

- End users :-

END users are those who actually reap the benefits of having a DBMS.

- Administrators :-

Administrators maintain the DBMS and are responsible for administrating the database.

- Designers :-

Designers are the group of people who actually works on the designing part of the database.

4. DBMS Architecture

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical.

The architecture of a DBMS can be seen as either single tier or mult-tier.

In 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.

DBMS BASICS

By - @codes.learning

1. What is Database?

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

2. What is DBMS ?

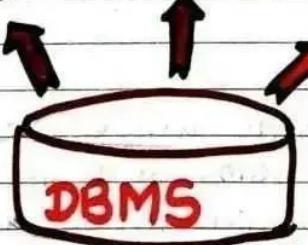
A DataBase Management System stores the data in a such a way that it becomes easier to retrieve, manipulate and produce information.

3. The users of a DBMS can be broadly categorized as :

ADMINISTRATORS

END USERS

DESIGNERS



If the Architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application.

A 3 tier Architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is most widely used architecture to design to a DBMS.

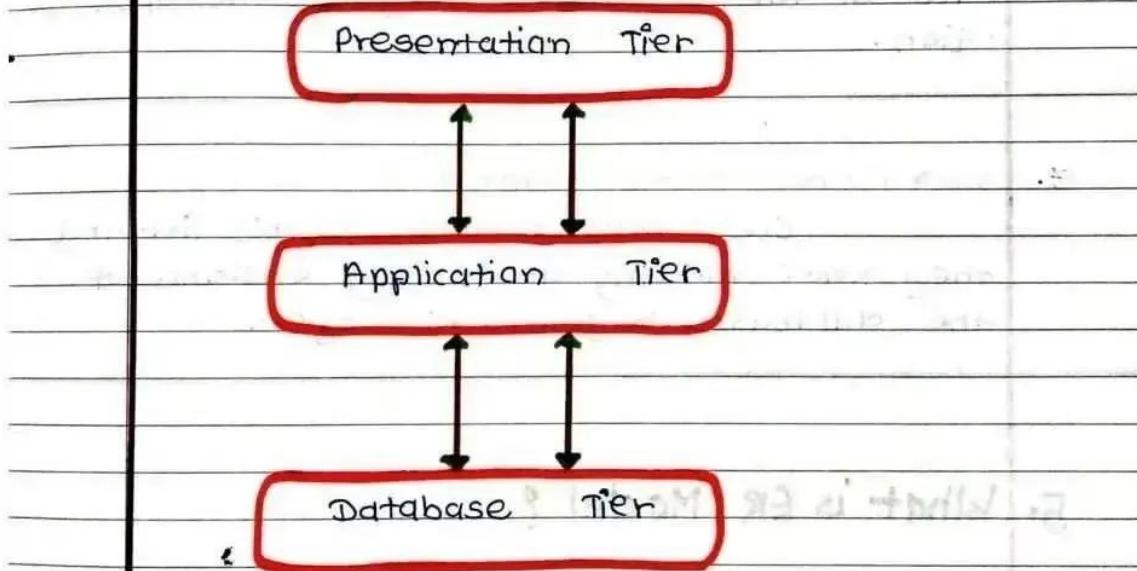


fig 1.2 (Architecture of DBMS)
3-tier.

1. Database (Data) Tier :-

- At this tier, the database resides along with its query processing languages.
- We also have the relations that define the data and their constraints at this level.

2. Application (Middle) Tier :-

- At this tier, reside the application server and programs that access the database for a user, this application tier presents an abstracted view of the database.
- END users are unaware of any existence of the database beyond the application.

3. User (presentation) Tier :-

- END users operate on this tier and they know nothing about any existence of the database beyond this layer.

5. What is ER Model ?

Entity-Relationship (ER) model is based on the notion of real-world entities and relationships among them.

ER model is best used for the conceptual design of a database

ER model is based on:

- Entities and their attributes
- Relationships among entities.

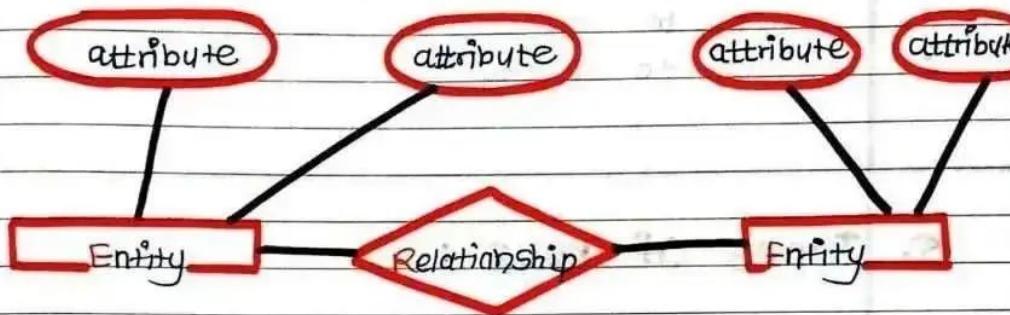


fig 1.3. (ER-model)

Entity :-

An entity in an ER-model is a real-world entity having properties called attributes. Every attribute is defined by the set of values called domain.

example :

database - student (school database) where

Entity - student

Attribute - name, age, class, etc.

Relationship :-

The logical association among entities is called Relationship. Relationships are mapped with entities in various ways.

Mapping cardinalities :-

- one to one
- one to many
- many to one
- many to many

6. Types of keys :

There are various types of keys in a relationship which are :

Candidate key :

The minimal set of attributes which can determine a tuple uniquely. There can be more than 1 candidate key of a relation and its proper subset can't determine tuple uniquely and it can't be null.

Super key :-

The set of attributes which can determine a tuple uniquely. A candidate key is always a super key but super key will

not be candidate key always.

● primary key :-

Among various candidate keys, one key is taken as primary key which is unique and other is alternate key.

● Foreign key :-

Foreign key is a set of attributes in a table which is used to prefer / refer the primary key or alternative key of the same or other table.

7. Operators :

i. select :-

The SELECT statement in SQL is used to retrieve or fetch data from a database. The data return is stored in the table called as result table. This result-table is called result-set.

syntax :-

SELECT column1, column2 FROM table-name

{ column1, column2 : names of fields of table }

table-name : from where we want to fetch.

②. from :-

From clause can be used to specify a sub-query expression in SQL. The relation produced by the sub-query is then used as a new relation on which outer query is applied.

Syntax :-

```
SELECT column1, column2 FROM  
(SELECT columnx AS c1, columny FROM table  
WHERE PREDICATE - x)  
WHERE PREDICATE ;
```

③. WHERE :-

WHERE keyword is used for fetching filtered data in a result set.

- It is used to fetch data according to a particular criteria.
- WHERE keyword can also be used to filter data by matching patterns.

Syntax :-

```
SELECT column1, column2 FROM  
table-name WHERE column-name  
operator value ;
```

④. EXISTS :-

The EXISTS condition in SQL is used to

check whether the result of a correlated nested query is empty or not.

Syntax :-

```
SELECT column-name(s)
FROM table-name
WHERE EXISTS
( SELECT column-name(s)
  FROM table-name
  WHERE condition);
```

⑤. GROUP BY :-

The GROUP BY statement in SQL is used to arrange identical data into groups with the help of some functions.

Syntax :-

```
SELECT column 1, function-name(column 2)
FROM table-name
WHERE condition
GROUP BY column 1, column 2
ORDER BY column 1, column 2;
```

⑥. ORDER BY :-

The ORDER BY statement in SQL is used to sort the fetched data in either