

## Question 1

a) **What is a database management system (DBMS)? List any FIVE (5) advantages of DBMS.**

Ans:- A Database Management System (DBMS) is a software system that allows users to define, create, maintain, and manage databases. It provides an interface for users and applications to interact with the database, ensuring data integrity, security, and efficient data retrieval and manipulation.

Here are five advantages of using a DBMS:

- 1. Data Centralization:** DBMS centralizes data storage, making it easier to manage and access data from a single location. This eliminates data redundancy and ensures data consistency.
- 2. Data Security:** DBMS offers robust security features, such as user authentication, authorization, and encryption, to protect sensitive data from unauthorized access and data breaches.
- 3. Data Integrity:** It enforces data integrity constraints, ensuring that the data stored in the database follows predefined rules, thereby maintaining data accuracy and reliability.
- 4. Concurrent Access:** DBMS allows multiple users and applications to access the database simultaneously without conflicts, thanks to built-in concurrency control mechanisms.
- 5. Data Recovery:** It provides backup and recovery mechanisms to safeguard against data loss due to hardware failures, human errors, or other disasters, ensuring data availability and reliability.

**b) Differentiate between:**

**i) Logical and physical data independence**

Ans:-

<b>Logical Data Independence</b>	<b>Physical Data Independence</b>
It refers to the ability to change the logical schema of a database without affecting the application programs that use the data.	It pertains to the ability to modify the physical storage structures or organization of data without impacting the logical schema or application programs.
It primarily deals with the high-level, abstract view of the data, including the schema, tables, and relationships.	It focuses on the low-level aspects of data storage and retrieval, such as file organization, indexing, and storage media.
Changes to the logical schema do not require changes to application code, as long as the data's logical structure remains consistent.	Modifications to the physical storage, like moving data to a different disk or changing indexing mechanisms, should not affect application programs.
Offers flexibility in adapting the database schema to evolving business requirements without disrupting application functionality.	Provides flexibility in optimizing data storage and retrieval mechanisms for performance, scalability, or cost considerations.
If you add a new attribute to a table or redefine relationships between tables in the logical schema, the applications can still access the data without modification.	Changing the data storage from a traditional hard disk drive to a solid-state drive or altering the indexing structure to improve query performance should not require changes to the logical schema or application code.

## ii) External and conceptual level.

Ans:-

<b>External Level</b>	<b>Conceptual Level</b>
The external level is primarily concerned with the individual user's perspective. It deals with how data is presented and accessed by end-users or specific applications.	The conceptual level focuses on the overall structure and organization of the entire database system, abstracting away user-specific details.
It involves a high level of abstraction and customization, as it tailors data views and interactions to suit specific user requirements.	This level maintains a more abstract and global view of the data, representing the entire database without concern for individual user needs.
It offers a degree of data independence for users, allowing changes in the underlying database structure (conceptual level) without affecting how users interact with the data.	Changes at the conceptual level can impact the external level, potentially requiring modifications to user interfaces and applications.
Security and permissions are typically enforced at this level, ensuring that users only access the data they are authorized to see.	Security and permissions are typically defined at the conceptual level but implemented at the external level to restrict user access.
Users at the external level can have customized views of the data, allowing them to work with specific subsets or presentations of the data.	It does not deal with view customization; rather, it defines the overall data model and relationships for the entire database system.

**c) Identify THREE (3) roles of data administration and database administration.**

Ans:- Roles of Data Administration and Database Administration:

**Three roles of data administration:**

- 1. Data Architect:** Responsible for designing the overall structure and organization of data within the database. This includes defining data models, relationships, and schemas to ensure data consistency and integrity.
- 2. Data Steward:** Manages and maintains data quality, ensures compliance with data policies and standards, and oversees data governance practices. Data stewards are responsible for data documentation and metadata management.
- 3. Data Analyst:** Analyzes data to derive insights, create reports, and support decision-making processes. Data analysts work with various tools and techniques to extract meaningful information from the database.

**Three roles of database administration:**

- 1. Database Administrator (DBA):** Manages the technical aspects of the database system. This includes installing, configuring, and maintaining the DBMS software, monitoring database performance, optimizing queries, and ensuring data security and backup strategies.
- 2. Database Developer:** Develops and maintains database applications, including writing SQL queries, designing database schemas, and creating stored procedures. Database developers work closely with software developers to integrate databases into applications.
- 3. Database Security Administrator:** Focuses on database security, ensuring that user access is controlled, authentication and authorization mechanisms are in place, and sensitive data is encrypted and protected from unauthorized access or breaches.

**d) Provide TWO (2) methods for securing a database**

Ans:- Methods for Securing a Database:

**1. Access Control:** Implement strict access control measures by defining user roles and permissions. Users should only have access to the data and operations necessary for their job functions. Use role-based access control (RBAC) to manage permissions effectively.

**2. Encryption:** Encrypt sensitive data both at rest (stored in the database) and in transit (while it's being transmitted between the database and client applications). This includes using techniques like Transparent Data Encryption (TDE) for data at rest and Secure Sockets Layer (SSL)/Transport Layer Security (TLS) for data in transit.

## Question 2

You are working as a System Analyst at Zahra Corporation Sdn. Bhd. Your company required to have E-Appointment Management System to manage all appointment with clients. This system will have to cater all the necessary requirements pertaining to the scheduling of appointments.

Before the project starts, you have to collect and analyze the data requirements for new system. Based on the interview with expected user, below are the lists of data requirements.

- The data held on appointment is the appointment ID, appointment types, description, event, date, time, contact number, person to contact, staff identification number and client identification number.
- Staff information should have data about staff identification number, staff name, user login, password, email, address and contact number.
- The data held on client is the client identification number, client name, address, contact number and email.
- Staff has many roles. Clerk will manage all information about client that has an appointment. Whereby, management staff will use the system for updating new appointment and to retrieve appointment information belongs to him/her.
- Management staff will have more than one appointment.
- Each appointment will have one client.

a) Draw the complete entity-relationship (ER) diagram by showing the type of relationships between entities and cardinality constraints for the scenario above.

Ans:- In an ER diagram, entities are represented as rectangles, attributes as ovals, and relationships as diamonds. Cardinality constraints are denoted using lines and numbers. Based on the given scenario, here is the ER diagram:

### Entities:

#### 1. Appointment

- Attributes: appointment ID (PK), appointment type, description, event, date, time, contact number

### **- Relationships:**

- Related to Staff (Many-to-One, as many appointments can be associated with one staff)
- Related to Client (One-to-One, as each appointment is for one client)

## **2. Staff**

- Attributes: staff identification number (PK), staff name, user login, password, email, address, contact number
- Specialization: Staff has two roles - Clerk and Management Staff

## **3. Clerk (Specialized Entity, Subtype of Staff)**

- No additional attributes
- Relationships:**
  - Manages Client (One-to-Many, as a clerk can manage multiple clients)

## **4. Management Staff (Specialized Entity, Subtype of Staff)**

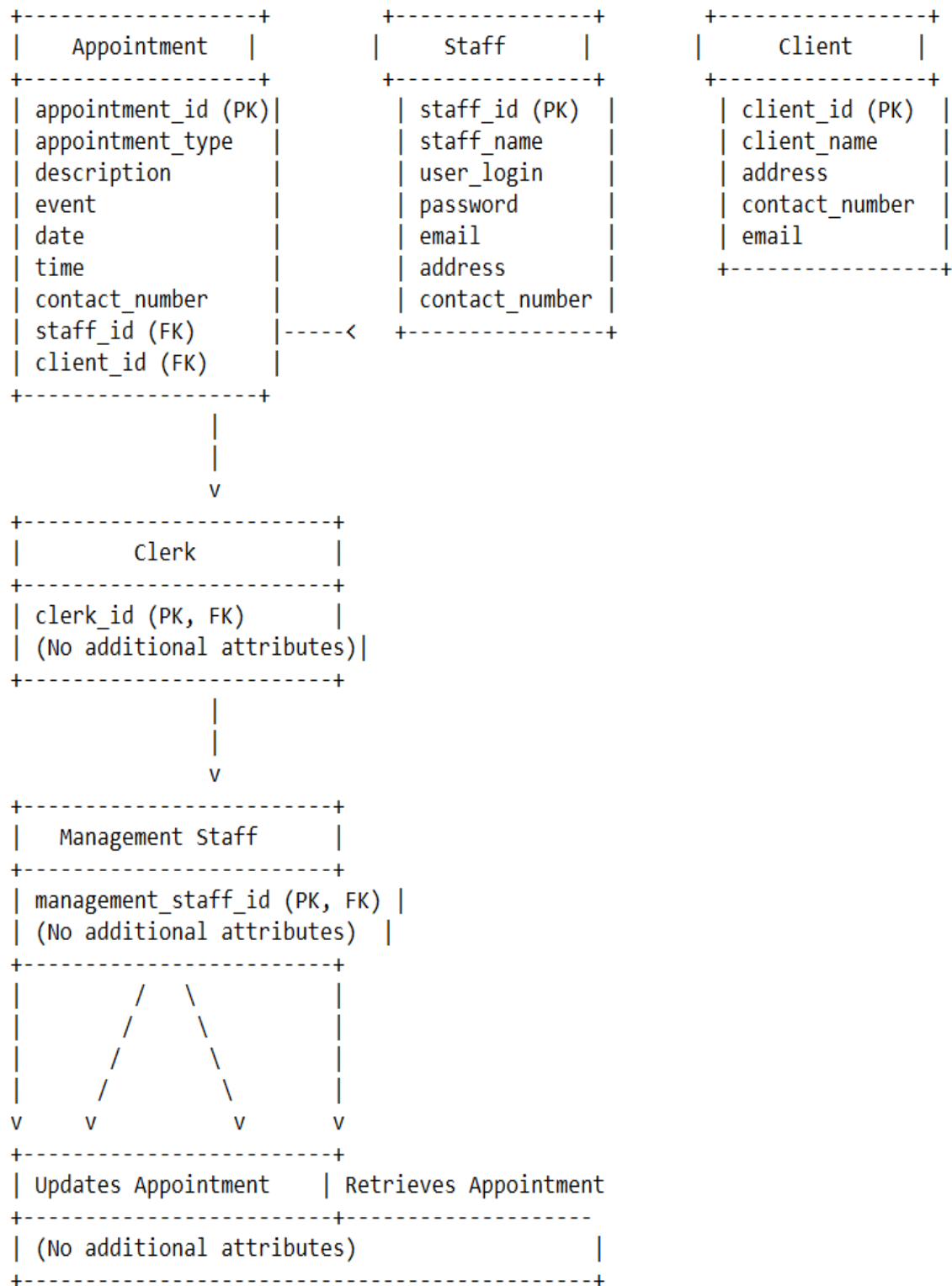
- No additional attributes
- Relationships:**
  - Updates Appointment (One-to-Many, as a management staff can update multiple appointments)
  - Retrieves Appointment (One-to-Many, as a management staff can retrieve multiple appointments)

## **5. Client**

- Attributes: client identification number (PK), client name, address, contact number, email

## Relationships:

- Appointment relates to Client (Many-to-One, as many appointments can be related to one client)





**b) Explain the TWO (2) types of constraints that may apply to specialization or generalization.**

Ans:- Types of Constraints for Specialization or Generalization:

**1. Disjointness Constraint:** This constraint defines how the subtypes (specialized entities) are divided or separated. There are two types of disjointness constraints:

- **Disjoint:** If entities in the specialization are mutually exclusive, meaning an instance of the superclass can be a member of only one subtype, it's called a disjoint specialization. In this case, a staff member can only be either a clerk or a management staff, not both.

- **Overlap:** If entities in the specialization can overlap, meaning an instance of the superclass can be a member of more than one subtype, it's called an overlapping specialization. In this case, a staff member can be both a clerk and a management staff.

**2. Completeness Constraint:** This constraint defines whether an instance of the superclass must belong to at least one subtype. There are two types of completeness constraints:

- **Total:** Every instance of the superclass must belong to one of the subtypes. In the given scenario, if all staff members must be either clerks or management staff (no staff member can exist without a subtype), it's a total specialization.

- **Partial:** Some instances of the superclass may not belong to any of the subtypes. In this case, it's a partial specialization.

**c) Define derived attribute. Provide an example for derived attribute.**

Ans:- A derived attribute is an attribute in a relational database that does not have its value stored explicitly but can be derived from other attributes in the database. It is typically calculated or derived using a defined algorithm or formula.

Example of a derived attribute:

In the context of your E-Appointment Management System, you could have a derived attribute called "Age" for clients. Instead of storing the client's age directly, you can calculate it based on the client's date of birth and the current date whenever needed. The "Age" attribute is derived from the "Date of Birth" attribute and the current date.

**Formula to calculate Age:  $\text{Age} = \text{Current Date} - \text{Date of Birth}$**

This approach ensures that the age is always up-to-date without the need to update it manually whenever a client's birthday passes.

### Question 3

**Table 1 shows list of dentist / patient appointment data. A patient is given an appointment at a specific time and date with a dentist located at a particular surgery. On each day of patient's appointments, a dentist is allocated to a specific surgery for that day.**

**Table 1 — Dentist Patient Uata Stall**

<b>Staff_No</b>	<b>Dentist_Name</b>	<b>Pat_No</b>	<b>Pat_Name</b>	<b>Appointment date time</b>	<b>Surgery_No</b>
S1011	Zara	P100	Roiana	12-Sep-17 10:00	S15
S1011	Zara	P105	Tiagu	12-Sep-17 12:00	S15
S1024	Aruna	P108	Andrew	12-Sep-17 10:00	S10
S1024	Aruna	P108	Andrew	14-Sep-17 12:00	S10
S1032	Robin	P105	Wong	14-Sep-17 10:00	S15

**a) The shown Table is susceptible to update anomalies. Provide ONE (1) example of insertion, deletion and update anomalies.**

Ans:-The shown Table 1 is susceptible to update anomalies. Here are examples of insertion, deletion, and update anomalies:

**Insertion Anomaly:** If you want to insert a new patient record but that patient doesn't have an appointment yet, you cannot do it in this table because it is structured to store both patient and appointment information together. This is an insertion anomaly.

**Deletion Anomaly:** If you delete a row for a patient's appointment, you would also delete information about the patient's dentist, which might be relevant for other appointments of the same patient. This is a deletion anomaly.

**Update Anomaly:** If a dentist changes their name, you would need to update multiple rows with the same dentist name. If you miss updating one of them, it can lead to inconsistency. This is an update anomaly.

**b) Illustrate the process of normalizing Table 1 to 3NF relations. Identify the primary key, alternate and foreign keys in your 3NF relations.**

Ans:- To normalize Table 1 to 3NF (Third Normal Form), we need to follow these steps:

**Step 1: Identify the Candidate Keys**

- First, let's identify the candidate keys in the table. A candidate key is a minimal set of attributes that can uniquely identify each tuple (row) in the table.
- In this table, 'Staff\_No' and 'Appointment date time' together can uniquely identify each appointment because a dentist can have multiple appointments on different dates and times. Therefore, (Staff\_No, Appointment date time) is a candidate key.

**Step 2: Eliminate Partial Dependencies**

- Partial dependencies occur when non-prime attributes (attributes not part of the candidate key) depend on only part of the candidate key. To eliminate partial dependencies, we'll create two new tables:

**1. Dentist\_Info (Dentist\_Name, Staff\_No, Surgery\_No)**

**2. Patient\_Appointments (Pat\_No, Pat\_Name, Appointment date time)**

**Step 3: Eliminate Transitive Dependencies**

- Transitive dependencies occur when non-prime attributes depend on other non-prime attributes. To eliminate transitive dependencies, we'll create a new table:

**3. Surgery\_Info (Surgery\_No, Staff\_No)**

Here are the three normalized tables in 3NF:

**Table 1: Dentist\_Info**

- Primary Key: Staff\_No
- Alternate Key: Dentist\_Name
- Foreign Key: Surgery\_No (references Surgery\_Info)

Staff_No	Dentist_Name	Surgery_No
S1011	Zara	S15
S1024	Aruna	S10
S1032	Robin	S15

**Table 2: Patient\_Appointments**

- Primary Key: Appointment date time
- Alternate Key: Pat\_No
- Foreign Key: Staff\_No (references Dentist\_Info)

Pat_No	Pat_Name	Appointment date time
P100	Roiana	12-Sep-17 10:00
P105	Tiagu	12-Sep-17 12:00
P108	Andrew	12-Sep-17 10:00
P108	Andrew	14-Sep-17 12:00
P105	Wong	14-Sep-17 10:00

**Table 3: Surgery\_Info**

- Primary Key: Surgery\_No
- Foreign Key: Staff\_No (references Dentist\_Info)

Surgery_No	Staff_No
S15	S1011
S10	S1024
S15	S1032

**c. Define Boyce Codd Normal Form (BCNF) and identify the relation you normalized in (b) is in BCNF already or not .**

Ans:- Boyce-Codd Normal Form (BCNF) is a higher level of normalization that ensures that for every non-trivial functional dependency  $X \rightarrow Y$  in a relation,  $X$  is a superkey. In other words, it ensures that there are no partial dependencies.

In the normalized relations from part (b), Table 4 (Appointments) is in BCNF because there are no partial dependencies, and for any functional dependency, the left side ( $X$ ) is a superkey.

**d) What Is the purpose of normalization?**

Ans:- The purpose of normalization in database design is to minimize data redundancy and eliminate data anomalies (insertion, deletion, and update anomalies) while preserving data integrity. Normalization achieves this by organizing data into multiple related tables, each with a specific purpose, and by ensuring that data is stored efficiently without unnecessary duplication. This leads to better data consistency, accuracy, and maintainability in the database, ultimately improving data quality and reducing the likelihood of errors.

#### Question 4

a) Define each of the following terms:

i) Relation Schema

ii) Data Definition Language

iii) Data Manipulation Language

Ans:-

**i) Relation Schema:** A relation schema, in the context of a relational database, defines the structure or blueprint of a table. It includes the table's name and a list of attributes (columns) along with their data types and constraints. It specifies what kind of data can be stored in the table and how it should be organized.

**ii) Data Definition Language (DDL):** DDL is a subset of SQL (Structured Query Language) used for defining and managing the structure of a database. It includes commands like CREATE, ALTER, and DROP, which are used to create and modify database objects such as tables, indexes, and constraints. DDL focuses on specifying the schema of a database.

**iii) Data Manipulation Language (DML):** DML, also part of SQL, is used to interact with the data stored in a database. It includes commands like SELECT, INSERT, UPDATE, and DELETE, which allow users to retrieve, insert, modify, and delete data in database tables. DML focuses on querying and modifying the data itself.

**b) Provide THREE (3) categories of data models.**

Ans:- Three categories of data models:

**1) Hierarchical Data Model:** In this data model, data is organized in a tree-like structure with a single parent for each data element. Each parent can have multiple children, forming a hierarchy. It's often used in file systems and older database systems.

**2) Network Data Model:** Similar to the hierarchical model, the network data model allows multiple relationships between data elements, creating a more flexible structure. It's represented as a graph, where records can be linked to multiple owner records. This model was popular in some early database systems.

**3) Relational Data Model:** The relational data model represents data as tables (relations) with rows and columns. It's based on set theory and predicate logic, providing a structured and easy-to-understand way to store and retrieve data. The relational model is widely used in modern database management systems (DBMS) like MySQL, PostgreSQL, and Oracle.



c) Write the SQL statements for the table illustrated below:

### TUTOR

TutorID	CertDate	Status
100	05/01/2016	Active
101	05/01/2016	Temp Stop
102	05/01/2016	Dropped
103	20/05/2016	Active
104	20/05/2016	Active
105	20/05/2016	Temp Stop

### STUDENT

StudentID	Read
3000	2.3
3001	5.6
3002	1.3
3003	3.3
3004	2.7
3005	4.8
3006	7.8

### MATCH HISTORY

MatchID	TutorID	StudentID	StartDate	EndDate
1	100	3000	10/01/2016	
2	101	3001	15/01/2016	15/05/2016
3	102	3002	10/02/2016	01/03/2016
4	105	3003	28/05/2016	
5	104	3004	06/01/2016	15/06/2016
6	104	3005	06/01/2016	28/06/2016
7	103	3006	06/01/2016	

i) List all the fields for the table STUDENT.

Ans:-

```
SELECT *
```

```
FROM STUDENT;
```

**ii) Count the number of tutors that have status of Temp Stop. Rename the title as 'NUMTSTOP'.**

Ans:-

```
SELECT COUNT(*) AS NUMTSTOP  
FROM TUTOR  
WHERE Status = 'Temp Stop';
```

**iii) Find out the tutors who are active.**

Ans:-

```
SELECT *  
FROM TUTOR  
WHERE Status = 'Active';
```

**iv) List the StudentID and student's Read score in descending order.**

Ans:-

```
SELECT StudentID, Read  
FROM STUDENT  
ORDER BY Read DESC;
```

**v) Count number of students that were taking the adult literacy program in first lry months of the year.**

Ans:-

```
SELECT COUNT(*) AS NumStudents  
FROM MATCH_HISTORY  
WHERE StartDate >= '01/01/2016' AND StartDate < '07/01/2016';
```