Diploma Engineering

Laboratory Manual

Database Management
4331603

[Information Technology / Semester-3]

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate of Technical Education
Gandhinagar - Gujarat

DTE’s Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls’ education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

DTE’s Mission:

Institute’s Vision:

Institute’s Mission:

Department’s Vision:

Department’s Mission:

Certificate

This is to certify that Mr./Ms ………………………………………………………………..

Enrollment No. ………….……….….……………. of **3rd** Semester of *Diploma in* ***Information Technology*** of Institute ……………………………………………………………….. (GTU Code: ………)

has satisfactorily completed the term work in course "**Database Management (4331603)"** for the academic year: …………………….…. Term: **Odd** prescribed in the GTU curriculum.

Place: ………………………..

Date: ………………………..

Signature of Course Faculty **Head of the Department**

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum - 2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities for each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

This course aims to develop skills in students to create, store, modify, manage and extract information from a database. Database system can be used as a backend for developing database applications.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs):

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the *engineering* problems.
2. **Problem analysis**: Identify and analyse well-defined *engineering* problems using codified standard methods.
3. **Design/ development of solutions:** Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
4. **Engineering Tools, Experimentation and Testing:** Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.
6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
7. **Life-long learning:** Ability to analyse individual needs and engage in updating in the context of technological changes in field of engineering.

Practical Outcome - Course Outcome matrix

Course Outcomes (COs):

a. Explain the basic concepts of databases.
b. Design database using Entity relationship approach.
c. Implement Relational algebra in a database.
d. Apply SQL Commands for creating, manipulating and controlling databases.
e. Apply concepts of normalization to design an optimal database.
f. Explain transaction management concepts for concurrent use of database.

Sr. No.	Practical Outcome/Title of experiment	CO1	CO2	CO3	CO4	CO5	CO6
1.	Basics of Database, its applications and Data Models a. Prepare a report on Current Database Trends. b. Prepare a list of most famous websites and Mobile Applications with Database Tools used in it.	√	-	-	-	-	-
2.	Entity Relationship Diagrams a. Draw ER-Diagram for Banking Management System b. Convert ER-Diagram to relational schema.	-	√	√	-	-	-
3.	SQL Commands a. Design the below given schemas using SQL Command – "Create". Decide the appropriate data type for each column. b. Insert data in above tables using SQL Command "Insert". c. Retrieve data from tables in Practical 2 using Data SQL command- "Select". a. Write SQL queries to use Update, alter, rename, delete, truncate and distinct	-	-	-	√	-	-
4.	In-built Functions (Part – 1) a. Write SQL queries to use various date functions. a. Write SQL queries to use various numeric functions	-	-	-	√	-	-

5.	In-built Function (Part 2) a. Write SQL queries to use various character functions. (1)Find out length of string “hello world”. (2)Change the case of ‘HELLO WORLD’ string to lower case. (3)Change the case of ‘hello world’ to upper case. (4)Display each word initial letter as capital letter : government polytechnic for girls. (5)Find ‘put’ from string ‘computer’. (6)Add 10 star on left side of India string. (7)Add 10 star on right side of India string. (8)Trim ‘ion’ from ‘information’. (9)Trim ‘info’ from ‘information’. b. Write SQL queries to use various conversion functions (a)Convert 100000 in format 99,99,99 a. (b)Convert today’s date to MM-DD-YY format	-	-	-	√	-	-
6.	Group Functions and Operators a. Write SQL queries to use various group function and operators using tables created.	-	-	-	√	-	-
7.	Set and Join Operators. a. Write SQL query for set operators and join operations.	-	-	-	√	-	-
8.	SQL Constraint a. Apply the concept of integrity/data constraints while creating/altering a table.	-	-	-	√	-	-
9.	Privilege Commands a. Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER command.	-	-	-	√	-	-

10.	PL/SQL Concepts a. Write a program in PL/SQL to show the user of Cursor b. Write a program in PL/SQL to show the user of Stored Procedures. c. Write a program in PL/SQL to show the user of Stored Function. a. d. Write a program in PL/SQL to show the user of Database Triggers.	-	-	-	√	-	-
11.	Refining database design through normalization a. Normalization of database. Consider the following relational schema: a)Employee(eid, ename, salary, mngr id, mngrname, mngrage,ecity) b) Student(no,name, bdate, city, semester, subname, submarks, percentage) For above schemas, answer the following questions:- a) Determine the primary key for given relation. b) Is there any pitfall in design of relation schema? Explain it. c) Find out various dependencies among given attributes of given relation schema. a. d) Normalize given relation schema up to possible normal form Normalization of database for a given relational schema.	-	-	-	-	√	-
12.	Transaction Management a. Prepare a report on transaction management concepts for concurrent access of database by multiple users.	-	-	-	-	-	√

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

- Data retrieval and analysis from relational databases.
- Query optimization and performance tuning for efficient data access.
- Basic database administration tasks, such as creating tables and managing user permissions.
- Simple data manipulation and transformation using SQL statements.
- Familiarity with database systems commonly used in various industries.

Guidelines to Course Faculty

1. Course faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Course faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should involve & give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all observations.
2. Students shall develop maintenance skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

Continuous Assessment Sheet

Enrollment No:

Name: **Term:**

Sr. No.	Practical Outcome/Title of experiment	Page	Date	Marks (10)	Sign
1.	Basics of Database, its applications and Data Models a. Prepare a report on Current Database Trends. b. Prepare a list of most famous websites and Mobile Applications with Database Tools used in it.				
2.	Entity Relationship Diagrams a. Draw ER-Diagram for Banking Management System b. Convert ER-Diagram to relational schema.				
3.	SQL Commands d. Design the below given schemas using SQL Command - ”Create”. Decide the appropriate data type for each column. e. Insert data in above tables using SQL Command ”Insert”. f. Retrieve data from tables in Practical 2 using Data SQL command- “Select”. g. Write SQL queries to use Update, alter, rename, delete, truncate and distinct				
4.	In-built Functions (Part – 1) b. Write SQL queries to use various date functions. c. Write SQL queries to use various numeric functions				

5.	In-built Function (Part 2) a. Write SQL queries to use various character functions. (1)Find out length of string “hello world”. (2)Change the case of ‘HELLO WORLD’ string to lower case. (3)Change the case of ‘hello world’ to upper case. (4)Display each word initial letter as capital letter : government polytechnic for girls. (5)Find ‘put’ from string ‘computer’. (6)Add 10 star on left side of India string. (7)Add 10 star on right side of India string. (8)Trim ‘ion’ from ‘information’. (9)Trim ‘info’ from ‘information’. b. Write SQL queries to use various conversion functions (a)Convert 100000 in format 99,99,99 a. (b)Convert today’s date to MM-DD-YY format				
6.	Group Functions and Operators a. Write SQL queries to use various group function and operators using tables created.				
7.	Set and Join Operators. a. Write SQL query for set operators and join operations.				
8.	SQL Constraint a. Apply the concept of integrity/data constraints while creating/altering a table.				
9.	Privilege Commands a. Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER command.				
10.	PL/SQL Concepts a. Write a program in PL/SQL to show the user of Cursor b. Write a program in PL/SQL to show the user of Stored Procedures. c. Write a program in PL/SQL to show the user of Stored Function. d. Write a program in PL/SQL to show the user of Database Triggers.				

11.	Refining database design through normalization a. Normalization of database. Consider the following relational schema: a)Employee(eid, ename, salary, mngr id, mngrname, mngrage,ecity) b) Student(no,name, bdate, city, semester, subname, submarks, percentage) For above schemas, answer the following questions:- a) Determine the primary key for given relation. b) Is there any pitfall in design of relation schema? Explain it. c) Find out various dependencies among given attributes of given relation schema. d) Normalize given relation schema up to possible normal form Normalization of database for a given relational schema.				
12.	Transaction Management a. Prepare a report on transaction management concepts for concurrent access of database by multiple users.				

Date: ______________

Practical No.1: Basics of Database, its applications and Data Models

a. Prepare a report on Current Database Trends.
b. Prepare a list of most famous websites and Mobile Applications with Database Tools used in it.

A. Objective:

DBMS helps to understand how data is stored, organized, and accessed in computer systems, and its applications in managing large amounts of information efficiently through the use of data models.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

C. Expected Skills to be developed based on competency:

"Design database for a given application"
This practical is expected to develop the following skills.

1. Development of research and technical skills about database.
2. Understanding the technical aspects of database systems.

D. Expected Course Outcomes(Cos)

CO1: Explain the basic concepts of databases.

E. Practical Outcome(PRo)

Students will be able understand recent database trends and tools.

F. Expected Affective domain Outcome(ADos)

1. Follow safety practices.
2. Increased appreciation and enthusiasm for the importance of databases in modern technology and business.
3. Follow ethical practices.

G. Prerequisite Theory:

What is Data?

Data refers to the raw facts and figures that are collected and stored for future use. It can be in any form, such as text, numbers, images, or sound.

What is Information?

Information is the meaningful interpretation of data. It is the output that is obtained after processing the data to extract meaning from it.

What is DBMS?

DBMS stands for Database Management System. It is a software system that allows users to create, maintain, and manage databases.

What is Metadata?

Metadata refers to the data that describes other data. It provides information about the structure, content, and context of data.

What are Data Items?

Data items are the smallest unit of information in a database. They represent a single fact or piece of information.

What are Fields?

Fields are the columns or attributes of a table in a database. They represent the different types of information that can be stored in a table.

What are Records?

Records are the rows or instances of a table in a database. They represent a collection of related data items.

What is a Data Dictionary?

A data dictionary is a centralized repository that stores metadata about the data elements and data structures in a database. It provides information about the data types, relationships, constraints, and other properties of the database.

What is a Schema in DBMS?

A schema is a logical structure that defines how data is organized and stored in a database. It includes the description of tables, attributes, relationships, and constraints.

What is a Sub-schema?

A sub-schema is a subset of the schema that describes the part of the database that is relevant to a particular user or group of users. It provides a view of the database that is tailored to the needs of the user.

What is an Instance in DBMS?

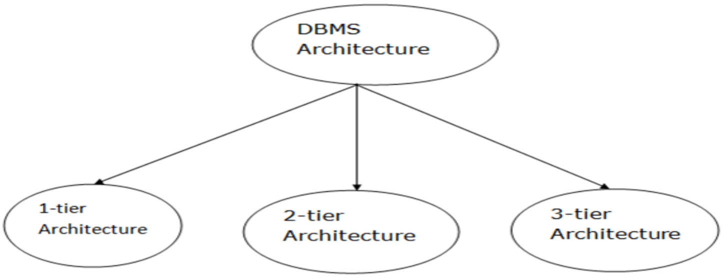
An instance is a snapshot of the data stored in a database at a particular point in time. It includes all the data values for the tables and records in the database. An instance is also referred to as a database state or database snapshot.

DBMS Architecture

The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.

The client/server architecture consists of many PCs and a workstation which are connected via the network. DBMS architecture depends upon how users are connected to the database to get their request done.

- Types of DBMS Architecture



Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: 2-tier architecture and 3-tier architecture.

- 1-Tier Architecture

In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it. Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users. The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

- 2-Tier Architecture

The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. The user interfaces and application programs are run on the client-side. The server side is responsible to provide the functionalities like: query processing and transaction management. To communicate with the DBMS, client-side application establishes a connection with the server side.

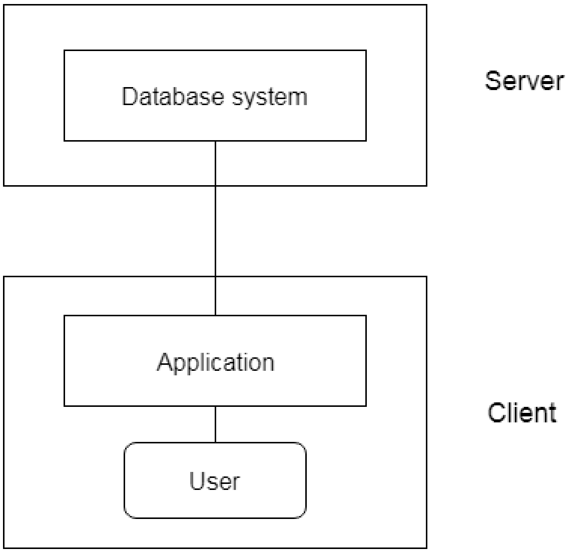


Fig: 2-tier Architecture

- 3-Tier Architecture

The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server. The application on the client-end interacts with an application server which further communicates with the database system. End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application. The 3-Tier architecture is used in case of large web application.

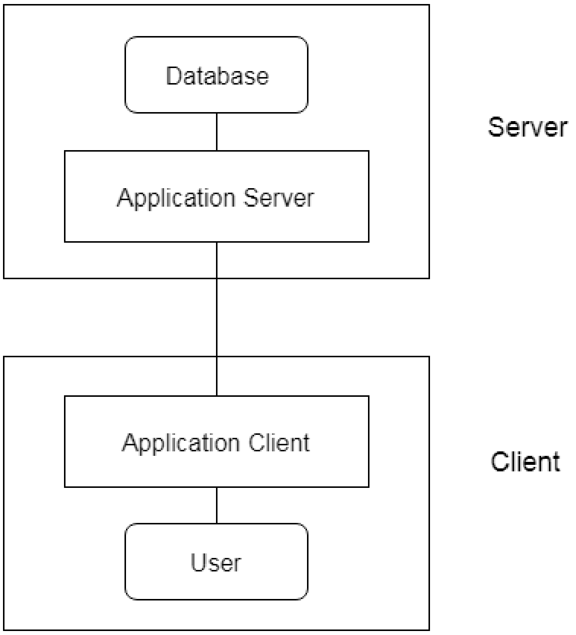


Fig: 3-tier Architecture

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Report:

Prepare a report on Current Database Trends.

Prepare a list of most famous websites and Mobile Applications with Database Tools used in it.

K. **Practical related Quiz.**

1) DBMS stands for ________________________________.

2) Metadata is ____.

3) A schema is a ________________________ structure.

4) In ____________ tier, the database is directly available to the user.

L. References / Suggestions

1. https://www.w3schools.com/sql/default.asp
2. https://www.tutorialspoint.com/dbms/
3. https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.2: Entity Relationship Diagram**s**

a. Draw ER-Diagram for Banking Management System
b. Convert ER-Diagram to relational schema.

A. Objective:

The objective of converting an ER-Diagram to a relational schema is to create a database structure that can be implemented in a database management system..

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Draw ER-Diagram for Banking Management System and Convert ER-Diagram to relational schema”

This practical is expected to develop the following skills.

1. Data modelling skills to represent the entities, attributes, and relationships in a clear and organized way.
2. Ability to think logically and analyse complex systems to identify the relationships and dependencies between different entities and attributes.

D. Expected Course Outcomes(Cos)

CO2: Design database using Entity relationship approach.

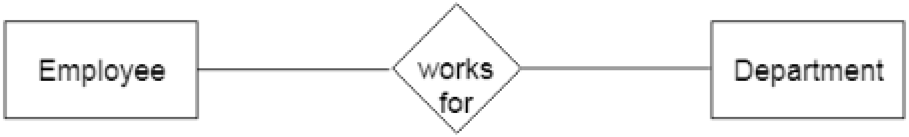
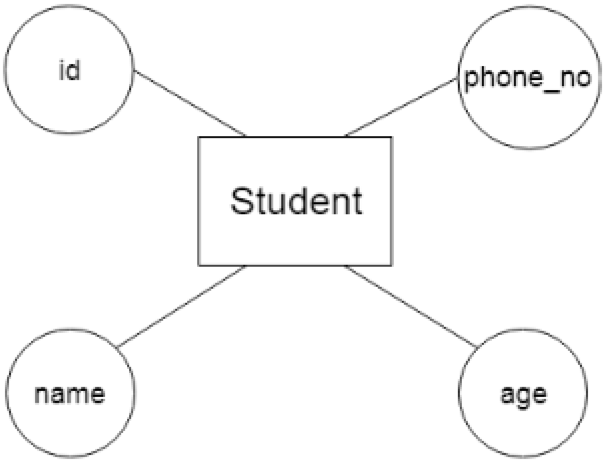
E. Practical Outcome(PRo)

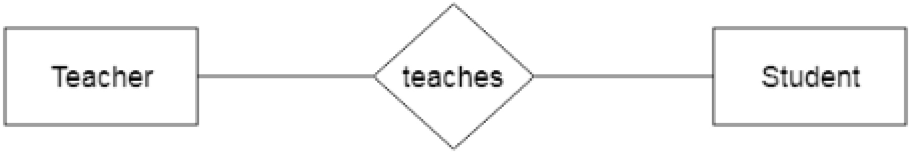
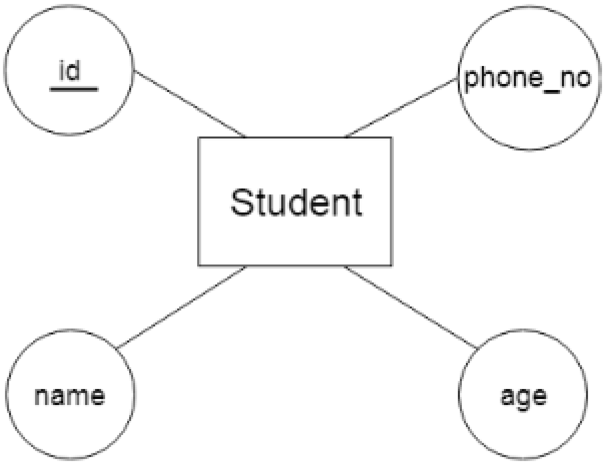
Students will be able to create E-R Diagram for any system and its conversion to relational schema.

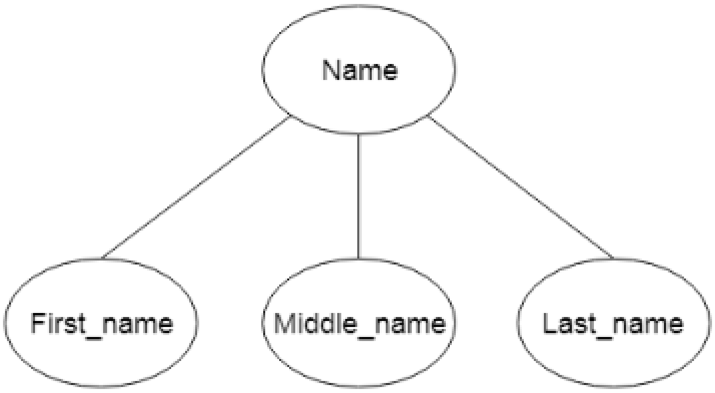
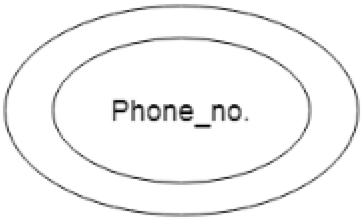
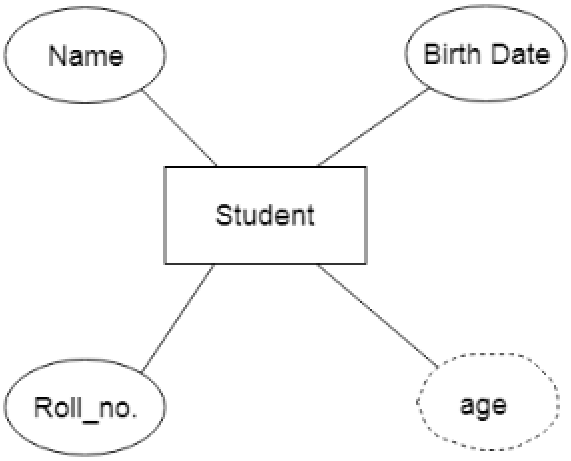
F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Entity	A real-world object or concept that has a unique identity and can be distinguished from other objects or concepts in the system. Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity. 
Attributes	Characteristics or properties of an entity that describe its features, such as name, address, age, or gender. For example, id, age, contact number, name, etc. can be attributes of a student. 

<table>
<tr><td><b>Relationship</b></td><td>An association between two or more entities that represents a meaningful connection or interaction, such as "works for", "is a part of", or "belongs to".
Diamond or rhombus is used to represent the relationship.

 </td></tr>
<tr><td><b>Primary key</b></td><td>A unique identifier for an entity that is used to ensure its uniqueness and enable efficient data retrieval, such as an employee ID or a customer account number. Primary key is represented by and ellipse with the text underlined.

 </td></tr>
<tr><td><b>Foreign key</b></td><td>A field in one table that refers to the primary key of another table, creating a relationship between the two tables and enabling data retrieval and manipulation.</td></tr>
<tr><td><b>Super key</b></td><td>A combination of one or more attributes that uniquely identifies an entity in a table, such as a customer's name and account number.</td></tr>
<tr><td><b>Candidate key</b></td><td>A minimal super key that uniquely identifies an entity, meaning that it cannot be further reduced without losing its uniqueness.</td></tr>
<tr><td><b>Weak entity set</b></td><td>An entity set that cannot be uniquely identified by its own attributes alone, but depends on a related entity set and a partial key, called a discriminator.</td></tr>
<tr><td><b>Composite</b></td><td>An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an</td></tr>
</table>

Attribute	ellipse, and those ellipses are connected with an ellipse. 
Multivalued Attribute	An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute. For example, a student can have more than one phone number. 
Derived Attribute	An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse. For example, A person's age changes over time and can be derived from another attribute like Date of birth. 

Sub class & Superclass

Superclass

A general entity type that has one or more specialized entity types that inherit its attributes and relationships.

Subclass

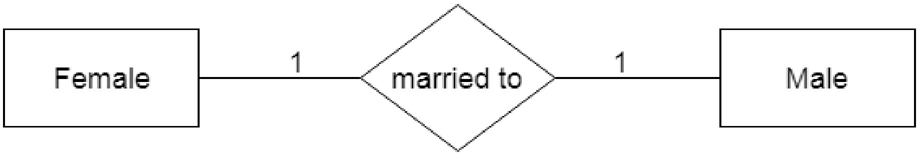
A specialized entity type that inherits attributes and relationships from a superclass and may also have its own unique attributes and relationships.

In object-oriented programming and database design, a subclass is a type of entity that is more specific than a superclass. The subclass inherits the attributes and relationships of the superclass, but it may also have its own additional attributes and relationships. The superclass is a more general entity type that defines common attributes and relationships that are shared by all its subclasses. Subclasses are used to represent more specific types of entities, such as different types of vehicles (car, truck, motorcycle) that share common attributes and relationships (e.g., make, model, year, manufacturer), but also have their own unique features and characteristics.

Types of Relationships

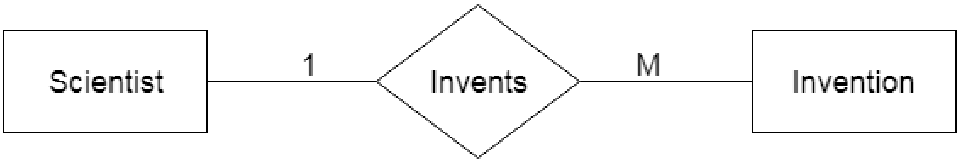
a. One-to-One Relationship

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship. For example, A female can marry to one male, and a male can marry to one female.



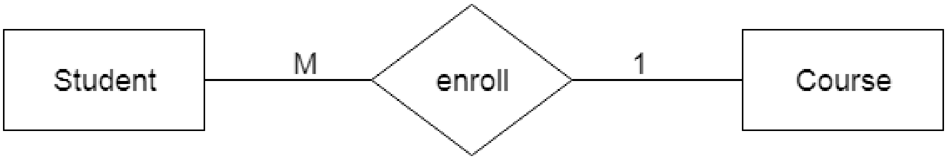
b. One-to-many relationship

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship. For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.



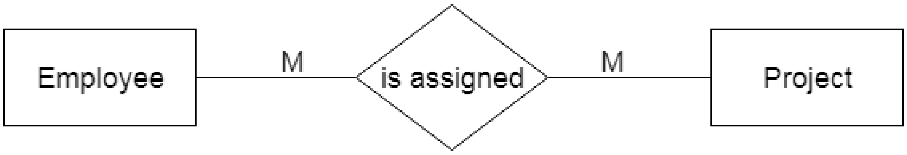
c. Many-to-one relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship. For example, Student enrolls for only one course, but a course can have many students.



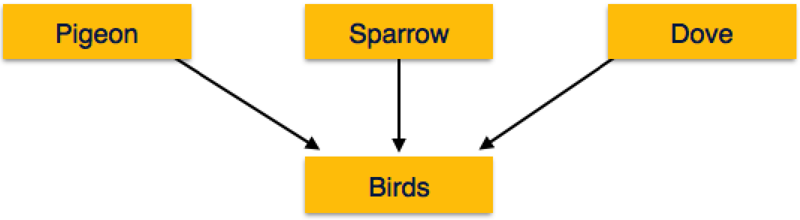
d. Many-to-many relationship

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship. For example, Employee can assign by many projects and project can have many employees.



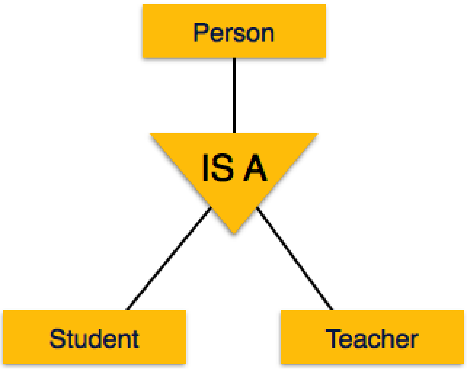
Generalization

As mentioned above, the process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called generalization. In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics. For example, pigeon, house sparrow, crow and dove can all be generalized as Birds.



Specialization

Specialization is the opposite of generalization. In specialization, a group of entities is divided into sub-groups based on their characteristics. Take a group ‘Person’ for example. A person has name, date of birth, gender, etc. These properties are common in all persons, human beings. But in a company, persons can be identified as employee, employer, customer, or vendor, based on what role they play in the company.

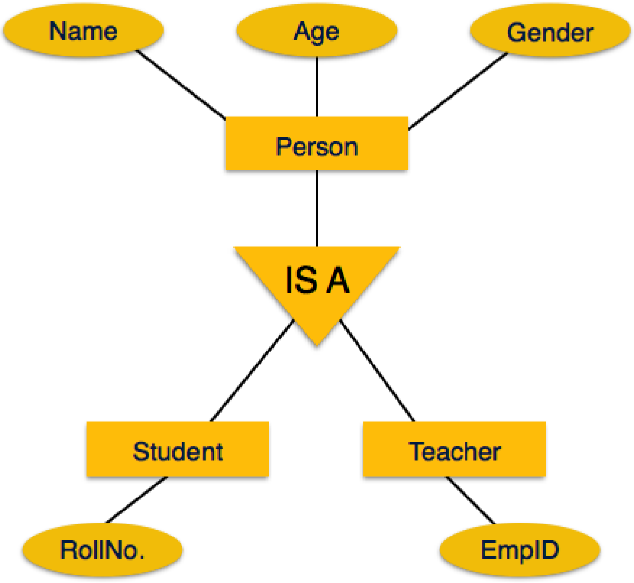


Similarly, in a school database, persons can be specialized as teacher, student, or a staff, based on what role they play in school as entities.

Inheritance

We use all the above features of ER-Model in order to create classes of objects in object-oriented programming. The details of entities are generally hidden from the user; this process known as abstraction.

Inheritance is an important feature of Generalization and Specialization. It allows lower-level entities to inherit the attributes of higher-level entities.



For example, the attributes of a Person class such as name, age, and gender can be inherited by lower-level entities such as Student or Teacher.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) Draw ER – Diagram for Banking Management System.

B) Convert ER-Diagram of Banking Management System to relational schema

K. Practical related Quiz.

1) What does ER stand for?
   a) Entity Relationship b) Enterprise Resource
   c) External Representation d) Exception Reporting

2) What is a cardinality?
   a) Attribute type b) Entity type
   c) Relationship type d) Primary key

3) What is an attribute?
   a) Column name b) Primary key
   c) Foreign key d) Data type

4) What is a candidate key?
   a) Unique identifier b) Alternate key
   c) Primary key d) Foreign key

5) What is a referential integrity?
   a) Database property b) Data validation
   c) Primary key d) Attribute type

L. References / Suggestions

https://www.w3schools.com/sql/default.asp
https://www.tutorialspoint.com/dbms/
https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _______________

Practical No.3: SQL Commands

a. Design the below given schemas using SQL Command - ”Create”. Decide the appropriate data type for each column.
b. Insert data in above tables using SQL Command ”Insert”.
c. Retrieve data from tables in Practical 2 using Data SQL command- “Select”.
d. Write SQL queries to use Update, alter, rename, delete, truncate and distinct.

A. Objective:

SQL commands is to facilitate efficient data management in databases by providing functionalities such as inserting new data, retrieving specific information, updating existing records, deleting unnecessary data, and altering table structures.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“SQL Commands”

This practical is expected to develop the following skills.

1. Developing the skill to retrieve specific information from a database.
2. Gaining proficiency in manipulating data by using SQL commands.
3. Learning to effectively manage and organize data by utilizing SQL commands.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to create and manipulate SQL tables.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

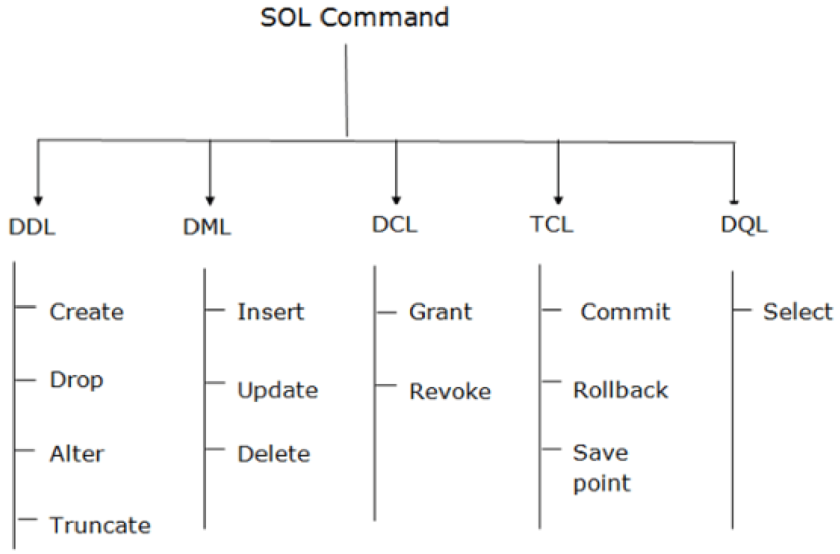
G. Prerequisite Theory:

SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

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



1. 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:

a. CREATE
b. ALTER
c. DROP
d. 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 table_name;

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 table_name MODIFY(column_definitions....);

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;

2. 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:

a. INSERT
b. UPDATE
c. 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 students WHERE Author="Sonoo";

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.
Here are some commands that come under DCL:

a. Grant
b. Revoke

a. Grant : It is used to give user access privileges to a database.

Example : GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

b. Revoke : It is used to take back permissions from the user.

Example : REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

4. Data Query Language

DQL is used to fetch the data from the database. It uses only one command:

SELECT : This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax: SELECT expressions FROM TABLES WHERE conditions;
For example: SELECT emp_name FROM employee WHERE age > 20;

4. **Resources/Equipment Required**

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

H. Safety and necessary Precautions followed

NA

I. Source code:

A) Design the below given schemas using SQL Command - ”Create”. Decide the appropriate data type for each column.

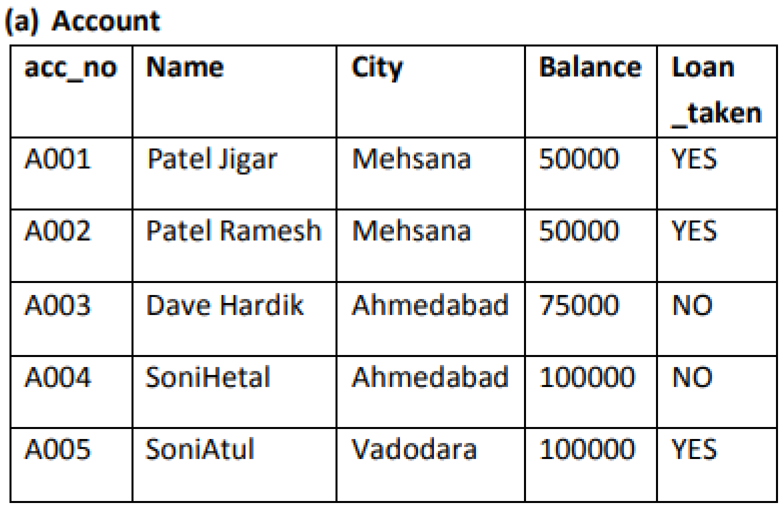
(a) Create a table ACCOUNT with column account number, name, city, balance, loan taken.

(b) Create a LOAN table with column loan number, account number, loan amount, interest rate, loan date, remaining loan. (c)Create table INSTALLMENT with column loan number, installment number, installment date and amount.

(d) Create table TRANSACTION with column account number, transaction date, amount, type of transaction, mode of payment.

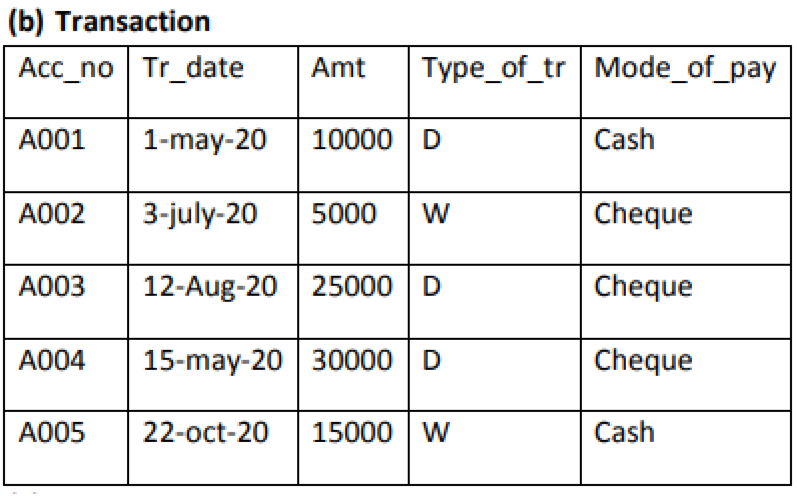
(e) Show the structure of above tables using “Describe” command.

B) Insert data in above tables using SQL Command ”Insert”



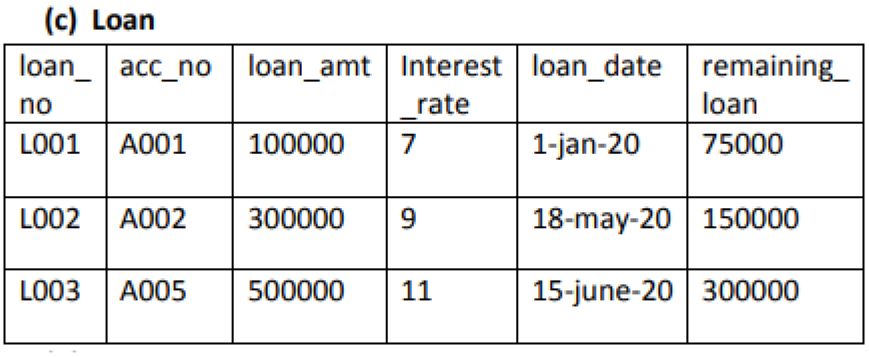
(a) Account

acc_no	Name	City	Balance	Loan _taken
A001	Patel Jigar	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	YES
A003	Dave Hardik	Ahmedabad	75000	NO
A004	SoniHetal	Ahmedabad	100000	NO
A005	SoniAtul	Vadodara	100000	YES



(b) Transaction

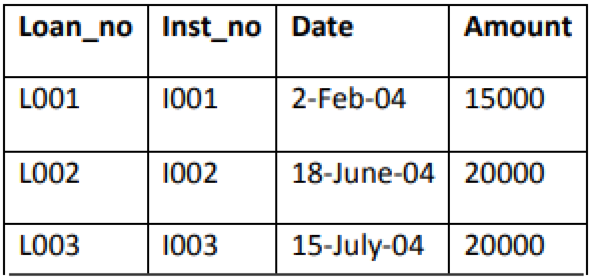
Acc_no	Tr_date	Amt	Type_of_tr	Mode_of_pay
A001	1-may-20	10000	D	Cash
A002	3-july-20	5000	W	Cheque
A003	12-Aug-20	25000	D	Cheque
A004	15-may-20	30000	D	Cheque
A005	22-oct-20	15000	W	Cash



(c) Loan

loan_ no	acc_no	loan_amt	Interest _rate	loan_date	remaining_ loan
L001	A001	100000	7	1-jan-20	75000
L002	A002	300000	9	18-may-20	150000
L003	A005	500000	11	15-june-20	300000

(d) Installment



Loan_no	Inst_no	Date	Amount
L001	I001	2-Feb-04	15000
L002	I002	18-June-04	20000
L003	I003	15-July-04	20000

B) Input-Output:

C) Retrieve data from tables in Practical 2 using Data SQL command- “Select”.

(a)Display all rows and all columns of table Transaction.

(b)Display all rows and selected columns of table Installment.

(c)Display selected rows and selected columns of table Account.

(d)Display selected rows and all columns of table loan.

(e)Display the branch wise balance from account table.

(f)Display list of those branches that have balance greater than 1 Lakh rupees.

(g)Display the list of customers in descending order of their name from account table.

(h)Display those records where mode of payment is “cheque”.

C) Input-Output:

D) Write SQL queries to use Update, alter, rename, delete, truncate and distinct.

Table: ACCOUNT.

(a)Change the name ‘pateljigar’ to ‘patelhiren’.

(b)Change the name and city where account number is A005. (new name = ‘kothari nehal’and new city = ‘patan’).

(c)Display only those records where loan taken status is ‘YES’.

(d)Add the new column (address varchar2 (20)) into table ACCOUNT.

(e)Create another table ACCOUNT_TEMP (acc_no, name, balance) from table ACCOUNT.

(f)Rename the table ACCOUNT to ACCOUNT_MASTER.

(g)Update the column balance for all the account holders. (Multiply the balance by 2 foreach account holders)

(h)Delete the records whose account no is A004

Table: LOAN.

(a)For each loan holders Add 100000 Rs. Amount into the column loan_amt.

(b)For each loan holders Increase the interest rate 2%.

(c)Display only those records where loan holder taken a loan in month of January.

(d)Modify the structure of table LOAN by adding one column credit_no varchar2 (4).

(e)Display the Loan amount*2 of table LOAN.

(f)Display the records of table LOAN by date wise in ascending order.

(g)Display the records of table LOAN by account number wise in descending Order.

(h)Increase the size 5 to 7 of column acc_no.

Table: INSTALLMENT.

(a)Change the Inst_Date ‘2-Feb-04’ to ’3-Mar-04’.
(b)Reduce 5000 amount from all Installment holders.
(c)Add the amount 5000 where loan no is ‘L003’ and ‘L002’.
(d)Change the column size of 5 to 7 where column name is Loan_no.
(e)Delete row where inst_no is ‘I001’.
(f)Only create a structure of table installment1 from table installment.

Table: TRANSACTION.

(a)Insert any duplicate value and display all the records without any duplicate rows.
(b)Select all the records in descending order(account number wise).
(c)Display amt, date, and type of transaction by date wise

D) Input-Output:

J. Practical related Quiz.

1) Which SQL command is used to add new data into a database table?

   a) SELECT b) INSERT

   c) DELETE d) UPDATE

2) Which SQL command is used to modify the structure of a table in a database?

   a) ALTER b) UPDATE

   c) MODIFY d) CHANGE

3) Which SQL command is used to remove all data from a table in a database while keeping the table structure intact?

   a) TRUNCATE b) DELETE

   c) REMOVE d) CLEAN

K. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

L. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student’s engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.4: In-built Functions (Part – 1)

a. Write SQL queries to use various date functions.
b. Write SQL queries to use various numeric functions

A. Objective:

To proficiently use SQL queries with various date functions and numeric functions to effectively manipulate and analyse date-related and numeric data in the database.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Write SQL queries to use various date functions.”

This practical is expected to develop the following skills.

1. Ability to perform date-based calculations, comparisons, and aggregations to analyze and report on time-related data accurately.

“Write SQL queries to use various numeric functions”

This practical is expected to develop the following skills.

1) Skill in using advanced numeric functions such as POWER, SQRT, LOG, and TRUNC to perform complex calculations and generate insights from numerical data.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to understand and use various date and numeric function.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Numeric Function :-

1. ABS() : Returns the absolute value of numeric expression.

2. ACOS(): Returns the arccosine of numeric expression. Returns NULL if the value is not in the range -1 to 1.

3. ASIN() : Returns the arcsine of numeric expression. Returns NULL if value is not in the range -1 to 1

4. ATAN() : Returns the arctangent of numeric expression.

5. ATN2() : Returns the arctangent of the two variables passed to it.

6. CEILING() : Returns the smallest (closest to negative infinity) integer value that is greater than or equal to this value.

7. COS() : Returns the trigonometric cosine of the given value.

8. COT() : Returns the trigonometric cotangent of the given value.

9. DEGREES() : Returns numeric expression converted from radians to degrees.

10. EXP() : Returns the base of the natural logarithm (e) raised to the power of passed numeric expression.

11. FLOOR() : Returns the largest integer value that is not greater than passed numeric expression.

12. LOG() : Returns the natural logarithm of the passed numeric expression.

13. LOG10() : Returns the base-10 logarithm of the passed numeric expression.

14. PI() : Returns the value of pi

15. POWER() : Returns the value of one expression raised to the power of another expression

16. RADIANS() : Returns the value of passed expression converted from degrees to radians.

17. RAND() : Returns the random value between 0 and 1.

18. ROUND() : Returns numeric expression rounded to an integer. Can be used to round an expression to a number of decimal points

19. SIGN() : Returns the sign of a number, indicating whether it is positive, negative, or zero.

20. SIN() : Returns the sine of numeric expression given in radians.

21. SQRT() : Returns the non-negative square root of numeric expression.

22. TAN() : Returns the tangent of numeric expression expressed in radians.

Date Function :

1. ADDDATE(): It returns a date after a certain time/date interval has been added.
2. ADDTIME(): It returns a time / date time after a certain time interval has been added.
3. CURDATE(): It returns the current date.
4. CURRENT_DATE(): It returns the current date.
5. CURRENT_TIME(): It returns the current time.
6. CURRENT_TIMESTAMP(): It returns the current date and time.
7. CURTIME(): It returns the current time.
8. DATE(): It extracts the date value from a date or date time expression.
9. DATEDIFF(): It returns the difference in days between two date values.
10. DATE_ADD(): It returns a date after a certain time/date interval has been added.
11. DATE_FORMAT(): It formats a date as specified by a format mask.

12. DATE_SUB(): It returns a date after a certain time/date interval has been subtracted.
13. DAY(): It returns the day portion of a date value.
14. DAYNAME(): It returns the weekday name for a date.
15. DAYOFMONTH(): It returns the day portion of a date value.
16. DAYWEEK(): It returns the weekday index for a date value.
17. DAYOFYEAR(): It returns the day of the year for a date value.
18. EXTRACT(): It extracts parts from a date.
19. FROM_DAYS(): It returns a date value from a numeric representation of the day.
20. HOUR(): It returns the hour portion of a date value.
21. LAST_DAY(): It returns the last day of the month for a given date.
22. LOCALTIME(): It returns the current date and time.
23. LOCALTIMESTAMP(): It returns the current date and time.
24. MAKEDATE(): It returns the date for a certain year and day-of-year value.
25. MAKETIME(): It returns the time for a certain hour, minute, second combination.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write SQL queries to use various date functions.*
(a)Add 3 months in current date.

(b)Display months between date: ’17-02-04’ and ’17-02-05’.
(c)Display last date of month ‘2-feb-06’.
(d)If the date is 31-jul-13 then find out the date of friendship day.

A) Input-Output:

B) *Write SQL queries to use various numeric functions*

(a)Display integer value of 125.25.

(b)Display absolute value of(-15)

(c)Display ceil value of 55.65

(d)Display floor value of 100.2

(e)Display the square root of 16.

(f)Display square root of 20.

(g)Display result of 12 raised to 6.

(h)Display result of 24 mod 2.

(i)Show value of e3.

(j)2 is angle in radius find out cos, sin, tan value of 2.

(k)Show output of sign(-25), sign(25), sign(0).

B) Input-Output:

K. Practical related Quiz.

1) Which SQL function can be used to add or subtract a specified interval of time from a given date?

   a) DATEDIFF b) DATEADD
   c) DATEPART d) DATETIME

2) Which SQL function can be used to add or subtract a specified interval of time from a given date?

   a) DATEDIFF b) DATEADD
   c) DATEPART d) DATETIME

3) Which Oracle numeric function can be used to calculate the absolute value of a number?

   A) ROUND B) CEIL
   C) ABS D) MOD

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student’s engagement in practical activities (3) R3	Total (10)		

Date: _______________

Practical No.5: In-built Function (Part 2)

a. Write SQL queries to use various character functions.
(1)Find out length of string “hello world”.
(2)Change the case of ‘HELLO WORLD’ string to lower case.
(3)Change the case of ‘hello world’ to upper case.
(4)Display each word initial letter as capital letter : government polytechnic for girls.
(5)Find ‘put’ from string ‘computer’.
(6)Add 10 star on left side of India string.
(7)Add 10 star on right side of India string.
(8)Trim ‘ion’ from ‘information’.
(9)Trim ‘info’ from ‘information’.

b. Write SQL queries to use various conversion functions
(a)Convert 100000 in format 99,99,99
(b)Convert today’s date to MM-DD-YY format

A. Objective:

To gain the ability to manipulate and transform character data efficiently, perform data formatting and validation, and enhance sorting and comparison operations for character data within the database. These functions enable effective data manipulation, formatting, and analysis, improving overall data management and decision-making processes.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"SQL queries to use various character functions"

This practical is expected to develop the following skills.

1. You can develop skills in string manipulation, substring extraction, case conversion, and string replacement. These skills enable effective data manipulation and formatting for improved data analysis and management.

"SQL queries to use various conversion functions"

This practical is expected to develop the following skills.

1. Developing skills in conversion functions allows you to manipulate data formats effectively, convert between different data types, and perform data validation tasks within your SQL queries, enhancing your overall data management and analysis capabilities.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to understand and use various character and conversion function.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Character Function :

ASCII() : Returns numeric value of left-most character
CHAR() : Returns the character for each integer passed
CHARINDEX() : Returns the position of a substring within the given string.
CONCAT_WS() : Returns concatenate with separator
CONCAT() : Returns concatenated string

DIFFERENCE() : returns an integer value measuring the difference between the SOUNDEX() values of two different expressions(strings).
ESCAPE() : Returns a text with escaped characters.
FORMAT() : Returns the formatted string.
LEFT() : Returns the extracting string.
LEN() : Returns the length of the given string.
LOWER() : Returns returns the lower case of the given string.
LTRIM() : Returns a string after removing all the white spaces and characters from the string found on the left side.
NCHAR() : Returns the Unicode character based on the number code.
PATINDEX() : Returns the position of a pattern in a string.
QUOTENAME() : Returns a string with a delimiter
REPLACE() : Returns a new string by replacing all the occurrences of the particular part of a string (substring) with a specified string.
REPLICATE() : Returns the repeated value.
REVERSE() : Returns a reversed string.
RIGHT() : Returns the rightmost characters from the actual(current) string.
RTRIM() : Returns a string after removing all trailing blanks.
SOUNDEX() : Returns the Soundex string.
SPACE() : returns a string consisting of N number of space characters.
STR() : Returns a number as string.
SUBSTRING() : Returns the part of the character.
TRIM() : Returns a trimmed string.

<u>Conversion function :</u>

TO_CHAR :TO_CHAR function is used to typecast a numeric or date input to character type with a format model
TO_NUMBER : Convert a character string to a number format using the TO_NUMBER function :TO_NUMBER(char[, 'format_model'])
TO_DATE : Convert a character string to a date format using the TO_DATE function:
TO_DATE(char[, 'format_model'])

H. Safety and necessary Precautions followed

NA

I. Source code:

A) *Write SQL queries to use various character functions.*

(1)Find out length of string "hello world".
(2)Change the case of 'HELLO WORLD' string to lower case.
(3)Change the case of 'hello world' to upper case.

(4)Display each word initial letter as capital letter : government polytechnic for girls.

(5)Find ‘put’ from string ‘computer’.

(6)Add 10 star on left side of India string.

(7)Add 10 star on right side of India string.

(8)Trim ‘ion’ from ‘information’.

(9)Trim ‘info’ from ‘information’.

A) Input/Output:

B) *Write SQL queries to use various conversion functions*
(a)Convert 100000 in format 99,99,99
(b)Convert today’s date to MM-DD-YY format

B) Input/Output:

J. Practical related Quiz.

1: Which Oracle conversion function can be used to convert a string to uppercase?

A) UPPER B) LOWER
C) INITCAP D) SUBSTR

2: Which Oracle conversion function can be used to convert a numeric value to a character string?

A) TO_CHAR B) TO_NUMBER
C) TO_DATE D) LENGTH

3: Which Oracle string function can be used to concatenate multiple strings together?

A) LENGTH B) SUBSTR
C) CONCAT D) INSTR

K. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

L. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student’s engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.6: Group Functions and Operators

a. Write SQL queries to use various group function and operators using tables created.

A. Objective:

Gain proficiency in utilizing various group functions and operators in Oracle for data aggregation, analysis, and manipulation in SQL queries.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

"Write SQL queries to use various group function and operators using different tables"

This practical is expected to develop the following skills.

1. Proficiency in using group functions like COUNT, AVG, MAX, MIN, and SUM for data aggregation and analysis.
2. Ability to apply operators such as arithmetic operators, concatenation operators, logical operators, comparison operators, and pattern matching operators for data manipulation and filtering in SQL queries

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to understand group function and operators.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Group Functions in Oracle:

Group functions in Oracle are used to perform calculations on a set of rows and return a single result. These functions operate on a group of rows and return a single value for each group. Here are some commonly used group functions in Oracle:

COUNT: It is used to count the number of rows in a table or a specific column.

SUM: It calculates the sum of the values in a column.

AVG: This function computes the average value of a column.

MAX: It returns the maximum value from a column.

MIN: It retrieves the minimum value from a column.

Operators in Oracle:

Operators in Oracle are used to perform various operations on data, such as arithmetic calculations, logical comparisons, and string manipulations. Here are some commonly used operators in Oracle:

Arithmetic Operators:

Addition (+)

Subtraction (-)

Multiplication (*)

Division (/)

Comparison Operators:

Equal to (=)

Not equal to (<> or !=)

Greater than (>)

Less than (<)

Greater than or equal to (>=)

Less than or equal to (<=)

Logical Operators:

AND: Returns true if all conditions are true.

OR: Returns true if any of the conditions are true.

NOT: Negates a condition, returns true if the condition is false.

Concatenation Operator:

The concatenation operator (||) is used to concatenate two or more strings.

Pattern Matching Operator:

LIKE: It is used for pattern matching in SQL queries. It allows the use of wildcard characters (%) to match patterns.

Understanding and utilizing these group functions and operators in Oracle SQL queries allows for effective data analysis, manipulation, and filtering..

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write SQL queries to use various group function using tables created in Practical 3.*
(a)Retrieve specified information for the account holder who are not in ‘Ahmedabad’.
(b)Retrieve specified information for the account holder who are not in ‘Ahmedabad ’or ‘Vadodara’.
(c)Retrieve those records of Account holder whose balance between is 50000 and 100000.
(d)Retrieve those records of Account holder whose balance not between is 50000 and 100000.
(e)Display only those records whose amount is 5000, 25000, 30000.
(f)Display only those records whose amount not in 5000, 25000, 30000.
(g)Find the total transaction amount of account holder from transaction table.
(h)Find minimum amount of transaction.
(i)Find maximum amount of transaction.
(j)Count the total account holders.
(k)Count only that recordthat’s made of payment is ‘cash’.
(l)Count only those records whose transaction made in the month of ‘MAY’
(m)Find the average value of transaction.
(n)Display total balance for each branch from account table.
(o)Display total balance for account in Ahmadabad city.
(p)Find total amount of mode of pay ‘cash’ from transaction table.

A) Input/Output:

K. Practical related Quiz.

1) Which group function is used to find the total number of records in a table?

a) COUNT
b) SUM
c) AVG
d) MAX

2) Which operator is used to concatenate two or more strings in SQL?

a) +
b) *
c) /
d) –

3) Which group function is used to find the average value of a column in a table?

a) COUNT
b) SUM
c) AVG
d) MAX

4) Which operator is used for pattern matching in SQL?

a) =
b) LIKE
c) NOT
d) OR

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

4) **Assessment-Rubrics**

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.7: Set and Join Operators**.**

a. Write SQL query for set operators and join operations.

A. Objective:

The objective of learning set and join operators in SQL is to efficiently retrieve and combine data from multiple tables, enabling complex data analysis and reporting.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Write SQL query for set operators and join operations”

This practical is expected to develop the following skills.

1. develop skills in integrating and analyzing data from multiple tables
2. optimizing queries for improved performance, and enhancing your problem-solving abilities in working with complex databases..

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to use set and join operators in database tables.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Set Operators:

UNION:

The UNION operator combines the result sets of two or more SELECT statements, eliminating duplicate rows. The syntax is as follows:

```
SELECT column1, column2, ... FROM table1 UNION SELECT column1, column2, ... FROM table2;
```

INTERSECT:

The INTERSECT operator returns the common rows between two result sets, excluding duplicates. The syntax is:

```
SELECT column1, column2, ... FROM table1 INTERSECT SELECT column1, column2, ... FROM table2;
```

EXCEPT/MINUS:

The EXCEPT or MINUS operator returns the rows present in the first SELECT statement but not in the second SELECT statement. The syntax is:

```
SELECT column1, column2, ... FROM table1 EXCEPT SELECT column1, column2, ... FROM table2;
```

Join Operators:

INNER JOIN:

The INNER JOIN operator retrieves rows from two or more tables that have matching values in the specified columns. The syntax is:

```
SELECT column1, column2, ... FROM table1 INNER JOIN table2 ON table1.column = table2.column;
```

LEFT JOIN:

The LEFT JOIN operator returns all rows from the left table and the matching rows from the right table. If no match is found, NULL values are returned for the right table. The syntax is:

```
SELECT column1, column2, ... FROM table1 LEFT JOIN table2 ON
table1.column = table2.column;
```

RIGHT JOIN:

The RIGHT JOIN operator returns all rows from the right table and the matching rows from the left table. If no match is found, NULL values are returned for the left table. The syntax is:

```
SELECT column1, column2, ... FROM table1 RIGHT JOIN table2 ON
table1.column = table2.column;
```

FULL JOIN:

The FULL JOIN operator returns all rows from both tables, including the non-matching rows. If no match is found, NULL values are returned for the unmatched table. The syntax is:

```
SELECT column1, column2, ... FROM table1 FULL JOIN table2 ON
table1.column = table2.column;
```

These operators provide powerful ways to manipulate and combine data from multiple tables in SQL queries.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write SQL query for set operators and join operations.(Use tables of Practical 3)..*

A) Input/Output:

K. Practical related Quiz.

1. Which set operator in SQL is used to combine the results of two SELECT statements and remove duplicates?

a) UNION

b) INTERSECT

c) EXCEPT/MINUS

d) JOIN

2. Which join operator in SQL returns only the matching rows from both tables based on a specified condition?

a) INNER JOIN

b) LEFT JOIN

c) RIGHT JOIN

d) FULL JOIN

3. Which join operator in SQL returns all rows from the left table and the matching rows from the right table, with NULL values for non-matching rows?

a) INNER JOIN

b) LEFT JOIN

c) RIGHT JOIN

d) FULL JOIN

4. Which join operator in SQL returns all rows from both tables, including non-matching rows with NULL values?

a) INNER JOIN

b) LEFT JOIN

c) RIGHT JOIN

d) FULL JOIN

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: _______________

Practical No.8: SQL Constraint

a. Apply the concept of integrity/data constraints while creating/altering a table.

A. Objective:

The objective of applying the concept of integrity/data constraints while creating or altering a table is to ensure data consistency, accuracy, and reliability within the database. By defining appropriate constraints, you can enforce rules and restrictions on the data stored in tables, preventing invalid or inconsistent data from being inserted or updated. This helps maintain data integrity and improves the overall quality and usability of the database.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Apply the concept of integrity/data constraints while creating/altering a table”

This practical is expected to develop the following skills.

Create student registration form using tags.

1) you can develop skills in data validation, database design, error prevention and handling, query optimization.

2) you can develop skills in data consistency maintenance, enhancing your overall database management abilities and ensuring accurate and reliable data storage.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to use Integrity constraints on tables and also maintain data constraint.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Constraints in SQL:

1. Primary Key Constraint:

The primary key constraint ensures that each record in a table has a unique identifier. It is defined when creating a table or altering an existing table. The syntax is as follows:

-- Creating a table with a primary key constraint

CREATE TABLE table_name (column1 data_type PRIMARY KEY, column2 data_type, ...);

-- Adding a primary key constraint to an existing table

ALTER TABLE table_name ADD CONSTRAINT constraint_name PRIMARY KEY(column1);

2. Unique Constraint:

The unique constraint ensures that values in a column or a group of columns are unique across the table. It can be defined during table creation or added later using the ALTER TABLE statement. The syntax is:

-- Creating a table with a unique constraint

```
CREATE TABLE table_name (column1 data_type,column2 data_type,...

CONSTRAINT constraint_name UNIQUE (column1, column2, ...));
```

-- Adding a unique constraint to an existing table

```
ALTER TABLE table_name

ADD CONSTRAINT constraint_name UNIQUE (column1, column2, ...);
```

3. **Foreign Key Constraint:**

The foreign key constraint establishes a relationship between two tables by linking a column in one table to the primary key of another table. It is defined when creating a table or using the ALTER TABLE statement. The syntax is:

-- Creating a table with a foreign key constraint

```
CREATE TABLE table_name1 (column1 data_type, column2 data_type, ...

CONSTRAINT constraint_name FOREIGN KEY (column1, column2, ...)

REFERENCES table_name2 (column1, column2, ...));
```

-- Adding a foreign key constraint to an existing table

```
ALTER TABLE table_name1 ADD CONSTRAINT constraint_name FOREIGN
KEY (column1, column2, ...) REFERENCES table_name2 (column1, column2,
...);
```

4. **Check Constraint:**

The check constraint allows you to specify a condition that must be satisfied for the data in a column. It is defined when creating a table or using the ALTER TABLE statement. The syntax is:

-- Creating a table with a check constraint

```
CREATE TABLE table_name (column1 data_type, column2 data_type, ...

CONSTRAINT constraint_name CHECK (condition));
```

-- Adding a check constraint to an existing table

```
ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK
(condition);
```

These syntax examples demonstrate how to apply integrity and data constraints in SQL during table creation or alteration.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Apply the concept of integrity/data constraints while creating/altering a table:*
(a)Create table SalesPeople where Snumvarchar2(4) ***P.K.****(first letter should start with S) , Sname varchar2(20) NOT NULL,City Varchar2(15),Mobile_No Number(10).*

(b)Create table Customer where cnum varchar2(4) P.K(first letter should start with C), Cname varchar2(20) NOT NULL,city varchar2(20),Rating number(3) DEFAULT 10,Snum number(4) F.K (where snum refers salespeople table)

(c)Create table Order where Order_No number(4) P.K, Amount number(5), odate varchar2(10), cnum varchar2(4) F.K,(where cnum refers customer table), snum varchar2(4) F.K (where snum refers Salespeople table)

A) Input/Output:

K. Practical related Quiz.

1. Which constraint ensures that each record in a table has a unique identifier?

a) Primary Key Constraint b) Unique Constraint
c) Foreign Key Constraint d) Check Constraint

2. Which constraint ensures that values in a column or a group of columns are unique across the table?

a) Primary Key Constraint b) Unique Constraint
c) Foreign Key Constraint d) Check Constraint

3. Which constraint establishes a relationship between two tables by linking a column in one table to the primary key of another table?

a) Primary Key Constraint b) Unique Constraint
c) Foreign Key Constraint d) Check Constraint

4. Which constraint allows you to specify a condition that must be satisfied for the data in a column?

a) Primary Key Constraint b) Unique Constraint
c) Foreign Key Constraint d) Check Constraint

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. sAssessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student’s engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.9: Privilege Commands

a. Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER command.

A. Objective:

To gain the ability to manage user accounts and access privileges in a database, including creating new users, granting permissions, revoking permissions, and removing user accounts. To understand and effectively control user access to the database, ensuring security and maintaining data integrity.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER command”

This practical is expected to develop the following skills.

1. develop skills in user management, access control, security management, and user administration.
2. enabling you to effectively manage user accounts, control permissions, maintain security, and troubleshoot access-related issues in a database environment.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to perform privilege commands onto database.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

CREATE USER Command:

The CREATE USER command is used to create a new user account in the database. It specifies the username and password for the user.
The syntax is: CREATE USER username IDENTIFIED BY 'password';
Example: CREATE USER johnsmith IDENTIFIED BY 'password123';

GRANT Command:

The GRANT command is used to grant specific privileges or permissions to a user. It allows users to perform operations on database objects.
The syntax is: GRANT privilege_type ON object_name TO username;
Example: GRANT SELECT, INSERT ON Employees TO johnsmith;

REVOKE Command:

The REVOKE command is used to revoke or remove specific privileges or permissions from a user. It restricts the user's ability to perform certain operations on database objects.
The syntax is: REVOKE privilege_type ON object_name FROM username;
Example: REVOKE INSERT ON Employees FROM johnsmith;

DROP USER Command:

The DROP USER command is used to remove a user account from the database. It permanently deletes the user and any associated privileges.
The syntax is: DROP USER username;
Example: DROP USER johnsmith;

These examples demonstrate the usage of CREATE USER, GRANT, REVOKE, and DROP USER commands in SQL. These commands are essential for user management, access control, and security administration in a database environment.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP USER commands.*
(a)Create a new user User1.
(b)Grant connect, resource and dba rights to user1.
(c)Grant all permissions of account table to user1.
(d)Remove select permission on account table for user1.
(e)Delete user1.

A) Input/Output:

K. Practical related Quiz.

1. Which command is used to create a new user account in the database?
a) CREATE USER b) GRANT
c) REVOKE d) DROP USER

2. Which command is used to grant specific privileges or permissions to a user?
a) CREATE USER b) GRANT
c) REVOKE d) DROP USER

3. Which command is used to remove specific privileges or permissions from a user?
a) CREATE USER b) GRANT
c) REVOKE d) DROP USER

4. Which command is used to remove a user account from the database?
a) CREATE USER b) GRANT
c) REVOKE d) DROP USER

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.10: PL/SQL Concepts

a. Write a program in PL/SQL to show the user of Cursor
b. Write a program in PL/SQL to show the user of Stored Procedures.
c. Write a program in PL/SQL to show the user of Stored Function.
d. Write a program in PL/SQL to show the user of Database Triggers

A. Objective:

Understand how to iterate over result sets and perform row-level operations in database queries. Learn to create reusable code blocks for executing complex database operations with parameters and error handling. Gain knowledge of creating and using functions to encapsulate logic for data manipulation and retrieval within SQL statements. Understand how to enforce data integrity, automate tasks, and perform actions based on predefined events or conditions in the database.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.
Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
Engineering Tools, Experimentation and Testing: Apply modern *engineering* tools and appropriate technique to conduct standard tests and measurements.
Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Write a program in PL/SQL to show the user of Cursor”
This practical is expected to develop the following skills.

1. Row-level data manipulation and processing.
2. Efficient traversal and retrieval of data from result sets.

“Write a program in PL/SQL to show the user of Stored Procedures”
This practical is expected to develop the following skills.

1. Creating reusable code blocks for complex database operations, improving code organization and maintainability.
2. Implementing parameterized procedures for flexibility in executing database operations with varying input values.

“Write a program in PL/SQL to show the user of Stored Function”

This practical is expected to develop the following skills.

1. Creating modular and reusable logic for data manipulation and retrieval within SQL statements.
2. Incorporating stored functions within SQL expressions for enhanced query flexibility and efficiency.

“Write a program in PL/SQL to show the user of Database Triggers”

This practical is expected to develop the following skills.

1. Implementing modular and reusable code for complex data processing and manipulation.
2. Leveraging stored functions to enhance database performance by reducing round-trips to the database and promoting code reusability.

D. Expected Course Outcomes(Cos)

CO4: Apply SQL Commands for creating, manipulating and controlling databases.

E. Practical Outcome(PRo)

Students will be able to create, store, fetch, delete cookies and session.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Cursor:

- A cursor in Oracle is a database object that provides a way to retrieve and manipulate result sets in a systematic manner.

- Cursors are used to iterate over rows returned by a SQL query.
- They can be used to perform row-level operations such as fetching data, updating records, or deleting rows within the result set.
- Cursors can be defined and used within PL/SQL blocks to process data sequentially or selectively based on specific conditions.
- Syntax for defining and using a cursor:

```
DECLARE

  cursor_name CURSOR [(cursor_parameters)]

IS

    select_statement;

BEGIN

  -- Cursor operations

END;
```

Example:

```
DECLARE

  CURSOR emp_cursor IS SELECT employee_id, first_name, last_name FROM employees;

  emp_record emp_cursor%ROWTYPE;

BEGIN

  OPEN emp_cursor;

  LOOP

    FETCH emp_cursor INTO emp_record;

    EXIT WHEN emp_cursor%NOTFOUND;

    -- Process the current row

    -- ...

  END LOOP;

  CLOSE emp_cursor;

END;
```

Stored Function:

- A stored function in Oracle is a named PL/SQL block that can accept input parameters, perform calculations, and return a single value.
- Stored functions encapsulate reusable logic that can be called within SQL statements or other PL/SQL blocks.
- They are typically used to perform complex calculations, data manipulation, or data retrieval tasks.

Syntax for creating a stored function:

```
CREATE [OR REPLACE] FUNCTION function_name
  [(parameter_name [IN | OUT] data_type [, ...])]
  RETURN return_data_type
IS
  [local_variable_declarations]
BEGIN
  -- Function logic and calculations
  [RETURN expression;]
END;
```

Example:

```
CREATE FUNCTION calculate_salary_bonus(emp_id NUMBER) RETURN NUMBER
IS
  bonus NUMBER := 0;
BEGIN
  -- Calculate bonus based on employee ID
  -- ...
  RETURN bonus;
END;
```

Stored Procedure:

- A stored procedure in Oracle is a named PL/SQL block that can accept input parameters and perform a series of actions.
- Stored procedures encapsulate a set of SQL statements or business logic that can be executed as a unit.
- They are often used to perform database operations such as inserting, updating, or deleting records.
- Syntax for creating a stored procedure:

```
CREATE [OR REPLACE] PROCEDURE procedure_name

  [(parameter_name [IN | OUT] data_type [, ...])]

IS

  [local_variable_declarations]

BEGIN

  -- Procedure logic and actions

END;
```

Example:

```
CREATE PROCEDURE insert_employee(emp_id NUMBER, first_name VARCHAR2, last_name VARCHAR2)

IS

BEGIN

  -- Insert employee record into the database

  INSERT INTO employees (employee_id, first_name, last_name)

  VALUES (emp_id, first_name, last_name);

END;
```

Trigger:

- A database trigger in Oracle is a named PL/SQL block that is automatically executed in response to a specific event or action on a table.
- Triggers are used to enforce business rules, maintain data integrity, and automate actions within the database.
- They can be triggered by events such as insertions, updates, or deletions on a table.
- Syntax for creating a database trigger:

```
CREATE [OR REPLACE] TRIGGER trigger_name
    {BEFORE | AFTER | INSTEAD OF} {INSERT | UPDATE | DELETE}
    ON table_name
    [FOR EACH ROW]
  DECLARE
    -- Declaration of variables
  BEGIN
    -- Trigger logic and actions
  END;
  /
```

- BEFORE/AFTER/INSTEAD OF: Specifies the timing of the trigger execution. "BEFORE" triggers execute before the triggering event, "AFTER" triggers execute after the triggering event, and "INSTEAD OF" triggers are used with views to replace the triggering event.
- INSERT/UPDATE/DELETE: Specifies the triggering event that activates the trigger.
- ON table_name: Specifies the table on which the trigger is defined.
- [FOR EACH ROW]: Optional clause indicating that the trigger is a row-level trigger and can access the affected row.
- Example of a trigger:

```
CREATE OR REPLACE TRIGGER audit_employee_changes
  AFTER UPDATE ON employees
  FOR EACH ROW
```

```
DECLARE
  audit_message VARCHAR2(100);
BEGIN
  audit_message := 'Employee ' || :OLD.employee_id || ' details updated.';

  -- Log the audit message in the audit table
  INSERT INTO audit_table (message) VALUES (audit_message);
END;
/
```

- In this example, the trigger named audit_employee_changes is defined to execute after an update operation on the employees table. The trigger captures the old values of the updated row using the :OLD pseudorecord and generates an audit message. The audit message is then inserted into an audit_table to log the changes.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) Write a program in PL/SQL to show the user of Cursor.

A) Input/Output:

B) *Write a program in PL/SQL to show the user of Stored Procedures.*

B) Input/Output:

C) Write a program in PL/SQL to show the user of Stored Function**.**

C) Input/Output:

D) Write a program in PL/SQL to show the user of Database Triggers**.**

D) Input/Output:

K. Practical related Quiz.

1. What is a cursor in Oracle?
   a) A database object that stores data temporarily
   b) A control structure used to iterate over result sets
   c) A database trigger used to enforce data integrity
   d) A stored procedure used to encapsulate business logic

2. What is the primary purpose of a stored function in Oracle?
   a) To retrieve and manipulate data from the database
   b) To perform complex calculations and return a single value
   c) To execute a series of SQL statements in a transaction
   d) To handle events and automate tasks within the database

3. Which of the following best describes a stored procedure in Oracle?
   a) A database object used to trigger automatic actions based on events
   b) A container for a set of SQL statements that can be executed as a unit
   c) A function that returns a result set for further processing
   d) A mechanism used to define and enforce referential integrity rules

4. What is the purpose of a database trigger in Oracle?
   a) To encapsulate business logic and calculations
   b) To handle errors and exceptions during data manipulation
   c) To enforce data integrity and perform automatic actions
   d) To retrieve and modify data stored in database tables

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.11: Refining database design through normalization

a. Normalization of database.

Consider the following relational schema:

a)Employee(eid, ename, salary, mngr id, mngrname, mngrage,ecity)

b)Student(no,name, bdate, city, semester, subname, submarks, percentage)

For above schemas, answer the following questions:-

a) Determine the primary key for given relation.

b) Is there any pitfall in design of relation schema? Explain it.

c) Find out various dependencies among given attributes of given relation schema.

d) Normalize given relation schema up to possible normal form Normalization of database for a given relational schema.

A. Objective:

Normalization of database to reduce data redundancy, improve data integrity, and enhance database performance through efficient organization and structuring of data.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

Design/ development of solutions: Design solutions for *engineering* well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

Life-long learning: Ability to analyze individual needs and engage in updating in the context of technological changes in field of engineering.

C. Expected Skills to be developed based on competency:

“Normalization of database.”

This practical is expected to develop the following skills.

1. Analyzing data dependencies and relationships.
2. Designing and creating normalized database schemas.

3. Applying normalization techniques to eliminate data redundancy and improve data integrity.

D. Expected Course Outcomes(Cos)

CO5: Apply concepts of normalization to design an optimal database.

E. Practical Outcome(PRo)

Students will be able to perform normalization the database to improve integrity.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Functional Dependency:

Functional dependency describes the relationship between attributes in a table. A functional dependency exists when the value of one or more attributes determines the value of another attribute.

For example, consider a table called "Employees" with attributes: EmployeeID, FirstName, LastName, and Department. Here, EmployeeID uniquely identifies each employee, so we can say that EmployeeID functionally determines the FirstName, LastName, and Department attributes.

Normal Forms:

a. First Normal Form (1NF): This form ensures that each attribute within a table contains only atomic values. In other words, there are no repeating groups or arrays within an attribute. To achieve 1NF, we break down the table into smaller, more granular tables.

For example, let's say we have a table called "Students" with attributes: StudentID, Name, and Courses (which stores multiple courses for each student). To convert it to 1NF, we split it into two tables:

Table 1: Students (StudentID, Name)

Table 2: Courses (StudentID, Course)

b. Second Normal Form (2NF): In addition to 1NF, 2NF ensures that each non-key attribute is functionally dependent on the entire primary key. If an attribute is only functionally dependent on part of the primary key, it should be moved to a separate table.

For example, consider a table called "Orders" with attributes: OrderID, ProductID, ProductName, and Quantity. Here, OrderID and ProductID together form the primary key. Since ProductName is functionally dependent only on ProductID, we split the table into two:

Table 1: Orders (OrderID, ProductID, Quantity)

Table 2: Products (ProductID, ProductName)

c. Third Normal Form (3NF): In addition to 2NF, 3NF ensures that there are no transitive dependencies. Transitive dependencies occur when an attribute depends on another attribute through a separate attribute. To achieve 3NF, we further split tables to remove transitive dependencies.

For example, consider a table called "Employees" with attributes: EmployeeID, DepartmentID, DepartmentName, and ManagerName. Here, DepartmentName is transitively dependent on EmployeeID through DepartmentID. We split the table into three:

Table 1: Employees (EmployeeID, DepartmentID)

Table 2: Departments (DepartmentID, DepartmentName)

Table 3: Managers (DepartmentID, ManagerName)

Primary Key:

A primary key uniquely identifies each record in a table. It ensures that there are no duplicate entries and allows establishing relationships with other tables. Primary keys can be single attributes or a combination of attributes.

For example, in the "Customers" table, the primary key could be "CustomerID," which uniquely identifies each customer.

Functional Dependencies and Normalization Rules:

Functional dependencies help determine normalization rules and identify potential issues such as partial dependencies and transitive dependencies. By identifying these dependencies, we can apply normalization rules to eliminate anomalies and ensure data integrity.

Partial Dependency Example:

Consider a table called "Books" with attributes: BookID, Title, Author, and Publisher. If the primary key is BookID, but the attribute Author depends only on Title, we have a partial dependency issue. To resolve this, we split the table into two:

Table 1: Books (BookID, Title, Publisher)

Table 2: Authors (Title, Author)

Transitive Dependency Example:

In the "Employees" table, if the attribute ManagerName depends on the DepartmentID, which depends on the EmployeeID, we have a transitive dependency issue. To resolve this, we split it into three tables:

Table 1: Employees (EmployeeID, DepartmentID)

Table 2: Departments (DepartmentID, DepartmentName)

Table 3: Managers (DepartmentID, ManagerName)

Now, the "Managers" table contains only the DepartmentID and ManagerName attributes, removing the transitive dependency between EmployeeID and ManagerName.

Denormalization:

While normalization aims to reduce redundancy, there are cases where denormalization may be used to improve performance. Denormalization involves selectively reintroducing redundancy by combining tables or duplicating data.

For example, let's consider a normalized database with separate tables for "Customers" and "Orders." If frequent joins between these tables impact performance, we may denormalize by adding the "CustomerName" attribute directly into the "Orders" table. This denormalization eliminates the need for joins in some scenarios, improving query performance.

It's important to note that denormalization should be used judiciously and carefully to balance performance gains with the potential for data inconsistency.

Understanding the theory of normalization and applying it correctly enables the design of efficient, maintainable, and reliable databases that minimize data redundancy, ensure data integrity, and optimize performance.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Normalization of database.*

Consider the following relational schema:
a)Employee(eid, ename, salary, mngr id, mngrname, mngrage,ecity)
b)Student(no,name, bdate, city, semester, subname, submarks, percentage)

For above schemas, answer the following questions:-
a) Determine the primary key for given relation.
b) Is there any pitfall in design of relation schema? Explain it.
c) Find out various dependencies among given attributes of given relation schema.
d) Normalize given relation schema up to possible normal form.

A) Input/Output:

K. Practical related Quiz.

1. Which of the following is the primary goal of database normalization?
a) Increasing data redundancy
b) Improving data consistency
c) Introducing data anomalies
d) Reducing data integrity

2. Which normal form ensures that there are no repeating groups or arrays within an attribute?
a) 1NF (First Normal Form)
b) 2NF (Second Normal Form)
c) 3NF (Third Normal Form)
d) BCNF (Boyce-Codd Normal Form)

3. Which attribute(s) is/are typically used as a primary key in a table?
a) Any attribute in the table
b) The attribute with the most unique values
c) A combination of attributes that uniquely identifies each record
d) The attribute with the longest length

4. Denormalization is the process of:
a) Breaking down a table into smaller tables
b) Combining tables or duplicating data to improve performance
c) Removing duplicate records from a table
d) Creating complex relationships between tables

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				**Faculty Signature**	**Date**
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student’s engagement in practical activities (3) R3	Total (10)		

Date: ______________

Practical No.12: Transaction Management

a. Prepare a report on transaction management concepts for concurrent access of database by multiple users.

A. Objective:

The objective of preparing a report on transaction management concepts for concurrent access of a database by multiple users is to examine the techniques and strategies used to handle simultaneous transactions, ensuring data integrity and preventing conflicts. The report aims to provide insights into concurrency control mechanisms and their implications for efficient and reliable database operations.

B. Expected Program Outcomes (POs)

Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods.

C. Expected Skills to be developed based on competency:

"Prepare a report on transaction management concepts for concurrent access of database by multiple users."

This practical is expected to develop the following skills.

1. Research and analysis skills to gather and evaluate information about transaction management and concurrency control mechanisms.
2. Communication skills to effectively convey complex transaction management concepts and their implications in multi-user database environments.

D. Expected Course Outcomes(Cos)

CO6:)Explain transaction management concepts for concurrent use of database.

E. Practical Outcome(PRo)

Students will be able to ensure data integrity, consistency, and reliability by effectively managing and controlling database transactions.

F. Expected Affective domain Outcome(ADos)

1) Follow safety practices.
2) Follow Coding standards and practices.
3) Demonstrate working as a leader/ a team member.
4) Follow ethical practices.
5) Maintain tools and equipment.

G. Prerequisite Theory:

Transactions are a set of operations used to perform a logical set of work. It is the bundle of all the instructions of a logical operation. A transaction usually means that the data in the database has changed. One of the major uses of DBMS is to protect the user's data from system failures. It is done by ensuring that all the data is restored to a consistent state when the computer is restarted after a crash. The transaction is any one execution of the user program in a DBMS. One of the important properties of the transaction is that it contains a finite number of steps. Executing the same program multiple times will generate multiple transactions.

Example: Consider the following example of transaction operations to be performed to withdraw cash from an ATM vestibule.

Steps for ATM Transaction

1. Transaction Start.
2. Insert your ATM card.
3. Select a language for your transaction.
4. Select the Savings Account option.
5. Enter the amount you want to withdraw.
6. Enter your secret pin.
7. Wait for some time for processing.
8. Collect your Cash.
9. Transaction Completed.

A transaction can include the following basic database access operation.

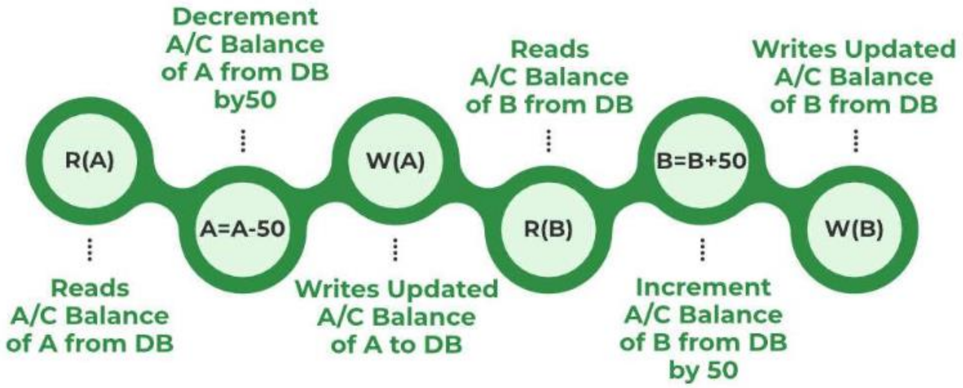
Read/Access data (R): Accessing the database item from disk (where the database stored data) to memory variable.

Write/Change data (W): Write the data item from the memory variable to the disk.

Commit: Commit is a transaction control language that is used to permanently save the changes done in a transaction

Example: Transfer of 50₹ from Account A to Account B. Initially A= 500₹, B= 800₹. This data is brought to RAM from Hard Disk.

```
R(A) -- 500      // Accessed from RAM.
A = A-50        // Deducting 50₹ from A.
W(A)--450        // Updated in RAM.
R(B) -- 800      // Accessed from RAM.
B=B+50          // 50₹ is added to B's Account.
W(B) --850       // Updated in RAM.
commit          // The data in RAM is taken back to Hard Disk
```



All instructions before committing come under a partially committed state and are stored in RAM. When the commit is read the data is fully accepted and is stored on Hard Disk.

If the transaction is failed anywhere before committing we have to go back and start from the beginning. We can’t continue from the same state. This is known as Roll Back.

Desirable Properties of Transaction (ACID Properties)

For a transaction to be performed in DBMS, it must possess several properties often called ACID properties.

A – Atomicity

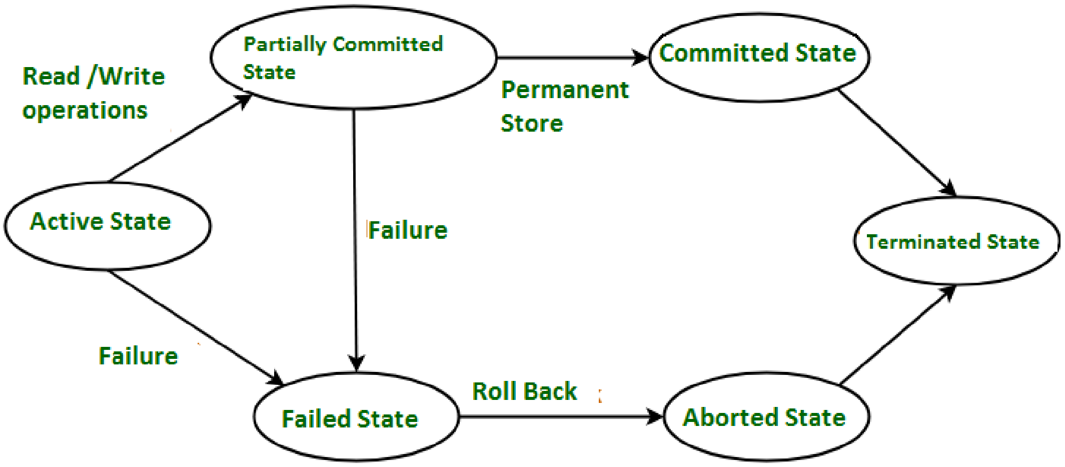
C – Consistency

I – Isolation

D – Durability

Transaction States

Transactions can be implemented using SQL queries and Servers. In the below-given diagram, you can see how transaction states work.



Transaction States in DBMS

The transaction has the four properties. These are used to maintain consistency in a database, before and after the transaction.

Property of Transaction:

1. Atomicity
2. Consistency
3. Isolation
4. Durability

Atomicity:

- It states that all operations of the transaction take place at once if not, the transactions aborted.
- There is no midway, i.e., the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.
- Atomicity involves the following two operations:
- Abort: If a transaction aborts, then all the changes made are not visible.
- Commit: If a transaction commits then all the changes made are visible.

Consistency:

- The integrity constraints are maintained so that the database is consistent before and after the transaction.
- The execution of a transaction will leave a database in either its prior stable state or anew stable state.
- The consistent property of database states that every transaction sees a consistent database instance.
- The transaction is used to transform the database from one consistent state to another consistent state.
-

Isolation:

- It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.
- In isolation, if the transaction T1 is being executed and using the data item X, then that data item can't be accessed by any other transaction T2 until the transaction T1ends.
- The concurrency control subsystem of the DBMS enforced the isolation property

Durability:

- The durability property is used to indicate the performance of the database's consistent state. It states that the transaction made the permanent changes.
- They cannot be lost by the erroneous operation of a faulty transaction or by the system failure. When a transaction is completed, then the database reaches a

state known as the consistent state. That consistent state cannot be lost, even in the event of a system's failure.

- The recovery subsystem of the DBMS has the responsibility of Durability property.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment/ Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	*As Per Batch Size*
2	Operating System	Windows/ Linux/ MAC	
3	Software	Oracle	
4	Text Editor	Notepad, Notepad++, Sublime Text or similar	

I. Safety and necessary Precautions followed

NA

J. Source code:

A) *Prepare a report on transaction management concepts for concurrent access of database by multiple users.*

A) Input/Output:

K. Practical related Quiz.

1. Which of the following is a key property of a transaction in SQL?

a) Consistency	b) Complexity
c) Concurrency	d) Conformity

2. What does the ACID acronym stand for in the context of transaction management?

a) Atomicity, Commitment, Isolation, Durability

b) Authentication, Consistency, Isolation, Durability

c) Atomicity, Consistency, Integrity, Durability

d) Atomicity, Consistency, Isolation, Durability

3. Which concept in transaction management ensures that all changes made within a transaction are saved permanently, even in the event of a system failure?

a) Concurrency control	b) Atomicity
c) Durability	d) Isolation

L. References / Suggestions

1) https://www.w3schools.com/sql/default.asp
2) https://www.tutorialspoint.com/dbms/
3) https://www.geeksforgeeks.org/dbms/

M. Assessment-Rubrics

Marks Obtained				Faculty Signature	Date
Program Correctness (4) R1	Implementation and Presentation Methodology (3) R2	Student's engagement in practical activities (3) R3	Total (10)		

Database Management

4331603

Lab manuals are prepared by

Miss Darshana Trivedi

Lecturer - IT

R.C.T.I., Ahmedabad

Shri Jay Raval

Lecturer - IT

R.C.T.I., Ahmedabad

Shri Shashikant Sarvaiya

Lecturer - IT

R.C.T.I., Ahmedabad

Branch Coordinator

Shri Nandu Fatak

(HOD-IT)

Government Polytechnic for Girls, Ahmedabad

Committee Chairman

Shri R. D. Raghani

(HOD-EC)

Principal (I/C)

Government Polytechnic, Gandhinagar