K.S. SCHOOL OF ENGINEERIN AND MANAGEMENT, BENGALURU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATABASE MANAGEMENT SYSTEMS

BCS403

INTEGRATED PROFESSIONAL CORE COURSE

Prepared by: -

Mrs. Amitha S
Associate Professor
Dept of CSE, KSSEM

Mrs. Soungandhika N
Assistant Professor
Dept of CSE, KSSEM

LAB MANUAL

INSTITUTION VISION & MISSION

VISION:

"To impart quality education in engineering and management to meet technological, business and societal needs through holistic education and research"

MISSION:

K.S. School of Engineering and Management shall,

- Establish state-of-art infrastructure to facilitate effective dissemination of technical and Managerial knowledge.
- Provide comprehensive educational experience through a combination of curricular and experiential learning, strengthened by industry-institute-interaction.
- Pursuesocially relevant research and disseminate knowledge.
- Inculcate leadership skills and foster entrepreneurial spirit among students.

DEPARTMENT VISION & MISSION

VISION:

"To produce quality Computer Science professional, possessing excellent technical knowledge, skills, personality through education and research."

MISSION:

Department of Computer Science and Engineering shall,

- Provide good infrastructure and facilitate learning to become competent engineers who meet global challenges.
- Encourages industry institute interaction to give an edge to the students.
- Facilitates experimental learning through interdisciplinary projects.
- Strengthen soft skill to address global challenges.

SYLLABUS

DBMS LABORATORY
[As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2023 -2024) SEMESTER – IV

Subject Code	BCS403	CIE Marks	50
Number of Lecture Hours/Week	0:0:2:0	SEE Marks	50
Total Number of Lecture Hours	8-10 lab slots	Exam Hours	03

CREDITS – 04

Course objectives:

- To Provide a strong foundation in database concepts, technology, and practice.
- To Practice SQL programming through a variety of database problems.
- To Understand the relational database design principles.
- To Demonstrate the use of concurrency and transactions in database.
- To Design and build database applications for real world problems.
- To become familiar with database storage structures and access techniques.

Teaching-Learning Process

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) needs not to be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding
9. Use any of these methods: Chalk and board, Active Learning, Case Studies

LABORATORY COMPONENT

1	Create a table called Employee & execute the following. Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION) 1. Create a user and grant all permissions to the user. 2. Insert the any three records in the employee table contains attributes EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2	Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR, SAL & execute the following. 1. Add a column commission with domain to the Employee table. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.
3	Queries using aggregate functions (COUNT, AVG, MIN, MAX, SUM), Groupby, Orderby. Employee (E_id, E_name, Age, Salary) 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employee table 3. Find the Maximum age from employee table. 4. Find the Minimum age from employee table. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
5	Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extrct the values from the cursor. Close the cursor.Employee (E_id, E_name, Age, Salary)
6	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.
7	Install an Open Source NoSQL Data base Mango DB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute Mango DB basicQueries using CRUD operations.

Course outcomes (Course Skill Set):

At the end of the course, the student will be able to:

- Describe the basic elements of a relational database management system
- Design entity relationship for the given scenario.
- Apply various Structured Query Language (SQL) statements for database manipulation.
- Analyse various normalization forms for the given application.
- Develop database applications for the given real world problem.
- Understand the concepts related to NoSQL databases.

Suggested Learning Resources:

Text Books:

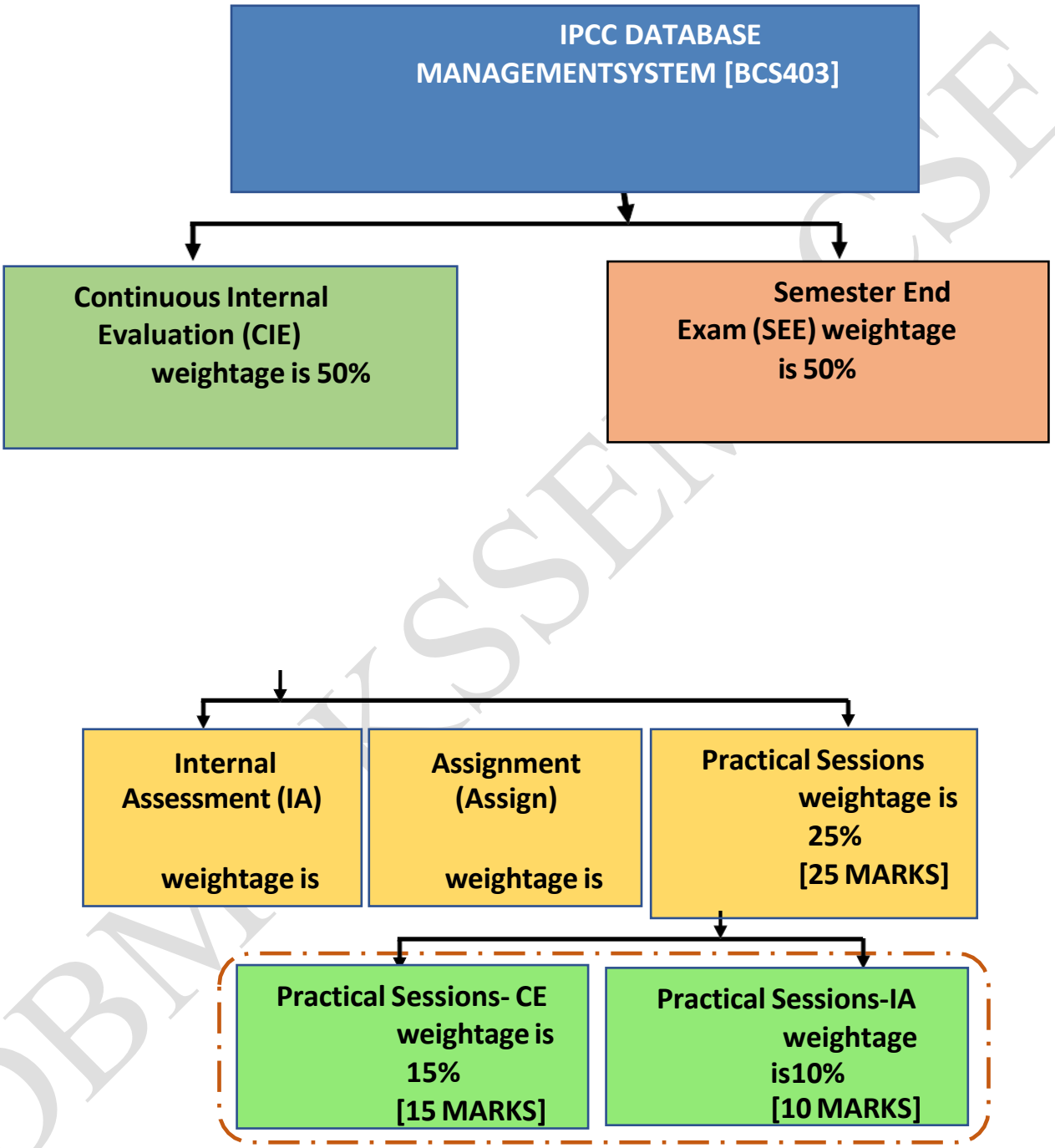
1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning
Mini Project:

• Project Based Learning

ASSESSMENT PROCESS FOR INTEGRATED PROFESSIONAL CORE COURSE (IPCC)



ASSESSMENT DETAILS (BOTH CIE AND SEE)

- The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%.
- The minimum passing mark for the CIE is 40% of the maximum marks (20 marks).
- A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE of Theory:

- **Two Unit Tests** each of **30 Marks (duration 01.30 hour**)
  -scale down the average marks to **15 marks**.
- **Two assignments** each of **10 Marks (Any Two Assessment Methods are followed)**
  -scale down the average marks to **10 marks.**
- **Total CIE=25 Marks (minimum 10 marks)**

CIE of Practical:

- **Practical Sessions** need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **25 marks (minimum 10 marks)**.

Note: Minimum of 80% of the laboratory components have to be covered *Note: Minimum of 80% of the laboratory components have to be covered.*

Split-up of Marks used Practical Sessions:

Split up of Marks	Practical Sessions- Continuation Evaluation (CE) Methodology / Process Steps per Experiment	Marks
#R1	Observation, Write up of Procedure / Algorithm/ Program and Execution of experiment	07
#R2	Record writing	05
#R3	Viva – Voce (Questions & Answers on relevant Experiment /Topic)	03
	Total Marks for each experiment	**15 marks**
	Practical Sessions-Internal Assessment (IA)	

#R1	Write-up of Procedure/Program/Algorithm	15
#R2	Conduction/Execution	25
#R3	Viva-Voce	10
	Total Marks	**50(Scale down to 10 marks)**

- The sum of two tests, two assignments, and practical sessions will be out of 50 marks and will **CIE methods /question paper has to be designed to attain the different levels of Bloom'staxonomy as per the outcome defined for the course**

SEMESTER END EXAMINATION(SEE):

- Theory SEE will be conducted by University as per the scheduled timetable, with commonquestion papers for the subject (**duration 03 hours)**
  1. The question paper will have ten questions. Each question is set for 20 marks.
  2. There will be 2 questions from each module. Each of the two questions under a module(with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
- The students have to answer 5 full questions, selecting one full question from each module

INTRODUCTION TO DBMS COMMAND'S

INTRODUCTION TO ORACLE

SQL

SQL stands for Structured Query Language. SQL is used to create, remove, alter the database and database objects in a database management system and to store, retrieve, update the data in a database.

SQL is a standard language for creating, accessing, manipulating database management system. SQL works for allmodern relational database management systems, like SQL Server, Oracle, MySQL, etc.

Different types of SQL commands

SQL commands can be categorized into five categories based on their functionality

DDL (Data Definition language)

A Data Definition Language (DDL) statement is used to define the database structureor schema.

Aim: To study and execute the DDL commands in RDBMS.DDL commands:

1. CREATE
2. ALTER
3. DROP
4. RENAME
5. TRUNCATE

1. CREATE Command

CREATE is a DDL command used to create databases, tables, triggers and otherdatabase objects.

Syntax to create a new table:

CREATE TABLE table_name(
column_Name1 data_type (**size of** the **column**) ,
column_Name2 data_type (**size of** the **column**) ,
column_Name3 data_type (**size of** the **column**) ,
... ……..column_NameN data_type (**size of** the **column**)
) ;

Suppose, you want to create a **Student** table with five columns in the SQLdatabase. To do this, you have to write the following DDL command:

Example 1:

```
CREATE TABLE Student(
    Roll_No. int,
    First_Name    varchar    (20),
    Last_Name     varchar    (20),
    Age int,
    Marks int,
    Dob Date
    ) ;
SQL>desc Student;
```

Example 2:

```
create table Employee(
 empid varchar (10),
 empname varchar (20),
 gender varchar (7),
 age number (3),
dept varchar (15),
doj Date
);
  SQL> desc Employee
```

Example 3:

```
create table BOOK
 (
Book_id varchar (4),
Title varchar (10),
Publisher_name varchar (10),
Pub_year int
);
SQL> desc BOOK;
```

2.ALTER:

This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify takseasily.

Syntax: –

ALTER TABLE table_name ADD column_name datatype;

ADD:

SQL> alter table employee add (designation varchar (15));Table altered.

II.MODIFY

SQL> alter table employee modify (designation varchar (20));Table altered

Example 1:

ALTER TABLE StudentADD

CGPA number; SQL>desc

Student; **Example 2:**

ALTER TABLE Employee

ADD Salary number;

SQL>desc Employee;

Example 3:

ALTER TABLE BOOK

ADD Author_nmae varchar (20);

SQL>desc Student;

3. RENAME:

It is possible to change name of table with or without data in it using simpleRENAME command. We can rename any table object at any point of time.

Syntax –
RENAME <Table Name> To <New_Table_Name>;

Example:

RENAME TABLE Employee To EMP;

4. TRUNCAT:

This command is used to remove all rows from the table, but the structure of thetable still exists.

Syntax –
Syntax to remove an existing table.

TRUNCATE TABLE table_name;

Example:

TRUNCATE TABLE Student;

5. DROP:

This command is used to remove an existing table along with its structure fromthe Database.

Syntax –

Syntax to drop an existing table.DROP TABLE table_name;

Example: DROP TABLE Student_info;

DML (DATA MANIPULATION LANGUAGE):

Data manipulation language allows the users to query and manipulate data inexisting schema in object.

It allows following data to insert, delete, update and recovery data in schemaobject.

DML COMMANDS:

- ❖ INSERT
- ❖ UPDATE
- ❖ DELETE

1.INSERT

This command is used to enter the information or values into a row. We canconnect one or more records to a single table within a repository using thisinstruction.

Syntax:

Insert into Table_ Name Values (column1, column2,.....);

Example:

CREATE TABLE Student(
Roll_No **int,**
First_Name **varchar** (20),
Last_Name **varchar** (20),
Marks **int,**
Dob **Date**
) ;

SQL>desc Student;

SQL> insert into Student values('01','Adit,'k'',25,'11-02-2004');

SQL> insert into Student values (02,"Arpitha","S", 20,'21-12-2003'); SQL> insert into Student values(03,"Jorge","D", 20, 18,'10-08-2001'); Insert 2 more rows.

SQL>desc Student;

2.UPDATE

This allows the user to update the particular column value using the whereclause condition. This command is used to alter existing table records.

Syntax:

UPDATE <table_ name>
SET <column_ name = value>WHERE
condition;

Example:

```
UPDATE Students
  SET  Marks= 21
     WHERE First_ name = “Arpitha”;

SQL>desc Students;
```

2. DELETE

a) Delete some rows

DELETE statement is used to delete rows from a table. Generally, DELETEstatement removes one or more records from a table.

Syntax:

DELETE FROM table_name WHERE condition;

Example:

Delete from Students where Roll_no=’111’;

b) Delete All rows:

- It will remove all the rows from the table and does not free the spacecontained by the table.

Syntax:

DELETE FROM table_name;

Example:

Delete from student;

DQL (Data Query Language)

DQL stands for the. DQL command is used for fetching the data. DQL command is used for selecting data from the table, view, temp table, table variable, etc. Thereis only one command under DQL which is the SELECT command.

Syntax: **SELECT * FROM Employee;**

The SELECT statement can be used in many ways.

1. Selecting some columns:

To select specified number of columns from the table the Following command isused.

Syntax:

SELECT column_ name FROM table_ name;

Example:

Select First_ name, Last_ name from Students;

SQL> desc Students;

2. Select All Columns:

To select all columns from the table * is used instead of column names.Syntax:

SELECT * FROM table_name;

Example:

Select * from Students;

SQL> desc Students;

3. Select using DISTINCT:

The DISTINCT keyword is used to return only different values (i.e.) thiscommand does not select the duplicate values from the table.

Syntax:

SELECT DISTINCT column name(s) FROM table_name;

Example:

SELECT DISTINCT Roll_No FROM Students;

4. Select using IN:

If you want to get the rows which contain certain values, the best way to do it isto use the IN conditional expression.

Syntax:

SELECT column name(s) FROM table_name
WHERE Column name IN (value1, value2,……,value-n);

Example:

SELECT * FROM students
WHERE students_name IN (Arpitha, Jorge);

5.Select using BETWEEN:

BETWEEN can be used to get those items that fall within a range.Syntax:

SELECT column name FROM table_name

WHERE Column name BETWEEN value1 AND value2

Example:

SELECT * FROM **student** WHERE mark **BETWEEN** 80 and 100;

6.Renaming:

The select statement can be used to rename either a column or the entire table.Syntax:

Renaming a column:

SELECT column name AS new name FROM table_name;

Example:

SELECT **First_name** As **Name** FROM Students;

Renaming a table:

RENAME old_table_name TO **new_table_name;**

Example:

RENAME Student TO **Stu details;**

7.SELECT DATE

It is used to retrieve a date from a database. If you want to find a particular datefrom a database, you can use this statement.

Syntax:

SELECT Column_Names(S) from **table-name**
WHERE condition(date_column);

Example: **SELECT * FROM Students WHERE** DOB >= '11-12-2000';

8. SELECT NULL

Null values are used to represent missing unknown data.There can be two conditions:

1. Where SQL is NULL
2. Where SQL is NOT NULL

If in a table, a column is optional, it is very easy to insert data in column or updatean existing record without adding a value in this column. This means that field hasnull value.

Where SQL is NULL:

Syntax:

SELECT COLUMN_NAME(S) FROM TABLE_NAMEWHERE COLUMN_NAME IS NULL;

Example:

SELECT Student_name, Marks **FROM** STUDENTS
WHERE MARKS **IS NULL;**

Where SQL is NOT NULL:

Syntax:

SELECT COLUMN_NAME(S) FROM TABLE_NAMEWHERE COLUMN_NAME IS NOT NULL;

Example:

SELECT Student_name, Marks **FROM** STUDENTS
WHERE MARKS **IS NOT NULL**;

ORDER BY Clause

ORDER BY is a clause in SQL which shows the result-set of the SELECT statementin either ascending or descending order.

a) with one row

Syntax:

SELECT Column_Name FROM Table_NameORDER BY Column_Name;

Example:

SELECT * FROM Students ORDER BY Reg_no;

b) With Multiple row

Syntax:

SELECT Column_Name(S) FROM Table_NameORDER BY Column_Name(S);

Example:

SELECT reg_no,first_name,marks FROM Students ORDER BY first_name,marks;

c) Ascending Order(ASC)/Descending Order(DESC)

SELECT Column_Name(S) FROM Table_Name ORDER BY Column_Name(S) ASC;

Example:

SELECT * FROM Students ORDER BY Marks ASC;

d) with WHERE Clause

Syntax:

SELECT Column_Name(S)FROM Table_Name
WHERE [condition] ORDER BY Column_Name [ASC | DESC];

Example:

SELECT * FROM Students WHERE Marks >80 ORDER BY Marks;

GROUP BY clause

GROUP BY is an SQL keyword used in the SELECT query for arranging the samevalues of a column in the group by using SQL functions.

Syntax:

SELECT Column_Name(S) **FROM** Table_Name
GROUP BY Column_Name(S);

Example:

SELECT COUNT (marks), Student_name **GROUP BY** marks;

a) with MIN clause

Group by with MIN clause shows the minimum value for the given whereclause.

Syntax:

SELECT MIN(Column_Name) **FROM** Table_Name;

Example:

SELECT MIN (Marks) FROM Students;

b) with MAX clause

Group by with MIN clause shows the minimum value for the given whereclause.

Syntax:

SELECT Column_Name, **MAX**(Column_Name) **FROM** Table_Name;

Example:

SELECT Stu_Subject, MAX (Marks) FROM Students;

c) COUNT() Function

The COUNT() function returns the number of rows in a database table.

Syntax:

COUNT(*)

Or

COUNT([ALL|DISTINCT] expression)

Example:

1. **SELECT COUNT (*)**
   FROM Student;

2. **SELECT COUNT(Marks)**
   FROM Student
   GROUP BY marks
   HAVING COUNT (*) > 50;

d) SUM() clause

It is used to return the total summed value of an expression.Syntax:

SELECT SUM (aggregate_expression)
FROM tables

[**WHERE** conditions];

Example:

```
SELECT SUM(Marks)
    FROM Student
        Where roll_no='111';
```

DCL (DATA CONTROL LANGUAGE)

DCL stands for data control language. DCL commands are used for providing andtaking back the access rights on the database and database objects. DCL command used for controlling user's access on the data. Most used DCL commands are GRANT and REVOKE

GRANT

used to provide access right to the user.

Syntax: GRANT INSERT, DELETE ON Employee TO user;

REVOKE

REVOKE command is used to take back access right from the user, it cancelsaccess right of the user from the
database object.

Syntax
REVOKE ALL ON Employee FROM user;

TCL (Transaction Control Language)

TCL commands are used for handling transactions in the database.Transactions ensure data integrity in the multi-user environment.

TCL commands can rollback and commit data modification in the database. Themost used TCL commands are COMMIT, ROLLBACK, SAVEPOINT, and SET TRANSACTION.

COMMIT
COMMIT command is used to save or apply the modification in the database.

ROLLBACK
ROLLBACK command is used to undo the modification.

SAVEPOINT
SAVEPOINT command is used to temporarily save a transaction, the transactioncan roll back to this point when it's needed.

Syntax:

Just write COMMIT or ROLLBACK or SAVEPOINT

Experiment-1

Create a table called Employee & execute the following. Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION)

1. Create a user and grant all permissions to the user.

2. Insert the any three records in the employee table contains attributes EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback.Check the result.

3. Add primary key constraint and not null constraint to the employee table.

4. Insert null values to the employee table and verify the result.

Solutions:

Create a table employee with the given constraints:

SQL> create table employee (empno number, ename varchar2(10), job varchar2(10), mgr number, sal number);

SQL> create table employee (empno number,
ename varchar (10),job
varchar (10), mgr_no
number, sal number,
commission number);

Table created

```
Name                      Null?    Type
------------------------------------ -----------------
EMPNO                              NUMBER
ENAME                              VARCHAR2(10)
JOB                                VARCHAR2(10)
MGR                                NUMBER
SAL                                NUMBER
```

SQL>desc employee;

Insert any five records into the table: -

insert into Employee values(101,'abhi','manager',1234,10000,'70');

insert into employee values(102,'rohith','analyst',2345,9000,'65');

insert into employee values(103,'david','analyst',3456,9000,'65');

insert into employee values(104,'rahul','clerk',4567,7000,'55');

insert into employee values(105,'pramod','salesman',5678,5000,'50');

SQL>select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50

Solutions:

1. Create a user and grant all permissions to the user.

//connect to oracle database first

Provide user name as: **'/' as sysdba**

//create user

Create user c##dbms identified by dbms403;

User created.

//grant the permission

Grant connect, resource, dba to c##dbms;Grant

succeeded.

//connect to the user now Connect

c##dbms/dbms403;

Connected.

//check the user now

Show user;

2. Insert the any three records in the employee table and use rollback.Check the result

SQL>select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50

Insert new row

SQL> insert into employee values(106,'shashi','HR',5509,50000,'80');

SQL>select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50
106	Shashi	HR	5509	50000	80

SQL>rollback Rollback

completed

1. Add primary key constraint and not null constraint to the employeetable.

2.

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50

SQL> alter table employee modify (empno number **primary key**,
ename varchar (10) **not null);**

Table altered SQL>desc

Employee;

Name	Null?	Type
EMPNO	NOT NULL	NUMBER
ENAME	NOT NULL	VARCHAR2(10)
JOB		VARCHAR2(10)
MANAGER_NO		NUMBER
SAL		NUMBER
COMMISSION		NUMBER

3. Insert null values to the employee table and verify the result.

insert into employee values (106,'shashi','HR',5509,' ' ,80);

1 row inserted

SQL> select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50
106	Shashi	HR	5509	NULL	80

Viva Question:

1. **What is data?**

   Data is a collection of information gathered by observations, measurements,research or analysis.

2. **What is database?**

   A database is an electronically stored, systematic collection of data. It can containany type of data, including words, numbers, images, videos, and files.

3. **What is DBMS?**

   Database Management Systems (DBMS) are software systems used to store,retrieve, and run queries on data.

4. **What is a Database system?**

   A database is an organized collection of structured information, or data, typicallystored electronically in a computer system.

5. **What are the advantages of DBMS?**

   The advantages of database management include improved data integrity, consistency, and security, efficient data access and sharing, and reduced data redundancy and inconsistency.

6. **What is relational database?**

   A relational database is a collection of information that organizes data in predefined relationships where data is stored in one or more tables (or "relations") of columns and rows.

7. **What is Table?**

   A table is an arrangement of data in rows and columns, or possibly in a more complex

structure.

8. What is a Tuple?

A tuple is an ordered sequence of values. The values can be repeated, but their number is always finite.

9. What is Columns?

column or pillar in architecture and structural engineering is a structural element that transmits, through compression, the weight of the structure above to other structural elements below.

10. What is a query?

A query is a question or a request for information expressed in a formal manner.

Experiment-2

Create a table called Employee that contain attributes EMPNO, ENAME, JOB,MGR, SAL) execute the following.

1. **Add a column commission with domain to the Employee table.**
2. **Insert any five records into the table.**
3. **Update the column details of job**
4. **Rename the column of Employee table using alter command.**
5. **Delete the employee whose Empno is 105.**

Solution:

```
SQL> create table employee
(empno number,
ename varchar (10),
job varchar (10), mgr number,
sal number);
```

Table created.

SQL> desc employee;

Name	Null?	Type
EMPNO		NUMBER
ENAME		VARCHAR2(10)
JOB		VARCHAR2(10)
MGR		NUMBER
SAL		NUMBER

1. Add a column commission with domain to the Employee table.

SQL> alter table employee add (commission number);

Table altered

SQL> desc employee

Name	Null?	Type
EMPNO		NUMBER
ENAME		VARCHAR2(10)
JOB		VARCHAR2(10)
MGR		NUMBER
SAL		NUMBER
COMMISSION		NUMBER

2. Insert any five records into the table.

Insert any five records into the table

insert into Employee values(101,'abhi','manager',1234,10000,'70'); insert into employee values(102,'rohith','analyst',2345,9000,'65');insert into employee values(103,'david','analyst',3456,9000,'65');insert into employee values(104,'rahul','clerk',4567,7000,'55');

insert into employee values(105,'pramod','salesman',5678,5000,'50');

SQL>select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50

3. Update the column details of job

SQL> update employee set job='trainee' where empno=103;

1 row updated.

SQL> select * from employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	trainee	3456	9000	65
104	rahul	clerk	4567	7000	55
105	pramod	salesman	5678	5000	50

4. Rename the column of Employee table using alter command.

SQL> alter table employee rename column mgr to manager_no;

Table altered.

Name	Null?	Type
EMPNO		NUMBER
ENAME		VARCHAR2(10)
JOB		VARCHAR2(10)
MANAGER_NO		NUMBER
SAL		NUMBER
COMMISSION		NUMBER

SQL>desc employee;

5. Delete the employee whose Empno is 105

SQL> delete employee where empno=105; 1

row deleted

SQL> select * from Employee;

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	abhi	manager	1234	10000	70
102	rohith	analyst	2345	9000	65
103	david	analyst	3456	9000	65
104	rahul	clerk	4567	7000	55

Viva Questions

1. **What is an Attribute**?

   A quality, character, or characteristic ascribed to someone or something has leadership attributes.

2. **What is Single valued Attributes** ?

   Single-valued attributes Single-valued attributes accept only one value. For single-valued attributes, the syntax is: attribute = value attribute = "value with spaces" Multi-valued attributes.

3. **What is Multi valued Attributes?**

   A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.

4. **What is Compound /Composite Attribute?**

   A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.

5. **What is Simple/Atomic Attributes?**

   A simple, or atomic, attribute is one that cannot be decomposed into meaningfulcomponents.

6. What is Stored Attribute?

Stored attributes are those attributes that are stored in the physical database for e.g date of birth.

7. What is Derived Attribute?

A derived attribute is one that can be figured out from other information. Anexample is "age". A person's age can be derived from date of birth.

8. What are Complex Attributes?

Complex attributes are formed by grouping together the attributes of compositeand multi-valued attributes.

9. What is Key Attribute?

In DBMS, key attributes refer to the specific fields or columns in a table that areused to uniquely identify each record in the table.

10. What are Non-Key Attributes?

The values of a primary key cannot be duplicated. Non-prime (non-key)attributes are those that are not the primary key attributes.

Experiment-3

Queries using aggregate functions (COUNT, AVG, MIN, MAX, SUM) Groupby, Orderby.

Employee (E_id, E_name, Age, Salary)

1. **Create Employee table containing all Records E_id, E_name, Age, Salary.**
2. **Count number of employee names from employee table**
3. **Find the Maximum age from employee table.**
4. **Find the Minimum age from employee table.**
5. **Find salaries of employee in Ascending Order.**
6. **Find grouped salaries of employees.**

Solution:

1. **Create Employee table containing all Records E_id, E_name, Age, Salary.**

```
SQL> create table employee (E_id number,
                            E_name varchar (10),age
                            number,
                            sal number);
```

Table created.

SQL>desc Employee;

Name	NULL?	Type
E_id		NUMBER
E_name		VARCHAR(10)
Age		NUMBER
Sal		NUMBER

2. **Count number of employee names from employee table**

Insert any five records into Employee table

insert into Employee values (10,'abhi',25 ,10000); insert into

employee values(20,'rohith',30,9000); insert into employee

values(30,'david',28,9000); insert into employee

values(40,'rahul',29,7000); insert into employee

values(50,'pramod',31,8000);

SQL>select * from employee;

E_id	E_name	Age	Sal
10	Abhi	25	10000
20	Rohith	30	9000
30	David	28	9000
40	Rahul	29	7000
50	Pramod	31	8000

SQL> select count(E_name) from Employee;

Count(E_name)
5

3.Find the Maximum age from employee table.

SQL>select max(age) from Employee;

Max(Age)
31

4. Find the Minimum age from employee table

SQL>select min(age) from Employee;

Min(Age)
25

5. Find salaries of employee in Ascending Order.

SQL> SELECT * FROM Students ORDER BY salary ASC;

E_id	E_name	Age	Sal
40	Rahul	29	7000
50	Pramod	31	8000
20	Rohith	30	9000
30	David	28	9000
10	Abhi	25	10000

6. Find grouped salaries of employees.

SQL> select salary from employee group by salary;

Sal
7000
8000
9000
10000

Viva Questions

1. What is an Attribute?

A table consists of several records(row), each record can be broken down into several smaller parts of data known as **Attributes**. The above **Employee** table consist of four attributes, **ID**, **Name**, **Age** and **Salary**.

2. What is Single valued Attributes?

An attribute, that has a single value for a particular entity. For example, age of aemployee entity.

3. What is Multi valued Attributes?

An attribute that may have multiple values for the same entity. For example colors of a car entity.

4. What is Compound /Composite Attribute?

Attribute can be subdivided into two or more other Attribute. For Example, Namecan be divided into First name, Middle name and Last name.

5. What is Simple/Atomic Attributes?

The attributes which cannot be divided into smaller subparts are called simple oratomic attributes. For example, age of employee entity

6. What is Stored Attribute?

An attribute, which cannot be derived from other attribute, is known as storedattribute. For example, BirthDate of employee.

7. What is Derived Attribute ?

Attributes derived from other stored attribute. For example age from Date ofBirth and Today's date.

8. What is Complex Attributes?

If an attribute of an entity, is built using composite and multivalued attributes, then these attributes are called complex attributes. For example, a person can havemore than one residence and each residence can have multiple phones, an addressphone for a person entity can be specified as – {Addressphone (phone
{(Area Code, Phone Number)}, Address(Sector Address (Sector Number,HouseNumber), City, State, Pin))}.Here {} are used to enclose multivalued attributes and () are used to enclose composite attributes with comma separating individualattributes.

9. What is Key Attribute ?

It represents primary key. It is an attribute, that has distinct value for each entity/element in an entity set. For example, Roll number in a Student Entity Type.

10. What is Non Key Attributes ?

These are attributes other than candidate key attributes in a table. For example Firstname is a non key attribute as it does not represent the main characteristics ofthe entity.

Experiment 4:

for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary.
CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)

Solution:

1. Create Customer Table:

```
create table Customers (id number,
                        name varchar (10),age
                        number,
                        sal number,
                        address varchar (50));
```

Table created.

2.Creating trigger

```
SQL> CREATE OR REPLACE TRIGGER salary_difference_trigger
BEFORE INSERT OR UPDATE OR DELETE ON CUSTOMERS
FOR EACH ROW
DECLARE
old_salary NUMBER;
new_salary NUMBER;
BEGIN
IF INSERTING OR UPDATING THEN
old_salary := NVL(:OLD.SALARY, 0);
new_salary := NVL(:NEW.SALARY, 0);
DBMS_OUTPUT.PUT_LINE('Salary difference:'|| (new_salary - old_salary));
ELSIF DELETING THEN
old_salary := NVL(:OLD.SALARY, 0);
DBMS_OUTPUT.PUT_LINE('Salary before deletion: ' || old_salary);
END IF;
END;
/
```

Trigger created.

3.Finding salary difference

```
insert into Customers values (10,'abhi',25 ,10000," Bangalore");
```

Salary difference:10000

```
Update customer set salary=40000 where id=10;
```

Salary difference:3000

Viva Questions

1. **What is a primary key?**

A primary key is a column whose values **uniquely identify every row** in a table.

2. **What are the conditions for a field to be a primary key?**

- No two rows can have the same primary key value.
- Every row must have a primary key value.
- The primary key field cannot be null.
- Value in a primary key column can never be modified or updated, if anyforeign key refers to that primary key.

3. **What is a Foreign Key?**

When a "one" table's primary key field is added to a related "many" table inorder to create the common field which relates the two tables, it is called a foreign key in the "many" table.
For example, the salary of an employee is stored in salary table. The relation isestablished via foreign key column “Employee_ID_Ref” which refers “Employee_ID” field in the Employee table.

4. **What is Super Key**?

A set of attributes (one or more) that collectively identifies an entity in an entityset.

5. **What is Candidate Key**

A minimal super key is called a candidate key. An entity set may have more thanone candidate key.

6. **What is a query?**

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

7. Define SQL Update Statement?

SQL Update is used to update data in a row or set of rows specified in the filtercondition.

8. Define SQL Delete Statement?

SQL Delete is used to delete a row or set of rows specified in the filter condition.

11.What is order by clause?

ORDER BY clause helps to sort the data in either ascending order to descending

Experiment-5:

Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extract the values from the cursor.Close the cursor. Employee (E_id, E_name, Age, Salary)

Solution:

1. **Create table Employee**

```
create table employee (E_id number,
                       E_name varchar (10),age
                       number,
                       sal number);
```

Table created.

2. **Insert values into the table**

Insert any five records into Employee table

```
insert into employee values (10,'abhi',25 ,10000);
insert into employee values(20,'rohith',30,9000);
select * from employee;
```

E_id	E_name	Age	Sal
10	Abhi	25	10000
20	Rohith	30	9000

3. **Create the cursor, extracting the value from Employee table and closethe cursor.**

```
DECLARE
  E_id Employee.E_id%TYPE;
  E_name Employee.E_name%TYPE;
 Age Employee.Age%TYPE;
Sal Employee.Salary%TYPE;

-- Declare cursor
CURSOR employee_cursor IS
SELECT E_id, E_name, Age, Sal
FROM employee;

-- Open the cursor
BEGIN
OPEN employee_cursor;

-- Fetch data from cursor
LOOP
FETCH employee_cursor INTO E_id, E_name, Age, Sal;
EXIT WHEN employee_cursor%NOTFOUND;
```

Output or use the fetched values
DBMS_OUTPUT.PUT_LINE ("Employee ID: " || E_id || ', Name:'; || E_name || ', Age: '; || Age ||,' Salary: '|| Sal);

END LOOP;
-- Close the cursor
CLOSE employee_cursor;
END;

Output:

Employee Id:10, Name: Abhi, Age=25, Salary=10000
Employee Id:20, Name: Rohith, Age=30, Salary=1000

Viva Questions

1. **Define Normalization.**

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

2. **Enlist the advantages of normalizing database.**

Advantages of normalizing database are:

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

3. **What is Entity?**

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities.

4. **What is entity set?**

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a students set may containall the students of a school; like wise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

5. **What is Relationship?**

The association among entities is called a relationship. For example, an employee **works_at** a department, a student **enrolls** in a course. Here, Works_at and Enrolls are called relationships.

6. **What is Relationship Set?**

A set of relationships of similar type is called a relationship set.

7. **What is Degree of Relationship?**

The number of participating entities in a relationship defines the degree of therelationship

8. Name the Degree of Relationship?

- Binary = degree 2
- Ternary = degree 3
- n-ary = degree n

9. What is Data Model?

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

10. What is E-R model?

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

Experiment-6:

Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist inthe second table, then that data should be skipped.

Solution:

Create table O-Rollcall:

```
Create table O_Rollcall
            ( roll int,  name varchar(20));
```

insert values into the table

```
insert into O_Rollcall('10','AJIET'); insert
into O_Rollcall('20'.'MITE'); insert into
O_Rollcall('30'.'NITTE');insert into
O_Rollcall('40'.'RVC'); insert into
O_Rollcall('50'.'IIT');
```

SQL>select * from O_Rollcall;

ROLL	NAME
10	AJIET
20	MITE
30	NITTE
40	RVC
50	IIT

Create table N_Rollcall:

```
Create table N_Rollcall
            (roll int,  name varchar (20));
```

insert values into the table

```
insert into N_Rollcall('60','ALIET');insert
into N_Rollcall('70'.'NITK'); insert into
N_Rollcall('80'.'MIT'); insert into
N_Rollcall('40'.'RVC'); insert into
N_Rollcall('50'.'IIT');
```

SQL>select * from N_Rollcall;

ROLL	NAME
60	ALIET
70	NITK
80	MIT
40	RVC
50	IIT

Create Procedure roll_details:

```
DECLARE
v_count NUMBER;
CURSOR c_new_rollcall IS
SELECT roll, name
FROM N_RollCall;
BEGIN
FOR new_rec IN c_new_ rollcall  LOOP
-- Check if the record already exists in O_RollCall2
SELECT COUNT (*)
INTO v_count
FROM O_RollCall2
WHERE roll= new_rec.roll;

-- If record doesn&#39;t exist, insert it
IF v_count = 0 THEN
INSERT INTO O_RollCall (roll, name)
VALUES (new_rec_roll, new_rec.name);
DBMS_OUTPUT.PUT_LINE ('Record inserted:' || new_rec. roll);
ELSE
DBMS_OUTPUT.PUT_LINE ('Record skipped:' || new_roll);
END IF;
END LOOP;
COMMIT;
END;
```

Output is

Procedure is created

Run 1:

Record inserted :60
Record inserted :70
Record inserted :80
Record Skipped:40
Record Skipped:50

SQL>select * from O_Rollcall;

ROLL	NAME
60	ALIET
70	NITK
80	MIT
40	RVC
50	IIT
10	AJIET
20	MITE
30	NITTE

Viva Questions

1. What is Mapping Cardinalities

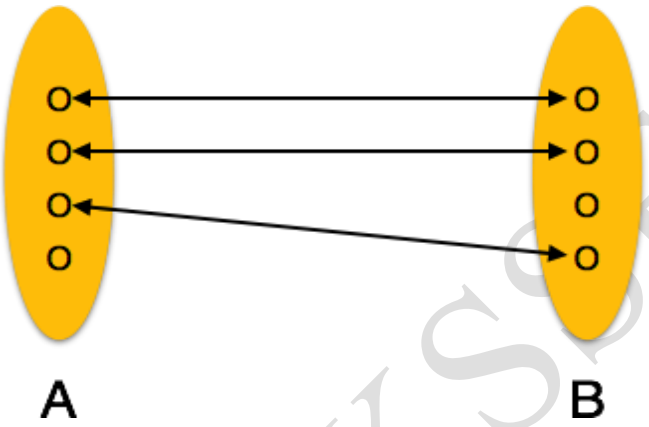
Cardinality defines the number of entities in one entity set, which can be associatedwith the number of entities of other set via relationship set.

2. What are the different types of Mapping

- One to one
- One to many
- Many to one
- Many to many

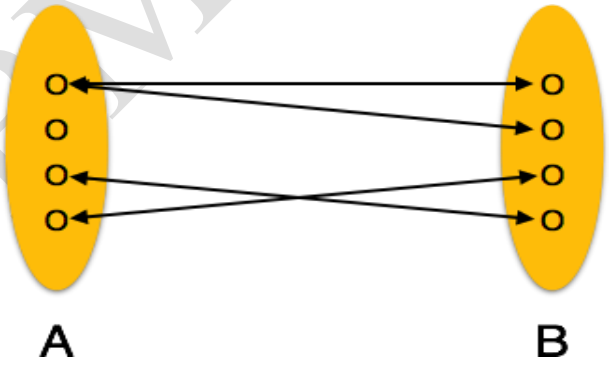
3. What is One-to-one mapping?

One entity from entity set A can be associated with at most one entity of entity set Band vice versa.



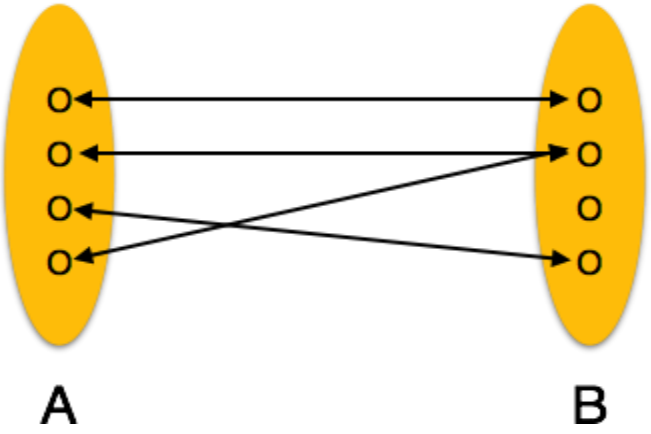
4. What is One-to-many mapping?

One entity from entity set A can be associated with more than one entities of entity setB however an entity from entity set B, can be associated with at most one entity.



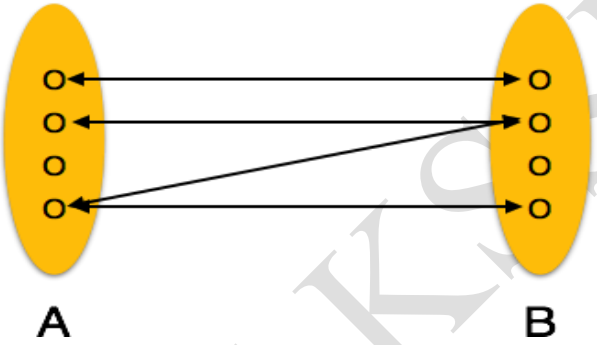
5. What is Many-to-one mapping?

More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



6. What is Many-to-many mapping?

One entity from A can be associated with more than one entity from B and viceversa.



7. What is DDL?

DDL stands for Data Definition Language. SQL queries like CREATE,ALTER, DROP and RENAME come under this.

8. What is DML?

DML stands for Data Manipulation Language. SQL queries like SELECT,INSERT and UPDATE come under this.

9. What is DCL?

DCL stands for Data Control Language. SQL queries like GRANT andREVOKE come under this.

Experiment-7:

Install an Open-Source NoSQL Data base Mongo DB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MongoDB basicQueries using CRUD operations.

How to Install and Configure MongoDB in Ubuntu?

MongoDB is a popular NoSQL database offering flexibility, scalability, and ease of use. Installing and configuring MongoDB in Ubuntu is a straightforward process, but it requires careful attention to detail to ensure a smooth setup.
In this guide, we'll learn how to install and configure MongoDB in Ubuntu. We'll walk you through each step, from installation to configuration, enabling you to harness the power of MongoDB on your Ubuntu system.
Let's look at the requirements for installing MongoDB in Ubuntu.

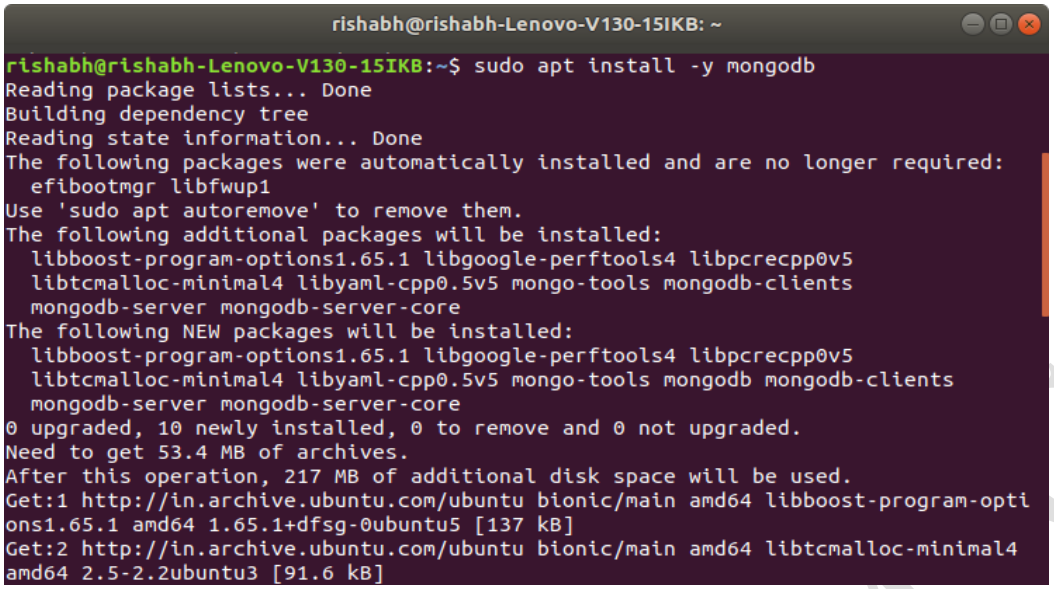
Steps to Install and Configure MongoDB in Ubuntu

MongoDB can be installed on Ubuntu with the use of the following commands. These commands are easy to run on the terminal and make the installation process handy. Follow the steps given below to install MongoDB:
Step 1: First you need to update and upgrade your system repository to install MongoDB. Type the following command in your terminal and then press Enter.

sudo apt update && sudo apt upgrade

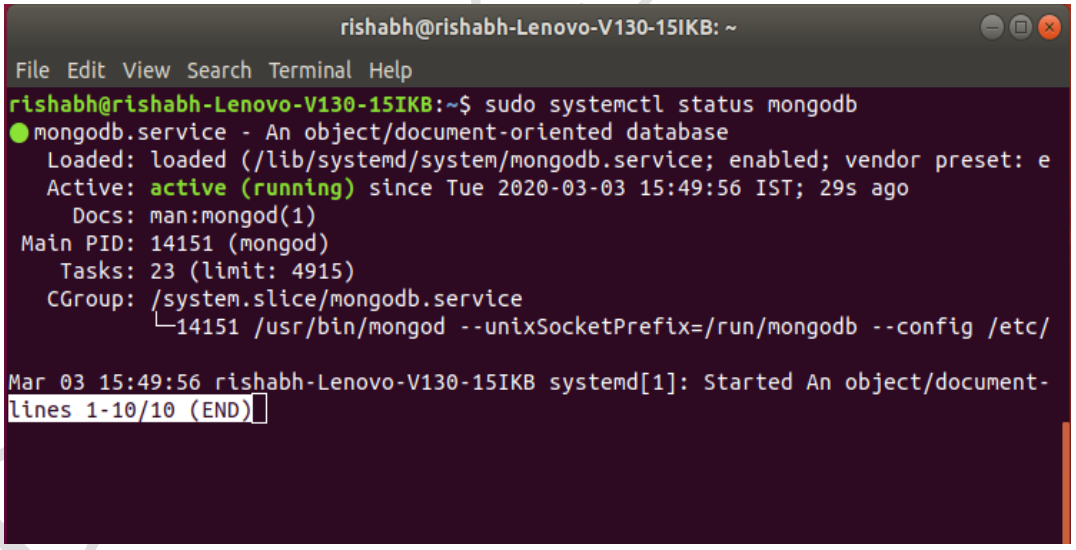
Step 2: Now, install the MongoDB package using 'apt'. Type the following command and press Enter.

sudo apt install -y mongo db



Step 3: Check the service status for MongoDB with the help of following command:

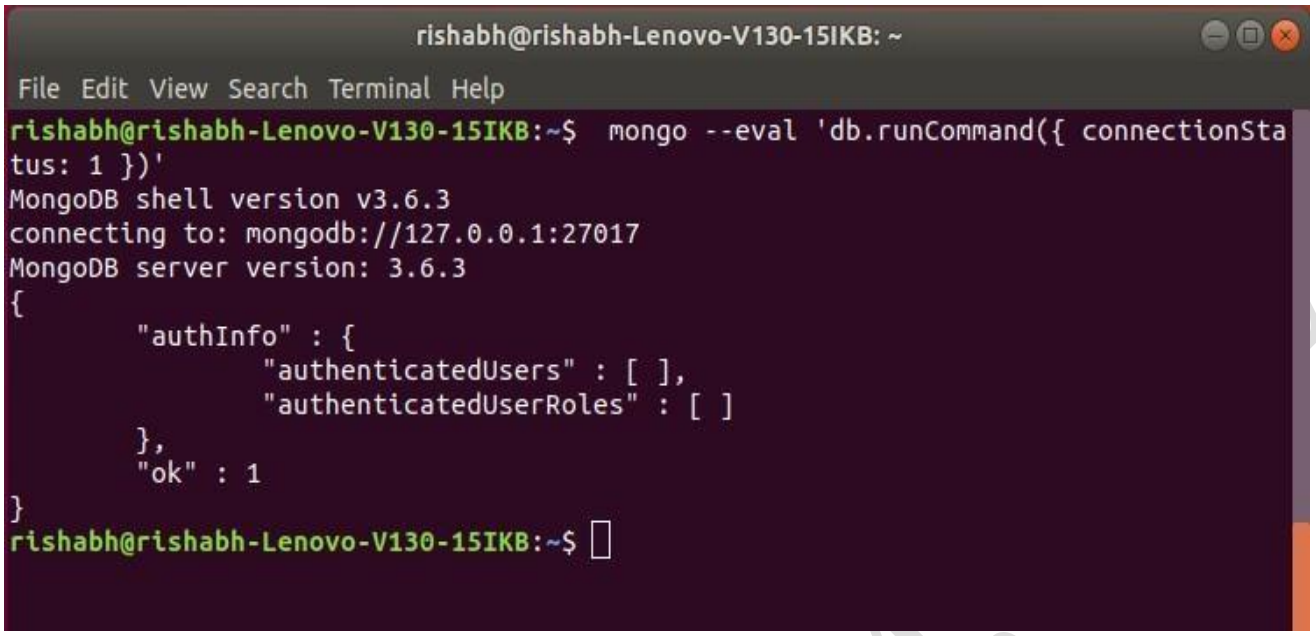
sudo systemctl status mongo db



systemctl verifies that MongoDB server is up and running.

Step 4: Now check if the installation process is done correctly and everything is working fine. Gothrough the following command:

mongo --eval 'db.runCommand ({ connectionStatus: 1 })'



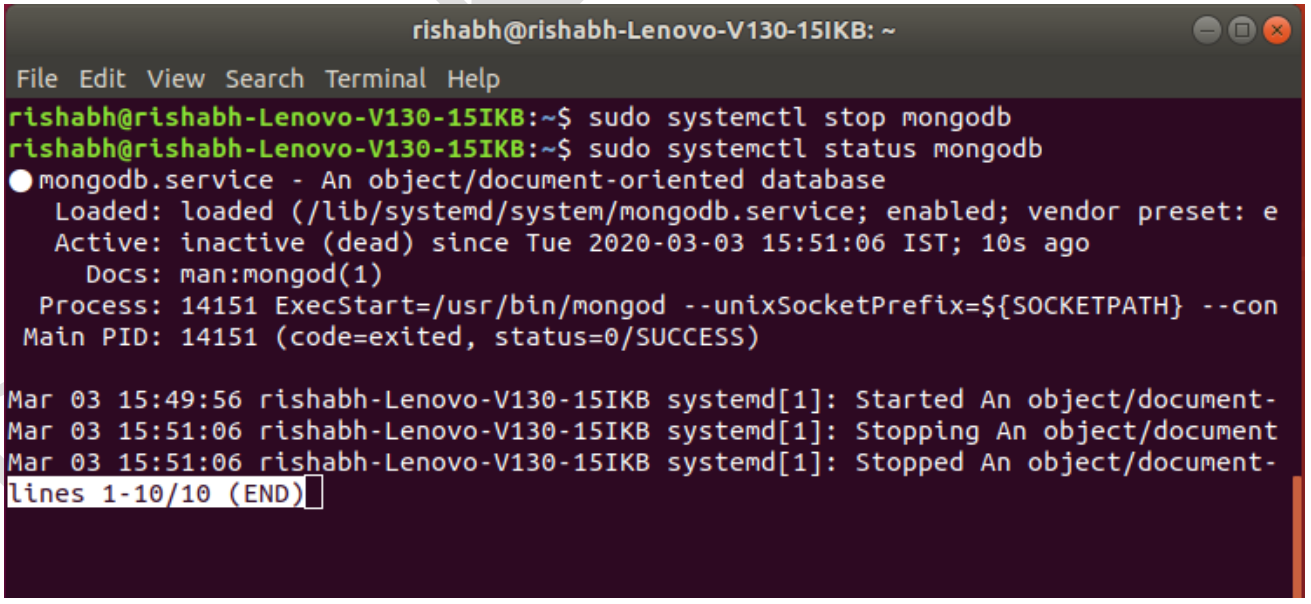
the value "1" in ok field indicates that the server is working properly with no errors.

Step 5: MongoDB services can be started and stopped with the use of following commands: To stoprunning the MongoDB service, use command :

sudo systemctl stop mongo db

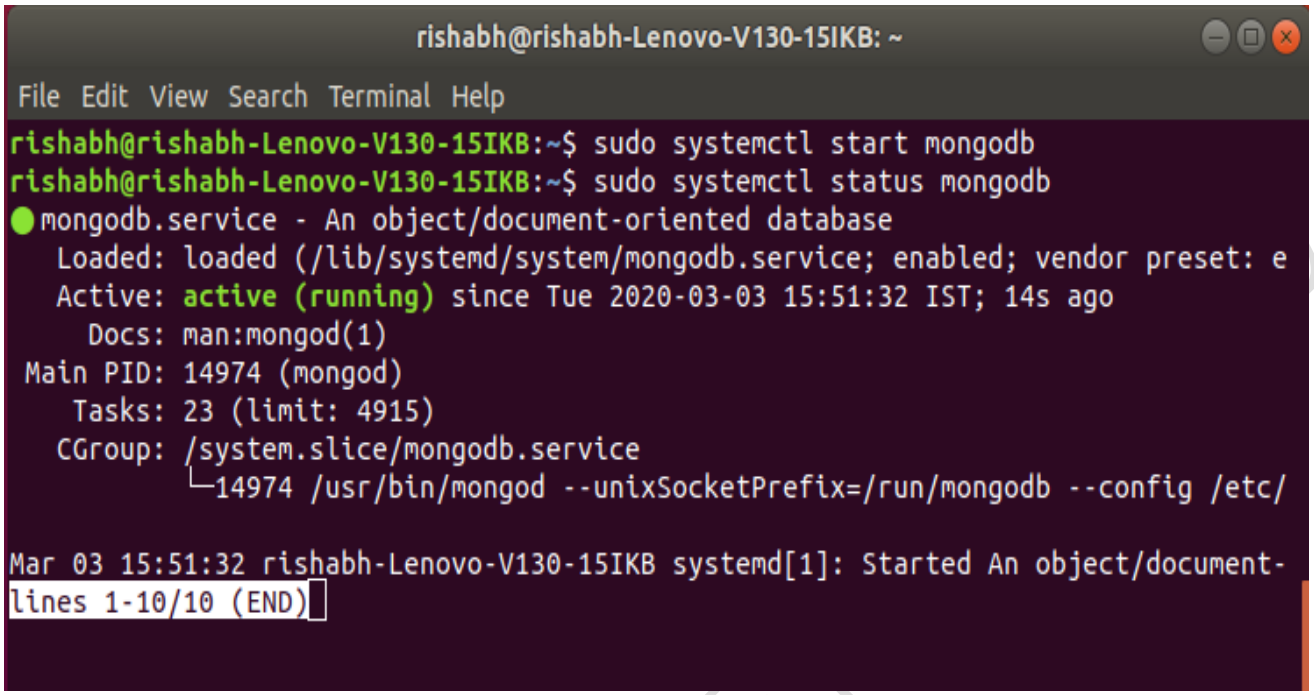
MongoDB service has been stopped and can be checked by using the status command:

sudo systemctl status mongo db



As it can be seen that the service has stopped, to start the service we can use:

sudo systemctl start mongo db



Step 6: Accessing the MongoDB Shell

MongoDB provides a command-line interface called the MongoDB shell, which allows you to interact with the database.
To access the MongoDB shell, simply type the following command in your terminal:

mongo

You are now connected to the MongoDB server, and you can start executing commandsto create databases, collections, and documents.

CRUD Operations:

1. **Create (Insert)**

To create or insert data into a MongoDB collection, you use the insertOne () orinsertMany () methods.

Insert a single document:

```
db.collection ('yourCollection').insertOne({ key: value });
```

Insert multiple documents: db.collection ('yourCollection').insertMany([
{ key1: value1 },
{ key2: value2 }, // more documents
]);

2. Read (Query)

To read or retrieve data from a MongoDB collection, you use the find() method.

Find all documents:

```
db.collection('yourCollection').find();
```

Find documents with a specific condition:

```
db.collection('yourCollection').find({ key: value });
```

3. Update

To update existing documents in a MongoDB collection, you use the updateOne()or updateMany() methods.

Update a single document:

```
db.collection('yourCollection').updateOne({ key: value }, // filter
  { $set: { newField: newValue } } // update operation
);
```

Update multiple documents:

```
db.collection('yourCollection').updateMany(
  { key: value }, // filter
  { $set: { newField: newValue } } // update operation
);
```

4. Delete

To delete documents from a MongoDB collection, you use the deleteOne()or deleteMany()methods.

Delete a single document:

```
db.collection('yourCollection').deleteOne({ key: value });
```

Delete multiple documents:

```
db.collection('yourCollection').deleteMany({ key: value });
```

Viva Questions

1. How do you perform CRUD operations create, read, update, delete MongoDB?

MongoDB CRUD Operations

The Create operation is used to insert new documents in the MongoDB database. The Read operation is used to query a document in the database.

The Update operation is used to modify existing documents in the database. The Delete operation is used to remove documents in the database.

2. What are the CRUD operations in NoSQL database?

CRUD is the acronym for CREATE, READ, UPDATE and DELETE. These terms describe the four essential operations for creating and managing persistent data elements, mainly in relationaland NoSQL databases.

3. How to update in CRUD operations?

You can perform update, insert and delete operation in the Grid. While performing hese operations, the corresponding event is invoked. In that event SQL query is used to update the database. The events for performing CRUD operation are declared.

4. How can we create updating and deleting documents in MongoDB?

The MongoDB shell provides the following methods to update documents in a collection:

1. To update a single document, use db. collection. updateOne() .
2. To update multiple documents, use db. collection. updateMany() .
3. To replace a document, use db. collection. replaceOne() .

5. What is the full form of CRUD in MongoDB?

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

6. What are the CRUD methods in REST API?

CRUD stands for Create, Read, Update, and Delete. These are the four fundamental operations of persistent storage. In the context of RESTful APIs, they correspond to the HTTP methods POST, GET, PUT/PATCH, and DELETE.

7. How to create a collection in MongoDB?

Several ways can be employed to create and remove collections in MongoDB. Of which one way is by using db. Create Collection (name, options). MongoDB creates a collection for an inserted command automatically if no similar collection already exists in the MongoDB database.