DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
VISION & MISSION OF THE INSTITUTE

Vision of the Institute

To be a globally recognized institution that fosters innovation, excellence, and leadership in education research, and technology development, empowering students to create sustainable solutions for the advancement of society.

Mission of the Institute

- To foster a transformative learning environment that empowers students to excel in engineering innovation, and leadership.
- To produce skilled, ethical, and socially responsible engineers who contribute to sustainable technological advancements and address global challenges.
- To shape future leaders through cutting-edge research, industry collaboration, and community engagement.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DEPARTMENT VISION & MISSION

Vision:

To empower the students to be technologically adept, innovative, self-motivated and responsible globa citizen possessing human values and contribute significantly towards high quality technical education by harmonizing innovation with sustainability.

Mission:

- To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
- To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
- To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various co-curricular and extra curricular programmers.
- To provide the students with theoretical and applied knowledge, and adopt an education approach tha promotes lifelong learning and ethical growth.

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this manual is a **bonafide record of practical work** carried out in **Databas Management Systems Lab** for the B.Tech Computer Science and Engineering (CSE) III Semeste Programme during the academic year **2026–2027**.

This manual has been prepared by the **Department of Computer Science and Engineering (CSE)** with ou own efforts and to the best of our knowledge.

Signature of Lab Faculty **Signature of HOD**

INDEX

GENERAL INSTRUCTIONS

1. Students are instructed to come to laboratory on time. Late comersare not entertained in the lab.
2. Students should be punctual to the lab. If not, the conducted experiments will not be repeated.
3. Students are expected to come prepared at home with the experiments which are going to be performed.
4. Students are instructed to display their identity cards before entering into the lab.
5. Students are instructed not to bring mobile phones to the lab.
6. Any damage/loss of system parts like keyboard, mouse during the lab session, it is student's responsibility and penalty or fine will be collected from the student.
7. Students should update the records and lab observation books session wise. Before leaving the lab the student should get his lab observation book signed by the faculty.
8. Students should submit the lab records by the next lab to the concerned faculty members in the staffroom for their correction and return.
9. Students should not move around the lab during the lab session.
10. If any emergency arises, the student should take the permission from faculty member concerned in written format.
11. The faculty members may suspend any student from the lab session on disciplinary grounds.
12. Never copy the output from other students. Write down your own outputs.

2230574: DATABASE MANAGEMENT SYSTEMS LAB

B.Tech. II Year I Sem. **L T P C**
0 0 2 1

Prerequisites: A course on Data Structures

Course Objectives:

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

Course Outcomes:. The students should be able to

1. Understand conceptual database design using E–R modeling and relational database structure principles
2. Apply relational model concepts, normalization techniques, and dependency rules for efficient schema design.
3. Execute database definition and manipulation commands for schema creation, data insertion, modification, and retrieval.
4. Develop complex SQL queries using joins, subqueries, set operations, aggregate functions, views, and constraints.
5. Implement database programming constructs including triggers, procedures, and cursors for automated data processing

LIST OF EXPERIMENTS:

1. Concept design with E-R Model

2. Relational Model

3. Normalization

4. Installation of MYSQL and 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

DATABASE MANAGEMENT SYSTEMS LABORATORY

RUBRICS USED TO ASSESS LEARNING IN LABORATORIES

1. RUBRICS FOR DAY-TO-DAY EVALUATION

Parameter	MaxMark	Level-1 (VeryPoor)	Level-2 (Poor)	Level-3 (Average)	Level-4 (Good)	Level-5 (Excellent)
Observation Book	05	No observationsor irrelevant data.(0–1)	Incompleteor incorrectdata. (2)	Basicvalueswith some errors. (3)	Mostlycorrect withgood format.(4)	Fully correct, clear,andwell-formatted.(5)
RecordWriting	05	Not submitted.(0-1)	Submitted butmostly incomplete.(2)	Submitted withsome missing/wrongparts. (3)	Submittedwith minor issues. (4)	Fully complete, correct algorithm& flowchart.(5)
Result	05	Noresultormajor errors.(0–1)	Resultpartially obtained.(2)	Acceptable resultwith limitederror.(3)	Near- correct resultand reasonableerror.(4)	Accurateresult. (5)
Viva-Voce	05	Didnot answerany questions.(1)	Answered veryfewquestions.(2)	Answered some questionswithhelp.(3)	Answeredmost questions correctly.(4)	Answeredall questions accurately.(5)

DATABASE MANAGEMENT SYSTEMS LABORATORY

2. RUBRICS FOR INTERNAL EVALUATION

Criterion	Max Marks	Level-1 *(VeryPoor)*	Level-2 *(Poor)*	Level-3 *(Average)*	Level-4 *(Good)*	Level-5 *(Excellent)*
Design/Tool/Apparat us Selection	2Marks	Incorrect tool/designandno reasoning. **(0)**	Tool/design Selection attemptedwith unclearlogic. **(0.5)**	Satisfactory Selectionwith partial justification. **(1)**	Correctselection andproper analysiswithfew errors.**(1.5)**	Smartselection with accurate, relevantanalysis. **(2)**
Execution (Code/Debug/Run) /Analysis/Method Used	4 Marks	Didnotattempt or complete lyfailed to execute. **(0)**	Attemptedbut unableto proceed orwithmajor errors. **(1)**	Partial execution withsome logic/syntax errors. **(2)**	Mostlycorrect executionwith minimal help.**(3)**	Fully correct and Independentlye xecuted program. **(4)**
Results&Documentation	2 Marks	Incomplete orpoorly presented. **(0)**	Basic structure butlacks clarityor formattin g. **(0.5)**	Completebut genericor with formattin g issues. **(1)**	Well-structure dand mostlyclear. **(1.5)**	Well-organized, professional ,and engaging documentatio n.**(2)**
Viva-Voce (Understandin gofConcepts)	2 Marks	Nounderstandi ng;could notanswerquesti ons. **(0)**	Answeredafe w with difficulty .**(0.5)**	Answeredhalft hequestion swithbasic clarity. **(1)**	Good understandingw ithconfident answers. **(1.5)**	Answered allquestions withclarityand depth. **(2)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SOFTWARE ENGINEERING LABORATORY

3. RUBRICS FOR SEMESTER END EXAMINATIONS

Criterion	Max Marks	Level-1 (VeryPoor (*0–2marks)*	Level-2(Poor)*(3–4 marks)*	Level-3 (Average)*(5–6 marks)*	Level-4 (Good)*(7–9 marks)*	Level-5 (Excellent)*(1 0–12marks)*
Preparednessfor the Experiment	12 marks	Noclarityon objective o procedure. Unableto explain basics.	Limited ideao f the objective/procedu re. Needed prompting.	Has basic understanding; minor gaps in concept or preparation.	Well-prepared, with clear understandingofstepsand background.	Fully preparedwith strong conceptual clarity and confident explanation.
Performanceinthe Laboratory	12 marks	Unable to perform experiment. Relied entirely on examiner'shelp.	Performed with multiple errors and constant support.	Performed with some errors; required occasional help.	Performed mostly independently with minimal support.	Performed independently,efficiently, and with precision.
Calculations & Graphs	12 marks	No or incorrect calculations. Graphs missing or irrelevant.	Multiple calculationerrors. Graphs/plots inaccurate c	Calculations partially correct. Graphspresent butwithsome flaws.	Correct calculations and graphs with minor errors.	Accurate calculations and well-labeled graphswith proper interpretation.
Results& Error Analysis	12 marks	Noresultor invalidresult. No error analysis attempted.	Incorrect result with vague or no error discussion.	Acceptable result. Error analysis attempted but limited.	Correct resultwith sounderror discussion.	Accurate result with detailedand relevant error analysis.
Viva-Voce (Subject Knowledge)	12 marks	Unable to answer any questions. No conceptual understanding.	Answered few questions with poor logic.	Answeredhalf of the questionswith average understanding.	Answered most questions with clarity and confidence.	Answeredall questions with depth, clarity,and reasoning.

2230574: DATABASE MANAGEMENT SYSTEMS LAB

B.TECH II Year I Sem. **L T P C**

0 0 2 1

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

WEEK: 1

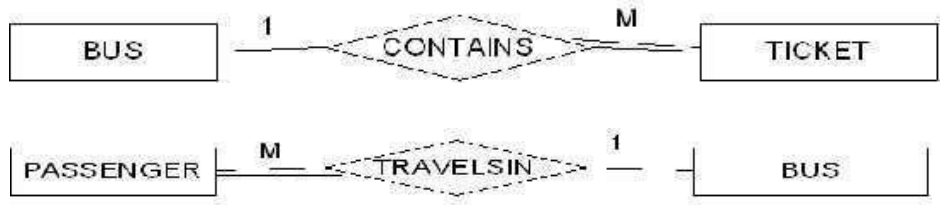
Concept Design with ER Model

Concept design with E – R model

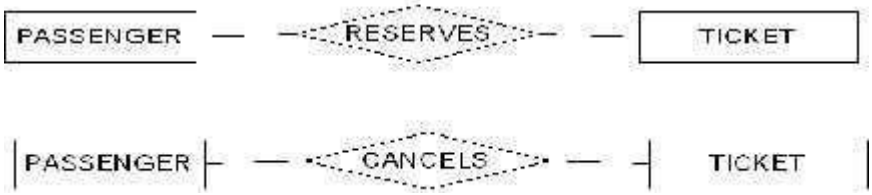
Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationship (total/partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

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.



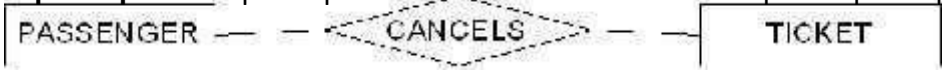
Weak and strong entity: - an entity set may not have sufficient attributes to form a primary key. Such an entity set is termed a weak entity set. An entity set that has primary key is termed a strong entity set.

Total participation:-

Ex: - if a travel agency states that every passenger must make reservation then every passenger travels in bus. Than 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

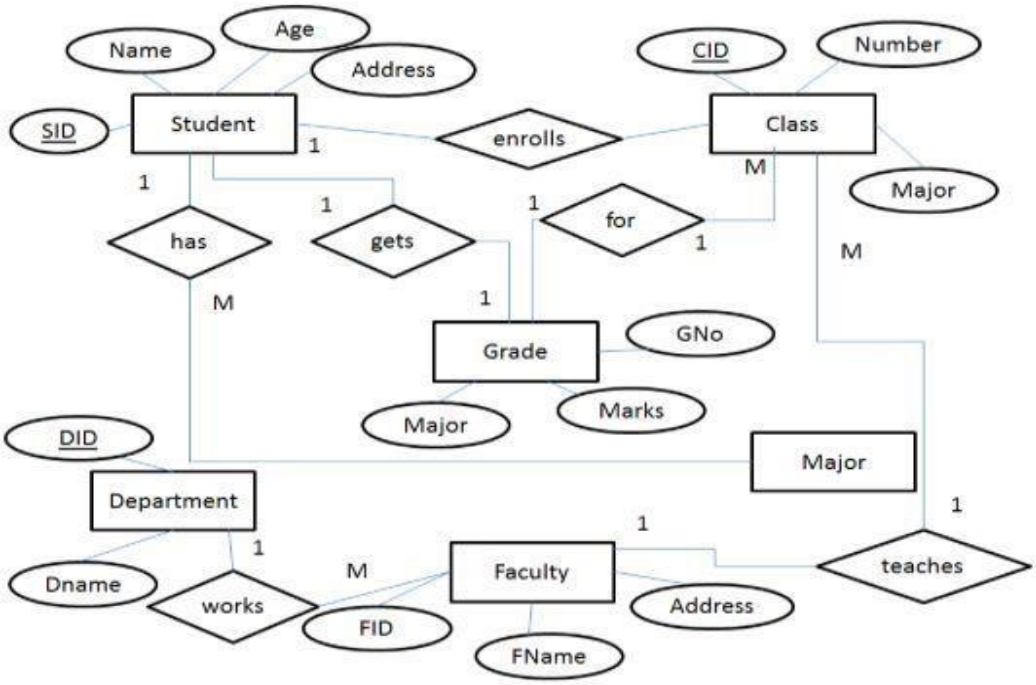
Relationship between Passenger and Bus entities

M:1 binary relationship

Relationship between Passenger and Ticket entities

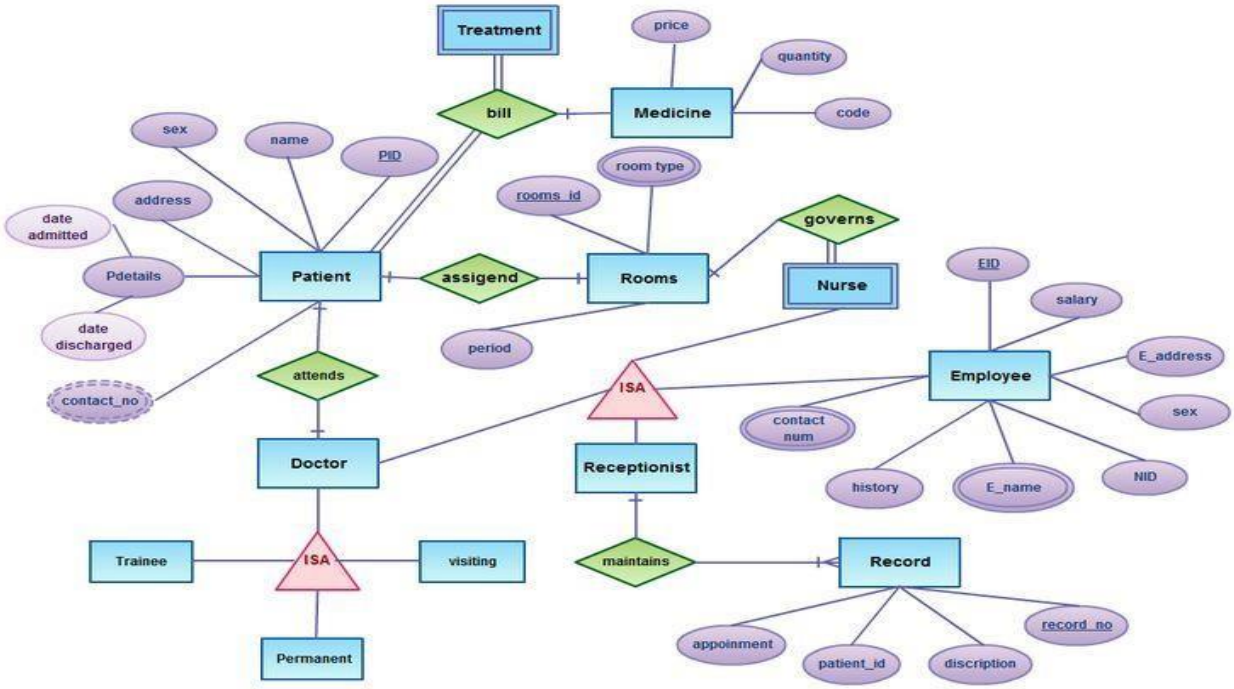
M:N binary relationship

1. **Draw the ER Diagram for UNIVERSITY**

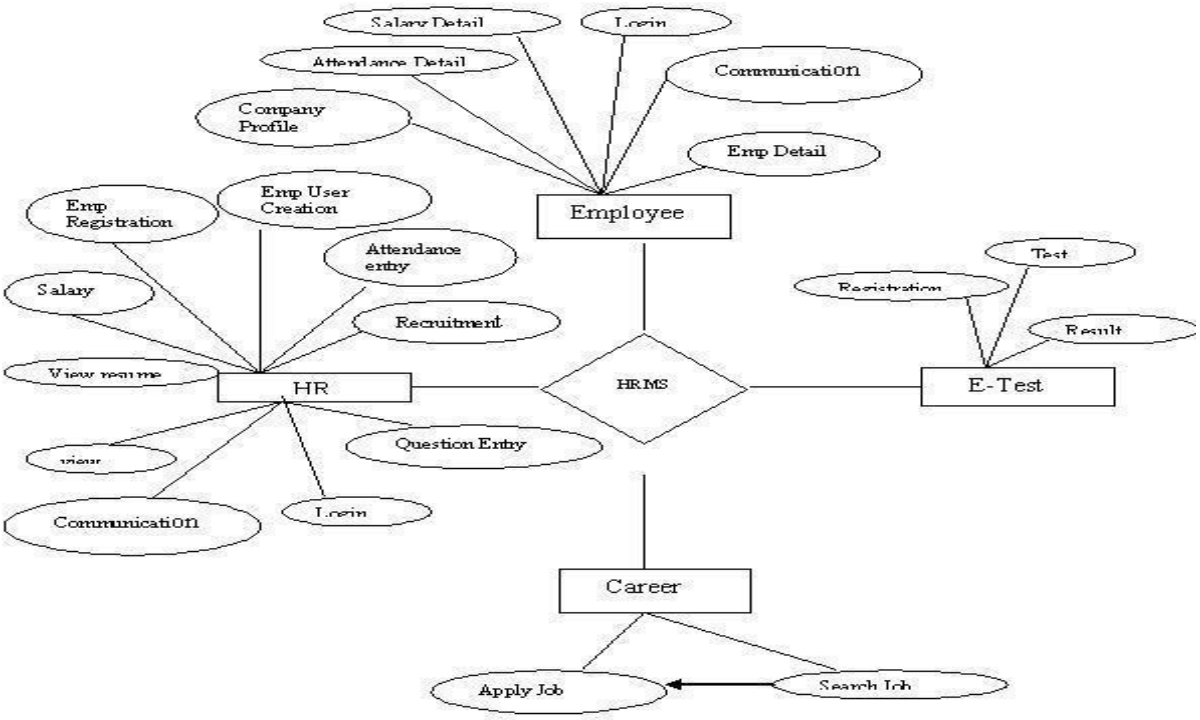


2. **Draw the ER Diagram for Hospital management Systems.**

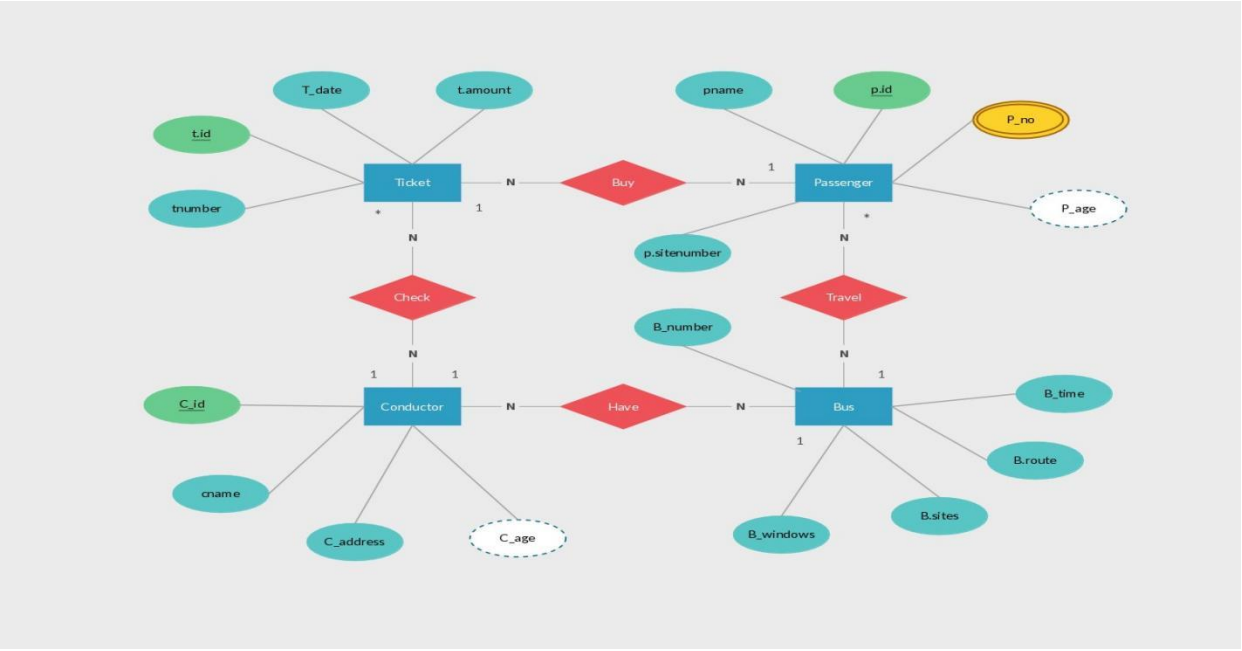
E-R Diagram for Hospital Management System



3. Draw the ER diagram for HR management systems



4. Draw the ER diagram for HR management systems



VIVA QUESTIONS

1. What is database?
2. What is DBMS?
3. What is a Database system?
4. What are the advantages of DBMS?
5. What is normalization?
6. What is Data Model?
7. What is E-R model?
8. What is Object Oriented model?
9. What is an Entity?
10. What is data?
11. What Is Information?
12. Define DBMS.
13. What is Entity?
14. Define attribute
15. What is relationship?
16. What is link?
17. What is strong Entity?
18. What is Weak Entity?
19. Define Entity Set.
20. What is Relationship set?
21. Define Simple attribute.
22. Define Composite Attribute.
23. What is single valued attribute?
24. What is multi valued attribute?
25. Define derived attribute.

26. Write the notation for strong entity.

27. Write the notation for weak Entity.

28. What is the difference between strong entity and Weak entity?

29. What is the difference between simple and Composite attribute?

30. What is the difference between multi valued and derived attribute?

WEEK: 2

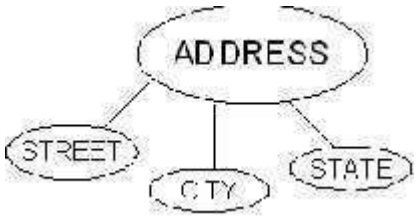
Relational Model

Relational model

Represent all entities (strong, week) 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:

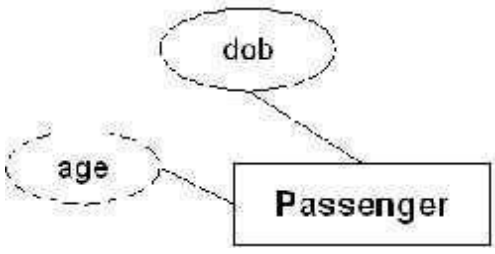
Composite attributes: can be divided into smaller sub parts which represent more basic attributes with independent meaning.



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 Ai must have a domain, dom(Ai)

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 Ai must have a domain, dom(Ai)

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

1. What is relational model?
2. What is degree of a relation?
3. What is cardinality of a relation?
4. What is tuple?
5. What is Constraint?
6. Define Attribute?
7. What is primary key?
8. What is Unique Key constraint?
9. What is not Null Constraint?
10. Define Check constraint.
11. What is integrity constraint over relation?
12. What is Domain?
13. What is domain integrity constraint?
14. What is Entity integrity constraint over Relation?
15. What is Foreign key constraint?
16. How to specify primary key in schema of database?
17. How to specify Not Null in Schema of database?

18. What is Key constraint over relation?
19. What is Simple Attribute in a relation?
20. What is Default constraint in relation?

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 an 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 → 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

1. What is normalization?
2. What is Functional Dependency?
3. What is Lossless join property?
4. What is 1 NF?
5. What is Fully Functional dependency?
6. What is 2NF?
7. What is 3NF?
8. What is BCNF (Boyce-Codd Normal Form)?
9. Define Entity Set.
10. What do you mean by Entity type extension?
11. What is 4NF?
12. What is 5NF?
13. What is Domain-Key Normal Form?
15. What is Partial Key?
16. What is Alternate Key?
17. What is Artificial Key?
18. What is Compound Key?
19. What is Trivial Functional Dependency ?
20. What is Non trivial Functional Dependency?
21. What is multi valued Dependency?
22. What is PJNF?
23. What is partial dependency?
24. What is Transitivity Property?
25. What is full form of PJNF?
26. What is full form of BCNF?
27. Why Normalization is used in DBMS?
28. What is Redundancy?
29. What is Decomposition?
30. What is lossless decomposition?

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;

VIVA QUESTIONS

1) Write The Syntax to create Table in Database?
2) Give an example to create the table in database
3) Why alter Command is used in SQL?
4) Write the syntax to add new column in table using alter Command.
5) Write the syntax to modify column using alter command.
6) Explain about truncate command.
7) Give an examples to modify the column using alter command.
8) Write the syntax for DROP command in sql
9) List out the types of DDL commands.
10) Explain about alter Command
11) What is SQL?
12) Why create command is used as DDL command?
13) How to add new column to a table after any existing column?
14) How to drop column using alter command?
15) How to add multiple columns into a table using alter command?
16) How to drop single column from existing table?
17) How to drop multiple columns from the table using alter command?
18) How to change the specified length for any column datatype using alter command?
19) What is DDL?
20) How to rename column in the database?
21) How to rename table using alter command?
22) Write the syntax to rename column in database.
23) Write the syntax to rename table in database.
24) Write the syntax ti modify datatype of column in relation.
25) What is Drop Command?

26) How Drop command differs from Delete Command?

27) What is truncate command?

28) How truncate command Differs from delete command?

29) Write the syntax for truncate command?

30) Write the syntax to create a database.

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

1. What is DDL (Data Definition Language)?
2. What is VDL (View Definition Language)?
3. What is SDL (Storage Definition Language)?
4. What is Data Storage – Definition Language?
5. What is DML (Data Manipulation Language)?
6. What is DML Compiler?
7. What is Query evaluation engine?
8. What is DDL Interpreter?
9. List out the commands used in DML.
10. Why insert command is used in database?
11. What is the syntax to insert a new record in the database?
12. What is Delete command?
13. What is the syntax to delete the records in database based on condition?
14. What is update command?
15. Write the syntax to update a single column data in database.
16. What is the syntax to update multiple columns of a single record?
17. What is on delete cascade?
18. How to specify on delete cascade in relational databases?
19. What is On delete set null?
20. How to specify on delete set null in relational databases?
21. What is On Update cascade?
22. How to specify on update Cascade in relational databases?
23. What is the full form of DCL?
24. What is full form of TCL?
25. What commands are used in DCL ?
26. List out the commands used in TCL?
27. What is the use of Grant Command?
28. What is the use of revoke command?
29. What is the use of commit command?
30. What is the use of rollback command?

WEEK: 6

Querying (using ANY, IN,NOTIN 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 **IN** operator	SELECT column_name1, column_name2, etc FROM table_name WHERE column_name1 IN (value1, value2, etc);
Syntax for SQL **NOT IN** 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

1. What is a query?
2. What do you mean by Correlated subquery ?
3. What are the primitive operations common to all record management systems?
4. Write the syntax for IN operator.
5. Why IN operator is used in database?
6. Why all operator is used in sql?
7. What is the syntax for all operator?
8. Why Union operator is used?
9. Why union all operator is used?
10. What is the use of Not in Opearator?
11. Write the syntax for Not in operator.
12. Write an example for IN operator.
13. Write an example for NOT IN operator.
14. Write an example for ANY operator.
15. Write an example for ALL operator.
16. What are the primitive operations common to all record management systems?
    Addition, deletion and modification.
17. Name the buffer in which all the commands that are typed in are stored.
18. What are the unary operations in Relational Algebra?
19. Are the resulting relations of PRODUCT and JOIN operation the same?
20. What is RDBMS KERNEL?
21. Name the sub-systems of a RDBMS
22. Which part of the RDBMS takes care of the data dictionary?
23. What is the job of the information stored in data-dictionary?
24. Not only RDBMS takes care of locating data it also
25. How do you communicate with an RDBMS?
26. State the differences between SQL and other conventional programming Languages
27. Name the three major set of files on disk that compose a database in Oracle.
28. What is the difference between any and all operator?
29. What is the difference between union and union all operator?
30. What is the difference between In and Not in operator?

WEEK: 7

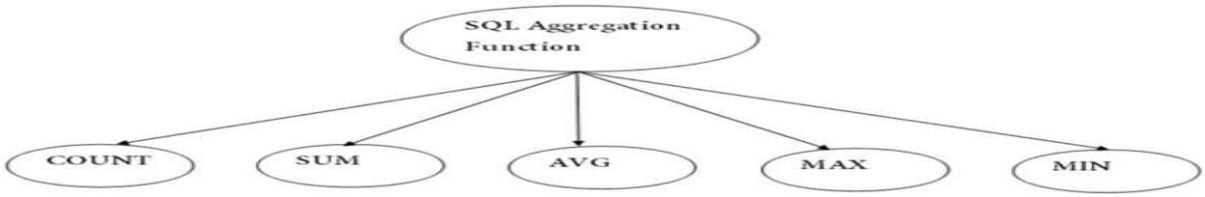
Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.

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;
```

VIVA QUESTIONS AND ANSWERS

1.List out the aggregate functions supported by sql?
2.Write the syntax for count function.
3. Write the syntax for Max function.
4. Write the syntax for Min function.
5. Write the syntax for Average function.
6. Write the syntax for Sum function.
7. Why Views are used in SQL?
8. Write the syntax to create view in sql.
9. Why group by clause is used in SQL?
10. Why having clause is used in SQL?
11. Write the syntax to apply having clause in sql.
12. What is Storage Manager?
13. What is Buffer Manager?
14. What is Transaction Manager?
15. What is File Manager?
16. What is Authorization and Integrity manager?
17. What are the four objectives of the selection of a data type?
18. Describe the four types of indexes.
19. What is denormalization and why would someone consider doing so?
20. Compare a hierarchical and network database model?
21. Describe the differences between vertical and horizontal portioning.
22. Explain the difference between a dynamic and materialized view.
23. Discuss some of the techniques that can be used to tune operational performance.
24. Briefly describe the three types of SQL commands.
25. What are the steps to follow when preparing to create a table?
26. What are some disadvantages of a standard language such as SQL?
27. Explain a join between tables
28. Describe and contrast a trigger and a procedure.
29. Briefly describe an outer join.
30. Describe a 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)
```

VIVA QUESTIONS

1).What is a Triggers in SQL Server?
2). What are the type of Triggers?
3). Explain what is after Trigger?
4). What is the maximum depth level for nested triggers?
5). What are magic tables in SQL Server?
6.How many types of Magic tables available in SQL server?
7. Suppose you have a view that is formed by joining two tables together, what database object would allow you to insert a new row into both of the two tables joined together? .
8). If a trigger aborts due to a run-time error and no exception handler exists, what will occur?
9). Which T-SQL statement can you include in the code of the trigger to indicate a normal exit from the trigger?
10). If there is an AFTER INSERT trigger on a table, how many times will that trigger fire if you insert 50 rows using a single INSERT?
11). If a DML trigger fires and executes another DML statement also contains a trigger, what will happen to that second trigger, it will fire or not?
12). What command explicitly fires a trigger, means can we fire trigger forcefully?
13). How many triggers are possible per table?
14). You are working with some Transact-SQL code that causes an error. You want to determine the name of the trigger or stored procedure that caused a specific error. You configure the appropriate TRY..... CATCH construct, using the code so that you can further diagnose the error. What will you do for that?
15.You have created a DML trigger that fires when an UPDATE operation is performed. You want to ensure that this DML trigger does not fire in the event that no rows are affected by the UPDATE operation. What you will do to implement same?
16).Is it possible to create trigger on views?
17). Suppose you need to save/get the count of rows inserted into a table, How you will achieve that?
18.How can you use @@ROWCOUNT to improve the performance of a trigger?
19). You notice a system uses a lot of triggers to enforce foreign key constraints, and the triggers are error-prone and difficult to debug. What changes can you recommend to reduce the use of triggers?
20. Suppose there is an important table that requires some simple logging actions to take place after any changes to the data. The logging is to a custom table built especially to meet application requirements. What recommendation might you make to help implement such a logging action?

21. What are the difference between Trigger and Stored Procedure?
22. What is the difference between trigger and function in SQL Server?
23. What are the two types of Triggers in SQL Server?
24. What are the special tables used by Triggers in SQL Server?
25. What are some use cases of Triggers??
26. Difference between Triggers and Constraints?
27. What is a DML Trigger?
28. What happens when a trigger is created with compilation errors and a triggering event occurs?
29. What happens when a DML is executed on a view and base table has trigger?
30. Types of DML Triggers in Oracle?

WEEK-9
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

1. Differentiate PL/SQL and SQL?
2. Enlist the characteristics of PL/SQL?
3. What are the data types available in PL/SQL?
4. Explain the purpose of %TYPE and %ROWTYPE data types with the example?
5. What do you understand by PL/SQL packages?
6. What do you understand by PL/SQL cursors?
7. Explain cursor types.
8. When do we use triggers?
9. Explain the difference in the execution of triggers and stored procedures?
10. Explain the difference between Triggers and Constraints?
11. What is a PL/SQL block?
12. Differentiate between syntax and runtime errors?
13. What are COMMIT, ROLLBACK, and SAVEPOINT?
14. What is the mutating table and constraining table?
15. What are actual parameters and formal parameters?
16. What is the difference between ROLLBACK and ROLLBACK TO statements?
17. Write a PL/SQL script to display the following series of numbers: 99,96,93……9,6,3?
18. What are the 3 modes of parameter?
19. Why is %ISOPEN always false for an implicit cursor?
20. When a DML statement is executed, in which cursor attributes, the outcome of the statement is saved?
21. What are the ways of commenting in a PL/SQL code?
22. What do you understand by Exception handling in PL/SQL?
23. Enlist some predefined exceptions?
24. What is PL/SQL cursor exceptions?

25. Explain the difference between cursor declared in procedures and cursors declared in the package specification?

26. What are INSTEAD OF triggers?

27. What are expressions?

28. List different type of expressions with the example.

29. Write a program that shows the usage of the WHILE loop to calculate the average of user entered numbers and entry of more numbers are stopped by entering number 0?

30. What do you understand by PL/SQL Records?

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              |
+----------------------+
```

```
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)
```

VIVA QUESTIONS

1) What is PL SQL ?

2) Differentiate between % ROWTYPE and TYPE RECORD.

3) Explain uses of cursor.

4) Show code of a cursor for loop.

5) Explain the uses of database trigger.

6) What are the two types of exceptions.

7) Show some predefined exceptions.

8) Explain Raise_application_error.

9) Show how functions and procedures are called in a PL SQL block.

10) Explain two virtual tables available at the time of database trigger execution.

11) What are the rules to be applied to NULLs whilst doing comparisons?

12) How is a process of PL SQL compiled?

13) Differentiate between Syntax and runtime errors.

14) Explain Commit, Rollback and Savepoint.

15) Define Implicit and Explicit Cursors.

16) Explain mutating table error.

17) When is a declare statement required?

18) How many triggers can be applied to a table?

19) What is the importance of SQLCODE and SQLERRM?

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

21) Show the two PL/SQL cursor exceptions.

22) What operators deal with NULL?

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

24) What packages are available to PL SQL developers?

25) Explain 3 basic parts of a trigger.

26) What are character functions?

27) Explain TTITLE and BTITLE.

28) Show the cursor attributes of PL/SQL.

29) What is an Intersect?

30) What are sequences?