Unit:-1

Introduction To Database Management System

By:- @curious_programmer

* Introduction:-

- In today's world as the information technology has changed rapidly, many computing applications deal with large amount of information regularly.

- As the end user applications has changed significantly in last few decades, there is a challenge to store the large amount of information, retrive and manage this Information in timely manner.

- This can be achieving today by making use of services of Database Management system (DBMS).

* What is Database Management System:-

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc.

These are a very popular commercial database which is used in different applications.

- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.

- It provides protection and security to the database. In the case mutiple users, it also maintains data consistency.

- If the data has to be shared among number of users there are highly chances that the data might not remain consistent because too many users might try to access it at same time and may try to change the value.

- The DBMS must ensure that the chances of getting anomalous results when the Data is used by more than one set of user.

DBMS systems can be used extensively in the following fields:

Transportation
Education
Banking
Sales
Manufacturing
Human Resource

* History of Database System:-

The following are the historical perspective of DBMS system:

- In Early 1960s, the first general purpose DBMS was designed by Charles Bachman at General Electric, which was Later, called as IDS (Integrated Data store).

- This IDS formed groundwork for introduction of Network Data Model, which was later, standardized by CODASYL (Conference on Data Systems Languages).

- In late 1960s, IBM Developed the IMS (Information Management System) which was widely used.

- This IMS formed groundwork For Introduction of Hierarchical Data Model.

- By the joint venture of of IBM and American Airlines, the SABREE was launched which help the people to reserve the tickets.

- The New Data representation Framework was Initialy Launched by Edgar codd, called the Relational Data Model.

★ Purpose of the Database System

- A database system provides a data defination Language to specify the database schema and a data manipulation language to express database queries & updates.

- One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data.

- To see why database management system is necessary, Let us look at a typical "file processing system" supported by conventional operating system.

e.g.

The application is a saving bank:-

- Saving account & customer records are kept in permanent system files.

- Application programs are written to manipulate files to perform following task.

1. To debit or credit an account.

2. To add a new account

3. To find the account balance

4. To generate the monthly statements

- System programmers wrote these application programs to meet the needs of the bank.

- File system has several disadvantages and the following problems are associated with file system:

1. Data redundancy and inconsistency:

- The major problem with file processing system is that it maintains several versions of same file i.e; duplication of data is possible at multiple places.

- Also there are several copies of files are stored, if any one of the file is changed, the different versions of same file may not be updated which leads to in consistency of data.

2. Difficulty in accessing the data

- Consider the airline reservation system. If the senior management of company wants to access the information of all its customers who are living in the same postal code, it has to be done manually because current file processing system does not allow the user to obtain this

Information.

- So in the above case, there are two options. Either the application programmer has to write a new application program to satisfy the usual request or could get this information manually.

- If a query changes, a new application program should be written to get the needed information.

3. Data isolation:-

- One of the major problems with the file system is that the data is scattered and stored in multiple locations and in different formats.

- Hence in order to retrive the needed information from multiple location and in different formats is a very difficult to proceed with the help of application program.

4. Concurrent access anomalies:-

- In Order to speed up the performance of the system and faster response to applications, many systems allow the user to update the data

concurrently.

- Suppose two users located at different locations wants to book the tickets, there might be situation that both of the people will be given the same seat because the data is stored in multiple locations and both of them will be given a seat from individual copy of the data.

- Therefore there should be some protection mechanism to avoid this concurrent updates.

5. Security problems:-

- Every user in this system should be able to access the data which he is allowed to access and not all the data.

- For example, the salesperson in an organization should be allowed to access the data related to him and should not be allowed to access data which is used HR team or finance department in an organization.

6. Integrity problems:-

- Data stored in the database should be allowed to satisfy certain constraint checking.

- For e.g. before adding a new employee in the Employee table, if we check the age of the employee and if we apply constraint such that only these employee whose age is grater than 18 years should be allowed to enter in the table which means that before the new data is inserted the age of the employee should be calculated.

7. Atomicity problems:-

- Every Application System is assumed to fail at some point in near future.

- In many applications, if the system fails, the data should be rolled back to the state before the failure occurs.

- Consider the customer is withdrawing some cash from the ATM machine from his own accound and if the faliure happens in the system, it should not happen that the amount is deducted from customer

account but the customer is not getting any cash from the machine.

- Another disadvantage with file processing system is that it becomes difficult to ensure atomicity.

✱ Advantages And Disadvantages of Database Systems :-

The DBMS is preferred ever the conventional file processing system due to the following advantages :-

1. Controlling Data Redundancy :-

- In the conventional file processing system, every user group maintains its own files for handaling its data files. This may Lead to:

- Duplication of same data in different files.

- Wastage of storage space, since duplicated data is stored.

- Time in entering data again and again is wasted.

2. Elimination of Inconsistency:

- Let Us consider the following example of student:

- Imagine that a particular student has opted for embedded system as one of the elective subject in sem-V for TYBSc IT sem V examination while filling up the examination from.

-If, after getting hallticket ~~that~~ the rather than expecting Embedded system as the choice of elective subjet in the hall ticket, if some other subject is highlighted, it means that the data for that sudden has not correctly inserted in the database.

- On centralizing the database the duplication will be controlled and hence inconsistency will be removed.

3. Better service to the users:

- A DBMS is often used to provide better service to the users.

- In conventional system, availability of Information is often poor, since it normally difficult to obtain information in a timely

manner because our existing systems are not capable to produce the same.

- Centralizing the data in the database also means that user can obtain new and combined information easily that would have been impossible to obtain otherwise.

- Also use of DBMS should allow users that don't know programming to interact with the data more easily, unlike file processing system where the programmer may need to write new programs to meet every new every new demand.

4. Flexibility of the System is Improved:

- Since changes are often necessary to the contents of the data stored in any system, these changes are made more easily in a centralized database than in a conventional system.

- Application programs need not to be changed on changing the data in the database.

5. Integrity can be improved:

- Since data of the organization using database approach is centralized and would be used by a number of users ata time, it is essential to enforce integrity-constraints.

For Example:-

The Example of Hall Ticket Generation system that we have already discussed, since multiple files are to maintained, so sometimes you may enter a value for subject may not exist. Suppose Elective Subjects can have values but we enter a value 'Mathematics-1' For it, it may lead to database inconsistency.

- Even if we centralized the database it may still contain incorrect data For example

- Salary of full time clerk may be entered as 1500 rather than Rs 4500

- The above problems can be avoided by defining the validation procedures whenever any update operation is attempted.

6. Standards can be enforced:-

- Standards are easier to enforce in database system because all the data in database is access through centralized DBMS.

- Here we can relate to the naming of data, structure of data, format of the data etc.

- Standardizing stored data formats is usually desirable for the purpose of data interchange or migration between systems.

7. Security can be improved:-

- On conventional systems, applications are developed in an adhoc manner.

- Often different system of an organization would access different components of the operational data, in such an environment enforcing security can be quiet difficult.

- Setting up of a database makes it easier to enforee security restrications since data is now centralized.

- For example, a clerk may be given the authority to know only the names of all the customer who have a loan in bank but not the details of each loan the customer may have.

- This can be accomplished by giving the privileges to each employee.

8. Organization's requirement can be easily identified :-

- All organization have sections and departments and each of these units often consider the work of their unit as the most important and therefore consider their need as the most important.

- Once a database has been setup with centralized control, it will be necessary to identify organization's requirement and to balance the needs of the competition units.

- For example,— a DBA must choose best file structure and access method to give fast response for the high critical application as compared to less critical applications.

9. Data Model must be developed:-

- Perhaps the most important advantage of setting up of database system is the requirement that an overall data model for an organization be build. In conventional system, it is more likely that files will be desined as per need of particular applications demand.

- The overall view is often not considered. Building an overall view of an organization's data is usual cost effective in the long terms.

10. Provides backup and Reqovery:-

- Centralizing a database provides the schemes such as recovery and backups from the failures including disk crash, power faliures, software errors which may help the database to recover from the inconsistent state to the state that existed prior to the occurrence of the faliure, though methods are very complex.

★ Disadvantages of Database systems :-

The following are the disadvantages of Database systems

1. Database complexity

The design of the database system is complex, difficult and is very time consuming task to perform.

2. Substantial hardware and software start-up costs.

Huge amount of investment is needed to setup the required hardware and the software needed to run those applications.

3. Damage to database affects virtually all applications programs

If one part of the database is corrupted or damaged because of the hardware or software faliure, since we don't have many versions of the file, all the application programs which are dependent on this database are implicity affected.

4. Extensive conversion cost in moving from a file-based system to a database system

If you are corrently working on fire based system and need to upgrade it to database system, then large amount of cost is incurred in purchasing different tools, adopting different techniques as per the requirement.

5. Initial traning reduired for all programmers and user.

Large amount of humans efforts, the time and cost is needed to train the end users and application programmers in order to get used to the database systems.

Introduction To Relational Database Management Systems

By: Curious_.programmer

Introduction To Relational Database Management System:

- A relational DBMS is special software that is used to manage the organization, storage, access, security and integrity of data.

- This specialized software allows application system to focous on the interface, data validation and screen navigation.

- When there is a need to insert, modify, delete or display data, the application system simply makes a "call" to the RDBMS

- Although there are many different types of database management systems, relational database are by far the most common.

- Other types include hierarchical database and network databases.

- A relational DBMS stores information in a set of "tables", each of which has a unique identifier or "primary key".

Rdbms are widely used in real life applns such as:

1. Airlines:
   It can be used to keep the status of the flights and schedules and for reservation and cancellation of tickets.

2. Banking:
   It is useful in storing the customer information, account details, loan details and banking transactions.

3. Universities:
   It is useful in storing the student information, coarse registrations, grades etc.

The Relational Model

- The relational model is a collections of relations required to build a database. Informally, each relation resembles a table of values or, to some extent, a "flat" file of records.

- In relational model, each row in the table consists of set of related data values.

- In this model, each row in the table shares some reality which corresponds to the real world entity or relationship.

- Every table and the columns present in the table is given a unique table name and column names which can be used to extract the relevant values from the tables.

Consider the IDOLSYIT table given below:

Relation

Table name: IDOLSYIT

Domain

Attributes

S_ID (Int)	S_NAME (String)	Contact no (Int)	Email (string)
1001	Amit	5253461512	amit@gmail.com
1002	Akash	16 1913 9246	rakesh@hotmail.com
1003	Devang	3816761521	devang@yahoo.com
1004	Ritesh	4523689713	om@rediffmail.com
1005	Rakesh	982365 7412	adit@gmail.com

Tuples

The relationship betn domains, attribute, tuples and Relation.

- A row of records in the given table is called as tuple. In the above example, the individual records for students starting with 1000 to 1005

- In above example, the individual columns are called as attributes of the system

- The table itself is called as the relation.

- The data type describing the types of values that can appear in each column is called a domain.

- On the above example, if we define the relation schema it would look like this

  IDOLSYIT (S-ID:integer, S_Name:String, contact no: integer, email:string)

- In a table, if every row in a given table is different from all other rows is called the relational in mathematical terms.

- The term relational database come because relations are the base of a relational model.

- A column in one table whose value matches with the primary key of another table is called as a foreign key of the table.

Introduction To SQL

- SQL is a standard computer database programming language and its popularity has popularity has explored since past two decades.

- It is portable language which supports right from mainframe system to personal computers and even to hand held devices.

- Today most of company's software products lie on SQL for its data management and SQL is the nucleus of database products from Microsoft and Oracle, two of the Largest software companies in the world.

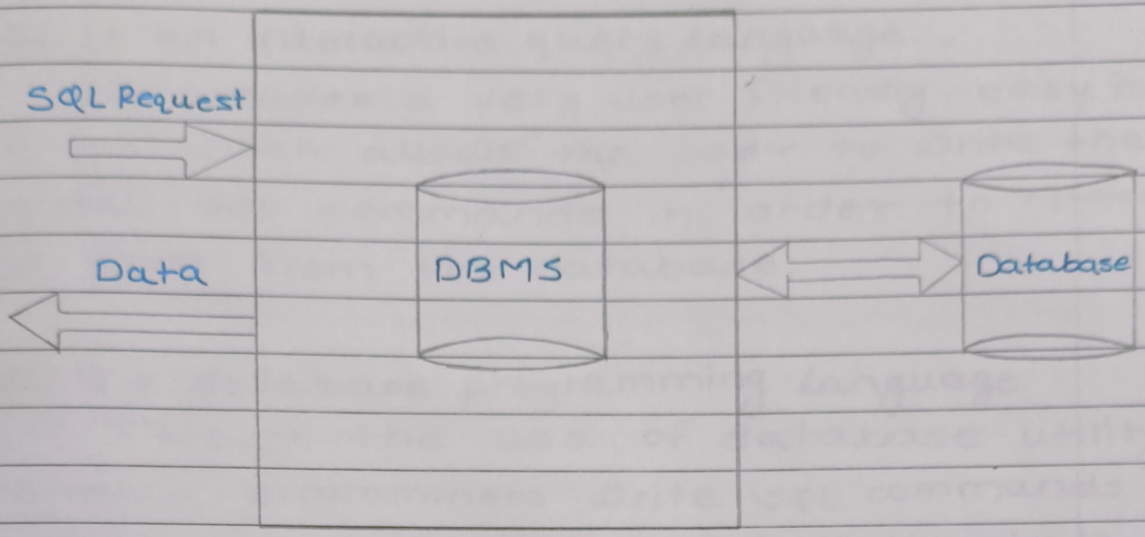
- The journey of SQL is a tremendous right from the beginning as as IBM research project; SQL has become a powerful market force.

- SQL is a vehicle for structuring, organizing managing and retrieving data stored in the database.

@ Curious_. programmer

- It acts as an interpreter which allows the user to interact directly with database through computer language.

- The figure below shows how actually SQL works with databases:



Computer System

The Working of SQL

- In the above system, the computer system has a database which stores all the needed information.

- If the above database is for a company, it might store the information of manufacturing, finance, human resource inventory, payroll etc.

- This process of requesting data from a database and receiving back the results is called a database query - hence the name structured query Language.

- There are various roles which are played by SQL. Some of them are discased below.

1. SQL is an interactive query language.
   SQL provides a very user friendly, easy to use tool which allows the user to write the typical SQL commands in order to retrive the data from the database.

2. SQL is a database programming language.
   Through the use of database utility programs, programmers write SQL commands in their own application to retrive the data stored in database

3. SQL is a database administration language.
   It allows the administrators to define database structures and also control the access to the stored data.

4. SQL is client/server language.
   In the server architecture, the client programs uses SQL to communicate through a network to access the shared data stored in database.

5. SQL is a distributed database language
   Many DDBMS (Distributed) uses SQL to the data across many connected computer systems. The DBMS software running on the local system makes use of SQL to

communicate with other systems by sending request for data access.

7. SQL is a database gateway language.
SQL is most of the time used as a gateway which allows one brad of DBMS to communicate with the other brands.

* Working with Relations of RDBMS
This section highlights how to create, modify or delete relations which may exist in relational model. This can understood by the following SQL statements

1. Creating Relations (Create Tabel statement)

2. Modifying Relations (Alter table statement)

3. Integrity constraints over the relation

1. Creating Relation (create table statement)
- The CREATE TABLE statement defines a new table (Relation) in the database and prepares it to accept data.

- For example, if we want to create a new table IDOLTYIT, the table is created as follows:

Create table TOOLTYIT (S-ID integer not null, S_name varchar (25) not null, contact no integer not null, email varchar (30) not null).

- When the user is creating the above table, the user now became the owner of the newly created table, which is given the name specified in the above statement.

2. Modifying a relation (Alter table statement)

- Alter the table is ready, at times user feels the need to store additional information about the entries in the table.

- The alter table allows the user to change or modify the relation (schema) of the table which is already created by a Create table syntax.

- The alter table statement allow the user to do the following:

1. Add a column defination to the table.

2. Drop a column from the table.

3. Change the default value to the table.

4. Add a drop primary key for the table.

5. Add or drop the foreign key for a table.

6. Add or drop the uniqueness constraint for a table.

7. Add or drop check constraints for a table.

Some of the examples are discussed below

```
1. Alter Table IDOLSYIT
   Add Subject char (15)
```

- On the above example, an existing table of IDOLYSIT is modified with a new column is added as subject which was not there earlier.

```
2. Alter Table IDOLTYIT
   Drop Email
```

- On the above example, an existing table of IDOLTYIT is modified with a existed column is removed known as email which was earlier present in the table.

3. Integrity constraints over the Relation.

- The term data integrity refers to the correctness and completeness of the data in a database. When the contents of a database are modified with the INSERT, DELETE, or UPDATE statements, the integrity of the stored data can be lost in many different ways.

- To preserve the consistency and correctness of its stored data, a relational DBMS typically imposes one or more data integrity constraints.

- These constraints restrict the data values that can be inserted into the database or created by a database update.

- Several different types of data integrity constraints are commonly found in relational database, includes the following

1. Required data checking:

- There are instances when some columns in a database must contain a valid data value in every row; they are not allowed to contain missing or NULL values.

2. Validity checking:

Every column in a database has a domain, a set of data values that are legal for that column. The DBMS can be asked to prevent other data values in these columns.

3. Entity integrity:

The primary key of a table must contain a unique value in each row, which is different from the values in all other rows.

- Duplicate values are illegal, because they wouldn't allow the database to distinguish one entity from another. The DBMS can be forced to enforced this unique values constraint.

4. Referential integrity:

A foreign key in a relational database links each row in the child table containing the foreign key to the row of the parent table containing the matching primary key value.

The DBMS can be asked to enforce this foreign key/primary key constraint.

5. Other data relationships:

- The real-world situation modelled by a database often have additional constraints that govern the legal data values that may appear in the database.

- The DBMS can be asked to check modification to the tables to make sure that their values are constrained in this way.

6. Business rules

- Updates to a database may be constrained by business rules governing the real-world transactions that are represented by the updates.

- For example, there might be a business rule such as the new employee should be added only if the age of the employee is between 18 to 35 years.

Camlin Page
Date / /

* Advantages and Disadvantages of RDBMS

* Advantages

1. Simple data Structures:

- By storing the data in the table format, it becomes easier for the user to understand the structure of database and use it.

- RDBMS provides data access using a natural structure and organization of the data.

- When the user are writing a queries, database queries cam search any columns for any matching entries.

2 Multi-user database access monitoring:

RDBMS allows the multiple database users to access a database simultaneously.

- By taking advantage of services of transaction management and locking, it allows the user to access the data without being changed, prevents collisions between two users updating the same data, and keeps users from accessing partially updated records.

3. Well defined privileges

Authorization and privilege control features in a RDBMS allow the database administrator to restrict access to authorized users and grant privileges to individual users based on the types of database tasks they need to perform.

4. Network Access:

RDBMSs provide access to the database through a server daemon, a specialized software program that listens for request on a network, and allows database clients to connect to and use the database.

5. Speed:

The relational database model is not the fastest data structure. RDBMS advantages, such as simplicity, make the slower speed a fair trade-off

INTRODUCTION TO DATABASE STRUCTURE

INTRODUCTION :

- In traditional system, each collection of application programs had it own independent master file. The duplication of data over master files could lead to inconsistent data.

- In early days, efforts were discovered to use a common master file for a number of application programs resulted in problems of integrity and security.

- As a first step towards a DBMS, packages of subrutines were introduced to reduce programmer effort in maintaining these data structures.

- However, the use of these packages still requires knowledge of the physical organization of the data.

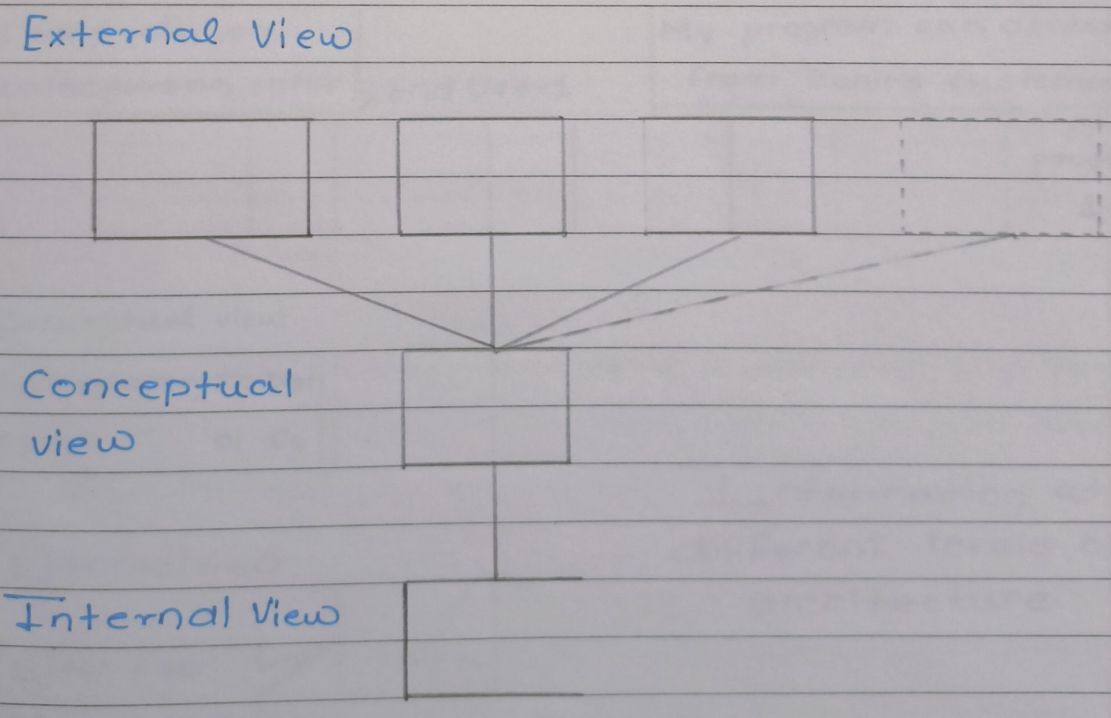
- The database can hold a variety of different things. The database concepts are divided into two concepts.
    1. Schema
    2. Data

- The schema is the structure of the database and the data is the facts of the database.

Levels Of Abstraction In DBMS:

- Database management can be defined in the way in which they use their data dictionary.

- Data dictionaries are helpful for all human users, especially the database administrator, as well as invaluable to the application programs and report generators that might access the database.



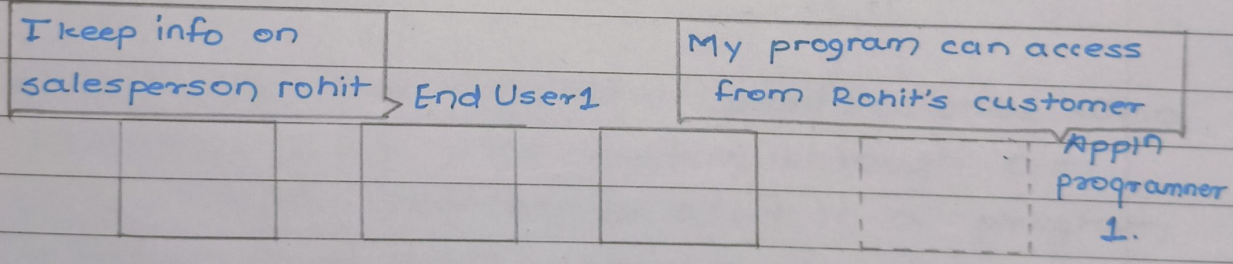
The Three Level Database Architecture

- The three levels of database architecture are.

1. External Level: It is concerned with the way individual user observe the data.

2. Conceptual Level: It can be regarded as a community user view a formal description of data of intrest to the organisation, independent of any storage considerations.

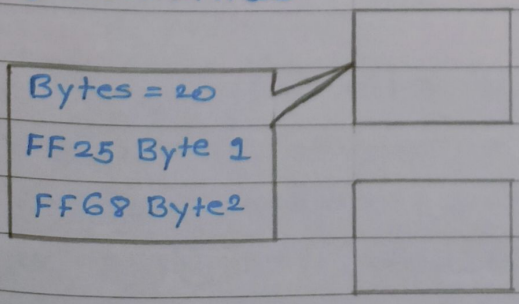
3. Internal Level: It is a concerned with the way in which the data is actually used.



Conceptual view

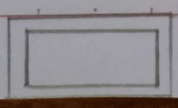
Salesperson	ID	Dept
Rohit	01	64

Internal view



Information at different levels of architecture

@Curious_.programmer



Camlin Page
Date / /

Role of Database Users

1. Naive user:

- They are unsophisticated users who interact with the system by invoking one of the permanent application programs that have been written previously.

Example:

Suppose the bank teller wants to transfer the money after maturity of the fixed deposit amount of a particular customer, needs to invoke a program called transfer.

2 Application programmers:

- They are the computer professional who interact with the system through DML calls, which are embedded in a program written in a host programming language.

- A special pre-processor, called the DML precompiler, converts the DML statement to normal procedure calls in the host language.

@ Curious_programmer

- Since the DML syntax is different from the host language syntax, DML calls are usually prefaced by a special character so that the appropriate code can be generated

3. Sophisticated users:

- These users interact with the database using database query language.
- They submit their query to query processor
- Then Data Manipulation Language (DML) functions are performed on the database to retrieve the data.
- Tools used by these users are OLAP (Online Analytical Processing) and data mining tools.

@ Curious_.programmer

4. Specialized users

- These users write specialized database application to retrive data.
- These application can be used to retrive data with complex data types e.g. graphics data and audio data.

* Role of Database Administrator.

A person having who has central control over data and program that access the data is called DBA. Following are the functions of the DBA:

- Schema defination: DBA creates database schema by executing Data Definition Language (DDL) statements.

- Storage structure and access method defination.

- Granting of authorization for data access: DBA can decide which parts of data can be accessed by which users. Before any user access the data, DBMS checks, which rights are granted to the user by the DBA.

Transaction Management

What is Transaction?

- A transaction is an event which occurs on the database. Generally a transaction reads a value from the database or writes a value to the database.

- Although a transaction is an event which occurs on the database. Generally a transaction reads a value from the database or writes a value to the database.

- A read operations does not change the image of these database in any way.

The four Properties of Transactions:

- Every transaction, for whatever purpose it is being used, has the following four properties. Taking the initial letters of these four properties we collectively call them the ACID properties.

1. Atomicity:

This means that either all of the instructions within the transaction will be reflected in the database, or none of them will be reflected.

Example:

2. Consistency:

@Curious-.programmer

If we execute a particular transaction in isolation or together with other transaction, (i.e. presumably in a multiprogramming environment), the transaction will yeild the same expected results.

3. Isolation:

In case multiple transaction are executing concurrently and trying to access a sharable resource at the same time, the system should create an ordering in their execution so that they should not create any anomaly in the value stored at the sharable resource.

4. Durability: It states that once a transaction has been complete the changes it has made should be permanent.

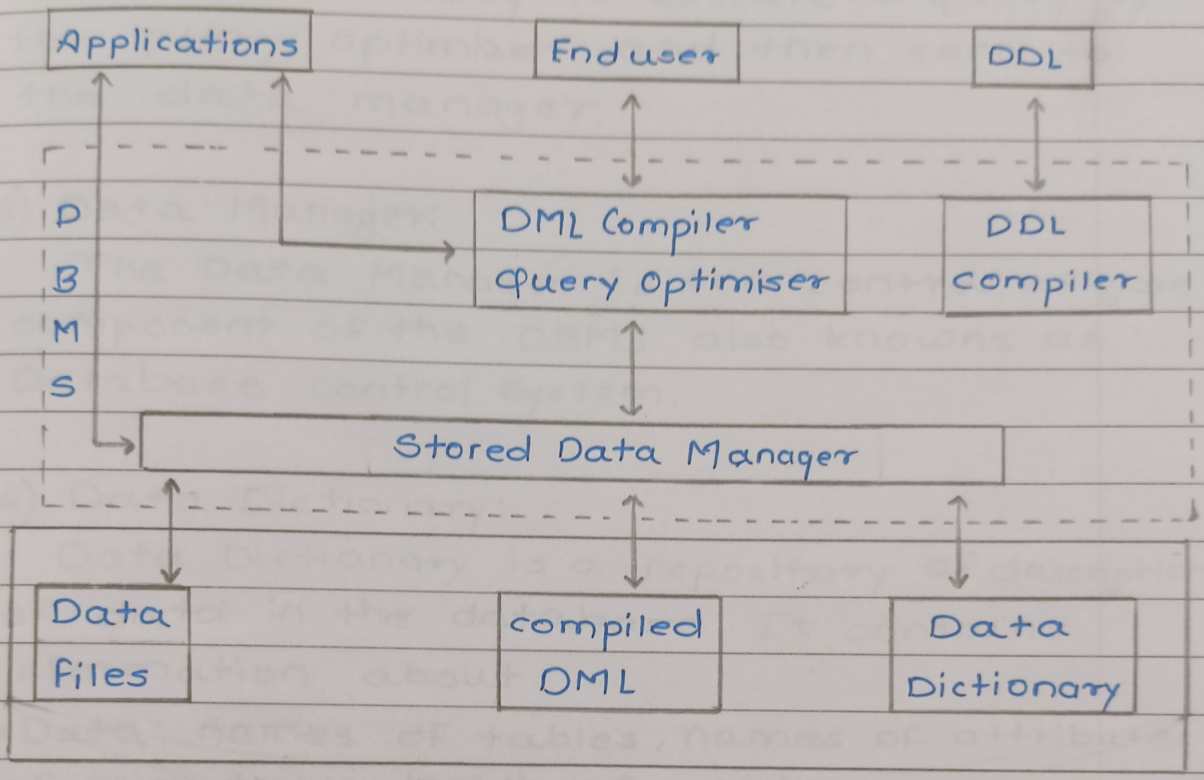
Database Structure:

In a database structure, the DBMS acts as an interface between the user and the database

The user requests the DBMS to perform various operations such as insert, delete, update and retrieval on the database.

The components of DBMS perform these requested operations on the database and provide necessary data to the users.

The various components of DBMS are shown below:-



Structure of DBMS

1) DDL Compiler

Data Description Language compiler processes schema definitions specified in the DDL. It includes metadata information such as the name of the files, data items, storage details of each file, mapping information and constraints etc.

2) DML Compiler and query optimiser.

The DML commands such as insert, update, delete, retrive from the application into

program are sent to the DML compiler for compilation into object code for database access. The object code is then optimized in the best way to execute a query by the query optimizer and then send to the data manager.

3) Data Manager:

The Data Manager is the central software component of the DBMS also knowns as Database control system.

4) Data Dictionary:

Data Dictionary is a repository of description of data in the database. It contains information about

- Data: names of tables, names of attributes of each table, length of attributes, and number of rows in each table.

- Relationships: between database transactions and data items referenced by them which are useful in determining which transaction are affected when certain data definitions are changed

- constraints on data i.e. range of values permitted.

5) Data Files: It contains the data portion of the database.

6) Compiled DML: The DML compiler converts the high level queries into low level file access commands know as compiled DML.

7) End Users: They are the users of the system who is going to use the system for their day to day activities.

@ Curious_.programmer

Introduction To Data Models

Introduction:

- A data model is a picture of description which shows how the data is to be arranged to achieve a given task.

- It is a clear model which specifies how the data items are arranged in a given model.

- DBMS organize and structure data so that it can be retrived and manipulated by different users and application programs.

- A data model determined both the personality of a DBMS and the applications for which it is particularly well suited.

- The data structures and access techniques provide by a particular DBMS are called its data model.

@curious-.programmer

Camlin | Page
Date / /

Evolution of Data Models:

- The first non-proprietary programming language was COBOL and with COBOL, & later FORTRAN, programming became the foundation of creating enterprise computer systems.

- The system developed, needed to store its data somewhere and the programmers designed more or less proprietary and specialized solutions for this purpose.

- In 1964 the first commercial database management system was born; IDS - Integrated Data store, developed at General Electric, based upon an early network data model developed by C.W. Bachman (Bachman 1965)

- In 1970 Edgar F. Codd published an article which offered a fundamentally different approach (Codd 1970)

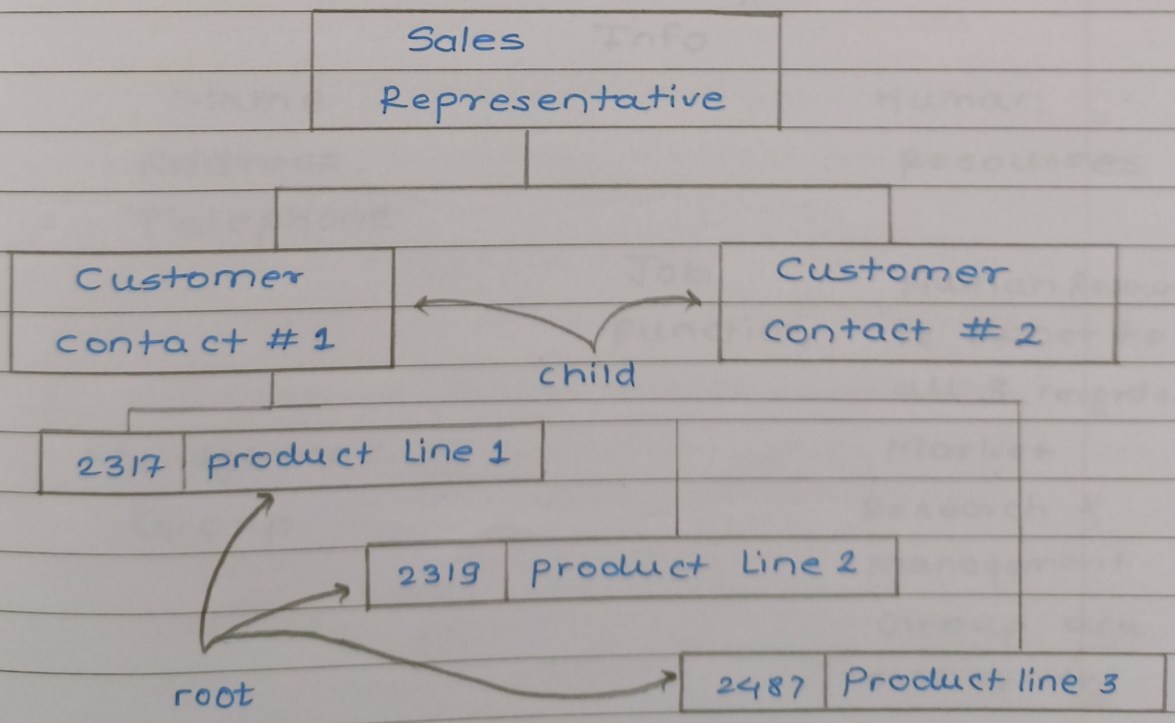
@Curious_.programmer

Types of Data Model:

There are four different types of data models:

1. Hierarchical database
2. Network database
3. Relational database
4. Object oriented database

Hierarchical Database:



- Hierarchical Database is most commonly used with mainframe system.

- It is one of oldest methods of organizing and storing data and it is still by some organizations for making travel reservations.

- A hierarchical database is organised in pyramid fashion, like the branches of a tree extending downwords.

Network Database:

Market Research

Social security Info

Name
Address,
Telephone

Job function

Human Resources

Management Group

Human Resources is owner to all 3 records.
Market Research & management Group are owner to only certain members.

- Network database are similar to hierarchical database by also having a hierarchical structure. There are a few key differences, however.

- Instead of looking like an upside-down tree, a network database looks more like a cobweb or interconnected network of records. In network database, children are called members and parents are called owners.

Relational database

- Pre-relational models depended upon being able to determine explicity where and how individual records were stored.

- Early relational proponents argued that the relational data model viewed information logically rather than physically, but this is not quite correct.

- The relational data model looks at information as an unordered collection of "relations".

- Each relation is popwated with unordered "tuples" of the same unordered "field" structure.

Object Oriented Database:

- A data model is a logic organization of the real world objects (entities), constraints on them, and the relationships among objects. A DB language is a concrete syntax for a data model.
A DB system implements a data model.

- A core object-oriented data model consists of the following basic object oriented concepts.

1) Object and object identifier

2) attributes and methods

3) class

4) class hierarchy and inheritance.

Camlin Page
Date / /

Advantages And Disadvantages of Data Models:

1. Hierarchial Database:-

Advantages:

1. Simplicity:

Since the database is based on the hierarchical structure, the relationship between the various layers is logically simple

2. Data Security:

Hierarchical model was the first database model that offered the data security that is provided by the DBMS.

Disadvantages:

1. Implementation complexity:

Although it is simple and easy to design, it is quite complex to implement.

2. Database Management Problem:

If you make any changes in the database structure, then you need to make changes in the entire application program that access the database.

Camlin Page
Date / /

Network Model:

Advantages

1. Conceptual simplicity:
   just like hierarchical model it also simple and easy to implement.

2. capability to handel more relationship types:
   the Network model can handel one to one 1:1 and many to many N:N relnship

3. Ease to access data: the data access is easier than the hierarchial model.

Disadvantages:

1. System Complexity:
   All the records have to maintain using pointers thus the database structure becomes more complex.

2. Operation Anomalies:
   As discussed earlier in network model large number of pointers is required so insertion, deletion and updating more complex.

Camlin Page
Date / /

Relational Model:

Advantages:

1. Conceptual simplicity:
We have seen that both the hierarchial and network models are conceptually simple, but relational model is simpler than both of these too.

2. Structural Independence:
In the Relational model, changes in the structure do not affect the data access.

Disadvantages:

1. Hardware overheades:
The relational database systems hide the implementation complexities and the physical data storage details from the user. For doing this, the relational database system need more powerful hardware computers and data storage devices.

2. Ease to design can lead to bad design.
The relational database is easy to design and use. The user needs not to know the complexities of the data storage.

@ Curious-programmer

Business Rules:

- Business Rules helps employees focous on and implement the actions within the organizations environment.

- Some things to think about when creating business rules to keep them simple, easy to understand, keep them broad so that everyone can have a similar interpretation To be considered true business rules must be in writing and kept up to date.

Some examples of business rules:

Departments --- offers ---course

course ---------generates ---class

Professors------teaches ----class

- Another important aspect of business rules consist of how the rules are shared within the company.

Database Design:

The database design process consists of a number of steps listed below:

Step 1: Requirements Collections & Analysis:

- Prospective users are interviewed to understand and document data requirements

- This step results in a concise set of user requirements, which should be detailed and complete.

- Functional requirements can be documented using diagrams such as sequence diagrams, data flow diagrams, scenarios, etc.

Step 2: Conceptual Design:

- Once the requirements are collected and analyzed the designers go about creating the conceptual schema.

- Conceptual schema: consists description of data requirements of the users, and include a detailed description of the entity types, relationships and constraints.

Camlin Page
Date / /

ER Model:

In 1976, Entity relationship model developed by Chen,

ER Model is high level conceptual model which used Conceptual design of database where as relational model are used to logical design of database.

- ER diagram:

- A database can be modeled as
  - A collection of entities
  - Relationships among the entities
-
- An entity is an real world object that exist and it is distinguishable from other entities.

  Example Person, company, event, plant

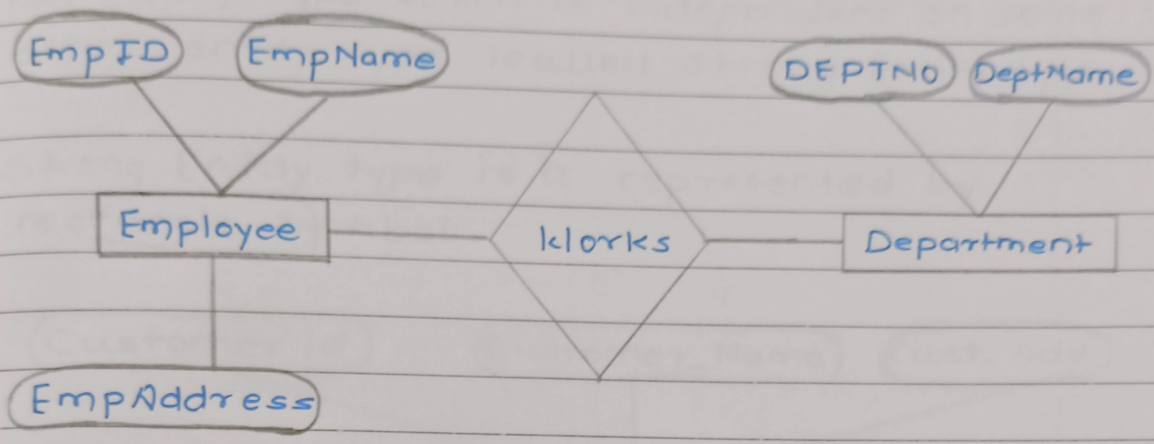
- All the entities in the data model have attributes as known as properties of an entities

  Example: people have names and addresses

An Entity set is a set of an entities of all same type that share the same properties

Example: set of all persons, companies, tress, holidays

ER Diagram:

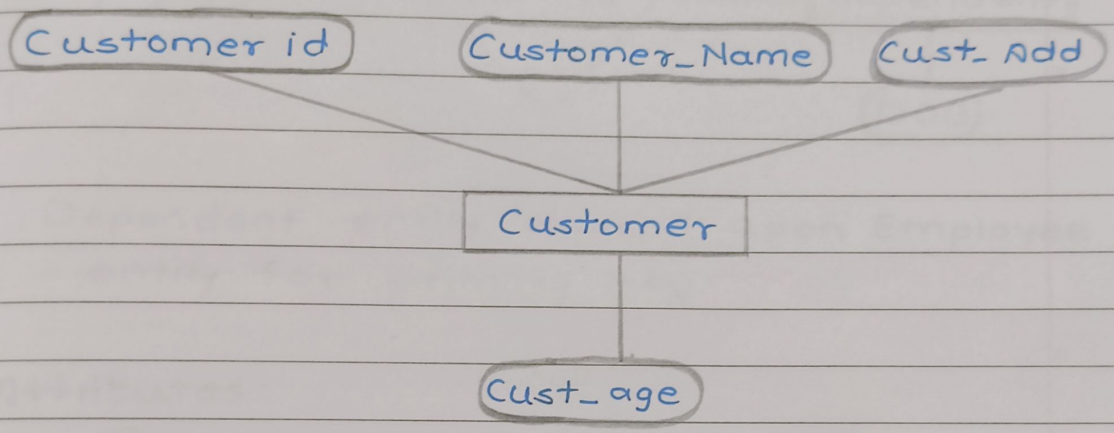


- Rectangles represents entity sets.
- Diamonds represents relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets
- Underline indicates primary key attributes.
- Ellipses represents an attributes
- Double lines represents total participation of an entity in a relationship set
- Double rectangle represent a weak entity sets

Strong Entity type:

An Entity type which is independent on some other entity type is called strong Entity type.

Strong Entity type is a represented by rectangle symbol



Customer Entity

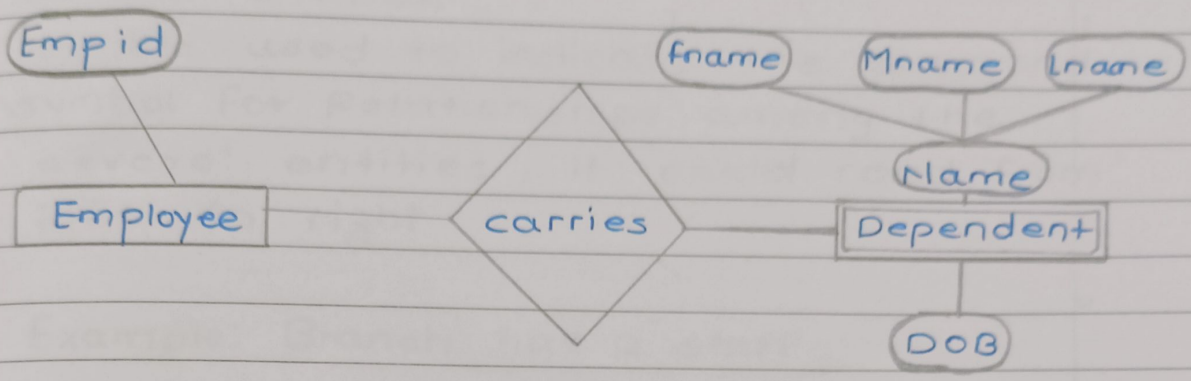
kleak entity Type:

Entity type which is dependent on a strong other entity type is called as kleak entity type.

- kleak entity type is dependent on ~~some other~~ strong entity type ~~is~~ and cannot exist on its own
- It does not have a unique identifier that has partial identifier.

Camlin Page
Date / /

Weak entity type is represented by double rectangle.



Dependent entity depend upon Employee entity for primary key.

Attributes:

Properties of an entity or relationship type is called as attribute.

Attribute domains:

The set of allowable values which is assigned to one or more attribute is known as Attribute domains.

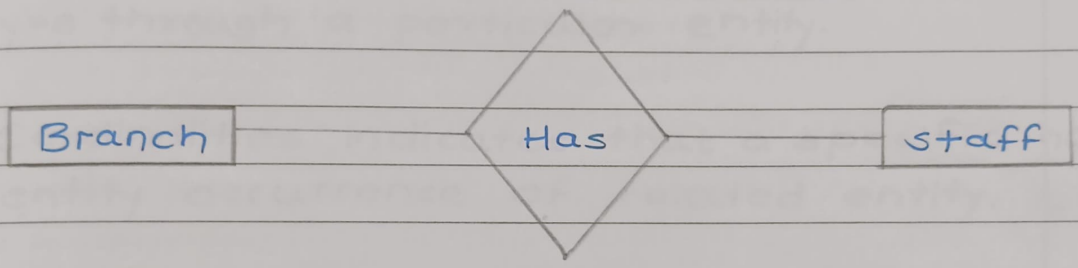
There are types of attributes has been classified such as simple and composite type, single valued and multi valued attributes stored and derived attributes and key attributes

Relationships:

A set of meaningful relationship among several entities,

We used to indicate the diamond symbol for Relationships among the several entities, it could read from left to right

Example: Branch has a staff

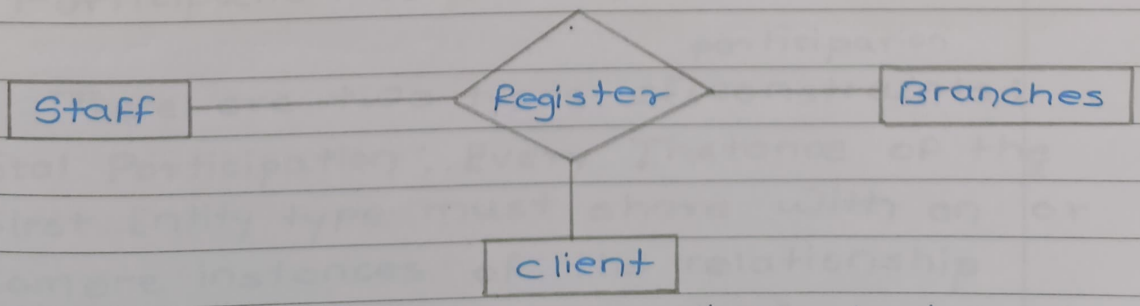


Degree of relationship:

It is the number of entities participated in a particular relational model.

There are two type of degree of relationships.

- Binary relationship
- Ternary relationship



Staff registers a client at a branch

Relationship set:

The collection of similar relationship is known as Relationship set.

Constraints on relationship

1) Mapping constraints / cardinalities

The number (or range) of possible entity type that is associated to another entity type through a particular entity.

Cardinalities indicates that a specific no. entity occurrence of related entity.

Type of Mapping Constraints

One - to - one (1:1)

One - to - many (1:*)

Many - to - one (*:1)

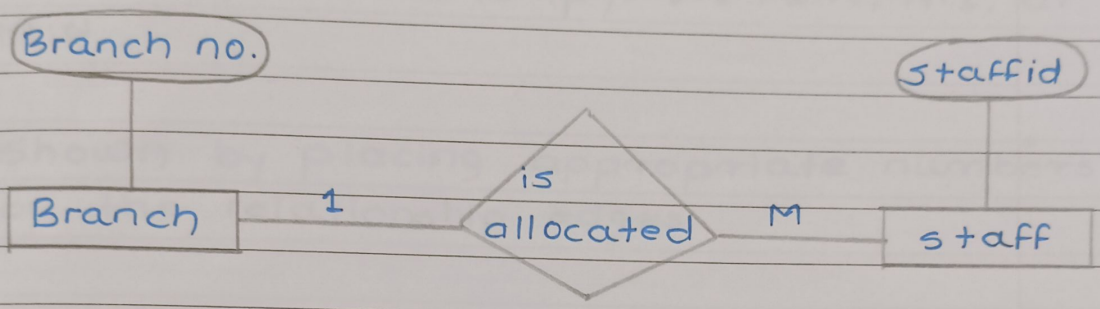
Many - to - many (*:*)

2. Participation Constraints

There are two types of participation constraints:
Total Participation: Every Instance of the first Entity type must share with on or some instances of the relationship

type with the other entity type.

The total participation is represented by a dark line or double line between the relationship and entity.

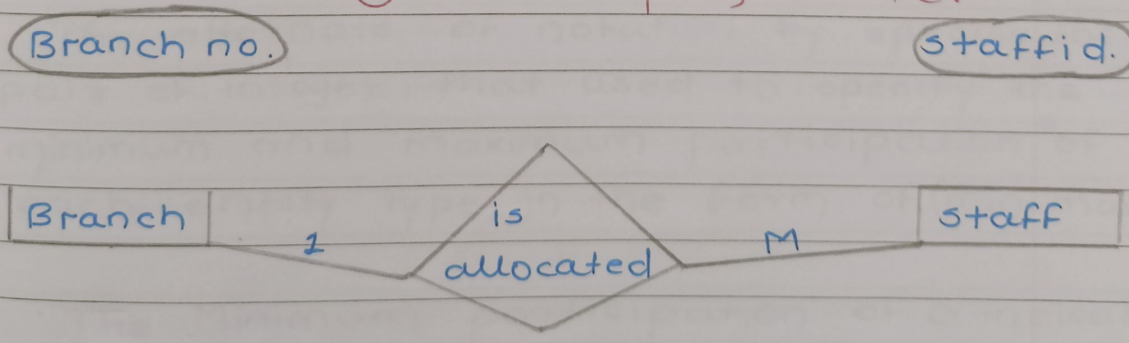


Every Branch office is allocated members of staff.

- Partial Participation:

There exist an instance of the first entity type that don't share an instance of the relationships type with the other entity type.

@ Curious-.programmer



Every A member of staff need not work at a Branch office.

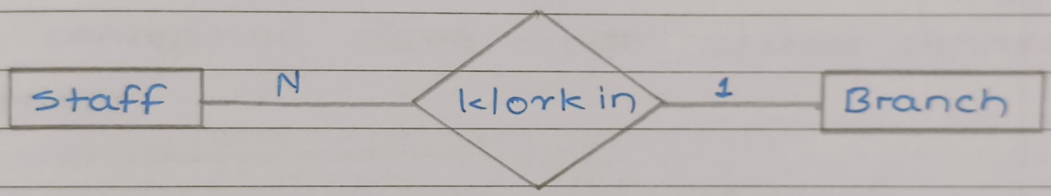
- Notations used in ER Diagrams for Representing Relations:

1) Cardinality Ratio Notation:

In this method, Cardinality ratio (of binary relationship): 1:1, 1:N, N:1, or M:N

shown by placing appropriate numbers on the relationship edges

Eg.

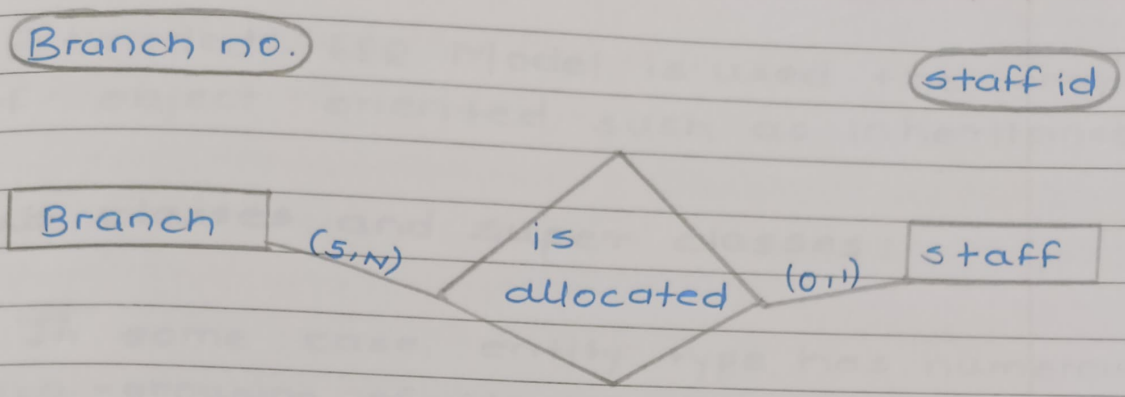


Number of staffs working in Branch

2) Min-Max notation:

The alternate of notation by specify the pair of integer, that used to specify the minimum and maximum participation of each entity type in the form of (min, max)

The Minimum participation of 0 indicate partial participation where as maximum participation of 1 or more indicates total participation.



At least 5 staff is allocated to branch.

- Limitation of Entity Relationship Model :

Problems may arise when designing a conceptual data model called connection traps.

- Often due to a misinterpretation of the meaning of certain relationships.

- Extended Entity Relationship Model:

Since 1980 there has been increase in the emergence of new database application with more demanding application.

@ Curious-.programmer

Basic concepts of ER modelling are not sufficient to represent the requirement of newer, more complex operation.

Extended EER Model is used the concept of object oriented such as inheritance.

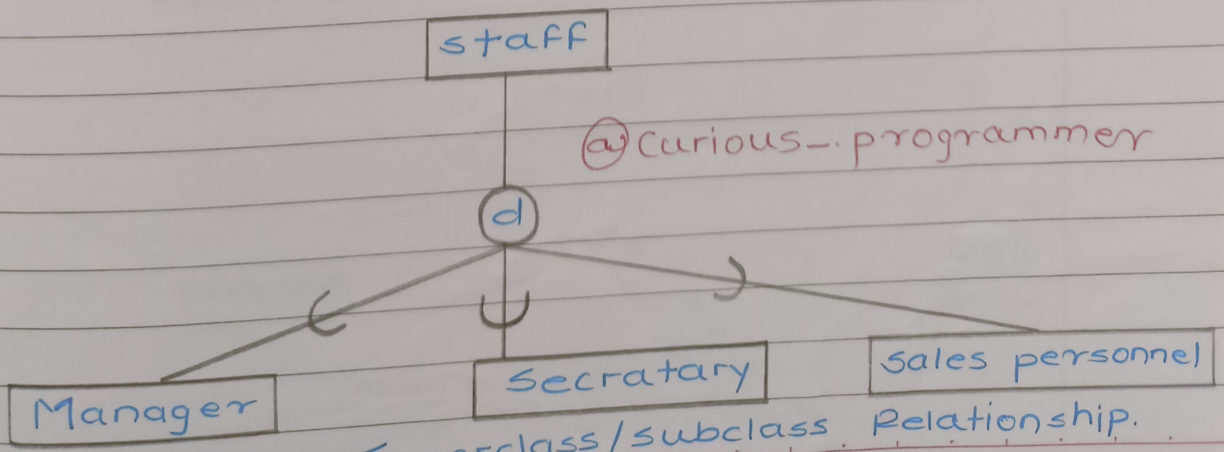
- Sub classes and super classes:

In some case, entity type has numerous sub-grouping of its entities because that are meaningful way for representation and need to be explicitly defined because of their importance.

An entity type that includes distinct subclass that require to be represented in a data model is called a super class.

A subclass is a entity type that has a distinct role and is also a member of the superclass

staff is the super class where as manager, secretary, sales personnel is the subclass.



Superclass/subclass Relationship.

Camlin Page
Date / /

Type Inheritance:

- The type of an entity is defined by the attributes it possesses, and the rel^nship types it participates in.

- Because an entity in a subclass represents the same entity from the super class, it should posses all the values for its attributes, as well as the attributes as a member of the super class.

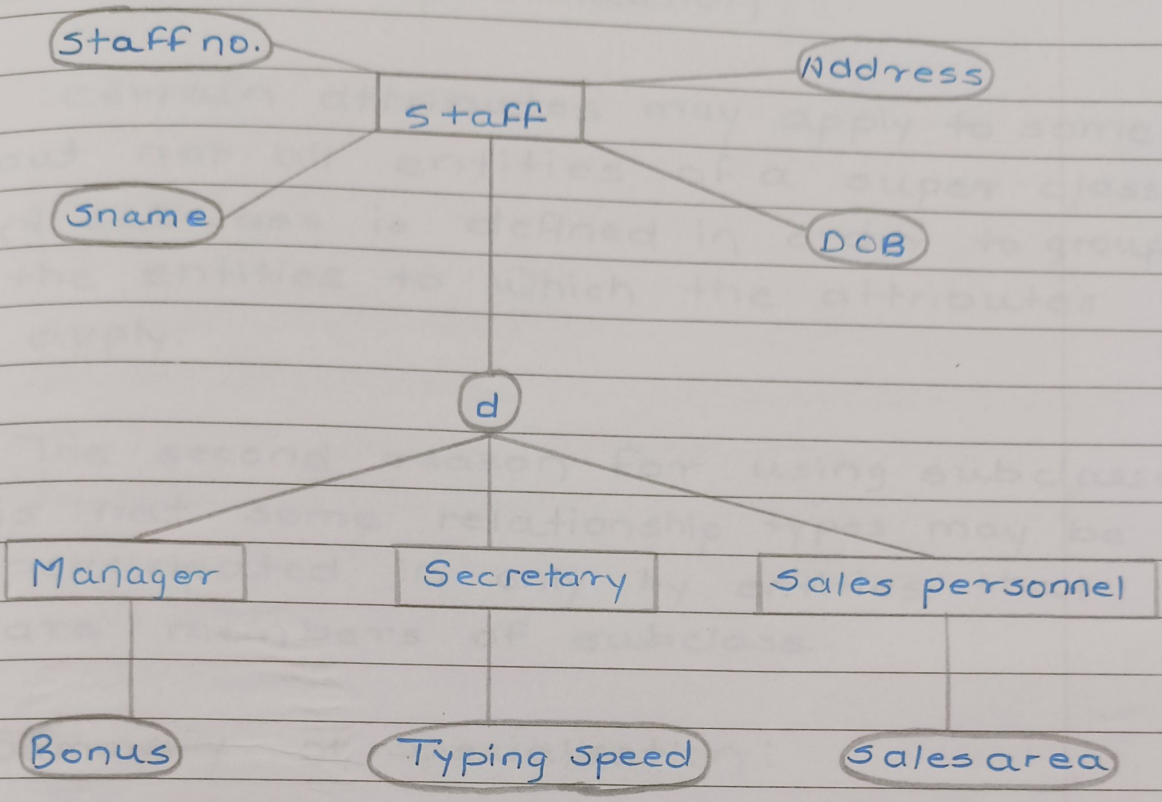
Specialization:

The process of defining a set of sub class of super class.

@curious_.programmer

The specialization is a top down approach of super class and subclasses

The set of sub classes is based on some distinguishing characteristic.



- Notation for Specialization:

- To represent a specialization, the sub classes that define a specialization are attached by lines to a circle that represents the specialization, and is connected to the super class.

- Attribute that only apply to subclass are attached to the rectangle representing the subclass. They are called specific attributes.

@ Curious-. programmer.

(X) sub class can also participate in specific relationship types

Camlin Page
Date / /

- Reasons for specialization:

  - certain attributes may apply to some but not all entities of a super class. A subclass is defined in order to group the entities to which the attributes apply.

  - The second reason for using subclasses is that some relationship types may be participated in only by entities that are members of subclass.

- Summary of specialization:

  Allows for:

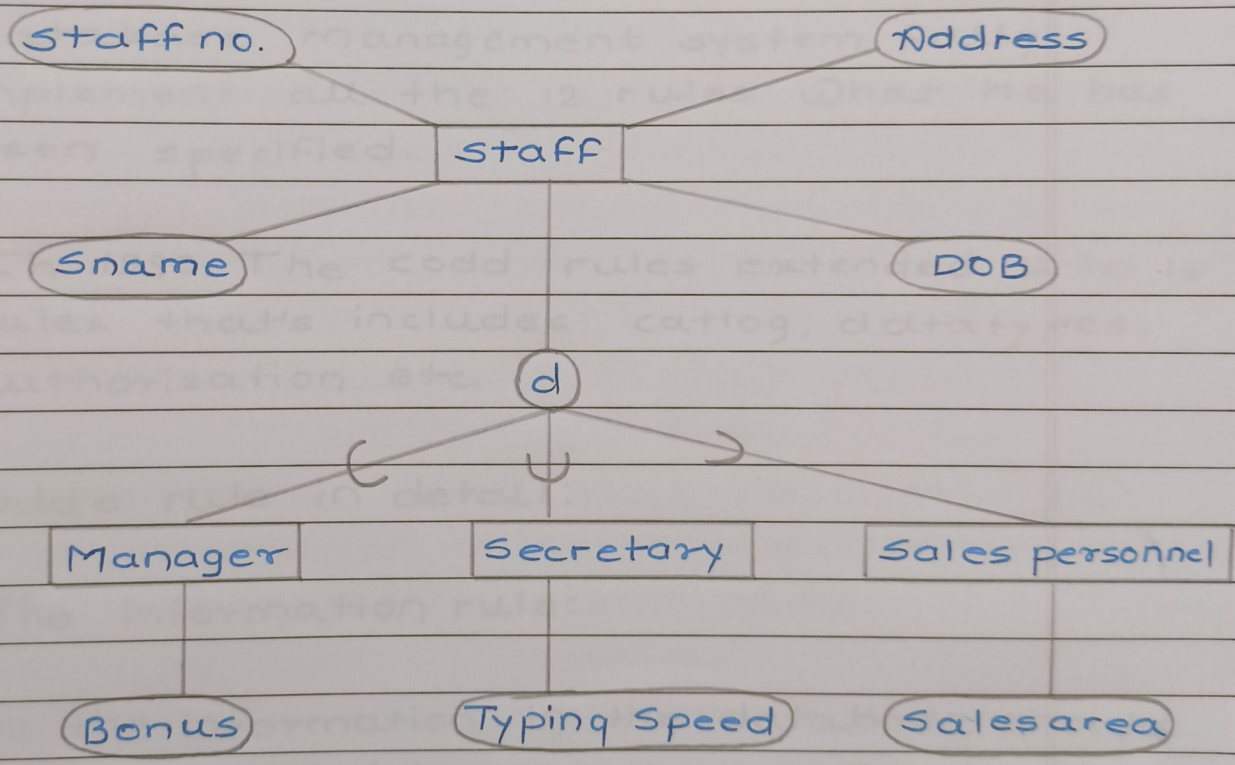
  - Defining set of subclass of entity type

  - Create additional specific attributes for each sub class.

  - Create additional specific relationship types betⁿ types each sub class and other entity types or other subclasses.

- Generalization:-

- Generalization is the reverse of specialization and this is a bottom-up approach
In Generalization, there are several classes with common features and generalizing into a super class.



Attribute Inheritance:

- An entity in a subclass may possess subclass specific attributes, as well as those associated with the superclass.

- CODD'S Rule:

Dr. E.F Codd was inventor of the relational database model. This model say that whether the Database management system follow the relational model or not and what extends model is relational.

The article mentioned by Dr. E.F. Codd that according to these rule, there is no database management system fully implement all the 12 rules what he has been specified.

In 1990, The codd rules extended 12 to 18 rules that's includes catlog, datatypes, authorization etc.

Codd's rule in detail:

1) The information rule:

i) All the information in the database should be represented in the term of relational or table. Information should be stored as an values in a tables.

ii) Data should be stored in form a table and no other means to stored the data.

- Guaranteed access rule:

i) The guaranted access All data must be accessible. The Rule say that there is fundamental requirement of primary key for each record in table, and there should be no ambiguity by stating the table name and its primary key of the each record in the table along with columns name to be accessed rule.

ii) For accessing the data from the table, we must provides Table name, Primary key and other column names in the record table to be accessed.

- Systematic Treatment of Null values

i) A null values means that we don't know what information must be provided or entered in to this field name.

iv) Null values must be handled logically and consistent manner.

- Active online catlog based on the relational model.

i) The system must support an online catlog based data dictionary which hold the information or description about the

table in the database.

ii) User Tables: The user table contains the data about the table which is created by any users in the database systems.

iii) System tables: The system table contains the data about the structure of the database and database object.

iv) The collection of the system tables is known as the system catlogs or data dictionary.

5) The comprehensive data sublanguage rule:

i) The system must support at least one relational language that through which the data in the database must be accessed

ii) The language must support all the operation of the following items:

Data defination
View defination
Data Manupulation
Integrity constraints
Authorization
Transaction boundaries
(begin, commit & rollback)

6) The view updating rule:

i) All the view must be theoretically updatable can be updated by the system.

ii) There is ambiguity in this rule, the structured query language support a single updation at a time suppose if we try combine two or more tables a for a complex views and try to update the views and the DBMS would fail to update the records to the respective tables, thereby violating this rule.

Eg. If Roll_no column is not present in the view then it is not possible update the view of the student table.

7) High-level, update, and delete:

i) This rules states that in the relational model, the structured query language must performed data manipulation such as inserting, updating and deleting record on sets of row in the table.

ii) Consider an example, if you want to delete the record of the invoices table which are older than six years, you don't have locate position each record and delete them individually, you should able to delete set of records in the table using one single command.

iii) The same concept can be apply to inserting and updating the record.

8) Physical data independence:

i) Any change made in the data is physically stored in the file system through array and link list must not effect application that access the data structure.

This rule say that any change is made in the back end must not effect front end application

If the database file renamed or database location is change, then this should not have effect on the application.

9) Logical data independence:

This rule state that change in the logical level must not change to the application's structure.

This rule state that it should possible to change the database design or alter the database ~~but the~~ design ~~the~~ without the user begin aware of it.

10) Integrity independence:

Data integrity constraints should be considered as seperated from application program, the structured query Language which defines data integrity constraints must be stored in the database in term of data in table that is, in the catlog and not in the application.

Referential integrity and entity integrity is integral part of the relational database, in more specific term, the following two integrity should be apply to the relational database.

Relational Database Model

E.F. Codd first proposed the relational database Model also he is known as the father of Relational model.

Relational model was attempt of specify the database structure in term of matrix. ie the database should contain tables. The tables is in form of set of Columns and Rows. The relational model is set of 2 dimensional table consists of rows and columns.

In the relational database model consist of set of tables having the unique name.

- Logical View of Data:

1) Introduction:

Logical structure of tables is consist of 2- diemensional tables consist of no.s of horizontal rows and vertical columns

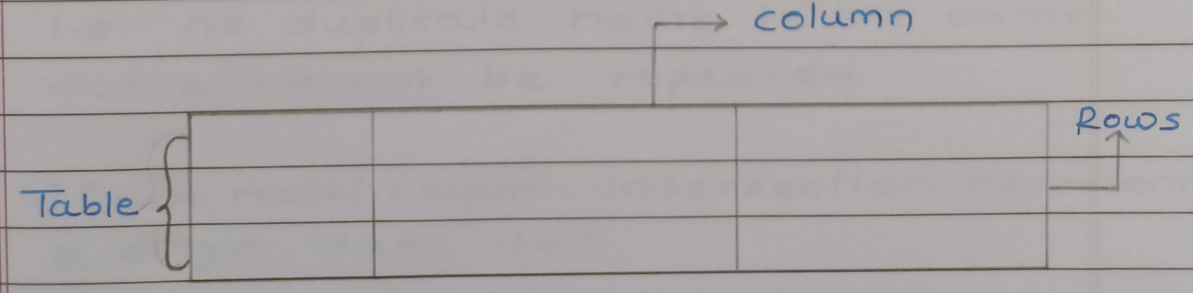


Table is an abstract entity which does not say how the data is stored in the physical memory of the computer system

Each table in the database has its own unique name trough which we can refer the content of the table by the unique name.

2) Characteristics of an table:

i) A tables in the database must be in the two-diemensional structure which consist number of rows and columns.

ii) Each row in the table is called as record or tuple can represent as a single entity which is occur within the entity set i.e. Customer record in the Customer table.

iii) Each column name in the table is called as attribute and each row in the table is called as record. Each column name in the table is unique name i.e. no duplicate name in the same table cannot be repeated.

iv) Each rows/ column intersection represent a single data item.

v) All the value in the column must be represent in the same data format

vi) Each columns has the specific range of values, and also refer as the domain attribute.

vii) The order of rows and columns is not limited to the DBMS.

3) Example:

There is customer Table contain all information about the

customer customer_id

Customer_Name cust_Age

cust_Address cust_Mobile No.

Cust_ id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_ Mobile
1.	Yogesh	20	Klorli	9892946123
2.	Ramesh	30	Bandra	9324689573
3.	Ram	18	Mahim	8283435561
4.	Pramod	24	khar	7292917761
5.	Yatin	25	Dadar	9897867427
6	Tushar	26	Matunga	9873476231

Attribute:

- Each column in the above table represent the data item in the database

- Each column in the table represent the attribute in the table.

- Atleast one column consist in the table.

- The ANSI/SQL standard does not specify a maximum numbers of rows and columns in the table.

Eg. Cust_id , Cust_Name , cust_Age, Cust_Address , Cust_Mobile_No.

- Attribute

- Each column in the above table represent the data item in the database.

- Each column in the table represent the attribute in the table

- At least one column consist in the table.

Records / Tuples:

- A single Record consist all the information of the single entity.
- Each horizontal row in the Customer table represented a single entity.
- A Table consists any numbers of rows, The ANSI/SQL standard does not specify the limits of rows in the table.
- Empty table is called when there is zero row consist in the table.

- **Key:**

Definition:

A column value in the table that uniquely identifies a single record in the table is called key of an table.

A attribute or the set of attribute in the table that uniquely identifies each record in the entity set is called a key for that entity set.

- Types of Keys:

Simple key:

∅ key which has the single attribute is known as the simple key.

Composite key:-

∅ key which consist two or more attributes is called a composite key.

Example:

Cust_id is a key attribute of customer Table it is possible to have a single key for one customer i.e. cust_id ie cust_id = 1 is only for the Cust_Name = "Yogesh"

Types of key	Definition of key
Super key	Super key is sufficient to identify the unique record in the table.
Candidate Key	∅ mimimul super key is called Candidate key. ∅ super key has no proper sub set of Candidate key.
Primary key	∅ candidate key is choosen as a principal to identify

	a unique.
Foreign key	an column (or combination of columns) in the one tables whose values is match the primary key in the another table.

Referential Integrity:

Referential integrity say the column which contain foreign key in one table must be primary key of another table.

In general term, Foreign key of Table A must be Primary key Table B

Example

Customer Table:

Cust_ id	Cust_ Name	Cust_ Age	Cust_ Address	Cust_Mobile No.

Account Table

Account No	cust_ id	Account type	Balance	Description

In The above example cust_id is the primary key for the customer Table While cust_id is the foreign key for the Account table

- Relational Integrity Rules:

1) Entity Integrity:

Entity Integrity ensure that there is no duplicate records in the table and each field that recognizes each record in the table must have and not having null values

Entity Integrity specifies that every instance of entity have the unique values ie primary key must be kept and must have the values other than null values.

Entity Integrity must have two properties for primary keys:

- The primary key must be unique for each row in the table that is no two primary key having the same value in the same table, The primary key values must be distinct i.e. the value could not be repeated.

- The primary key values should not contain null values, primary key must be NOT NULL

2) Referential Integrity:

Referential Integrity is a property of data which, when satisfied, requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different relation.

For referential integrity to hold in a relational database, any field in a table that is declared a foreign key can contain only values from a parent table's primary key or a candidate key. For Instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

3) Other integrity rules:

As the integrity rules that states column which specify the NOT NULL values means these column must contain some values which should not contain any NULL values.

Unique. In this rules no two record or tuples have same values for the same attribute.

Check. In this rule we can apply own integrity rules by applying CHECK constraint.

- Relational Database Design Process

The Relational Database model was proposed by E.F. Codd in 1969. The Relational Database Model is based on branch of mathematics called set theory and predicate logic. The idea behind to design the Relational Database model is that the database consist of series of unordered table of relation that can be manipulated using non-procedural process that return tables.

- The benefits of a relational Database Design process.

- Data entry, updating and deleting would be efficient and simple in manner.

- Data retrieval, summarization and reporting will be efficient.

- Database must follows a well designed model hence it behave predictably

- Change the database structured are easy to make er creating database, tables, views.

- Feature of Good Relational Database Design - Normalization.

i) In the Relational Database Design, the process of organizing data to minimizing redundancy is known as Normalization

ii) The main aim of the Normalization is to decompose complex relation into smaller, well-structured relation

iii) Normalization is the process that involves dividing a large table into smaller table and stating the relationship among the tables.

iv) Data normalization or Database Normalization is also canonical synthesis is mean for preventing the inconsistent in a set of data by using unique values to reference common information.

v) If we apply the normalization on the table we can reduce the problem of data inconsistency for some extent.

Definition of Normalization:

In the Relational Database Design, the process of organizing data to minimizing redundancy is known as Normalization.

- Main aim of the Normalization:

1) Ensure data integrity.

i) The correct data should be stored in the database.

ii) This can be achieved by applying integrity rules in the database.

iii) Integrity rules prevent duplicate values in the database

2) Prevent Data Redundancy in database:

i) Non-Normalised data is more vulnerable to data anomalies. The same set of information is present in the multiple rows, now if we applying the updating rule on the table then it lead to logical inconsistence this is known as update anomaly.

An insufficiently normalized table might have one or more of the following characteristics:

i) Update anomaly

ii) An insertion anomaly

iii) An deletion anomaly

- Advantages of Normalization:

1) Avoids data modification anomalies as each data item lives in one place

2) Greater flexibility in getting the expected data in atomic granular

3) Normalization is conceptually cleaner & easier to maintain and change as your needs change.

4) Fewer null values and less opportunity for inconsistency

5) A better handle on database security.

6) Increased storage efficiency

Disadvantages of Normalization:

1) Requires much more CPU, memory, and I/O to process thus normalised data gives reduced database performance.

2) Requires more joins to get the desired result. A poorly-written query can bring the database down.

3) Maintenance overhead. The higher the level of normalization, the greater the number of tables in the database

- Normal Form:

Normal form are designed for addressing potential problem in the database such that inconsistent and redundant data which is stored in the database.

Normal form is based on relation rather than table. The normal form has a set of attribute which table should be satify. The following attribute are:

1) They describe one entity

2) They do not have duplicate rows, hence there must a primary key for each row.

3) The columns are unordered

4) The rows are unordered

- Types of Normal Forms:

1) Edgar F. Codd, the inventor of the relational model, introduced the concept of normalization and what we now know as the First Normal Form (1NF) in 1970

2) Second Normal Form (2NF) and Third Normal Form (3NF) in 1971,

3) Codd and Raymond F. Boyce defined the Boyce-Codd Normal Form (BCNF) in 1974.

4) Fourth normal form (4NF)

5) Fifth Normal form (5NF)

6) Higher normal forms where defined by other theorists in subsequent years, the most recent beg being the Sixth Normal Form (6NF) introduced by Chris Date, Hugh Darwen, and Nikos Lorentzos in 2002

- Boyce-Codd Normal Form (BCNF)

- Based on functional dependencies that takes into account all candidate keys in a relation.

- For a relation with only one candidate key, 3NF & BCNF are equivalent

- A relation is in BCNF, if and only if every determinant is a candidate key.

- Violation of BCNF may occur in a relation that
  - contains 2 (or more) composite keys
  - which overlap and share at least 1 attribute 3NF to BCNF

- Identify all candidate keys in the relation

- Identify all functional dependencies in the relation.

- If functional dependencies exist in the relation where their determinants are not candidate keys for the relation, remove the functional dependencies by placing them in a new relation along with a copy of their determinant.

Introduction To UML

- Objectives:

UML or Unified Modeling language is a specification which is used in the software engineering field. It can defined as a general purpose language which is used to design as graphical notation which is used an abstract model and this abstract model is used in the system. That system is called as UML or Unified Model language.

UML is a modeling language but not a methodology or process, the first concept is developed by Grady Booch, James Rambaugh and Ivar Jacobson at Rational Software.

This model is accepted as a standard by the Object Management Group (OMG), in 1997

Types of UML:

- The Main purpose of the class diagram is include the classes within a model. In the object oriented programming, the classes has certain attributes, operations (member function) and relationship among the objects.

- In the UML the class diagram can be include very easily.

- The fundamental part of the diagram is the class icon which is shown in the figure

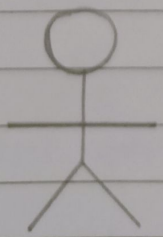
class
attributes
Member function

- A class icon is simply a rectangle divided into three compartments. The topmost compartment contains a list of attributes and the bottom compartment contains a list of operations.

- Usecase Diagram:

- A use case is a set of scenarios that shows an interaction betn a user a system. A use case diagram shows the relationship among actors and use cases.

- The two main components of a use case diagram are use cases and actors.



Use Case

Activity Diagrams:

- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

- In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step by step workflows of components in a system.

- An Activity diagram shows the overall flow of control.

- Activity diagram are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

→ rounded rectangles represent activities;

→ diamonds represent decisions;

→ bars represent the start (split) or end (join) of concurrent activites;

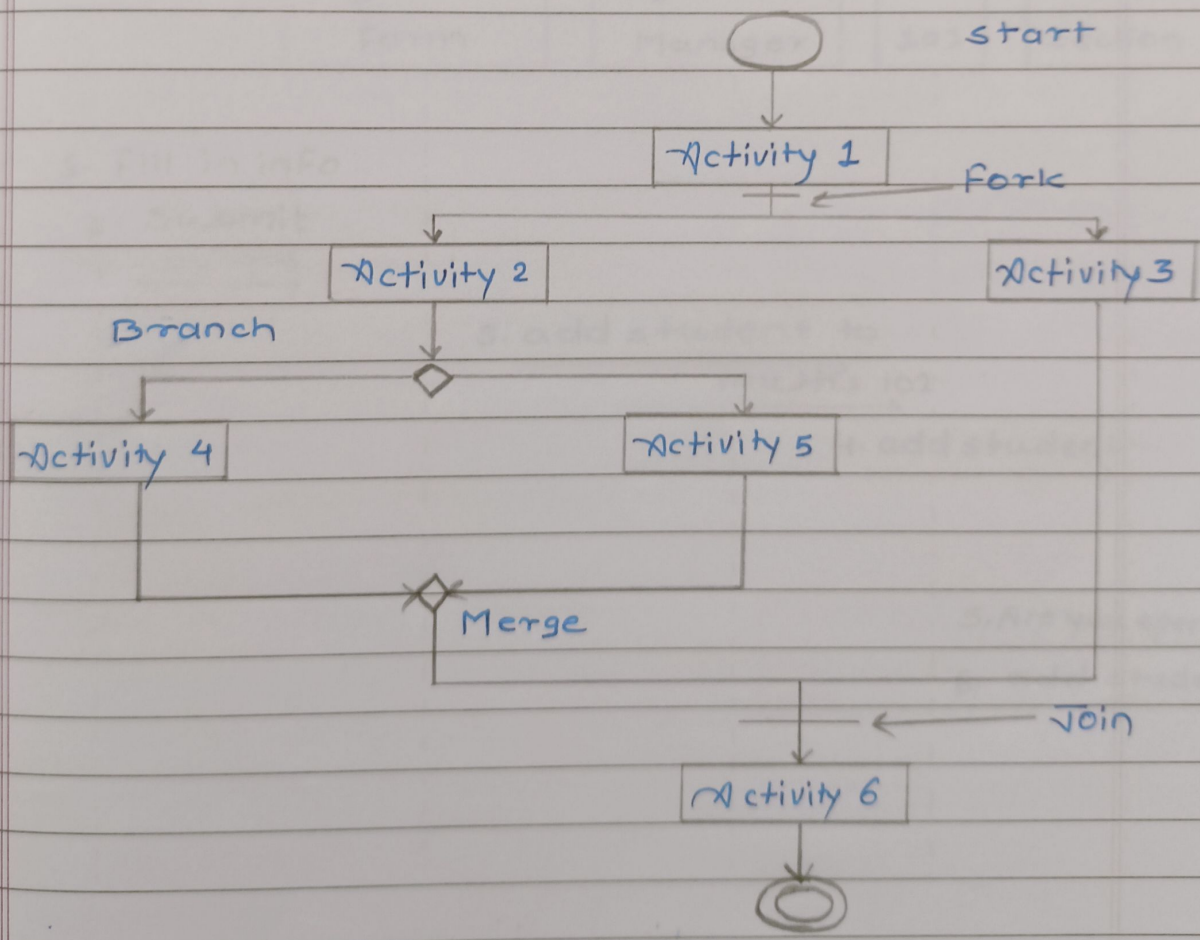
→ a black circle represents the start (initial state) of the workflow;

→ an encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen.

- Hence They can be regarded as a form of Flowchart.

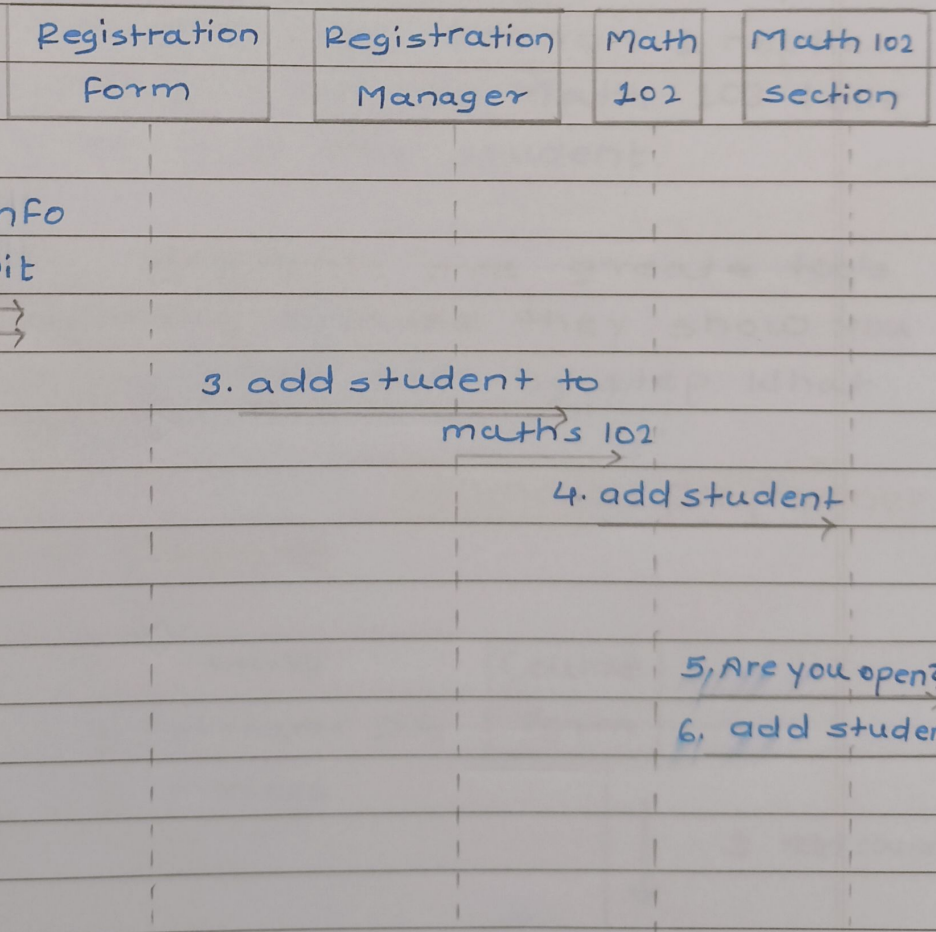
- Typical Flowchart techniques lack constructs for expressing concurrency.



Sequence Diagrams:

- Sequence diagrams is involved how to the object are interacted which are arranged in a time sequence.

- The sequence Diagram which is use the flow of events to determine what objects and interactions I will need to accomplish the functionality specified by the flow of events.



- Figure shows how a student successfully gets added to a course. The student let's call him Mahesh) fills in some information and submit the form.

- The form then talks to the manager and says "add Joe to Mahesh 102!"

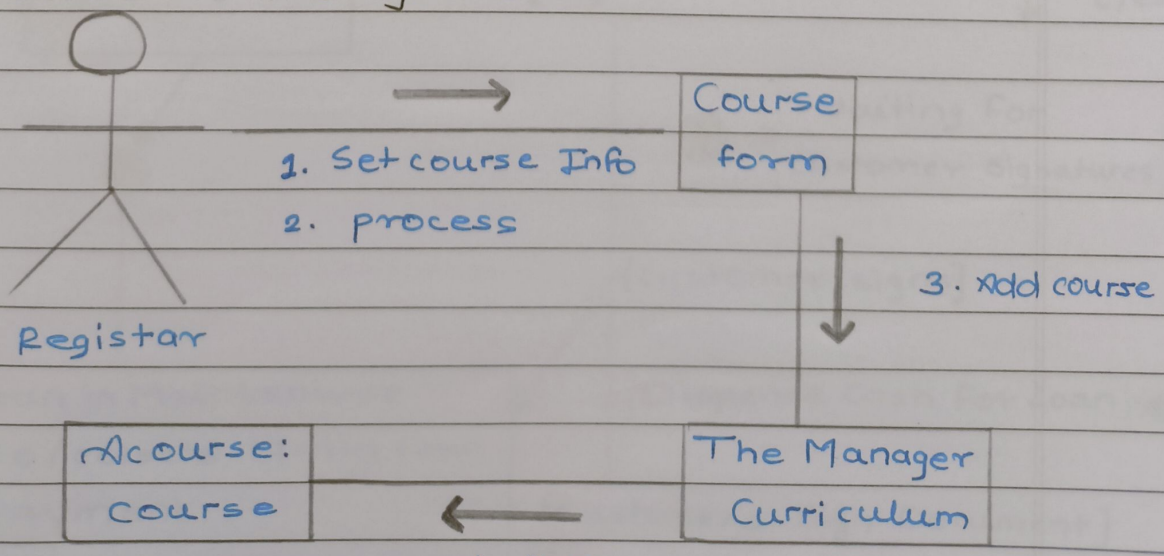
- The manager tells Math 102 that it has to add a student.

- Math 102 says to section 1 "are you open"? In this case, Section 1 replies that they are open, so Math 103 tells section 1 to add this student.

∴ sequence diagrams are great tools in the beginning because they show you and your customer step by step what has to happen.

@curious_.programmer

Collaboration Diagrams:

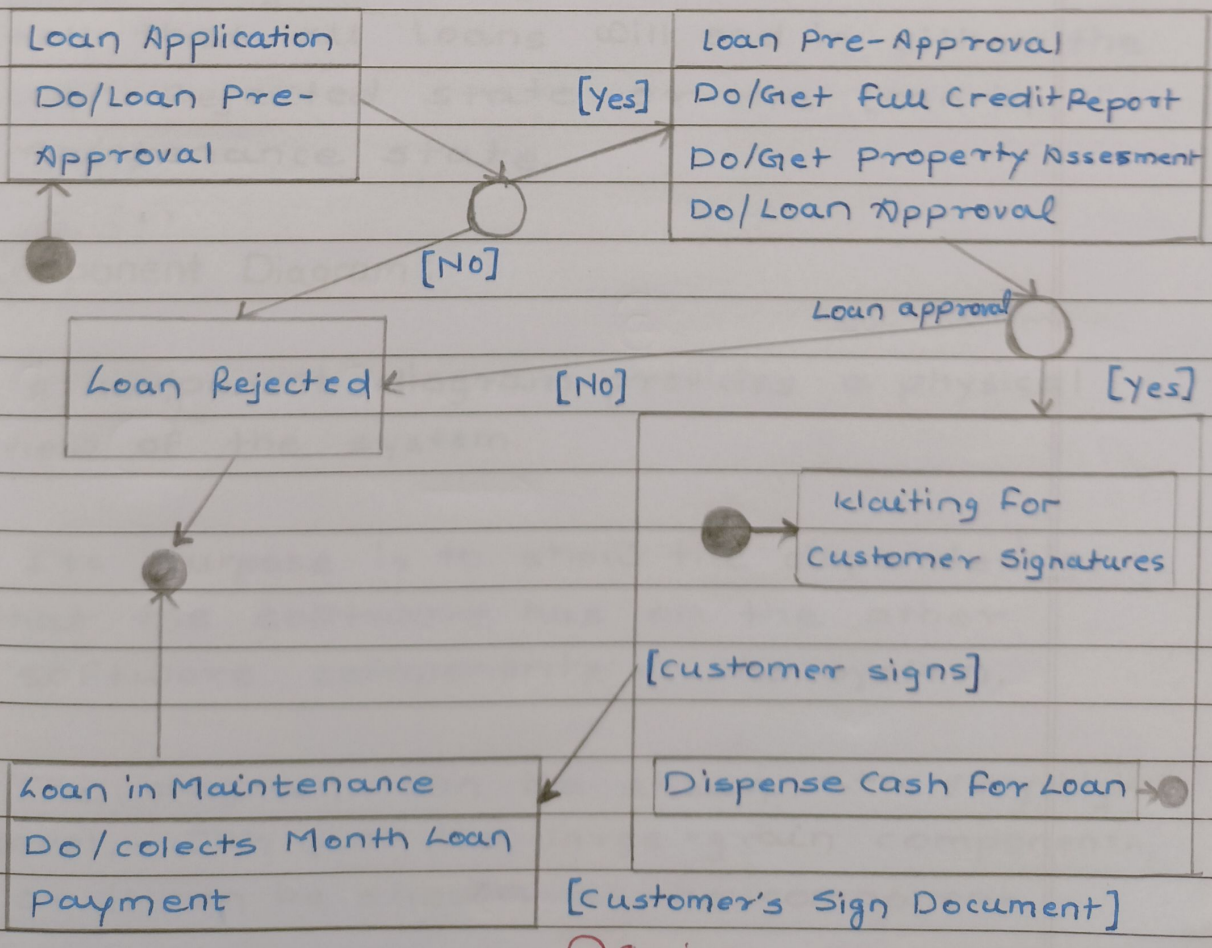


Statechart Diagram:

- The statechart diagram models the different states that a class can be in and how that class transitions from state to state.

- It can be argued that every class has a state, but that every class shouldn't have a statechard diagram.

- Only classes with "intresting" states -- that is, classes with three or more potential states during system activity -- should be modeled.



- The example statechart diagram in figure shows some of the vital information they can communicate.

- For example, you can tell that loan processing department to begin in the loan application state.

- When the pre-approval process is over, depending whatever output comes, and then you move to either the Loan pre-approved state or the Loan Rejected state.

- By Looking at diagram, a person can tell that all loans will end in either the Loan Rejected state or the Loan in Maintenance state.

- Component Diagram :

- A component diagram provides a physical view of the system.

- Its purpose is to show the dependencies that the software has on the other software components in the system.

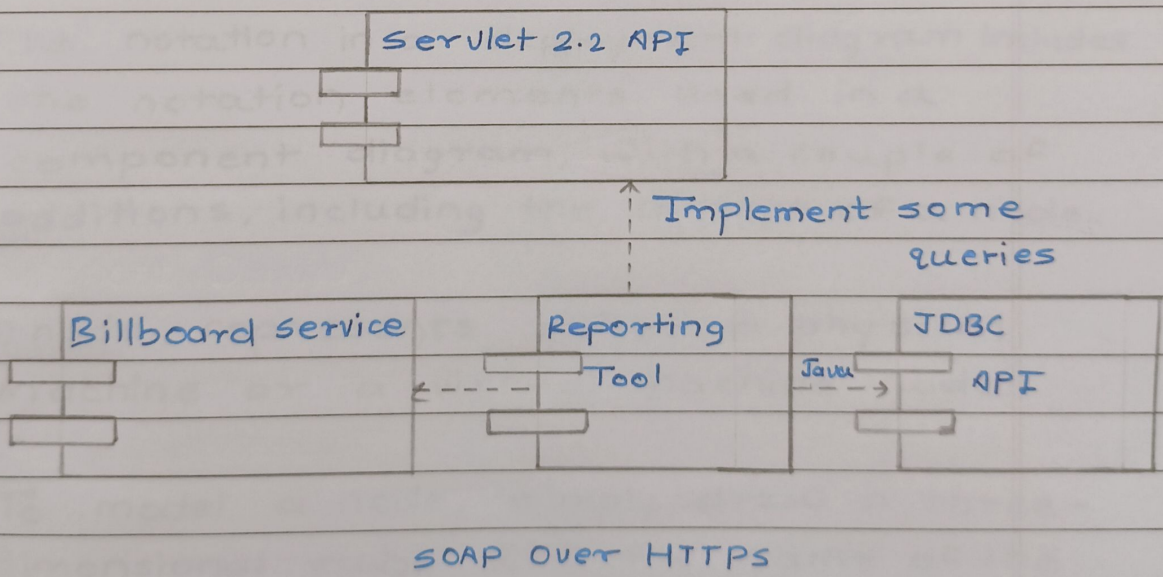
- The diagram can be shown at a very high level, with just the large-grain components, or it can be shown at the component

package level.

- Modeling a component diagram is best describe through an example.

- Figure shows four components : Reporting Tool, Billboard service, servlet 2.2. API, and JDBC API.

- The arrowed lines from the Reporting Tool component to the Billboard service, servlet 2.2 API, and JDBC API components mean that the Reporting Tool is dependent on those three components.



classmate
Date
Page

- Deployment Diagram:

- The deployment diagram shows how a system will be physically deployed in the hardware environment.

- It purpos is to show where the different components of the system will physically run and how they will communicate with each other.

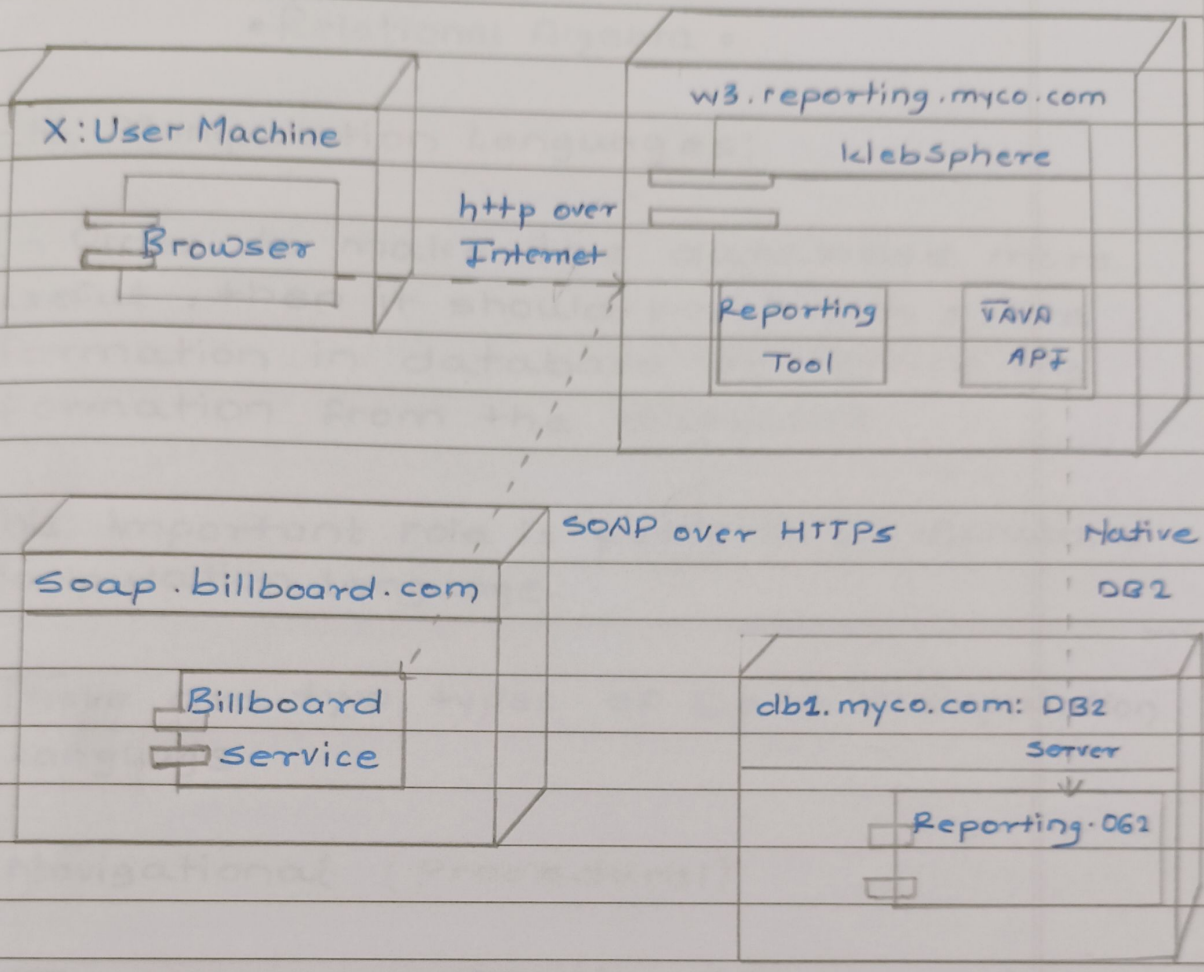
- since the diagram models the physical runtime, a system's production staff will make considerable use of this diagram.

- The notation in a deployment diagram includes the notation elements used in a component diagram, with a couple of additions, including the concept of a node.

- A node represents either a physical machine or a virtual machine node.

- To model a node, simply draw a three-dimensional cube with the name of the name of the node at the top of the cube.

- Use the naming convention used in sequence diagrams: [instance name] : [instance type] (e.g., "w3reporting.myco.com : Application Server").



- The deployment diagram in figure shows that the users access the Reporting Tool by using a browser running on their local machine and connecting via HTTP over their company's internet to the Reporting Tool.

- This tool physically runs on the Application server named w3reporting.myco.com. The diagram shows the Reporting Tool component drawn inside of IBM Web Sphere, which in turn is drawn inside of the node w3.reporting.myco.com.

• Relational Algebra •

- Data Manipulation Languages:

- In Order to make the database more useful, then it should possible to store information in database or retrive the information from the database.

- This important role is perform by database Manipulation language.

There are two types of Data Manipulation Language

→ Navigational (Procedural)

- The query specifies (to some extent) the statergy use to find the desired result e.g. relational algebra.

→ Non-Navigational (non-procedural)

- The query only specifies what data is wanted, not how to find it. e.g. relational calculus.

Introduction:

- Codd proposed a number of algebric operation for the relational database model.

- In the Relationship algebra there are two type of operation one is Unary operation and second one Binary operation

- Unary operation takes as input a single table and produces an output another table.

- Binary operations take as input two tables and produce as output another table.

- Fundamental operations:
  - Unary operation
    - projection operation (π)
    - Select operation (σ)
    - Rename Operation (ρ)
  - Binary Operation
    - SET operation
      - Union operation (U)
      - Intersection operation (∩)
      - Difference Operation (-)

- Join Operation ($\bowtie$)
- Cartesian Product Operation (x)
- Division Operation (÷)

- Selection operation

- The Selection operator select the row from the table that satisfy a given predicate.

- This operation allows to manipulate data in the single relation.

- The Selection operation is defined by the symbol called sigma (σ).

- The predicate is appear at subscript of sigma symbol (σ). The argument relation is present in the paranthesis after σ

Syntax:

σ

<predicate><comparsion_operator><Constant_value>(<input_table_name>)

Where Predicate: Name of the column in the Table

Comparsion_Operators: <, =, <=, >, =>, <>

Example:

Select all the student from the student table who's Roll no is greater than 300

Roll No	student Name	student_Address
134	Mary	Curry road
356	John	Dockyard
500	steve	Hepen Sea road

σ Roll No >300 (student)

Roll No	student_Name	Student_Address
356	John	Dockyard
500	Steve	Hepen sea road

- We can combine several predicates into a larger predicate by using the connectives and ($\wedge$), or ($\vee$), and not ($\neg$)

@ Curious_programmer

To Find the tuple in the student table where student name is john & roll no is greater than 300

σ Students_Name = "John" $\wedge$ RollNo > 300 (student)

Roll no	Student Name	Students_Address
356	John	Dockyard

• Projection Operation (π)

• This operator is used to select some of the attribute from the table to produce a desired result set.

Note that projection operation is used to eliminates the duplicates records in the table.

Syntax.

$$\Pi <attributes> (<Input\_Table\_Name>)$$

Example:

1) Find the all record from the student table

Π Roll No, Students_Name, Students_Address (Student)

Roll No	students Name	student_Address
134	Mary	3 Curry Road
356	John	Dockyard
500	steve	Nepen Sea Road