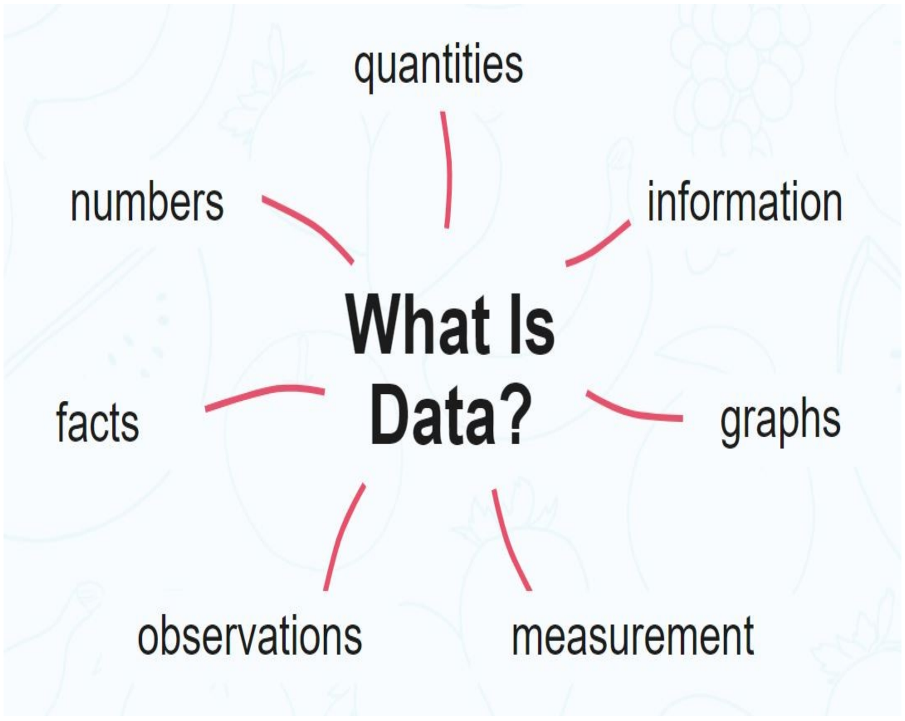
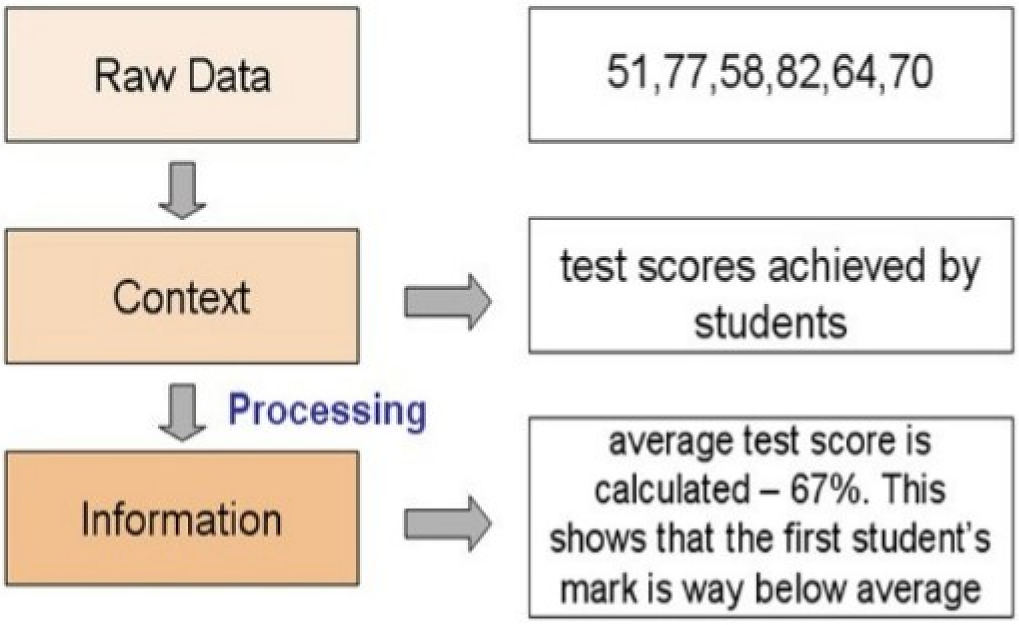
Database: Introduction, Concepts & Architecture

(Course Instructor: Bidur Devkota)

Database: collection of Data

- Details of friends you have in Facebook.
- Your record stored in the Collage.




Raw Data
51,77,58,82,64,70
Context
test scores achieved by students
Processing
Information
average test score is calculated – 67%. This shows that the first student's mark is way below average

Data and Information

Data:

- Unit of information
- raw, unprocessed facts and figures
- lowest level of abstraction in representation.

Information

- Data with context
- When data is processed, organized , structured of presented in a given context to make it useful, it is called information.

Traditional File Processing System

- Before the advent of Computerized system, a manual file system used to maintain the records and files (in hard-copy format).

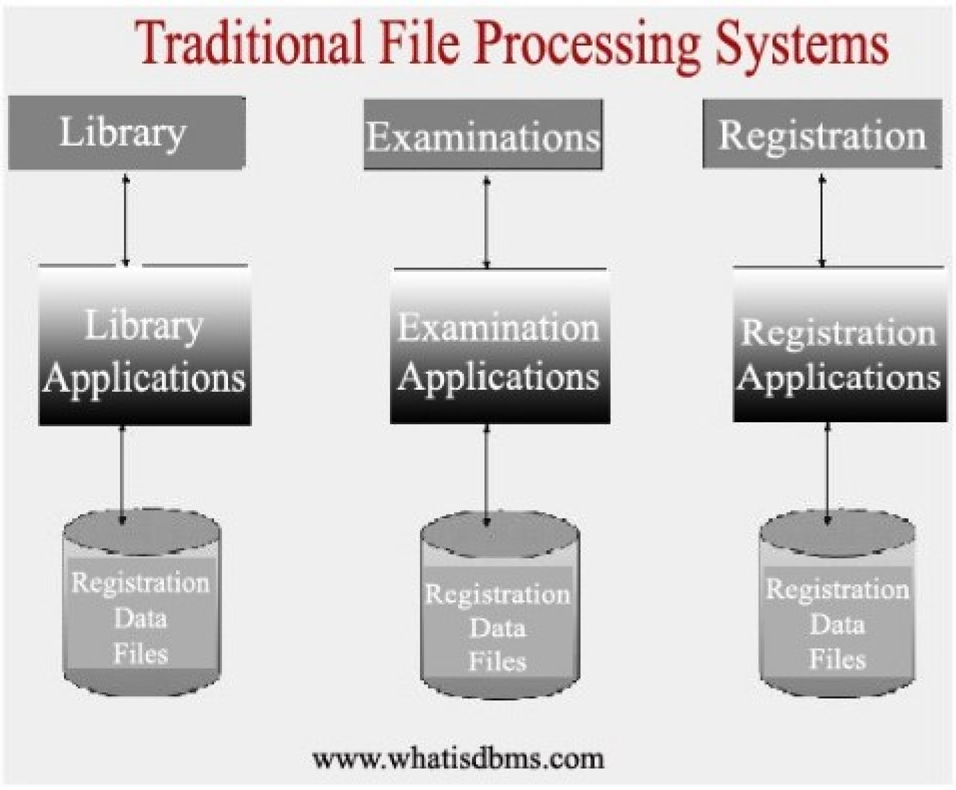
- All the data stored in files and it makes it easy to find any information.
    - good **only** for small organizations having small number of items.

- It has many **disadvantages**:
    - It was **time consuming**.
    - **Inefficient** to maintain the record of big firm having large number of items.
    - It requires a **lots of labor work** to do.
    - It becomes more complex when anyone requires **changing** the information

- **Computerized File Processing System** was introduced after the invention of computers. It was totally computer based system where all the **information is store in different computer files**.

    - Also traditional files system stores data in a manner that all the departments of an organization have their own set of files that creates **data redundancy**.
    - The data and program depends upon each other. And due to this, any change in one file will affect all the others files too. => i.e. **NO DATA INDEPENDENCE.**

- It is **less flexible** and has many limitations.
- It is very difficult to maintain file processing system.
- Any change in one file affects all the files that creates burden on the programmer.
- File in Traditional File Processing Systems are called **flat files**.



Disadvantages of File system over DBMS

Most explicit and major disadvantages of file system when compared to database management system are as follows:

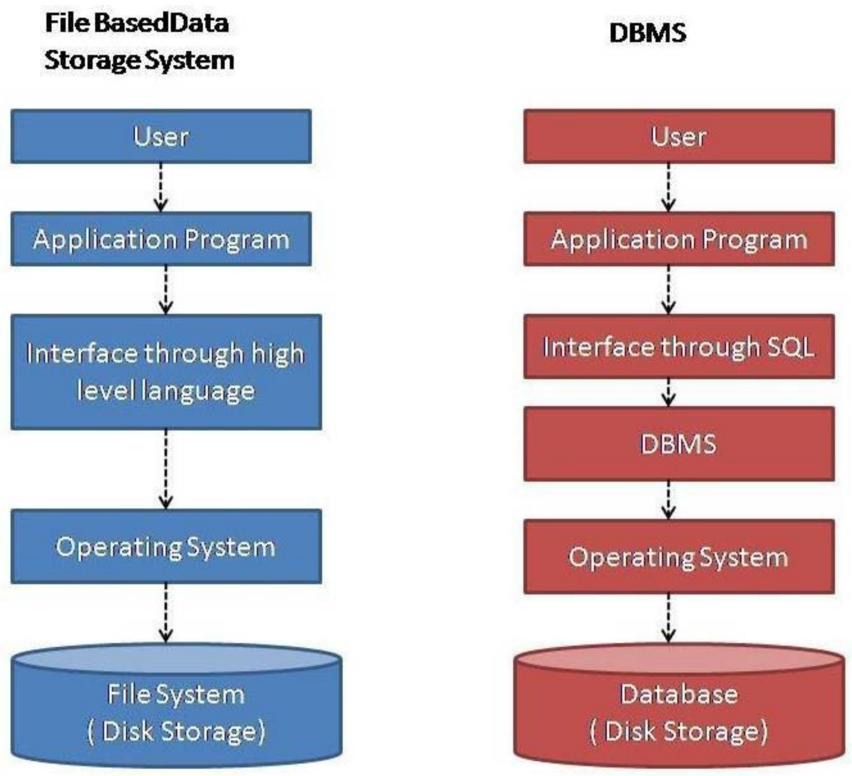
- ***Data Redundancy:*** The files are created in the file system as and when required by an enterprise over its growth path. So in that case the repetition of information about an entity cannot be avoided.
  - For Eg. The addresses of customers will be present in the file maintaining information about customers holding savings account and also the address of the customers will be present in file maintaining the current account. Even when same customer have a saving account and current account his address will be present at two places.

- ***Data Inconsistency*****:** Data redundancy leads to greater problem than just wasting the storage i.e. it may lead to inconsistent data. Same data which has been repeated at several places may not match after it has been updated at some places.
  - For example: Suppose the customer requests to change the address for his account in the Bank and the Program is executed to update the saving bank account file only but his current bank account file is not updated. Afterwards the addresses of the same customer present in saving bank account file and current

bank account file will not match. Moreover there may not be a way to find out which address is latest out of these two.

- ***Difficulty in Accessing Data*:** For generating ad hoc reports the programs will **not already be present** and only options present will to write a new program to generate requested report or to work manually. This is going to take impractical time and will be more expensive.
  - For E.g.: Suppose all of sudden the administrator gets a request to generate a list of all the customers holding the saving banks account who lives in particular locality of the city. Administrator will not have any program already written to generate that list but say he has a program which can generate a list of all the customers holding the savings account. Then he can either provide the information by going through the list manually to select the customers living in the particular locality or he can write a new program to generate the new list. Both of these ways will take large time which would generally be impractical.

- ***Data Isolation*:** Since the data files are created at different times and supposedly by different people the structures of different files generally will not match. The data will be scattered in different files for a particular entity. So it will be difficult to obtain appropriate data.
  - For e.g.: Suppose the Address in Saving Account file have fields: ***Address line1, Address line2, City, State, Pin*** while the fields in address of Current account are: ***House No.,Street No., Locality*, City, State, Pin**.
  - **Administrator is asked to provide the list of customers from a particular locality.** Providing consolidated list of all the customers will require looking in both files. But they both have different way of storing the address. Writing a program to generate such a list will be difficult.



- ***Integrity Problems****:* All the consistency constraints have to be applied to database through appropriate checks in the coded programs. This is very difficult when number such constraint is very large.
    - For e.g.: An account should not have balance less than Rs. 500. To enforce this constraint appropriate check should be added in the program which add a record and the program which withdraw from an account. Suppose later on this amount limit is increased then all those check should be updated to avoid inconsistency. These time to time changes in the programs will be of inconvenience for the administrator.

- ***Security and access control:*** Database should be protected from unauthorized users. Every user should not be allowed to access every data. Since application programs are added to the system
    - For e.g.: The Payroll Personnel in a bank should not be allowed to access accounts information of the customers.

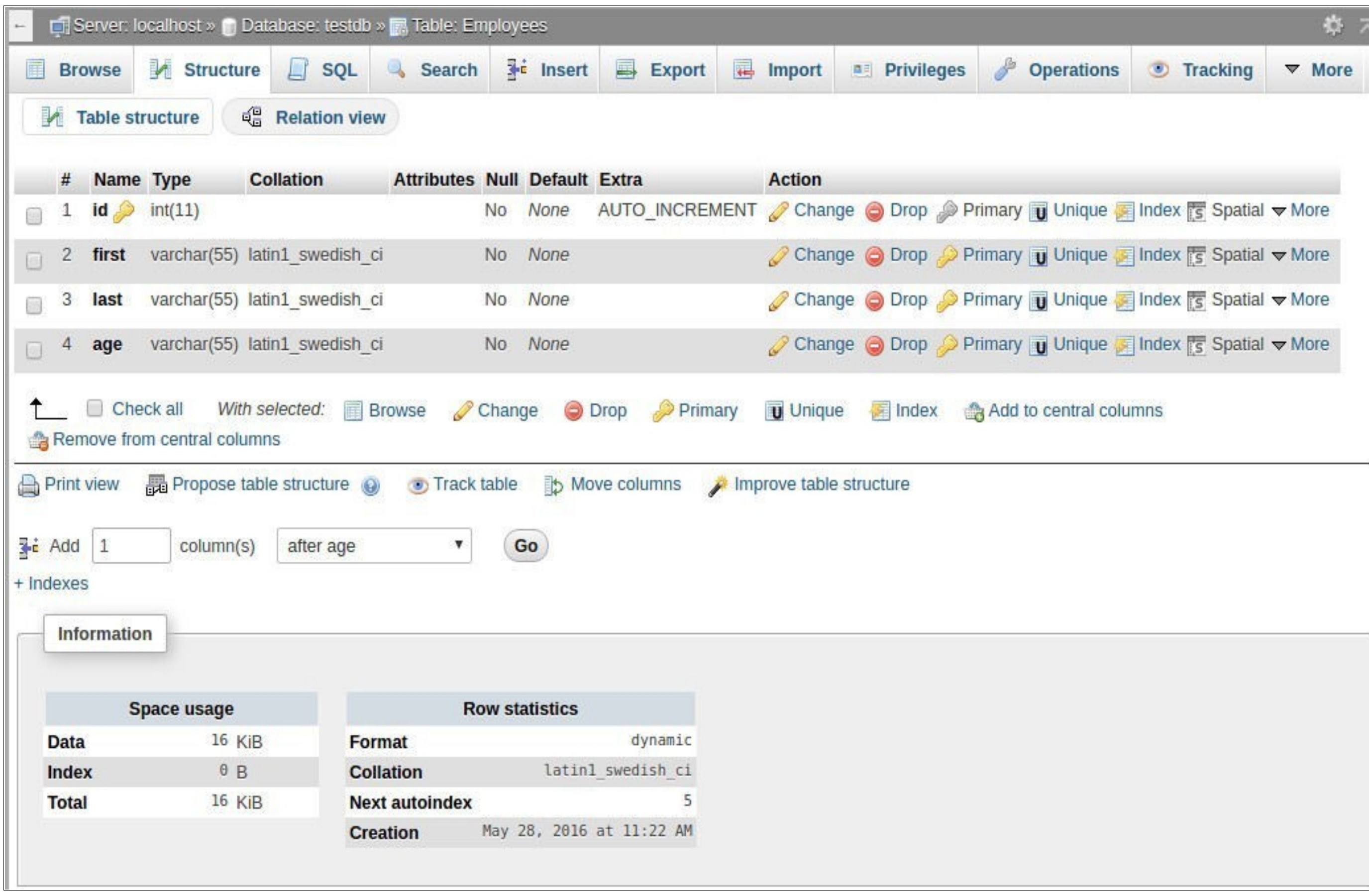
- ***Concurrency Problems:***
    - Data **access** operation with **more than one users** at the **same time**.
    - If in that environment two or more users try to **update a shared data element** at about the **same time** then it may result into inconsistent data.
    - For e.g.: Suppose Balance of an account is Rs. 500. And User A and B try to withdraw Rs 100 and Rs 50 respectively at almost the same time using the Update process.
        - **Update**:
            - Read the balance amount.
            - Subtract the withdrawn amount from balance.
            - Write updated Balance value.
        - Suppose A performs Step 1 and 2 on the balance amount i.e it reads 500 and subtract 100 from it. But at the same time B withdraws Rs 50 and he performs the Update process and he also reads the balance as 500 subtract 50 and writes back 450. User A will also write his updated Balance amount as 400. They may update the Balance value in any order depending on various reasons concerning to system being used by both of the users. So finally the balance will be either equal to 400 or 450. Both of these values are wrong for the updated balance and so now the balance amount is having inconsistent value forever.

Database Approach

- Database is an organized **collection of data.**
    - repository or container for a collection of data files.
- In database, data may be classified according to their organizational approach such as distributed database, object oriented database, relational database, etc.
- A database is the collection of related **persistent data** and contains information relevant to an enterprise.
- For e.g., library database for maintaining information about students, books, etc.
- Database consists of following four components
    i. Data item
    ii. Relationships
    iii. Constraints (i.e. criteria)
    iv. Schema (plan of data)

Characteristics of Database Approach

1) **Self describing nature of database system:**
   - Database system contains
     - target data
     - description about the data (database structure).
   - This definition is stored in the **DBMS catalog** which provides information such as:
     - the structure of each file
     - the type of data
     - storage format
   - In traditional file management system data definition was not the part of application program.



Server: localhost » Database: testdb » Table: Employees

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐	1	**id**	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
☐	2	**first**	varchar(55)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial More
☐	3	**last**	varchar(55)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial More
☐	4	**age**	varchar(55)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index Spatial More

Check all *With selected:* Browse Change Drop Primary Unique Index Add to central columns Remove from central columns

Print view Propose table structure Track table Move columns Improve table structure

Add 1 column(s) after age Go

+ Indexes

Information

Space usage	
Data	16 KiB
Index	0 B
Total	16 KiB

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	5
Creation	May 28, 2016 at 11:22 AM

2) **Represent Some Aspects of real world applications**

- A database represents some features of real world applications.
- Any change in the real world is reflected in the database.
- If we have some changes in our real applications like Library Management System (LMS) then it will be reflected in database too.
- **For example**, In LMS; we have in our mind certain applications of maintaining records of book borrower list, due dates, etc. related to each book.

3) **Insulation between programs and data** :

- In file processing system,
  - the **structure of data** file is **embedded** in the application program.
  - any changes in the format of the data → the whole **application program will also have to be changed.**
- In DBMS, database and the application program are **separately** situated.
  - **The structure of data files is stored in the DBMS catalog separately from the access programs.**
  - Hence in most cases DBMS access programs do not need such changes.
  - This property is **program-data independence**:
    - **System data (Meta Data) descriptions are separated from the application programs**.
    - Changes to the data structure is handled by the DBMS and not embedded in the program.
- For example, **adding an extra filed in existing data**:
    - a file access program may be written in such a way that it can just access the **name** of a customer of any company but if we want to add another field, say, **address** of the customer, then in such a case this program will no longer be useful. But in the case of the DBMS, we just have to add the new field "address" in the catalog and the next time DBMS refers to the catalog, the new structure of the records will be accessed and hence we do not have to make change in the program

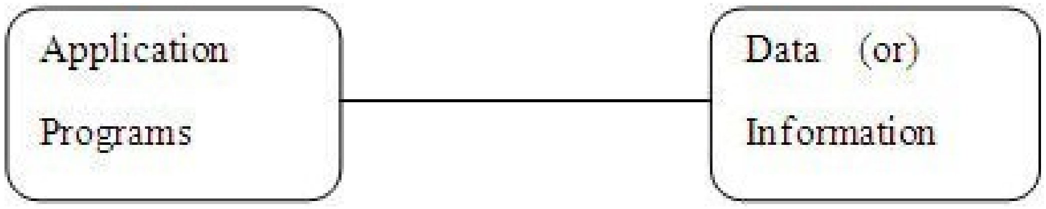


Fig : Conventional data processing without data independence

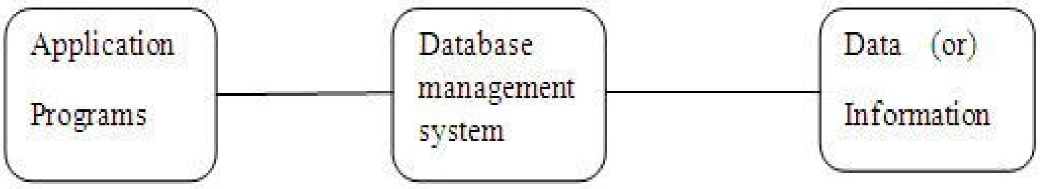


Fig : Database system with data independence

- The advantages of data independence in DBMS are as follows:
  - Ability of improving performance
  - Alterations in data structure does not requires alterations in application programs
  - Implementation details can be hidden from the users
  - Affordable prices of maintaining system
  - Permit users to focus on general structure
  - Enforcement of standards
  - Improvement of security
  - The state of being undamaged or undivided can be improved.

4) **Support of Multiple view of data:**
- A view is a **subset of the database**.
- A view is defined and devoted for a particular user of the system.
- **A database typically has many users, and their perspective of viewing the database is different.**
- Consider an example of **student information system of a college**, where all the data of student, courses, information of college etc are stored in a database.
- It is obvious that there is more than one user of this system and also their **interest is different.**
  - student are generally interested in finding out the marks obtained
  - But, teachers are interested to add/update marks, view information about students,
  - Similarly, outsiders/newcomers/visitors are interested to find the information of the college.
- In the example given below, every user is accessing data from the same database but what they see and can access is different.
- **A view may be subset of the database or it may contain virtual data that is derived from the database files but is not stored explicitly.**

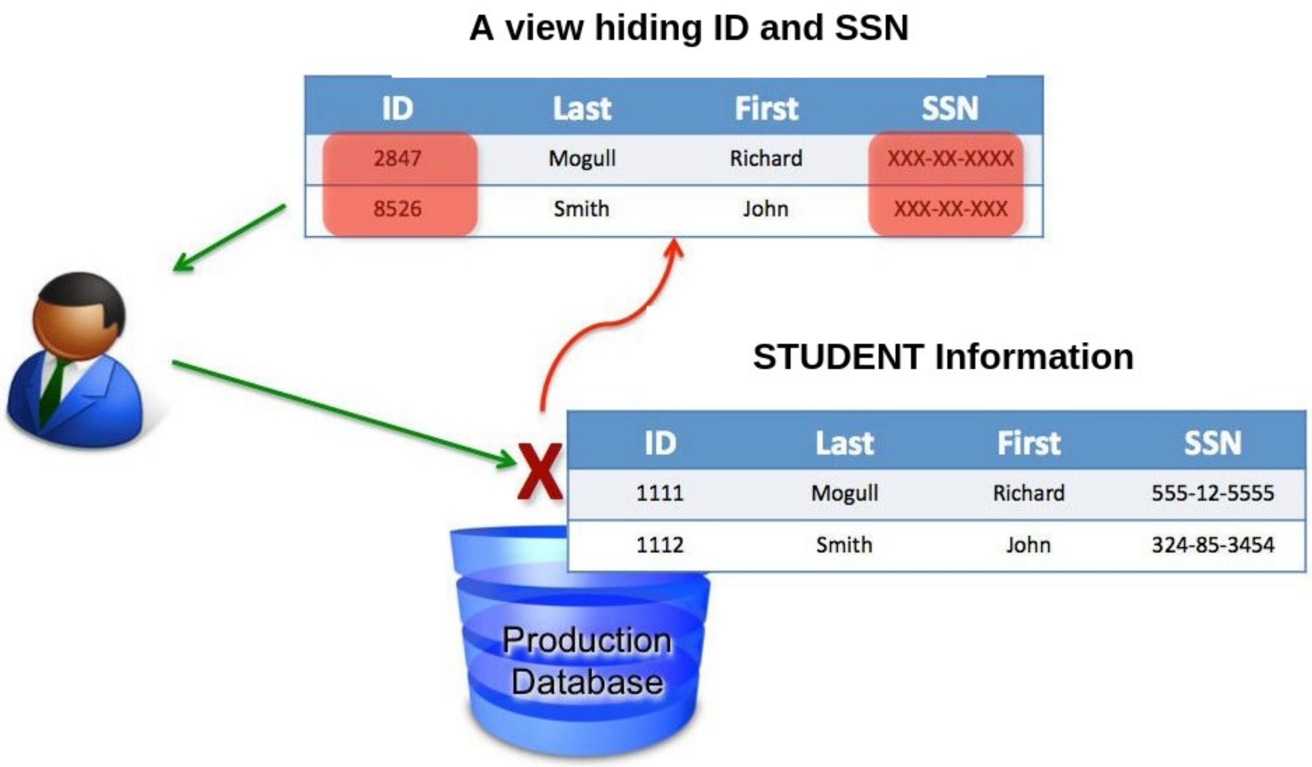


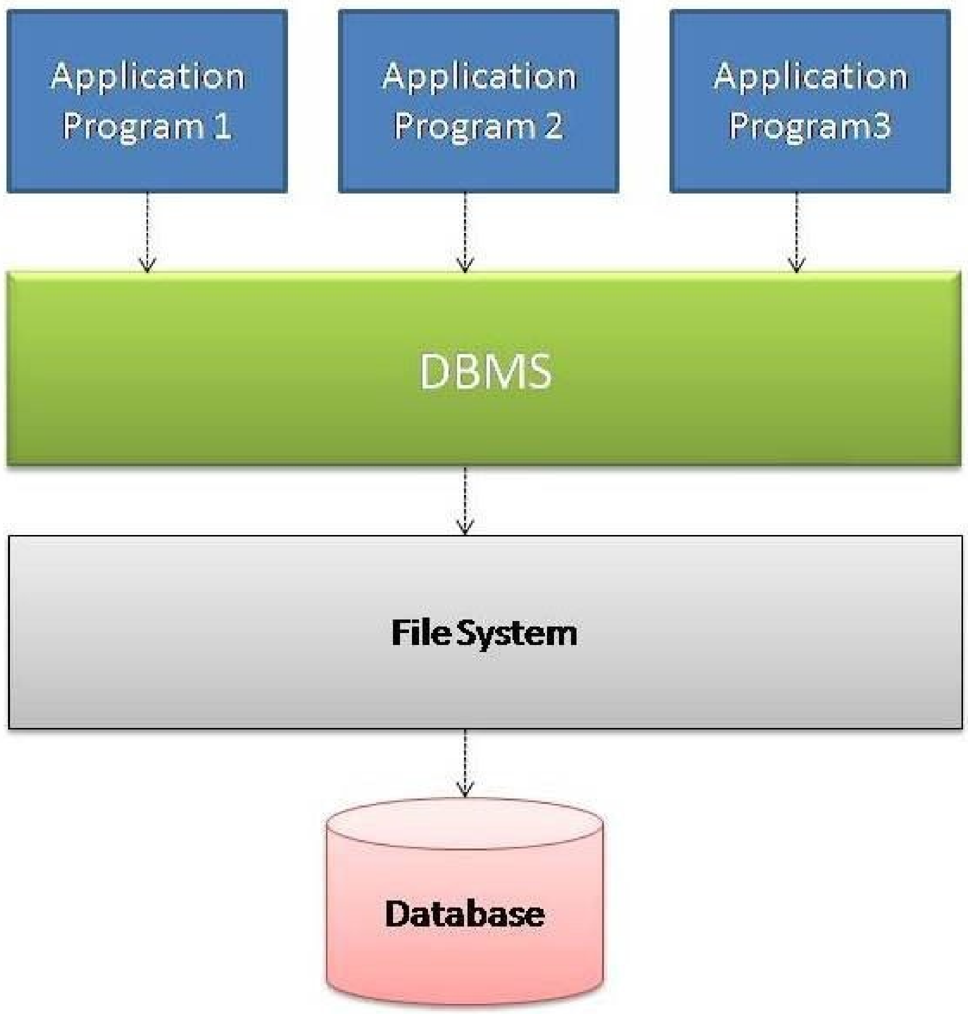
Fig: Database View as per the specific user needs (Unnecessary details is masked)

5) **Sharing of data and multi user transaction processing:**
   - Database system should allow **multiple users to access** same database at the same time.
   - Hence, the multiuser DBMS must have **concurrency control** strategies to ensure several users access to the same data item at the same time, and to do so in a manner that the data will always be correct –**data integrity**.
   - Example: **online air ticket reservation system:**
     - many users are accessing the same site at the same time.
     - Hence for every seat, DBMS should ensure that only one user should be given access to reserve the seat.
     - Hence to ensure the correctness of this type of transaction DBMS must include **concurrency control** software to ensure that several users trying to update same data do so in a controlled manner so that the result of the updates is correct.
     - In order to do so, DBMS uses lock based protocol and time stamp based protocol.

DBMS (Database Management System)

- Management of data basically involves
  - store data
  - provides a mechanism to access and manipulation data.

- Just a computer software program that allows a user to perform a database functions of storing, retrieving, deleting and modifying data.



- Database management systems are basically designed to manage large volume of information.
- DBMS is the interface between the users and the database.
- Example of DBMS is MySQL, Oracle, MS-SQL, etc.

Application of DBMS

- Computerized Library System:
  used to store the information of books, borrowers, etc.
- Banking
  Used for recording the customer information and transaction information.
- Content management systems (CMS):
  CMS like wordpress, Laravel, etc store websites as collections of webpages in a database.
- Airlines:
  For reservations and schedule information, etc.
- Telecommunication:
  To keep records of customers, call made, balance left, generating monthly bills, etc.
- Universities:
  To keep records of students, courses, marks of students etc.
- Sales:
  To keep information of customers, products list, purchase information etc.
- Manufacturing:
  To store orders, tracking production of items etc.
- Human Resources:
  To keep records of employee, their salary, bonus etc.

Data Abstraction:

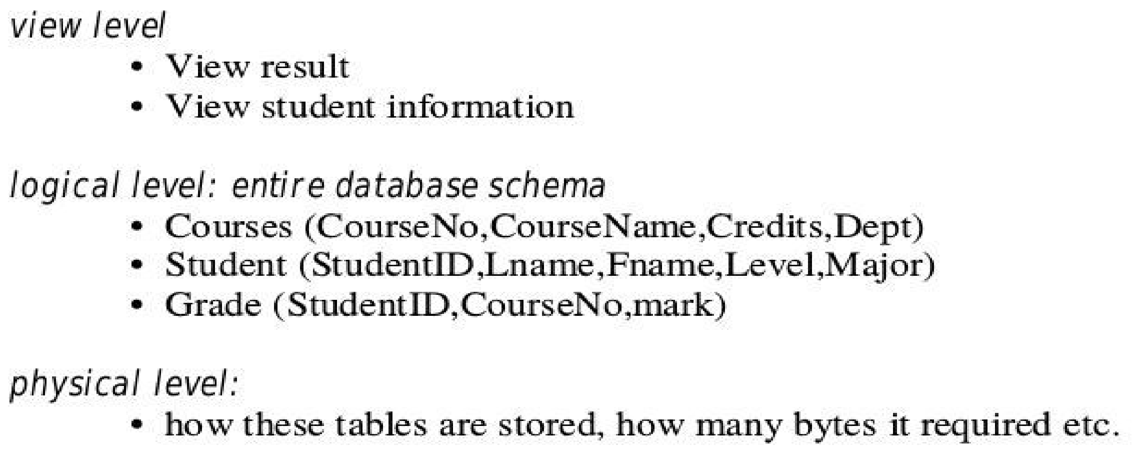
- The technique of hiding the complexity of the database to its users.
- There are three levels of data abstraction:

1. **Physical Level or Internal Level:**
    - lowest **level of abstraction**
    - describes **how the data** in the database are **actually stored**.
    - This level describes complex low-level data structures in detail and is concerned with the way the data is physically stored.
    - Data only exists at physical level.

2. **Logical Level or Conceptual Level:**
    - Next higher level of abstraction
    - describes
        - what data are stored in the database,
        - what relationships exist among those data.
    - It describes the structure of whole database and **hides details of physical storage structure**.
    - Describes entities, data types, relationships, attributes and constraints.
    - All **views** must be **derivable** from this **conceptual schema.**
    - **It describes the actual data stored in the database in the form of tables and relates them by means of mapping.**
    - This level will not have any information on what a user views at external level. This level will have all the data in the database.

3. **View Level or External Level:**
    - **highest level of abstraction**
    - concerned with the way the data is **seen by individual users**.
    - users see the data in the form of **rows and columns**.
    - Users see how the data is stored in terms of **tables and relations.**
    - Users view full or partial data based on the business requirement.

- **The users will have different views here, based on their levels of access rights.**
- For example, student will not have access to see Lecturers salary details, one employee will not have access to see other employees details, unless he is a manager. At this level, one can access the data from database and perform some calculations based on the data.



view level

- View result
- View student information

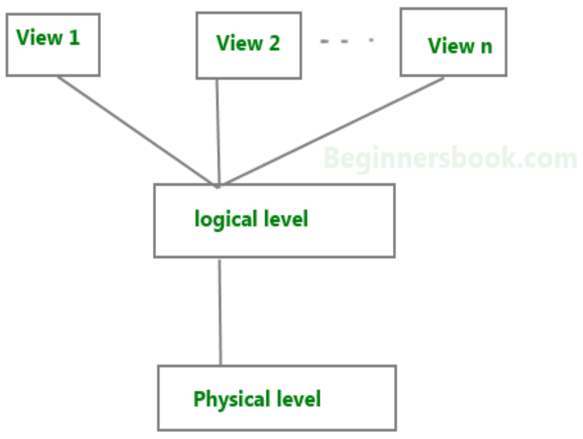
logical level: entire database schema

- Courses (CourseNo,CourseName,Credits,Dept)
- Student (StudentID,Lname,Fname,Level,Major)
- Grade (StudentID,CourseNo,mark)

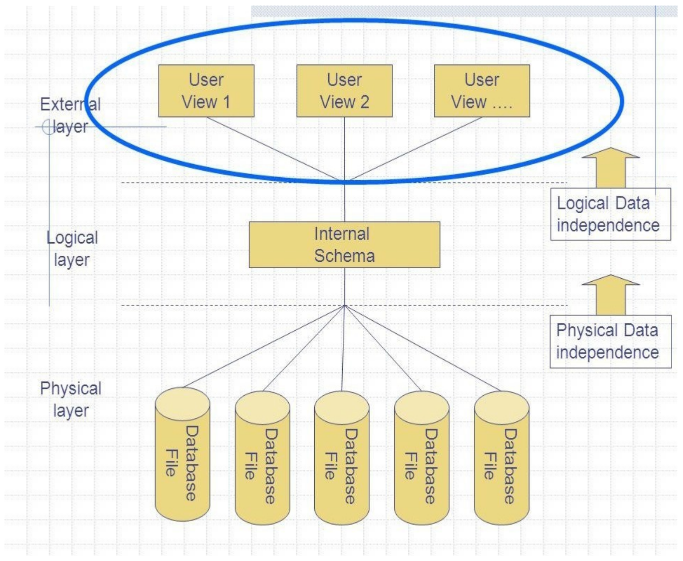
physical level:

- how these tables are stored, how many bytes it required etc.

- **Example**:
    - Let's say we are storing customer information in a customer table. At **physical level** these records can be described as **blocks of storage** (bytes, gigabytes, terabytes etc.) in memory. These details are **often hidden from the programmers.**
    - At the **logical level** these records can be described as **fields and attributes** along with their **data types, their relationship** among each other can be logically implemented. The p**rogrammers generally work at this level because** they are aware of such things about database systems.
    - At **view level**, user just **interact with system with the help of GUI** and enter the details at the screen, they are **not aware of how the data is stored and what data is stored;** such details are hidden from them.
- The **DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the database.** The process of transforming requests and results between levels is called **mapping**.



Three Levels of data abstraction



Physical Data Independence means the change in the physical storage/level can be made without affecting the conceptual or external view of the data. The new changes are absorbed by mapping techniques. **Logical data independence** is the ability to modify the logical schema without causing application program to be rewritten.

Instances and Schemas:

- **Instances (database state):**
  - Databases change over time as information is inserted and deleted.
  - The **collection of information stored in the database at a particular moment is called an instance** of the database. It is also known as database state.
- **Schema:**
  - The overall design of the database which is **not expected to change frequently** is called the database schema.
  - There are three schemas, partitioned according to the levels of abstraction.
  - The physical schema describes the database design at physical level.
  - The logical schema describes the database design at the logical level.
  - The schema at the view level is sometimes called subschema and describes the view of the database.
  - A database may have several subschemas.

Data Models:

- Data models define how the **logical structure of a database** is modeled.
- The **basic design structure of the database is the data model**.
- A data model is a **collection of conceptual tools** for **describing data, data relationships, data semantics, and consistency constraints**.
- Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system.
- Popular Models – **ER Model,** Relational Model, etc

Relational Model:

- In the relational model of a database, all **data is represented in terms of tuples, grouped into relations** (table).
- Structured query language (**SQL**) is used to manipulate data stored in tables.
- A database organized in terms of the relational model is a relational database.
- It is more scientific a model than others.
- This model is based on first-order predicate logic and defines a table as an n-ary relation.
- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

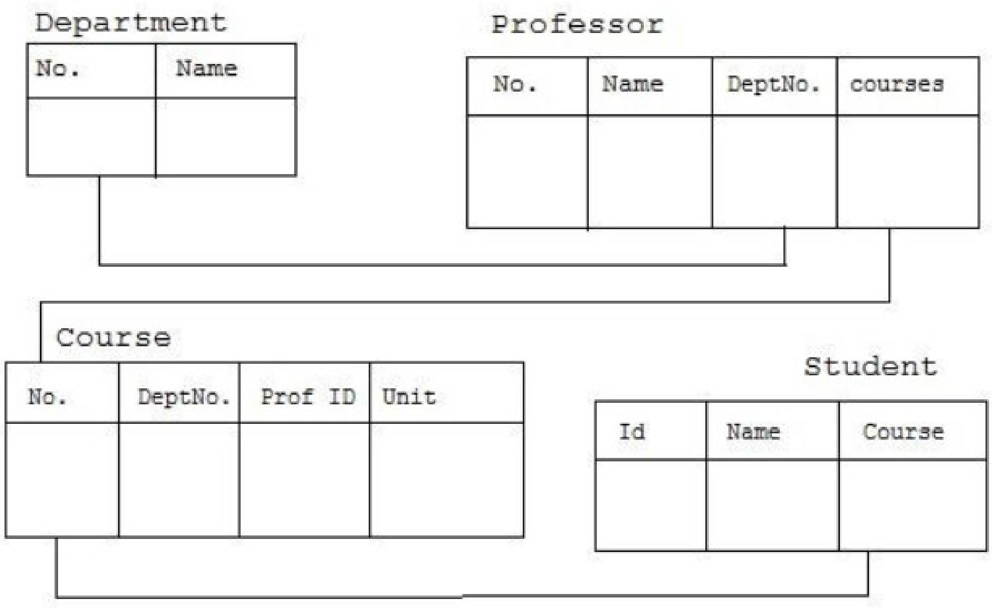
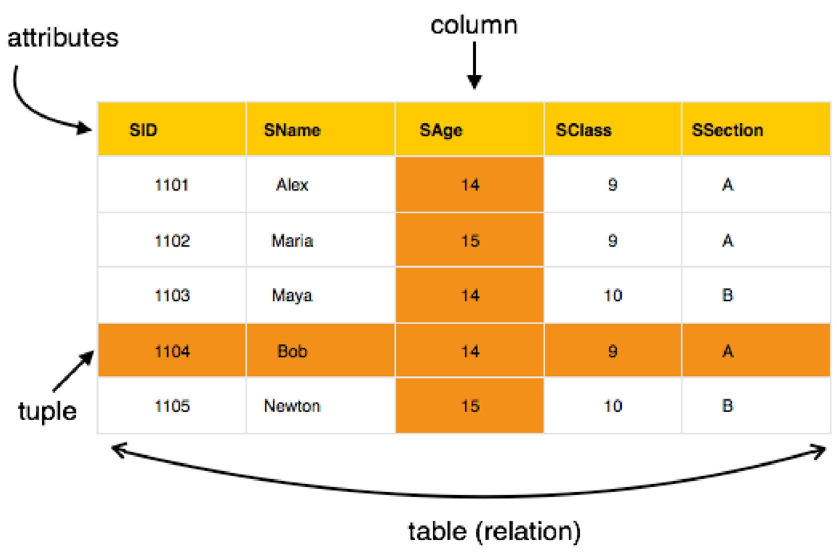


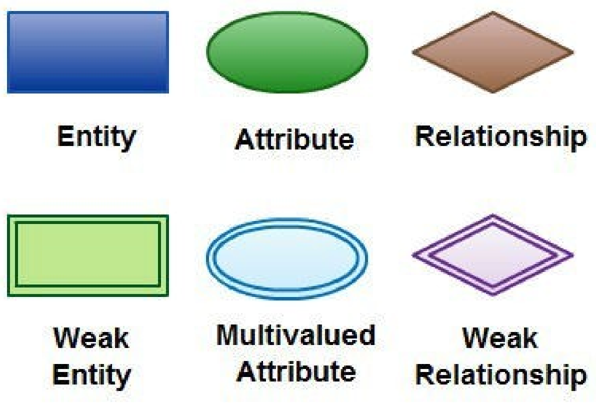
Illustration 1: Fig: Relational Data Model


attributes
column
tuple
table (relation)

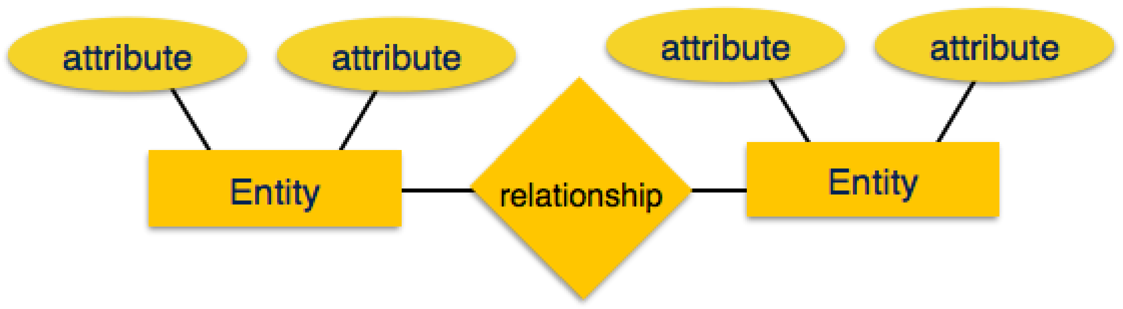
SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

Entity-Relationship (ER) Model:

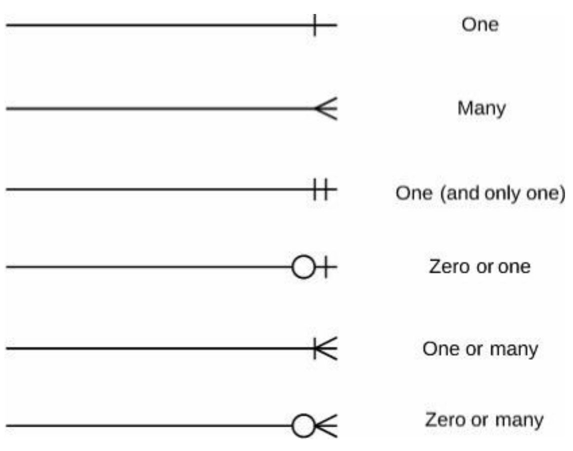
- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them.
- The ER Model creates entity set, relationship set, general attributes and constraints.
- Overall logical structure of a database can be expressed graphically by E-R diagram.



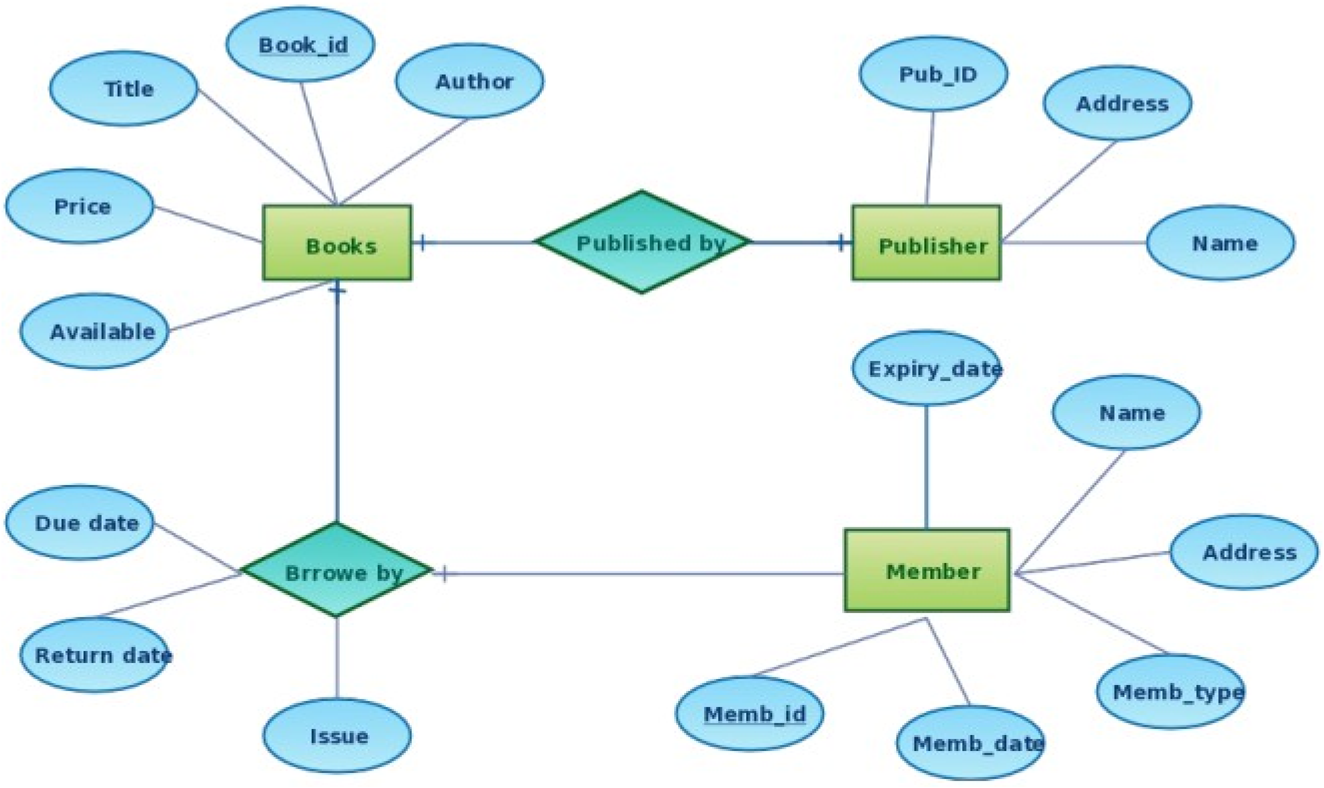
- ER Model is best used for the conceptual design of a database.ER Model is based on:
  - **Entities** and their attributes.
  - **Relationships** among entities.
- The basic components of this diagram are:
  - Rectangles (represent entity sets)
  - Ellipses (represent attributes)
  - Diamonds (represent relationship sets among entity sets)
  - Lines (link attributes to entity sets and entity sets to relationship sets)



- **Entity:**
  - An entity in an ER Model is a real-world entity having properties called **attributes**.
  - Every **attribute** is defined by its set of values called **domain**.
  - For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship:**
  - The logical association among entities is called **relationship**.
  - Relationships are mapped with entities in various ways.
  -
  - Mapping cardinalities define the number of association between two entities.A relationship is an association among several entities. The set of all entities of the same type is called an entity set and the set of all relationships of the same type is called a relationship set.
  - Mapping cardinalities:
    - one to one
    - one to many
    - many to one
    - many to many

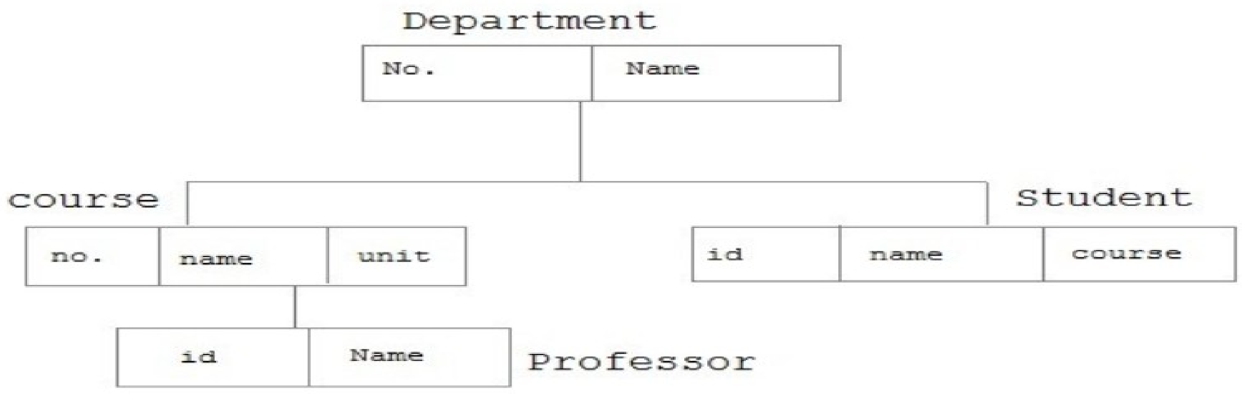


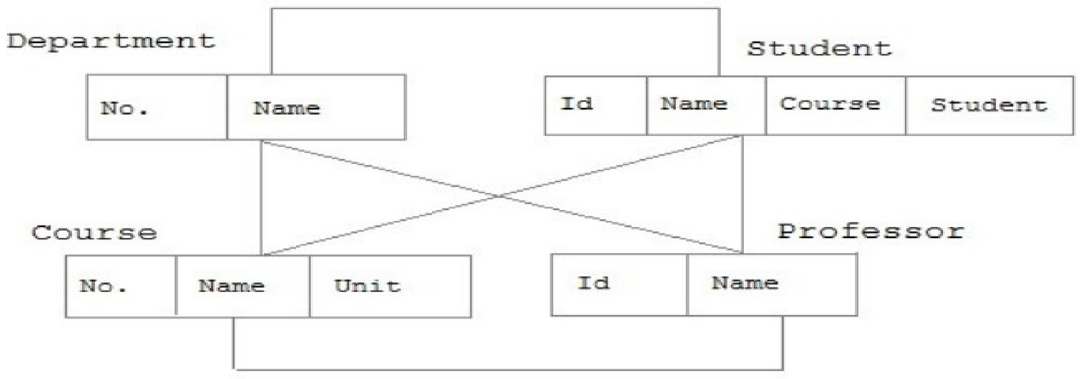
E-R Diagram for Library Management System



Other Data Models:

- **Object-based data models (Object-oriented and Object-relational):**
  - Object oriented data model is extension to E-R model with the notion of encapsulation, methods (functions) and object identity.
  - It is based on collection of objects, like the E-R model.
  - An object contains values stored in instance variables within the object.
- **Network model:**
  - In network model, data are represented by the set of records and relationships among data are represented by links.
- **Hierarchical model:**
  - Hierarchical model also represents data by a set of records but records are organized in hierarchical or order structure and database is a collection of such disjoint trees.
  - The nodes of the tree represent record types.
  - Hierarchical tree consists one root record type along with zero more occurrences of its dependent subtree and each dependent subtree is again hierarchical.





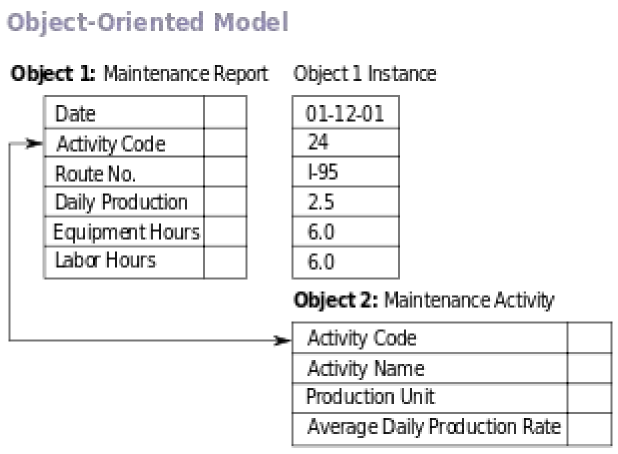
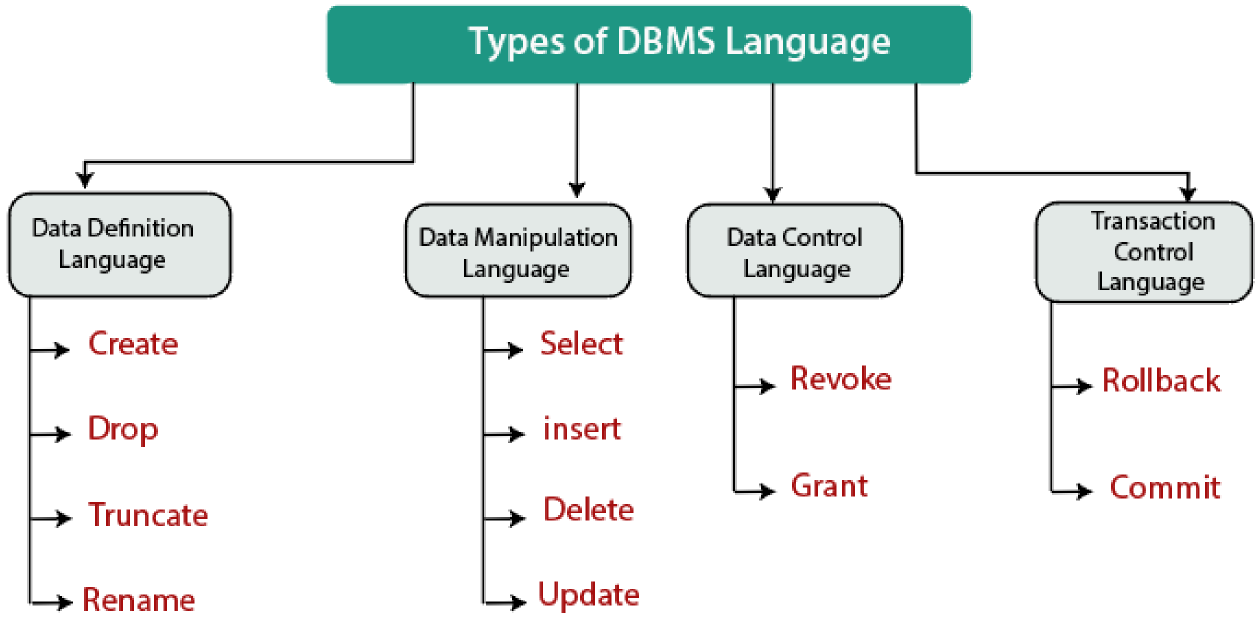


Fig: OO Data Model

Database Languages



- **Definition Language (DDL)**
  - Data definition language used to specify database scheme.
  - For example, following DDL statement in SQL defines account relation.

```
            create table account
     (
           id integer,  balance
           integer
                         )
```

    - The execution of above DDL statement creates table account.
  - The **output of the DDL is placed in the data dictionary,** which contains metadata i.e. , data about data.
  - The data dictionary is considered to be a **special type of table that can only be accessed and updated by the database system itself** (not a regular user).The database system consults the data dictionary before reading or modifying actual data.
  - DDL also **allows defining storage structure and access methods** for database system, such special set of DDL statement called **data storage and definition language.**
  - The data values stored in the database must satisfy certain **constraints, i.e. Integrity Constraints** (types: domain constraints, referential integrity, Assertions, authentication)
  - Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency.
    - For example, Suppose a Bank requires that any account **must not have negative balance**. The DDL provides facilities to specify such constraints.

- **Data Manipulation language (DML)**
  - Data manipulation language allow database user to **access (query) and manipulate data.**
  - DML allows **communication between user and database.**
  - DML allows to:
    - **Insert** new information in the database
    - **Read** information from the database
    - **Update** information in the database
    - **Delete** information in the database
  - There are basically two types Procedural and Non Procedural DML.
  - **Procedural** DMLs require a user to specify **what** data are needed and **how** to get those data.
  - **Declarative** DMLs (nonprocedural DMLs) require a user to specify **what** data are needed without specifying how to get those data.
  - Using Declarative DML a user does **not** have to specify **how** to get the data, But the database system has to deal with accessing data in an efficient way.
  - A **query** is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a **query language**.

- **Data Control language (DCL):**
  - DCL is used to access the stored data.
  - Mostly used for **revoking** and **granting** user access on a database.
  - DCL Commands:
    - **Grant**: This command allows user's access privileges to the database.
    - **Revoke**: This command removes the accessibility of users from the database objects

- **Transaction Control language (TCL):**
  - TCL manages transactions within the database.
  - Used to **execute the changes made by the DML statements**.
  - TCL Commands:
    - **Commit**: used to save the transactions in the database.
    - **Rollback**: used to restore the database to that state which was last committed.

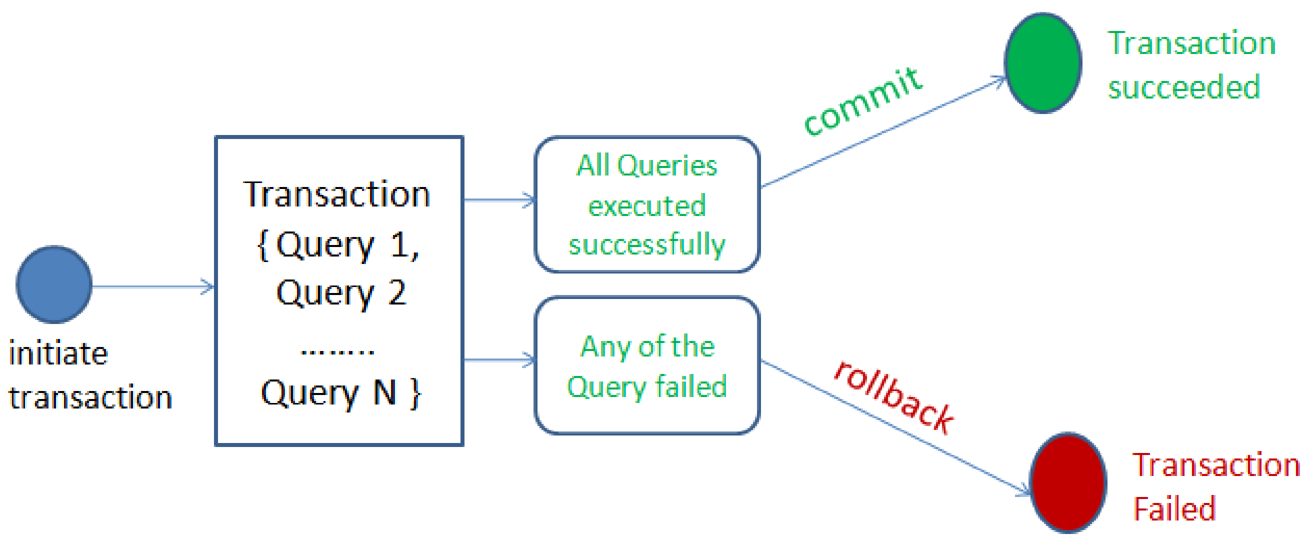
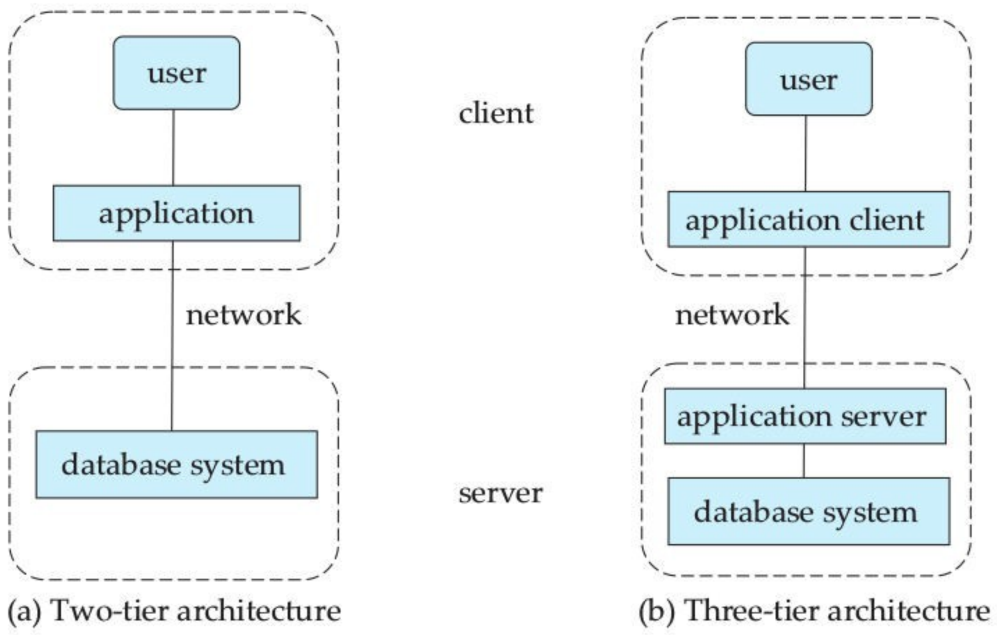


Figure: Example of transaction management using commit and rollback

DBMS Utilities:

1. Load Data
   - Load existing data files
2. Backup Data
   - Creates a backup copy of the database
3. Storage reorganization
   - Reorganize database files into different file organizations
4. Performance Monitoring
   - Monitors database usage and avail statistics.

Database Architecture



- It can be **centralized**, **decentralized** or **hierarchical**.
- The **architecture** of a DBMS can be seen as either **single-**tier or **multi-tier**.
- **1-tier architecture:**
  - the DBMS is the only entity where the user directly sits on the DBMS and uses it.
  - Any changes done here will directly be done on the DBMS itself.
  - It does not provide handy tools for end-users.
  - Database designers and programmers normally prefer to use single- tier architecture.
- **2 tier Architecture:**
  - it must have an application through which the DBMS can be accessed.
  - the application resides at the client machine, where it invokes database system functionality at the server machine through query language statements.
  - Application program interface standards like ODBC and JDBC are used for interaction between the client and the server.
- **3 tier Architecture:**
  - the client machine acts as merely a front end and does not contain any direct database calls.
  - Instead, the client end communicates with an application server, usually through a forms interface.
  - The application server in turn communicates with a database system to access data.
  - The business logic of the application, which says what actions to carry out under what conditions, is embedded in the application server, instead of being distributed across multiple clients.
  - Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.

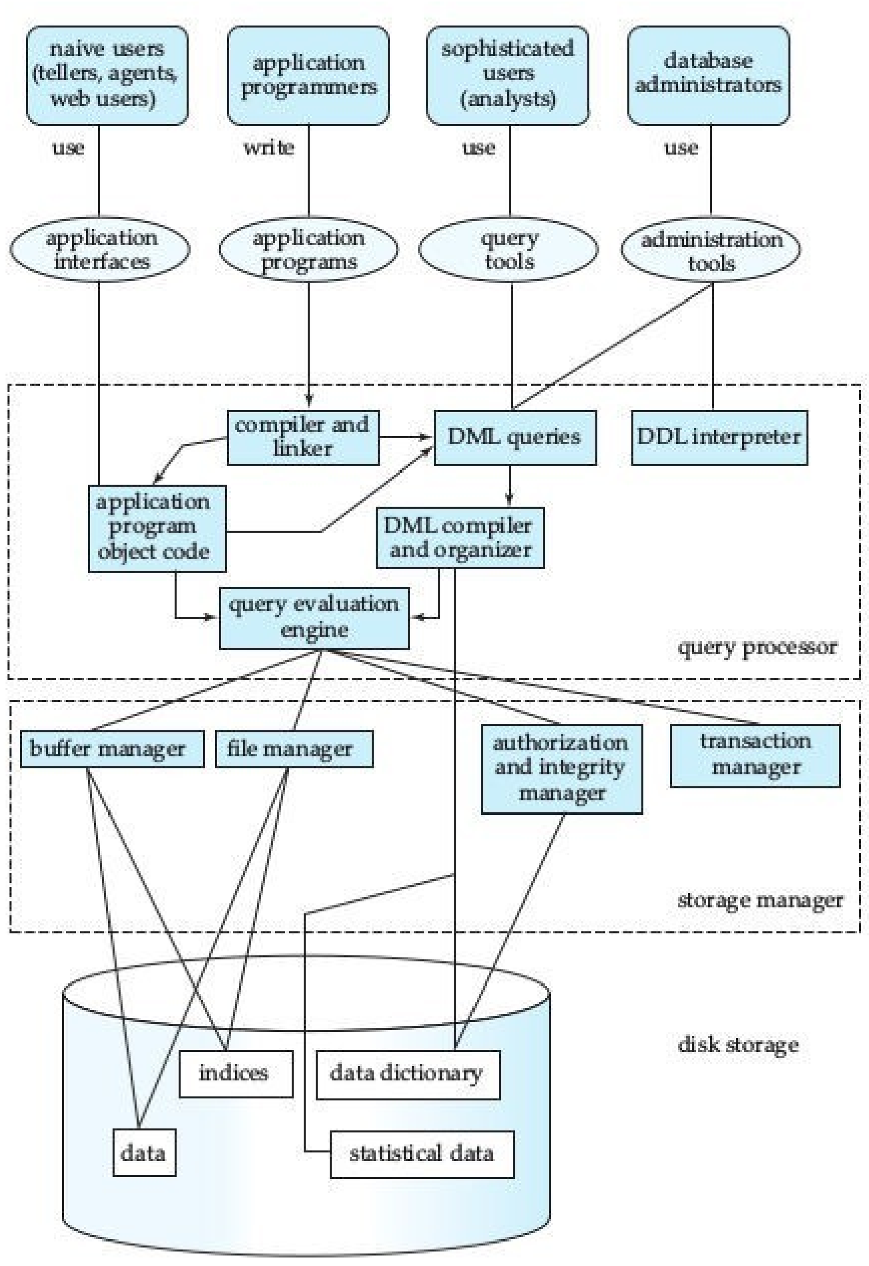


Fig: System Structure (Components of a Database System)

Overall System Structure

The functional component of the database system is divided into storage manager and query processor component.

- **Storage Manager:**
  - Storage manager is a program module that provides **interface** between the **low level data stored in the database and the application programs and queries submitted to the system.**
  - The storage manager is responsible for:
    - **interaction with the file manager.**
    - **storing, retrieving, and updating data in the database.**
  - The storage manager uses various DML statements into low level file system command.
  - Storage manager consist following **components**:
    - **Authorization and integrity manager:**
      - ensure **integrity constraint** does not violate
      - checks **the data access authority** of users to access data.
    - Transaction Manager:
      - ensure database remain in a **consistent state even system failure** occurs.
      - manage **concurrent transactions** so that they could not conflict, which also helps to ensure consistency of database.
    - File Manager:
      - manage the **allocation of space** on disk storage and the **data structures used** to represent information stored on disk.
    - Buffer Manager:
      - **fetching data** from disk storage into main memory,
      - decides what **data to cache** in main memory.

    - The storage manager implements several data structure for physical system implementation:
      - Data files: stores database itself,
      - Data dictionary: stores **meta data** about structure of database, in particular schema of database.
      - Indices: provides **fast access** to data items that holds particular values.

- **Query Processor:**
  - The query processor is responsible to **simplify and facilitate access data.**
  - It is responsible to **translate updates and queries** written in nonprocedural language **at the logical level**, **into an efficient sequence of operations at the physical level.**

- The **query processor component** includes the following components:
  - **DDL interpreter:**
    - **interprets DDL statements and records the definitions in the data dictionary.**
  - **DML Compiler:**
    - **translate DML statements** in a query language **into low level instructions that query evaluation engine understands.**
- The steps of query processing are:
  - **Query Parsing and translation:**
    Query is generally translated into **number of alternative evaluation plan**s that produce the same result.
  - **Query optimization:**
    Select the lowest cost evaluation plan among the alternatives
  - **Query evaluation:**
    execute low level instruction generated by DML compiler.

- The first step that must be taken in query processing is to translate a given query into its internal form. In generating the internal form of the query, the parser checks the syntax of the users query, verifies that the relation names appearing in the query are names of the relations in the database. The system constructs the parse tree and then generates the relational algebra expression. The given query can be computed in various ways.

- For example:

  - **Select balance from account Where balance<4000**

  - This query can be translated into either of the following relational algebra expressions.
    - ${}_{balance}(\quad {}_{balance<4000}(account))$
    - ${}_{balance<4000}(\quad {}_{balance}(account))$

  - The same query can be executed with different algorithms. The sequence of primitive operations that can be used to evaluate query is a query execution or query evaluation plan. Optimizer evaluates all the query plan and then finds out the cost of each query evaluation plan and selects the appropriate plan. Finally evaluation engine executes the plan and gives the output. The figure below shows the query processing steps.

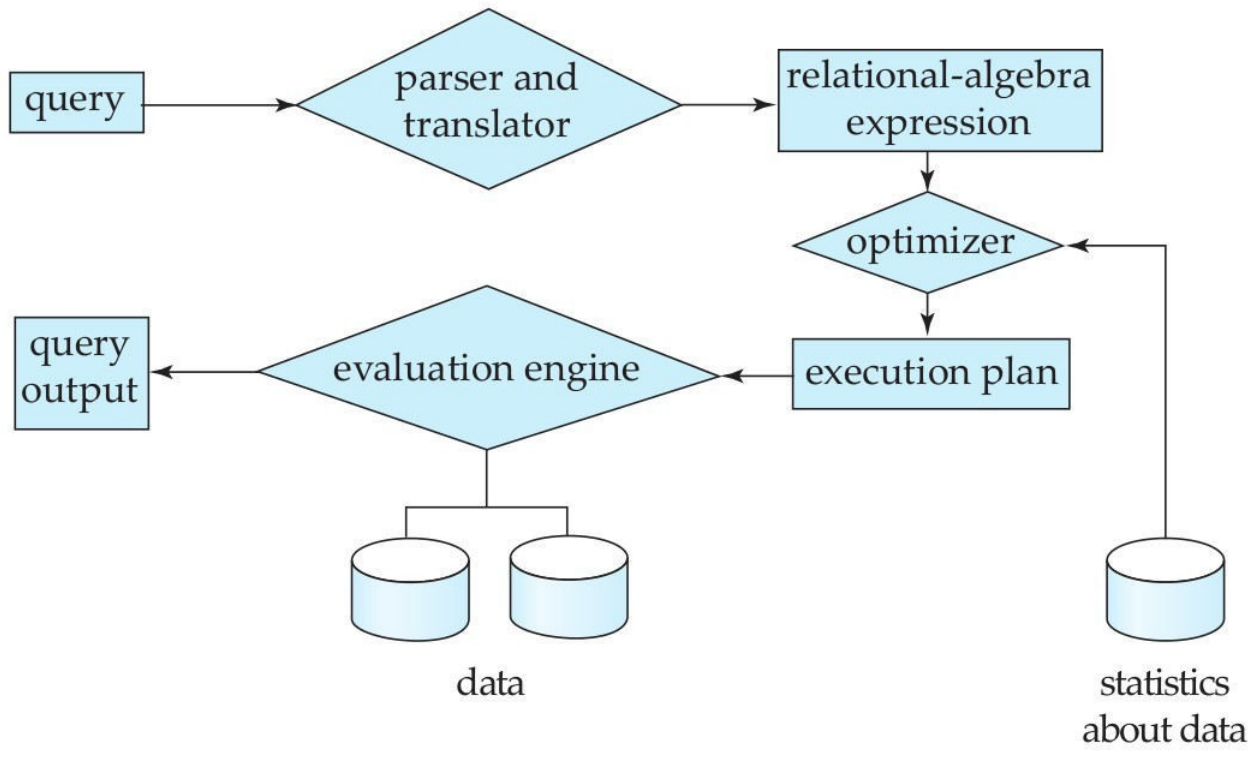


Fig: Steps in Query Processing

Database Users:

The various users of the database are discussed below:

1. **Naïve users:**

Naïve users are the simple users who just uses the application that have been built previously. For example, a customer who uses ATM simply invokes the program which checks his username, password and balance: and finally allows to withdraw certain amount from his account.

2. **Application programmer:**

These are the computer professionals who write application programs. They build user interfaces to interact with the database and hence it makes naïve users easy. Application programmers uses many application development tools for the quick development of the application.

3. **Sophisticated users:**

Sophisticated users do not interact with the database through the application programmers since the interest of such users in vast and hence they generate their own query in a database query language. Analyst who submit queries to explore data in the database fall in this category.

4. **Specialized users:**
Specialized users are the sophisticated users. These are highly advanced users and write specialized database applications. Scientists, researchers fall in this category.

Database Administrators:

The person who has control over both data and the program that accesses those data are called database administrator(DBA). The functions of database administrator are:

- **Schema definition:**
  The DBA creates the original structure of the database by using DDL.

- **Schema and physical organization modification:**
  The changes needed in any organization is analyzed and then appropriate change is made in the database schema by the DBA.

- **Granting of authorization:**
  By granting different types of the authorization, DBA can regulate which part of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.

- **Runtime maintenance:**
  DBA periodically backup the database, ensures that enough free disk space is available for normal operations.

References:

The Instructor do not own the contents (text and images) of this documents collected from various sources:

- A. Silberschatz, H.F. Korth, and S. Sudarshan, Database System Concepts, 4th Edition, McGraw Hill (ISBN: 0-07-120413)
- C. J. Date, An Introduction to Database Systems, 8 th Edition, Addison Wesley
- Raghu Ramakrishnan, and Johannes Gehrke, Database Management Systems, McGraw-Hill, 2003. (ISBN: 0-07-246563-8)
- Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 4th Edition, Pearson Addison Wesley; 2003, (ISBN: 0321122267)
- http://www.yourarticlelibrary.com/accounting/computerized-accounting/database- management-system-dbms-applications-uses-and-other-details/63278/
- https://opentextbc.ca/dbdesign/chapter/chapter-3-characteristics-and-benefits-of-a- database/
- http://whatisdbms.com/characteristics-of-database-approach/
- https://securosis.com/blog/understanding-and-selecting-data-masking-how-it-works
- http://www.careerbless.com/db/rdbms/c1/intdbms.php
- https://www.tutorialcup.com/dbms/database-abstraction.htm

- http://beginnersbook.com/2015/04/levels-of-abstraction-in-dbms/
- http://dcm.uhcl.edu/yue/courses/csci5333/Fall2015/notes/intro/IntroArch.html
- https://www.twinkl.de/teaching-wiki/data
- https://www.javatpoint.com/dbms-language
- https://pynative.com/python-mysql-transaction-management-using-commit-rollback/
- https://www.tutorialandexample.com/dbms-languages/
- The images used in this documents were obtained via Google Search.