DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LAB MANUAL

Name: Database Management System LAB

Course Code: 4CS4-22

Course: B.TECH

LIST OF EXPERIMENTS

Sr. No.	Particulars	Date of Experiment	Faculty Signature
1	To study about the DDL commands.		
2	To study about the DML commands.		
3	To study about Aggregate Functions.		
4	To study about Set Operations.		
5	To study about Joins operations.		
6	To study about Views operations.		

Experiment - One

(a) Objective: To study about the DDL commands in DBMS.

(b) Coding & Result :

Table Creation:

Syntax: **Create table** table-name(A_1D_1,A_2D_2………AnDn)

Description: To create a new table.

Table Description:

Syntax: **Desc** table-name.

Description: To display the structure of the table.

Table Deletion:

Syntax: **Drop table** table-name

Description: To delete the table.

Table Renaming:

Syntax: **Alter table** old-table-name **rename to** new-table-name

Description: To rename the table .

Rename Column:

Syntax: **Alter table** table-name **rename column** old-column-name to New-column-name

Description: To rename the column in the table.

Adding one Column:

Syntax: **Alter table** table-name **add** A D

Description: To alter the table by adding one column

Deleting One Column:

Syntax: **Alter table** table-name **drop column** A

Description: To delete one column from the table.

Datatype or Domaintype modification:

Syntax:

Alter table table-name **modify**(A_1 new data-type(size),……..An new data-type(size))

Description:

To change the datatype or domaintype.

Table Truncation:

Syntax: **Truncate table** table-name

Description: To delete all the records from the table.

Example:-

Creating the table:

Create table student (Regno int,Name char(20),Course char(9),Year int,Remarks char(10))

Table created.

Description of the table:

Desc student

Output-

Renaming the column:

Alter table student rename column course to degree

Table altered.

Desc student;

Name	Null?	Type

Adding one column:

Alter table student add Age int

Table altered.

Desc student

Name	Null?	Type

Deleting one column:

Alter table student drop column age

Table altered.

Desc student

Name	Null?	Type

Domain type modification:

Alter table student **modify**(Year varchar(10))

Table altered.

Desc student

Name	Null?	Type

Table renaming:

Alter table student rename to class

Table altered

Desc class

Name	Null?	Type

Table truncation:

Truncate table class
Table truncated.

Create table with primary keys:

```
Create table employ(Rno int primary key,Name varchar(25),Dept char(18),Salary int)
```

Table created.

```
desc employ
```

Name	Null?	Type

Table deletion:

Drop table class

Table dropped.

Result:

Thus the DDL commands were executed.

Experiment - Two

(a) Objective: To study about the DML commands.

(b) Coding & Result :

1.Select:

Syntax:

Select A_1,A_2,…….An

From r_1,r_2,r_3…….r_n

Where **p**

Description: It selects the rows from the table.

2.Insert:

Syntax:

Insert into table-name **values**(value1,value2,…….valuen)

Description: It is used to insert values into specified table.

3.Update:

Syntax 1:

Update tablename

Set A = value **Where** P

Description: It is used to update a particular value in the table.

Syntax 2:

Update tablename

Set A = **case**

when P1 **then** result1

```
when P2 then result2
when Pn then result n
else return 0
end                  .
```

Description:

It is used to update a set of values in the table by checking certain conditions.

4.Delete:

Syntax:

```
Delete from table-name
Where P
```

Description: It is used to delete the values from specified table.

5.Group by:

Syntax:

```
Select    <(columns)>
From      <(tables)>
Where     <(condition)>
Group by <groupcolumns>
```

Description: It is used to group the rows that have certain properties and then to apply an aggregate function on one column for each group separately.

6.OrderBy:

Syntax:

```
Select [distinct]<columns>
From <table>
```

Where <condition>

[**order by** <columns[asc/desc]>]

Description: It is used to retrieve the information stored in database in a particular order.

Output:

1. Select loannumber,amount*1000

   From loan

LOANNUMBER	AMOUNT*1000

2. Select loannumber

   from loan

   where amount between 900 and 1000

LOANNUMBER

3. Select customername,loan.loannumber,amount

```
From loan,Borrower
Where loan.loannumber=Borrower.loannumber and

Branchname = 'Perryridge'
```

CUSTOMERNAME	LOANNUMBER	AMOUNT

4.

```
select Customername,loannumber

from Borrower

order by Customername
```

CUSTOMERNAME	LOANNUMBER

5.

```
Update account Set balance = balance+50  Where branchname='Brighton'

2 rows updated.

 select * from account
```

ACCOUNTNUMBER	BRANCHNAME	BALANCE

6. Delete from loan

   Where branchname='Roundhill'
   1 row deleted.

   Select * from loan

LOANNUMBER	BRANCHNAME	AMOUNT

7. select customername

   from customer

   where customername like '_a%'

CUSTOMERNAME

Experiment - Three

(a) Objective: To study about the Aggregate Functions.

(b) Coding & Result:

Count()-To count the number of rows in a table.

Sum()-To find the total value from the row..

Avg()-To find the average value from the row.

Min()-To find the minimum value from the row.

Max()-To find the maximum value from the row.

Syntax

Select function-name **(A)**

from table-name

select * from accrelation;

select * from borrower;

select * from branrelation;

select * from custrelation;

select * from loan;

select * from depositer;

ACCNUMBER	BRANCHNAME	BALANCE

CNAME	LNUMBER

BRANCHNAME	BRANCHCITY	ASSETS

CUSTOMERNAME	CUSTOMERSTREET	CUSTOMERCITY

LNUMBER	BNAME	AMOUNT

CNAME	ACCNUMBER

Select avg(balance)from accrelation where branchname='perryridge'

AVG(BALANCE)

Select branchname,avg(balance)from accrelation group by branchname

BRANCHNAME	AVG(BALANCE)

Select count(*) from custrelation

COUNT(*)

Select count(distinct branchname)from branrelation

COUNT(DISTINCTBRANCHNAME)

Select branchname,avg(balance)from accrelation group by branchname having avg(balance)>700;

BRANCHNAME	AVG(BALANCE)

Select max(balance)from accrelation

MAX(BALANCE)

Select max(balance)-min(balance) from accrelation

MAX(BALANCE)-MIN(BALANCE)

Select min(amount)from loan

MIN(AMOUNT)

Select sum(amount) from loan

SUM(AMOUNT)

Experiment - four

(a) Objective: To study about the Set Operations.

(b) Coding & Result:

Union-Displays all the values from both the table.

Intersection- Displays common values from both the table.

Minus- Displays the values from the left side table.

Select lnumber from loan Intersect (Select lnumber from borrower)

LNUMBER

Select cname from depositer minus (Select cname from borrower)

CNAME

Select cname from depositer union (Select cname from borrower)

CNAME

Set Membership Operations

In- Displays all the values from both the table.

Not in -Displays the values from the left side table.

Select customername from custrelation where customername in('smith','hayes')

CUSTOMERNAME

Select cname from depositer where cname in(select cname from borrower)

CNAME

Select cname from depositer where cname not in(select cname from borrower)

CNAME

Experiment - FIVE

(a) Objective: To perform JOIN operations.

(b) Coding & Result :

INNER JOIN:

DESCRIPTION: Returns the matching rows from the tables that are being joined.

OUTER JOIN:

SELECT *

FROM<tablename1> join type <tablename2>

ON tablename1.aattribute=tablename2.attribute

LEFT OUTER JOIN: Returns the matching row from the table that are being joined and non matching rows from the left table

RIGHT OUTER JOIN: Returns the matching row from the table that are being joined and non matching rows from the right table

FULL OUTER JOIN: Returns the matching row from the table that are being joined and non matching rows from the left and right table

INITIAL TABLES:

1)Select * from loan

LOANNO	BNAME	AMOUNT

2)select * from borrow

CNAME	LOANNO

INNER JOIN:

OUTPUT: select * from loan inner join borrow on loan.loanno=borrow.loanno

LOANNO	BNAME	AMOUNT	CNAME	LOANNO

LEFT OUTER JOIN:

OUTPUT: select * from loan left outer join borrow on loan.loanno=borrow.loanno

LOANNO	BNAME	AMOUNT	CNAME	LOANNO

RIGHT OUTER JOIN:

OUTPUT: select * from loan right outer join borrow on loan.loanno=borrow.loanno

LOANNO	BNAME	AMOUNT	CNAME	LOANNO

FULL OUTER JOIN:

OUTPUT: select * from loan full outer join borrow on loan.loanno=borrow.loanno

LOANNO	BNAME	AMOUNT	CNAME	LOANNO

RIGHT OUTER JOIN WITH SELECTED ATTRIBUTES:

OUTPUT:

select borrow.loanno,cname,amount,bname from loan right outer join borrow on loan.loanno=borrow.loanno

LOANNO	CNAME	AMOUNT	BNAME

LEFT OUTER JOIN WITH SELECTED ATTRIBUTES:

OUTPUT:

```
select loan.loanno,bname,cname,amount from loan left outer join borrow on
loan.loanno=borrow.loanno
```

LOANNO	BNAME	CNAME	AMOUNT

Experiment – SIX

(a) Objective: To study View operations.

(b) Coding & Result :

SYNTAX:

CREATE VIEW viewname as

SELECT columnname,column name

FROM tablename

WHERE columnname=expressionlist;

OUTPUT:

select*from account;

ACCNO	BNAME	BALANCE

OUTPUT:

select*from depositor;

CNAME	ACCNO

OUTPUT:

Create a view consisting of branch name and name of customer

create view a as(select cname,bname from account,depositor where account.accno=depositor.accno);

view created

OUTPUT:

select *from a;

CNAME	BNAME

OUTPUT:

Selcect customer name of branch ‘perryridge’ from the view

QUERY :

select cname from a where bname='perryridige’

CNAME

Result:

Thus the join and view operations were performed.

Program1

Table creation

Create table employee(ssn varchar(4),salary int,superssn varchar(4));

Output:

Value insertion

insert into employee values('s1',10000,'s3');

insert into employee values('s2',15000,'s3');

insert into employee values('s3',12000,");

insert into employee values('s4',16000,'s5');

insert into employee values('s5',13000,");

insert into employee values('s6',11000,'s3');

Display data stored in the table.

Select * from employee;

Output:

SSN	SALARY	SUPE

DBMS Basic VIVA Questions

1. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

2. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

3. What is a Database system?

The database and DBMS software together is called as Database system.

4. What are the advantages of DBMS?

1. Redundancy is controlled.
2. Unauthorised access is restricted.
3. Providing multiple user interfaces.
4. Enforcing integrity constraints.
5. Providing backup and recovery.

5. What are the disadvantage in File Processing System?

1. Data redundancy and inconsistency.
2. Difficult in accessing data.
3. Data isolation.
4. Data integrity.
5. Concurrent access is not possible.
6. Security Problems.

6. Describe the three levels of data abstraction?

The are three levels of abstraction:

1. **Physical level:** The lowest level of abstraction describes how data are stored.
2. **Logical level:** The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
3. **View level:** The highest level of abstraction describes only part of entire database.

7. Define the "integrity rules"?

There are two Integrity rules.

1. **Entity Integrity:** States that "Primary key cannot have NULL value"
2. **Referential Integrity:** States that "Foreign Key can be either a NULL value or should be Primary Key value of other relation.

8. What is extension and intension?

1. **Extension:** It is the number of tuples present in a table at any instance. This is time dependent.
2. **Intension:** It is a constant value that gives the name, structure of table and the constraints laid on it.

9. What is Data Independence?

Data independence means that "the application is independent of the storage structure and access strategy of data". In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.
Two types of Data Independence:

1. **Physical Data Independence:** Modification in physical level should not affect the logical level.
2. **Logical Data Independence:** Modification in logical level should affect the view level.

10. What is a view?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary.

11. What is Data Model?

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

12. 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.

13. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables with in the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

14. What is an Entity?

It is a 'thing' in the real world with an independent existence.

15. What is an Entity type?

It is a collection (set) of entities that have same attributes.

16. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

17. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organised by appropriate data model.

1. **Procedural DML or Low level:** DML requires a user to specify what data are needed and how to get those data.
2. **Non-Procedural DML or High level:** DML requires a user to specify what data are needed without specifying how to get those data.

18. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

19. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

20. What is DDL Interpreter?

It interprets DDL statements and record them in tables containing metadata.

21. What are the different phases of transaction?

Different phases are
1.) Analysis phase,
2.) Redo Phase,
3.) Undo phase.

22. What are the unary operations in Relational Algebra?

PROJECTION and SELECTION.

23. Are the resulting relations of PRODUCT and JOIN operation the same?

No.
PRODUCT: Concatenation of every row in one relation with every row in another.
JOIN: Concatenation of rows from one relation and related rows from another.

24. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

25. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

26. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency X Y is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

27. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

28. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD X A either of the following is true

1. X is a Super-key of R.
2. A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

29. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD X A, X must be a candidate key.

30. What is Relational Algebra?

It is procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

31. What is Relational Calculus?

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. E.g. of languages based on it are DSL ALPHA, QUEL.

32. How does Tuple-oriented relational calculus differ from domain-oriented relational calculus?

1. The **tuple-oriented calculus** uses a tuple variables i.e., variable whose only permitted values are tuples of that relation. E.g. QUEL
2. The **domain-oriented calculus** has domain variables i.e., variables that range over the underlying domains instead of over relation. E.g. ILL, DEDUCE.

33. What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties
(1).Minimizing redundancy, (2). Minimizing insertion, deletion and update anomalies.

34. What is Functional Dependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.