Q1. Write down the difference between DBMS and RDBMS.

DBMS	RDBMS
DBMS stands for Database Management System.	RDBMS stands for Relational Database Management System.
DBMS is a set of related records that used for accessing, storing and managing a data.	RDBMS is a set of software programs that used for creating, maintaining, modifying & determining relational database.
A General-purpose software provides the users with the process of defining, constructing & manipulating the database for various applications.	It can used to create the application that a user will require for interacting with the data that stored within the database.
The performance of DBMS operation is not good	The RDBMS operation gives better performance
It is not based on the concept of relationships	It is based on the concept of relationships
The speed of DBMS operation is very slow	The speed of RDBMS operation is very faster
Hardware and Software requirements are less	Hardware and Software requirements are high
Facilities and Utilities are limited.	Provide High Facilities and Utilities.
It uses the concept of file	It uses the concept of table
DBMS uses a 3GL (Third Graphical Language)	RDBMS uses a 4GL (Fourth Graphical Language)
Provides less security as compared to RDBMS	Provides high security as compared to DBMS.
Normalization is not present	Normalization is present
It deals with small quantity of data.	It can work with large amount of data.

Q2. Write a note on database users

- There are several types of database users.

❖ **Actors on the Scene:**

- Database Administrator
- Database Designers
- End Users
  - Casual End User
  - Native or Parameter End User
  - Sophistical End User
  - Standalone User
- System Analysts and Application Programmers (Software Engineers)

Database Administrator:-

- Administrating these resources is the responsibility of database administrator (DBA).
- The DBA is responsible for authorizing the access to the database coordinating and monitoring its use and acquiring software and hardware resources, as they required.
- The DBA is responsible for handling the problems such as breach of security or poor response time.

Database Designer:-

- The database designers are responsible for identify the data to be stored in the database and for choosing the appropriate structure to represent and store these data.

- Database designers communicate with potential users to understand their needs and create a design that meets their demands.

End User:-

- The end user are the persons whose jobs required access to the database for querying, updating and for generating the reports.
- Following are the various categories of End User.

1. Casual End Users:-

- This type of user occasionally accesses the database but they may require different information at each time.

2. Naive or Parametric End Users:-

- This type of user makes a sizeable portion of the database and the user.
- Their main job function revolves around consistently query and updating the database.
- The common examples of such type of user are bank clerks and reservation clerks.

3. Sophisticated End Users:-

- This category of end users include engineers, scientists, businessmen, and others who possess extensive knowledge of DBMS to effectively implement application that answer their intricate needs.

4. Standalone User:-

- This user maintains a personal database using a ready-made program with a simple menu or graphics interface.
- An example is the user of accounting package that stores variety of personal information and financial information for the tax planning purpose.

System Analysts and Application Programmers (Software Engineers):

- System analysts evaluate the needs of end users, especially naive and parametric ones, and provide canned transaction specifications.
- Application programmers convert these specifications into code, test, debug, document, and maintain scripted transactions.
- To do their jobs, analysts and programmers should know the DBMS's entire capabilities.

Workers behind the Scene:

1. DBMS system designers and implementers:

- They design and implement the DBMS modules and interfaces as a software package.
- The DBMS must interface with other system software, such as the operating system and compilers for various programming languages.

2. Tool developers:

- They design and implement tools—the software packages that facilitate database modeling and design, database system design, and improved performance.
- Tools are optional packages that often purchased separately.

3. Operators and maintenance personnel (system administration personnel):

- They are responsible for the actual running and maintenance of the hardware and software environment for the database system.

Q3. Explain functional dependency in detail

- The functional dependency is a relationship that exists between two attributes.
- Introduced by E. F. Codd, it helps in preventing data redundancy and gets to know about bad designs.
- Functional Dependency is represented by -> (arrow sign)
- let us consider P is a relation with attributes **A** and **B**.

A - > B

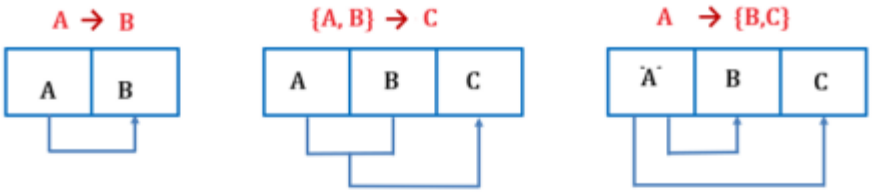
- The left side of this arrow is a Determinant.
- The right side of this arrow is a Dependent.
- In above example:
  - **A** is called **Determinant**, **B** is called the **Dependent** and B is functionally dependent on A
- Lets take one more example:
  - Here department table has two fields deptno and dept_name where deptno is primary key

Deptno	Dept_name
011	Marketing
022	HR
033	Finance
044	Accounting
055	Sales
066	Telecom

  - So if you want to know department name you need to know deptno.
  - So it is denotated by:

Deptno - > Dept_name

- Diagrammatic representation



Types of functional dependency:

- Following are the types of functional dependency
  - Full Functional Dependency
  - Partial Functional Dependency
  - Transitive Functional Dependency
  - Trivial Functional Dependency
  - Nontrivial Functional Dependency

Full Functional Dependency

- An attribute is fully functional dependent on another attribute, if it is Functionally Dependent on that attribute and not on any of its proper subset(attribute).
- Example:

supplier_id	item_id	price
1	1	540
2	1	545
1	2	200
2	2	201
1	1	540
2	2	201

- Price is fully functionally dependent on {supplier_id, item_id}.

{supplier_id, item_id}→ price

Q4. Explain normalization in details with example.

- Normalization is the process of organizing the data in the database.
- Normalization is use to minimize the redundancy from a relation or set of relations.
- Normalization is the process of organizing data in a database.
- Normalization helps to minimize data redundancy.
- The normal form is use to reduce redundancy from the database table.

Normal forms:

- Normal forms are:
  - **1NF (First Normal Form)**
  - **2NF (Second Normal Form)**
  - **3NF (Third Normal Form)**
  - **BCNF (Boyce–Codd normal form)**
  - **4NF (Forth normal form)**
  - **5NF (Fifth normal form)**

1NF (First Normal Form)

- A relation R is in first normal form (1NF) if and only if
  - Contains only atomic /single values
  - There are no **composite attribute** or multi-valued attributes

Customer ID	**Name**	**Address**
C01	Raj	Jamnagar Road, Rajkot
C02	Meet	Nehru Road, Jamnagar

- In above relation **address** is **composite attribute** which is further divided into sub-attributes as "Road" and "City".
- So above relation is not in 1NF.
- Solution:
- **Divide composite attributes** into **number of sub-attributes** and insert value in proper sub-attribute

Customer ID	Name	Road	City
C01	Raj	Jamnagar Road	Rajkot
C02	Meet	Nehru Road	Jamnagar

2NF (Second Normal Form)

- The second normal form must satisfy 2 conditions:
  - The table should be in first normal form.
  - There should be no partial dependency.
- Example:

TEACHER_ID	SUBJECT	TEACHER_AGE
111	JAVA	38
111	C++	38
222	PHP	36
333	DBMS	40
333	PHP	40

- This table is in 1NF
- It not in 2NF because non prime attribute Teacher_Age is dependent on Teacher id

- Solution: split down into two table based on prime attribute

Teacher details

TEACHER_ID	TEACHER_AGE
111	38
222	36
333	40

Teacher Subject

TEACHER_ID	SUBJECT
111	JAVA
111	C++
222	PHP
333	DBMS
333	PHP

3NF (Third Normal Form)

- The third normal form must satisfy 2 conditions:
  - It should be in the Second Normal form.
  - In addition, it should not have Transitive Dependency.
- Example:

RollNo	State	City
101	punjab	Maholi
102	gujarat	baroda
103	gujarat	Surat
104	punjab	Maholi
105	Nayan	Baroda
106	Bihar	Maholi

- Primary key: rollno
- Here the state →determine by city
- State is non prime so its called transitive dependency

Q5. A booking table contains following fields. Write down query to construct the table using constraints and write down queries for all instructions also.

Field Name	Data Type	Size	Constraint
BookingID	INT	—	PRIMARY KEY,
CustomerName	VARCHAR	20	NOT NULL
CustomerEmail	VARCHAR	20	NULL allowed
CustomerPhone	VARCHAR	15	NULL allowed
ServiceType	VARCHAR	20	NOT NULL
BookingDate	DATE	—	NOT NULL
BookingTime	TIME	—	NULL allowed
Status	VARCHAR	20	DEFAULT 'Pending'
PaymentStatus	VARCHAR	20	DEFAULT 'Unpaid'
Amount	DECIMAL	(10,2)	NULL allowed

❖ Create table

```
CREATE TABLE Booking (
  BookingID INT PRIMARY KEY,
  CustomerName VARCHAR2(20) NOT NULL,
  CustomerEmail VARCHAR2(20),
  CustomerPhone VARCHAR2(15),
  ServiceType VARCHAR2(20) NOT NULL,
  BookingDate DATE NOT NULL,
  BookingTime TIME,
  Status VARCHAR2(20) DEFAULT 'Pending',
  PaymentStatus VARCHAR2(20) DEFAULT 'Unpaid',
  Amount DECIMAL(10,2)
);
```

1. Describe table

DESC Booking;

OUTPUT 👇

Name	Null?	Type
BOOKINGID	NOT NULL	NUMBER(38)
CUSTOMERNAME	NOT NULL	VARCHAR2(20)
CUSTOMEREMAIL		VARCHAR2(20)
CUSTOMERPHONE		VARCHAR2(15)
SERVICETYPE	NOT NULL	VARCHAR2(20)
BOOKINGDATE	NOT NULL	DATE
BOOKINGTIME		TIME
STATUS		VARCHAR2(20)
PAYMENTSTATUS		VARCHAR2(20)
AMOUNT		NUMBER(10,2)

2. Insert statement to insert record

INSERT INTO Booking VALUES (1, 'Samir', 'samir@gmail.com', '9876543210', 'Cleaning', '2024-04-10', '10:00:00', 'Pending', 'Unpaid', 500.00);

INSERT INTO Booking VALUES (2, 'Suresh', NULL, '9876500000', 'Painting', '2024-04-12', '12:30:00', 'Completed', 'Paid', 1500.00);

INSERT INTO Booking VALUES (3, 'Ravi', 'ravi@yahoo.com', '9876511111', 'Plumbing', '2024-04-14', '09:30:00', 'Pending', 'Paid', 1200.00);

INSERT INTO Booking VALUES (4, 'Sneha', 'sneha@gmail.com', '9876522222', 'Electric', '2024-04-15', '15:00:00', 'Completed', 'Unpaid', 900.00);

INSERT INTO Booking VALUES (5, 'Amit', NULL, '9876533333', 'Repair', '2024-04-17', '11:00:00', 'Pending', 'Unpaid', 800.00);

OUTPUT 👇

5 rows created.

3. Display customer name, amount and payment status

SELECT CustomerName, Amount, PaymentStatus FROM Booking;

OUTPUT . 👇

CustomerName	Amount	PaymentStatus
Samir	500.00	Unpaid
Suresh	1500.00	Paid
Ravi	1200.00	Paid
Sneha	900.00	Unpaid
Amit	800.00	Unpaid

4. Display all customers whose name begin with ‘S’

SELECT * FROM Booking WHERE CustomerName LIKE 'S%';

Output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
1	Samir	samir@gmail.com	9876543210	Cleaning	2024-04-10	10:00:00	Pending	Unpaid	500
2	Suresh		9876500000	Painting	2024-04-12	12:30:00	Completed	Paid	1500
4	Sneha	sneha@gmail.com	9876522222	Electric	2024-04-15	15:00:00	Completed	Unpaid	900

5. Display all whose email id is NULL

SELECT * FROM Booking WHERE CustomerEmail IS NULL;

Output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
2	Suresh		9876500000	Painting	2024-04-12	12:30:00	Completed	Paid	1500
5	Amit		9876533333	Repair	2024-04-17	11:00:00	Pending	Unpaid	800

6. **Display all where status is ‘Pending’ and Payment status is ‘Paid’**

   SELECT * FROM Booking WHERE Status = 'Pending' AND PaymentStatus = 'Paid';

   output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
3	Ravi	ravi@yahoo.com	9876511111	Plumbing	2024-04-14	09:30:00	Pending	Paid	1200

7. **Display all where status is ‘Pending’ OR Payment status is ‘Unpaid’**

   SELECT * FROM Booking WHERE Status = 'Pending' AND PaymentStatus = 'Paid';

   Output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
1	Samir	samir@gmail.com	9876543210	Cleaning	2024-04-10	10:00:00	Pending	Unpaid	500
3	Ravi	ravi@yahoo.com	9876511111	Plumbing	2024-04-14	09:30:00	Pending	Paid	1200
4	Sneha	sneha@gmail.com	9876522222	Electric	2024-04-15	15:00:00	Completed	Unpaid	900
5	Amit		9876533333	Repair	2024-04-17	11:00:00	Pending	Unpaid	800

8. **Display all and arrange by payment status**

   SELECT * FROM Booking ORDER BY PaymentStatus;

   Output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
2	Suresh		9876500000	Painting	2024-04-12	12:30:00	Completed	Paid	1500
3	Ravi	ravi@yahoo.com	9876511111	Plumbing	2024-04-14	09:30:00	Pending	Paid	1200
1	Samir	samir@gmail.com	9876543210	Cleaning	2024-04-10	10:00:00	Pending	Unpaid	500
4	Sneha	sneha@gmail.com	9876522222	Electric	2024-04-15	15:00:00	Completed	Unpaid	900
5	Amit		9876533333	Repair	2024-04-17	11:00:00	Pending	Unpaid	800

9. **Display all and arrange by booking date**

   SELECT * FROM Booking ORDER BY BookingDate;

   Output 👇

BookingID	CustomerName	CustomerEmail	CustomerPhone	ServiceType	BookingDate	BookingTime	Status	PaymentStatus	Amount
1	Samir	samir@gmail.com	9876543210	Cleaning	2024-04-10	10:00:00	Pending	Unpaid	500
2	Suresh		9876500000	Painting	2024-04-12	12:30:00	Completed	Paid	1500
3	Ravi	ravi@yahoo.com	9876511111	Plumbing	2024-04-14	09:30:00	Pending	Paid	1200
4	Sneha	sneha@gmail.com	9876522222	Electric	2024-04-15	15:00:00	Completed	Unpaid	900
5	Amit		9876533333	Repair	2024-04-17	11:00:00	Pending	Unpaid	800

10. **Display booking id, customer name, booking date and time and arrange as per booking time**

    SELECT BookingID, CustomerName, BookingDate, BookingTime FROM Booking ORDER BY BookingTime;

    Output 👇

BookingID	CustomerName	BookingDate	BookingTime
3	Ravi	2024-04-14	09:30:00
1	Samir	2024-04-10	10:00:00
5	Amit	2024-04-17	11:00:00
2	Suresh	2024-04-12	12:30:00
4	Sneha	2024-04-15	15:00:00

11. Find out total amount according to payment status

SELECT PaymentStatus, SUM(Amount) AS TotalAmount FROM Booking GROUP BY PaymentStatus;

Output 👇

PaymentStatus	TotalAmount
Paid	2700
Unpaid	2200

12. Find out average amount according to status

SELECT Status, AVG(Amount) AS AverageAmount FROM Booking GROUP BY Status;

Output 👇

Status	AverageAmount
Completed	1200
Pending	833.3333333333334