GOVT.COLLEGE FOR MENA(A)::KADAPA

Department of Computer Science / Applications

Year: 3rd **Semester: Vth**

DBMS Lab

INDEX

Govt.College for Men(A)::Kadapa

LAB MANUAL

	Course Name:DBMS Lab.	**Experiment No.** 1	
		Group: CS	**Semester:**V

AIM: Write the queries for DataManipulation and Data Definition Language.

Theory:

DML: A **data manipulation language** (**DML**) is a family of syntax elements similar to a computer programming language used for selecting, inserting, deleting and updating data in a database. Performing read-only queries of data is sometimes also considered a component of DML.

Commands in DML are:

- **a.** INSERT
- **b.** UPDATE
- **c.** DELETE
- **d.** SELECT

DML COMMANDS:

SYNTAX:

INSERT Statement:

Single Row into a Table: INSERT INTO table – name [column- identifier-comma-list)] VALUES (column-valuecomma-list);

Multiple Row into a Table: insert into <table name> values (&col1, &col2, ...);

UPDATE Statement: *UPDATE table-name SET update- column-list [WHERE search-condition];*

DELETE Statement: *DELETE FROM table-name [WHERE search- condition];*

DDL: A **data definition language** or **data description language** (**DDL**) is syntax similar to a computer programming language for defining data structures, especially database schemas.-

Commands in DDL are:

- **a.** CREATE
- **b.** DROP
- **c.** TRUNCATE
- **d.** RENAME
- **e.** ALTER

DDL COMMANDS:

SYNTAX:

CREATE Statement: *Create table tablename (column_name1 data_ type constraints, column_name2 data_ type constraints);*

DROP:*DROP TABLE table_name;*

TRUNCATE: *TRUNCATE TABLE table_name;*
RENAME: *RENAME TABLE {tbl_name} TO {new_tbl_name};*
ALTER:
Add column to Table: ALTER TABLE table_name ADD column_name column-definition;
Modify column in Table: ALTER TABLE table_name MODIFY column_namecolumn_type;
Drop column in Table:ALTER TABLE table_name DROP COLUMN column_name;

DDL QUERIES:

Q1. Write a query to create a table employee with empno, ename, designation, andsalary.

```
SQL>CREATE TABLE EMP (EMPNO NUMBER (4),
        ENAME VARCHAR2 (10),
        DESIGNATIN VARCHAR2 (10),
        SALARY NUMBER (8,2));
```

Table created.

Q2. Write a query for create a from an existing table with all the fields.

SQL> CREATE TABLE EMP1 AS SELECT * FROM EMP;

Table created.

SQL> DESC EMP1

Name	Null?	Type
EMPNO	NUMBER	(4)
ENAME	VARCHAR2	(10)
DESIGNATIN	VARCHAR2	(10)
SALARY	NUMBER	(8,2)

Q3. Write a Query to Alter the column EMPNO NUMBER(4) TO EMPNO NUMBER(6).

SQL>ALTER TABLE EMP MODIFY EMPNO NUMBER (6);

Table altered.

Q4. Write a query to add a new column in to employee.

SQL> ALTER TABLE EMP ADD QUALIFICATION VARCHAR2(6);

Table altered.

Q5. Write a query to drop a column from an existing table employee.

SQL> ALTER TABLE EMP DROP COLUMN DOJ;

Table altered.

Q6. Write a query to drop an existing table employee.

SQL> DROP table employee;

Table deleted.

DML QUERIES:

Q1. Write a query to insert the records in to employee.
SQL>INSERT INTO EMP VALUES(103,'Saurabh','ASST_PROF',25000);
1 row created.

Q2. Write a query to display the records from employee.

```
SQL> SELECT * FROM EMP;
 EMPNO ENAME      DESIGNATIN          SALARY
---------- ------------  ----------          ----------
103      SAURABH     ASST_PROF           25000
```

Q3. Write a query to insert the records in to employee using substitution method.

```
SQL> INSERT INTO EMP
VALUES(&EMPNO,'&ENAME','&DESIGNATIN','&SALARY');
Enter value for empno: 102
Enter value for ename: DHAJVEER
Enter value for designatin: ASST_PROF
Enter value for salary: 35000
old 1: INSERT INTO EMP
VALUES(&EMPNO,'&ENAME','&DESIGNATIN','&SALARY')
new 1: INSERT INTO EMP VALUES(102,'DHAJVEER','ASST_PROF','35000')
1 row created.
SQL> /

Enter value for empno: 101
Enter value for ename: ABHILASHA
Enter value for designatin: ASST_PROF
Enter value for salary: 40000
old 1: INSERT INTO EMP
VALUES(&EMPNO,'&ENAME','&DESIGNATIN','&SALARY')
new 1: INSERT INTO EMP VALUES(101,'ABHILASHA','ASST_PROF','40000')
1 row created.
```

Q4. Write a query to update the records from employee.

```
SQL> UPDATE EMP SET SALARY=45000 WHERE EMPNO=101;
1 row updated.
SQL> SELECT * FROM EMP;
EMPNO        ENAME                DESIGNATIN          SALARY
----------        ------------              ----------          ----------
 101     ABHILASHA          ASST_PROF 45000
 102     DHAJVEER           ASST_PROF          35000
 103     SAURABH            ASST_PROF          30000
```

Outcome:

To understand the basic commands of DML and DDL and their use in database.

LAB MANUAL			
	Course Name:DBMS Lab.	**Experiment No.** 2	
		Branch: CS	**Semester:**V

AIM: Write SQL queries using logical operations and operators.

Theory:

An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations. These Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.

- Arithmetic operators
- Comparison operators
- Logical operators
- Operators used to negate conditions

Pre-Requisite Data:

CUSTOMER TABLE

ID	NAME	AGE	ADDRESS	SALARY
1	Akshay	25	Delhi	30000
2	Manish	27	Mumbai	35000
3	Kushagra	26	Kolkata	30000
4	Mukesh	31	Hyderabad	32000
5	Himanshu	29	Chennai	40000
6	Neeraj	30	Noida	36000
7	Nishant	32	Delhi	30000

Queries:

Q1. Write a query to find the salary of a person where age is <= 26 and salary >= 25000 from customer table.

SQL>SELECT * FROM CUSTOMERS WHERE AGE <= 26 AND SALARY >= 25000;

Output:

ID	NAME	AGE	ADDRESS	SALARY
1	Akshay	25	Delhi	30000
3	Kushagra	26	Kolkata	30000

2 rows selected.

Q2. Write a query to find the salary of a person where age is <= 26 or salary > =33000 from customer table.

SQL>SELECT * FROM CUSTOMERS WHERE AGE <= 26 or SALARY > =33000;

Output:

ID	NAME	AGE	ADDRESS	SALARY
1	Akshay	25	Delhi	30000
2	Manish	27	Mumbai	35000
3	Kushagra	26	Kolkata	30000
5	Himanshu	29	Chennai	40000
6	Neeraj	30	Noida	36000

5 rows selected.

Q3.Write a query to find the name of customer whose name is like “Ku%”.

SQL>SELECT * FROM CUSTOMERS WHERE NAME LIKE 'Ku%';

Output:

ID	NAME	AGE	ADDRESS	SALARY
3	Kushagra	26	Kolkata	30000

1 row selected.

Q4. Write a query to find the customer details using “IN” and “Between” operator where age can be 25 or 27.

SQL>SELECT * FROM CUSTOMERS WHERE AGE IN (25, 27);
SQL>SELECT * FROM CUSTOMERS WHERE AGE BETWEEN 25 AND 27;

Output:

ID	NAME	AGE	ADDRESS	SALARY
1	Akshay	25	Delhi	30000
3	Kushagra	26	Kolkata	30000

2 rows selected.

Outcome:

To understand the implementation of SQL queries using logical operations and operators.

	LAB MANUAL		
	Course Name:DBMS Lab.	**Experiment No.** 3	
		Branch: CS	**Semester:**V

AIM: Write SQL query using group by function.

Theory:

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

GROUP BY Syntax:

SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
ORDER BY *column_name(s);*

QUERY:

SELECT COUNT(ID),Address
FROM Customers
GROUP BY address;

Output:

COUNT(ID)	ADDRESS
2	Delhi
1	Mumbai
1	Kolkata
1	Hyderabad
1	Chennai
1	Noida

6 rows selected.

Outcome:

To understand the SQL query using group by function.

	LAB MANUAL		
	Course Name:DBMS Lab.	**Experiment No.** 4	
		Branch: CS	**Semester:**V

AIM: Write SQL queries for group functions.

Theory:

An SQL group function or aggregate functions performs an operation on a group of rows and returns a single result. You may want retrieve group of item-prices and return total-price. This type of scenario is where you would use a group functions. The following table is summary of some SQL group function & query examples.

Function	**Description**	**Query Example**
AVG(fieldname)	Returns average value of a column	SELECT avg(price) FROM inventory;
COUNT(fieldname)	Returns number of items in Table or queried items	SELECT count(product_id) from Product;
MAX (fieldname)	Returns maximum value ofColumn	SELECT max(price)FROM inventory;
MIN(fieldname)	Returns minimum value of Column	SELECT min(price)FROM inventory;
SUM(fieldname)	Returns total value of Column	SELECT sum(price)FROM inventory;

To use a group function in a SQL query, list the function name followed by numericcolumn name within parentheses. AVG averages the column, COUNT counts the numberof items, MAX returns maximum number of the column, and MIN returns minimumnumber of the column .The following is query to retrieve total price, average price, maximum price, and minimum price from the table "product" assuming the product table has the followingvalues.

QUERY:

PRODUCT TABLE

Product ID	Name	Description	Price	colour
100000000	Printer	Inkjet 300 colour Printer	120	80
100000001	Printer	1220XI Inkjet Printer	200	130
100000002	Printer	Photo 890 Inkjet Printer	250	200
100000003	Printer	Photo 890 Inkjet Printer	300	270

Q1. Write a query find the total price of the product.

```
SQL>SELECT sum(price)
FROM product;
```

SUM(PRICE)
870

This statement will returns the total amount for the column price which is 870.

Q2. Write a query find the average price of the product.

```
SQL>SELECT avg(price)
 FROM product;
```

Avg(price)
217.50

This statement will returns the average amount for the column price which is 870/4 or 217.50.

Q3. Write a query find the max price of the product.

```
SELECT max(price)
FROM product;
```

Max(price)
300

This statement will returns the maximum amount for the column price which is 300.

Outcome:

To understand the implementation of SQL queries for group functions.

LAB MANUAL			
	Course Name:DBMS Lab.	**Experiment No.** 5	
		Branch: CS	**Semester:**V

AIM: Write SQL queries for sub queries, nested queries.

Theory:

Nested Queries: Nesting of queries one within another is known as a nestedqueries.
Sub queries. The query within another is known as a sub query. A statementcontaining sub query is called parent statement. The rows returned by sub query areused by the parent statement.

Types

1. Sub queries that return several valuesSub queries can also return more than one value. Such results should be made usealong with the operators in and any.

2. Multiple queries

Here more than one sub query is used. These multiple sub queries are combined bymeans of 'and' & 'or' keywords

3. Correlated sub query

A sub query is evaluated once for the entire parent statement whereas a correlatedSub query is evaluated once per row processed by the parent statement.

Relating Data through Join Concept

The purpose of a join concept is to combine data spread across tables. A join isactually performed by the 'where' clause which combines specified rows of tables.Syntax; select columns from table1, table2 where logical expression;

Types of Joins 1.Simple Join 2.Self Join 3. Outer Join 4. Inner Join

1. Simple Join

a) Equi-join: A join, which is based on equalities, is called equi-join.

b) Non Equi-join: It specifies the relationship between

Table Aliases

Table aliases are used to make multiple table queries shorted and more readable. Wegive an alias name to the table in the 'from' clause and use it instead of the namethroughout the query.

Self join: Joining of a table to itself is known as self-join. It joins one row in a tableto another. It can compare each row of the table to itself and also with other rows ofthe same table.

Outer Join: It extends the result of a simple join. An outer join returns all the rowsreturned by simple join as well as those rows from one table that do not match anyrow from the table. The symbol (+) represents outer joins.

Inner join: Inner join returns the matching rows from the tables that are beingjoined

Queries:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

Q1. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'A'.

SQL>select ename,sal from emp where sal>(select min(sal) from emp where job like 'A%');

Output:

ENAME	SALARY
Arjun	32000
Gugan	40000
Karthik	35000

3 rows selected.

Outcome:

To understand the SQL queries for sub queries, nested queries.

	Course Name:DBMS Lab.	**Experiment No.** 6	
		Branch: CS	**Semester:**V

AIM: Write programme by the use of PL/SQL.

Theory:

The PL/SQL programming language was developed by Oracle Corporation in the late 1980s as procedural extension language for SQL and the Oracle relational database. PL/SQL has the following features −

- PL/SQL is tightly integrated with SQL.
- It offers extensive error checking.
- It offers numerous data types.
- It offers a variety of programming structures.
- It supports structured programming through functions and procedures.
- It supports object-oriented programming.
- It supports the development of web applications and server pages.

Query:

```
DECLARE
a number (2) := 21;
b number (2) := 10;
BEGIN
IF (a = b) then
dbms_output.put_line('Line 1 - a is equal to b');
ELSE
dbms_output.put_line('Line 1 - a is not equal to b');
END IF;
IF (a < b) then
dbms_output.put_line('Line 2 - a is less than b');
ELSE
dbms_output.put_line('Line 2 - a is not less than b');
END IF;
IF ( a> b ) THEN
dbms_output.put_line('Line 3 - a is greater than b');
```

```
ELSE
dbms_output.put_line('Line 3 - a is not greater than b');
END IF;
END;
/
```

Output:
Line 1 - a is not equal to b
Line 2 - a is not less than b
Line 3 - a is greater than b

Outcome:
To learn the programming using PL/SQL.

	LAB MANUAL		
	Course Name:DBMS Lab.	**Experiment No.** 7	
		Branch: CS	**Semester:**V

AIM:Write SQL queries to create views.

Theory:

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Views, which are a type of virtual tables allow users to do the following −

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

Syntax:

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

Query:

Q1. Write a SQL query to create a view of customer table created in PRACTICAL no 1.

```
SQL>CREATE VIEW CUST as
Select ID, Name, Address
From Customer;
```

Output:

ID	NAME	ADDRESS
1	Akshay	Delhi
2	Manish	Mumbai
3	Kushagra	Kolkata
4	Mukesh	Hyderabad
5	Himanshu	Chennai
6	Neeraj	Noida
7	Nishant	Delhi

Outcome:

To understand the implementation of SQL queries to create views

LAB MANUAL			
	Course Name:DBMS Lab.	**Experiment No.** 8	
		Branch: CS	**Semester:**V

AIM: Write an SQL query to implement JOINS.

Theory:

A SQL join clause combines columns from one or more tables in a relational database. It creates a set that can be saved as a table or used as it is. A JOIN is a means for combining columns from one (self-table) or more tables by using values common to each. ANSI-standard SQL specifies five types of JOIN: INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER and CROSS. As a special case, a table (base table, view, or joined table) can JOIN to itself in a self-join.

A programmer declares a JOIN statement to identify rows for joining. If the evaluated predicate is true, the combined row is then produced in the expected format, a row set or a temporary table.

QUERIES:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

DEPARTMENT TABLE

DEPTNO	DNAME	LOCATION
1	ACCOUNTING	NEW YORK
2	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Q1. Display the employee details, departments that the departments are same in both the emp and dept.

SQL>select * from emp,dept where emp.deptno=dept.deptno;

EMPNO	ENAME	JOB	DEPTNO	SALARY	DEPTNO	DNAME	LOCATION
1	Mathi	AP	1	30000	1	ACCOUNTING	NEW YORK
2	Arjun	ASP	2	32000	2	RESEARCH	DALLAS
3	Gugan	ASP	2	40000	2	RESEARCH	DALLAS
4	Karthik	AP	1	35000	1	ACCOUNTING	NEW YORK

Outcome:

To understand the implementation of JOINS using SQL.

LAB MANUAL			
	Course Name:DBMS Lab.	**Experiment No.** 9	
		Branch: CS	**Semester:**V

AIM: Write a queryfor extracting data from more than one table.

Query:

EMPLOYEE TABLE

EMPNO	ENAME	JOB	DEPTNO	SALARY
1	Mathi	AP	1	30000
2	Arjun	ASP	2	32000
3	Gugan	ASP	2	40000
4	Karthik	AP	1	35000

DEPARTMENT TABLE

DEPTNO	DNAME	LOCATION
1	ACCOUNTING	NEW YORK
2	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Q1. Write a query to extract empno, ename, salary, dname and location from employee and department table where empno = deptno without using joins.

SQL> select employee.empno, employee.ename, employee.salary, department.dname, department.location
From department, employee
Where department.deptno = employee.empno;

Output:

EMPNO	ENAME	SALARY	DNAME	LOCATION
1	Mathi	30000	ACCOUNTING	NEW YORK
2	Arjun	32000	RESEARCH	DALLAS

2 rows selected.

Q2. Write a query to extract ename, salary and location from employee and department table where is like 30, 40.

SQL> selectemployee.ename, employee.salary, department.location
From department, employee
Where department.deptnoIN (30,40);

Output:
No rows Selected.

Outcome:

To understand the queryfor extracting data from more than one table.

	LAB MANUAL		
	Course Name:DBMS Lab.	**Experiment No.** 10	
		Branch: CS	**Semester:**V

AIM: Write a query to understand the concepts for ROLL BACK, COMMIT & CHECK POINTS.

Theory:

Transaction Control Language(TCL) commands are used to manage transactions in database.These are used to manage the changes made by DML statements. It also allows statements to be grouped together into logical transactions.

Commit command

Commit command is used to permanently save any transaaction into database.

Following is Commit command's syntax,

COMMIT;

Rollback command

This command restores the database to last commited state. It is also use with savepoint command to jump to a savepoint in a transaction.

Following is Rollback command's syntax,

rollback to savepoint-name;

Savepoint command

Savepoint command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.

Following is savepoint command's syntax,

savepointsavepoint-name;

QUERY:

Q1. Write a query to implement the save point.

SQL> select employee.empno, employee.ename, employee.salary, department.dname, department.location
From department, employee
Where department.deptno = employee.empno;

SQL> SAVEPOINT S1;
Savepoint created.

Q2. Write a query to implement the Rollback.

SQL>ROLL BACK S1;

Rollback complete.

Outcome: To understand the concept of rollback, save-points and commit statements.