

## DATABASE MANAGEMENT SYSTEMS LAB



### MCA(OLODL)

Mr. Gobinda Chandra Das

Asst. Prof. MCA

KL Deemed to be University

#### Co-requisites:

- Co-requisite of course “Database Management Systems”

#### Course Objectives:

- Introduce ER data model, database design and normalization
- Learn SQL basics for data definition and data manipulation

#### Course Outcomes:

- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers

#### LIST OF EXPERIMENTS:

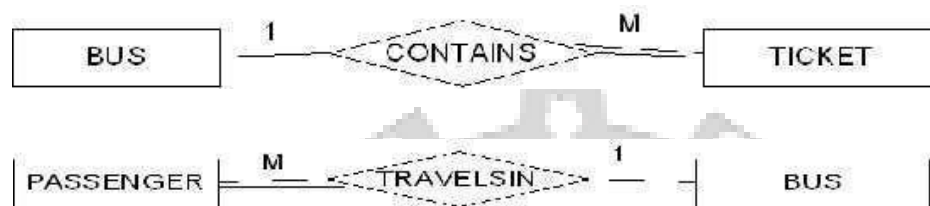
1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

### Concept design with E – R model

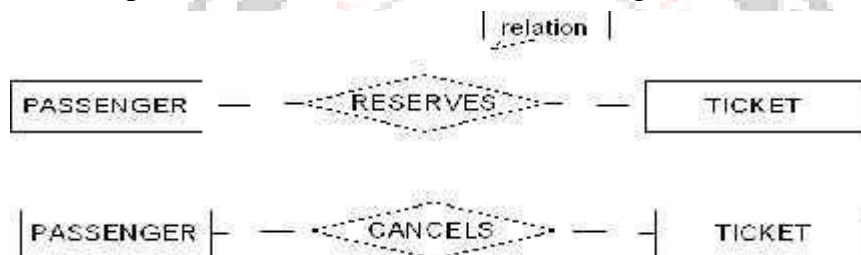
Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationship (total/partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

#### Definitions:

**The cardinality ratio:** - for a binary relationship specifies the maximum number of relationships that an entity can participate in.



**Relationship:** - it is defined as an association among two or more entities.



**Weak and strong entity:** - an entity set may not have sufficient attributes to form a primary key. Such an entity set is termed a weak entity set. An entity set that has primary key is termed a strong entity set.

#### Total participation:-

Ex: - if a travel agency states that every passenger must make reservation then every passenger travels in bus. Then a passengers entity can exist only if it participates in atleast one travels relationship instances. Thus the participation of passenger in travel is called total participation meaning that every entity in the “total set” passenger entities must be related to bus via travels relationship.



All passengers travel in one bus so it is total participation

**Partial participation:** a participation that is not total is called as partial participation.



Some passengers cancel ticket so it is partial participation

**Generalization:** consists of identifying some common characteristics of a collection of entity set and creating new entity set that contains entities possessing these common characteristics.

**Aggregation:** allows us to indicate that a relationship set participates in another relationship set.

**Specialization:** in the process of identifying subsets of an entity set (the super set) that share some distinguishing characteristics. This entity type is called the super class of the specialization.

### Relationship between different entities:

#### *Relationship between Bus and Ticket entities*

1:M binary relationship



#### *Relationship between Passenger and Bus entities*

M:1 binary relationship

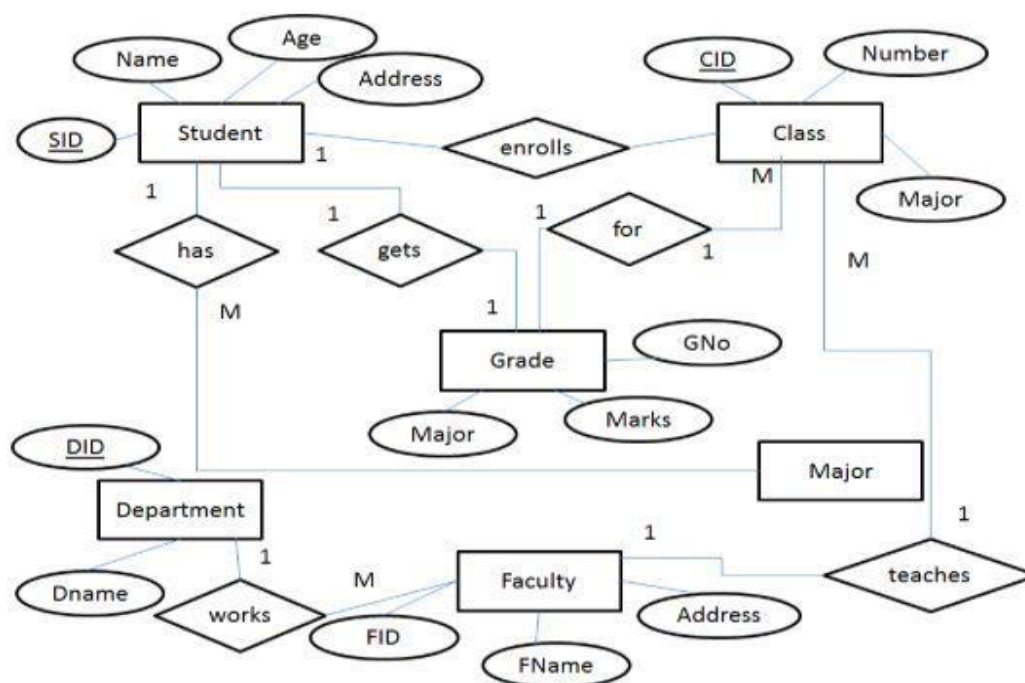


#### *Relationship between Passenger and Ticket entities*

M:N binary relationship

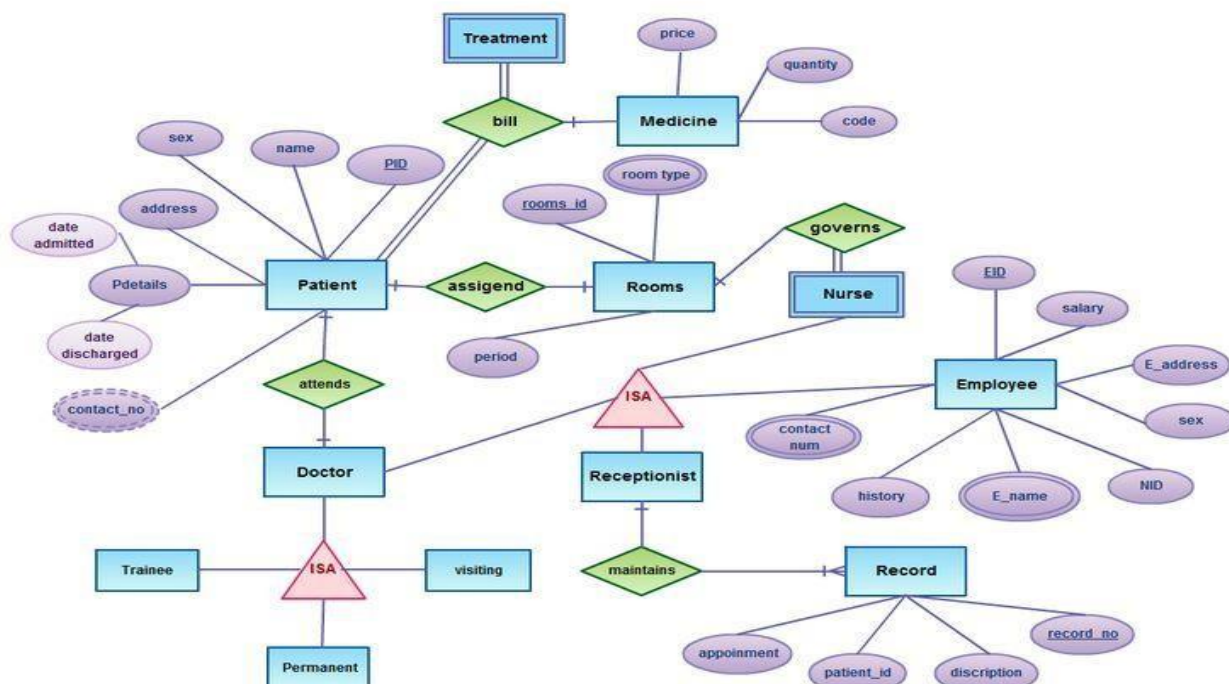


## 1. Draw the ER Diagram for UNIVERSITY



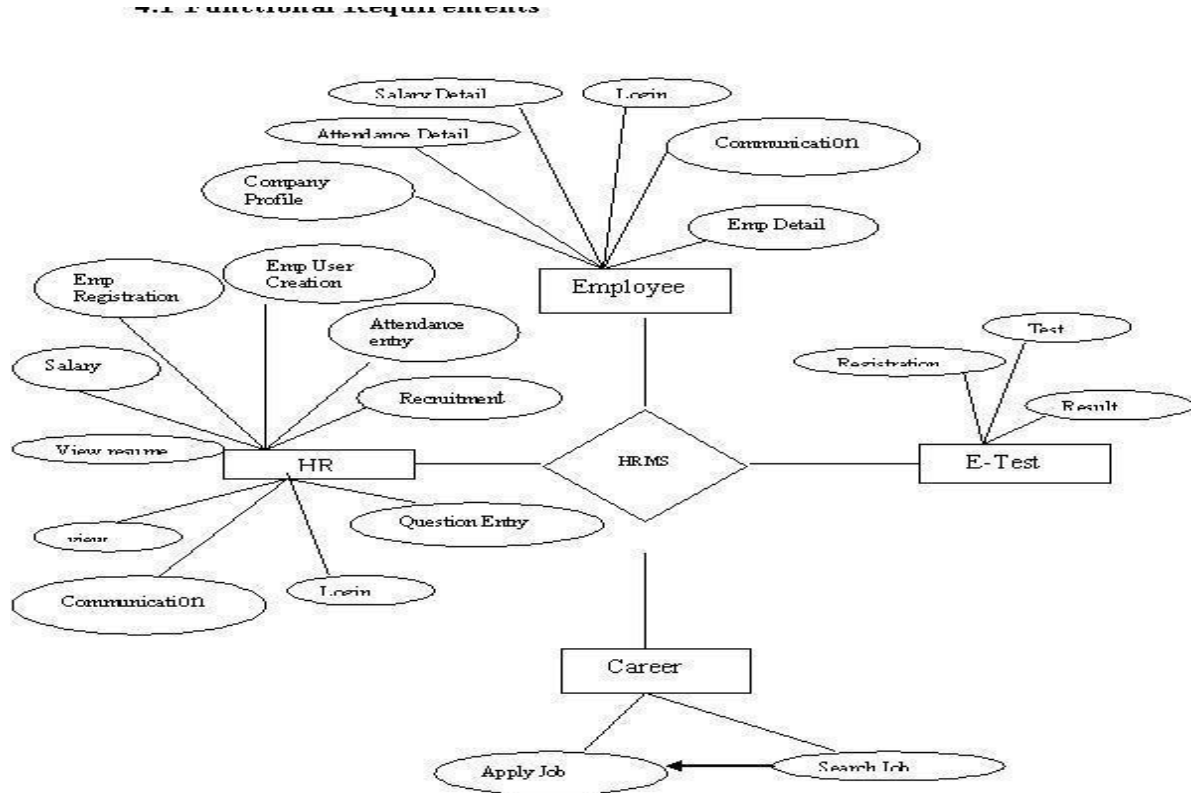
## 2. Draw the ER Diagram for Hospital management Systems.

E-R Diagram for Hospital Management System

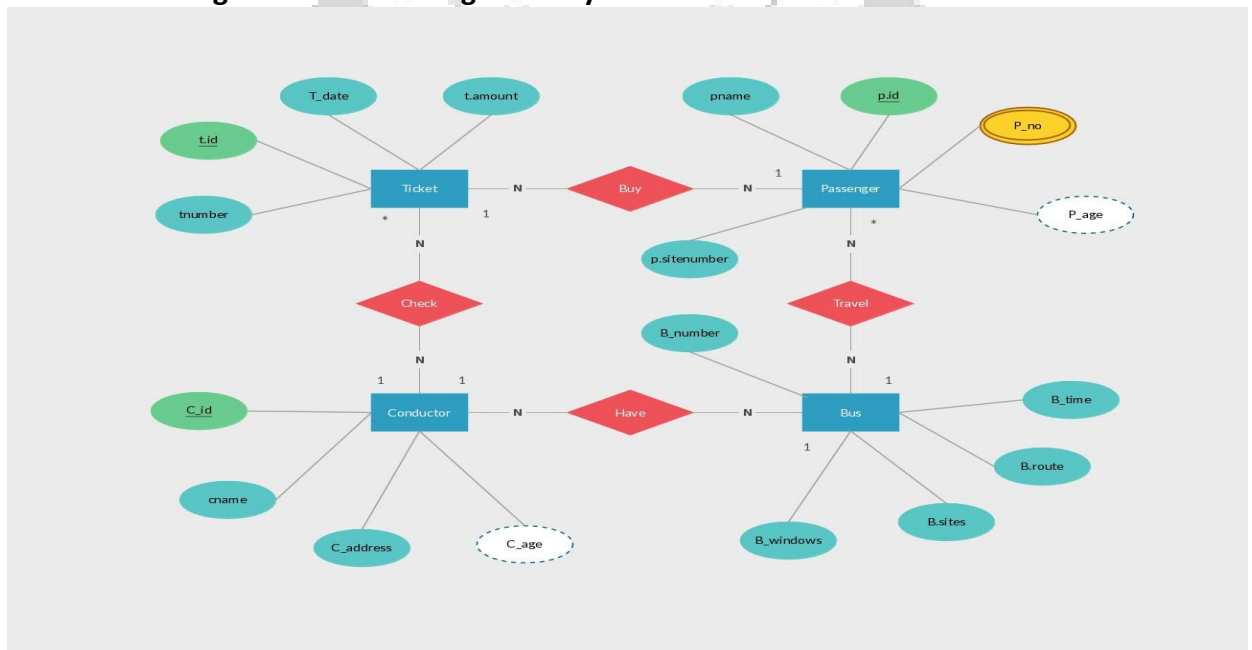


### 3. Draw the ER diagram for HR management systems

4



### 4. Draw the ER diagram for HR management systems



## **VIVA QUESTIONS AND ANSWERS**

### **1. What is database?**

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

### **2. What is DBMS?**

DBMS stands for Database Management System. It is a collection of programs that enables user to create and maintain a database.

### **3. What is a Database system?**

The database and DBMS software together is called as Database system.

### **4. What are the advantages of DBMS?**

- I. Redundancy is controlled.
- II. Providing multiple user interfaces.
- III. Providing backup and recovery
- IV. Unauthorized access is restricted.
- V. Enforcing integrity constraints.

### **5. What is normalization?**

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties  
(1).Minimizing redundancy, (2). Minimizing insertion, deletion and update anomalies.

### **6. What is Data Model?**

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

### **7. What is E-R model?**

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

### **8. What is Object Oriented model?**

This model is based on collection of objects. An object contains values stored in instance variables with in the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

### **9. What is an Entity?**

An entity is a thing or object of importance about which data must be captured.

### **10. What is DDL (Data Definition Language)?**

A data base schema is specifies by a set of definitions expressed by a special language called DDL.

### **11. What is DML (Data Manipulation Language)?**

This language that enable user to access or manipulate data as organised by appropriate data model. Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data. Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data

### **12. What is DML Compiler?**

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

### **13. What is Query evaluation engine?**

It executes low-level instruction generated by compiler.

### **14. Under what conditions should indexes be used?**

Indexes can be created to enforce uniqueness, to facilitate sorting, and to enable fast retrieval by column values. A good candidate for an index is a column that is frequently used with equal conditions in WHERE clauses.

### **15. What is difference between SQL and SQL SERVER?**

SQL is a language that provides an interface to RDBMS, developed by IBM. SQL SERVER is a RDBMS just like Oracle, DB2.



## WEEK: 2

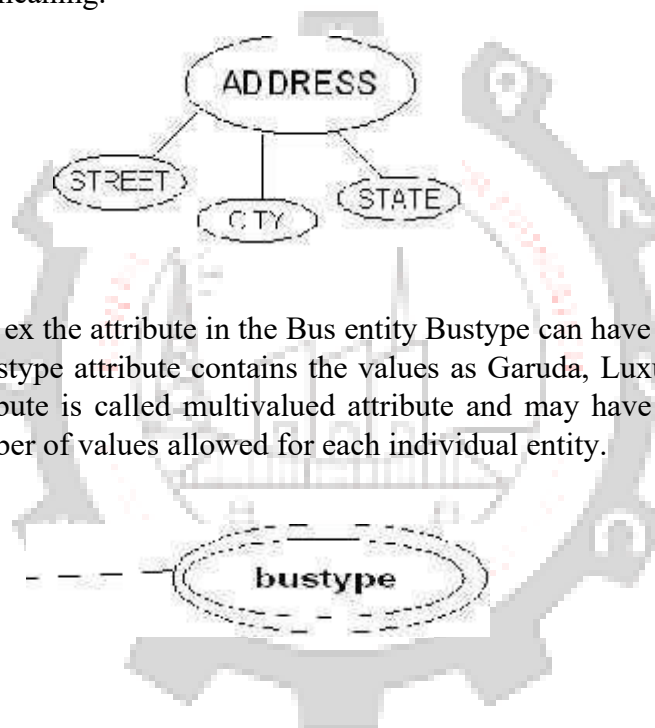
### Relational Model

#### Relational model

Represent all entities (strong, weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing as tables based on the cardinality. Represent attributes as columns in the tables or as tables based on the requirement. Different types of attributes (composite, multivalued and derived).

#### Definitions:

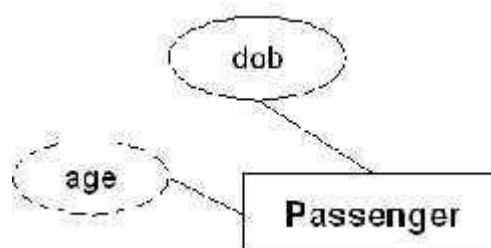
**Composite attributes:** can be divided into smaller sub parts which represent more basic attributes with independent meaning.



**Multivalued attributes:** for ex the attribute in the Bus entity Bustype can have different types of buses according that the Bustype attribute contains the values as Garuda, Luxury, Express, and Ordinary. This type of attribute is called multivalued attribute and may have lower and upper bounds to constrain the number of values allowed for each individual entity.

#### Derived attributes:

In some cases, two or more attribute values are related. With the help of one attribute we get the value of another attribute. Age and DOB attributes. With the DOB we get the age of the person to the current date.





## Relational Model concept

Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.

**Domain:** It contains a set of atomic values that an attribute can take.

**Attribute:** It contains the name of a column in a particular table. Each attribute  $A_i$  must have a domain,  $dom(A_i)$

### Relational Model concept

Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.

**Domain:** It contains a set of atomic values that an attribute can take.

**Attribute:** It contains the name of a column in a particular table. Each attribute  $A_i$  must have a domain,  $dom(A_i)$

**Relational instance:** In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

**Relational schema:** A relational schema contains the name of the relation and name of all columns or attributes.

**Relational key:** In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

**Example: STUDENT Relation**

NAME	ROLL_NO	PHONE_NO	ADDRESS	AGE
Ram	14795	7305758992	Noida	24
Shyam	12839	9026288936	Delhi	35
Laxman	33289	8583287182	Gurugram	20
Mahesh	27857	7086819134	Ghaziabad	27
Ganesh	17282	9028 9i3988	Delhi	40

- In the given table, NAME, ROLL\_NO, PHONE\_NO, ADDRESS, and AGE are the attributes.
- The instance of schema STUDENT has 5 tuples.
- t3 = <Laxman, 33289, 8583287182, Gurugram, 20>

### Integrity constraints

- NOT NULL
- UNIQUE
- DEFAULT
- CHECK
- Key Constraints - PRIMARY KEY, FOREIGN KEY
- Domain constraints

#### **NOT NULL:**

NOT NULL constraint makes sure that a column does not hold NULL value. When we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default. By specifying NULL constraint, we can be sure that a particular column(s) cannot have NULL values.

#### **UNIQUE:**

UNIQUE Constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.

#### **DEFAULT:**

The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

#### **CHECK:**

This constraint is used for specifying range of values for a particular column of a table. When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

### Key constraints:

#### **PRIMARY KEY:**

[Primary key](#) uniquely identifies each record in a table. It must have unique values and cannot contain nulls. In the below example the ROLL\_NO field is marked as primary key, that means the ROLL\_NO field cannot have duplicate and null values.

#### **FOREIGN KEY:**

Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

Read more about it [here](#).

### ***Domain constraints:***

Each table has certain set of columns and each column allows a same type of data, based on its data type. The column does not accept values of any other data type.

```
Example:create table student(sid varchar(30)primary key, sname  
char(20) unique,age int(20) check(age>16),city varchar(20) default  
'hyderabad')
```

```
Create table department(did varchar(20) primary key,dname  
char(20),sid varchar(30),Foreign key(sid)references student(sid));
```

### **Company database**

```
CREATE TABLE employee (  
  emp_id INT PRIMARY KEY,  
  first_name VARCHAR(40),  
  last_name VARCHAR(40),  
  birth_day DATE,  
  sex VARCHAR(10),  
  salary INT check(salary>0),  
  super_id INT,  
  branch_id INT);
```

```
CREATE TABLE branch (  
  branch_id INT PRIMARY KEY,  
  branch_name VARCHAR(40),  
  mgr_id INT,  
  mgr_start_date DATE,  
  FOREIGN KEY(mgr_id) REFERENCES employee(emp_id));
```

```
CREATE TABLE client (  
  client_id INT PRIMARY KEY,  
  client_name VARCHAR(40),  
  branch_id INT,  
  FOREIGN KEY(branch_id) REFERENCES branch(branch_id));
```

## **VIVA QUESTIONS AND ANSWERS**

### **1. What is Specialization?**

It is the process of defining a set of subclasses of an entity type where each subclass contain all the attributes and relationships of the parent entity and may have additional attributes and relationships which are specific to itself.

### **2. What is generalization?**

It is the process of finding common attributes and relations of a number of entities and defining a common super class for them.

### **3. What is meant by Proactive, Retroactive and Simultaneous Update?**

Proactive Update: The updates that are applied to database before it becomes effective in real world.

Retroactive Update: The updates that are applied to database after it becomes effective in real world.

Simultaneous Update: The updates that are applied to database at the same time when it becomes effective in real world.

### **4. What is RAID Technology?**

Redundant array of inexpensive (or independent) disks. The main goal of raid technology is to even out the widely different rates of performance improvement of disks against those in memory and microprocessor. Raid technology employs the technique of data striping to achieve higher transfer rates.

### **5. What are serial, non serial schedule?**

A schedule S is serial if, for every transaction T participating in the schedule, all the operations of T is executed consecutively in the schedule, otherwise, the schedule is called non-serial schedule.

### **6. What are conflict serializable schedules?**

A schedule S of n transactions is serializable if it is equivalent to some serial schedule of the same n transactions.

### **7. What is view serializable?**

A schedule is said to be view serializable if it is view equivalent with some serial schedule.

### **8. What is a foreign key?**

A key of a relation schema is called as a foreign key if it is the primary key of some other relation to which it is related to.

## WEEK: 3

### Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in doing so, to safeguard the database against certain types of logical or structural problems namely data anomalies.

The normalization forms are:

- 1) **First Normal Form:** 1NF requires that the values in each column of a table are atomic. By atomic we mean that there are no sets of values within a column.
- 2) **Second Normal Form:** where the 1NF deals with atomicity of data, the 2NF deals with relationships between composite key columns and non-key columns. To achieve 2NF the tables should be in 1NF. The 2NF any non-key columns must depend on the entire primary key.
- 3) **Third Normal Form:** 3NF requires that all columns depend directly on the primary key. Tables violate the third normal form when one column depends on another column, which in turn depends on the primary key (transitive dependency). One way to identify transitive dependency is to look at your tables and see if any columns would require updating if another column in the table was updated. If such a column exists, it probably violates 3NF.

#### 1NF

Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP\_PHONE.

**EMPLOYEE table:**

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389, 8589830302	Punjab

Relation can be converted into 1NF as follows

## Employee\_details

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385	UP
14	John	9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389	Punjab
12	Sam	8589830302	Punjab

## 2NF

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

## Relation not in 2nf

### TEACHER table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

Convert the above relation into 2NF as follows

Create the teacher\_detail table with teacher\_id as primary key

**TEACHER\_DETAIL table:**

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

Create teacher\_subject table with teacher\_id as foreign key

**TEACHER\_SUBJECT table:**

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

### 3NF

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency  $X \rightarrow Y$ .

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.



**EMPLOYEE\_DETAIL table:**

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

The above table can be converted into 3NF as follows

Create the employee table with emp\_id as primary key

**EMPLOYEE table:**

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

Create employee\_zip table with emp\_zip as primary key

**EMPLOYEE\_ZIP table:**

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

## VIVA QUESTIONS AND ANSWERS

### 1. What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs)

and primary key to achieve the properties

(1).Minimizing redundancy, (2). Minimizing insertion, deletion and update anomalies.

### 2. What is Functional Dependency?

A Functional dependency is denoted by  $X \rightarrow Y$  between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if  $t1[X] = t2[X]$  then they have  $t1[Y] = t2[Y]$ . This means the value of X component of a tuple uniquely determines the value of component Y.

### 3. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

### 4. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

### 5. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency  $X \rightarrow Y$  is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

### 6. What is 2NF?

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

### 7. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD  $X \rightarrow A$  either of the following is true

1. X is a Super-key of R.

2. A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

### 8. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD  $X \rightarrow A$ , X must be a candidate key.

### 9. Define Entity Set.

Compilation of all entries of any particular type of entry in the database is called Entity Set.

### 10. What do you mean by Entity type extension?

Compilation of similar entity types into one particular type which is grouped together as an entity set.

### 11. What is 4NF?

A relation schema R is said to be in 4NF if for every Multivalued dependency X Y that holds over R, one of following is true

- X is subset or equal to (or)  $XY = R$ .
- X is a super key.

### 12. What is 5NF?

A Relation schema R is said to be 5NF if for every join dependency  $\{R_1, R_2, \dots, R_n\}$  that holds R, one the following is true

- $R_i = R$  for some i .
- The join dependency is implied by the set of FD, over R in which the left side is key of R.

### 13. What is Domain-Key Normal Form?

A relation is said to be in DKNF if all constraints and dependencies that should hold on the the constraint can be enforced by simply enforcing the domain constraint and key constraint on the relation.

### 14. What are partial, alternate,, artificial, compound and natural key?

#### 1. 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.

#### 2. Alternate Key:

All Candidate Keys excluding the Primary Key are known as Alternate Keys.

#### 3. Artificial Key:

If no obvious key, either stand alone or compound is available, then the last resort is to simply create a key, by assigning a unique number to each record or occurrence. Then this is known as developing an artificial key.

#### 4. Compound Key:

If no single data element uniquely identifies occurrences within a construct, then combining multiple elements to create a unique identifier for the construct is known as creating a compound key.

Natural Key:

When one of the data elements stored within a construct is utilized as the primary key, then it is called the natural key.

## WEEK: 4

### Practicing DDL commands

#### Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. **CREATE** It is used to create a new table in the database.

##### Syntax:

```
CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);
```

##### Example:

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);
```

b. **DROP**: It is used to delete both the structure and record stored in the table.

##### Syntax

```
DROP TABLE ;
```

##### Example

```
DROP TABLE EMPLOYEE;
```

c. **ALTER**: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

##### Syntax:

To add a new column in the table

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

To modify existing column in the table:

```
ALTER TABLE MODIFY(COLUMN DEFINITION....);
```

##### EXAMPLE

```
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));  
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));
```

d. **TRUNCATE:** It is used to delete all the rows from the table and free the space containing the table.

**Syntax:**

```
TRUNCATE TABLE table_name;
```

**Example:**

```
TRUNCATE TABLE EMPLOYEE;
```



## WEEK: 5

### Practicing DML commands

#### Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

**a. INSERT:** The INSERT statement is a SQL query. It is used to insert data into the row of a table.

**Syntax:**

```
INSERT INTO TABLE_NAME (col1, col2, col3,.... col N) VALUES (value1, value2, value3,..... valueN);
```

Or

```
INSERT INTO TABLE_NAME VALUES (value1, value2, value3,.....valueN);
```

**For example:**

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

**b. UPDATE:** This command is used to update or modify the value of a column in the table.

**Syntax:**

```
UPDATE table_name SET [column_name1= value1, column_nameN = valueN] [WHERE CONDITION]
```

**For example:**

```
UPDATE students  
SET User_Name = 'Sonoo'  
WHERE Student_Id = '3'
```

**c. DELETE:** It is used to remove one or more row from a table.

**Syntax:**

```
DELETE FROM table_name [WHERE condition];
```

**For example:**

```
DELETE FROM javatpoint WHERE Author="Sonoo";
```

## VIVA QUESTIONS AND ANSWERS



### 1) What is DDL (Data Definition Language)?

A data base schema is specifies by a set of definitions expressed by a special language called DDL.

### 2. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

### 3. What is SDL (Storage Definition Language)?

This language is to specify the internal schema. This language may specify the mapping between two schemas.

### 4. What is Data Storage – Definition Language?

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage-definition language.

### 5. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organised by appropriate data model.

1. Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data.
2. Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data.

### 6. What is DML Compiler?

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

### 7. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

### 8. What is DDL Interpreter?

It interprets DDL statements and record them in tables containing metadata.

### 9. What is Record-at-a-time?

The Low level or Procedural DML can specify and retrieve each record from a set of records. This retrieve of a record is said to be Record-at-a-time.

### 10. What is Set-at-a-time or Set-oriented?

The High level or Non-procedural DML can specify and retrieve many records in a single DML statement. This retrieve of a record is said to be Set-at-a-time or Set-oriented.

### 11. What is Relational Algebra?

It is procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

### 12. What is Relational Calculus?

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. E.g. of languages based on it are DSL ALPHA, QUEL.



**13. How does Tuple-oriented relational calculus differ from domain-oriented relational calculus?**

1. The tuple-oriented calculus uses a tuple variables i.e., variable whose only permitted values are tuples of that relation. E.g. QUEL
2. The domain-oriented calculus has domain variables i.e., variables that range over the underlying domains instead of over relation. E.g. ILL, DEDUCE.

**14. What is a Phantom Deadlock?**

In distributed deadlock detection, the delay in propagating local information might cause the deadlock detection algorithms to identify deadlocks that do not really exist. Such situations are called phantom deadlocks and they lead to unnecessary aborts.

**15. What is a checkpoint and When does it occur?**

A Checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS can reduce the amount of work to be done during restart in the event of subsequent crashes.

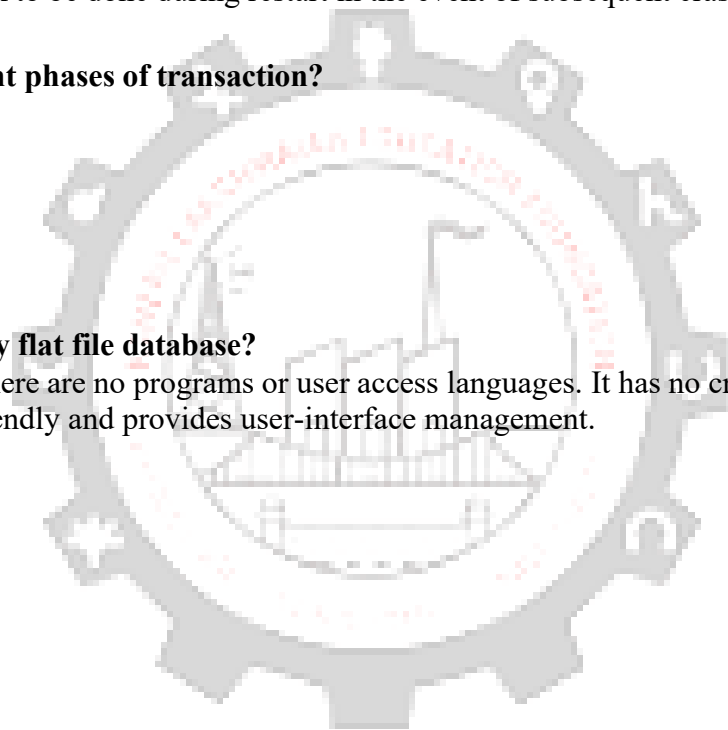
**16. What are the different phases of transaction?**

Different phases are

1. Analysis phase
2. Redo Phase
3. Undo phase

**17. What do you mean by flat file database?**

It is a database in which there are no programs or user access languages. It has no cross-file capabilities but is user-friendly and provides user-interface management.



## WEEK: 6

### Querying (using ANY, IN, NOT IN UNION UNION ALL.)

IN, NOT IN operators in SQL are used with SELECT, UPDATE and DELETE statements/queries to select, update and delete only particular records in a table those meet the condition given in WHERE clause and conditions given in IN, NOT IN operators. I.e. it filters records from a table as per the condition. Syntax for SQL IN & NOT IN operators are given below.

#### SQL SYNTAX FOR AND, OR OPERATORS:

Syntax for SQL <b>IN</b> operator	SELECT column_name1, column_name2, etc FROM table_name WHERE column_name1 IN (value1, value2, etc);
Syntax for SQL <b>NOT IN</b> operator	SELECT column_name1, column_name2, etc FROM table_name WHERE column_name1 NOT IN (value1, value2, etc);

#### UNION

The **UNION** command combines the result set of two or more SELECT statements (only distinct values)

The following SQL statement returns the cities (only distinct values) from both the "Customers" and the "Suppliers" table:

#### Example

```
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;
```

#### UNION ALL

The **UNION ALL** command combines the result set of two or more SELECT statements (allows duplicate values).

The following SQL statement returns the cities (duplicate values also) from both the "Customers" and the "Suppliers" table:

#### Example

```
SELECT City FROM Customers
UNION ALL
SELECT City FROM Suppliers
ORDER BY City;
```

```
mysql> create database nested;
```

```
mysql> use nested;
```

Database changed

```
mysql> select * from employee;
```

```
+---+-----+-----+-----+
| eid | ename  | dept  | salary |
+---+-----+-----+-----+
| 1 | rahul  | hr    | 10000  |
| 2 | ramesh | mrkt  | 20000  |
| 3 | rajat  | hr    | 30000  |
| 4 | rakesh | marketing | 40000  |
| 5 | rharshit | it    | 50000  |
| 6 | himesh | marketing | 60000  |
+---+-----+-----+-----+
```

6 rows in set (0.00 sec)

```
mysql> select * from project;
```

```
+---+-----+-----+-----+
| eid | pid | pname  | plocation |
+---+-----+-----+-----+
| 1 | p1 | iot    | hyd    |
```

```
| 5 | p2 | android | pune |
| 4 | p3 | networking | bengalore |
| 4 | p3 | database | mangalore |
| 3 | p4 | database | mangalore |
```

```
+----+----+-----+-----+
```

5 rows in set (0.00 sec)

### 1) IN

The IN operator is used when you want to retrieve a column that has entries in the table or referencing table.

#### Syntax

expression IN (value1, value2, ..... value\_n);

#### query

- 1) select \* from employee where ename In('ramesh','mahesh','suresh');
- 2) select ename from employee where eid In(select eid from project where employee.eid=project.eid);
- 3) select ename,salary from employee where eid In(select eid from project where employee.eid=project.eid);

### 2) NOT IN

The NOT IN operator is used when you want to retrieve a column that has no entries in the table or referencing table.

```
select * from employee where ename Not In('ramesh','mahesh','suresh');
```

### 3) ANY

The ANY operator returns true if any of the subquery values meet the condition.

#### Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
(SELECT column_name FROM table_name WHERE condition);
```

```
select * from employee where eid=ANY(select eid from project where employee.eid=project.eid);
```

### 4) All

The ALL operator returns true if all of the subquery values meet the condition.

## ALL Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
(SELECT column_name FROM table_name WHERE condition);
```

## UNION

The UNION operator is used to combine the result-set of two or more SELECT statements.

- Each SELECT statement within UNION must have the same number of columns
- The columns must also have similar data types
- The columns in each SELECT statement must also be in the same order

## UNION Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

Example

```
Create table emp1(eid int,ename varchar(20),address varchar(20),salary int(20));
```

```
Create table emp2(eid int,ename varchar(20),address varchar(20),salary int(20));
```

```
mysql> select * from emp1;
```

eid	ename	address	salary
1	rahul	pune	10000
2	rajat	hyd	20000
3	rakesh	mangalore	30000
4	harshit	umerga	40000
5	harshit	mangalore	30000
6	somesh	solapur	50000
1	saket	aaa	10000
2	sanket	bbb	20000

```
mysql> select * from emp2;
```

eid	ename	address	salary
1	saket	aaa	10000
2	sanket	bbb	20000
3	rishi	ccc	40000
4	sanket	ddd	50000
1	saket	aaa	10000
2	sanket	bbb	20000

```
mysql> select eid,ename from emp1 union select eid,ename from emp2;
```

```
mysql> select eid,ename from emp1 union all select eid,ename from emp2;
```

## VIVA QUESTIONS AND ANSWERS

### 1. What is a query?

A query with respect to DBMS relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

### 2. What do you mean by Correlated subquery ?

Subqueries , or nested queries, are used to bring back a set of rows to be used by the parent query. Depending on how the subquery is written, it can be executed once for the parent query or it can be executed once for each row returned by the parent query. If the subquery is executed for each row of the parent, this is called a correlated subquery .

A correlated subquery can be easily identified if it contains any references to the parent subquery columns in its WHERE clause. Columns from the subquery cannot be referenced anywhere else in the parent query. The following example demonstrates a non-correlated subquery .

E.g. Select \* From CUST Where '10/03/1990' IN (Select ODATE From ORDER Where CUST.CNUM = ORDER.CNUM)

### 3. What are the primitive operations common to all record management systems?

Addition, deletion and modification.

### 4. Name the buffer in which all the commands that are typed in are stored

'Edit' Buffer

### 5. What are the unary operations in Relational Algebra?

PROJECTION and SELECTION.

**6. Are the resulting relations of PRODUCT and JOIN operation the same?**

No.

**PRODUCT:** Concatenation of every row in one relation with every row in another.

**JOIN:** Concatenation of rows from one relation and related rows from another.

**7. What is RDBMS KERNEL?**

Two important pieces of RDBMS architecture are the kernel, which is the software, and the data dictionary, which consists of the system-level data structures used by the kernel to manage the database

You might think of an RDBMS as an operating system (or set of subsystems), designed specifically for controlling data access; its primary functions are storing, retrieving, and securing data. An RDBMS maintains its own list of authorized users and their associated privileges; manages memory caches and paging; controls locking for concurrent resource usage; dispatches and schedules user requests; and manages space usage within its table-space structures .

**8. Name the sub-systems of a RDBMS**

I/O, Security, Language Processing, Process Control, Storage Management, Logging and Recovery, Distribution Control, Transaction Control, Memory Management, Lock Management

**9. Which part of the RDBMS takes care of the data dictionary? How**

Data dictionary is a set of tables and database objects that is stored in a special area of the database and maintained exclusively by the kernel.

**10. What is the job of the information stored in data-dictionary?**

The information in the data dictionary validates the existence of the objects, provides access to them, and maps the actual physical storage location.

**11. Not only RDBMS takes care of locating data it also**

determines an optimal access path to store or retrieve the data

**12. How do you communicate with an RDBMS?**

You communicate with an RDBMS using Structured Query Language (SQL)

**13. Define SQL and state the differences between SQL and other conventional programming Languages**

SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

**14. Name the three major set of files on disk that compose a database in Oracle**

There are three major sets of files on disk that compose a database. All the files are binary. These are

- Database files
- Control files
- Redo logs

The most important of these are the database files where the actual data resides. The control files and the redo logs support the functioning of the architecture itself.

All three sets of files must be present, open, and available to Oracle for any data on the database to be useable.

Without these files, you cannot access the database, and the database administrator might have to recover some or all of the database using a backup, if there is one.



## WEEK: 7

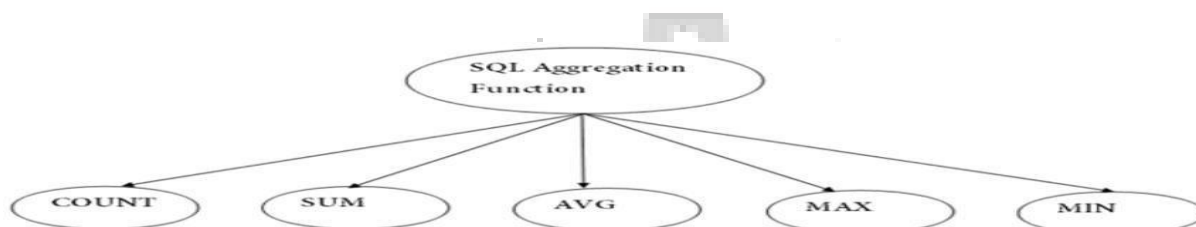
Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.

Create the tables in the database with different attributes insert the records in the tables and perform the execution of the following functions

### SQL Aggregate Functions

- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
- It is also used to summarize the data.

### Types of SQL Aggregation Function



#### 1. COUNT FUNCTION

- COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
- COUNT function uses the COUNT(\*) that returns the count of all the rows in a specified table. COUNT(\*) considers duplicate and Null.

#### Syntax

**COUNT(\*)**

or

**COUNT( [ALL|DISTINCT] expression )**

Examples

- 1) `SELECT COUNT(*)`  
`FROM PRODUCT_MAST;`
- 2) `SELECT COUNT(*)`  
`FROM PRODUCT_MAST;`  
`WHERE RATE>=20;`

#### Example: COUNT() with DISTINCT

```

SELECT COUNT(DISTINCT COMPANY)

FROM PRODUCT_MAST;
  
```

## 2. SUM Function

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

### Syntax

```
SUM()  
or  
SUM( [ALL|DISTINCT] expression )
```

### Example: SUM()

```
SELECT SUM(COST)  
FROM PRODUCT_MAST;
```

### Example: SUM() with WHERE

```
SELECT SUM(COST)  
FROM PRODUCT_MAST  
WHERE QTY>3;
```

## 3. AVG function

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

### Syntax

```
AVG()
```

### Example:

```
SELECT AVG(COST)  
FROM PRODUCT_MAST;
```

## 4. MAX Function

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

### Syntax

```
MAX()
```

### Example:

```
SELECT MAX(RATE)  
FROM PRODUCT_MAST;
```

## 5. MIN Function

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

### Syntax

```
MIN()  
SELECT MIN(RATE)  
FROM PRODUCT_MAST;
```

## Group By

The MYSQL GROUP BY Clause is used to collect data from multiple records and group the result by one or more column. It is generally used in a SELECT statement.

### Syntax:

```
SELECT expression1, expression2, ... expression_n,  
aggregate_function (expression)  
FROM tables  
[WHERE conditions]  
GROUP BY expression1, expression2, ... expression_n;
```

### Examples

```
SELECT address, COUNT(*)  
FROM officers  
GROUP BY address;
```

```
SELECT emp_name, SUM(working_hours) AS "Total working hours"  
FROM employees  
GROUP BY emp_name;
```

## MySQL HAVING Clause

MySQL HAVING Clause is used with GROUP BY clause. It always returns the rows where condition is TRUE.

### Syntax:

```
SELECT expression1, expression2, ... expression_n,  
aggregate_function (expression)  
FROM tables  
[WHERE conditions]  
GROUP BY expression1, expression2, ... expression_n  
HAVING condition;
```

### **example**

```
SELECT emp_name, SUM(working_hours) AS "Total working hours"  
FROM employees  
GROUP BY emp_name  
HAVING SUM(working_hours) > 5;
```

### **CREATE VIEW Statement**

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

### **CREATE VIEW Syntax**

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Example

```
CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = "Brazil";  
  
SELECT * FROM [Brazil Customers];
```

### **SQL Dropping a View**

A view is deleted with the DROP VIEW command.

### **SQL DROP VIEW Syntax**

```
DROP VIEW view_name;
```

**95. Select 'NORTH', CUSTOMER From CUST\_DTLS Where REGION = 'N' Order By CUSTOMER Union Select 'EAST', CUSTOMER From CUST\_DTLS Where REGION = 'E' Order By CUSTOMER**

The above is

- a) Not an error
- b) Error – the string in single quotes 'NORTH' and 'SOUTH'
- c) Error – the string should be in double quotes
- d) Error – ORDER BY clause

**(d) Error – the ORDER BY clause. Since ORDER BY clause cannot be used in UNIONS**

**1. What is Storage Manager?**

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

**2. What is Buffer Manager?**

It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory.

**3. What is Transaction Manager?**

It is a program module, which ensures that database, remains in a consistent state despite system failures and concurrent transaction execution proceeds without conflicting.

**4. What is File Manager?**

It is a program module, which manages the allocation of space on disk storage and data structure used to represent information stored on a disk.

**5. What is Authorization and Integrity manager?**

It is the program module, which tests for the satisfaction of integrity constraint and checks the authority of user to access data.

**6. What are the four objectives of the selection of a data type?**

A data type should be selected so that all possible values are represented using minimal storage space. The data type should help to ensure data integrity and support all possible data manipulations (i.e., cannot place a letter in a field such as salary where a number is required).

**7. Describe the four types of indexes.**

A unique primary index is unique and is used to find and store a row. A nonunique primary index is not unique and is used to find a row but also where to store a row (based on its unique primary index). A unique secondary index is unique for each row and used to find table rows. A nonunique secondary index is not unique and used to find table rows.

**8. What is denormalization and why would someone consider doing so?**

Denormalization is the process of taking normalized relations and changing them so that they are not longer normalized. This process may lead to anomalies and create data redundancy as negative consequences. However, the revised relations should improve database performance.

**9. Compare a hierarchical and network database model?**

The hierarchical model is a top-down structure where each parent may have many children but each child can have only one parent. This model supports one-to-one and one-to-many relationships. The network model can be much

more flexible than the hierarchical model since each parent can have multiple children but each child can also have multiple parents. This model supports one-to-one, one to- many, and many-to-many relationships.

**10. Describe the differences between vertical and horizontal portioning.**

Horizontal portioning is where the rows in a relation are separated by some criteria and placed into a new relation or file with the same layout as the original relation (in this case only the records in each file differ). Vertical portioning is where the columns in a relation are separated by some criteria and placed into a new relation or file with a different layout as the original relation.

**11. Explain the difference between a dynamic and materialized view.**

A dynamic view may be created every time that a specific view is requested by a user. A materialized view is created and or updated infrequently and it must be synchronized with its associated base table(s).

**12. Discuss some of the techniques that can be used to tune operational performance.**

Choosing primary and secondary keys can increase the speed of row selection, joining, and row ordering. Selecting the appropriate file organization for base tables and indexes can also improve performance. Clustering related rows together and maintaining statistics about tables and indexes can lead to increased efficiency.

**13. Briefly describe the three types of SQL commands.**

Data definition language commands are used to create, alter, and drop tables. Data manipulation commands are used to insert, modify, update, and query data in the database. Data control language commands help the DBA to control the database.

**14. What are the steps to follow when preparing to create a table?**

1. Identify the data type, length, and precision for each attribute. 2. Identify the columns that can accept a null value. 3. Identify the columns that need to be unique. 4. Identify primary and related foreign keys with the parent table being created before the child. 5. Determine default values. 6. Determine where the domain values are that need to be constrained. 7. Create the indexes.

**15. What are some disadvantages of a standard language such as SQL?**

A standard language can hinder the effort to create a new language. One standard is never enough to meet all of the business needs. A standard can be a compromise among interested parties which can cause the standard to not be ideal. If a standard is altered by some, than portability between

**16. Explain a join between tables**

A join allows tables to be linked to other tables when a relationship between the tables exists. The relationships are established by using a common column in the tables and often uses the primary/foreign key relationship.

**17. Describe and contrast a trigger and a procedure.**

Triggers are stored and controlled in the DBMS. A trigger is executed automatically when a condition is met (INSERT, UPDATE, or DELETE). A procedure is also stored in a database. A procedure is not executed automatically.

**18. Briefly describe an outer join.**

An outer join includes the records that match and those that do not have a matching value in another table. Outer joins can be a LEFT outer join (includes all records from the first table listed) or a RIGHT outer join (includes all records from the second table listed). Outer joins are not easily used with more than two tables.

**19. Describe a subquery.**

A subquery is a query that is composed of two queries. The first query (inner query) is within the WHERE clause of the other query (outer query). In some cases the inner query provides results for the outer query to process. In other cases, the outer query results provide results for the inner query (correlated subquery).

## WEEK: 8

### Triggers

#### Triggers

Triggers are stored programs, which are automatically executed or fired when some event occurs.

#### **Syntax for creating trigger:**

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
    Declaration-statements
BEGIN
    Executable-statements
EXCEPTION
    Exception-handling-statements
END;
```

```
mysql> use triggers;
```

Database changed

```
mysql> create table student(sid int(20) primary key,sname varchar(20),address varchar(20),marks
int(20));
```

Query OK, 0 rows affected (0.17 sec)

```
mysql> create trigger sample_insert before insert on student for each row set new.marks=new.marks+5;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> insert into student values(1,'aaa','pune',76,'patil');
```

Query OK, 1 row affected (0.06 sec)

```
mysql> insert into student values(2,'bbb','hyd',70,'rao');
```

Query OK, 1 row affected (0.06 sec)

```
mysql> insert into student values(3,'ccc','chd',36,'balude');
```

Query OK, 1 row affected (0.07 sec)



```
mysql> insert into student values(4,'ddd','delhi',35,'malude');
```

Query OK, 1 row affected (0.06 sec)

```
mysql> select * from student;
```

```
+---+-----+-----+-----+-----+
```

```
| sid | sname | address | marks | lname |
```

```
+---+-----+-----+-----+-----+
```

```
| 1 | aaa | pune | 81 | patil |
```

```
| 2 | bbb | hyd | 75 | rao |
```

```
| 3 | ccc | chd | 41 | balude |
```

```
| 4 | ddd | delhi | 40 | malude |
```

```
+---+-----+-----+-----+-----+
```

4 rows in set (0.00 sec)

```
mysql> create table doctor(did int(30) primary key,dname varchar(20),address varchar(20),age
int(20),city varchar(20));
```

Query OK, 0 rows affected (0.10 sec)

```
mysql> insert into doctor values(1,'ramesh','himayatnagar',34,'hyderabad');
```

Query OK, 1 row affected (0.07 sec)

```
mysql> insert into doctor values(2,'jogit','dilsukhnagar',35,'hyderabad');
```

Query OK, 1 row affected (0.37 sec)

```
mysql> insert into doctor values(3,'gupta','mehandipatnam',37,'hyderabad');
```

Query OK, 1 row affected (0.06 sec)

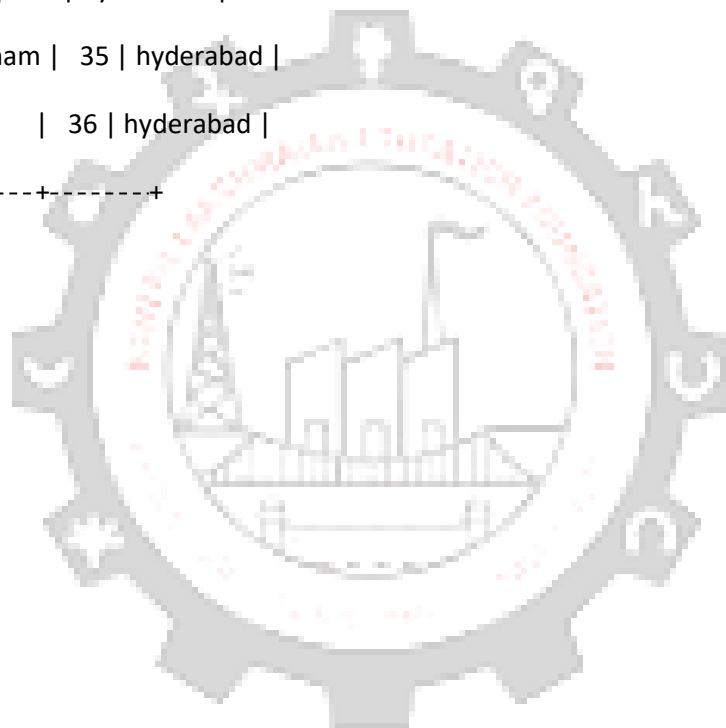
```
mysql> insert into doctor values(4,'atmaram','begumpet',38,'hyderabad');
```

Query OK, 1 row affected (0.07 sec)

mysql> select \* from doctor;

```
+---+-----+-----+---+-----+
| did | dname  | address    | age | city    |
+---+-----+-----+---+-----+
| 1 | ramesh | himayatnagar | 32 | hyderabad |
| 2 | jogit  | dilsukhnagar | 33 | hyderabad |
| 3 | gupta  | mehandipatnam | 35 | hyderabad |
| 4 | atmaram | begumpet    | 36 | hyderabad |
+---+-----+-----+---+-----+
```

4 rows in set (0.00 sec)



### **1. Explain the difference between two and three-tier architectures.**

A three-tier architecture includes a client and two server layers. The application code is stored on the application server and the database is stored on the database server. A two-tier architecture includes a client and one server layer. The database is stored on the database server.

### **2. Describe and contrast SQL and QBE.**

QBE is a direct-manipulation database language that uses a graphical approach to query construction. Some database systems translate QBE queries into SQL. QBE does not adhere to a standard but SQL does. Both SQL and QBE are relational database languages.

### **3. Describe ODBC**

ODBC is a standard that contains an interface that provides a common language for application programs to access and process SQL databases. In order to use ODBC, a driver, server name, database name, user id, and password are required. ODBC is important for Internet applications and has gained wide acceptance.

### **4. Compare a thin and a fat client.**

A fat client is a PC that is responsible where most processing occurs on the client rather than the server. Some of the fat client activities include processing, presentation logic, and business rules logic. A thin client is a PC where only limited processing occurs. In these cases, more processing should occur on the application server.

### **5. Describe some reasons that someone familiar with Microsoft Access will want to learn VBA.**

You can perform more complex functions and error handling can be accommodated by VBA. VBA code will execute faster since code runs faster than macros and maintenance is easier because VBA modules are stored with the forms and reports. Reading VBA is easier because you can see the entire code. Finally, you can pass parameters and use OLE automation better.

### **6. Describe middleware applications that ease the connection of databases to Web applications.**

Both ASP and ColdFusion are middleware that ease the connection of databases to Web applications. Both require the use of several files. Both use tags and are executed on the server. Both Internet Explorer and Netscape can access these files. The programmer does not need to be concerned with the client when they work with this middleware.

### **7. Describe JavaScript and VBScript.**

JavaScript shares many features with Java. Developers use it to achieve interactivity. JavaScript is an open language and does not require a license. It is supported by both Internet Explorer and Netscape. VBScript is similar to JavaScript. It is based on Visual Basic but is simpler. Microsoft developed this language.

### **8. Describe Web Services.**

Web Services are improving the ability of computers to communicate over the Internet. These services use XML programs and usually run in the background. Easier integration of applications may be possible because developers do not need to be familiar with the technical details with applications that are being integrated. UDDI is a technical specification for creating a distributed registry of Web services and businesses that are open to communicating through Web services.

**9. Provide an overview of XML.**

XML is used to structure and manipulate data involved with a browser and is becoming the standard for e-commerce. XML uses tags that are similar to HTML in that they use the angle brackets, but XML describes the content whereas HTML describes the appearance. The XML schema standard was published in May 2001 by W3C.

**10. Describe Website security issues.**

Website security issues include unauthorized access to the several aspects of one's Website. Security measures should include all aspects of the system such as the network, operating level, database, and Web server. Regular monitoring and security testing by a company should help to avoid intrusion into one's system.

**11. Explain the role of metadata for the three-layer architecture.**

Each of the three layers has a metadata layer linked with it. The metadata layer describes the properties or characteristics of the data. The operational metadata describe the data used in the various operational and external systems. The enterprise data warehouse metadata describe the reconciled data layer. The data mart metadata describes the derived data layer.

**12. Describe why operational and informational systems are separate.**

Operational systems are used to run the business in real time on a daily basis and contain current data. Non-managers and non-analysts work in this type of system. Operational systems must process large volumes of data. Informational systems are used to support decisions and contain historical data. Managers and analysts work in this type of system. Informational systems have periodic batch updates.

**13. Describe a data warehouse.**

A data warehouse is organized around specific subjects. The data is defined using consistent naming, format, and encoding structure standards. The data contains a time element, so that the data can be studied for trends. No data in a data warehouse can be updated by end users.

**14. Explain why an information gap exists in most organizations.**

One reason that an information gap exists is the fact that systems have been developed in separate, segmented efforts. This has helped the data from being stored in an integrated database and thus the data is in an inconsistent structure. The other reason for the gap is that most systems are created to support the operational aspect of an organization. The systems were not developed for decision making.

**15. Describe the differences between a data warehouse and data mart.**

A data warehouse is for the enterprise and contains multiple subjects. A data mart is for a specific functional area and focuses on one subject. A data warehouse is flexible and data-oriented and contains detailed historical data. A data mart is restrictive and project-oriented and contains some historical data.

**16. Describe the difference between data and database administration.**

Data administration is responsible for the overall management of data resources. Some of the core roles include the creation of data policies, procedures and standards, resolve data ownership issues, and manage the information repository. Database administration is physical database oversight. Some of the core duties include the selection of the DBMS and software tools, the installation and upgrade of the DBMS, and database performance tuning.

**17. What are some of the important security features of a DBMS?**

One of the features includes the use of views which allows the presentation of only data needed by someone and limits the capability of database updates. The use of integrity controls includes such things as domains, assertions, and checks. Also authorization rules, user-defined procedures, encryption, authentication schemes, and backups are important.

**18. Describe concurrency control.**

Concurrency control is the process managing simultaneous operations against a database so that database integrity is not compromised. There are two approaches to concurrency control. The pessimistic approach involves locking and the optimistic approach involves versioning.

**19. Explain locking.**

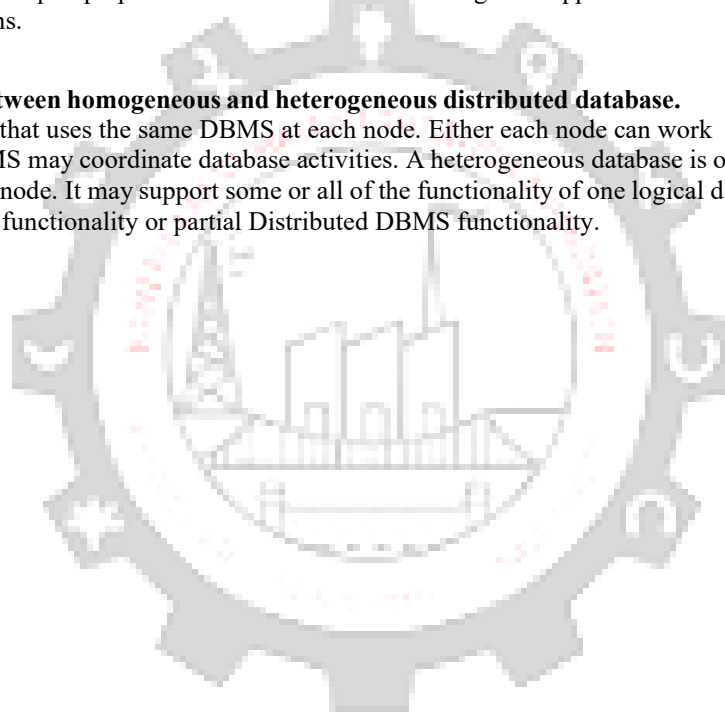
Locking is denying others the ability to update a record until someone completes the update or releases the record. Locking can occur on many different levels in a database. It can occur at the database, table, record, or field level. A lock can be shared (another can read the record while an update is in progress) or exclusive (no one can read the record while an update is in progress).

**20. Explain issues for database performance.**

The installation of the database is an important issue since the better the installation, the better the performance could be. Memory usage is learning how the DBMS uses main memory and then using that knowledge to enable better performance. I/O is usually very intense in a DB, so understanding how users will use the data will help to prepare the database better. CPU usage and application tuning are also important considerations.

**21. Describe the difference between homogeneous and heterogeneous distributed database.**

A homogenous database is one that uses the same DBMS at each node. Either each node can work independently or a central DBMS may coordinate database activities. A heterogeneous database is one that may have a different DBMS at each node. It may support some or all of the functionality of one logical database. It may support full Distributed DBMS functionality or partial Distributed DBMS functionality.



## PROCEDURES

The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks. It is just like procedures in other programming languages.

### Syntax for creating procedures

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS}
BEGIN
    < procedure_body >
END procedure_name;
```

Mysql>Create table doctor(did int(20),dname varchar(20),address varchar(20),age int(20),city varchar(20));

mysql> delimiter /

mysql> create procedure disp\_doctor()

-> begin

-> select \* from doctor;

-> end;

-> /

Query OK, 0 rows affected (0.14 sec)

mysql> call disp\_doctor();

```
+---+-----+-----+---+-----+
| did | dname  | address  | age | city   |
+---+-----+-----+---+-----+
| 1 | ramesh | himayatnagar | 32 | hyderabad |
| 2 | jogit  | dilsukhnagar | 33 | hyderabad |
| 3 | gupta  | mehandipatnam | 35 | hyderabad |
| 4 | atmaram | begumpet   | 36 | hyderabad |
```

```
+---+-----+-----+-----+
```

4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

```
mysql> create procedure doctor_details()
```

```
-> begin
```

```
-> select did,dname from doctor;
```

```
-> end;
```

```
-> /
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> call doctor_details()/
```

```
+---+-----+
```

```
| did | dname |
```

```
+---+-----+
```

```
| 1 | ramesh |
```

```
| 2 | jogit |
```

```
| 3 | gupta |
```

```
| 4 | atmaram |
```

```
+---+-----+
```

4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

```
mysql> create procedure find_doctorid(in id int)
```



```
-> begin
-> select * from doctor where did=id;
-> end;
-> /
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> call find_doctorid(2)/
```

```
+---+-----+-----+-----+
| did | dname | address | age | city |
+---+-----+-----+-----+
| 2 | jogit | dilsukhnagar | 33 | hyderabad |
```

1 row in set (0.10 sec)

Query OK, 0 rows affected (0.11 sec)

```
mysql> call find_doctorid(4)/
```

```
+---+-----+-----+-----+
| did | dname | address | age | city |
+---+-----+-----+-----+
| 4 | atmaram | begumpet | 36 | hyderabad |
```

1 row in set (0.01 sec)

```
mysql> create procedure doctor_info()
```

```
-> begin
-> select did,dname,city from doctor;
-> end;
-> /
```



Query OK, 0 rows affected (0.01 sec)



```
mysql> call doctor_info()/
```

```
+---+-----+-----+  
| did | dname | city |  
+---+-----+-----+  
| 1 | ramesh | hyderabad |  
| 2 | jogit | hyderabad |  
| 3 | gupta | hyderabad |  
| 4 | atmaram | hyderabad |  
+---+-----+-----+  
4 rows in set (0.00 sec)
```

Query OK, 0 rows affected (0.01 sec)



## VIVA QUESTIONS AND ANSWERS

### 1. What is a distributed database?

A distributed database is a single logical database that is spread across more than one node or locations that are all connected via some communication link. It requires multiple DBMSs, running at each remote site. A distributed database can be either homogenous (same DBMS at each node) or heterogeneous (different DBMS at some nodes).

### 2. What is the difference between horizontal and vertical partitioning?

Horizontal partitioning is where some rows of a table are placed into the base relations at one site and other rows are placed at another site. Vertical partitioning is where some columns of a table are placed into the base relations at one site and other columns are placed at another site but each all of these relations must share a common domain.

### 3. Explain concurrency transparency.

Concurrency transparency is where each transaction in a distributed database is treated as if it is the only one in the system. Therefore if several transactions are running at one time, the results will be the same as if each transaction was run in serial order. The transaction manager helps to provide concurrency control. The three methods that may be used are locking, versioning, and timestamping.

### 4. Explain snapshot replication.

Snapshot replication can be used when an application does not require that the data always be current. These applications can be updated by periodic table copying or snapshots. As part of the snapshot effort, all of the data to be included in the snapshot is collected at a primary point. Then a read-only snapshot is taken and the snapshot is sent to each site so that the update can be made.

### 5) Enlist the various relationships of database.

The various relationships of database are:

- One-to-one: Single table having drawn relationship with another table having similar kind of columns.
- One-to-many: Two tables having primary and foreign key relation.
- Many-to-many: Junction table having many tables related to many tables.

### 6) Define Normalization.

Organized data void of inconsistent dependency and redundancy within a database is called normalization.

### 7) Enlist the advantages of normalizing database.

Advantages of normalizing database are:

- No duplicate entries
- Saves storage space
- Boasts the query performances.

### 13) Define Denormalization.

Boosting up database performance, adding of redundant data which in turn helps rid of complex data is called denormalization.

### 8) Define DDL and DML.

Managing properties and attributes of database is called Data Definition Language(DDL).

Manipulating data in a database such as inserting, updating, deleting is defined as Data Manipulation Language. (DML)

### 9. Explain a join between tables

A join allows tables to be linked to other tables when a relationship between the tables exists. The relationships are established by using a common column in the tables and often uses the primary/foreign key relationship.

### 10. Describe and contrast a trigger and a procedure.

Triggers are stored and controlled in the DBMS. A trigger is executed automatically when a condition is met (INSERT, UPDATE, or DELETE). A procedure is also stored in a database. A procedure is not executed automatically.

### 11) Define Union All operator and Union.

Full recordings of two tables is Union All operator. A distinct recording of two tables is Union.

### 12. What is Storage Manager?

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

### 13. What is Buffer Manager?

It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory.

### 14. What is Transaction Manager?

It is a program module, which ensures that database, remains in a consistent state despite system failures and concurrent transaction execution proceeds without conflicting.

### 15. What is File Manager?

It is a program module, which manages the allocation of space on disk storage and data structure used to represent information stored on a disk.

## WEEK 10

### USAGE OF CURSORS

Cursors In MySQL, a cursor allows row-by-row processing of the result sets. A cursor is used for the result set and returned from a query. By using a cursor, you can iterate, or by step through the results of a query and perform certain operations on each row. The cursor allows you to iterate through the result set and then perform the additional processing only on the rows that require it. In a cursor contains the data in a loop. Cursors may be different from SQL commands that operate on all the rows in the returned by a query at one time.

There are some steps we have to follow, given below :

- Declare a cursor
- Open a cursor statement
- Fetch the cursor
- Close the cursor

```
mysql> select * from students;
```

```
+----+-----+-----+-----+
| sid | sname | age | marks |
+----+-----+-----+-----+
| 1 | ravi | 15 | 25 |
| 2 | ramu | 20 | 30 |
| 2 | rahul | 18 | 26 |
| 5 | kiran | 19 | 28 |
| 6 | varun | 21 | 32 |
| 8 | ramesh | 22 | 33 |
| 8 | rohit | 10 | 20 |
+----+-----+-----+-----+
```

7 rows in set (0.00 sec)



```
mysql> delimiter $$
```

```
mysql> create procedure p1(in_customer_id int)
```

```
-> begin
```

```
-> declare v_id int;
```

```
-> declare v_name varchar(20);
```

```
-> declare v_finished integer default 0;
```

```
-> declare c1 cursor for select sid,sname from students where sid=in_customer_id;
```

```
-> declare continue handler for NOT FOUND SET v_finished=1;
```

```
-> open c1;
```

```
-> std:LOOP
```

```
-> fetch c1 into v_id,v_name;
```

```
-> if v_finished=1 then
```

```
-> leave std;
```

```
-> end if;
```

```
-> select concat(v_id,v_name);
```

```
-> end loop std;
```

```
-> close c1;
```

```
-> end;
```

```
-> $$
```

Query OK, 0 rows affected (0.10 sec)

```
mysql> call p1(2);$$
```

```
+-----+
```

```
| concat(v_id,v_name) |
```

```
+-----+
```

```
| 2ramu      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
+-----+
```

```
| concat(v_id,v_name) |
```

```
+-----+
```

```
| 2rahul     |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> call p1(1) $$
```

```
+-----+
```

```
| concat(v_id,v_name) |
```

```
+-----+
```

```
| 1ravi       |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> call p1(5) $$
```

```
+-----+
```



```
| concat(v_id,v_name) |
```

```
+-----+
```

```
| 5kiran      |
```

```
+-----+
```

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```
mysql> call p1(6);$$
```

```
+-----+
```

```
| concat(v_id,v_name) |
```

```
+-----+
```

```
| 6varun      |
```

```
+-----+
```

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)



### 1) How is a process of PL SQL compiled?

Compilation process includes syntax check, bind and p-code generation processes.

Syntax checking checks the PL SQL codes for compilation errors. When all errors are corrected, a storage address is assigned to the variables that hold data. It is called Binding. P-code is a list of instructions for the PL SQL engine. P-code is stored in the database for named blocks and is used the next time it is executed.

### 2) Differentiate between Syntax and runtime errors.

A syntax error can be easily detected by a PL/SQL compiler. For eg, incorrect spelling.

A runtime error is handled with the help of exception-handling section in an PL/SQL block. For eg, SELECT INTO statement, which does not return any rows.

### 3) Explain Commit, Rollback and Savepoint.

For a COMMIT statement, the following is true:

- Other users can see the data changes made by the transaction.
- The locks acquired by the transaction are released.
- The work done by the transaction becomes permanent.

A ROLLBACK statement gets issued when the transaction ends, and the following is true.

- The work done in a transaction is undone as if it was never issued.
- All locks acquired by transaction are released.

It undoes all the work done by the user in a transaction. With SAVEPOINT, only part of transaction can be undone.

### 4) Define Implicit and Explicit Cursors.

A cursor is implicit by default. The user cannot control or process the information in this cursor.

If a query returns multiple rows of data, the program defines an explicit cursor. This allows the application to process each row sequentially as the cursor returns it.

### 5) Explain mutating table error.

It occurs when a trigger tries to update a row that it is currently using. It is fixed by using views or temporary tables, so database selects one and updates the other.

### 6) When is a declare statement required?

DECLARE statement is used by PL SQL anonymous blocks such as with stand alone, non-stored procedures. If it is used, it must come first in a stand alone file.



**7) How many triggers can be applied to a table?**

A maximum of 12 triggers can be applied to one table.

**8) What is the importance of SQLCODE and SQLERRM?**

SQLCODE returns the value of the number of error for the last encountered error whereas SQLERRM returns the message for the last error.

**9) If a cursor is open, how can we find in a PL SQL Block?**

the %ISOPEN cursor status variable can be used.

**10) Show the two PL/SQL cursor exceptions.**

Cursor\_Already\_Open

Invalid\_cursor

**11) What operators deal with NULL?**

NVL converts NULL to another specified value.

var:=NVL(var2,'Hi');

IS NULL and IS NOT NULL can be used to check specifically to see whether the value of a variable is NULL or not.

**12) Does SQL\*Plus also have a PL/SQL Engine?**

No, SQL\*Plus does not have a PL/SQL Engine embedded in it. Thus, all PL/SQL code is sent directly to database engine. It is much more efficient as each statement is not individually stripped off.

**13) What packages are available to PL SQL developers?**

DBMS\_ series of packages, such as, DBMS\_PIPE, DBMS\_DDL, DBMS\_LOCK, DBMS\_ALERT, DBMS\_OUTPUT, DBMS\_JOB, DBMS\_UTILITY, DBMS\_SQL, DBMS\_TRANSACTION, UTL\_FILE.

**14) Explain 3 basic parts of a trigger.**

- A triggering statement or event.
- A restriction
- An action

**15) What are character functions?**

INITCAP, UPPER, SUBSTR, LOWER and LENGTH are all character functions. Group functions give results based on groups of rows, as opposed to individual rows. They are MAX, MIN, AVG, COUNT and SUM.

**16) Explain TTITLE and BTITLE.**

TTITLE and BTITLE commands that control report headers and footers.

**17) Show the cursor attributes of PL/SQL.**

%ISOPEN : Checks if the cursor is open or not

%ROWCOUNT : The number of rows that are updated, deleted or fetched.

%FOUND : Checks if the cursor has fetched any row. It is true if rows are fetched

%NOT FOUND : Checks if the cursor has fetched any row. It is True if rows are not fetched.

**18) What is an Intersect?**

Intersect is the product of two tables and it lists only matching rows.

**19) What are sequences?**

Sequences are used to generate sequence numbers without an overhead of locking. Its drawback is that the sequence number is lost if the transaction is rolled back.

**20) How would you reference column values BEFORE and AFTER you have inserted and deleted triggers?**

Using the keyword "new.column name", the triggers can reference column values by new collection. By using the keyword "old.column name", they can reference column values by old collection.

**21) What are the uses of SYSDATE and USER keywords?**

SYSDATE refers to the current server system date. It is a pseudo column. USER is also a pseudo column but refers to current user logged onto the session. They are used to monitor changes happening in the table.

**22) How does ROWID help in running a query faster?**

ROWID is the logical address of a row, it is not a physical column. It composes of data block number, file number and row number in the data block. Thus, I/O time gets minimized retrieving the row, and results in a faster query.

**23) What are database links used for?**

Database links are created in order to form communication between various databases, or different environments like test, development and production. The database links are read-only to access other information as well.