SURESH ANGADI EDUCATION FOUNDATION'S
ANGADI INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Savagaon Road, BELAGAVI – 590 009.
Approved by AICTE, New Delhi &Affiliated to Visvesvaraya Technological University, Belagavi,
(Accredited by NAAC)
Department of Artificial Intelligence and Data Science

LABORATORY MANUAL

DATABASE MANAGEMENT SYSTEM

(2022 Scheme)

Subject Code: BCS403A

Prepared By:

Prof. Sagar. Birje
Assistant Professor Dept.AI & DS

Institute Vision

To become a premier institute committed to academic excellence and global competence for the holistic development of students.

Institute Mission

M1: Develop competent human resources, adopt outcome based education (OBE) and implement cognitive assessment of students.

M2: Inculcate the traits of global competencies amongst the students.

M3: Nurture and train our students to have domain knowledge, develop the qualities of global professionals and to have social consciousness for holistic development.

Department Vision

To deliver a quality and responsive education in the field of artificial intelligence and data science emphasizing professional skills to face global challenges in the evolving IT paradigm.

Department Mission

- Leverage multiple pedagogical approaches to impart knowledge on the current and emerging AI technologies.
- Develop an inclusive and holistic ambiance that bolsters problem solving, cognitive abilities and critical thinking.
- Enable students to develop trust worthiness, team spirit, understanding law-of-the-land, social behaviour to be a global stake holder

Program Specific Outcomes (PSOs)

- **PSO 1**: To apply core knowledge of Artificial Intelligence, Machine Learning, Deep Learning, Data Science, Big Data Analytics and Statistical Learning to develop effective solutions for real-world problems.

- **PSO 2**: To demonstrate proficiency in specialized and emerging technologies such as Natural Language Processing, Cloud Computing, Robotic Process Automation, Storage Area Networks and the Internet of Things to meet the stringent and diverse professional challenges.

- **PSO 3**: To imbibe managerial skills, social responsibility, ethical and moral values through courses in Management and Entrepreneurship, Software Engineering Principles, Universal Human Values and Ability Enhancement Programs to meet the industry and societal expectations.

Program Educational Objectives (PEOs)

PEO1 : Build a strong foundation in mathematics, core programming, artificial intelligence, machine learning, and data science to enable graduates to analyze, design, and implement intelligent systems for solving complex real-world problems.

PEO2 : Foster creativity, cognitive and research skills to analyze the requirements and technical specifications of software to articulate novel engineering solutions for an efficient product design.

PEO3 : Prepare graduates for dynamic career opportunities in AI and Data Science by equipping them with interdisciplinary knowledge, adaptability, and practical exposure to tools and techniques required for industry and research.

PEO4 : Instill a strong sense of ethics, professional responsibility, and human values, empowering graduates to contribute positively to society and lead with integrity in their professional domains.

PEO5 : Encourage graduates to pursue higher education, certification program, entrepreneurial ventures, etc. by nurturing a mindset of continuous learning and awareness of global trends and challenges.

Program Outcomes (POs)

PO 1: Engineering Knowledge: Apply the Knowledge of Mathematics, Science, Engineering Fundamentals, and an Engineering specialization to the solution of complex Engineering problems.

PO 2: Problem Analysis: Identify, Formulate, Review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of Mathematics, natural sciences and engineering sciences.

PO 3: Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental conditions.

PO 4: Conduct investigations on complex problems: Use research based knowledge and research methods including design of Experiments, analysis and interpretation of data, and synthesis of Information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate technique, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess society, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s work, as a member and leader in a team, to manage projects and in multidisciplinary environments

PO 12: Lifelong learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

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.

Course Outcomes:

- 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.

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 all modern 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

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 structure or 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 other database 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 SQL database. 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 tasks easily.

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 Student

ADD 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 simple RENAME 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 the table 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 from the 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 in existing schema in object.

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

DML COMMANDS:

- INSERT
- UPDATE
- DELETE

1. INSERT

This command is used to enter the information or values into a row. We can connect one or more records to a single table within a repository using this instruction.

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 where clause 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;
```

3. DELETE

a) Delete some rows

DELETE statement is used to delete rows from a table. Generally DELETE statement 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 space contained 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. There is 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 is used.

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.) this command 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 is to 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 date from 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 update an existing record without adding a value in this column. This means that field has null value.

Where SQL is NULL:

Syntax:

SELECT COLUMN_NAME(S) FROM TABLE_NAME
WHERE 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_NAME
WHERE 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 statement in either ascending or descending order.

a) with one row

Syntax:

SELECT Column_Name FROM Table_Name
ORDER BY Column_Name;

Example:

SELECT * FROM Students ORDER BY Reg_no;

b) With Multiple row

Syntax:

SELECT Column_Name(S) FROM Table_Name
ORDER 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 same values 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 where clause.

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 where clause.

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 and taking 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 cancels access 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. The most 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 transaction can 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 employeetable.
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

SQL>desc employee;

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

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

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

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

4. 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 contain any 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, typically stored 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 Employeetable.
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.

SQL>desc employee;

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

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 meaningful components.

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. An example is "age". A person's age can be derived from date of birth.

8. What is Complex Attributes?

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

9. What is Key Attribute ?

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

10. What is 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),Group by,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 a employee entity.

3. What is Multi valued Attributes?

An attributes 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, Name can 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 or atomic attributes. For example, age of employee entity

6. What is Stored Attribute?

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

7. What is Derived Attribute ?

Attributes derived from other stored attribute. For example age from Date of Birth 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 have more 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,House Number), City, State, Pin))}.Here {} are used to enclose multivalued attributes and () are used to enclose composite attributes with comma separating individual attributes.

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 of the 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:**

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

Table created.

2. **Insert values into the table**

Insert any five records into Customers table

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

insert into Customers values(20,'rohith',30,9000, "Delhi");

insert into Customers values(30,'david',28,9000, "Pune");

insert into Customers values(40,'rahul',29,7000, "Mumbai");

insert into Customers values(50,'pramod',31,8000,"Mysore");
```

ID	Name	Age	Sal	Address
10	Abhi	25	10000	Bangalore
20	Rohith	30	9000	Delhi
30	David	28	9000	Pune
40	Rahul	29	7000	Mumbai
50	Pramod	31	8000	Mysore

3.Creating trigger

```
SQL>set serveroutput on;
 1  create or replace TRIGGER sal_diff
 2  Before Delete or INSERT OR UPDATE on Customer

 3  for each row
 4  when(new.id>0)
 5  Declare
 6     sal_diff number;
 7  BEGIN
 8        sal_diff := :NEW.sal - :OLD.sal;
 9        dbms_output.put_line('Previous salary: ' || : OLD.sal);
10        dbms_output.put_line('Current salary ' || : NEW.sal);
11        dbms_output.put_line ('salary difference: ' || sal_diff);
12  END;
13 /
```

Trigger created.

4.Finding salary difference

SQL>UPDATE CUSTOMERS SET salary =’10000’ where id=’50’;

Previous salary:8000
Current salary: 10000
Salary difference:2000

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 any foreign 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 in order 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 is established 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 entity set.

5. **What is Candidate Key**

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

6. **What is a query?**

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

7. Define SQL Insert Statement ?

SQL INSERT statement is used to add rows to a table.

8. Define SQL Update Statement ?

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

9. 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);

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

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

```
SQL> set serveroutput on;
SQL> declare cursor c1 is select id, sal from Cust;
 2  vid int;
 3  vsal int;
 4  begin
 5  dbms_output.put_line('Emp ID' ||'                    '||'Emp sal');
 6  dbms_output.put_line('------------------------------------ '));
 7    open c1;
 8  loop
 9    fetch c1 into vid,vsal;
10  exit when c1%notfound;
11  dbms_output.put_line(vid || '                    ' || vsal);
12  end loop;
13  close c1;
14  end;
15  /
```

Output:

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

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 contain all the students of a school; likewise 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 the relationship.

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 semantics and constraints.

10. What is E-R model?

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

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 in the 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:

```
SQL>create procedure roll_details AS
 2  rno1 int;
 3  nm1  varchar(20);
 4  rno2 int;
 5  nm2 varchar(20);
 6  done number :=0;
 7  cursor c1 IS select roll,name from O_Rollcall;
 8 cursor c2 IS select roll,name from N_Rollcall
 9  begin
10  open c1;
11  loop
```

```
12  fetch c1 into rno1,nm1;
13  exit when c1%notfound;
14  done :=0;
15  open c2;
16  loop
17  fetch c2 into rno2,nm2;
18  exit when c2%notfound;
19  if rno1=rno2 then
20  exit;
21  end if;
22  end loop;
23  if c2%notfound then
24  insert into N_Rollcall values(rno1,nm1);
25  end if;
26  close c2;
27  end loop;
28  close c1;
29  end;
30  /
```

Procedure created.

SQL>call roll_details();

Call completed.

SQL>select * from N_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

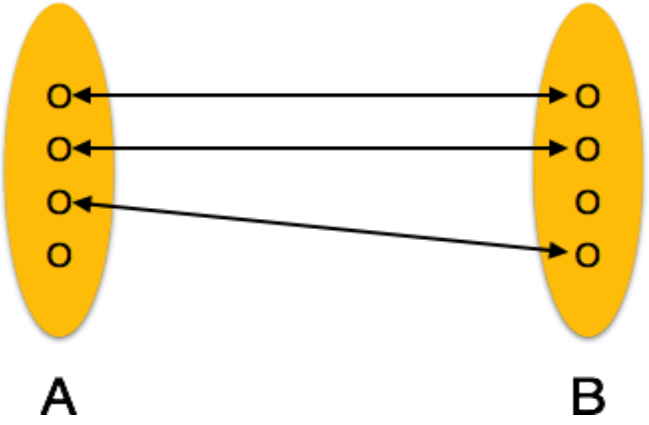
Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

2. What are the different types of Mapping

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

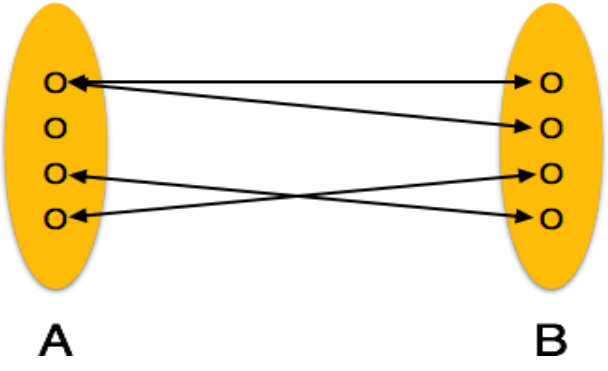
3. What is One-to-one mapping?

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



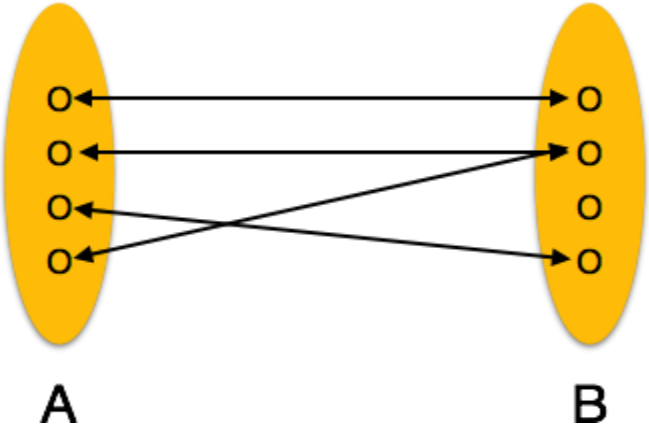
4. What is One-to-many mapping?

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



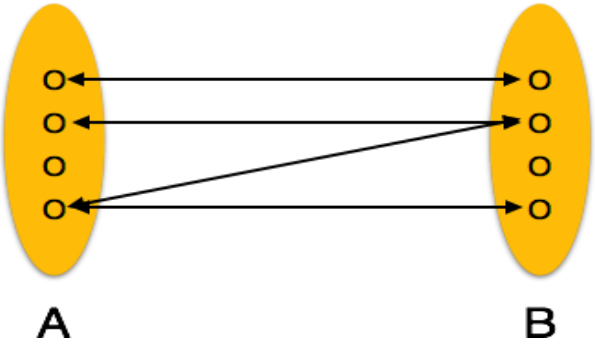
5. What is Many-to-one mapping?

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 vice versa.



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 and REVOKE come under this.

Experiment-7:

Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read,Update & Delete) operations. Execute MangoDB basic Queries 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.

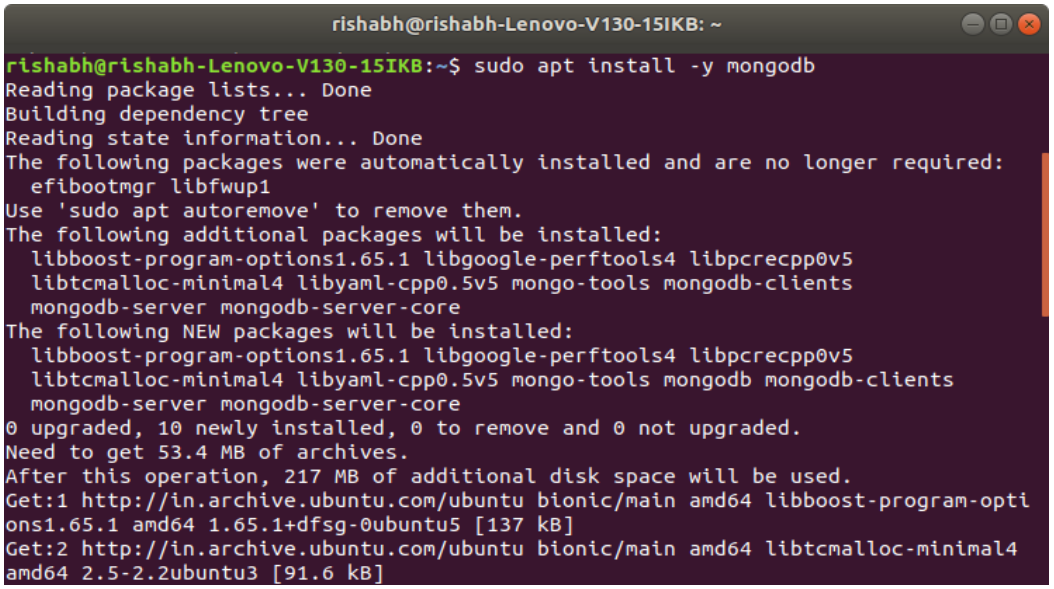
Steps to Install and Configure MongoDB in Ubuntu

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
```

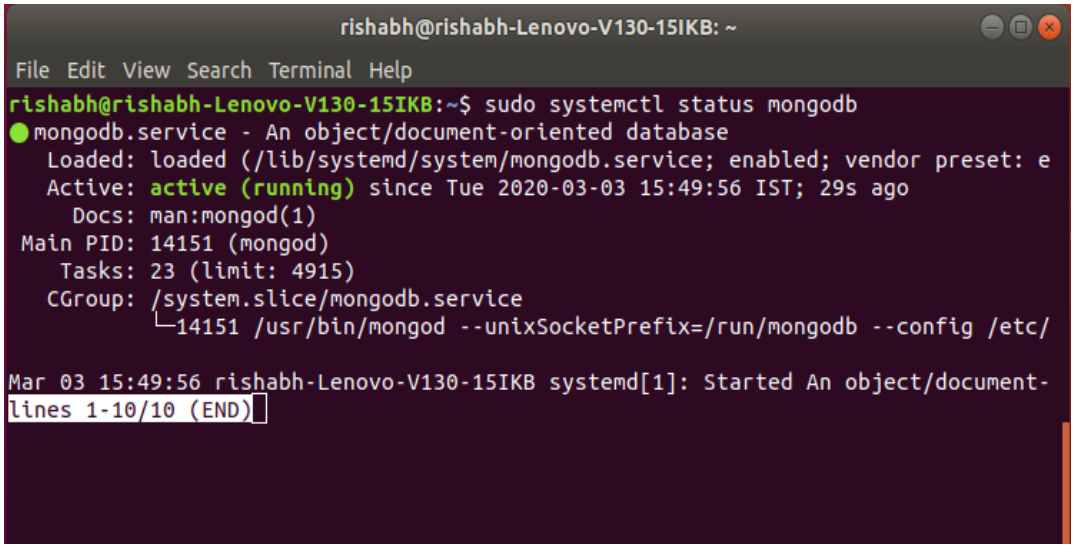
Step 2: Now, install the MongoDB package using 'apt'. Type the following command and press Enter.

```
sudo apt install -y mongodb
```



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

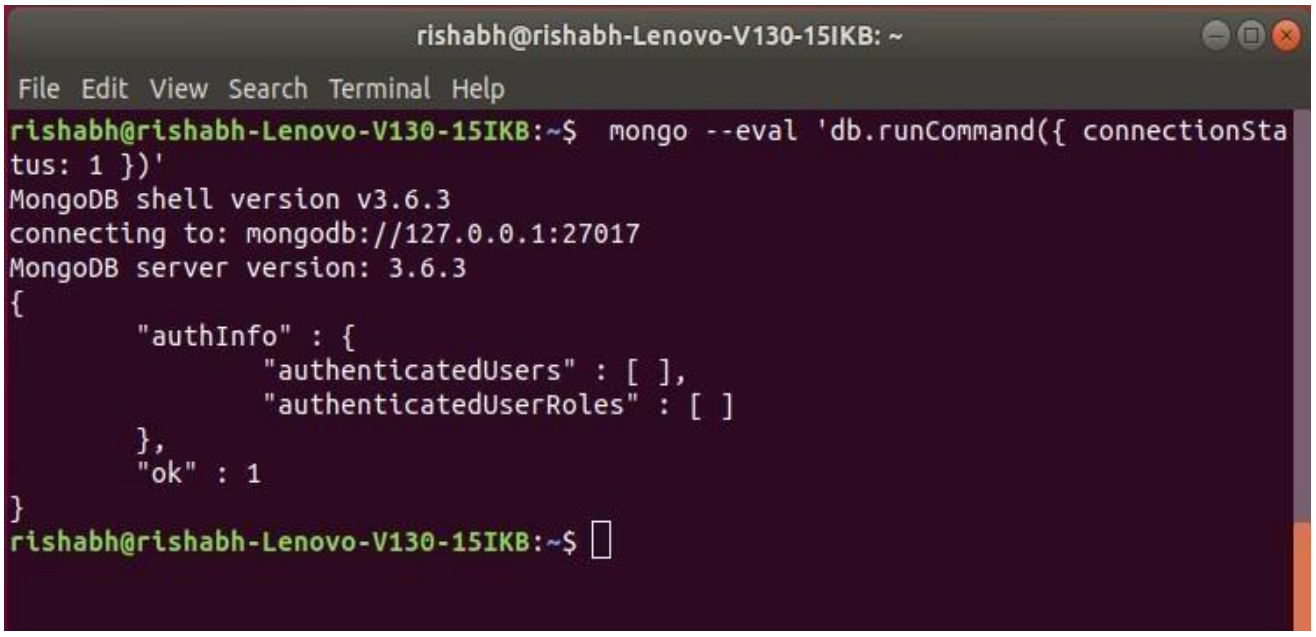
sudo systemctl status mongodb



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. Go through the following command:

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



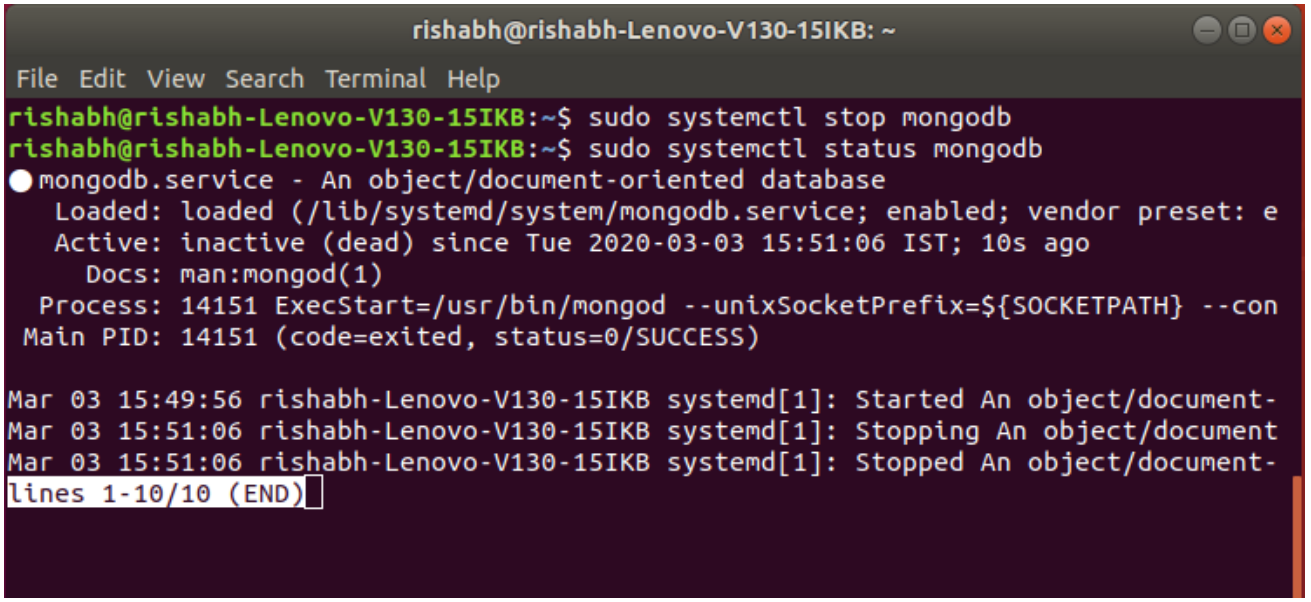
the value "1" in ok field indicates that the server is working properly with no errors.

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

sudo systemctl stop mongodb

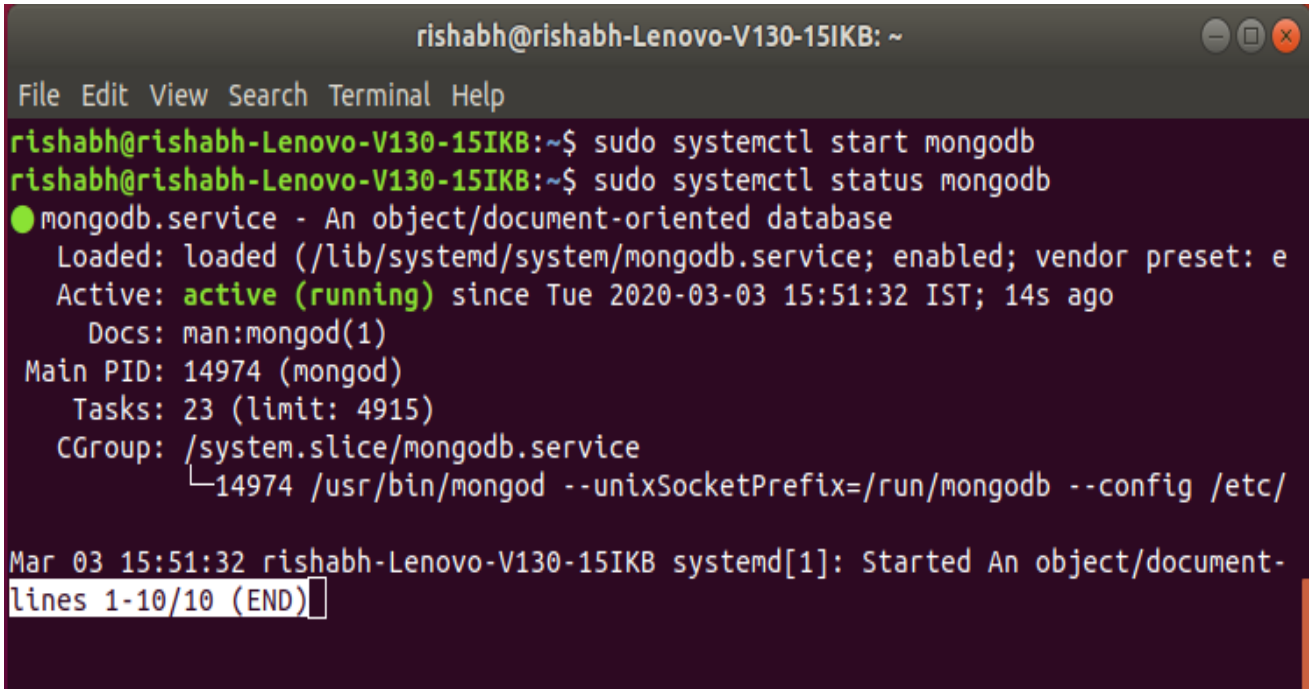
MongoDB service has been stopped and can be checked by using the status command:

sudo systemctl status mongodb



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

sudo systemctl start mongodb



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 commands to createdatabases, collections, and documents.

CRUD Operations:

1. Create (Insert)

To create or insert data into a MongoDB collection, you use the insertOne() or insertMany() 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, deleteMongoDB?

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 persistentdata elements, mainly in relational and NoSQL databases.

3. How to update in CRUD operations?

You can perform update, insert and delete operation in the Grid. While performingthese 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 acollection:

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 CRUDmethods 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 collectionalready exists in the MongoDB database.