Database Management Systems (3130703)

Lab Manual

B.E. Information Technology, 3rd Semester
(2023– 2024)

L. D. COLLEGE OF ENGINEERING
INFORMATIONTECHNOLOGY DEPARTMENT
AHMEDABAD

Practical: 1

Aim: To study DDL-create and DML-insert commands.

(1) Create tables according to the following definition.
(2) CREATE DATABASE bank;
(3) USE bank;

1. **Table** – **DEPOSITE**

```
CREATE TABLE deposit(

actno VARCHAR(5),

cname VARCHAR(18),

bname VARCHAR(18),

amount DECIMAL(8,2),

adate DATE);
```

COLUMN NAME	DATA TYPE
ACTNO	VARCHAR(5)
CNAME	VARCHAR(18)
BNAME	VARCHAR(18)
AMOUNT	DECIMAL(8,2)
ADATE	DATE

2. **Table** – **BRANCH**

```
CREATE TABLE branch(

bname VARCHAR(18),

city VARCHAR(18));
```

COLUMN NAME	DATA TYPE
BNAME	VARCHAR(18)
CITY	VARCHAR(18)

3. **Table** – **CUSTOMERS**

```
CREATE TABLE customer(
cname VARCHAR(19),
city VARCHAR(18));
```

COLUMN NAME	DATA TYPE
CNAME	VARCHAR(19)
CITY	VARCHAR(18)

4. Table – BORROW

```
CREATE TABLE borrow(
loanno VARCHAR(5),
cname VARCHAR(18),
bname VARCHAR(18),
amount DECIMAL(8,2));
```

COLUMN NAME	DATA TYPE
LOANNO	VARCHAR(5)
CNAME	VARCHAR(18)
BNAME	VARCHAR(18)
AMOUNT	DECIMAL(8,2)

(4) Insert the data as shown below.

Table – DEPOSIT

```
INSERT INTO deposit
(actno,cname,bname,amount,adate)
VALUES
("100","anil","vrce",1000.00,"1995-03-01"),
("101","sunil","ajni",5000.00,"1996-01-04"),
("102","mehul","karolbagh",3500.00,"1995-09-17"),
("104","madhuri","chandi",1200.00,"1995-12-17"),
("105","pramod","m.g.road",3000.00,"1996-03-27"),
("106","sandip","andheri",2000.00,"1996-03-31"),
("107","shivani","virar",1000.00,"1995-09-05"),
("108","kranti","nehru place",5000.00,"1995-04-02"),
("109","minu","powai",7000.00,"1995-08-10");
```

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000.00	1-MAR-95
101	SUNIL	AJNI	5000.00	4-JAN-96
102	MEHUL	KAROLBAGH	3500.00	17-NOV-95
104	MADHURI	CHANDI	1200.00	17-DEC-95
105	PRMOD	M.G.ROAD	3000.00	27-MAR-96
106	SANDIP	ANDHERI	2000.00	31-MAR-96
107	SHIVANI	VIRAR	1000.00	5-SEP-95
108	KRANTI	NEHRU PLACE	5000.00	2-JUL-95
109	MINU	POWAI	7000.00	10-AUG-95

Table – BRANCH

```
INSERT INTO branch
(bname,city)
VALUES
("vrce","nagpur"),
("ajni","nagpur"),
("karolbagh","delhi"),
("chandi","delhi"),
("dharampeth","nagpur"),
("m.g.road","banglore"),
("andheri","bombay"),
("virar","bombay"),
("nehru place","delhi"),
("powai","bombay");
```

BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

Table – CUSTOMERS

```
INSERT INTO customer
(cname,city)
VALUES
("anil","calcuta"),
("sunil","delhi"),
("mehul","baroda"),
("mandar","patna"),
("madhuri","nagpur"),
("pramod","nagpur"),
("sandip","surat"),
("shivani","bombay"),
("kranti","bombay"),
("naren","bombay");
```

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

➢

Table – BORROW

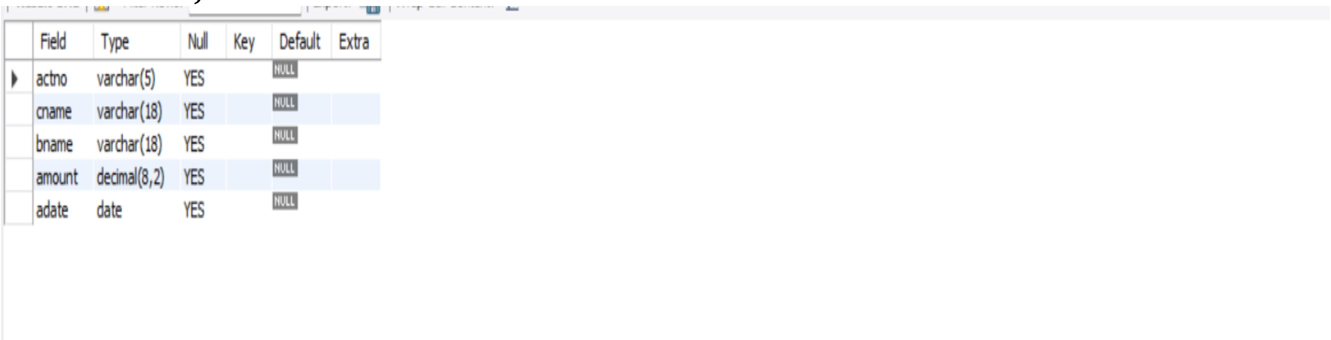
```
INSERT INTO borrow
(loanno,cname,bname,amount)
VALUES
("201","anil","vrce",1000.00),
("206","mehul","ajni",5000.00),
("311","sunil","dharampeth",3000.00),
("321","madhuri","andheri",2000.00),
("375","pramod","virar",8000.00),
("481","kranti","nehru place",3000.00);
```

LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

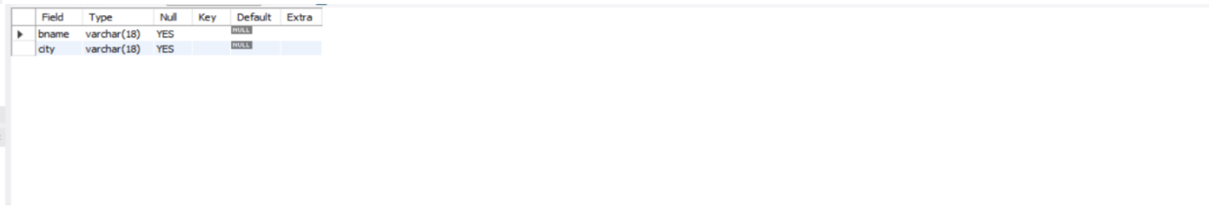
From the above given tables perform the following queries:

(1) Describe deposit, branch.

```
DESC  deposit;
DESC branch;
```



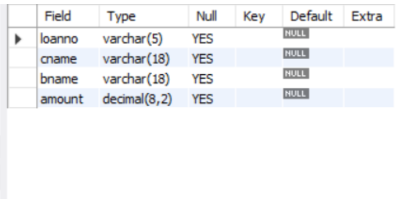
Field	Type	Null	Key	Default	Extra
actno	varchar(5)	YES		NULL	
cname	varchar(18)	YES		NULL	
bname	varchar(18)	YES		NULL	
amount	decimal(8,2)	YES		NULL	
adate	date	YES		NULL	



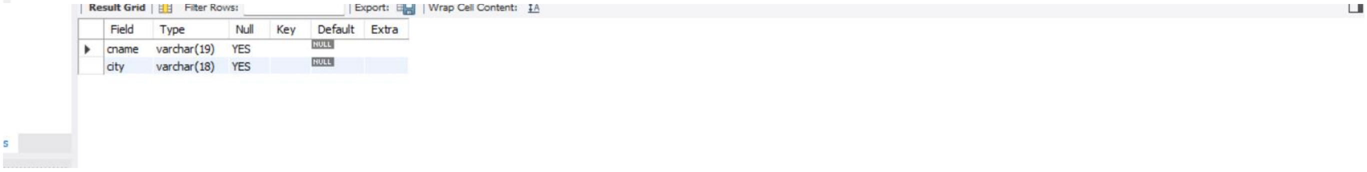
Field	Type	Null	Key	Default	Extra
bname	varchar(18)	YES		NULL	
city	varchar(18)	YES		NULL	

(2) Describe borrow, customers.

```
DESC borrow;
DESC customer;
```



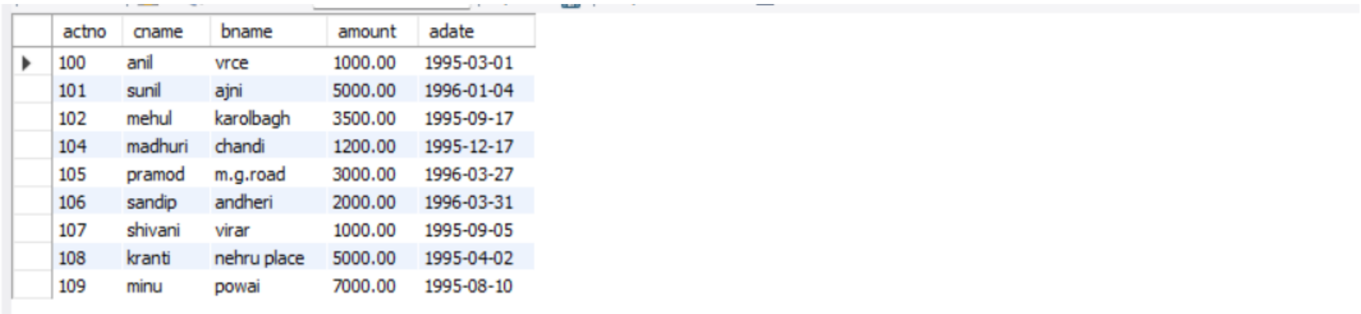
Field	Type	Null	Key	Default	Extra
loanno	varchar(5)	YES		NULL	
cname	varchar(18)	YES		NULL	
bname	varchar(18)	YES		NULL	
amount	decimal(8,2)	YES		NULL	



Field	Type	Null	Key	Default	Extra
cname	varchar(19)	YES		NULL	
city	varchar(18)	YES		NULL	

(3) List all data from table DEPOSIT.

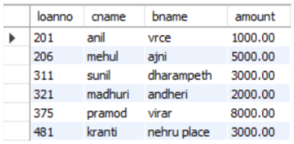
SELECT*FROM deposit;



actno	cname	bname	amount	adate
100	anil	vrce	1000.00	1995-03-01
101	sunil	ajni	5000.00	1996-01-04
102	mehul	karolbagh	3500.00	1995-09-17
104	madhuri	chandi	1200.00	1995-12-17
105	pramod	m.g.road	3000.00	1996-03-27
106	sandip	andheri	2000.00	1996-03-31
107	shivani	virar	1000.00	1995-09-05
108	kranti	nehru place	5000.00	1995-04-02
109	minu	powai	7000.00	1995-08-10

(4) List all data from table BORROW.

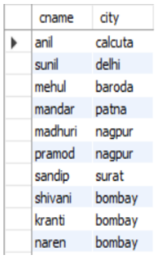
SELECT*FROM borrow;



loanno	cname	bname	amount
201	anil	vrce	1000.00
206	mehul	ajni	5000.00
311	sunil	dharampeth	3000.00
321	madhuri	andheri	2000.00
375	pramod	virar	8000.00
481	kranti	nehru place	3000.00

(5) List all data from table CUSTOMERS.

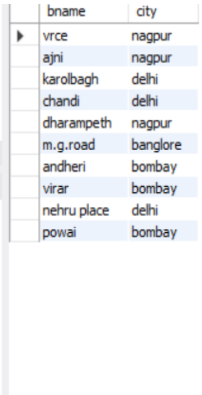
SELECT*FROM customer;



cname	city
anil	calcuta
sunil	delhi
mehul	baroda
mandar	patna
madhuri	nagpur
pramod	nagpur
sandip	surat
shivani	bombay
kranti	bombay
naren	bombay

(6) List all data from table BRANCH.

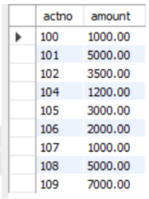
SELECT*FROM branch;



bname	city
vrce	nagpur
ajni	nagpur
karolbagh	delhi
chandi	delhi
dharampeth	nagpur
m.g.road	banglore
andheri	bombay
virar	bombay
nehru place	delhi
powai	bombay

(7) Give account no and amount of depositors.

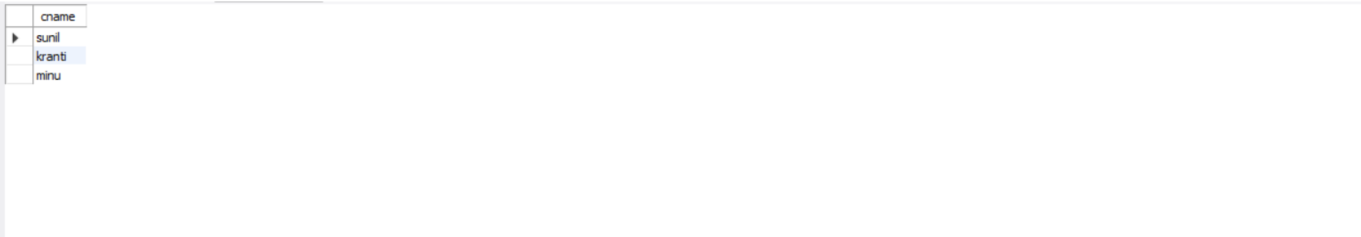
SELECT actno,amount FROM deposit;



actno	amount
100	1000.00
101	5000.00
102	3500.00
104	1200.00
105	3000.00
106	2000.00
107	1000.00
108	5000.00
109	7000.00

(8) Give name of depositors having amount greater than 4000.

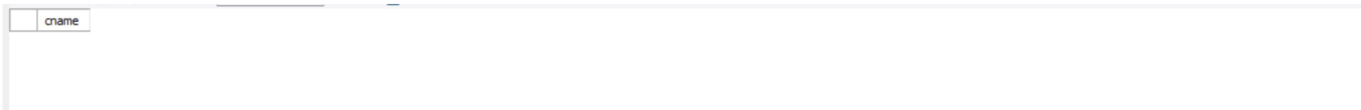
SELECT cname FROM deposit
WHERE amount>4000.00;



cname
sunil
kranti
minu

(9) Give name of customers who opened account after date '1-12-96'.

SELECT cname FROM deposit
WHERE adate>"1996-12-01"



cname

****

Practical :2

Aim : Create the below given tables and insert the data accordingly.

Create given tables

1. Table - Job

COLUMN NAME	DATA TYPE
job_id	Varchar2(15)
job_title	Varchar2(30)
min_sal	Number(7,2)
max_sal	Number(7,2)

2. Table - Employee

COLUMN NAME	DATA TYPE
emp_no	Number(3)
emp_name	Varchar2(30)
emp_sal	Number(8,2)
emp_comm	Number(6,1)
dept_no	Number(3)

3. Table - Deposit

COLUMN NAME	DATA TYPE
a_no	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Number(7,2)
a_date	Date

4. Table - Borrow

COLUMN NAME	DATA TYPE
Loanno	Varchar2(5)
Cname	Varchar2(15)
Bname	Varchar2(10)
Amount	Varchar2(7,2)

Insert following values in the table Employee.

emp_n	emp_name	emp_sal	emp_comm	dept _no
101	Smith	800		20
102	Snehal	1600	300	25
103	Adama	1100	0	20
104	Aman	3000		15
105	Anita	5000	50,000	10

106	Sneha	2450	24,500	10
107	Anamika	2975		30

Insert following values in the table job.

job_id	job_name	min_sal	max_sal
IT_PROG	Programmer	4000	10000
MK_MGR	Marketing manager	9000	15000
FI_MGR	Finance manager	8200	12000
FI_ACC	Account	4200	9000
LEC	Lecturer	6000	17000
COMP_OP	Computer Operator	1500	3000

Insert following values in the table deposit.

A_no	cname	Bname	Amount	date
101	Anil	andheri	7000	01-jan-06
102	sunil	virar	5000	15-jul-06
103	jay	villeparle	6500	12-mar-06
104	vijay	andheri	8000	17-sep-06
105	keyur	dadar	7500	19-nov-06
106	mayur	borivali	5500	21-dec-06

Perform following queries

1. Retrieve all data from employee, jobs and deposit.

2. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

3. Display all jobs with minimum salary is greater than 4000.

4. Display name and salary of employee whose department no is 20. Give alias name to name of employee.

5. Display employee no,name and department details of those employee whose department lies in(10,20)

To study various options of LIKE predicate

6. Display all employee whose name start with ‘A’ and third character is ‘a’.

7. Display name, number and salary of those employees whose name is 5 character long and first three character are ‘Ani’.

8. Display the non-null values of employee and also employee name second character should be ‘n’ and string should be 5 character long.

9. Display the null values of employee and also employee name’s third character should be ‘a’.

10. what will be output if you are giving LIKE predict as ‘%_%’.

****

Practical: 3

Aim: To perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

1. List total deposit from deposit.

2. List total loan from karolbagh branch.

3. Give maximum loan from branch vrce.

4. Count total number of customers.

5. Count total no. of customer's cities.

6. Create table supplier from employee with all the columns.

7. Create table sup1 from employee with first two columns.

8. Create table sup2 from employee with no data.

9. Insert the data in sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.

10. Delete all the rows from sup1.

11. Delete the detail of supplier whose sup_no is 103.

12. Rename the table sup2.

13. Destroy table sup1 with all the data.

14. Update the value dept_no to10 where second character of emp name is'm'.

15. Update the value of employee name whose employee number is 103.

****

Practical :4

Aim : To study Single-row functions.

1. Write a query to display the current date. Label the column Date.

2. For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary.

3. Modify your query no 4.(2) to add a column that subtracts the old salary from the new salary. Label the column Increase.

4. Write a query that display the employee’s name with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employee’s last name.

5. Write a query that produces the following for each employee:

<employee last name> earns <salary> monthly

6. Display the name, hire date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.

7. Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM.

8. Write a query to calculate the annual compensation of all employees (sal--comm.).

****

Practical: 5

Aim: Displaying data from Multiple Tables (join).

1. Give details of customers ANIL.

2. Give name of customer who are borrowers and depositors and having living city Nagpur.

3. Give city as their city name of customers having same living branch.

4. Write a query to display the last name, department number, and department name for all employees.

5. Create a unique listing of all jobs that are in department 30. Include the location of the department in the output.

6. Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.

7. Display the employee last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

8. Create a query to display the name and hire date of any employee hired after employee Aman.

****

Practical :6

Aim : To apply the concept of Aggregating Data using Group functions.

1. List total deposit of customer having account date after 1-jan-96.

2. List total deposit of customers living in city Nagpur.

3 .List maximum deposit of customers living in bombay.

4. Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

5. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

6. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998.

7. Find the average salaries for each department without displaying the respective department numbers.

8. Write a query to display the total salary being paid to each job title, within each department.

9. Find the average salaries > 2000 for each department without displaying the respective department numbers.

10. Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.

11. List the branches having sum of deposit more than 5000 and located in city bombay.

****

Practical: 7

Aim: To solve queries using the concept of sub query.

1. Write a query to display the last name and hire date of any employee in the same department as AMAN. Exclude AMAN.

2. Give name of customers who are depositors having same branch city of mr. sunil.

3. Give deposit details and loan details of customer in same city where pramod is living.

4. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

5. Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000.

6. Display the last name and salary of every employee who reports to ford.

7. Display the department number, name, and job for every employee in the Accounting department.

8. List the name of branch having highest number of depositors.

9. Give the name of cities where in which the maximum numbers of branches are located.

10. Give name of customers living in same city where maximum depositors are located.

****

Practical :8

Aim : To apply the concept of Manipulating Data.

1. Give 10% interest to all depositors.

2. Give 10% interest to all depositors having branch vrce.

3. Give 10% interest to all depositors living in nagpur and having branch city bombay.

4. Write a query which changes the department number of all employees with empno 7788’s job to employee 7844’s current department number.

5. Transfer 10 Rs from account of anil to sunil if both are having same branch.

6. Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.

7. Delete depositors of branches having number of customers between 1 to 3.

8. Delete deposit of vijay.

9. Delete borrower of branches having average loan less than 1000.

****

Practical: 9

Aim: PL/SQL block for loop, arithmetic operations, function and procedure

1. Write a PL/SQL program to illustrates the use of varrays.

2. Write a PL/SQL program that illustrates the concept of loop

3. Write a PL/SQL program to create a simple procedure that displays the string 'Hello World!' on the screen when executed

4. Write a PL/SQL program that makes the use of cursor to display CUSTOMER data in the customers table.

5. Write a PL/SQL function that returns the total number of CUSTOMERS in the customers table.

6. Write a PL/SQL Function that computes and returns the maximum of two values.

7. Write a PL/SQL program to calculates the factorial of a given number by calling itself recursively

****

Practical :10

Aim : To apply the concept of security and privileges.

Database Security

- Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks.
- Database security covers and enforces security on all aspects and components of databases. This includes:
  - Data stored in database
  - Database server
  - Database management system (DBMS)
  - Other database workflow applications
- Database security is generally planned, implemented and maintained by a database administrator and or other information security professional.
- Database security is the technique that protects and secures the database against intentional or accidental threats. Security concerns will be relevant not only to the data resides in an organization's database: the breaking of security may harm other parts of the system which may ultimately affect the database structure. Consequently, database security includes hardware part, software part, human resource, and data. To efficiently do the uses of security needs appropriate controls, which are distinct in a specific mission and purpose for the system. The requirement for getting proper security while often having been neglected or overlooked in the past days; is now more and more thoroughly checked by the different organizations.
- We consider database security about the following situations:
  - Theft and fraudulent.
  - Loss of confidentiality or secrecy.
  - Loss of data privacy.
  - Loss of data integrity.
  - Loss of availability of data.

How to improve MYSQL Security:

1. Drop the Test database.
2. Remove all anonymous accounts.

3. Change default port mappings.
4. Alter which hosts have access to MYSQL.
5. Do not run MYSQL with root level privileges.
6. Remove and disable the MYSQL history file.
7. Limit or disable SHOW DATABASES.
8. Disable the use of LOAD DATA LOCAL INFILE command.
9. Set the proper file permissions.

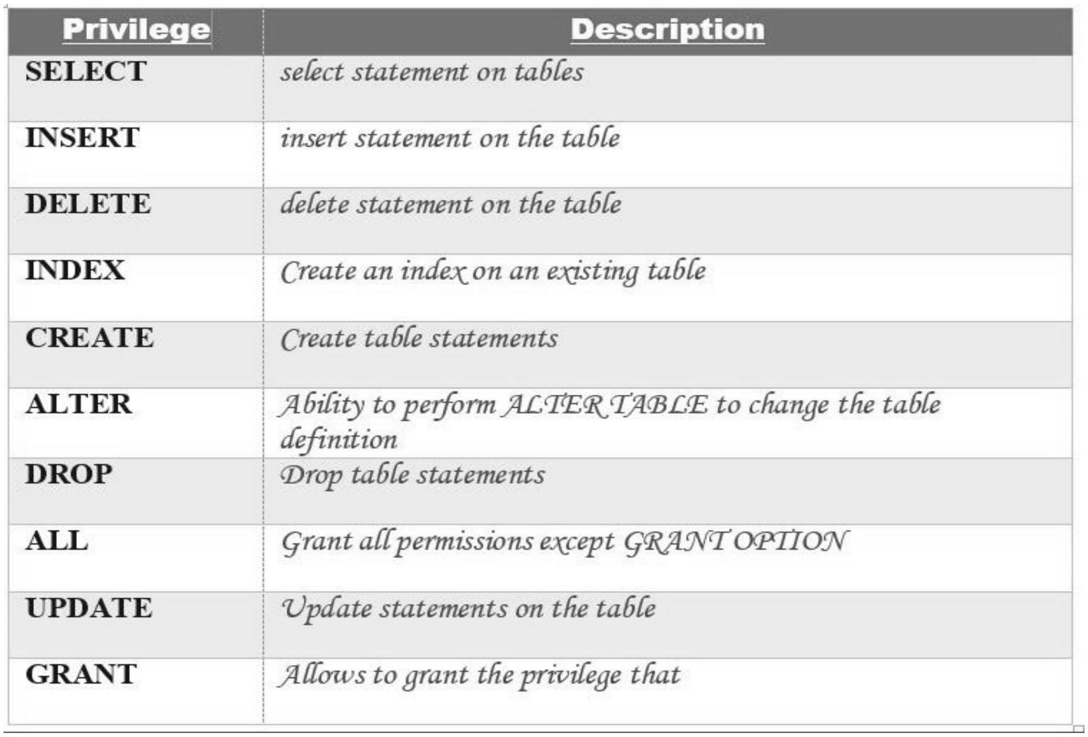
Database Privileges

- A privileges is a right to execute a particular type of SQL statement or to access another user's object.
- Some of the examples of privileges include the right to :
  - Connect to the database(create a session).
  - Create a table.
  - Select rows from another user's table.
  - Execute another user's stored procedure.
- You can grant privileges to user so these users can accomplish tasks required for their job. You should grant a privilege only to a user who absolutely requires the privilege to accomplish necessary work. Excessive granting of unnecessary privileges can compromise security. A user can receive a privilege in two different ways:
  - You can grant privileges to users explicitly. For example, you can explicitly grant the privileges to insert records into the 'EMPLOYEE' table to the user 'SCOTT'.
  - You can also grant privileges to a role (a named group of privileges), and then grant the role of one or more users. For example, you can grant the privileges to select, insert, update and delete records from the 'EMPLOYEE' table to the role named clerk, which in turn you can grant to the users 'SCOTT' and 'BRAIN'.

Granting Privileges

- **SQL GRANT is a command used to provide access or privileges on the database objects to the users.**

  Syntax: GRANT privileges_names ON object TO user;
- **Privileges**:
  The privileges that can be granted to the users are listed below along with description:



Privilege	Description
SELECT	*select statement on tables*
INSERT	*insert statement on the table*
DELETE	*delete statement on the table*
INDEX	*Create an index on an existing table*
CREATE	*Create table statements*
ALTER	*Ability to perform ALTER TABLE to change the table definition*
DROP	*Drop table statements*
ALL	*Grant all permissions except GRANT OPTION*
UPDATE	*Update statements on the table*
GRANT	*Allows to grant the privilege that*

- **Object** − The name of an object to which to grant access. The possible objects are: table, view, sequence
- **PUBLIC** − A short form representing all users.
- GROUP **group** − A group to whom to grant privileges.
- **username** − The name of a user to whom to grant privileges. PUBLIC is a short form representing all users.

EXAMPLE:

GRANT SELECT ON person_details TO user1;

- This query will grant the SELECT permission to person_details table to user named user1.

- For granting all privileges to user we use **sysdba.**

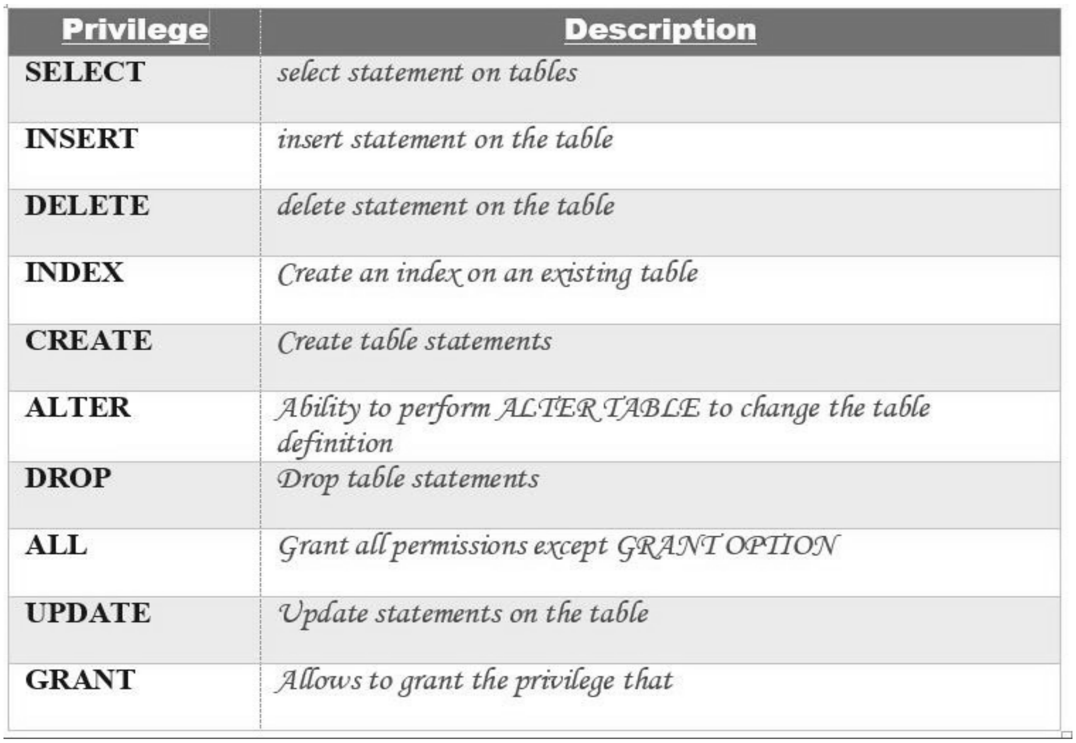
GRANT sysdba to user1;

Revoking Privileges

- The Revoke statement is used to revoke some or all of the privileges which have been granted to a user in the past.

Syntax: REVOKE privileges ON object FROM user;

- **Privileges**
  Privileges can be of the following values:



Privilege	Description
SELECT	*select statement on tables*
INSERT	*insert statement on the table*
DELETE	*delete statement on the table*
INDEX	*Create an index on an existing table*
CREATE	*Create table statements*
ALTER	*Ability to perform ALTER TABLE to change the table definition*
DROP	*Drop table statements*
ALL	*Grant all permissions except GRANT OPTION*
UPDATE	*Update statements on the table*
GRANT	*Allows to grant the privilege that*

- **Object** − The name of an object to which to grant access. The possible objects are: table, view, sequence
- **PUBLIC** − A short form representing all users.
- GROUP **group** − A group to whom to grant privileges.
- **username** − The name of a user to whom to grant privileges. PUBLIC is a short form representing all users.

EXAMPLE:

- To remove access right for SELECT to the table ‘person_details’ for ‘user1’ we write the query

```
REVOKE SELECT ON person_details FROM user1;
```

****

Practical :11

Aim : To Study Transaction control command.

- A transaction is a unit of work that is performed against a database. Transactions are units or sequences of work accomplished in a logical order, whether in a manual fashion by a user or automatically by some sort of a database program.
- A transaction is the propagation of one or more changes to the database. For example, if you are creating a record or updating a record or deleting a record from the table, then you are performing a transaction on that table. It is important to control these transactions to ensure the data integrity and to handle database errors.
- Practically, you will club many SQL queries into a group and you will execute all of them together as a part of a transaction.

PROPERTIES OF TRANSACTION:

- Transactions have the following four standard properties, usually referred to by the acronym **ACID**.
  - **Atomicity** − ensures that all operations within the work unit are completed successfully. Otherwise, the transaction is aborted at the point of failure and all the previous operations are rolled back to their former state.
  - **Consistency** − ensures that the database properly changes states upon a successfully committed transaction.
  - **Isolation** − enables transactions to operate independently of and transparent to each other.
  - **Durability** − ensures that the result or effect of a committed transaction persists in case of a system failure.

TRANSACTION CONTROL:

- The following commands are used to control transactions.
- **COMMIT** − to save the changes.
- **ROLLBACK** − to roll back the changes.
- **SAVEPOINT** − creates points within the groups of transactions in which to ROLLBACK.
- **SET TRANSACTION** − Places a name on a transaction.

TRANSACTION CONTROL COMMAND:

- Transactional control commands are only used with the **DML Commands** such as - INSERT, UPDATE and DELETE only. They cannot be used while creating tables or dropping them because these operations are automatically committed in the database.

COMMIT COMMAND:

- The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.

- The COMMIT command is the transactional command used to save changes invoked by a transaction to the database. The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

- The syntax for the COMMIT command is as follows.

```
COMMIT;
```

ROLLBACK COMMAND:

- The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database. This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

- The syntax for a ROLLBACK command is as follows −

```
ROLLBACK;
```

SAVEPOINT COMMAND:

- A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

- The syntax for a SAVEPOINT command is as shown below.

```
SAVEPOINT SAVEPOINT_NAME;
```

- This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

- The syntax for rolling back to a SAVEPOINT is as shown below.

```
ROLLBACK TO SAVEPOINT_NAME;
```

RELEASE SAVEPOINT COMMAND:

- The RELEASE SAVEPOINT command is used to remove a SAVEPOINT that you have created.

- The syntax for a RELEASE SAVEPOINT command is as follows.

```
RELEASE SAVEPOINT SAVEPOINT_NAME;
```

- Once a SAVEPOINT has been released, you can no longer use the ROLLBACK command to undo transactions performed since the last SAVEPOINT.

SET TRANSACTION COMMAND:

- The SET TRANSACTION command can be used to initiate a database transaction. This command is used to specify characteristics for the transaction that follows. For example, you can specify a transaction to be read only or read write.

- The syntax for a SET TRANSACTION command is as follows.

```
SET TRANSACTION [ READ WRITE | READ ONLY ];
```

EXAMPLE:

- Consider the CUSTOMERS table having the following records-

ID	NAME	AGE	ADDRESS	SALARY
1	RAMESH	32	Ahmedabad	2000
2	KHILAN	25	Delhi	1500
3	KAUSHIK	23	Kota	2000
4	CHAITALI	25	Mumbai	6500
5	HARDIK	27	Bhopal	8500
6	KOMAL	22	MP	4500
7	MUFI	24	Indore	10000

- Following is an example which would delete those records from the table which have age = 25 and then COMMIT the changes in the database.

- MYSQL> DELETE FROM CUSTOMERS WHERE AGE = 25;

- MYSQL> COMMIT;

- Thus, two rows from the table would be deleted and the SELECT statement would produce the following result.

ID	NAME	AGE	ADDRESS	SALARY
1	RAMESH	32	Ahmedabad	2000
3	KAUSHIK	23	Kota	2000
5	HARDIK	27	Bhopal	8500
6	KOMAL	22	MP	4500
7	MUFI	24	Indore	10000

Following is an example, which would delete those records from the table which have the age = 25 and then ROLLBACK the changes in the database.

- MYSQL> DELETE FROM CUSTOMERS WHERE AGE = 25;
- MYSQL> ROLLBACK;
- Thus, the delete operation would not impact the table and the SELECT statement would produce the following result.

ID	NAME	AGE	ADDRESS	SALARY
1	RAMESH	32	Ahmedabad	2000
2	KHILAN	25	Delhi	1500
3	KAUSHIK	23	Kota	2000
4	CHAITALI	25	Mumbai	6500
5	HARDIK	27	Bhopal	8500
6	KOMAL	22	MP	4500
7	MUFI	24	Indore	10000

- MYSQL> SAVEPOINT SP1;

  Savepoint created.

****