

DATABASE MANAGEMENT SYSTEM

Introduction	2
Data Models & Architecture Of DBMS	14
ER Model	26
SQL	54
Relational Model & Relational Database Design	80
Normalization	104
File Organization & Query Optimization	131
Indexing	137
Transaction Management	148

NOTE:

MAKAUT course structure and syllabus of 4th semester has been changed from 2022. Few topics have been changed and redesigned in **DATABASE MANAGEMENT SYSTEM**. Taking special care of this matter we are providing chapterwise relevant MAKAUT university solutions and some model questions & answers for newly introduced topics, so that students can get an idea about university questions patterns.

INTRODUCTION

Multiple Choice Type Questions

1. DBA is a
 a) software b) hardware c) person d) others [WBUT 2003, 2012]
 Answer: (c)
2. The distinguishable parts of a record are called
 a) data b) files c) fields d) entity [WBUT 2006, 2007, 2015]
 Answer: (b)
3. Which of the following is not a relational database?
 a) Oracle b) RDBMS c) YGL d) DBMS IV [WBUT 2006, 2011]
 Answer: (c)
4. A database management system
 a) allows simultaneous access to multiple files
 b) allows simultaneous access to multiple records of database
 c) both (a) and (b)
 d) none of (a) and (b) [WBUT 2006, 2011]
 Answer: (d)
5. The values of the attributes describe a particular
 a) Attribute b) Entity c) Instance d) None of these [WBUT 2007, 2012]
 Answer: (c)
6. System catalog is a system created database that describes
 a) database objects b) data dictionary information
 c) user access information d) all of these [WBUT 2008, 2012]
 Answer: (d)
7. What separates the physicals of data storage from the logical aspects of data representation? [WBUT 2008]
 a) Data b) Schema c) Constraints d) Relationship
- Answer: (a)
8. What is a set of possible data values called? [WBUT 2008]
 a) Degree b) Domain c) Field d) Cardinality
- Answer: (b)
9. Name the buffer in which all the commands that are type are stored. [WBUT 2008]
 a) View b) Edit c) Window d) Tristate
- Answer: (b)

10. Which of the following levels of abstraction involves the views of data?
 a) External level b) Conceptual level c) Physical level d) none of these [WBUT 2009]
 Answer: (a)
11. The ability to modify the internal schema without causing any change to the external schema is
 a) Physical data independence b) Logical data independence
 c) External data independence d) none of these [WBUT 2010]
 Answer: (a)
12. Which of the following is not one of the four categories described in the data dictionary?
 a) Data structure b) Data store c) Process d) Data flow [WBUT 2010]
 Answer: (a)
13. A person who has central control over the system is called a
 a) data analyst b) data selector
 c) database administrator d) none of these [WBUT 2010]
 Answer: (c)
14. Log table is used for
 a) updating b) recovery c) query d) deadlock [WBUT 2012]
 Answer: (b)
15. Generalization is a approach.
 a) bottom up b) top down
 c) both (a) and (b) d) None of these [WBUT 2013, 2017]
 Answer: (a)
16. Which level of Abstraction describes how data are stored in the database?
 a) Physical level b) View level c) Abstraction level d) Logical level [WBUT 2015, 2016]
 Answer: (a)
17. The collection of information stored in a database at a particular moment is called as
 a) Schema b) instance of the database
 c) Data domain d) independence [WBUT 2017]
 Answer: (b)
18. Identify the correct statement:
 a) Physical level Abstraction : Describes how a record is stored.
 b) Physical level Abstraction : Describe how schema is stored in a data base.
 c) Physical level Abstraction : Hides details of data types.
 d) None of the above [WBUT 2018]
 Answer: (a)

19. The entity set person is classified as student and employee. This process is called _____ [WBUT 2018]

- a) Generalization
- b) Specialization
- c) Inheritance
- d) Constraint generalization

Answer: (b)

Short Answer Type Questions

1. Explain the difference between meta data and data dictionary. [WBUT 2007]

Answer:

Metadata is defined as data providing information about one or more other pieces of data, such as:

- Means of creation of the data
- Purpose of the data
- Time and date of creation
- Creator or author of data
- Placement on a computer network where the data was created
- Standards used

Data dictionary is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format.

A data dictionary is organized into five sections:

- Data elements
- Data Flows
- Data Stores
- Processes
- External Entities.

2. Explain data types of a DBMS. [WBUT 2009]

Answer:

Data Types

SL. No.	Data Type	Description
1	DATE	Used to represent date and Time.
2	Number (L, D)	Can be integer or floating point. Magnitude up to 38 digits. L indicates maximum length of the data. D determines the number of places to the right of the decimal.
3	Long	Used to store variable length character string of size 2 GB. Long values cannot be indexed. String functions can not be applied on Long data types.
4	RAW/Long Raw	Is used to store binary data like picture. Raw data type can be 255 bytes long. Long Raw can be up to 2GB. Indexing can not be done.

3. What do you mean by DBMS? How many types of DBMS are there? [WBUT 2011]

Answer:

A database is a collection of information that is organised so that it can easily be accessed, managed, and updated.

Types of DBMS: Based on the structure of database management system (DBMS) DBMS, it can be classified in three types,

Hierarchical DBMS: Data items are organized in hierarchical manner.

Network DBMS: Many-to-many relationship appears in the form of network.

Relational DBMS: Data items are organized in the form of a table

4. What are the differences between Traditional File Management System and Database Management System? [WBUT 2011]

Answer:

Traditional File Management System

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity and independence from applications programs
- Improved data access to users through use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs
- Facilitated development of new applications program

In DBMS:

1. Problem Associated with centralized system
2. Cost of software, hardware and migration
3. Complexity of backup and recovery

5. What is information? How does it differ from data?

[WBUT 2011]

Answer:

Data are a set of collected numbers, words, anything. They do not mean anything until they are organised, arranged or developed. Once that happens (after they have been processed), information is obtained.

Information actually makes sense and is expressed through some sort of comprehensible logic. It must be recognized that data and information are required at the three levels, policy formulation, formulation of management plans, and the determination of management actions to implement the policy and plans.

The collection of data and information is essential for decision-making. It is important to ensure that the collected data are analyzed correctly, disseminated to where they can best be used, and used appropriately in decision-making.

6. Describe any two limitations of file system.

Answer:

1. **Data redundancy and inconsistency:** The files and application programs are created by different programmers. The various files have different formats and the programs may be written in different programming languages. Moreover, the same information may be BC DB-5

duplicated in several files. The redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency; i.e., the various copies of the same data may no longer agree. For example, a changed customer address may be reflected in savings-account records but not elsewhere in the system.

2. Difficulty in accessing data: Suppose that one of the bank officers needs to find out the names of all customers who live within Kolkata city having Pin-Code 700001. This request was not anticipated when the original system was designed and there is no application program on hand to meet it. Suppose that such a program is written, and that, several days later, the same officer needs to trim that list to include only those customers who have an account balance of Rs. 10,000/- or more. As expected, a program to generate such a list does not exist.

7. Indicate disadvantages of DBMS. Write down the application of DBMS.

[WBUT 2011]

Answer:

Disadvantages of DBMS:

- Problem Associated with centralized system
- Cost of software, hardware and migration
- Complexity of backup and recovery

There are many different types of DBMSs ranging from small systems that run on personal computers to huge systems that run on mainframes. The following are some of the examples of database applications:

- Payroll Accounting
- Integrated Inventory Management
- ERP HR Systems
- ERP Financials
- Computerized library systems
- Automated teller machines
- Flight reservation systems
- Computerized parts inventory systems
- Supply chain Management Systems.

8. What is data abstraction?

[WBUT 2015]

OR,

Explain the different levels of abstraction of the data base management system.

[WBUT 2017]

Answer:

Data Abstraction:

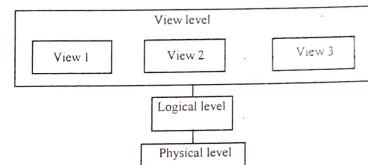


Fig: The three levels of data abstraction

Many database-systems users are not computers trained, so developers hide the complexity from users through several levels of abstraction, to simplify users interactions with the system:

Physical level: The lowest level of abstraction describes how the data are actually stored. At the physical level, complex low-level data structures are described in detail. **Logical level:** The next-higher level of abstraction describes what data are stored in the database, and what relationship exists among those data. The entire database is thus described in terms of a small number of relatively simple structures. Although implementation of the simple structures at the logical level may involve complex physical level structures, the user of the logical level does not need to be aware of this complexity. The logical level of abstraction is used by database administrators, who must decide what information is to be kept in the database.

View level: The highest level of abstraction describes only part of the entire database. Despite the use of simpler structures at the logical level, some complexity remains, because of the large size of the database. All users of the database system are not concerned with every information present in the system. Instead a common user is interested about some particular set of attributes of the data in the database to access. Accordingly their interaction with the system database can be simplified by defining the view level abstraction. The system may provide many views for the same database as per user requirement.

9. What is the difference between a database and a table?

[MODEL QUESTION]

Answer:

Database is a Pool of data. Tables are the containers of holding the data. As per relational database management system, tables are organized following the rules of total participation and/or partial participation. Implementation is done with the help of primary Key and foreign key relationship.

10. Describe the basic components of DBMS.

[MODEL QUESTION]

The major software modules or components of DBMS are as follows:

(i) Query processor: The query processor transforms user queries into a series of low level instructions. It is used to interpret the online user's query and convert it into an

efficient series of operations in a form capable of being sent to the run time data manager for execution. The query processor uses the data dictionary to find the structure of the relevant portion of the database and uses this information in modifying the query and preparing an optimal plan to access the database.

(ii) Run time database manager: Run time database manager is the central software component of the DBMS, which interfaces with user-submitted application programs and queries. It handles database access at run time. It converts operations in user's queries coming directly via the query processor or indirectly via an application program from the user's logical view to a physical file system. It accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the user's request. It enforces constraints to maintain the consistency and integrity of the data, as well as its security. It also performs backing and recovery operations.

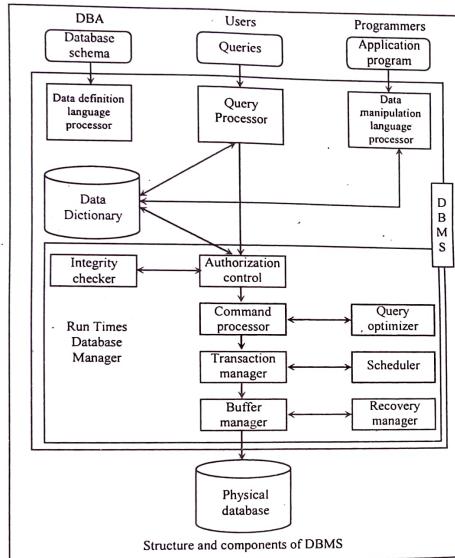
Run time database manager is sometimes referred to as the database control system and has the following components:

• Authorization control:

The authorization control module checks the authorization of users in terms of various privileges to users.

• Command processor:

The command processor processes the queries passed by authorization control module.



Integrity checker: It checks the integrity constraints so that only valid data can be entered into the database.

Query optimizer: The query optimizers determine an optimal strategy for the query execution.

- Transaction manager:** The transaction manager ensures that the transaction properties should be maintained by the system.

- Scheduler:** It provides an environment in which multiple users can work on same piece of data at the same time in other words it supports concurrency.

(iii) Data Manager: The data manager is responsible for the actual handling of data in the database. It provides recovery to the system which that system should be able to recover the data after some failure. It includes Recovery manager and Buffer manager. The buffer manager is responsible for the transfer of data between the main memory and secondary storage (such as disk or tape). It is also referred as the cache manager.

11. What are the different types of database end users? Discuss the main activities of each. [MODEL QUESTION]

Answer:

End users are the users, who use the applications developed. End users need not know about the working, database design, the access mechanism etc. They just use the system to get their task done. End users are of two types:

- Direct users
- Indirect users

a) Direct users: Direct users are the users who use the computer, database system directly, by following instructions provided in the user interface. They interact using the application programs already developed, for getting the desired result. E.g. People at railway reservation counters, who directly interact with database.

b) Indirect users: Indirect users are those users, who desire benefit from the work of DBMS indirectly. They use the outputs generated by the programs, for decision making or any other purpose. They are just concerned with the output and are not bothered about the programming part.

12. What are the significant features and objectives of DBMS? [MODEL QUESTION]

Answer:

Database management system (DBMS) consists of a collection of interrelated data and a set of programs to access that data. The collection data is usually referred to as data base which contains information about one particular enterprise.

- It is a mechanism for organizing, structuring and storing data.
- Data redundancy is reduced.
- Reducing physical storage requirements by separating the logical and physical aspects of the data base
- Damaged database can be recovered subject to pre-conditions
- Providing security to the user's data. Access is limited to authorized users.

Long Answer Type Questions**1. What is 'Data Dictionary'?****Answer:**

A data dictionary, or metadata repository is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format. A data dictionary is organized into five sections:

- Data elements
- Data Flows
- Data Stores
- Processes
- External Entities.

The format of the data dictionary contains:

- **Data Type:** Data Element/Data Flow/Data Store
- **Data Name:** Name of the Data Elements, Data Flow/Data Store.
- **Data Aliases:** Alternate names used for the convenience of multiple users.
- **Data Description:** A short description of data.
- **Data characteristics:** Frequency of the use. Data length, Range of data values etc.
- **Data Composition:** Various data elements contained in a data store of data flow

2. a) Describe briefly the role of DBA in the base design.

[WBUT 2006, 2010]

OR,

What are the functions of a Database Administrator?

[WBUT 2011, 2012]

b) What is data model? Describe different types of data model.

OR,

[WBUT 2006]

Write note on 'Data model'.**Answer:**

a) The functions of the DBA include the following:

(1) **Schema definition:** The DBA creates the original database schema by writing set of definitions that is translated by the DDL compiler to a set of tables that is stored permanently in the data dictionary.

(2) **Storage structure and access-method definition:** The DBA creates appropriate storage structures and access methods by writing a set of definitions, which is translated by the data-storage and data-definition-language compiler.

(3) **Schema and physical-organization modification:** Programmers accomplish the relatively rare modifications either to the database schema or to the description of the physical storage organization by writing a set of definitions which is used by either the DDL compiler or the data-storage and data-definition-language compiler to generate modifications to the appropriate internal system tables (for example, the data dictionary).

(4) **Granting of authorization for data access:** The granting of different types of authorization allows the database administrator to regulate which parts of the database various users can access. The authorization information is kept in special system

structure that is consulted by the database system whenever access to the data is attempted in the system.

(5) **Integrity-constraint specification:** The data values stored in the database must satisfy certain consistency constraints. Such a constraint must be specified explicitly by the database administrator. The integrity constraints are kept in a special system structure that is consulted by the database system whenever an update takes place in the system.

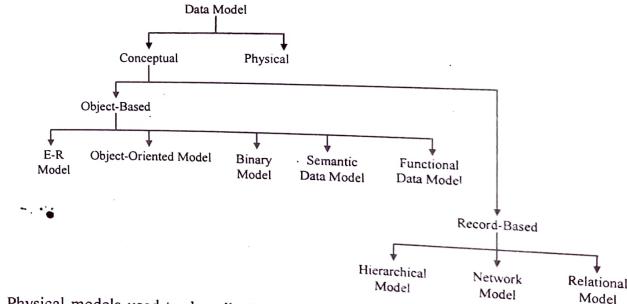
(6) **Designing security and integrity constraint specification:** The data values stored in the database must satisfy certain consistency constraints. Such a constraint must be specified explicitly by the database administrator. The integrity constraints are kept in a special system structure that is consulted by the database system whenever an update takes place in the system.

(7) **Routine Maintenance:** Periodically backing up the data either onto some movable devices and/or to remote servers to prevent and protect data from loss due to various reasons. Also ensures free disk space for regular operations. Monitoring jobs running on the database and ensuring that performance is not degraded by the method loosely called performance tuning.

b) Data Model

Data models describe the structure of a database. Data models are either conceptual data model or physical data model. The conceptual data model are of two categories namely, Object-Based Logical Models and record based models. Data models describe the structure of a database. Data models are either conceptual data model or physical data model. The conceptual data model can have two important categories.

- Object-based Logical Model
- Record-based Model.



Physical models used to describe higher level description of the storage structure of the database and their access mechanism. With the physical models it is possible to implement the data base at the system level. Two of these well-known models are the unifying model and the frame memory model.

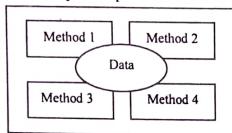
This model describes data at the conceptual and view levels. Some of them are:

- Entity-Relationship (E-R) model.
- Object-Oriented model.

Object oriented model

Used to specify the overall logical structure of the database. The data integrity constraints can't be specified explicitly with these models. The three widely accepted data models are:

1. Hierarchical Model
2. Network Model
3. Relational Model



These models are called record-based due to its fixed-format. It can contain varieties of record types with a set of attributes of fixed size. Separate languages are available to store and retrieve records in the respective data models.

3. Differentiate between various levels of data abstraction. List any two significant differences between a file processing system and a DBMS. [WBUT 2013]

Answer:

1st Part: Refer to Question No. 8 of Short Answer Type Questions.

2nd Part:

(i) Difficulty in accessing data: Suppose that one of the bank officers needs to find out the names of all customers who live within the city's 700035 pin code. This request was not anticipated when the original system was designed and there is no application program on hand to meet it. Suppose that such a program is written, and that, several days later, the same officer needs to trim that list to include only those customers who have an account balance of Rs 10,000/- or more. As expected, a program to generate such a list does not exist.

(ii) Integrity problems: The data values stored in the database must satisfy certain types of consistency constraints. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

4. What is Metadata? Explain with the help of a example.

[WBUT 2016]

Answer:

This Metadata is used for better access to the enormous amounts of data stored and managed by different companies. Metadata provides context for data. In data processing, for example, Metadata is definitional, it gives documentation of other data in the application or environment. The term "Metadata" should be used carefully since all data is about something and hence is "Metadata".

In Databases, Metadata defines data elements and attributes (Name, data type, size, etc), data could be registered about structures and records as well (Length, columns and fields). This is extremely helpful for the reliability of databases and their efficiency. In a library, for example, the data is the content of the titles, and the Metadata is about the title, the author, a description of the content, the physical location and the date of publication. In addition, it protects investment in data, helps the user in understanding data, enables discovery options, and limits Liability. All of these reasons, make Metadata the backbone in understanding a DBMS.

Examples

Examples of Metadata include; Photograph:

- Date and time
- Camera Settings Like:
- Focal Length, aperture, exposure

Web Pages:

- Descriptive Text
- Dates
- Keywords(1)

5. Write short notes on the following:

- a) Advantages of RDBMS
- b) Data Abstraction
- c) Role of DBA in database design

OR,

Functions of DBA

Answer:

a) Advantages of RDBMS:

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity and independence from applications programs
- Improved data access to users through use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs
- Facilitated development of new applications program

b) Data Abstraction: Refer to Question No. 8 of Short Answer Type Questions.

c) Role of DBA in database design:

Refer to Question No. 2(a) of Long Answer Type Questions.

[WBUT 2008]

[WBUT 2008, 2010]

[WBUT 2015]

[WBUT 2018]

DATA MODELS & ARCHITECTURE OF DBMS

Multiple Choice Type Questions

1. A relationship between the instances of a single entity type is called [WBUT 2008]
 a) binary relationship b) ternary relationship
 c) recursive relations d) none of these

- Answer: (c)
2. Overall logical structure of a database can be expressed graphically by [WBUT 2009]
 a) ER diagram b) records c) relations d) hierarchy

- Answer: (a)
3. Association among several entities is known as [WBUT 2010, 2014]
 a) attribute b) relationship c) field
 d) none of these

- Answer: (b)
4. In ER model symbol is used for [WBUT 2010]
 a) attribute b) entity c) relation
 d) none of these
5. Cardinality of a table is equal to the [WBUT 2012]
 a) number of rows in the table
 b) number of columns in the table
 c) number of domains of the table
 d) none of these

- Answer: (a)
6. The degree of a relationship describes [WBUT 2013]
 a) the number of attributes attached to a relation
 b) the number of entities attached to a relation
 c) the number of relations used to connect the entities
 d) None of these

- Answer: (b)
7. The set of permitted values for each attribute is called its [WBUT 2014]
 a) attribute set b) attribute range c) domain
 d) group

- Answer: (c)
8. What is the cardinality of a table with 100 rows and 10 columns? [WBUT 2014, 2017]
 a) 1000 b) 100 c) 10
 d) 10000

- Answer: (b)
9. The descriptive property possessed by each entity set is [WBUT 2018]
 a) entity b) relation c) model
 d) attribute
- Answer: (d)

Short Answer Type Questions

1. a) Describe the three – schema architecture. [WBUT 2003, 2004, 2008, 2010, 2012, 2014]
 OR,

- What do you mean by architecture of DBMS? Explain with diagram. [WBUT 2006]
 b) What is the difference between logical data independence and physical data [WBUT 2003, 2006, 2007, 2010, 2013]
 independence?

- OR,
 Briefly explain the terms: 'logical data independence' and 'physical data [WBUT 2005]
 independence'. OR,

- Distinguish between logical and physical data independence. [WBUT 2018]

Answer:

- a) Three-schema architecture of DBMS:
 Individual user view defined in consultation with DBA

External Level

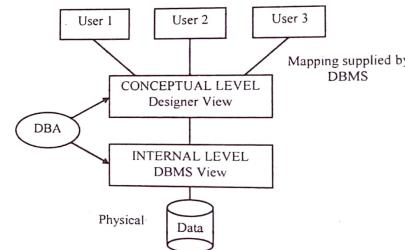


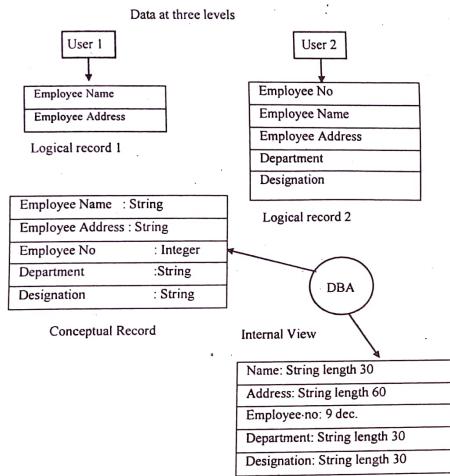
Fig: The three levels of Architecture

The Three Levels of the Architecture

We are now in a position to introduce an architecture of a database system. The aim is to present the architecture to provide a framework on which we can build our concepts on DBMS.

- The **internal level** is the one closest to physical storage—i.e., it is the one concerned with the way the data is physically stored on a particular media in a specific format.
- The **conceptual level** is a "level of indirection" between the other two.(Gives abstraction on to the upper level of the lower level)
- The **external level** is the one closest to the users i.e., it is the one concerned with the way the data is viewed by individual users.

If the external level is concerned with *individual* user views, then the conceptual level is concerned with a *community* user view. In other words, there will be many distinct external views, each consisting of a more or less abstract representation of some portion of the total database, and there will be precisely one conceptual view, consisting of a similarly abstract representation of the database in its entirety. There will be precisely one internal view, representing the total database as physically stored.



- **The internal level** is the one closest to physical storage—i.e., it is the one concerned with the way the data is physically stored.
- **The conceptual level** is a "level of indirection" between the other two.(Gives abstraction on to the upper level of the lower level)
- **The external level** is the one closest to the users i.e., it is the one concerned with the way the data is viewed by individual users.

If the external level is concerned with *individual* user views, then the conceptual level is concerned with a *community* user view. In other words, there will be many distinct external views, each consisting of a more or less abstract representation of some portion of the total database, and there will be precisely one conceptual view, consisting of a similarly abstract representation of the database in its entirety. There will be precisely one internal view, representing the total database as physically stored.

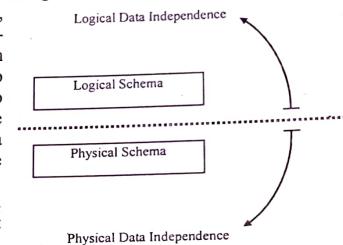
b) Refer to Question No. 2 of Short Answer Type Questions.

2. What do you mean by database independence?

Answer:

There is a lot of data in whole database management system other than user's data. DBMS comprises of three kinds of schemas, which is in turn data about data (Meta-Data). Meta-Data is also stored along with database, which once stored is then hard to modify. But as DBMS expands, it needs to be changed over the time satisfy the requirements of users. But if the whole data were highly dependent it would become tedious and highly complex.

Data about data itself is divided in layered architecture so that when we change data at one layer it does not affect the data layered at different level. This data is independent but mapped on each other.



Physical data independence: Physical data independence deals with hiding the details of the storage structure from user applications. The application should not be involved with these issues, since there is no difference in the operation carried out against the data. The ability to change the physical schema without changing the logical schema is called physical data independence. For example, a change to the internal schema, such as using different file organization or storage structures, storage devices, or indexing strategy, should be possible without having to change the conceptual or external schemas.

Logical data independence: The ability to change the logical (conceptual) schema without changing the External schema (User View) is called logical data independence. For example, the addition or removal of new entities, attributes, or relationships to the conceptual schema should be possible without having to change existing external schemas or having to rewrite existing application programs. The external schema may stay unchanged for most changes of the logical scheme. This is especially desirable as the application software does not need to be modified or newly translated.

Integrity independence: Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.

Distribution independence: The distribution of portions of the database to various locations should be invisible to users of the database. Existing applications should continue to operate successfully.

1. when a distributed version of the DBMS is first introduced; and
2. when existing distributed data are redistributed around the system.

3. Explain the difference between external, internal and conceptual schemas.

[WBUT 2007, 2010, 2012, 2014, 2018]

Answer:

Refer to Question No. 1 of Short Answer Type Questions.

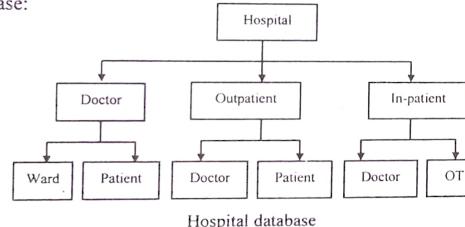
4. Explain hierarchical data model with suitable example. [WBUT 2008, 2010]
OR,

State the properties of Hierarchical Model.

Answer: [WBUT 2009]

Hierarchical Model

Hierarchical data model is record based and related to each other in a hierarchy of many to one relationship. The hierarchical database consists of a collection of hierarchical trees. The trees described could be physically stored trees or logical trees derived from the physically stored trees. The ordering of the tree is according to the hierarchical structure diagram. A many-to-many relationship between record types cannot be expressed directly in the hierarchical model. Such many to many relationship can be expressed by using data replication or virtual records. For example, the hierarchical representation of the hospital database:



The structure allows repeating information using parent/child relationship. Each parent can have many children but each child has one parent. Here, doctor is a parent, the ward and patient are the children. Similarly, Doctor, outpatient and In-patient are children to the parent entity Hospital and so on. One limitation is its ability to handle redundancy in Data which may invite huge disk storage and slow processing. In chapter-v "relational database design" we discuss the issues in detail.

5. Describe Record based data model.

[WBUT 2011]

Answer:

Record-Based Data Model:

It specifies the overall logical structure of the database. The data integrity constraints can't be specified explicitly with these models. The three widely accepted data models are:

1. Hierarchical Model
2. Network Model
3. Relational Model

These models are called record-based due to its fixed-format. It can contain varieties of record types with a set of attributes of fixed size. Separate languages are available to store and retrieve records in the respective data models.

6. What are the different types of Data Models?

[MODEL QUESTION]

Answer:

Data models describe the structure of a database. Data models are either conceptual data model or physical data model. The conceptual model is used to overview the application. The E-R model is a kind of conceptual graphical model used at the data base design level. Physical model operates at the lowest level of abstraction. It describes the storage structure of the database. Following diagram illustrates the model view.

7. What are the functions of Database Manager and DML pre-compiler?

[MODEL QUESTION]

Answer:

Database Manager (DM) is a component of DBMS. The major functions of DM are:

- Authorization control
- Command processor
- Integrity checker
- Query optimizer
- Transaction manager
- Scheduler
- Recovery manager
- Buffer manager

Data Manipulation Language (DML)

1. Provides basic data manipulation operations on data held in the database.
2. Procedural DML - allows user to tell system exactly how to manipulate data.
3. Non-Procedural DML - allows user to state what data is needed rather than how it is to be retrieved.

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

DML preprocessor: This module converts DML statements embedded in an application program into standard function calls in the host language. The DML preprocessor must interact with the query processor to generate the appropriate code.

8. What is mapping constraint? How does it influence the design of database?

[MODEL QUESTION]

Answer:

Mapping constraints can be explained in terms of mapping cardinality. In DBMS cardinality term is used at two different instances with two different meanings as well.

1. **In Context of Data Models:** Cardinality refers to the relationship between two tables. Relationship can be of three types one-to-one, one-to-many, and many-to-many.

2. In **In Context of Query Optimization:** In terms of query, the cardinality refers to the uniqueness of a column in a table. The column with all unique values would be having the high cardinality and the column with all duplicate values would be having the low cardinality. These cardinality scores helps in query optimization.

For example; we can have these constraints in place while creating tables that influence the design of database.

```
CREATE TABLE Customer (
    customer_id int PRIMARY KEY NOT NULL,
    first_name varchar(20),
    last_name varchar(20)
);
CREATE TABLE Order (
    order_id int PRIMARY KEY NOT NULL,
    customer_id int,
    order_details varchar(50),
    constraint fk_Customers foreign key (customer_id)
        references dbo.Customer
);
```

It is assumed that a customer orders more than once, the above relation represents **one to many** relation. Similarly we can achieve other mapping constraints based on the requirements.

9. List the types of privileges used in database access control.

[MODEL QUESTION]

Answer:

An access control constraints what a user can do directly. Access control seeks to prevent activity that could lead to a breach of security. Privileges are granted to users at the discretion of other users--hence the term discretionary access control.

There are two distinct categories of privileges:

- **System privileges:** A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. There are over 60 distinct system privileges. However, in general, grant system privileges only to administrative personnel and application developers. End users normally do not require the associated capabilities.
- **Schema object privileges:** A **schema object privilege** is a privilege or right to perform a particular action on a specific schema object:
 - Table
 - View
 - Sequence
 - Procedure
 - Function
 - Package

Long Answer Type Questions

1. a) Write down the comparison of three models in DBMS.

OR,

Differentiate between hierarchical, network and relational model. [WBUT 2017]

- b) What is schema of DBMS? What are the types of schema? Explain them. What is sub-schema? [WBUT 2009]

Answer:

a) **Hierarchical Model:** Hierarchical data model is record based and related to each other in a hierarchy of many to one relationship. The hierarchical database consists of a collection of hierarchical trees. The trees described could be physically stored trees or logical trees derived from the physically stored trees. The ordering of the tree is according to the hierarchical structure diagram. A many-to-many relationship between record types cannot be expressed directly in the hierarchical model. Such many to many relationship can be expressed by using data replication or virtual records.

Network Model: The network approach allows us to model many-to-many relationships more directly than the hierarchical model. The model is capable of handling redundant data more efficiently than in the hierarchical model. There are two basic data structures in the model: records and sets. Data is stored in records. Each record consists of a group of related data values. Records are classified into record types. Record types describe the structure of a group of records that store the same type of information. Each record type contains a name and format for each data item.

Relational Model

A relation has the following properties:

1. The columns of a table are of the same kind (columns are homogeneous)
2. Each item should contain simple value
3. All tuples of a relation must be distinct
4. The ordering of tuples within a relation depends on its key value
The columns of a relation are assigned distinct names and the ordering of this column is immaterial.

b) Schema:

The schema defines the tables, the attributes along with its size and type and the relationships between attributes and tables. Overall design of the database is called the database schema. Schemas are relatively fixed. However, changes in the schema can be made depending on the requirement. The different level of database schema are:

1. **Conceptual Schema:** A map of concepts and their relationship
2. **Logical Schema:** A map of entities and their attributes and relations
3. **Physical Schema:** A particular implementation of logical schema

At the lowest level is the physical schema; at the intermediate level is the logical schema; and at the highest level is a sub-schema. In general, database systems support one physical schema, one logical schema, and several sub-schemas.

Sub-Schema: The term sub-schema refers to an application programmer's view of portion of the database that is being used. Many different sub-schemas can be derived from one schema.

2. Write the features of network model. Compare network model and relational model.
[MODEL QUESTION]

Answer:

1st Part:

The network approach allows us to model many-to-many relationships more directly than the hierarchical model. The model is capable of handling redundant data more efficiently than in the hierarchical model. There are two basic data structures in the model: records and sets. Data is stored in records. Each record consists of a group of related data values. Records are classified into record types. Record types describe the structure of a group of records that store the same type of information. Each record type contains a name and format for each data item.

2nd Part:

Network Model	Relational Model
There are two basic data structures in the model: records and sets. Data is stored in records. Each record consists of a group of related data values.	There is no requirement of rigid structure on the data. The familiar structure of files and records are viewed as tables/relations and tuples/records respectively.
A set type is a description of a 1:N relationship between two record types. Each set type definition consists of three basic elements: 1. A name of the set type 2. An owner record type 3. A member record type	The columns of a table are of the same kind (columns are homogeneous). Each item should contain simple value. All tuples of a relation must be distinct.
In the database there will be many set occurrences or set instances corresponding to a set type. Set occurrences is composed of one owner record for the owner record type and a number of related member records. The stored representations of set instances are either linked lists or doubly linked list representation. For each set type an additional OWNER pointer is included in the member record type. The owner pointer points directly to the owner record of the set.	The ordering of tuples within a relation depends on its key value. The columns of a relation are assigned distinct names and the ordering of these column is immaterial. The columns of a relation are assigned distinct names and the ordering of these column is immaterial.

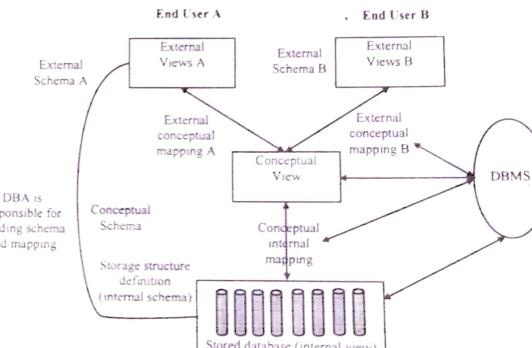
3. Why we need query three level architecture of a DBMS? Justify your answer with suitable example.
[MODEL QUESTION]

Answer:

The objective of the three level architecture is to separate each user's view of the database from the way the database is physically represented. There are several motives why this separation is desirable:

- Each user should be able to access the same data, but have a different customized view of the data. Each user should be able to change the way he or she views the data, and this change should not affect other users.
- Users should not have to deal directly with physical database storage details, such as indexing or hashing. In other words a user's interaction with the database should be independent of storage considerations.
- The Database Administrator (DBA) should be able to change the database storage structures without affecting the user's views.
- The internal structure of the database should be unaffected by changes to the physical aspects of storage, such as the changeover to a new storage device.
- The DBA should be able to change the conceptual structure of the database without affecting all users.

For example;



Three Level of Architecture of DBMS

In order to understand the difference between the three levels, consider again the database schema that describes College Database system. If User1 is a Library clerk, the external view would contain only the student and book information. If User2 is an account office clerk then he/she may be interested in students detail and fee detail. Shows specific information actually available at each level regarding a particular user. The external view would depend upon the user who is accessing the database. The conceptual level contain the logical view of the whole database, it represents the type of each required field. The internal view represents the physical location of each element on the disk of the servers well as how many bytes of storage each element needs.

4. Write short notes on the following:

- a) Three-level architecture of DBMS
- b) Logical and physical data independence
- c) Database Security
- d) Database language

Answer:

- a) Three-level architecture of DBMS:

Refer to Question No. 1(a) of Short Answer Type Questions.

- b) Logical and physical data independence:

Refer to Question No. 2 of Short Answer Type Questions.

- c) Database Security:

Database security usually refers to protection from malicious access, where integrity refers to the avoidance of accidental loss of consistency.

To protect the database, we must take security measures at several levels.

- (i) **Physical:** The site or sites containing the computer system must be physically secured against armed or surreptitious entry by intruders.
- (ii) **Human:** User must be authorized carefully to reduce the chance of any such user giving access to an intruder in exchange for a bribe or other favors.
- (iii) **Operating system:** No matter how secure the database system is, weakness in the operating system security may serve as a means of unauthorized access to the database.
- (iv) **Network:** Since almost all database system allow remote access through terminals or networks, software-level security within the network software is as important as physical security, both on the internet and in networks private to an enterprise.
- (v) **Database system:** Some database system users may be authorized to access only a limited portion of the database. Other users may be allowed to issue queries, but may be forbidden to modify the data. It is responsibility of the database system to ensure that these authorization restrictions are not violated.

- d) Database language:

The database system is an intermediate link between the physical database, the computer, the operating system, and the users. In order to provide the various facilities to different types of users, a DBMS provides one or more specialized programming languages called Database Languages. A language called Structured Query Language (SQL) has been taken as the standard database language. Depending on the level and degree of abstraction the language varies. Let us now introduce the concept of abstraction. The degree of abstraction of a database defined as follows:

If the database model is hardware and software dependent, we call the degree of abstraction is low. The hierarchical and network model falls in this category. If the database model is hardware-independent but software dependent the model may be identified as holding the medium level abstraction. Relational and object oriented models lie in this category, where as the model independent of both hardware and

[WBUT 2015, 2016]

[WBUT 2015, 2016]

[MODEL QUESTION]

[MODEL QUESTION]

software are identified as high degree of abstraction. ER models falls in this category. The three levels of abstractions of the database management system are external level which is actually the end-user view. The 2nd level is referred as conceptual level, which is actually the designers view and the 3rd level is the internal level, which is referred as DBMS view. To describe and manipulate the data in the database the two different components of SQL languages are used:

- Data Description Languages (DDL)
- Data Manipulation Languages (DML)

Data Definition Language (DDL)

A Data Definition Language (DDL) is used to describe the details of the data. The conceptual schema is specified by a set of definitions expressed by this special language. The DBMS will have a compiler whose function is to process DDL statements in order to identify description of the schema constructs and to store the schema descriptions as a set of tables in a special file called data dictionary. Thus, data dictionary is a file that contains data about data which is called metadata. This file is consulted before actual data are read or modified in the data base system.

Storage Definition Language: The SDL is used to specify the internal schema. The mapping between two languages can be specified by any one of the languages. In some DBMS, there is only one language which has both DDL and SDL capabilities.

Data Manipulation Language (DML)

Once the database schemas are compiled and the database is filled with data, the data manipulation needs to covers the following functions:

1. The retrieval of information stored in the database.
2. The insertion of new information into the database.
3. The deletion of information from the database.
4. The modification of information stored in the database.

The language that enables the users to access or manipulate data is called the data-manipulation language (DML). DML are of two types.

Procedural DML: which require users to specify what data are needed and how to get those data.

Nonprocedural DML: which require a user to specify what data are needed without specifying how to get it.

ER MODEL

Multiple Choice Type Questions

1. The data dictionary tells the DBMS

- a) what files are in the database
- b) what attributes are possessed by the data
- c) what these files contain
- d) all of these

Answer: (d)

2. To select a tuple from a relational database table, the symbol used in relational algebra is [WBUT 2007, 2011, 2012]

- a) pie
- b) lamda
- c) sigma
- d) gama

Answer: (c)

3. What Tuple relational calculus does is to [WBUT 2007, 2011]

- b) select entire table
- c) select domain variable
- d) Select tuple variables

Answer: (a)

4. Relational algebra is what type of language? [WBUT 2007]

- a) Non-procedural language
- b) Procedural language
- c) Data manipulation language
- d) Both (b) and (c)

Answer: (d)

5. The operation of a certain relation X, produces Y such that Y contains only selected attributes of X. Such an operation is [WBUT 2008, 2011, 2014]

- a) Projection
- b) Selection
- c) Union
- d) Difference

Answer: (a)

6. What is the smallest unit of data in relational model? [WBUT 2008]

- a) Data type
- b) Field
- c) Data value
- d) None of these

Answer: (b)

7. If a set of attributes K, in relation schema R1 is a foreign key for R1, then [WBUT 2008]

- a) every tuple of R1 has a distinct value of K
- b) K is a key for some other relation
- c) K cannot have a null value for tuples in R1
- d) K is a primary key for R1

Answer: (b)

8. The information about data in a database is called [WBUT 2009, 2016]

- a) Meta data
- b) Tera data
- c) Hyper data
- d) none of these

Answer: (a)

[WBUT 2007]

9. Which of the following features is supported in the relational database model?

- a) Complex data types
- b) Multi-valued attributes
- c) Associations with multiplicities
- d) Generalization relationships

Answer: (d)

[WBUT 2010]

10. Relational algebra is a

- a) procedural language
- b) non-procedural language
- c) object oriented language
- d) all of these

Answer: (a)

11. Any relation that is not part of the logical model, but is made visible to a user as a virtual relation, is called as [WBUT 2010, 2012, 2017]

- a) relation
- b) view
- c) tuple
- d) none of these

Answer: (b)

12. In relation algebra Π symbol is used for [WBUT 2010]

- a) selection
- b) union
- c) intersection
- d) projection

Answer: (d)

13. Relational algebra is a language. [WBUT 2013]

- a) non-procedural
- b) procedural
- c) programming
- d) none of these

Answer: (b)

14. An attribute of one table matching with the primary key of another table is called [WBUT 2014]

- a) foreign key
- b) secondary key
- c) candidate key
- d) surrogate key

Answer: (a)

15. If two relations have 5 and 10 rows respectively, then no. of tuples in Cartesian product will be [WBUT 2014]

- a) 50
- b) 5
- c) 10
- d) 15

Answer: (a)

16. A candidate key which is not a primary key is known as [WBUT 2014]

- a) super key
- b) alternate key
- c) foreign key
- d) non prime attribute

Answer: (b)

17. Which one is not a traditional set operator defined on relational algebra?

- a) Union
- b) Intersection
- c) Set Difference
- d) Join

Answer: (d)

18. Which operator performs pattern matching in SQL?

- a) Except
- b) Intersect
- c) Join
- d) Like

Answer: (d)

[WBUT 2014]

19. Referential integrity is directly related to

- a) relational key
b) foreign key
c) primary key
d) candidate key

[WBUT 2015, 2017]

20. To select a tuple from a relational database table, the symbol used in relational algebra is

- a) ρ (Row) b) σ (Sigma) c) Π (Project) d) none of these

Answer: (b)

21. If a set of attributes K, in a relation to schema R1 is a foreign key to R1 then [WBUT 2016]

- a) every tuple of R1 has a distinct value for K
b) K is key for some other relation
c) K cannot have a null value for tuples in R1
d) K is a primary key for R1

Answer: (b)

22. Which of the following features is supported in relational database model? [WBUT 2016]

- a) complex data types b) multi-valued attributes
c) associations with multiplicities d) generalization relationships

Answer: (c)

23. The entity integrity constraint states

- a) no primary key value can be null
b) a part of the key may be null
c) duplicate object values are allowed
d) none of these

Answer: (a)

24. Relational algebra is a language.

[WBUT 2017]

- a) non-procedural b) procedural
c) programming d) none of these

Answer: (b)

25. A relational database consists of a collection of

[WBUT 2018]

- a) tables b) fields c) records
d) keys

Answer: (a)

26. Suppose R is a relation of n attributes $\{A_1, A_2, \dots, A_n\}$ as a function of n. How many super keys R has if the only key is A_i ? [WBUT 2018]

- a) 2^n b) $2^{(n-1)}$ c) $2^{(n-1)}$
d) None of these

Answer: (c)

[WBUT 2015, 2017]

27. Consider the following SQL statements.

S1:

INSERT INTO employees (first_name, last_name, fname)
VALUES ('John', 'Capita', 'xcapit00');

S2:

SELECT instructor.ID, department.dept_name FROM instructor, department
WHERE instructor.dept_name = department.dept_name AND department.budget
> 95000;

Identify the correct statement related to S1 and S2:

- a) Both S1 and S2 are Data Definition (DDL) Queries.
b) S1 is a Data Control Query and S2 is a Data Manipulation (DML) Query
c) Both S1 and S2 are Data Manipulation (DML) Queries
d) S1 is a Data Definition (DDL) Query and S2 is a Data Control Query

Answer: (c)

28. To remove a relation from an SQL database, we use the command

[WBUT 2018]
a) delete b) purge c) remove d) drop

Answer: (d)

Short Answer Type Questions

1. Explain briefly about generalization and specialization.

[WBUT 2004, 2008, 2009, 2010]

OR,

Discuss the two main types of constraints on Specialization and Generalization.

[WBUT 2007]

Answer:

Generalization: The design process may also proceed in a bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features. The database designer may have first identified a customer entity set with the attributes name, street, city, and customer-id, and an employee entity set with the attributes name, street, city, employee-id, and salary. There are similarities between the customer entity set and the employee entity set in the sense that they have several attributes in common. This commonality can be expressed by generalization, which is a containment relationship that exists between a higher-level entity set and one or more lower-level entity sets. In our example, person is the higher-level entity set and customer and employee are lower-level entity sets.

Specialization: The graphical representation mechanism of an entity set having distinct subset of entities with shared set of attributes is called specialization. For e.g. consider an entity set person, with attributes name, employee-no, department, project-no. The person may be further classified as:

1. Project-member
2. Project-manager

Each of these person types is described by a set of attributes that includes all the attributes of entity set person plus some more attributes. The specialization of person allows us to distinguish among persons according to whether they are project-member or project manager.

2. a) What do you mean by degree of a relationship? [WBUT 2006, 2013]
 b) What do you mean by cardinality of relationship? [WBUT 2006, 2009, 2013]
 c) What is ternary relationship? [WBUT 2006, 2013]

Answer:

a) Degree of relationship refers to the number of participating entities in a relationship. If there are two entities involved in relationship then it is referred to as binary relationship. If there are three entities involved then it is called as ternary relationship and so on. On the other hand, it is the cardinality of relationship that defines the number of instances of one entity as it relates to the number of instances of the other entity. Based on the different combinations between two entities we can have either one-to-one, one-to-many or many-to-many relationship.

Three degrees are

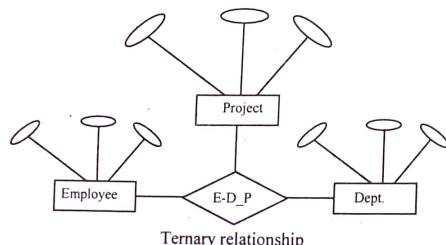
1. Unary;
2. Binary;
3. Ternary

b) Cardinality is the specification of the number of occurrences of one object that can be related to the number of occurrences of another object.

For example, one object can relate to only one other object (1:1 relationship); one object can relate to many objects (1:N relationship); Some number of occurrences of an object can relate to some other number of occurrences of another object (M:N relationship). Cardinality defines "the maximum number of objects that can participate in a relationship". However, it does not provide an indication of whether or not a particular data object must participate in the relationship. To specify this information, the data model adds modality to the object/ relationship pair.

c) Ternary relationship

The relationship defined by more than two entity sets can be represented as ternary relationship.



3. Discuss the entity integrity and referential integrity constraint. [WBUT 2007]

Answer:

Entity Integrity

In a relational database, entity integrity is a property that ensures that no records are duplicated and that no attributes that make up the primary key are NULL. It is one of the properties necessary to ensure the consistency of the database. Entity Integrity ensures that there are no duplicate records within the table and that the field that identifies each record within the table is unique and never null. The existence of the Primary Key is the core of the entity integrity. If one defines a primary key for each entity, they follow the entity integrity rule. Entity integrity specifies that the Primary Keys on every instance of an entity must be kept, must be unique and must have values other than NULL. Entity Integrity is the mechanism the system provides to maintain primary keys. The primary key serves as a unique identifier for rows in the table. The system enforces Entity Integrity by not allowing operations (INSERT, UPDATE) to produce an invalid primary key. Any operation that creates a duplicate primary key or one containing nulls is rejected.

Referential Integrity

It is a database management safeguard that ensures every foreign key matches a primary key. For example, customer numbers in a customer file are the primary keys, and customer numbers in the order file are the foreign keys. If a customer record is deleted, the order records must also be deleted; otherwise they are left without a primary reference. If the DBMS does not test for this, it must be programmed into the applications.

Referential integrity in a relational database is consistency between coupled tables. Referential integrity is usually enforced by the combination of a primary key(candidate key) and a foreign key. For referential integrity to hold, any field in a table that is declared a foreign key can contain only values from a parent table's primary key or a candidate key. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity. The RDBMS enforces referential integrity, normally either by deleting the foreign key rows as well to maintain integrity, or by returning an error and not performing the delete. Which method is used would be defined by the definition of the referential integrity constraint.

Example:

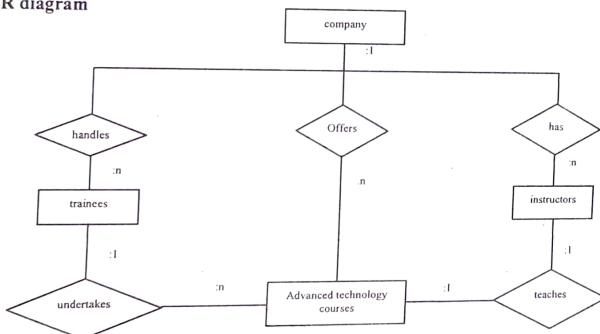
An employee database stores the department in which each employee works. The field "DepartmentNumber" in the Employee table is declared a foreign key, and it refers to the field "Index" in the Department table which is declared a primary key. Referential integrity would be broken by deleting a department from the Department table if those employees listed in the Employee table are listed as working for that department, unless those employees are moved to a different department at the same time.

POPULAR PUBLICATIONS

4. The IT Training Group (Kolkata) has contacted you to create a conceptual model by using the Entity-Relationship data model for a database that will meet the information needs for its training program. The Company Director gives you the following description of the training groups operating environment. The Company has twelve instructors and can handle up to one hundred trainees per training session. The Company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session. Given this information, draw an ER diagram for IT Training Group (Kolkata). Distinguish between single-valued vs multi-valued and stored vs Derived attributes.

Answer:

ER diagram



Last part:

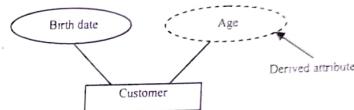
Single-valued: These attributes have a single value for a particular entity. For example, employment-number, age are single valued attribute of a person.

Multi-valued: The attribute has a multiple value for a particular entity. For example, a person may have multiple contact numbers, college degrees etc.

Stored Attributes: The attribute values which are stored against an entity are called stored attributes. For example, the date of birth of a person (dob) is a stored attribute.

Derived Attribute: The attribute values which are not a part of the entity but can be computed in association with some other stored attributes.

DATABASE MANAGEMENT SYSTEM



For example the age of a person which can be computed by subtracting the dob from the current date.

5. a) Draw a sample ERD for hospital management system.

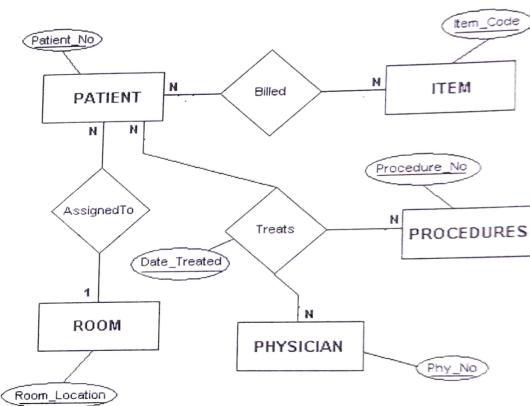
[WBUT 2007]

b) Explain Aggregation with reference to ER Model.

[WBUT 2007, 2009]

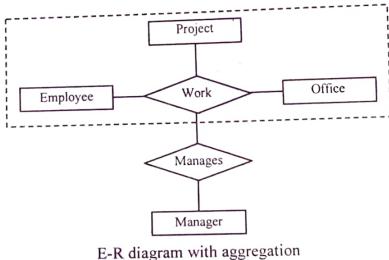
Answer:

a)



b) Aggregation:

An abstraction through which relationships are treated as higher-level entities is called aggregation. For example; the relationship set work relating the entity sets employee, office and project is considered as a higher-level entity set. Such an entity set is treated in the same manner as is any other entity set. Now we create a binary relationship manages between works and manager to represent who manages what tasks.



6. State the properties of Relational Model.

[WBUT 2008, 2010]

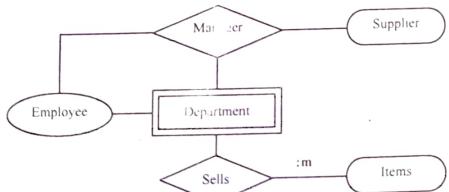
Answer:

1. Relation name is distinct from all other relations.
2. Each cell of relation contains exactly one atomic (single) value.
3. Each attribute has a distinct name.
4. Values of an attribute are all from the same domain.
5. Order of attributes has no significance.
6. Each tuple is distinct; there are no duplicate tuples.
7. Order of tuples has no significance, theoretically.

7. Consider E-R diagram for the database of a departmental store. There are various departments in the store. One department sells many items. Some items may be sold by more than one department. A department has many employees. An employee can belong to at most one department. A manager is an employee who may look after more than one department but a department may be looked after by only one manager. A unique member called internal_hern_no is assigned to every item by the store. A supplier may supply more than one item. Every item is supplied by only one supplier at a time. Make E-R diagram for the database mentioned indicating primary key and show one entity set as a weak entity set if possible.

[WBUT 2009]

Answer:



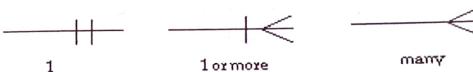
E-R Diagram of the Department Store

EMPLOYEE {Emp-ID, Emp-type, Name, department_No, skill, dob, Skill}
ITEMS {Internal_hern, description, uom, date-of-purchase, qty, rate}
DEPARTMENT {Department_no, Department_Name, Internal_hern, Rack_No}

8. Draw ER diagram showing the cardinality for the following problem:

- i) A bill is sent to a customer. A customer may receive many bills.
- ii) A clerk works in a bank. The bank has many clerks.
- iii) Students appear for seats in collage. Each student can get almost one seat. A college has many seats. A student can send many applications.

[WBUT 2010]



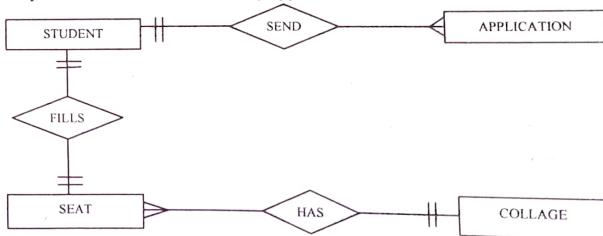
- i) A bill is sent to a customer. A customer may receive many bills.



- ii) A clerk works in a bank. The bank has many clerks.



- iii) Students appear for seats in collage. Each student can get almost one seat. A college has many seats. A student can send many applications.



9. What do you mean by strong and weak entities?

[WBUT 2011]

Answer:

Weak Entity Set:

When an entity set does not possess enough attributes to form a primary key is called weak entity set. A weak entity set can always be made into a strong entity set by adding to its attributes the primary-key attributes of its identifying entity set.

Strong Entity Set:

An entity set, which have sufficient attributes to form a primary key.

10. Let E1 and E2 be two entities in an E-R diagram with simple valued attributes. R1 and R2 are two relations between E1 and E2, where R1 is none-to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. What is the minimum number of tables required to represent the situation in the relational model? Give proper explanation for your answer. [WBUT 2016]

Answer:

Strong entities E1 and E2 are represented as separate tables.

In addition to that many-to-many relationships (R2) must be converted as separate table by having primary keys of E1 and E2 as foreign keys.

One-to-many relationship (R1) must be transferred to 'many' side table (i.e. E2) by having primary key of one side (E1) as foreign key (this way we need not to make a separate table for R1).

Let relation schema be E1(a1, a2) and E2(b1, b2).

Relation E1(a1 is the key)

a1 a2

1 3

2 4

3 4

Relation E2 (b1 is the key, a1 is the foreign key, hence R1(one-many) relationship set satisfy here)

b1 b2 a1

7 4 2

8 7 2

9 7 3

Relation R2 ({a1, b1} combined is the key here, representing many-many relationship R2)

a1 b1

1 7

1 8

2 9

3 9

Hence we will have minimum of 3 tables.

11. Tables:

Student			Department	
Roll(PK)	Name	Dept id (PK)	Dept id (PK)	Dept_Name
1	ABC	1	1	A
2	DEF	1	2	B
3	GHI	2	3	C
4	JKL	3		

What will happen if we try to execute the following two SQL statements?

Give proper explanation for your answer.

- a) Update Student set Dept id = Null where Roll no = 1;
- b) Update Department set Dept id = Null where Dept id = 1;

[WBUT 2016]

Answer:

Here a) will succeed but b) will fail.

Here in (a) when we update in Student table dept id =NULL then it will not cause any problem to referenced table.

But in (b) if we set in Department table dept id =NULL then it will produce inconsistency because in Student table we still have the tuples containing the dept id =1.

12. Let R (a, b, c) and S (d, e, f) be two relations in which d is the FK for S that refers to the primary key of R. Which of the following is true about the Referential integrity constraint? Give proper explanation for your answer for choosing or not choosing each of the option.

- a) Insert into R
- b) Insert into S
- c) Delete from R
- d) Delete from S.

[WBUT 2016]

Answer:

Both (b) and (c) can cause its violation

When 'Insert into S' operation will take place there will be inconsistency in the database, since it has a foreign key which refers to the primary key of R. When 'Delete from R' take place it will cause violation because since it's primary key is the foreign key for S so there will be inconsistency in the database. Before deleting from R, the corresponding rows from S has to be deleted. The other options (a) and (d) will not cause such problem.

13. What is Relationship? Explain different degrees of relationship. [WBUT 2017]

OR,

Explain the term Unary Relationship with example.

[WBUT 2017]

Answer:

A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

The **degree of relationship** (also known as cardinality) is the number of occurrences in one entity which are associated (or linked) to the number of occurrences in another.

There are three degrees of relationship, known as:

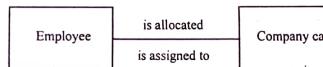
- 1) one-to-one (1:1)

- 2) one-to-many (1:M)
- 3) many-to-many (M:N)

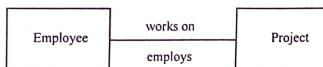
One-to-one (1:1)

This is where one occurrence of an entity relates to only one occurrence in another entity. A one-to-one relationship rarely exists in practice, but it can. However, you may consider combining them into one entity.

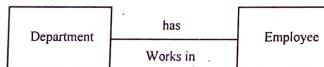
For example, an employee is allocated a company car, which can only be driven by that employee.

**One-to-Many (1:M)**

Is where one occurrence in an entity relates to many occurrences in another entity. For example, an employee works in one department but a department has many employees.

**Many-to-Many (M:N)**

This is where many occurrences in an entity relate to many occurrences in another entity. As with one-to-one relationships, many-to-many relationships rarely exist. Normally they occur because an entity has been missed. For example, an employee may work on several projects at the same time and a project has a team of many employees.



14. Explain generalization, specialization and aggregation in entity relation diagram. [WBUT 2012]

OR,

Define the concept of aggregation, generalization, and specialization and attribute inheritance. [WBUT 2015]

OR,

With proper diagram explain extended ER features (Generalization, Specialization and Aggregation). [WBUT 2018]

Answer:

Generalization, Specialization:

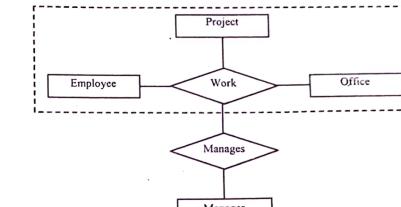
Refer to Question No. 1(b) of Short Answer Type Questions.

Aggregation:

An abstraction through which relationships are treated as higher-level entities is called aggregation. For example; the relationship set **work** relating the entity sets **employee**,

BC DB-38

office and project is considered as a higher-level entity set. Such an entity set is treated in the same manner as is any other entity set. Now we create a binary relationship **manages** between works and manager to represent who manages what tasks.

**Attribute Inheritance**

In attribute inheritance the attributes of a higher level entity set are inherited through a lower level entity set made by specialization-generalization hierarchy. Namely, all the attributes of the higher level entity set are as well the attributes of the lower level entity set.

15. What is constraint? Explain domain constraint and Entity Integrity constraint. [WBUT 2017]

Answer:

Constraints or Integrity Constraints ensures that the data in a database is both accurate and complete, in other words, that the data makes sense. There are at least five different types of integrity that needs to be considered:

- Domain constraints
- Entity integrity
- Column constraints
- User-defined integrity constraints
- Referential integrity

The data analysis stage will identify the requirements of these.

Domain Constraints:

A domain is defined as the set of all unique values permitted for an attribute. For example, a domain of date is the set of all possible valid dates, a domain of integer is all possible whole numbers, a domain of day-of-week is Monday, Tuesday ... Sunday. This in effect is defining rules for a particular attribute. If it is determined that an attribute is a date then it should be implemented in the database to prevent invalid dates being entered. If the system supports domain constraints then this invalid data would not have stored in the first place. That is, the integrity of the database is being preserved.

Entity integrity:

These constraints are the constraints that restrict the values of a tuple of a relation. The entity integrity constraint states that primary keys can't be null. There must be a proper value in the primary key field.

This is because the primary key value is used to identify individual rows in a table. If there were null values for primary keys, it would mean that we could not identify those rows. On the other hand, there can be null values other than primary key fields. Null value means that one doesn't know the value for that field. Null value is different from zero value or space.

16. Express division operation in terms of basic relational algebra operations.

[WBUT 2018]

Answer:

Union, difference, cross product, selection, projection form the "complete set of relational operations". That is, any other relational operation can be expressed as a combination of these.

Let $R(A, B)$ and $S(B)$ be two relations. Division should find all values of A in R that are connected with all values of B (in S).

$$R \div S = \pi_A(R) - \pi_A(\pi_A(R) \times S - R)$$

$\pi_A(R) \times S$: this contains all possible AB pairs.

R : this contains the actual AB pairs.

For the values of A that are connected to all values of B , after the difference, those will be gone. In the difference $\pi_A(R) \times S - R$ there are only values of A that are NOT connected to all B .

17. Explain the difference between weak entity set and strong entity set.

[MODEL QUESTION]

Answer:

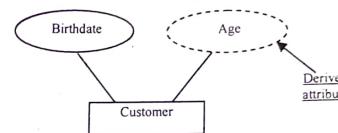
Strong entity sets	Weak entity sets
1) An entity set, which has a primary key, is termed as strong entity set.	1) An entity set is called a weak entity set if its existence depends on other entities i.e., strong entities..
2) A strong entity set have sufficient attributes to form a primary key.	2) A weak entity set does not have sufficient attributes to form a primary key.
3) A strong entity set is indicated in E-R diagram by a singly outlined rectangle.	3) A weak entity set is indicated in E-R diagrams by a doubly outlined rectangle.
4) The primary key of strong entity set is formed by the primary key of the strong entity set.	4) The primary key of weak entity set is formed by the primary key of the strong entity set plus its discriminator.

18. Give an example of derived attribute.

[MODEL QUESTION]

Answer:**Derived Attribute**

The attribute values which are not a part of the entity but can be computed in association with some other stored attributes.



For example the age of a person which can be computed by subtracting the dob from the current date.

19. Differentiate between the following:

[MODEL QUESTION]

- a) Partial key and Primary key
- b) Multi-valued attributes and Composite attributes
- c) Subclasses with disjoint constraints and overlapping constraints.

Answer:

a) **Partial Key**: It is a set of attributes that can uniquely identify weak entities and that are related to same owner entity. It is sometime called as Discriminator.

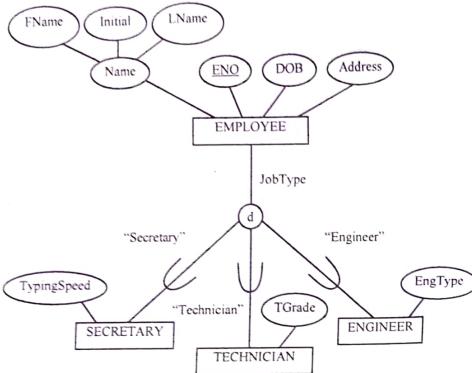
Primary key - A primary key is a value that can be used to identify a unique row in a table. Attributes are associated with it. Examples of primary keys are Social Security numbers (associated to a specific person) or ISBNs (associated to a specific book).

b) **Composite attributes** can be divided into subparts. For example, an attribute name could be organized as a composite attribute consisting of first-name, middle-initial, and last-name and can be divided into sub parts. E.g. Name (First name, Middle Name, last name)

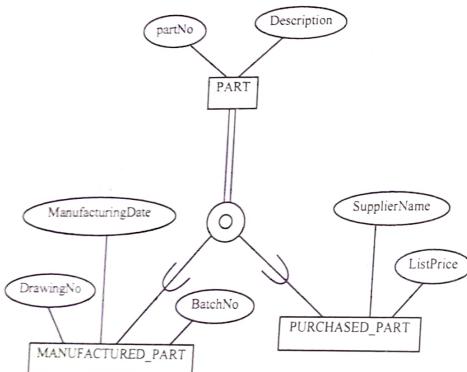
Multivalued Attributes may be instances where an attribute has a set of values for a specific entity. Consider an employee entity set with the attribute phone-number. An employee may have zero, one, or several phone numbers, and different employees may have different numbers of phones. This type of attribute is said to be Multivalued Attributes. Attribute having more than one values. E.g. Phone Number.

c) Disjoint constraint

This constraint specifies that the subclasses of the specialization must be disjoint. This means that an entity can be a member of at most one of the subclasses of the specialization. A specialization that is attribute-defined implies the disjoint ness constraint if the attribute used to define the membership predicate is single valued. It is displayed by placing d in the circle.

**Overlap constraint**

This constraint specifies that the subclasses of the specialization are not constrained to be disjoint. This means that an entity can be a member of more than one subclasses of the specialization. This case, which is the default, is displayed by placing **O** in the circle.

**Long Answer Type Questions**

1. a) What is multiple relationship?

[WBUT 2008, 2009, 2010]

- b) What is attribute inheritance?

[WBUT 2008, 2009, 2010, 2012]

- c) A post office has few postmen who go everyday to distribute letter. Every morning the post office receives a large number of registered letter. The post office intends to create a database to keep track of these letters.

Assumption:

- Every letter has a sender & origin post office, destination post office, a date of registration, a date of arrival of destination post office, the receiver & the status.
- Every sender has a name & address. Every receiver has a name & address.
- Every postman has a designated area where he delivers letter.
- The area consists of street under the jurisdiction of the post office.
- The status of the letter can be not yet taken for delivery. Delivered, address not available, receiver didn't accept the letter, redirected to the new address of the receiver and send it.

Give an Entity Relationship Diagram for the above application.

[WBUT 2008]

Answer:

- a) The best way to answer this question is to look at each relationship that is possible between tables (or entities). There are three. The one to one, one to many and many to many.

One to one is easy. For example, one person can only have one social security number (SIN in US). So one table contains the name of the person and another table contains the SSN.

One to many - Here again we compare two values, each in separate tables. For example one teacher has many students.

Many to many - There's a simple way to look at this. Let's take our example from above. One teacher has many students. But isn't it also the case that one student has many teachers? So if we have two tables, one showing all the information about teachers and the other showing all the information about students. If we connect the table thus –

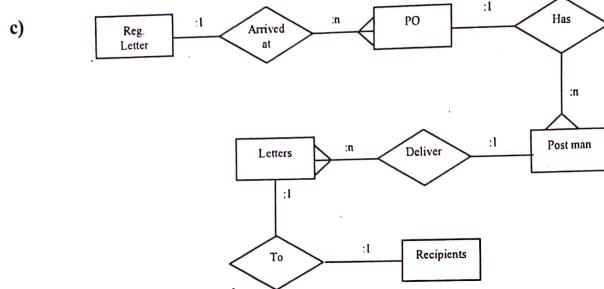
Teacher - 1 → many – Student

Then look at it from the other direction

Teacher - many → 1 – Student

But really the relationship is – Teacher - many → many - Student

- b) The higher and lower level entities created by specialization and generalization is **attribute inheritance**. The lower level entities are inherited from the higher level entities. For example, the entities project-member and the entities project-manager are inherited from the entity person.

**Attributes:**

REG-LETTER {sender, origin-post=office, destination-po, REG-date, date-of arrival, destination-po, Receiver-Add, Status}

PO {POcode, list of pdstman, service offered, area covered}

POSTMAN{PO reference, Postman Code, Delivery area}

LETTER{Sender address, Receiver address, Post code, Date written}

RECEIPIENTS{Receiver address ,Sender address, Date written, Date received}

2. Define ER model. What is an entity? What is multi-valued attribute? Draw the ER diagram from the following information:

"A store has different counters managed by different employee. A counter has item but no two counters have common items. Customers buy from different counters but bill are prepared at the bill counter only. Once in a month the performance of the persons managing different counters are evaluated in terms of sale. Items are also reviewed and slow moving items are identified." [WBUT 2009, 2012]

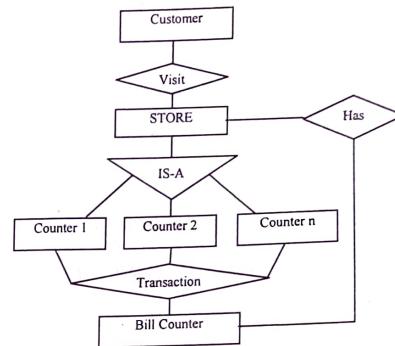
Answer:

ER model:

An entity-relationship (ER) model is a theoretical and conceptual way of showing data relationships in software development. ERM is a database modeling technique that generates an abstract diagram or visual representation of a system's data that can be helpful in designing a relational database.

An entity is an object with a physical existence or it may be conceptual – a particular course, student, faculty or a university course. It has certain properties. It can be concrete or abstract. In simple word it is the place where a data is stored. It may be organized or not.

Multi-Valued Attribute: The attribute have a multiple value for a particular entity. For example, a person may have multiple college degrees attribute.



Customer{Customer_ID, PhoneNo, Address, last Purchase_date, Time, BillNO, BillAMT,BillempNo}
 STORE{CounterNo, ItemNO, EmployeeNO, Date,Time}
 Transaction{Transno,Counterno,Employeeeno,Itemno,date,time}
 Transaction{BillCounter}

[WBUT 2013]

3. Consider a relation-

Bank (Customer_name,account_no, account_type, balance, branch)

Solve the following queries using SQL, Relational Algebra and Tuple Relational Calculus.

- Retrieve total balance amount for individual branch.
- Retrieve the name of the customers who have an account in "Dunlop" branch and balance less than Rs. 10,000.
- List the information of all customers of savings branch.
- Who have the minimum balance among all customers?
- Display the balance of those customers whose balance starts with the letter 'A'.

Answe:

(i) SQL: Select branch_name, sum(balance) as balance
 From BANK Group By branch_Name;

Algebra:(branch_name G sum(balance)) (Bank)

Calculus:{t| tε Bank ∧ Σ balance }

(ii) SQL: Select Customer_name

From Bank
 Where branch_name="Dunlop" ∧ balance<10,000;

Algebra: $\Pi_{customer_name} \sigma_{branch_name="Dunlop"} \wedge balance < 10,000(Bank)$

Calculus:

{t[Customer_name]| t ε Bank ∧ t[branch_name] = "Dunlop" } ∧ t[balance]<10,000}

(iii) Select Customer_name
From Bank
Where account_Type="Savings"

Algebra: $\Pi_{customer_name} \sigma_{account_type = "savings"}(Bank)$

Calculus: $\{t[Customer_name] \mid t \in Bank \wedge t[account_type] = "Savings"\}$

(iv) Select Customer_name
From Bank
Where balance=min(balance);

Algebra: $\Pi_{customer_name} \sigma_{balance = \min(balance)}(Bank)$

Calculus:
 $\{t[Customer_name] \mid t \in Bank \wedge t[balance] = \min(t[balance])\}$

(v) SQL: Select Customer_name, balance
From Bank

Where Customer_name LIKE 'S%'; **Algebra:** $\Pi_{customer_name, balance}(customer_name \text{ LIKE 'S%' } \wedge customer_name \text{ LIKE 'S%' } (Bank))$

Calculus:
 $\{t[Customer_name], t[balance] \mid t \in Bank \wedge t[customer_name] \text{ LIKE 'S%' }\}$

4. Define concept of aggregation. Give two examples where this concept is useful.
[WBUT 2014]

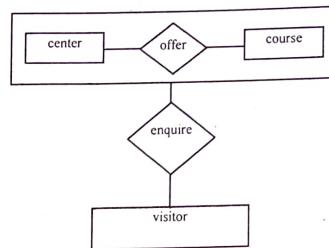
Answer:

Concept of aggregation

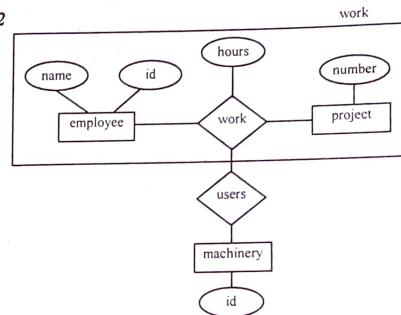
- Aggregation refers to performing an operation on a group of values to get a single result.
- One limitation of the E-R model is that it cannot express relationships among the relationships so to overcome this we use aggregation.
- Aggregation is a process when the relation between two entity is treated as a single entity.
- Aggregation represents abstract entities by allowing relationship between relationships.
- Aggregation is a special type of Association.
- Aggregation is also called a "Has-a" relationship.

UML Notation: ——————

Example - 1



Example - 2



5. Let $R = (A, B)$ and $S = (A, C)$ and let $r(R)$ and $r(S)$ be relations. Write relational algebra expression equivalent to the domain relational calculus expressions:

- a) $\{<a> \mid \text{there exist } b (<a, b> \text{ belongs to } r \wedge b = 17)\}$
 b) $\{<a, b, c> \mid <a, b> \text{ belongs to } r \wedge <a, c> \text{ belongs to } s\}$

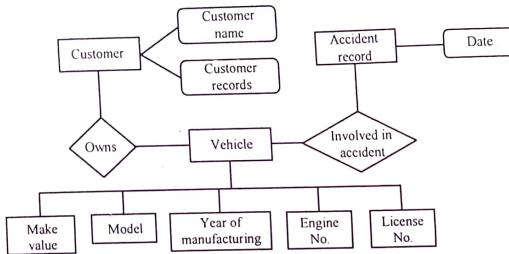
[WBUT 2014]

Answer:

- a) $\Pi_A (\sigma_B = 17 (r))$
 b) $r \sqcup s$

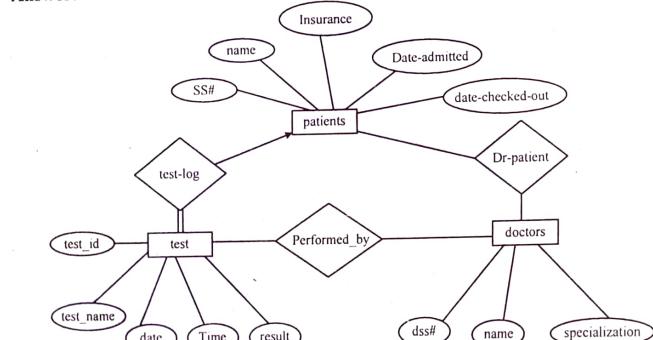
6. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. State all your assumptions.
[WBUT 2015]

Answer:



7. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

[WBUT 2016]

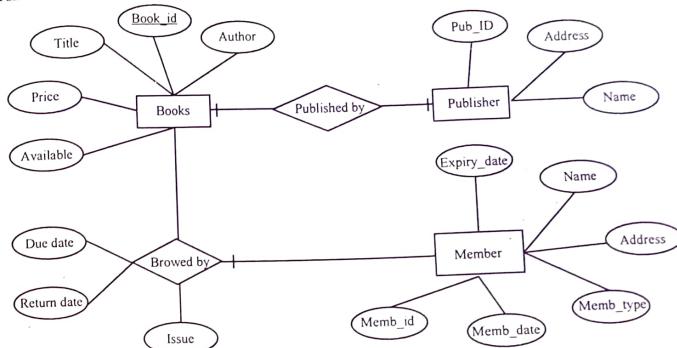


E-R diagram for a hospital

8. Draw an E-R Diagram for a library management system.

[WBUT 2017]

Answer:



E-R Diagram of Library Management System

9. Find out the candidate Keys for the following relation R:

R(A,B,C,D,E,H) , F = {A→B, BC→D, E→C, D→A}

Answer:

$(H)^+ \rightarrow H$
 $(HA)^+ \rightarrow HAB \dots \text{as, } A \rightarrow B$
 $(HAE)^+ \rightarrow HAEBCD \dots \text{as, } BC \rightarrow D, E \rightarrow C, D \rightarrow A$
 $(HD)^+ \rightarrow HDAB \dots \text{as, } A \rightarrow B, D \rightarrow A$
 $(HDE)^+ \rightarrow HDEABC \dots \text{as, } BC \rightarrow D, E \rightarrow C,$
 $(HE)^+ \rightarrow HEC \dots \text{as, } E \rightarrow C$
 $(HEB)^+ \rightarrow HEBBCDA \dots \text{as, } BC \rightarrow D, D \rightarrow A$
 Thus, {H, D, E}, {H, A, E}, {H, E, B} composite candidate keys.

10. Explain the following terms with example: Aggregation, Specialization, Generalization, Derived Attribute.

[WBUT 2017]

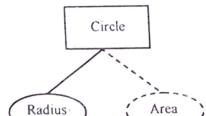
Answer:

Aggregation, Specialization, Generalization:

Refer to Question No. 14 of Short Answer Type Questions.

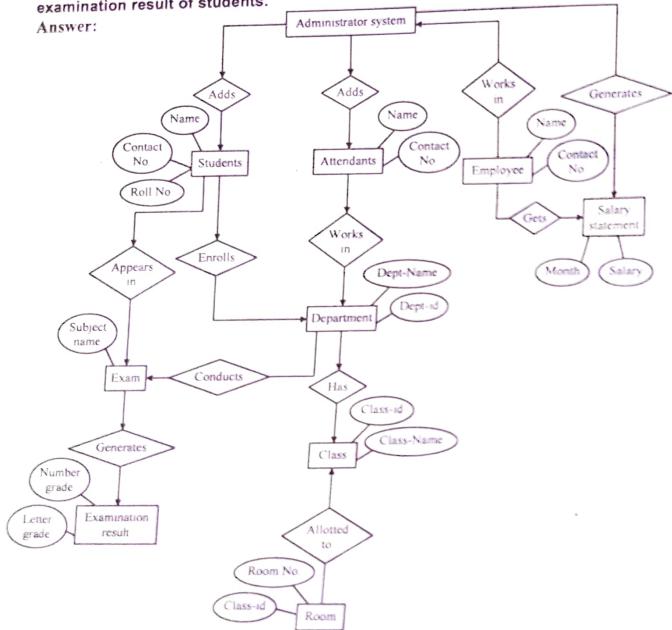
Derived Attribute:

An attribute based on another attribute is derived attribute. This is found rarely in ER diagrams. For example, for a circle, the area can be derived from the radius.



11. Draw a sample ER diagram for a college administration system. It should keep information like name and contact number of employees and students, attendants, examination result of students, [WBUT 2018]

Answer:



12. What is the application of Entity Relationship diagram? What is Generalization and Specialization? Design a ER diagram for Shopping Mall with all its entities, attributes and cardinality. [MODEL QUESTION]

Answer:

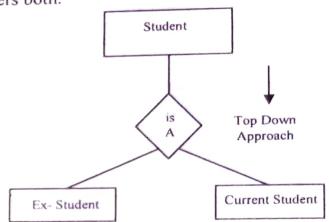
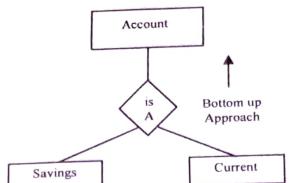
An ER model describes interrelated things of interest in a specific domain of knowledge. ER model is used to perform business processes. It is an abstract data model that defines a data or information structure which can be implemented in a relational database. The entities in the ER schema become tables, and attributes. With the increase in complexity of data Extended ER Features become necessary to model the complex applications.

BC DB-50

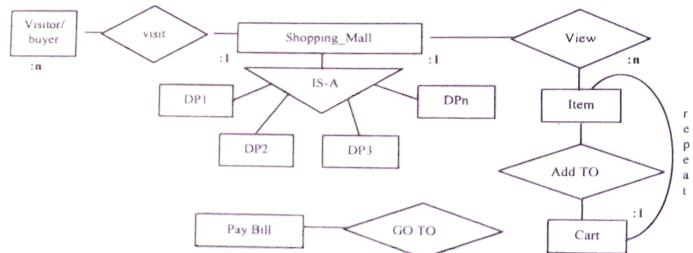
Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity. It's more like Superclass and Subclass system, but the only difference is the approach, which is bottom-up. Hence, entities are combined to form a more generalized entity, in other words, sub-classes are combined to form a super-class.

For example, **Saving** and **Current** account types entities can be generalized and an entity with name **Account** can be created, which covers both.

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.



E-R diagram of a shopping Mall



Attributes of the entities:

Visitor{VID, Name, Address, Phone}

Shopping_Mall{Name, Address, Phone, DID(fk), VID(fk)}

DP1{DID, ItemID, owner, phone}

Item{ItemID, ITDesc, ITname, unit, Rate, Date-of-manu, make, DID(fk)}

Cart{cartno, ItemID, Rate}

PayBill{Boothno, ItemID, Qty, Price, cartno(fk)}

BC DB-51

Primary key and foreign keys are described by '_', (FK) respectively.

13. Write short notes on the following:

[WBUT 2014]

[WBUT 2017]

- a) Mapping cardinalities
- b) Strong entity and weak entity

Answer:

a) **Mapping cardinalities:**

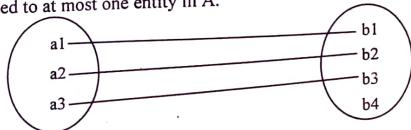
A mapping cardinality is a data constraint that specifies how many entities an entity can be related to in a relationship set.

Example: A student can only work on two projects, the number of students that work on one project is not limited.

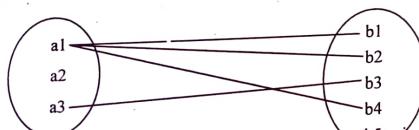
A binary relationship set is a relationship set on two entity sets. Mapping cardinalities on binary relationship sets are simplest.

Consider a binary relationship set R on entity sets A and B. There are four possible mapping cardinalities in this case:

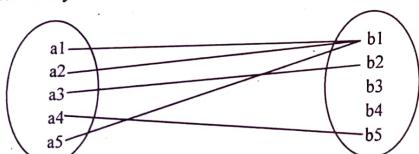
1. **One-to-one** - an entity in A is related to at most one entity in B, and an entity in B is related to at most one entity in A.



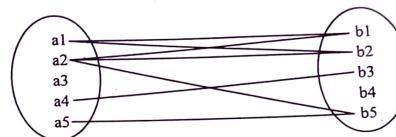
2. **One-to-many** - an entity in A is related to any number of entities in B, but an entity in B is related to at most one entity in A.



3. **Many-to-one** - an entity in A is related to at most one entity in B, but an entity in B is related to any number of entities in A.



4. **Many-to-many** - an entity in A is related to any number of entities in B, but an entity in B is related to any number of entities in A.



b) **Strong entity and weak entity:**

Refer to Question No. 9 of Short Answer Type Questions.

SQL**Multiple Choice Type Questions**

1. Which is the SQL command to remove rows from a table?

[WBUT 2006, 2007, 2011, 2015, 2017]
a) REMOVE b) DELETE c) TRUNCATE d) all of these

Answer: (b)

2. Which is the command in SQL to change a data in table?

[WBUT 2006, 2011]
a) UPDATE b) INSERT c) BROWSE d) APPEND

Answer: (a)

3. Which language has recently become the de-facto standard for interfacing application programs with relational database system? [WBUT 2007]

a) Oracle b) SQL c) dBASE d) 4GL

Answer: (b)

4. Which of the following is a comparison operation used in SELECT statement?

[WBUT 2008]
a) LIKE b) BETWEEN c) IN d) None of these

Answer: (c)

5. Four DML commands are

[WBUT 2009]
a) CREATE, UPDATE, DELETE, SELECT
b) INSERT, UPDATE, DROP, SELECT
c) CREATE, ALTER, DELET, SELECT
d) INSERT, MODIFY, DELETE, SELECT
e) INSERT, UPDATE, DELETE, SELECT

Answer: (e)

6. Updating a database means

[WBUT 2011]
a) revisiting the file structure
b) revisiting the database structure or schema
c) changing one or more database
d) normalization of database

Answer: (c)

7. Which is the command in SQL to change the structure in a table? [WBUT 2012]

a) Alter b) Update c) Append d) none of these

Answer: (a)

8. Which of the following keyword is used in SQL to eliminate duplicate rows from the query result? [WBUT 2013]

a) NO DUPLICATE
c) UNIQUE
b) DISTINCT
d) none of these

Answer: (b)

9. The command returns the number of rows deleted. [WBUT 2013]
a) Truncate b) Delete c) Drop d) None of these

Answer: (b)

10. Which of the following clauses is used to enforce a condition on a SQL statement containing "group by" clause? [WBUT 2013, 2017]
a) Where b) Having c) Order by
d) None of these

Answer: (b)

11. Which of the following is not an aggregate function? [WBUT 2013]
a) SUM b) MIN c) MAX
d) DISTINCT

Answer: (d)

12. The full form of CODASYL is
a) Correlated Data System Language
b) Conference on Data System Language
c) Cohesion on Data System Language
d) None of these

Answer: (d)

13. What is the default format of date in Oracle? [WBUT 2016]
a) dd-mm-yy
b) dd-m-yyyy
c) dd-mon-yy
d) none of these

Answer: (c)

14. Which of the following aggregate functions works with characters? [WBUT 2016]
a) max b) avg c) count d) none of these

Answer: (d)

15. If we do not specify the constraint name for a constraint, the default name is in the format [WBUT 2016]

a) SYS_Cn b) Cn_SYS c) Cn d) none of these

Answer: (a)

Short Answer Type Questions

1. Briefly explain Natural join with an example.

OR,

Explain natural join operation in relational algebra with requisite example.

[WBUT 2003]

[WBUT 2008]

Answer:

A natural join is a join statement that compares the common columns of both tables with each other. One should check whether common columns exist in both tables before doing a natural join. Natural joins may cause problems if columns are added or renamed. In natural join one need not to specify column names for the join – it will automatically join same name columns in two different tables.

For Example

Natural Join
SQL> select dptno from dept natural join emp;

2. Write a query to find all customers who do have a loan at the bank, but do not have an account at the bank.

```
borrower(customer_name, loan_no)
loan(branch_name, loan_no, amount)
Account(branch_name, account_no, balance)
depositor(customer_name, account_no)
```

[WBUT 2007]

Answer:

Select customer_name from borrower where loan-no in (select loan-no from loan) and loan-no in(select loan-no from loan where branch_name not in (select branch_name from account));

3. What are the types of Database Language? Explain them.

[WBUT 2009]

Answer:

Database languages are procedural types. Foundation is relational Algebra and the corresponding implementation is SQL.

4. Explain the database languages with SQL command.

[WBUT 2010]

Answer:

Because a database supports a number of user groups, as mentioned previously, the DBMS must have languages and interfaces that support each user-group.

Database Languages

DDL – the data definition language, used by the DBA and database designers to define the conceptual and internal schemas.

DDL commands: CREATE, ALTER, DROP etc.

In databases where there is a separation between the conceptual and internal schemas, DDL is used to specify the conceptual schema, and SDL, storage definition language, is used to specify the internal schema.

For a true three-schema architecture, VDL, view definition language, is used to specify the user views and their mappings to the conceptual schema. But in most DBMSs, the DDL is used to specify both the conceptual schema and the external schemas.

Once the schemas are compiled, and the database is populated with data, users need to manipulate the database. Manipulations include retrieval, insertion, deletion and modification.

The DBMS provides operations using the DML, data manipulation language.

DML commands: UPDATE, DELETE, INSERT.

In most DBMSs, the VDL, DML and the DML are not considered separate languages, but a comprehensive integrated language for conceptual schema definition, view definition and data manipulation. Storage definition is kept separate to fine-tune the performance, usually done by the DBA staff.

An example of a comprehensive language: SQL, which represents a VDL, DDL, DML as well as statements for constraint specification, etc

DCL(data control language) commands: COMMIT,ROLLBACK.

DQL(data query language) commands: SELECT

5. Let R = (A, B, C) and let r1 and r2 both be relation on schema R. Give an expression in SQL that is equivalent to each of the following queries: [WBUT 2011]

- i) $r1 \cup r2$
- ii) $r1 \cap r2$
- iii) $r1 - r2$
- iv) $\prod_{AB}(r1) \times \prod_{BC}(r2)$

Answer:

i) SELECT * FROM r1 UNION

SELECT * FROM r2

ii) SELECT * FROM r1 INTERSECT

SELECT * FROM r2

In the SQL the columns of r2 have to be in the same order as those of r1

iii) SELECT * FROM r1 EXCEPT

SELECT * FROM r2

In the SQL the columns of r2 have to be in the same order as those of r1.

iv) SELECT * FROM r1 FULL [OUTER] JOIN r2 on r1.B = r2.B

6. Differentiate between the following:

- a) Delete and truncate operations
- b) Referential integrity and entity integrity

[WBUT 2013, 2016]
[WBUT 2013]

Answer:

a) Delete is used to remove rows from a table

TRUNCATE is used to delete complete data from an existing table.

For example:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00

SQL>DELETE FROM CUSTOMERS WHERE ID = 3;
// will return the table

BC DB-57

```

ID | NAME      | AGE | ADDRESS     | SALARY
+---+-----+-----+-----+
| 1 | Ramesh   | 32 | Ahmedabad | 2000.00 |
| 2 | Khilan   | 25 | Delhi       | 1500.00 |

```

TRUNCATE TABLE CUSTOMERS;
 // will return
 SQL> SELECT * FROM CUSTOMERS; //Empty set (0.00 sec)

b) Referential integrity:

When two relations have common attributes and it is to be ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This condition is called referential integrity.

Let r1(R1) and r2(R2) be relations with primary keys K1 and K2, respectively. There is a subset α of R2 is a foreign key referencing K1 in relation r1 if it is required that, for every t_2 in r_2 , there must be a tuple t_1 in r_1 s.t $t_1[K_1]=t_2[\alpha]$.

Requirements of this form are called referential integrity constraints.

Entity Integrity:

Entity integrity constraints are the constraints that restrict the values of a tuple of a relation. There are two types of constraints namely Domain Constraints and key constraints.

Domain constraint specifies that the value of each attribute must be an atomic value from the domain of the respective attribute.

Key Constraint: A *unique constraint* or *unique key constraint* is a rule that forbids duplicate values in one or more columns within a table. Unique and primary keys are the supported unique constraints.

A *primary key constraint* is a column or combination of columns that has the same properties as a unique constraint.

A *foreign key constraint* is a logical rule about values in one or more columns in one or more tables.

7. What is a view?**[MODEL QUESTION]**

Answer:

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that directly represents the view instead a definition of view is stored in data dictionary.

Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

8. View does not take any memory space. Justify how do you create an insertable and Updateable View? **[MODEL QUESTION]**

Answer:

Views are organized over one or more tables. A view can also be considered as a stored query. Views do not actually contain data. It derives data from the tables on which they are based, referred to as the base tables of the views. Like tables, views can be queried, updated, inserted into, and deleted from, with some restrictions. All operations performed on a view actually affect the base tables of the view.

Views provide an additional level of table security by restricting access to a predetermined set of rows and columns of a table. They also hide data complexity and store complex queries. So the question of occupying additional memory space does not arise.

```

CREATE TABLE demotab (
pid NUMBER(3),
fname VARCHAR2(15),
lname VARCHAR2(15));

```

```

CREATE OR REPLACE VIEW updview AS
SELECT * FROM demotab;
INSERT INTO demotab
pid, fname, lname)
VALUES
(1, 'Dey', 'Moni');
INSERT INTO demo_tab
(pid, fname, lname)
VALUES
(2, 'Hanu', 'Lakhotia');
COMMIT;
SELECT * FROM updview;

```

View can be used to manipulate data. The update conditions of an insertable and updateable views are as follows:-

1. Views defined from a single table
2. To INSERT record through the view the primary key columns and all the not Null columns must be included in the view

3. UPDATE operations on the tuples can be made with the help of a view even if the primary key column and NOT NULL column(s) are excluded from the view definition.

An insert operation is attempted through view

Insert Into View-1 **values** (1, 'CESE', 5000);

//Output 1 row created

A modify operation is attempted through view

Update View-1 **SET** course='IT' **where** studentid=1;

//Output 1 row Updated

9. Consider the following relational schema:

[MODEL QUESTION]

EMP (EmpNo, EmpName, City, Sal, DeptNo)

DEPT (DeptNo, DeptName)

Write down the following queries in SQL:

- find the names of cities of all employees working for the 'Research' department
- Display the number of employees in each department
- Display the names of employees in department number 20
- Find the names of employees who have the second highest salary
- Find the names of employees whose salary is greater than anyone's salary of department 10.

Answer:

i) Select EmpName, City
from EMP, DEPT
where
EMP. DeptNo = DEPT.DeptNo
and
DEPT. DeptName = "Research";

ii) Select count(EmpName), DeptNo from EMP, DEPT
groupby(DeptName) where EMP. DeptNo = DEPT.DeptNo;

iii) Select EmpName from EMP,DEPT
Where EMP. DeptNo = DEPT.DeptNo
and DEPT. DeptNo = '20';

iv) Select max(Sal) from EMP where Sal NOT IN(select max(Sal) from EMP);

v) Select EmpName from EMP
Where Sal > (Select max(Sal)
From EMP where DEPT. DeptNo = '10');

10. What restrictions apply to the use of the aggregate functions within the SELECT statement? How do nulls affect the aggregate functions?

[MODEL QUESTION]

Answer:

The aggregate functions can be used in a select list or in the HAVING clause of a select statement that includes a GROUP BY clause.

One cannot use aggregate functions in a WHERE clause or in a JOIN condition. However, a SELECT statement with aggregate functions in its select list often includes a WHERE clause that restricts the rows to which the aggregate is applied.

If a SELECT statement includes a WHERE clause, but not a GROUP BY clause, an aggregate function produces a single value for the subset of rows that the WHERE clause specifies.

Whenever an aggregate function is used in a SELECT statement that does not include a GROUP BY clause, it produces a single value. This is true whether it is operating on all the rows in a table or on a subset of rows defined by a where clause.

One can use more than one aggregate function in the same select list, and produce more than one scalar aggregate in a single SELECT statement.

NULL values may cause aggregate functions to produce unexpected or incorrect results because these functions ignore NULL values for calculations. For example, if the Weight column of a table has 504 records, of which 299 records have NULL values, the average of this column calculated by using the AVG function may return incorrect values.

11. Consider a PL / SQL code to display the employee number and name of top 5 highest paid employees with CURSOR FOR LOOP statement. [MODEL QUESTION]

Answer:

```
declare
cursor c1 is select * from (select distinct basicsal,empno,ename from emp order by
basicsal desc) where rownum <=5;
rec c1%rowtype;
begin
dbms_output.put_line('empno|ename|basicsal');
for r in c1 loop
dbms_output.put_line(r.empno||"|"||r.ename||"|"||r.basicsal);
end loop;
end;
```

12. What is the difference between Procedural DML and Non-procedural DML?
[MODEL QUESTION]

Answer:

Data manipulation is

- Retrieval of Information Stored in Database
- Insertion of Information to the database
- Deletion of information from the database
- Updating of information stored in the database

Types of Data Manipulation Language (DML)

- Procedural DML
- Declarative DML (Non Procedural DML)

Procedural DML:

In procedural Data manipulation language user has to specify what data are needed and how to get it.

Declarative DML (Non Procedural DML):

In declarative Data manipulation language user has to specify what data are needed without specifying how to get it.

Example: SQL (Structured Query Languages)

13. What are the differences between Embedded SQL and Dynamic SQL?
[MODEL QUESTION]

Answer:

Embedded SQL	Dynamic SQL
i) In Embedded (static) mode the bind is done prior to the execution and is stored in a PLAN.	i) In dynamic mode the BIND occurs at execution time.
ii) It is hardcoded in the program itself and the value is changeable.	ii) It has the capability of changing the columns tables during the execution of program.
iii) It can contain any host variables.	iii) It does not contain any host variables.
iv) The embedded means to coding the sql query in the general purpose programming language. and programmer must access to a database.	iv) The dynamic sql means component of sql allows program to construct and submit sql queries at run time.
v) Embedded sql statement must be completely present at the compile time.	v) Dynamic sql statement may not be completely present at the compile time.

14. What are the recovery implications of:**[MODEL QUESTION]****a) forcing buffers to the database at COMMIT?****b) fever physically writing buffers to the database prior to COMMIT?****Answer:**

a) Provides durability without REDO logging. Never force the buffer from physical writing prior to commit.

b) A transaction ends when any of the following actions occurs:

1. User issue COMMIT/ROLLBACK command *without* [SAVEPOINT] option. Commit, makes the change permanent. It is visible to other users only after commit is executed.
2. When user issues a DDL command like; CREATE, DROP, RENAME, or ALTER. The database issues an implicit COMMIT statement before and after every DDL statement. If the current transaction contains DML statements, then it first commits the transaction and then runs and commits the DDL statement as a new, single-statement transaction.
3. Applications should always explicitly commit or undo transactions before program termination. A client process terminates abnormally, causing the transaction to be implicitly rolled back using metadata stored in the transaction table and the undo segment.

Thus after a transaction commits, users can view the changes. A commit is a fast operation, regardless of the transaction size and the speed of commit does not change with the size of the modified data in the transaction. The lengthiest part of the commit is the physical disk I/O performed by redo buffer e.g., LGWR in oracle. However, the amount of time spent by LGWR is reduced because it has been incrementally writing the contents of the redo log buffer in the background. However, for lower transaction commit latency, application developers can specify that redo be written asynchronously so that transactions need not wait for the redo to be on disk and can return from the COMMIT call immediately and from that perspective it is a fever.

15. Explain DDL, DML and DCL.**[MODEL QUESTION]****Answer:****Data Definition Language (DDL)**

A Data Definition Language (DDL) is used to describe the details of the data. The conceptual schema is specified by a set of definitions expressed by this special language. The DBMS will have a compiler whose function is to process DDL statements in order to identify description of the schema constructs and to store the schema descriptions as a set of tables in a special file called *data dictionary*. Thus, data dictionary is a file that contains data about data which is *called metadata*. This file is consulted before actual data are read or modified in the data base system.

Storage Definition Language: The SDL is used to specify the internal schema. The mapping between two languages can be specified by any one of the languages. In some DBMS, there is only one language which has both DDL and SDL capabilities. To construct, modify and administer the database the following commands are used.

- CREATE TABLE
- ALTER TABLE
- DROP TABLE

Data Manipulation Language (DML)

Once the database schemas are compiled and the database is filled with data, the data manipulation needs to covers the following functions:

1. The retrieval of information stored in the database.
2. The insertion of new information into the database.
3. The deletion of information from the database.
4. The modification of information stored in the database.

The language that enables the users to access or manipulate data is called the *data-manipulation language* (DML). DML are of two types.

Procedural DML: which require users to specify what data are needed and how to get those data.

Nonprocedural DML: which require a user to specify what data are needed without specifying how to get it.

The following statements are used for data manipulation.

- SELECT is used to retrieve data from the table.
- UPDATE is used to modify the data in the table.
- DELETE is used to delete one or more records from a relation.
- INSERT is used to insert a new tuple into a specified relation.

Data Control Language (DCL)

DCL is used to control the data population. It enables users to specify the beginning and ending of transactions. It deals with the authorization, access rights, recovery and integrity issues. The following statements are used to control the data in the data base.

Recovery and Concurrency:

- COMMIT
- ROLLBACK

Security:

- GRANT
- REVOKE

Integrity Constraints:

Integrity constraints are enforced by the system. For example, one can specify that an attribute of a relation will not accept null values. Detail syntaxes and examples are discussed later.

16. What is the database cursor? What is the difference between cursor and trigger? [MODEL QUESTION]

Answer:

1st Part:

An object used to store the output of a query for row-by-row processing by the application programs. Cursors are constructs that enable the user to name a private memory area to hold a specific statement for access at a later time. Cursors are used to process multi-row result sets one row at a time. Additionally, cursors keep track of which row is currently being accessed, which allows for interactive processing of the active set

2nd Part:

A trigger is a procedure (code segment) that is executed automatically when some specific events occur in a table/view of a database, while a cursor is a control structure used in databases to go through the database records. A cursor can be declared and used within a trigger. In such a situation, the declare statement would be inside the trigger. Then the scope of the cursor would be limited to that trigger. Within a trigger, if a cursor is declared on an inserted or a deleted table, such a cursor would not be accessible from a nested trigger. Once a trigger is completed, all the cursors created within the trigger will be de-allocated.

17. Write SQL Statements on the following tables: [MODEL QUESTION]

salespeople (snum, sname, city, comm.)

customer (cnum, cname, city, rating, snum)

orders (onum, amt, odate, cnum, snum).

a) Show the commission of all the salesperson who receive at least one order of amount greater than Rs. 5000.

b) Find all customers located in cities where salesperson "AMIT" has customers.

Answer:

a) Select sname, amt, comm
from orders, salespeople
where amt > 5000 and
salespeople.snum = orders.snum;

b) Select cname
from customer

```
where city in ( select city
    from customer, orders
    where customer.cnum = orders.cnum and
        orders.snum in ( select snum
            from salespeople
            where sname = 'AMIT'));
```

18. Write a PL/SQL code to calculate the total salary of first n records of emp (eno, ename, salary) table. The value of n is passed to cursor as parameter. [MODEL QUESTION]

Answer:

```
SQL> select * from emp;
1 declare
2 no_of_employee number;
3 total_salary number:=0;
4 cursor ec(n number) is select * from employee_salary where eno<=n;
5 rw ec%rowtype;
6 begin
7 no:=&no;
8 open ec(no_of_employee);
9 loop
10 fetch ec into rw;
11 exit when ec%notfound;
12 total_salary:=rw.salary+total_salary;
13 end loop;
14 dbms_output.put_line('Total salary of'||no||' employee is'||total_salary);
15* end;
```

19. What is the aggregate function in SQL? Give example of each of them.

[MODEL QUESTION]

Answer:

The SQL aggregate functions are used to retrieve minimum and maximum values from a column, to sum values in a column, to get the average of a column values, or to simply count a number of records according to a search condition. Let us take the examples on SQL aggregate functions:

Carmaker	Model	Year	Color	Amount
Toyota	Camry XLE	2005	white	500000
Honda	Accord EX	2002	Black	600000
Lexus	ES 350	2008	white	400000
BMW	3Series Coupe	2009	Red	500000

The most commonly used SQL aggregate function is the COUNT function. Here is an example:

SELECT COUNT() FROM Cars WHERE Color = 'white';*

The result of this will be the number 2.

The minimum and maximum can be selected on Year from Cars as follows:

`SELECT MIN (Year) FROM Cars;`

`SELECT MAX (Year) FROM Cars;`

The results will be 2001 and 2009 respectively.

Average Year from the Cars table can be selected:

`SELECT AVG (Year) FROM Cars;`

The result will be 506,

Finally you can use the SUM SQL aggregate function to get the sum of values in a certain column:

`SELECT SUM (Amount) FROM Cars;`

The SUM will be 20,000, to get the total value of all the cars in the above table.

20. Explain the features of SQL.

[MODEL QUESTION]

Answer:

The features of Structured Query Language (SQL) are:

- (i) It is easy to learn.
- (ii) SQL is used to access data from relational database management systems.
- (iii) It is used to describe the data and defines the data in the database and manipulate it when needed.
- (iv) It is used to create and drop the database and table. It is also used to create a view, stored procedure, function in a database.
- (v) SQL allows users to set permissions on tables, procedures and views.

Long Answer Type Questions

1. Consider the following two schemas:

[WBUT 2010, 2012, 2017]

`EMP (EMP#, ENAME, JOB, HIREDATE, MANAGER#, SALARY, COMM, DEPT#)`

`DEPT (DEPT#, DNAME, LOCATION)`

Perform the following queries on the tables (write appropriate SQL statement):

- i) List the name, salary and PF amounts of all employees (PF is calculated as 10% of the basic)
- ii) List the number of employees and average salary in DEPT# 20
- iii) List of the department number and total salary payable in each department
- iv) List the names of the employees who are more than twenty years old in the company
- v) List the names of the employees whose name either starts or ends with S.

Answer:

i) `SELECT ENAME, SALARY, SALARY*0.1
FROM EMP`

ii) `SELECT COUNT(EMP#),AVG(SALERY)
FROM EMP WHERE DEPT#=20`

iii) `SELECT DEPT#, SALERY`

`FROM EMP,DEPT WHERE EMP.DEPT#=DEPT.DEPT#
GROUP BY DEPT#`

iv) `SELECT ENAME
FROM EMP WHERE HIREDATE>25`

v) `SELECT ENAME
FROM EMP WHERE ENAME LIKE 'S%'
OR ENAME LIKE '%S'`

[WBUT 2014]

2. Consider the following tables

`employee (emp_name, street, city)`

`Works (emp_name, company_name, salary)`

`company (company_name, city)`

`managers (emp_name, manager_name)`

Give SQL expression for the following queries

- a. Find the names and cities of residence of all employees who work for first Bank Corporation.
- b. Find the name, street address and cities of residencies of all employees who work for first Bank Corporation and earn more than Rs. 100000.
- c. Find all employees in the database who earn more than each employee of Small Bank Corporation.

Answer:

a) $\Pi_{\text{person-name}} (\sigma_{\text{company-name} = \text{"First Bank Corporation"}} (\text{works}))$

SQL expression

`select e.emp_name, city from employee e, works w where w.company_name = 'First Bank Corporation' and w.emp_name = e.emp_name`

b) $\Pi_{\text{person_name}, \text{street}, \text{city}}$

$(\sigma_{(\text{company_name} = \text{"First Bank Corporation"}) \wedge (\text{salary} > 10000)} \text{works} \bowtie \text{employee})$

SQL expression

`select * from employee where emp_name in (select emp_name from works where company-name = 'First Bank Corporation' and salary > 10000)`

c) $\Pi_{\text{person_name}} (\text{works}) - \Pi_{\text{works_person_name}} (\text{works} \bowtie (\text{works.salary} \leq \text{works2.salary} \wedge \text{works2.company_name} = \text{"Small Bank Corporation"})) \rho_{\text{works2}(\text{works})}$

SQL expression

`select emp_name from works where salary > all (select salary from works where company-name = 'Small Bank Corporation')`

3. Consider the following schema:

[WBUT 2015]

Book(acc_no, yr_pub, title)
User(card_no, bname, baddress)

Borrow(acc_no, doi, card_no)

Where acc_no is account number, yr_pub is year of publication, bname is borrower name, baddress is borrower address, doi is date of issue.

Perform the following queries on the table.(In SQL)

- Find the account number whose year of publication is 1985.
- Display the title of the book which has been borrowed by "Vijoy"
- Find the borrower name who lives in same city as "Vijoy"
- find the borrower name and address who should issue book on 14-05-1988
- Find the acc_no of Book whose year of publication is 1992 and title is "Compiler Design"

Answer:

- select acc_no from Book where yr_pub = '1985'
- select b.title from Book b, User u, Borrow w where u.bname = 'Vijoy' and b.acc_no = w.acc_no and u.card_no = w.card_no
- select u.bname from User u,User v where v.bname = 'Vijoy' and u.baddress=v.baddress
- select u.bname,u.baddress from User u,Borrow b where u.card_no = b.card_no and b.doi = '14-05-1988'
- select acc_no. from Book where yr_pub = '1992' and title = 'Compiler Design'

4. Write and explain GROUP BY, LIKE, DISTINCT, INNER JOIN and UPDATE commands in SQL. Also give one example for each.

[WBUT 2016]

GROUP BY Statement

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.Example:

Consider the CUSTOMERS table is having the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

If we want to know the total amount of salary on each customer, then GROUP BY query would be as follows:

SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS
GROUP BY NAME;

This would produce the following result:

NAME	SUM(SALARY)
Chaitali	6500.00
Hardik	8500.00
kaushik	2000.00
Khilan	1500.00
Komal	4500.00
Muffy	10000.00
Ramesh	2000.00

LIKE Statement

The SQL LIKE clause is used to compare a value to similar values using wildcard operators. There are two wildcards used in conjunction with the LIKE operator:

- The percent sign (%)
- The underscore (_)

The percent sign represents zero, one, or multiple characters. The underscore represents a single number or character. The symbols can be used in combinations.
Let us take a real example, consider the CUSTOMERS table having the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Following is an example, which would display all the records from CUSTOMERS table where SALARY starts with 200:

SQL> SELECT * FROM CUSTOMERS

WHERE SALARY LIKE '200%';

This would produce the following result:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
3	kaushik	23	Kota	2000.00

The SQL DISTINCT keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records.

There may be a situation when you have multiple duplicate records in a table. While fetching such records, it makes more sense to fetch only unique records instead of fetching duplicate records.

Consider the CUSTOMERS table having the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

First, let us see how the following SELECT query returns duplicate salary records:

```
SQL> SELECT SALARY FROM CUSTOMERS
      ORDER BY SALARY;
```

This would produce the following result where salary 2000 is coming twice which is a duplicate record from the original table.

SALARY
1500.00
2000.00
2000.00
4500.00
6500.00
8500.00
10000.00

Now, let us use DISTINCT keyword with the above SELECT query and see the result:

```
SQL> SELECT DISTINCT SALARY FROM CUSTOMERS
      ORDER BY SALARY;
```

This would produce the following result where we do not have any duplicate entry:

SALARY
1500.00
2000.00
4500.00
6500.00
8500.00
10000.00

INNER JOIN statement

The most frequently used and important of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.

The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

Consider the following two tables, (a) CUSTOMERS table is as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

(b) Another table is ORDERS as follows:

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

Now, let us join these two tables using INNER JOIN as follows:

```
SQL> SELECT ID, NAME, AMOUNT, DATE
      FROM CUSTOMERS
      INNER JOIN ORDERS
      ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

This would produce the following result:

ID	NAME	AMOUNT	DATE
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

UPDATE statement

The SQL UPDATE Query is used to modify the existing records in a table.

You can use WHERE clause with UPDATE query to update selected rows otherwise all the rows would be affected.

Consider the CUSTOMERS table having the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Following is an example, which would update ADDRESS for a customer whose ID is 6:

```
SQL> UPDATE CUSTOMERS
SET ADDRESS = 'Pune'
WHERE ID = 6;
```

Now, CUSTOMERS table would have the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	Pune	4500.00
7	Muffy	24	Indore	10000.00

If you want to modify all ADDRESS and SALARY column values in CUSTOMERS table, you do not need to use WHERE clause and UPDATE query would be as follows:

```
SQL> UPDATE CUSTOMERS
SET ADDRESS = 'Pune', SALARY = 1000.00;
```

Now, CUSTOMERS table would have the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Pune	1000.00
2	Khilan	25	Pune	1000.00
3	kaushik	23	Pune	1000.00
4	Chaitali	25	Pune	1000.00
5	Hardik	27	Pune	1000.00
6	Komal	22	Pune	1000.00
7	Muffy	24	Pune	1000.00

5. Consider the following schemas:

[WBUT 2016]

Employee_master (EmpNo, Name, Job, Hiredate, Salary, manager_id, Dept_no, Age, E_sal)

Perform the following queries on table (write appropriate SQL statement) (any five)

- List all employees' names and jobs whose job includes 'M' or 'P'.
- List all employees' names and their salaries whose salary lies between 15000 and 35000. (using between clause)
- List all employees' names, salaries and 25% raise in salary.
- Find how much amount the company is spending towards salary head.
- List all employees' names and their manager_id whose manager_id is 7902, 7566 or 7789.
- List the difference between minimum and maximum salaries of employees.

Answer:

- select Name, Job from Employee_master where Job like 'M%' or Job like 'P%';
- select Name from Employee_master where Salary between 1500 and 3500;
- select Name , Salary , Salary+0.25* Salary from Employee_master;
- select sum (Salary) from Employee_master;
- select Name, manager_id from Employee_master where manager_id in(7602,7566,7789);
- select max(Salary)-min(Salary) from Employee_master;

6. Consider the following two schemas:

[WBUT 2018]

EMPLOYEE (EMP_ID, FNAME, ADDRESS, JOIN_DATE, SALARY, MANAGER#, DEPT_ID). DEPT(DEPT_ID, DNAME, LOCATION).

Write appropriate SQL statements based on above tables.

- List the details of employees whose salary is less than the average salary.
- List the department id and the number of employees working in that department.
- List the name of employees whose name have exactly five letters.
- List the details of employees who are more than 10 years old in the company.
- Display the minimum and maximum salary of the employee.

Answer:

- SELECT * FROM EMPLOYEE WHERE SALARY < ALL (SELECT AVG(SALARY) FROM EMPLOYEE);

```
b) SELECT DEPT_ID AS 'Department ID',
COUNT(*) AS 'No of Employees'
FROM DEPT
INNER JOIN EMPLOYEE
ON EMPLOYEE.DEPT_ID = DEPT.DEPT_ID;
```

```
c) SELECT FNAME FROM EMPLOYEE WHERE FNAME LIKE '_____';
```

```
d) SELECT * FROM EMPLOYEE WHERE EXTRACT (YEAR FROM AGE
(CURRENT_DATE, JOIN_DATE)) > 10;
```

```
e) SELECT MAX(SALARY), MIN(SALARY) FROM EMPLOYEE;
```

7. Write short note on Query language.**Answer:****Query language:**

SQL is the defacto standard of query language for creating and manipulating data in relational database. SQL can be used to map relational design to a database. Till now SQL is used in an interactive manner. However, **embedded SQL** and **Dynamic SQL** can be used for application development. The function can be called in any PL/SQL block where a primary key check is to be carried out and depending upon the return value of the function. A set of records resulting out of a query can be handled in the host language by declaring a cursor. The new SQL standard is SQL3 which will be easy to handle BLOB (Binary large objects).

8. Explain the importance of view in providing security to the database. What are the importance of GRANT and REVOKE commands?**[MODEL QUESTION]****Answer:****1st Part:**

Views can serve as security mechanisms by restricting the data available to users. Some data can be accessible to users for query and modification, while the rest of the table or database is invisible and inaccessible. Permission to access the subset of data in a view must be granted, denied, or revoked, regardless of the set of permissions in force on the underlying table(s).

For example, the salary column in a table contains confidential employee information, but the rest of the columns contain information that should be available to all users. You can define a view that includes all of the columns in the table with the exception of the sensitive salary column. As long as table and view have the same owner, granting SELECT permissions on the view allows the user to see non-confidential columns in the view without having any permissions on the table itself.

By defining different views and granting permissions selectively on them, users, groups, or roles can be restricted to different subsets of data. For example:

- Access can be restricted to a subset of the rows of a base table. For example, define a view that contains only rows for business and psychology books and keep information about other types of books hidden from users.
- Access can be restricted to a subset of the columns of a base table. For example, define a view that contains all the rows of the titles table but omits the royalty and advance columns because this information is sensitive.
- Access can be restricted to a row-and-column subset of a base table.
- Access can be restricted to the rows that qualify for a join of more than one base table. For example, define a view that joins the titles, authors, and title author tables to display the names of authors and books they have written. This view hides personal data about the authors, and financial information about the books.
- Access can be restricted to a statistical summary of data in a base table. For example, define a view that contains only the average price of each type of book.
- Access can be restricted to a subset of another view or of some combination of views and base tables.

[MODEL QUESTION]**2nd Part:****SQL GRANT Command**

SQL GRANT is a command used to provide access or privileges on the database objects to the users.

The Syntax for the GRANT command is:

```
GRANT privilege_name  
ON object_name  
TO {user_name |PUBLIC |role_name}  
[WITH GRANT OPTION];
```

For Example: GRANT SELECT ON employee TO user1; This command grants a SELECT permission on employee table to user1. You should use the WITH GRANT option carefully because for example if you GRANT SELECT privilege on employee table to user1 using the WITH GRANT option, then user1 can GRANT SELECT privilege on employee table to another user, such as user2 etc. Later, if you REVOKE the SELECT privilege on employee from user1, still user2 will have SELECT privilege on employee table.

SQL REVOKE Command

The REVOKE command removes user access rights or privileges to the database objects.

The Syntax for the REVOKE command is:

```
REVOKE privilege_name  
ON object_name  
FROM {user_name |PUBLIC |role_name}
```

For Example: REVOKE SELECT ON employee FROM user1; This command will REVOKE a SELECT privilege on employee table from user1. When you REVOKE SELECT privilege on a table from a user, the user will not be able to SELECT data from that table anymore. However, if the user has received SELECT privileges on that table from more than one users, he/she can SELECT from that table until everyone who granted the permission revokes it. You cannot REVOKE privileges if they were not initially granted by you.

9. Write short notes on the following:

- a) Inner join and Outer join
- b) Theta-join
- c) Equijoin
- d) Natural Join
- e) Triggers
- f) Embedded SQL

[WBUT 2011, 2013, 2018]**[WBUT 2013]****[WBUT 2013, 2017]****[WBUT 2013, 2017]****[MODEL QUESTION]****[MODEL QUESTION]**

Answer:**a) Inner join and Outer join:****• Inner Join**

Also known as a simple join. It returns all rows from both tables where there is a match. If there are rows in R1 which do not have matches in R2, those rows will not be considered. There are two possible ways of specifying this type of join:

`SELECT * FROM R1, R2 WHERE R1.r1_field = R2.r2_field;`

`SELECT * FROM R1 INNER JOIN R2 ON R1.field = R2.r2_field`

• Natural Join

A natural join is based on all columns in the two tables that have the same name. It is semantically equivalent to an INNER JOIN or a LEFT JOIN with a USING clause that names all columns that exist in both tables.

`SELECT * FROM R1 NATURAL JOIN R2`

• Left [Outer] Join

Returns all the rows from R1 even if there are no matches in R2. If there are no matches in R2 then the R2 values will be shown as null.

`SELECT * FROM R1 LEFT [OUTER] JOIN R2 ON R1.field = R2.field`

• Right [Outer] Join

Returns all the rows from R2 even if there are no matches in R1. If there are no matches in R1 then the R1 values will be shown as null.

`SELECT * FROM R1 RIGHT [OUTER] JOIN R2 ON R1.field = R2.field`

• Full [Outer] Join

Returns all the rows from both tables even if there are no matches in one of the tables. If there are no matches in one of the tables then its values will be shown as null.

`SELECT * FROM R1 FULL [OUTER] JOIN R2 ON R1.field = R2.field`

b) Theta-join:

Theta joins are those joins that use any of the comparison operators (i.e. less than, less than or equal to, greater than, greater than or equal to). A *theta-join* is any Cartesian product that's filtered by a condition which compares values from both Tables. The general theta-join form is:

`<Table_1.Column> relator <Table_2.Column>`

where the relator is almost always " $=$ ", as in this example:

`Sellers.seller_name = Sales.seller_name`

This special case of theta-join — where the relation is equality — is called an *equijoin*.

c) Equijoin:

SQ L EQUI JOIN performs a JOIN against equality or matching column(s) values of the associated tables. An equal sign ($=$) is used as comparison operator in the WHERE clause to refer equality.

We may also perform EQUI JOIN by using JOIN keyword followed by ON keyword and then specifying names of the columns along with their associated tables to check equality.

SYNTAX

```
SELECT column_list
FROM table1, table2...
WHERE table1.column_name =
table2.column_name;
```

There is difference between equijoin and innerjoin. An equijoin is a join with a join condition containing an equality operator. An equijoin returns only the rows that have equivalent values for the specified columns.

An inner join is a join of two or more tables that returns only those rows (compared using a comparison operator) that satisfy the join condition.

d) Natural Join:

A NATURAL JOIN is a JOIN operation that creates an implicit join clause for us based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.

A NATURAL JOIN can be an INNER join, a LEFT OUTER join, or a RIGHT OUTER join. The default is INNER join.

If the SELECT statement in which the NATURAL JOIN operation appears has an asterisk (*) in the select list, the asterisk will be expanded to the following list of columns (in this order):

- All the common columns
- Every column in the first (left) table that is not a common column
- Every column in the second (right) table that is not a common column

An asterisk qualified by a table name (for example, COUNTRIES.*) will be expanded to every column of that table that is not a common column.

If a common column is referenced without being qualified by a table name, the column reference points to the column in the first (left) table if the join is an INNER JOIN or a LEFT OUTER JOIN. If it is a RIGHT OUTER JOIN, unqualified references to a common column point to the column in the second (right) table.

SYNTAX

`TableExpression NATURAL [(LEFT|RIGHT) [OUTER] | INNER] JOIN`

e) Trigger:

A Trigger is a series of PL/SQL statements or code that is automatically and immediately fired when a certain event took place on a table in the database. They are pre built code and used to for logging purpose or to restrict people from accessing some data. A PL/SQL block consisting of one or more SQL commands which execute insert, update,

and delete statements on a table. The trigger can be defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. Triggers provide an alternative way to implement the referential integrity constraints provided by the foreign keys. For example; to define a trigger that is executed when someone attempts to insert a row into a table, and the trigger checks that the inserted data is valid.

e.g.,

```
CREATE TRIGGER TRIGGERNAME  
ON SCHEMA  
FOR {UPDATE | INSERT | DELETE}  
As SQL Statement.  
1. Create trigger delete_art  
2. on recording  
3. for delete as  
4. begin  
5. delete from art where art_id not in  
6. (select art_id from recording)  
7. end  
8. go
```

Strengths & Weakness of Trigger

Triggers are fired implicitly on the tables/views on which they are created.

Strengths

- Update Automatically one or more tables whenever a DML/DDL statement is executed for the table on which the trigger is created.
- Triggers can be used to enforce constraints. For e.g.: Any insert/update/ Delete statements should not be allowed on a particular table on holidays and after office hours.
- Triggers can be used to publish information about database events to subscribers. Database event like Database startup or shutdown or it can be a user even like User login or user logoff.

Weakness:

- It is easy to view table relationships, constraints, indexes, stored procedure in database but triggers are difficult to view.
- Triggers execute invisible. They are not visible or can be traced in debugging code.
- It is hard to follow the logic as it is fired before or after the database insert/update happens.
- As there is no documentation it will be difficult to figure out for new developers for their existence.
- Triggers run every time when the database fields are updated and it is an overhead on the system and it makes the system slower.

f) Embedded SQL:

Embedded SQL means SQL statements embedded in *host language* (C in our case). The original idea was for end-users to access a database through SQL Called *casual users*.

BC DB-78

But this is not a good idea, Takes too much concentration. Can you picture airline reservation clerk doing job with SQL?

Customers waiting for booking flight (update of seat table). Billing?

- Need to know all tables & columns, create complex syntax (FOR ALL)
- Too much risk of mistakes, especially with updates

Instead, we have an *Application Programmers* create menu applications, perform selects and updates programmatically. Programmers can spend a lot of time making sure the right SQL statement is used; programmers are temperamentally suited to this kind of work. Of course it is nice to have interactive SQL for some situations. Aim for this chapter is to be able to implement ANY CONCEIVABLE ALGORITHM using a C program that can get at data through SQL statements.