

**DEPARTMENT OF  
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**II YEAR – III SEMESTER**

**AD3391 DATABASE DESIGN AND MANAGEMENT**

**QUESTION BANK**

**ACADEMIC YEAR 2022-23**

# AD3391 – DATABASE DESIGN AND MANAGEMENT

## UNIT I

### CONCEPTUAL DATA MODELING

**Database environment – Database system development lifecycle – Requirements collection – Database design -- Entity-Relationship model – Enhanced-ER model – UML class diagrams.**

#### PART – A

**1. Explain the applications of DBMS?**

**Ans :** (a) Railway Reservation System,  
(b).library Management System,  
(c) Banking and  
(d) Education sector.

**2. What is a database environment?.**

**Ans :** A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data, which consist of software, hardware, people, techniques of handling database, and the data also.

**3. Mention the steps in Database development lifecycle.**

**Ans :** planning,  
requirement gathering,  
conceptual design,  
logical design,  
physical design,  
construction,  
implementation and rollout, and  
Ongoing support.

**4. What is database design?**

**Ans :** Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model.

**5. Define entity-relationship model.**

**Ans :** An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.

**6. List the types of entities in the ER model?**

**Ans :** A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

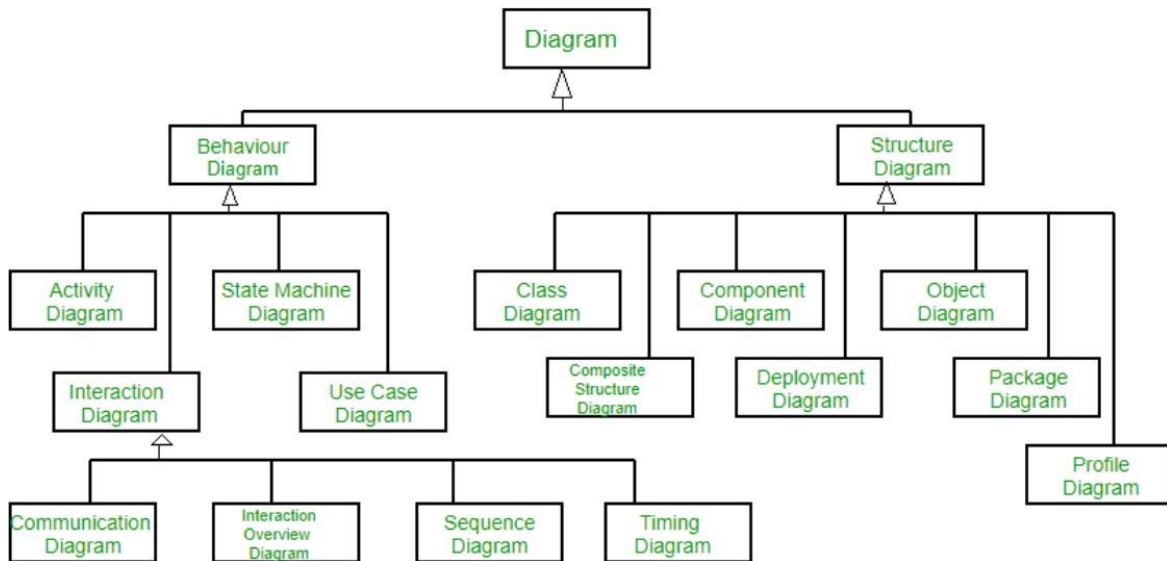
### 7. What is the enhanced ER model?

**Ans :** The enhanced entity–relationship (EER) model (or extended entity–relationship model) in computer science is a high-level or conceptual data model incorporating extensions to the original entity–relationship (ER) model, used in the design of databases.

### 8. What is the UML diagram?

**Ans :** a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects.

### 9. Draw the UML diagram hierarchy.



### 10. Mention the advantages of DBMS.

**Ans :** Better Data Transferring,  
Better Data Security,  
Better data integration,  
Minimized Data Inconsistency,  
Faster data Access,  
Better decision making,  
increased end-user productivity and Simple.

### 11. List the features of a database.

**Ans :** 1.It is a persistent (stored) collection of related data.  
2. The data is input (stored) only once.  
3. data is organized (in some fashion).  
4.The data is accessible and can be queried (effectively and efficiently).

**12. List the components of DBMS.**

**Ans :** The major components of database management system are,

- Software
- Hardware
- Data
- Procedures
- Database Access Language and
- Users.

**13. Give the limitations of ER model? How do you overcome this?**

**Ans :** The entity relationship model is a collection of basic objects called entities And relationship among those objects. An entity is a thing or object in the Real world that is distinguishable from other objects.

**14. What are the features of a good database design.**

**Ans :** Reflects real-world structure of the problem,  
Can represent all expected data over time,  
Provides efficient access to data,  
Avoids redundant storage of data items and  
Clean, consistent, and easy to understand.

**15. List the characters of conceptual data model.**

**Ans :** An overall view of the structure of the data in a business context.  
Features that are independent of any database or physical storage structure.  
Objects that may not ever be implemented in physical databases.

**16. Application of ER model.**

**Ans :** Database design,  
Database troubleshooting,  
Business,  
Education and  
Research.

**18. What is data model? Explain various data models with example?**

**Ans :** There are three main models of data modeling like conceptual, logical, and physical. A conceptual model is used to establish the entities, attributes, and relationships. A logical data model is to define the structure of the data elements and set the relationship between them. Finally, the physical model is used to specify the database-centric implementation of the model.

**19. What are the categories of data models.**

**Ans :** **High level/conceptual data models** –provide concepts close to the way Users perceive the data.

**Physical data models** –provide concepts that describe the details of how data is stored in the computer.

**Representational data models** –provide concepts that may be Understood by the end user but not far removed from the way data is Organized.

**20. What is relational model.**

**Ans :** The relational model represents the database as a collection of relations. A Relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship

**PART – B**

- 21.** List and describe the features and purpose of database?
- 22.** Compare database vs file processing system?
- 23.** Discuss the correspondence between the ER model construct and the Relational model constructs. Show how each ER model construct can be Mapped to the relational model. Discuss the option for mapping EER Model construct
- 24.** List and describe the components of database management system with neat Diagram.
- 25.** Explain the database system architecture with neat diagram?

# AD3391 – DATABASE DESIGN AND MANAGEMENT

## UNIT II

### RELATIONAL MODEL AND SQL

**Relational model concepts -- Integrity constraints -- SQL Data manipulation – SQL Data definition – Views -- SQL programming.**

#### PART – A

##### 1. What is Relational Model?

**Relational Model (RM)** represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

**Some popular Relational Database management systems are:**

- DB2 and Informix Dynamic Server – IBM
- Oracle and RDB – Oracle
- SQL Server and Access – Microsoft

##### 2. Relational Model Concepts in DBMS

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student\_Rollno, NAME, etc.
2. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
3. **Tuple** – It is nothing but a single row of a table, which contains a single record.
4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.
5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.
6. **Cardinality:** Total number of rows present in the Table.
7. **Column:** The column represents the set of values for a specific attribute.
8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. **Relation key** – Every row has one, two or multiple attributes, which is called relation key.
10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain

### 3. Relational Integrity Constraints

Relational Integrity constraints in DBMS are referred to conditions which must be present for a valid relation. These Relational constraints in DBMS are derived from the rules in the mini-world that the database represents.

There are many types of Integrity Constraints in DBMS. Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain Constraints
2. Key Constraints
3. Referential Integrity Constraints

### 4. Domain Constraints

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

#### Example:

```
Create DOMAIN CustomerName  
CHECK (value not NULL)
```

### 5. Key Constraints

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

#### Example:

In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =” Google”.

CustomerID	Customer Name	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

### 6. Referential Integrity Constraints

Referential Integrity constraints in DBMS are based on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

#### Example:

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Customer

InvoiceNo	CustomerID	Amount
1	1	\$100
2	1	\$200
3	2	\$150

Billing

In the above example, we have 2 relations, Customer and Billing.  
 Tuple for CustomerID =1 is referenced twice in the relation Billing. So we know  
 CustomerName=Google has billing amount \$30

## 7. Operations in Relational Model

Four basic update operations performed on relational database model are  
 Insert, update, delete and select.

- Insert is used to insert data into the relation
- Delete is used to delete tuples from the table.
- Modify allows you to change the values of some attributes in existing tuples.
- Select allows you to choose a specific range of data.

## 8. Advantages of Relational Database Model

- **Simplicity:** A Relational data model in DBMS is simpler than the hierarchical and network model.
  - **Structural Independence:** The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
  - **Easy to use:** The Relational model in DBMS is easy as tables consisting of rows and columns are quite natural and simple to understand
  - **Query capability:** It makes possible for a high-level query language like [SQL](#) to avoid complex database navigation.
  - **Data independence:** The Structure of Relational database can be changed without having to change any application.
  - **Scalable:** Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.



## 9. Disadvantages of Relational Model

- Few relational databases have limits on field lengths which can't be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

## 10. Integrity Constraints in the Relational Model Definition.

Integrity constraints are logical statements that restrict the set of allowable relations in a database.

Example.

- database schema,  $R = \{EMP, DEPT\}$ , with
- $schema(EMP) = \{ENAME, DNAME, ADDRESS, POSTCODE, LOC\}$
- $schema(DEPT) = \{DNAME, MNAME, NO EMPs, LOC\}$ .
- database,  $d = \{r1, r2\}$  OVER  $R$ ,
- $r1$  is a relation over  $EMP$ , and
- $r2$  is a relation over  $DEPT$

## 11. Functional Dependencies

- Stating that  $ENAME$  is a key of  $EMP$ , means that no two distinct tuples in  $r1$  have the same  $ENAME$ .
- Stating that  $DNAME$  is a key of  $DEPT$ , means that no two distinct tuples in  $r2$  have the same  $DNAME$ .? Keys are special cases of Functional Dependencies (FDs).
- An example of an FD which is not the result of a key, is the constraint that an  $ADDRESS$  has a unique  $POSTCODE$ .

## 12. SQL statements are divided into two major categories:

**Data Definition Language (DDL)** and **Data Manipulation Language (DML)**. Both of these categories contain far more statements than we can present here, and each of the statements is far more complex than we show in this introduction. If you want to master this material, we strongly recommend that you find a SQL reference for your own database software as a supplement to these pages.

- Few relational databases have limits on field lengths which can't be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

### 13. Data definition language

DDL statements are used to build and modify the structure of your tables and other objects in the database. When you execute a DDL statement, it takes effect immediately.

The create table statement does exactly that:

```
CREATE TABLE <table name> (  
  <attribute name 1> <data type 1>,  
  ...  
  <attribute name n> <data type n>);
```

the **data types** that you will use most frequently are character strings, which might be called **VARCHAR** or **CHAR** for **variable** or **fixed length strings**; **numeric types** such as **NUMBER** or **INTEGER**, which will usually specify a precision; and **DATE** or related types. **Data type syntax is variable from system to system; the only way to be sure is to consult the documentation for your own software.**

### 14. Data manipulation language

DML statements are used to work with the data in tables. When you are connected to most multi-user databases (whether in a client program or by a connection from a Web page script), you are in effect working with a private copy of your tables that can't be seen by anyone else until you are finished (or tell the system that you are finished). You have already seen the **SELECT** statement; it is considered to be part of DML even though it just retrieves data rather than modifying it.

The insert statement is used, obviously, to add new rows to a table.

```
INSERT INTO <table name>  
VALUES (<value 1>, ... <value n>);
```

### 15. Database views

A *database view* is a subset of a database and is based on a query that runs on one or more database tables. Database views are saved in the database as named queries and can be used to save frequently used, complex queries.

There are two types of database views: dynamic views and static views. Dynamic views can contain data from one or two tables and automatically include all of the columns from the specified table or tables. Dynamic views are automatically updated when related objects or extended objects are created or changed. Static views can contain data from multiple tables and the required columns from these tables must be specified in the **SELECT** and **WHERE** clauses of the static view. Static views must be manually updated when related objects or extended objects are created or changed.

## 16. SQL technique: views and indexes

A **view** is simply any SELECT query that has been given a name and saved in the database. For this reason, a view is sometimes called a **named query** or a **stored query**. To create a view, you use the SQL syntax:

```
CREATE OR REPLACE VIEW <view_name> AS  
SELECT <any valid select query>;
```

The view query itself is saved in the database, but it is not actually run until it is called with another SELECT statement. For this reason, the view does not take up any disk space for data storage, and it does not create any redundant copies of data that is already stored in the tables that it references (which are sometimes called the **base tables** of the view).

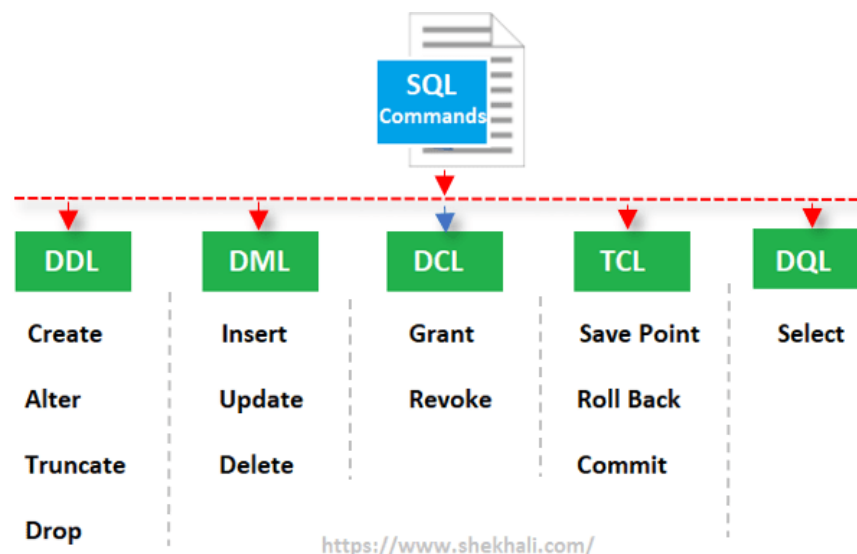
## 17. Materialized views

The execution speed of a query is so important that a developer is willing to trade increased disk space use for faster response, by creating a **materialized view**. Unlike the view discussed above, a materialized view *does* create and store the result table in advance, filled with data. The scheme of this table is given by the SELECT clause of the view definition.

## 18. Indexes

An **index**, as you would expect, is a data structure that the database uses to find records within a table more quickly. Indexes are built on one or more columns of a table; each index maintains a list of values within that field that are sorted in ascending or descending order. Rather than sorting records on the field or fields during query execution, the system can simply access the rows in order of the index.

## 19. What are the 5 basic SQL commands?



**There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.**

- Data Definition Language (DDL) DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc. ...
- Data Manipulation Language. ...
- Data Control Language. ...
- Transaction Control Language. ...
- Data Query Language.

**20. What is SQL example?**

Structured Query Language (SQL) is **a specialized language for updating, deleting, and requesting information from databases**. SQL is an ANSI and ISO standard, and is the de facto standard database query language.

### **PART - B**

1. What is relational data model explain with example?
2. What are the components of relational model explain each?
3. What do you mean by integrity constraints explain the two types of constraints?
4. What are the types of integrity constraints?
5. What are data manipulation statements explain in detail?
6. What is data definition language and data manipulation language give one example of each?
7. What is Data Definition Language (DDL) and Data Manipulation Language (DML)? Give one example of each?
8. What is the difference between Data Definition Language and Data Manipulation Language?
9. What is SQL explain with an example?
10. What is SQL explain DDL, DML and DCL?

# AD3391 – DATABASE DESIGN AND MANAGEMENT

## UNIT -III

### RELATIONAL DATABASE DESIGN AND NORMALIZATION

**ER and EER-to-Relational mapping – Update anomalies – Functional dependencies – Inference rules – Minimal cover – Properties of relational decomposition – Normalization (upto BCNF).**

#### PART - A

##### **1. Why 4NF in Normal Form is more desirable than BCNF?**

Ans: Because 4NF minimize the redundancy as well as make storage management. Redundancy is reduced as we normalize it further and this avoids consistency problems.

##### **2. Define functional Dependency?**

Ans: Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as  $X \rightarrow Y$ , which specifies Y is functionally dependent on X. Functional Dependency (FD) is a set of constraints between two attributes in a relation.

##### **3. State the anomalies of 1NF?**

Ans: 1NF databases have some problems: Most notable:

repetition of data

to change a department name all tuples of the relation need to be updated since the department name can exist in multiple rows.

##### **4. Explain entity relationship model?**

Ans: ER model defines the mapping between the entities in the database ER model is a graphical representation of real world objects with their attributes and relationship. It makes the system easily understandable. This model is considered as a top down approach for designing a requirement.

##### **5. What is meant by lossless-join decomposition?**

We claim the above decomposition is lossless. How can we decide whether decomposition is lossless? 1. Let R be a relation schema. 2. Let F be a set of functional dependencies on R. 3. Let R1 and R2 form a decomposition of R. 4. The decomposition is a lossless-join decomposition of R if at least one of the following functional dependencies are in :

a.  $R1 \cap R2 \rightarrow R1$

b.  $R1 \cap R2 \rightarrow R2$

**6) Define Boyce codd normal form? Why BCNF Stricter than 3NF?**

Ans : A relation schema R is in BCNF with respect to a set F of functional dependencies if, for all functional dependencies in F, BCNF is stricter than 3NF because each and every BCNF is relation to 3NF but every 3NF is not relation to BCNF. 4. BCNF non-translationally depends on individual candidate key but there is no such requirement in 3NF.Hence BCNF is stricter than 3NF.

**7)What is meant by functional dependencies? What are the uses of functional dependencies?**

Ans : :Consider a relation schema R and  $\alpha \subset R$  and  $\beta \subset R$ . The functional dependency  $\alpha \rightarrow \beta$  holds on relational schema R if in any legal relation  $r(R)$ , for all pairs of tuples  $t_1$  and  $t_2$  in r such that  $t_1[\alpha] = t_2[\alpha]$ , and also  $t_1[\beta] = t_2[\beta]$ .

To test relations to see whether they are legal under a given set of functional dependencies. To specify constraints on the set of legal relations.

**8) Explain trivial dependency?**

Ans :Functional dependency of the form  $\alpha \rightarrow \beta$  is trivial if  $\beta \subset \alpha$  Trivial functional dependencies are satisfied by all the relations.

**9) What is meant by normalization of data and Denormalization?**

Ans : It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties Minimizing redundancy Minimizing insertion, deletion and updating anomalies. Denormalization: It is the process of attempting to optimize the performance of a database by adding redundant data or by grouping data.

**10. Give the properties of decomposition.**

Ans :Lossless-join decomposition Dependency preservation Repetition of information

**11. What is 2NF?**

Ans : A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.00

**12. Define Domain / key normal form?**

It is a normal form used in database normalization which requires that the database contains no constraints other than domain constraints and key constraints.

**13) What are the desirable properties of decomposition?**

Lossless join and dependency preserving are the two desirable properties of decomposition. Lossless join decomposition property: Let R be the relational schema with instance r is decomposed into  $R_1, R_2, \dots, R_n$  with instance  $r_1, r_2, \dots, r_n$ . If  $r_1 \bowtie r_2 \bowtie \dots \bowtie r_n = r$ , then it is called Lossless Join Decomposition. i.e. if natural joins of all the decompositions gives the original relation, then it is said to be Lossless Join Decomposition. Dependency

preserving Property: The second property of decomposition is Dependency Preserving Decomposition

If the original table is decomposed into multiple fragments, then somehow, we suppose to get all original FDs from these fragments. In other words, every dependency in original table must be preserved or say, every dependency must be satisfied by at least one decomposed table.

**14) What is an entity relationship model?**

The entity relationship model is a collection of basic objects called entities and relationship among those objects. An entity is a thing or object in the real world that is distinguishable from other objects.

**15) What are attributes? Give examples.**

Ans : An entity is represented by a set of attributes. Attributes are descriptive properties possessed by each member of an entity set. Example: possible attributes of customer entity are customer name, customer id, customer street, customer city

**16) What is relationship? What is meant by the degree of relationship set?**

Ans : A relationship is an association among several entities. Example: A depositor relationship associates a customer with each account that he/she has. The degree of relationship type is the number of participating entity types

**17) Define the terms Entity set and Relationship set?**

Ans: Entity set: The set of all entities of the same type is termed as an entity set. Relationship set: The set of all relationships of the same type is termed as a relationship set.

**18) Define single valued and multivalued attributes.**

Ans : Single valued attributes: attributes with a single value for a particular entity are called single valued attributes. Multivalued attributes: Attributes with a set of value for a particular entity are called multivalued attributes.

**19) What are stored and derived attributes?**

Ans :Stored attributes: The attributes stored in a data base are called stored attributes. Derived attributes: The attributes that are derived from the stored attributes are called derived attributes.

**20) Define the terms i) Entity type ii) Entity set**

Ans : Entity type: An entity type defines a collection of entities that have the same attributes. Entity set: The set of all entities of the same type is termed as an entity set.

**21) Define weak and strong entity sets?**

Ans :Weak entity set: entity set that do not have key attribute of their own are called weak entity sets. Strong entity set: Entity set that has a primary key is termed a strong entity set.

## **PART – B**

**22)** Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time and has an associated due date, and the date when the payment was received?

**23)** Discuss the correspondence between the ER model construct and the relational model constructs. Show how each ER model construct can be mapped to the relational model. Discuss the option for mapping EER model construct?

**24)** Explain in detail about Functional Dependencies? Briefly discuss about the Functional Dependency Concepts?

**25)** A car rental company maintains a database for all vehicles in its current fleet. For all vehicles, it includes the vehicle identification number, license number, manufacturer, model, date of purchase and color. Special data are included for certain types of vehicles. Trucks: cargo capacity Sports car: horsepower, renter age requirement Vans: number of passengers Off-road vehicle: ground clearance, drivetrain (four or two-wheeler drive) Construct an ER model for the car rental company database.

**26)** State the need for normalization of a Database and Explain the various Normal Forms (1st, 2nd, 3rd, BCNF, 4th, 5th and Domain-key) with suitable examples?

(Or)

Exemplify multivalued dependency and fourth normal form (4NF) and join dependency and fifth 3 normal form (5NF) ?

((OR)

What is Normalization? Explain in detail about all Normal Forms

**27)** Draw E-R diagram for the “Restaurant menu ordering system” that will facilitate the food items ordering and services within a restaurant. The entire restaurant scenario is detailed as follows. The customer is able to view the food items menu, call the waiter, place orders and obtain the final bill through the computer kept in their table. The waiters through their wireless tablet PC are able to initialize a table for customer, control the table functions to assist customers, orders, send orders to food preparation staff (chef) and finalize the customer's bill. The food preparation staffs (chefs), with their touch-display interfaces to the system, are able to view orders sent to the kitchen by waiters. During preparation, they are able to let the waiter know the status of each item and can send notifications when items are completed. The system should have full accountability and logging facilities and should support supervisor actions to account for exceptional circumstances such as meal being refunded or walked out on?



**28)** Explain first normal form, second normal form, third normal form and BCNF with an example?

**29)** Design and draw an E-R diagram for university database?

# AD3391 – DATABASE DESIGN AND MANAGEMENT

## UNIT IV

### TRANSACTION MANAGEMENT

**Transaction concepts – properties – Schedules – Serializability – Concurrency Control – Two- phase locking techniques.**

#### PART – A

1. What is transaction?

**Ans :** Collections of operations that form a single logical unit of work are Called transactions.

2. Define lock?

**Ans :** Lock is the most common used to implement the requirement is to allow a Transaction to access a data item only if it is currently holding a lock on that item

3. What are the properties of transaction.

**Ans :** Atomicity, Consistency, Isolation and Durability.

4. What are the different modes of lock?

**Ans :** The modes of lock are:

- Shared /Read
- Exclusive /Write

5. When is a transaction rolled back?

**Ans :** Any changes that the aborted transaction made to the database must be Undone.. Once the changes caused by an aborted transaction have been undone, then the transaction has been rolled back.

6. What are the two statements regarding transaction?

**Ans :** The two statements regarding transaction of the form are  
Begin transaction and  
End transaction

7. Define atomicity?

**Ans :** Either all operations of the transaction are reflected properly in the database Or none are.

8. Define Durability?

**Ans :** A transaction completes successfully then changes It has made to the Database persist even if there are system failure.

9. What is transaction-management component?

**Ans :** Ensuring atomicity is the responsibility of the database system itself Specifically, it is handled by a component called the transaction-Management component.

**10.** What are the two operation for accessing data in transaction?

**Ans :** Read(x)- transfer data item x from database.

Write(x)- transfer data item x from the local buffer.

**11.** What do you mean by read only transaction?

**Ans :** The data base operation in a transaction do not update the database but only

Retrieve data, the transaction is called a read-only transaction.

**12.** What are the steps followed in Executing read(x) command in Transaction?

**Ans :** Find the address of the disk block that contains item x.

Copy that disk block in to a buffer in main memory.

Copy that disk block in to a buffer in main memory.

**13.** What are the steps followed in executing write(x) command in Transaction?

**Ans :** Find the address of disk block that contain item x.

Copy that disk block into buffer in main memory.

Copy item x from the program variable named x into its correct location

In the buffer

Store the update block from the buffer back to disk.

**14.** List out the transaction states?

**Ans :** i) active,

ii) Partially Committed,

iii) Failed,

iv) Abort and

v) committed.

**16.** What is serializable schedule? April/May 2017

**Ans :** To process transactions concurrently, the database server must execute some

Component statements of one transaction, then some from other

Transactions, before continuing to process further operations from the first.

The order in which the component operations of the various transactions are

Interleaved is called the schedule.

**17.** What are the three problems that any concurrency control mechanism must address?

**Ans:** The three problems are:

- The lost update problem
- The uncommitted dependency problem
- The inconsistent analysis problem

**18.** What are the need for concurrency?

**Ans :** Improved throughput and resource utilization and

Reduced waiting time.

**19.** What is the last update problem? (U)

**Ans :** Transaction A retrieves some tuple t at time t1; transaction B retrieves that same tuple t at time t2; transaction A updates the tuple at time t3; and transaction B updates the same tuple at time t4; Transaction A's update is lost at time t4, because transaction B overwrites it without even looking at it.

**20.** What is the uncommitted dependency problem? (U)

**Ans :** The uncommitted dependency problem arises if one transaction is allowed to retrieve-or, worse, update-a tuple that has been updated by another transaction but not yet committed by that other transaction.

### **PART – B**

**21.** Write Short Notes On Transaction Concept & Transaction State?

**22.** What is Serializability? Explain Its Types?

**23.** Explain testing for serializability with respect to concurrency control schemes. How will you determine, whether a schedule is serializable or not?

**24.** Why is Recovery needed? Discuss any two Recovery Techniques.

**25.** Explain why timestamp-based concurrency control allows schedules that are not recoverable. Describe how it can be modified through buffering to disallow such schedules.

# AD3391 – DATABASE DESIGN AND MANAGEMENT

## UNIT – V

### OBJECT RELATIONAL AND NO-SQL DATABASES

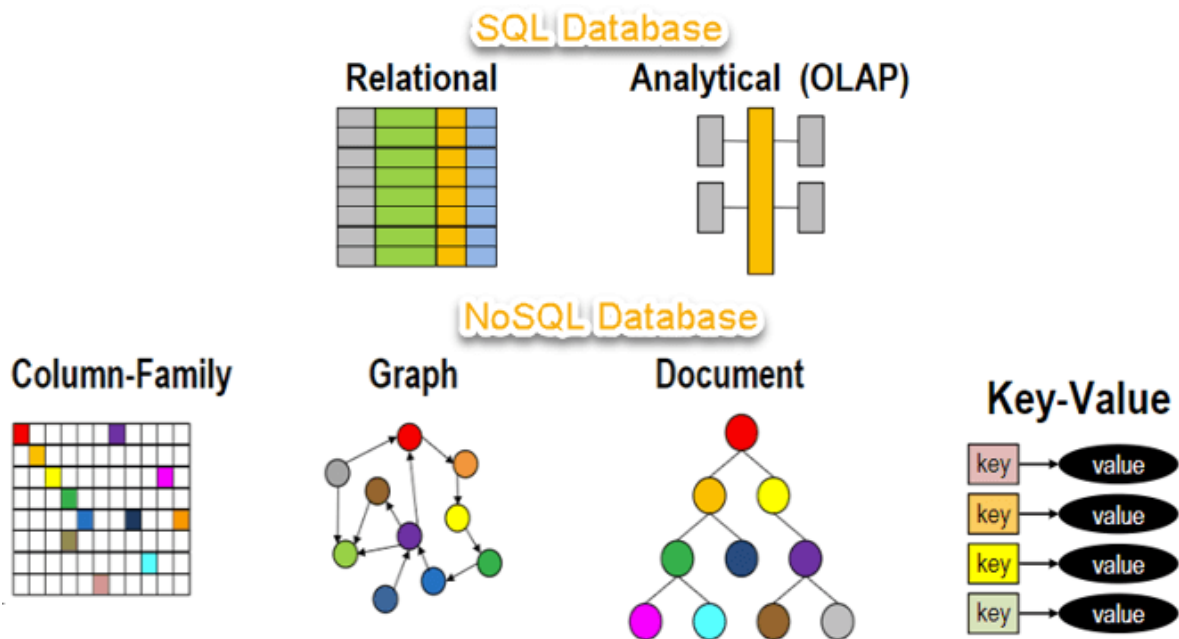
Mapping EER to ODB schema – Object identifier – reference types – rowtypes – UDTs – Subtypes and supertypes – user-defined routines – Collection types – Object Query Language; No-SQL: CAP theorem – Document-based: MongoDB data model and CRUD operations; Column-based:Hbase data model and CRUD operations.

#### PART - A

##### 1. What is NoSQL?

NoSQL, also referred to as “not only SQL”, “non-SQL”, is **an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases.**

##### 2. What is NoSQL and its types?

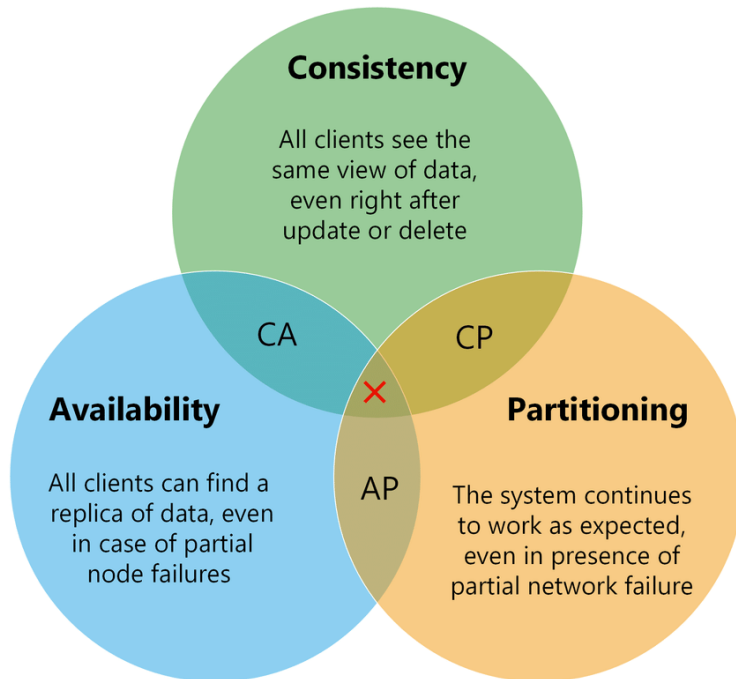


NoSQL Databases are mainly categorized into four types: **Key-value pair, Column-oriented, Graph-based and Document-oriented.** Every category has its unique attributes and limitations. None of the above-specified database is better to solve all the problems. Users should select the database based on their product needs.

##### 3. What is NoSQL vs SQL?

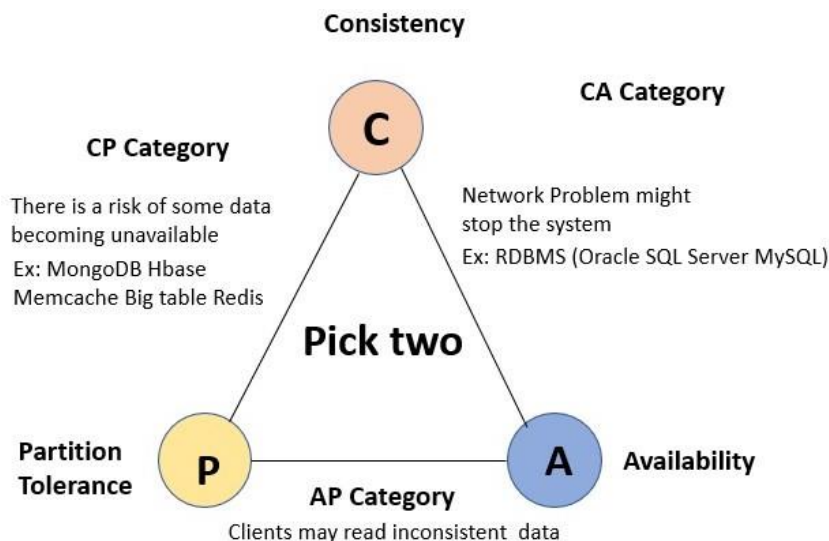
SQL is the programming language used to interface with relational databases. (Relational databases model data as records in rows and tables with logical links between them). **NoSQL is a class of DBMs that are non-relational and generally do not use SQL.**

#### 4. What is CAP theorem?



The CAP theorem is a belief from theoretical computer science about distributed data stores that claims, in the event of a network failure on a distributed database, it is possible to provide either consistency or availability—but not both.

#### 5. What is CAP in Brewer's CAP theorem?



CAP stands for **Consistency, Availability and Partition tolerance**. It wants system designers to make a choice between above three competing guarantees in final design. It's said that achieving all 3 in system is not possible, and you **MUST** choose at most two out of three guarantees in your system. OK.

#### 6. What are MongoDB CRUD operations?

The basic methods of interacting with a MongoDB server are called CRUD operations. CRUD stands for **Create, Read, Update, and Delete**. These CRUD methods are the primary ways you will manage the data in your databases.

## 7. What are MongoDB operations?

Edpresso Team. MongoDB is a document-oriented database program widely classified as a NoSQL database program. In MongoDB, the CRUD operation refers to the **creating, reading, updating, and deleting documents**.

## 8. What is CRUD function?

CRUD Meaning: CRUD is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: **create, read, update and delete**.

## 9. Which is used to perform CRUD operations on a table?

In CRUD operations, 'R' is an acronym for read, which means retrieving or fetching the data from the SQL table. So, we will use the **SELECT command** to fetch the inserted records from the SQL table. We can retrieve all the records from a table using an asterisk (\*) in a SELECT query.

## 10. What is CRUD form?

CRUD is an acronym that stands for **Create, Read, Update, and Delete**. These are the four most basic operations that can be performed with most traditional database systems and they are the backbone for interacting with any database.

## 11. Which language is used in MongoDB?

MongoDB uses the **MongoDB Query Language (MQL)**, designed for easy use by developers. The documentation compares MQL and SQL syntax for common database operations.

## 12. What is HBase data model?

The diagram illustrates the HBase data model as a table with columns grouped into two categories: personal data and professional data. The table has a 'Row key' column and four data columns: 'name', 'city', 'designation', and 'salary'. The 'personal data' group includes 'name' and 'city', while the 'professional data' group includes 'designation' and 'salary'.

Row key	personal data		professional data	
empid	name	city	designation	salary
1	raj	hydabad	manager	50,000
2	navi	chennai	stranger	30,000
3	rajesh	delhi	jr.engineer	25,000

HBase is a **distributed column-oriented database built on** top of the Hadoop file system. It is an open-source project and is horizontally scalable. HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data.

## 13. What are the main CRUD operations in HBase?

### Hbase CRUD Operations

- Create.
- Read.
- Update.
- Delete.

## 14. How do I create a HBase table?

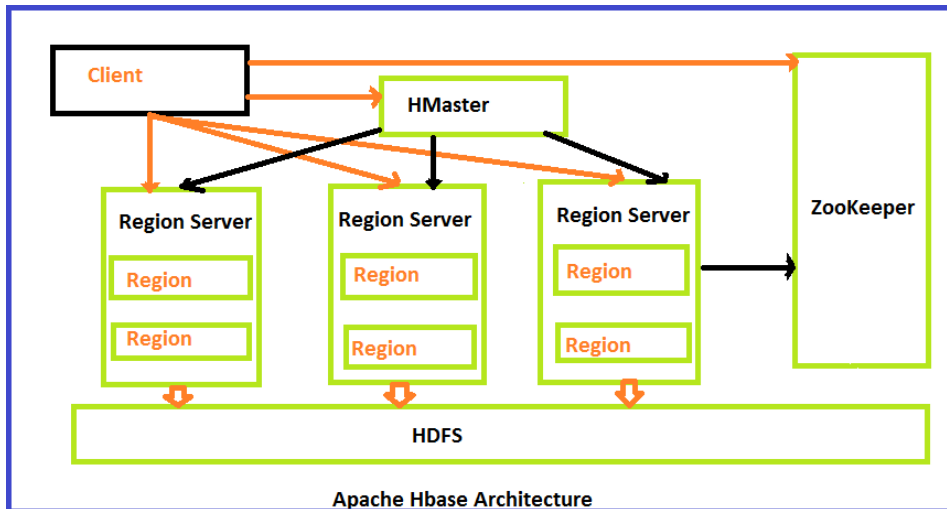
**Given below are the steps to create a table in HBase using java API.**

**Step 1:** Instantiate HBaseAdmin. This class requires the Configuration object as a parameter, therefore initially instantiate the Configuration class and pass this instance to HBaseAdmin. ...

**Step 2:** Create TableDescriptor. ...

**Step 3:** Execute through Admin.

### 15. What are the key components of HBase?



HBase architecture has 3 main components: **HMaster**, **Region Server**, **Zookeeper**. The implementation of Master Server in HBase is HMaster. It is a process in which regions are assigned to region server as well as DDL (create, delete table) operations.

### PART -B

1. Explain about CAP theorem?
2. Write about three criteria of CAP theorem?
3. Explain the properties of CAP theorem?
4. Write a note on MongoDB CRUD operations?
5. How do you perform CRUD operations in MongoDB compass?
6. Explain the main CRUD operations in HBase?
7. What is HBase explain HBase architecture?
8. What is HBase list out and explain the basic concepts of HBase in detail?
9. Which server handles read and write requests in HBase?
10. What are the data modeling explain the concepts used in MongoDB?