Define the following terms.

Database

- A Database is a collection of inter-related data.

DBMS (Database Management System)

- A database management system is a collection of inter-related data and set of programs to manipulate those data.
- DBMS = Database + Set of programs

Database instance:

- The collection of information stored in the database at a particular moment is called an instance of the database.

Database Schema:

- The overall design of the database is called the database schema.
- Based on the levels of abstraction: The **Physical schema** describes the database design at the physical level, while the **Logical schema** describes the database design at the logical level.

Metadata

- Metadata is data about data.
- Data such as table name, column name, data type, authorized user, user access privileges for any table is called metadata for that table.

Data dictionary

- Data dictionary is an information repository which contains metadata.
- It is usually a part of the system catalog.

Data warehouse

- Data warehouse is an information repository which stored data.
- It is design to facilitate reporting and analysis.

Field

- A field is a character or group of characters that have a specific meaning.
- It is also called a data item. It is represented in the database by a value.
- For Example customer id, name, society and city are all fields for customer Data.

Record

- A record is a collection of logically related fields.
- For examples, collection of fields (id, name, society & city) forms a record for customer.

Query:

- A query is statement requesting the retrieval of information from the database.
- The portion of Data manipulation language that involves information retrieval is called **query language.**

Transaction:

- A transaction is a collection of operations that perform a single logical function in a database system.
- Transaction accesses and possibly updates various data items.

Explain disadvantages of file system (file processing systems) compare to Database management system. OR Explain disadvantages of conventional file-based system compared to Database management system.

Data Redundancy

- It is possible that the same information may be duplicated in different files. This leads to data redundancy.
- Data redundancy results in memory wastage.
- For example, consider that some customers have both kinds of accounts - saving and current. In this case, data about customers - name, address, e-mail, contact number - will be duplicated in both files, file for saving accounts and file for current accounts. This leads to requirement of higher storage space. In other words, same information will be stored in two different locations (files). And, it wastes memory.

Data Inconsistency

- Due to data redundancy, it is possible that data may not be in consistent state.
- For example, consider that an address of some customer changes. And, that customer has both kinds of accounts. Now, it is possible that this changed address is updated in only one file, leaving address in other file as it is. As a result of this, same customer will have two different addresses in two different files, making data inconsistent.

Difficulty in Accessing Data

- Accessing data is not convenient and efficient in file processing system.
- For example, suppose, there is a program to find information about all customers. But, what if there is a need to find out all customers from some particular city. In this case, there are two choices here: One, find out all customers using available program, and then extract the needed customers manually. Second, develop new program to get required information. Both options are not satisfactory.
- For each and every different kind of data access, separate programs are required. This is neither convenient nor efficient.

Limited Data Sharing

- Data are scattered in various files.
- Different files may have different formats. And these files may be stored in different folders (directories) may be of different computers of different departments.
- So, due to this data isolation, it is difficult to share data among different applications.

Integrity Problems

- Data integrity means that the data contained in the database is both correct and consistent. For this purpose, the data stored in database must satisfy certain types of constraints (rules).
- For example, a balance for any account must not be less than zero. Such constraints are enforced in the system by adding appropriate code in application programs. But, when new constraints are added, such as balance should not be less than Rs. 5000, application programs need to be changed. But, it is not an easy task to change programs whenever required.

Atomicity Problems

- Any operation on database must be atomic. This means, operation completes either 100% or 0%.
- For example, a fund transfer from one account to another must happen in its entirely. But, computer systems are vulnerable to failure, such as system crash, virus attack. If a system failure occurs during the execution of fund transfer operation, it may possible that amount to be transferred, say, Rs. 500, is debited from one account, but is not credited to another

account.

- This leaves database in inconsistent state. But, it is difficult to ensure atomicity in a file processing system.

Concurrent Access Anomalies

- Multiple users are allowed to access data simultaneously (concurrently). This is for the sake of better performance and faster response.
- Consider an operation to debit (withdrawal) an account. The program reads the old balance, calculates the new balance, and writes new balance back to database. Suppose an account has a balance of Rs. 5000. Now, a concurrent withdrawal of Rs. 1000 and Rs. 2000 may leave the balance Rs. 4000 or Rs. 3000 depending upon their completion time rather than the correct value of Rs. 2000.
- Here, concurrent data access should be allowed under some supervision.
- But, due to lack of co-ordination among different application programs, this is not possible in file processing systems.

Security Problems

- Database should be accessible to users in a limited way.
- Each user should be allowed to access data concerning his application only.
- For example, a customer can check balance only for his/her own account. He/She should not have access to information about other accounts.
- But, in file processing system, application programs are added in an ad hoc manner by different programmers. So, it is difficult to enforce such kind of security constraints.

Explain advantages (benefits) of DBMS over file management system. OR Explain purpose of database system.

Minimal Data Redundancy (Duplication)

- Due to centralized database, it is possible to avoid unnecessary duplication of information.
- This leads to reduced data redundancy.
- It prevents memory wastage.
- It also reduced extra processing time to get required data.

Shared Data

- All authorized user and application program can share database easily.

Data Consistency

- Data inconsistency occurs due to data redundancy.
- With reduced data redundancy such type of data inconsistency can be eliminated.
- This results in improved data consistency.

Data Access

- DBMS utilizes a variety of techniques to retrieve data.
- Required data can be retrieved by providing appropriate query to the DBMS.
- Thus, data can be accessed in convenient and efficient manner.

Data Integrity

- Data in database must be correct and consistent.
- So, data stored in database must satisfy certain types of constraints (rules).
- DBMS provides different ways to implement such type of constraints (rules).
- This improves data integrity in a database.

Data Security

- Database should be accessible to user in a limited way.

- DBMS provides way to control the access to data for different user according to their requirement.
- It prevents unauthorized access to data.
- Thus, security can be improved.

Concurrent Access

- Multiple users are allowed to access data simultaneously.
- Concurrent access to centralized data can be allowed under some supervision.
- This result in better performance of system and faster response.

Guaranteed Atomicity

- Any operation on database must be atomic. This means, operation must be executed either 100% or 0%.
- This type of atomicity is guaranteed in DBMS.

List and explain the applications of DBMS.

Airlines and railways

- Airlines and railways use online databases for reservation, and for displaying the schedule information.

Banking

- Banks use databases for customer inquiry, accounts, loans, and other transactions.

Education

- Schools and colleges use databases for course registration, result, and other information.

Telecommunications

- Telecommunication departments use databases to store information about the communication network, telephone numbers, record of calls, for generating monthly bills, etc.

Credit card transactions

- Databases are used for keeping track of purchases on credit cards in order to generate monthly statements.

E-commerce

- Integration of heterogeneous information sources (for example, catalogs) for business activity such as online shopping, booking of holiday package, consulting a doctor, etc.

Health care information systems and electronic patient record

- Databases are used for maintaining the patient health care details in hospitals.

Digital libraries and digital publishing

- Databases are used for management and delivery of large bodies of textual and multimedia data.

Finance

- Databases are used for storing information such as sales, purchases of stocks and bonds or data useful for online trading.

Sales

- Databases are used to store product, customer and transaction details.

Human resources

- Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.

Describe functions (responsibility, roles, and duties) of DBA to handle DBMS.

DBA

- The full name of DBA is Database Administrator.
- Database Administrator is a person in the organization who controls the design and the use of database.

Functions or Responsibilities of DBA are as under:

Schema Definition

- DBA defines the logical schema of the database.
- A schema refers to the overall logical structure of the database.
- According to this schema, database will be designed to store required data for an organization.

Storage Structure and Access Method Definition

- DBA decides how the data is to be represented in the database.
- Based on this, storage structure of the database and access methods of data is defined.

Defining Security and Integrity Constraints

- DBA decides various security and integrity constraints.
- DDL provides facilities to specifying such constraints.

Granting of Authorization for Data Access

- The DBA determines which user needs access to which part of the database.
- According to this, various types of authorizations (permissions) are granted to different users.
- This is required to prevent unauthorized access of a database.

Assisting Application Programmers

- DBA provide assistance to application programmers to develop application programs.

Monitoring Performance

- The DBA monitors performance of the system.
- The DBA ensure that better performance is maintained by making change in physical or logical schema if required.

Backup and Recovery

- Database should not be lost or damaged.
- The task of DBA is to backing up the database on some storage devices such as DVD, CD or Magnetic Tape or remote servers.
- In case of failures, such as flood or virus attack, Database is recovered from this backup.

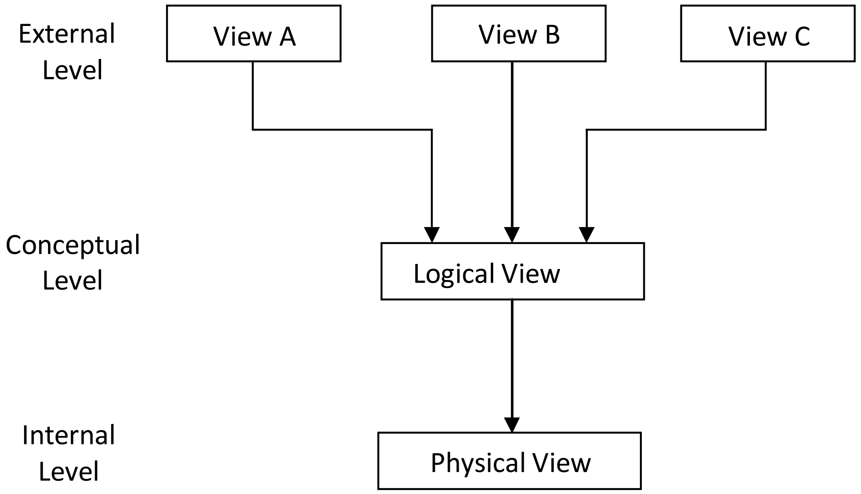
Explain three levels of Database System. OR Explain three level Data abstraction.

The ANSI SPARC architecture divided into three levels:

1) View level
2) Logical level
3) Physical level

Physical Level

- This is the lowest level of the data abstraction.
- It describes **how** the data are actually stored on storage devices.
- It is also known as a Internal level. The internal view is described by physical schema.
- Physical schema consists of definition of stored record, method of representing the data field and access method used.



Logical Level

- This is the next higher level of the data abstraction.
- It describes **what** data are stored in the database and what relationships exist among those data.
- It is also known as a logical level.
- Logical view is defined by Logical schema. It describes all records and relationship.

View Level

- This is the highest level of data abstraction.
- It is also known as External level.
- It describes only part of the entire database that a particular end user requires.
- External view is describes by external schema.
- External schema consists of definition of logical records, relationship in the external view and method of deriving the objects from the conceptual view.
- This object includes entities, attributes and relationship.

Explain Mapping. OR
Explain external and internal mapping. OR
What is mapping? Describe type of mapping.

Mapping

- The process of transforming requests and results between the three levels is called mapping.

Types of Mapping

- Logical (Conceptual) /Physical (Internal) Mapping
- View (External)/ Logical (Conceptual) Mapping

Logical (Conceptual) /Physical (Internal) Mapping

- It relates logical schema with physical schema.
- It defines correspondence between the logical schema and the database stored in physical devices.
- It specifies how logical records and fields are presented at the physical level.
- If the structure of stored database is changed, then logical / physical mapping must be changed accordingly and logical schema can remain invariant.
- There could be one mapping between logical and physical levels.

View (External)/ Logical (Conceptual) Mapping

- It relates each view schema with logical schema.
- It defines correspondence between a particular view and logical schema.
- If the structure of logical schema is changed, then view / logical mapping must be changed accordingly and view schema can remain invariant.
- There could be several mappings between view and logical levels.

Explain Data Independence.

Data Independence

- Data independency is the ability to modify a schema definition in one level without affecting a schema definition in the next higher level.

Types of data independence

- Physical data independence
- Logical data independence

Physical data independence

- Physical data independence allows changing in physical storage devices or organization of file without change in the logical view or logical view.
- Modifications at the physical level are occasionally necessary to improve performance.
- Physical data independence separates logical level from the physical level.
- It is easy to achieve physical data independence.

Logical data independence

- Logical data independence is the ability to modify the logical schema without requiring any change in application programs.
- Logical schema can be changed without affecting the existing external schema.
- Modifications at the logical level are necessary whenever the logical structure of the database is altered.
- Logical data independence separates external level from the logical view.
- It is difficult to achieve logical data independence.

Explain different database users.

There are four different database users.

Application programmers

- These users are computer professionals who write application programs using some tools.

Sophisticated users

- These users interact with system without writing program. They form their request in a database query language.

Specialized users

- These users write specialized database applications that do not fit into the traditional data processing framework.

Naive users

- These users are unsophisticated users who have very less knowledge of database system.
- These users interact with the system by using one of the application programs that have been written previously.
- Examples, people accessing database over the web, bank tellers, clerical staff etc.

Differentiate the DA and DBA.

DA (Data Administrator)	DBA (Database Administrator)
The data administrator is a person in the organization who controls the data of the database.	The database administrator is a person in the organization who controls the design and the use of the database.
DA determines what data to be stored in database based on requirements of the organization.	DBA provides necessary technical support for implementing a database.
DA is involved more in the requirements gathering, analysis, and design phases.	DBA is involved more in the design, development, testing and operational phases.
DA is a manager or some senior level person in an organization who understands organizational requirements with respect to data.	DBA is a technical person having knowledge of database technology.
DA does not need to be a technical person, but any kind of knowledge about database technology can be more beneficiary.	DBA does not need to be a business person, but any kind of knowledge about a functionality of an organization can be more beneficiary.
DA is a business focused person, but, he/she should understand more about the database technology.	DBA is a technically focused person, but, he/she should understand more about the business to administer the databases effectively.

What are the various Data Models of DBMS?

Data Model:

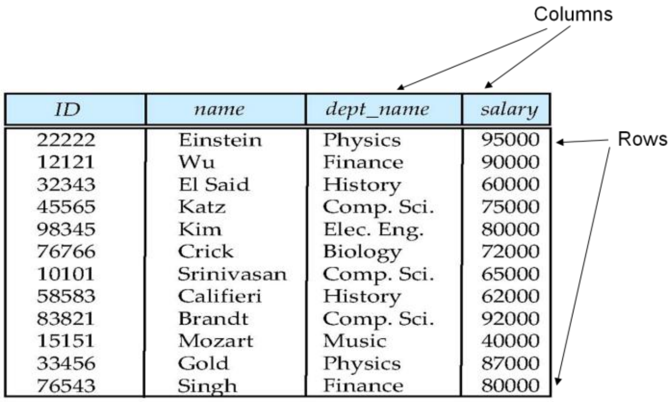
It is a collection of tools for describing Data, Data relationships, Data semantics and Data constraints.

Types of data models:

- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semistructured data model (XML)
- Other older models:
  - Network model
  - Hierarchical model

Relational Model:

- The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations**.**
- Each table contains records of a particular type. Each record type defines a fixed number of fields, or attributes.
- The columns of the table correspond to the attributes of the record type.



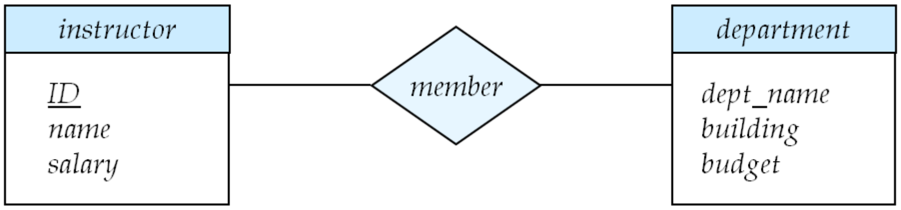
Columns
Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Entity-Relationship Model.

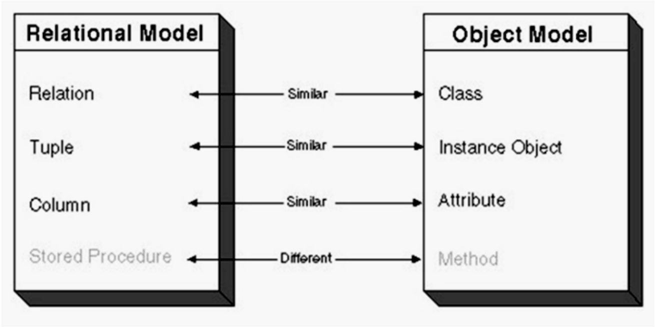
The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects. An entity is a “thing” or “object” in the real world that is distinguishable from other objects.



Object-Based Data Model

The object-relational data model combines features of the object-oriented data model and relational data model.

- Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
- Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
- Provide upward compatibility with existing relational languages.



Semi structured Data Model.

- The semi structured data 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

What are the types of 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.

Data-Manipulation Language

A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model. The types of access are:

- Retrieval of information stored in the database
- Insertion of new information into the database
- Deletion of information from the database
- Modification of information stored in the database

There are basically two types:

- Procedural DMLs require a user to specify what data are needed and how to get those data.
- Declarative DMLs (also referred to as nonprocedural DMLs) require a user to specify what data are needed without specifying how to get those data.

Data-Definition Language:

We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL).

Example:

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

The data values stored in the database must satisfy certain consistency constraints. For example, suppose the university requires that the account balance of a department must never be negative. The DDL provides facilities to specify such constraints.

- **Domain constraint**
  - A domain of possible values must be associated with every attribute : i.e. Data type
- **Integrity constraints**
  - Primary key and Foreign key relationships
- **Authorization**
  - Who can access what.
  - the most common being: read authorization, which allows reading, but not modification, of data;
  - insert authorization, which allows insertion of new data, but not modification of existing data;
  - update authorization, which allows modification, but not deletion, of data; and
  - delete authorization, which allows deletion of data.
- **Assertions**

  An assertion is any condition that the database must always satisfy. Domain constraints and referential-integrity constraints are special forms of assertions

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

Explain Database System Architecture.

Components of a DBMS

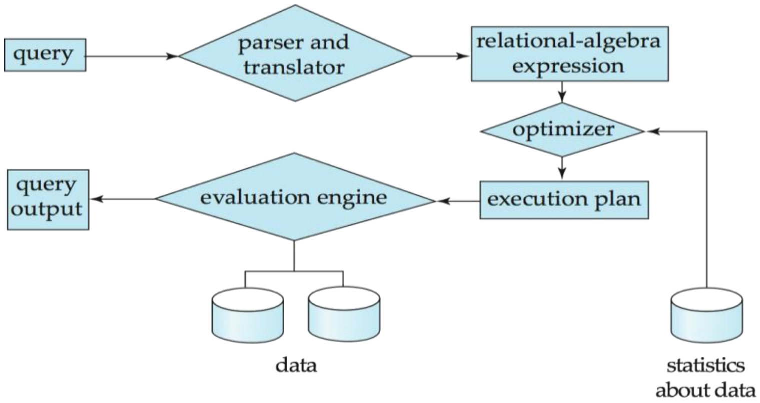
These functional units of a database system can be divided into two parts:

1. Query Processor Units (Components)
2. Storage Manager Units

Query Processor Units:

Query processor units deal with execution of DDL and DML statements.

- **DDL Interpreter —** Interprets DDL statements into a set of tables containing metadata.
- **DML Compiler —** Translates DML statements into low level instructions that the query evaluation engine understands.
- **Embedded DML Pre-compiler —** Converts DML statements embedded in an application program into normal procedure calls in the host language.
- **Query Evaluation Engine —** Executes low level instructions generated by DML compiler.



Storage Manager Units:

Storage manager units provide interface between the low level data stored in database and the application programs & queries submitted to the system.

- **Authorization Manager** — Checks the authority of users to access data.
- **Integrity Manager** — Checks for the satisfaction of the integrity constraints.
- **Transaction Manager** — Preserves atomicity and controls concurrency.
- **File Manager** — Manages allocation of space on disk storage.
- **Buffer Manager** — Fetches data from disk storage to memory for being used.

In addition to these functional units, several data structures are required to implement physical storage system. These are described below:

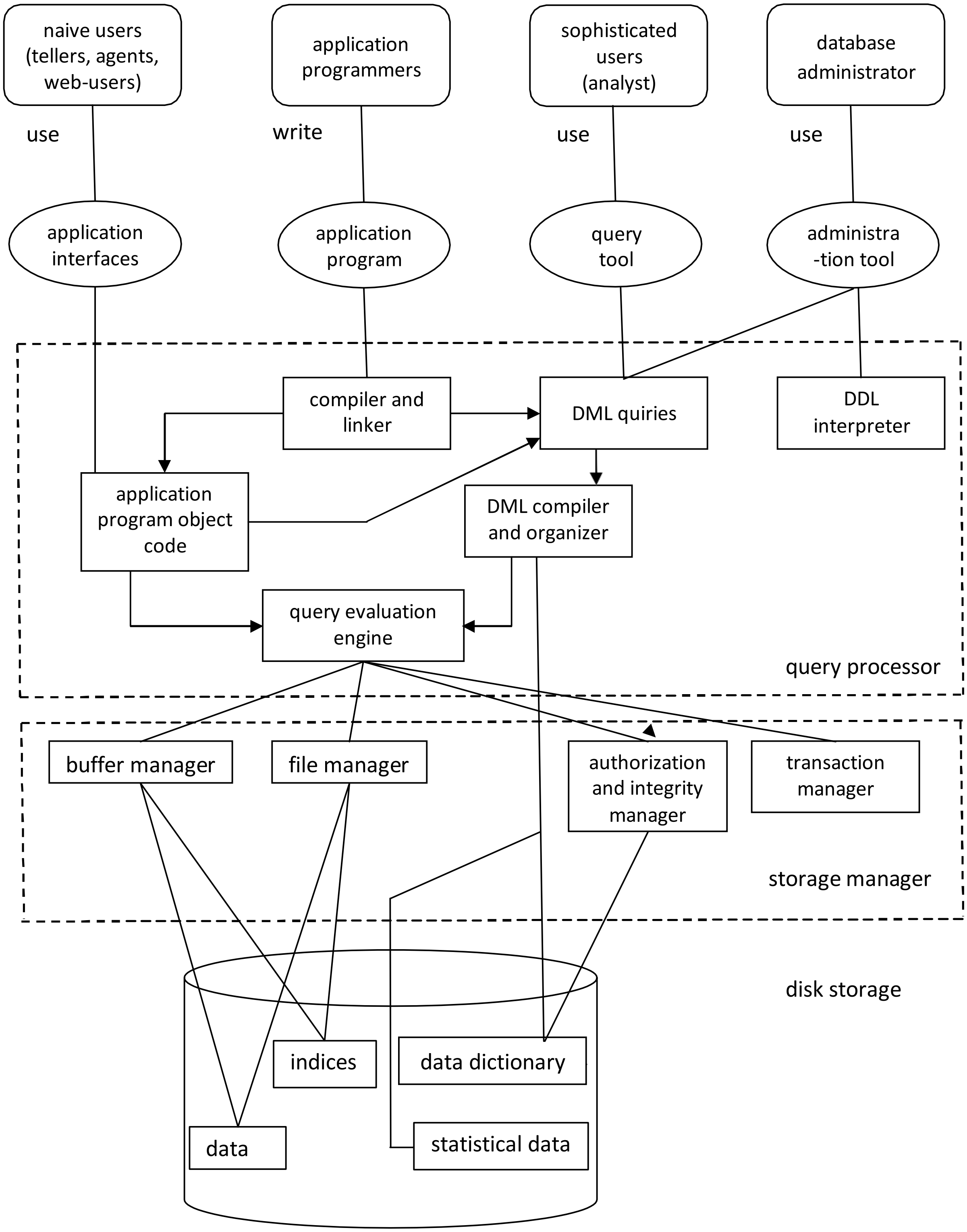
- **Data Files** — To store user data.
- **Data Dictionary and System Catalog** — To store metadata. It is used heavily, almost for each and every data manipulation operation. So, it should be accessed efficiently.
- **Indices** — To provide faster access to data items.
- **Statistical Data** — To store statistical information about the data in the database. This information is used by the query processor to select efficient ways to execute a query.

Transaction Manager:

- A transaction is a collection of operations that performs a single logical function in a database application.
- Transaction-management component ensures that the database remains in a consistent (correct)

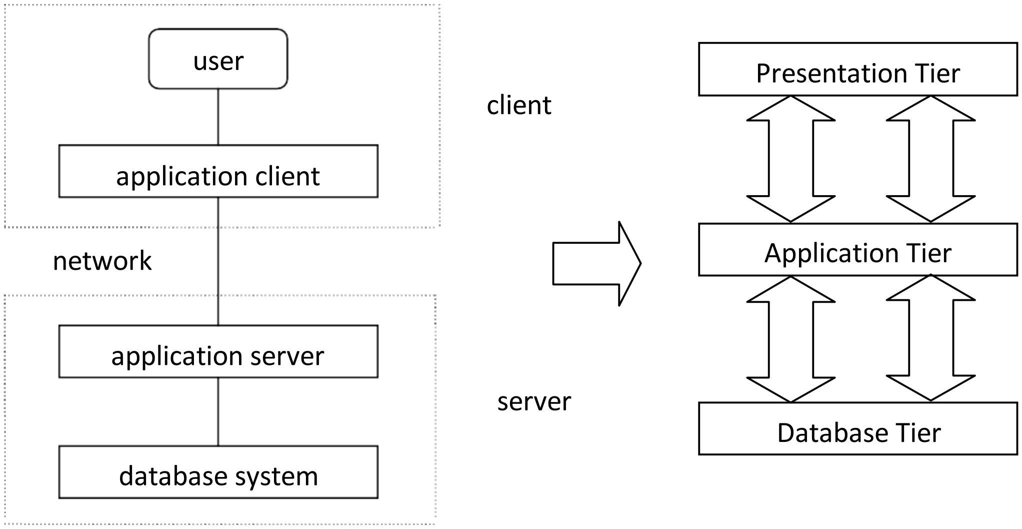
state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure consistency of the database.



Explain database system 3 tier architecture with clear diagram in detail.

- Most widely used architecture is 3-tier architecture.
- 3-tier architecture separates it tier from each other on basis of users.



Database (Data) Tier

- At this tier, only database resides.
- Database along with its query processing languages sits in layer-3 of 3-tier architecture.
- It also contains all relations and their constraints.

Application (Middle) Tier

- At this tier the application server and program, which access database, resides.
- For a user this application tier works as abstracted view of database.
- Users are unaware of any existence of database beyond application.
- For database-tier, application tier is the user of it.
- Database tier is not aware of any other user beyond application tier.
- This tier works as mediator between the two.

User (Presentation) Tier

- An end user sits on this tier.
- From a users aspect this tier is everything.
- He/she doesn't know about any existence or form of database beyond this layer.
- At this layer multiple views of database can be provided by the application.
- All views are generated by an application, which resides in application tier.

References :

- Chapter wise notes prepared by Prof. Firoz Sherasiya.
- Database System Concepts, Abraham Silberschatz, Henry F. Korth & S. Sudarshan.