Chapter-1 (Introduction)

① Data, Database, DBMS, Types of database
② Database system application
③ purpose of database system
④ Database users
⑤ Data instances and schema
⑥ Data Model, Relational database
⑦ Database languages - DDL, DML, DCL, SQL
⑧ Database Administrator
⑨ Database Abstraction
⑩ Database architecture
⑪ File system Vs database system
⑫ Database design, engine, sub-systems
⑬ Storage manager, Query processor, Transaction management.

Database management system: A database management system (DBMS) is a collection of data and a set of programs to access the data. The collection of data usually referred to as the database and contains information.

Database systems are designed to manage large bodies of information. Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data.

Data: Data is a collection of information gathered by observations, measurements and research or analysis. Data is the raw facts that forms information. There are two types of data. 1. Qualitative
2. Quantitative

Database: An organized collection of data/ records.

Database - system application:

1. Enterprise information
   - Sales : customers, products, purchases,
   - Accounting : payments, receipts, assets.
   - Human resources: Information about employees, salaries, payroll taxes.

2. Manufacturing: Management of production
   orders,
   supply chain
   inventory.

3. Banking and finance:
   - Customer information, accounts, loans, banking transactions.
   - Credit card transactions
   - Finance: Sales and purchases of financial instruments and bonds

4. Universities: registration, grades

5. Airlines: reservations, schedules

6. Telecommunication: records of calls, text and data usage, generating monthly bills.

7. Web based service: online retailers
   online advertisements

8. Document databases

9. Navigation system: For maintaining the locations of varies places, train systems, bus etc.

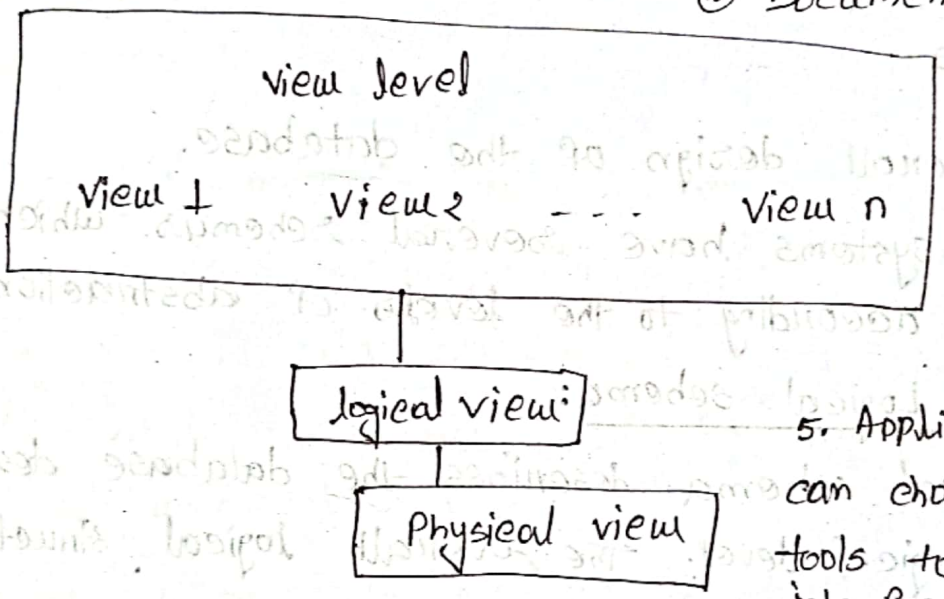
Purpose of database system:

1. Difficulty in accessing data
2. Data isolation
3. Atomicity problem
4. current access anomalies.
5. security problems

Types of database:

① centralized database
② Distributed database
③ cloud database
④ commercial database
⑤ object oriented database
⑥ Temporal database
⑦ spatial database
⑧ Network database
⑨ Graph database
⑩ Document Database

View of Data (Database user):



1. Naive users
2. Application programmers : are computer professionals who write application programs. It can choose from a many tools to develop user interfaces. Rapid application tools are tools that enable an application programer.
3. Sophisticated users interact with the system without writing programs.
4. Specialized users: This user will be using the database for a particular application not for a generic application.
5. Application programmers can choose from many tools to develop user interfaces.

Data instances and schemas:

Databases changes 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.

The overall design of the database is called the database schema. Schemas are changed infrequently.

- Similar to types and variables in programming languages

Schema: Overall design of the database.

Database systems have several schemas. which is partitioned according to the levels of abstraction.

Types: 1. **Logical schema:**

→ Logical schema describes the database design at the logical level. The overall logical structure of the databases.

Exp: The database consists of information about a set of customers and accounts in a bank and relationship between them.

2. **physical schema:** The overall physical structure of the database. Physical schema describes the database design at the physical level.

3. Sub schema: A database may also have several schemas at the view level. That describe different views of the database.

Physical data independence: The ability to modify the physical schema without changing the logical schema.

⇒ Applications depend on the logical schema.

⇒ Application programms are said to be exhibit physical data independence if they do not depend on the physical schema, and thus need not be rewritten if the physical schema changes.

Data Models: Underlying the structure of a database is the data model, a collection of conceptual tools for describing data, data relationships.

A data model provides a way to describe the design of a database at the physical, logical and view models

There are 4 types of data models:

1. Relational model: Relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each columns has a unique name.

Tables are also known as relations. The relational model is an example of a record-based model.

The relational data model is the most widely data model.

2. Entity Relationship model: The entity-relationship (E-R) data model uses a collection of basic objects called entities and relationship among these objects. An entity is a thing or object in the real world. The E-R model is widely used in database design.

3. Object Based data model: object oriented programming (especially in Java, C++ or C#) has become the dominant software-development methodology. The object relational data model combines features of the object-oriented data model and relational data model.

4. Semistructured model: The semistructured model permits the specification of data where individual data items of the same type may have different sets of attributes. The extensible markup language (XML) is widely used to represent semi-structured data.

Example of XML:

```
<department>
    <department>
        <name> CSE </name>
        <semester> 8 </semester>
```

```
    <building> CSE </building>
</department>
<department>
    <name> EEE </name>
    <semester> 8 </semester>
    <building> EEE </building>
</department>
<course>
    <name> CSE 2423 </name>
    <room> 101 </room>
    <lab> db-lab </lab>
</course>
</department>
```

Types of Relational model:

1. one to one
2. one-to many
3. Many to many

Relational model:

columns

ID	name	dept_name	salary
22	Wu	Finance	95000
23	Brick	Biology	60000
24	Katz	History	75000

— Rows

(Instructor table)

dept_name	building	budget
Biology	Taylor	10000
Music	Watson	50000

department table

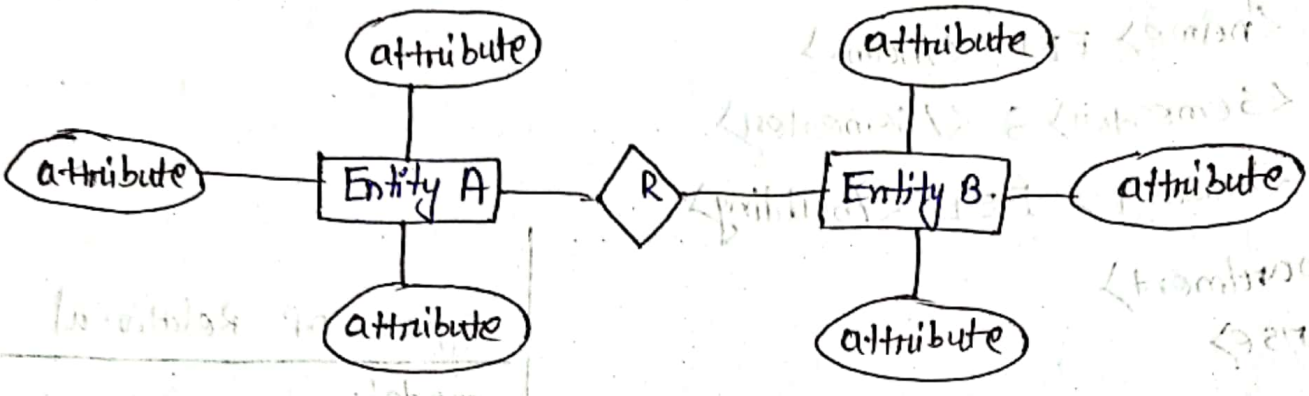
→ All the data is stored in various tables.

→ Tables are known as relations

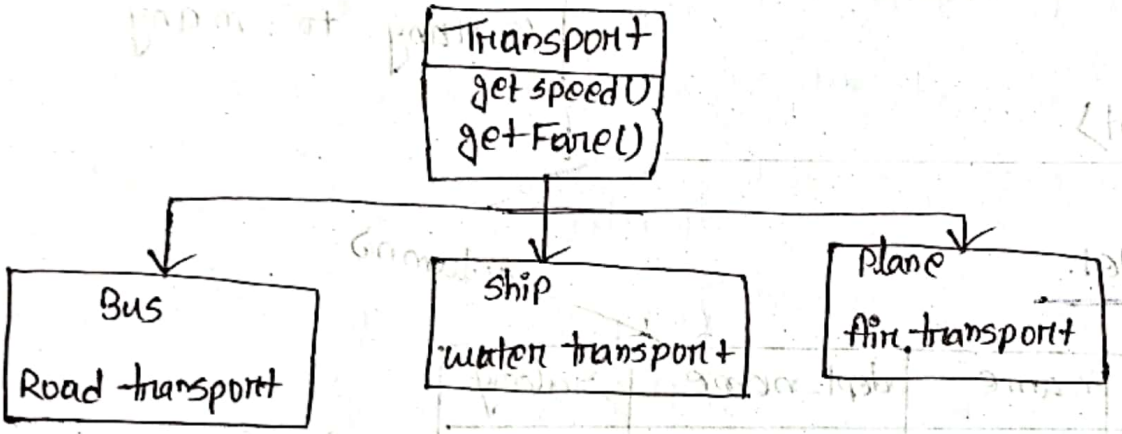
Relational database: It is based on the relational model and uses a collection of tables to represent both data and relationships.

→ It includes DML and DDL.

Entity relationship model: ⇒ collection of basic object called entities and relationships among these objects.



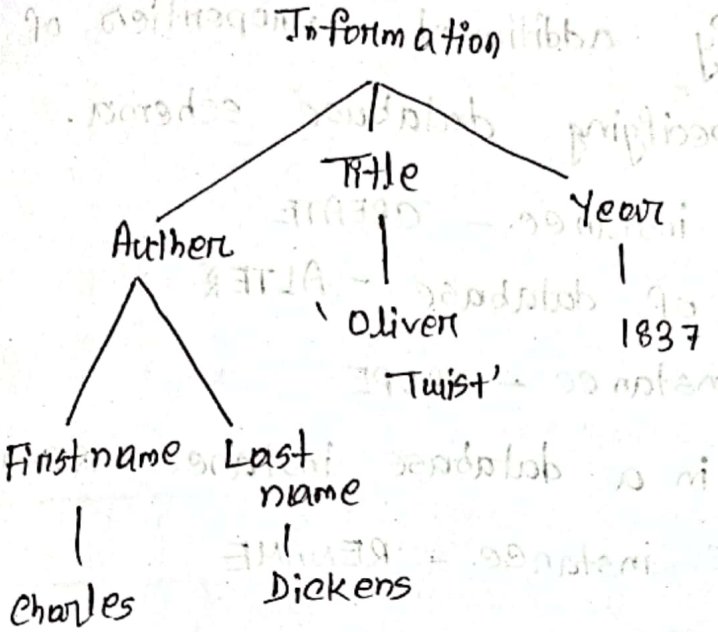
Object based data model:



Here,
Object - Transport, Bus, ship and plane
Bus has road transport as the attribute
ship has water transport as the attribute
plane has Air transport as the attribute

- Here, the transport object is the base object and the Bus, ship and plane objects derive from it.

Semi Structured data model:



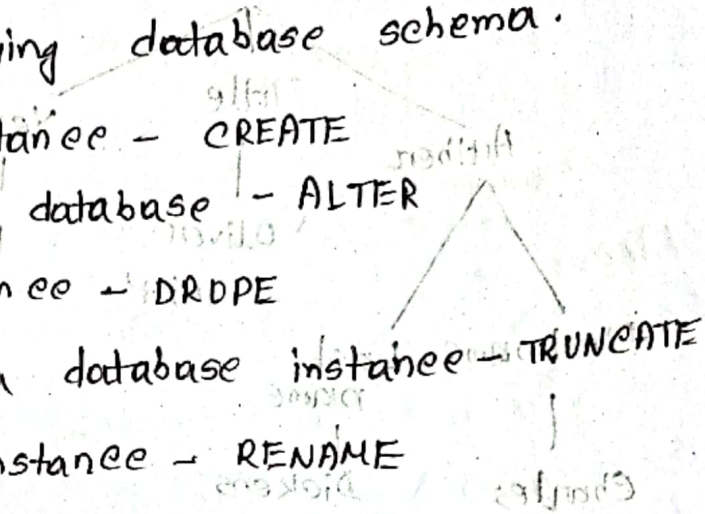
Database languages: A database system provides a data-definition language to specify the database schema and a data manipulation language to express database queries and updates. The data-definition and data-manipulation languages are not two separate languages instead they simply form parts of a single database such as they are widely used SQL language.

- Data definition language (DDL)
- Data Manipulation language (DML)
- Data control language (DCL)

Data definition language (DDL):

DDL is used to specify additional properties of data. DDL used for specifying database schema.

1. To create database instance - CREATE
2. To alter structure of database - ALTER
3. To drop database instance - DROPE
4. To delete table(s) in a database instance - TRUNCATE
5. To rename database instance - RENAME



- SQL provides a rich DDL that allows one to define tables, integrity, constraints, assertions etc.

Ex: create table department
(dept_name char(20),
building char(15),
budget numeric(12,2));

- create table instructor(
ID char(5),
name varchar(20),
dept_name varchar(20),
salary numeric(8,2))

DDL compiler generates a set of table templates stored in a data dictionary

DDL gets as input some instructions (statements) and generates some output.

The output of the DDL is placed in the data dictionary, which contains metadata that is data about data.

Database constraints :

1. Domain constraints : A domain of possible values must be associated with every attribute (ex: integer types, character types, date/time types)

2. Referential Integrity : Value that appears in one relation for a given set of attributes also appears in a certain set of attributes in another ~~section~~ relation.

   Ex: The dept_name value in a course record must appear in the dept_name attribute of some record of the department relation.

3. Assertions : ⇒ It is any condition that the database must always satisfy.

   ⇒ Domain constraints and referential integrity constraints are special forms of assertions.

4. Authorization : Differentiations in the type of access they are permitted on various data values in the database is called authorization.

5. ⊞ Read authorization: which allows reading but not modification.
   ⊞ Insert authorization: which allows insertion of new data but not modification.
   ⊞ update authorization: which allows modification but not deletion of data.
   ⊞ Delete authorization: which allows of data

Special type of DDL:

Data storage and definition language:

The storage structure and access methods used by the database system are specified by a set of statements in a special type of DDL called a data storage and definition language.

Data manipulation Language (DML)

DML is used for accessing and manipulating data in a database. It is known as query language. There are two types of DML. They are:

1. Procedural DML - Require a user to specify what data are needed how to get those data.
2. Declarative DML - Require a user to specify what data are needed without specifying how to get those data.

- Declarative DMLs are usually easier to learn and use than are procedural DMLs.
- Declarative DMLs are also referred to as non-procedural DMLs.
- The portion of a DML that involves information retrieval is called a query language.
- SQL query language is nonprocedural.
- SQL query that finds the names of all instructors in the history department.

```
select instructor name
from instructor
where instructor. dept_name = 'History';
```

1. To read records from the table(s) – SELECT
2. To insert records into the table(s) – INSERT
3. Update the data in table(s) – UPDATE
4. Delete (all) records from the table(s) – DELETE

Data control language (DCL)

DCL is used for granting and revoking user access on database 1. To grant access to user – GRANT
2. To revoke access from user – REVOKE

(SQL) **SQL Query language:** It is a non procedural. A query takes as input several tables (possible only one) and returns a single table. It is not a turning machine equivalent language.

```
select name
from instructor
where dept_name = 'Comp. Sci.'
```

Database Administrator: A person who has central control over the system is called a database administrator (DBA). One of the main reason for using DBMS is to have central control of both the data and the programs that access those data.

The functions of a DBA include:

① Schema definition: The DBA creates the original database schema by executing a set of data definition statements in the DDL.

② Storage structure and access method definition

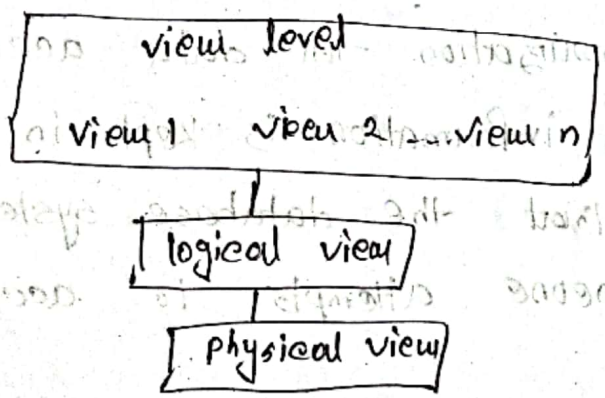
③ Schema and ~~of~~ physical organization modification:
The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization.

④ Granting of authorization for data access:
The authorization information is kept in a special system structure that the database system consult. whenever someone attempts to access the data in the system.

⑤ Routine maintenance:
- periodically backing up the database,
- ensuring that enough free disk space is available for normal operations,
- Monitoring jobs running on the database.

Data Abstraction: For the system to be usuable, it must be retrieve data efficiently. They need for efficiency has led designers to use complex data structures to represent data in the database.

- **Physical Level :** This level describes low-level data structures in detail. (highest level / lowest)
- **Logical level:** This level describes the entire database in terms of a small number of simple structure. This level describes what data are stored in the database and what relationships exist among these data.
- **View level:** The highest level of abstraction describes only part of the entire database.

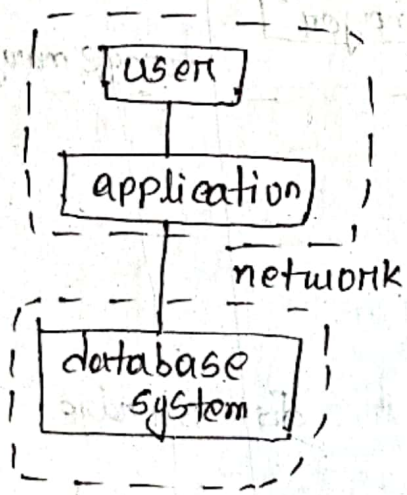


Database Architecture: The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs. It can be,

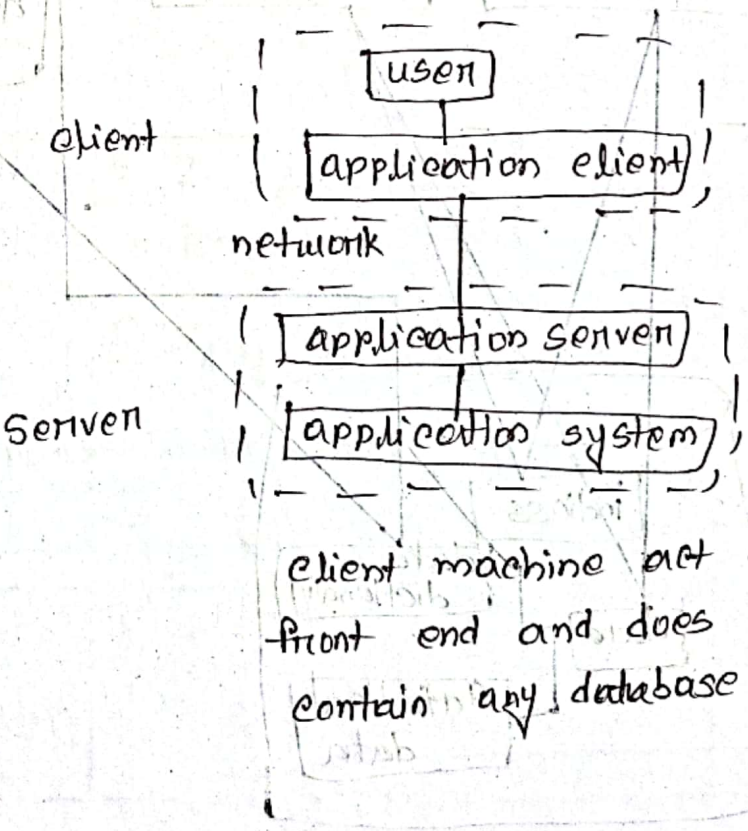
1. centralized (one to a few cores, shared memory)
2. client ~~server~~ server
3. parallel databases (many more shared memory, shared disk, shared nothing)
4. Distributed databases (Geographical distribution) (schema / data heterogeneity)

Database application: There are two or three parts.

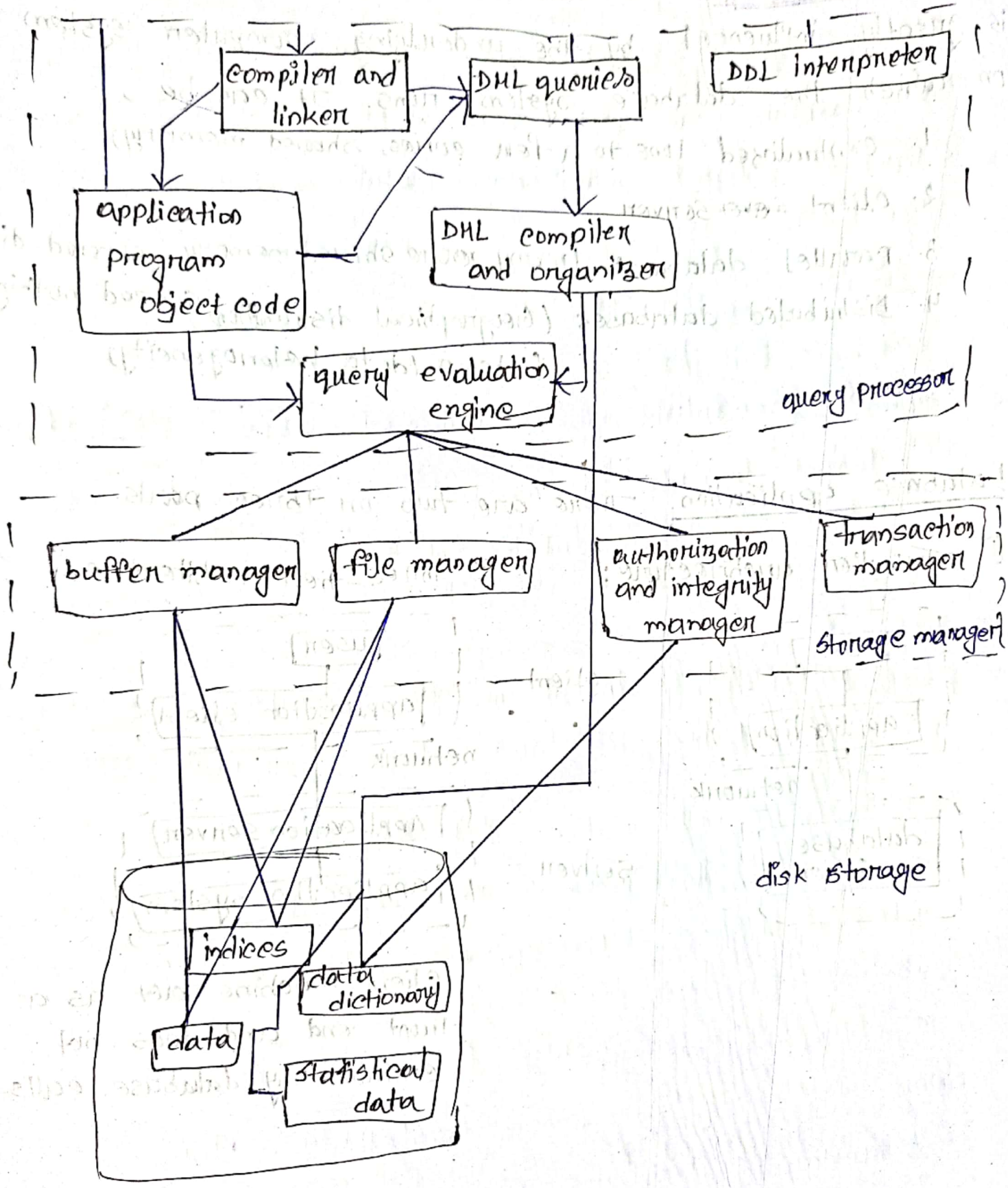
① Two-tier architecture:



② Three-tier architecture:



client machine act as a front end and does not contain any database calls.



Database architecture

File system Vs database system

File system	Database system
① It maintains data redundancy and inconsistency.	① It maintains data consistency
② Data isolation	② No problem in data isolation
③ Difficulties in data access	③ Easy data access
④ Atomicity problem	④ Allowing atomicity
⑤ security problem	⑤ No security problem

Database design: A database system provides a data definition language to specify the database schema and a data manipulation language to express database. Two types: ① logical design
② physical design

Logical design: Deciding on the database schema.

Physical design: Deciding on the physical layout of the database.

Database engine: The functional components of a database system can be divided into

⇒ Storage manager

⇒ Query processor component

⇒ Transaction management component

Database sub-systems: A database system has several subsystem

⇒ Storage manager - It provides the interface between the low level data stored in the database and the application programs

and queries submitted to the system.

⇒ Query language- It compiles and executes DDL and DML.

Storage Manager: A program that module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

⇒ The storage manager translates the various DML statements into low-level file system commands

⇒ As databases typically require a large amount of storage space.

- Storage manager is responsible to following the tasks.
  - ⇒ Interaction with the OS file manager.
  - ⇒ Efficient storing, retrieving and updating of data.
- Storage manager components include
  1. Authorization and integrity manager
  2. Transaction manager
  3. File manager
  4. Buffer manager.

⊞ The storage manager implements several data structures as part of the physical system implementation:

① Data files: Store the database itself

② Data dictionary: Store metadata about the structure of database.

③ Indices - can provide fast access to data items.

Query processor: It helps the database system to simplify and facilitate access to data.

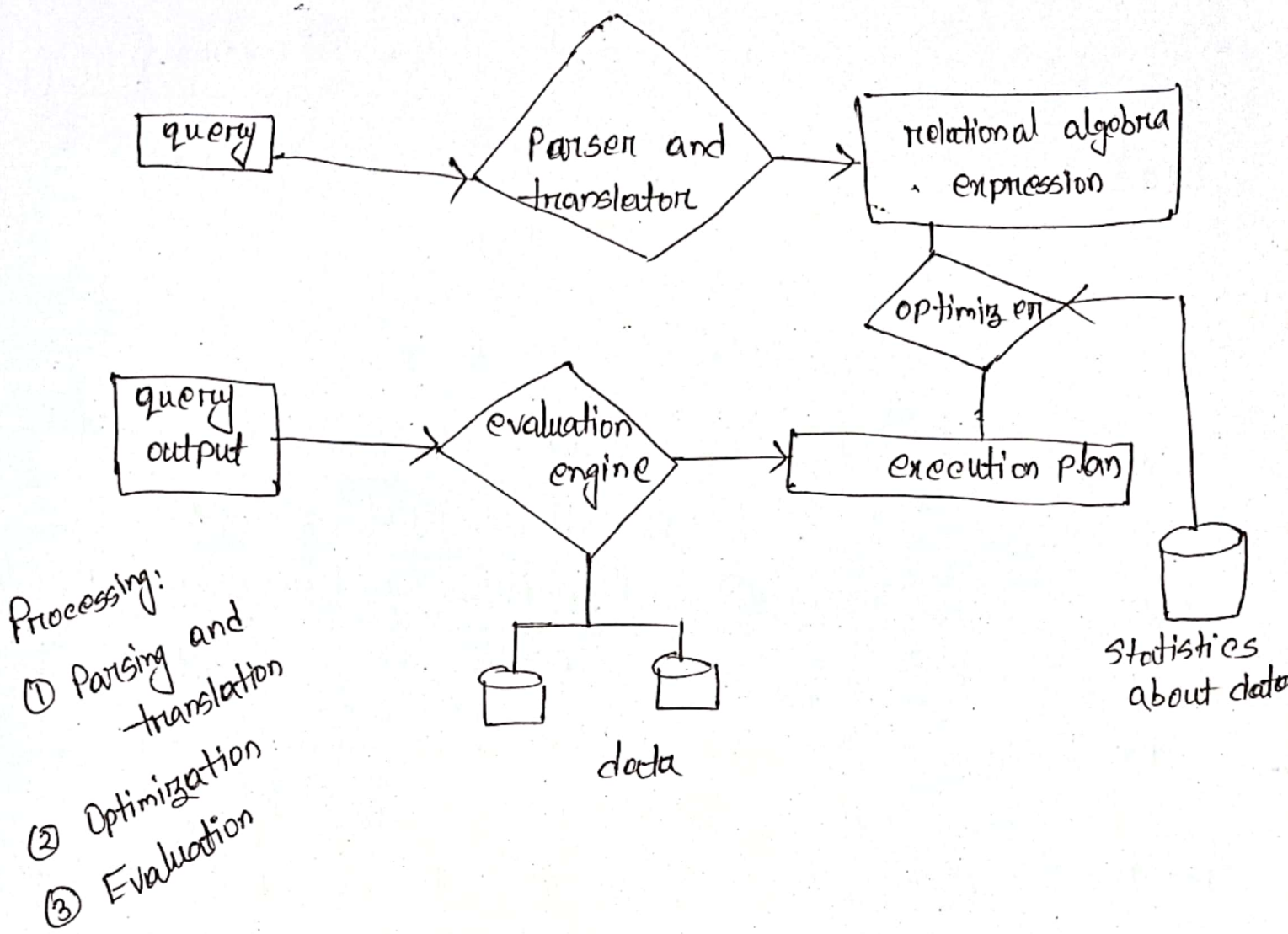
It allows database users to obtain good performance.

Query processor components:

① DDL interpreter: • Interprets DDL statements
• Records the definitions.

② DML compiler: Translate DML statement in a query language.

③ Query evaluation engine: executes low-level instructions generated by the DML compiler.



Processing:

① Parsing and translation

② Optimization

③ Evaluation

Transaction management: It is a collection of operations that performs a single logical function in a database application.

⇒ Transaction management component

⇒ Concurrency - control manager