new records
UPDATE – update/Modify existing records
DELETE – delete existing records

- **DDL Statements include**

CREATE – create a new Table, database, schema
ALTER – alter existing table, column description
DROP – delete existing Table from database

SQL Commands

- **DCL statements control the level of access**

GRANT – allows users to read/write on certain database objects
REVOKE – revoke users' read/write permission on database objects

- **TCL statements allow us to control and manage transactions**

COMMIT – commits a transaction
ROLLBACK – Rollback a transaction