

DBMS Lab Assignment – 3 PRN : 23070122074**Introduction to DDL and DML commands and its execution.**

Data definition language defines the schema for the database by specifying entities and the relationship among them. In addition to this, DDL even defines certain security constraints. The execution of DDL statements results in new tables which are stored in "system catalog" also called data dictionary or data directory.

Data Manipulation Language is a language that provides a set of operations to support the basic data manipulation operations on the data held in the databases. It allows users to insert, update, delete and retrieve data from the database. Data manipulations are applied at internal, conceptual and external levels of schemas. However, the level of complexity at each schema level varies from one another.

Data Control Language statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a User with the help of GRANT statement. The privileges assigned can be SELECT, ALTER, DELETE, EXECUTE, INSERT, INDEX etc. In addition to granting of privileges, you can also revoke (taken back) it by using REVOKE command.

DDL : Data Definition Language

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

DML : Data Manipulation Language

DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

DCL : Data Control Language

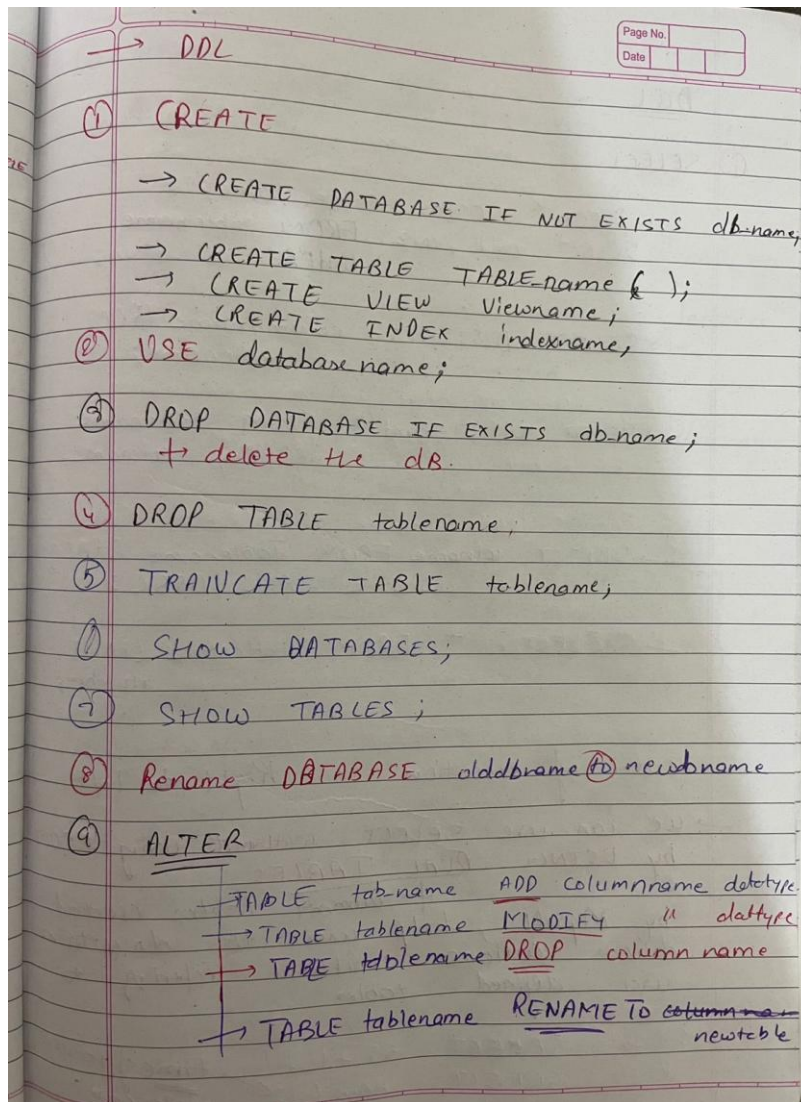
Data control language provides command to grant and take back authority.

Command	Description
grant	grant permission of right
revoke	take back permission.

Assignment 3 Lab Questions

Section- 1 DDL Commands

Q-1 Explain DDL commands and their syntax



DQL

① SELECT

→ SELECT * FROM table_name;
 → SELECT col1, col2 FROM tablename
 WHERE condition;

→ SELECT DISTINCT columnname FROM
 tablename;

→ SELECT col1, count(*) FROM tablename
 GROUP BY colname;

→ SELECT colname FROM Tablename ORDER BY
 colname ASC | DESC;

→ SELECT * FROM tablename LIMIT
 number;

→ Order of execution is Right to Left

→ We can use SELECT without using FROM
 by USING DUAL TABLES.

→ Dummy tables created
 by MySQL, help users to do certain
 obvious actions without referring to
 user defined tables

Eg → SELECT 58 + 11; ← time of server
 SELECT now(); ← uppercase

① Where → to apply condition
 + select * FROM CUSTOMER where sal > 1000

② Between →
 + SELECT * FROM customer WHERE age BETWEEN
 0 AND 100
 Inclusive

③ IN → Reduces OR condition

SELECT * FROM officers WHERE officername
IN ('Mahi', 'Vishu');

④ WHERE

condition 1 AND condition 2
condition 1 OR condition 2
columnname NOT IN (1, 2, 3, 4)

⑤ IS NULL
 S * F cust WHERE columnname IS NULL

⑥ Pattern Searching / (Wildcard) ('%', '-')

% , any character b/w 0 to n.

- , only one character

SELECT * FROM CUSTOMER ~~ORDER BY~~
 WHERE name LIKE '%p-';

Second last letter (p)

Wild cards

- ① %pa% → any no of characters
abc pa, bc pa etc.
- ② _pa_ → can be replaced by only one char.
Eg → a(pa)b

③ SORTING → Ascending to descending
↓
ORDER BY (ASC) → Default

Eg SELECT * FROM workers ORDER BY SALARY;

→ For Descending to use DESC

Eg → SELECT * FROM workers ORDER BY salary
DESC;

Page No.
 Date

③ DISTINCT values

will give all distinct values

Ex	Distinct
HR	
HR	HR
HR	Admin
Admin	Acc
Acc	

SELECT DISTINCT department FROM worker;

④ Data GROUPING

For eg → I have to group all workers of HR department etc etc

Keyword → GROUP BY

↓

Generally used with some Aggregation function Eg → COUNT, SUM, AVG, MIN, MAX

→ SELECT department FROM worker

↓

GROUP BY department;

Otherwise it will act as DISTINCT

Both should be same

now we have to use Aggregation

→ SELECT department, COUNT(department)

FROM Worker GROUP BY department;

Avg salary per department

```
SELECT department, AVG(salary) from
worker GROUP BY department;
```

(11) HAVING → similar to WHERE

→ Used for filtering in Group by

→ we want departments ~~with~~ having more than 2 workers

```
SELECT department, COUNT(department)
FROM worker GROUP BY department
HAVING COUNT(department) > 2;
```

→ WHERE VS. HAVING

(1) Both have same function of filtering row base on certain condition

(2) WHERE clause is used to filter rows from the table based on specific condition

Q-2 Create tables for the following relational model of library management. Apply constraints on the columns and also alter the structure according to your requirements.

1. **SIULIBRARY** (Slid,lname,location,noofbranches)

```

create table SIULIBRARY(
    Slid int primary key not null,
    lname varchar(50),
    location varchar(50),
    noofbranches numeric
);

use siulibrary;

-- Inserting data into SIULIBRARY
INSERT INTO SIULIBRARY VALUES (1, 'Central Library', 'Pune', 3);
INSERT INTO SIULIBRARY VALUES (2, 'City Library', 'Mumbai', 5);
INSERT INTO SIULIBRARY VALUES (3, 'State Library', 'Delhi', 4);
INSERT INTO SIULIBRARY VALUES (4, 'Regional Library', 'Chennai', 2);
INSERT INTO SIULIBRARY VALUES (5, 'District Library', 'Bangalore', 6);

```

	Slid	lname	location	noofbranches
▶	1	Central Library	Pune	3
	2	City Library	Mumbai	5
	3	State Library	Delhi	4
	4	Regional Library	Chennai	2
	5	District Library	Bangalore	6
✱	NULL	NULL	NULL	NULL

2. Ilibrary(Lid, lname, city, area, slid)

```

• CREATE TABLE Ilibrary (
    lid INT PRIMARY KEY NOT NULL,
    lname VARCHAR(50),
    city VARCHAR(50),
    area VARCHAR(50),
    slid INT,
    FOREIGN KEY (slid) REFERENCES SIULIBRARY (Slid)
);

```

```

INSERT INTO Ilibrary VALUES (1, 'Main Branch', 'Pune', 'Kothrud', 1);
INSERT INTO Ilibrary VALUES (2, 'Branch A', 'Mumbai', 'Andheri', 2);
INSERT INTO Ilibrary VALUES (3, 'Branch B', 'Delhi', 'Connaught', 3);
INSERT INTO Ilibrary VALUES (4, 'Branch C', 'Chennai', 'T Nagar', 4);
INSERT INTO Ilibrary VALUES (5, 'Branch D', 'Bangalore', 'Indiranagar', 5);

```

Result Grid					
Filter Rows:					
	lid	lname	city	area	slid
▶	1	Main Branch	Pune	Kothrud	1
	2	Branch A	Mumbai		2
	3	Branch B	Delhi	Connaught	3
	4	Branch C	Chennai	T Nagar	4
	5	Branch D	Bangalore	Indiranagar	5
•	NULL	NULL	NULL	NULL	NULL

3. BOOKS(Bid, Bname, Price, Lid)

```

CREATE TABLE BOOKS (
    Bid INT PRIMARY KEY NOT NULL,
    Bname VARCHAR(50),
    Price NUMERIC,
    Lid INT,
    FOREIGN KEY (Lid) REFERENCES Ilibrary (lid)
);

```

```

INSERT INTO BOOKS VALUES (1, 'Database Systems', 500, 1);
INSERT INTO BOOKS VALUES (2, 'Operating Systems', 600, 2);
INSERT INTO BOOKS VALUES (3, 'Networking Basics', 450, 3);
INSERT INTO BOOKS VALUES (4, 'Machine Learning', 700, 4);
INSERT INTO BOOKS VALUES (5, 'AI Fundamentals', 800, 5);

```

	Bid	Bname	Price	Lid
▶	1	Database Systems	500	1
	2	Operating Systems	600	2
	3	Networking Basics	450	3
	4	Machine Learning	700	4
	5	AI Fundamentals	800	5

4. Noofcopies(bnid, bid, lid)

```

CREATE TABLE Noofcopies (
    bnid INT PRIMARY KEY NOT NULL,
    bid INT,
    blid INT,
    FOREIGN KEY (bid) REFERENCES BOOKS (Bid),
    FOREIGN KEY (blid) REFERENCES Ilibrary (lid)
);

```

```

INSERT INTO Noofcopies VALUES (1, 1, 1);
INSERT INTO Noofcopies VALUES (2, 2, 2);
INSERT INTO Noofcopies VALUES (3, 3, 3);
INSERT INTO Noofcopies VALUES (4, 4, 4);
INSERT INTO Noofcopies VALUES (5, 5, 5);

```

	bnid	bid	blid
▶	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
✱	NULL	NULL	NULL

5. **AUTHOR**(Aid, Aname, email, phoneno)

```

CREATE TABLE AUTHOR (
    Aid INT PRIMARY KEY NOT NULL,
    Aname VARCHAR(50),
    email VARCHAR(50),
    phoneno VARCHAR(15)
);

```

```

INSERT INTO AUTHOR VALUES (1, 'John Doe', 'john@example.com', '1234567890');
INSERT INTO AUTHOR VALUES (2, 'Jane Smith', 'jane@example.com', '0987654321');
INSERT INTO AUTHOR VALUES (3, 'Mike Brown', 'mike@example.com', '1122334455');
INSERT INTO AUTHOR VALUES (4, 'Emma Wilson', 'emma@example.com', '5566778899');
INSERT INTO AUTHOR VALUES (5, 'Liam Davis', 'liam@example.com', '6677889900');

```

	Aid	Aname	email	phoneno
▶	1	ayaan	a@g.com	1234
	2	Jane Smith	jane@example.com	0987654321
	3	Mike Brown	mike@example.com	1122334455
	4	Emma Wilson	emma@example.com	5566778899
	5	shruti	liam@example.com	6677889900

6. **Writes**(Bid, Aid, pid)

```
CREATE TABLE Writes (
    Bid INT,
    Aid INT,
    pid INT,
    PRIMARY KEY (Bid, Aid),
    FOREIGN KEY (Bid) REFERENCES BOOKS (Bid),
    FOREIGN KEY (Aid) REFERENCES AUTHOR (Aid)
);

-- Inserting data into table --

INSERT INTO Writes VALUES (1, 1, 101);
INSERT INTO Writes VALUES (2, 2, 102);
INSERT INTO Writes VALUES (3, 3, 103);
INSERT INTO Writes VALUES (4, 4, 104);
INSERT INTO Writes VALUES (5, 5, 105);
```

	Bid	Aid	pid
▶	1	1	101
	2	2	102
	3	3	103
	4	4	104
	5	5	105
✱	NULL	NULL	NULL

7. PUBLISHER(Pid, Pname)

```
CREATE TABLE PUBLISHER (  
    Pid INT PRIMARY KEY NOT NULL,  
    Pname VARCHAR(50)  
);
```

```
INSERT INTO PUBLISHER (Pid, Pname) VALUES  
(1, 'Penguin'),  
(2, 'HarperCollins'),  
(3, 'Macmillan'),  
(4, 'Random House'),  
(5, 'Oxford Press');
```

	Pid	Pname
▶	1	Pearson
	2	HarperCollins
	3	Macmillan
	4	McGraw Hill
	5	McGraw Hill
●	NULL	NULL

8. **SELLER**(Sid, sname, city)

```

CREATE TABLE SELLER (
    Sid INT PRIMARY KEY NOT NULL,
    sname VARCHAR(50),
    city VARCHAR(50)
);

```

```

INSERT INTO SELLER (Sid, sname, city)
VALUES

```

```

    (1, 'BookWorld', 'Delhi'),
    (2, 'ReadersHub', 'Mumbai'),
    (3, 'Pages', 'Bangalore'),
    (4, 'LibraryMart', 'Chennai'),
    (5, 'BookBarn', 'Kolkata');

```

Result Grid			
	Sid	sname	city
▶	1	BookWorld	Delhi
	2	ReadersHub	Mumbai
	3	Pagesta	Bangalore
	4	LibraryMart	Chennai
	5	BookBarn	Kolkata
✱	NULL	NULL	NULL

9. **DEPARTMENT** (Deptid,deptname, Iname,lid)

```

CREATE TABLE DEPARTMENT (
    -- Deptid INT PRIMARY KEY NOT NULL,
    deptname VARCHAR(50),
    Iname VARCHAR(50),
    lid INT,
    FOREIGN KEY (lid) REFERENCES Ilibrary (lid)
);

```

```

INSERT INTO DEPARTMENT (Deptid, deptname, Iname, lid)
VALUES

```

```

    (1, 'Science', 'Central Library', 1),
    (2, 'Arts', 'City Library', 2),
    (3, 'Commerce', 'University Library', 3),
    (4, 'Engineering', 'Tech Library', 4),
    (5, 'Law', 'Legal Library', 5);

```

	Deptid	deptname	Iname	lid
▶	1	Computer Science	Central Library	1
	2	Arts	City Library	2
	3	Commerce	University Library	3
	4	Engineering	Tech Library	4
	5	Civil	Legal Library	5
✱	NULL	NULL	NULL	NULL

10. STUDENT(Stuid, Sname, email, memid, deptid)

```

CREATE TABLE STUDENT (
    Stuid INT PRIMARY KEY NOT NULL,
    Sname VARCHAR(50),
    email VARCHAR(50),
    memid INT,
    deptid INT,
    FOREIGN KEY (deptid) REFERENCES DEPARTMENT (Deptid)
);

```

```

INSERT INTO STUDENT (Stuid, Sname, email, memid, deptid)
VALUES

```

```

    (1, 'Alice', 'alice@example.com', 101, 1),
    (2, 'Bob', 'bob@example.com', 102, 2),
    (3, 'Charlie', 'charlie@example.com', 103, 3),
    (4, 'David', 'david@example.com', 104, 4),
    (5, 'Eva', 'eva@example.com', 105, 5);

```

	Stuid	Sname	email	memid	deptid
▶	1	Alice	alice@example.com	101	1
	2	Bob	bob@example.com	102	2
	3	Charlie	charlie@example.com	103	3
	4	David	david@example.com	104	4
	5	Eva	eva@example.com	105	5
✱	NULL	NULL	NULL	NULL	NULL

11. **STAFF**(Stid, sname, email, deptid, memid)

```

CREATE TABLE STAFF (
    Stid INT PRIMARY KEY NOT NULL,
    sname VARCHAR(50),
    email VARCHAR(50),
    deptid INT,
    memid INT,
    FOREIGN KEY (deptid) REFERENCES DEPARTMENT (Deptid)
);

```

```

INSERT INTO STAFF (Stid, sname, email, deptid, memid)
VALUES
(1, 'Mr. Smith', 'smith@example.com', 1, 201),
(2, 'Ms. Johnson', 'johnson@example.com', 2, 202),
(3, 'Mr. Lee', 'lee@example.com', 3, 203),
(4, 'Ms. Brown', 'brown@example.com', 4, 204),
(5, 'Mr. Davis', 'davis@example.com', 5, 205);

```

	Stid	sname	email	deptid	memid
▶	1	Mr. Smith	smith@example.com	1	201
	2	Ms. Johnson	johnson@example.com	2	202
	3	Mr. Lee	lee@example.com	3	203
	4	Ms. Brown	brown@example.com	4	204
	5	Mr. Davis	davis@example.com	5	205
•	NULL	NULL	NULL	NULL	NULL

12. **PURCHASE**(prid , lid, sid, pid, bid, quantity ,date, totalcost)

```

CREATE TABLE PURCHASE (
    prid INT PRIMARY KEY NOT NULL,
    lid INT,
    sid INT,
    pid INT,
    bid INT,
    quantity INT,
    date DATE,
    totalcost NUMERIC,
    FOREIGN KEY (lid) REFERENCES Ilibrary (lid),
    FOREIGN KEY (sid) REFERENCES SELLER (Sid),
    FOREIGN KEY (pid) REFERENCES PUBLISHER (Pid),
    FOREIGN KEY (bid) REFERENCES BOOKS (Bid)
);

```

```

INSERT INTO PURCHASE (prid, lid, sid, pid, bid, quantity, date, totalcost)
VALUES

```

```

(1, 1, 1, 1, 1, 10, '2024-01-01', 5000),
(2, 2, 2, 2, 2, 15, '2024-01-02', 7500),
(3, 3, 3, 3, 3, 20, '2024-01-03', 10000),
(4, 4, 4, 4, 4, 25, '2024-01-04', 12500),
(5, 5, 5, 5, 5, 30, '2024-01-05', 15000);

```

	prid	lid	sid	pid	bid	quantity	date	totalcost
▶	1	1	1	1	1	10	2024-01-01	5000
	2	2	2	2	2	15	2024-01-02	7500
	3	3	3	3	3	20	2024-01-03	10000
	4	4	4	4	4	25	2024-01-04	12500
	5	5	5	5	5	30	2024-01-05	15000
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13.ISSUE(Issueid, memid, bid, lid, issuedate, returndate)

```
CREATE TABLE ISSUE (
    Issueid INT PRIMARY KEY NOT NULL,
    memid INT,
    bid INT,
    lid INT,
    issuedate DATE,
    returndate DATE,
    FOREIGN KEY (memid) REFERENCES STUDENT (Stuid),
    FOREIGN KEY (bid) REFERENCES BOOKS (Bid),
    FOREIGN KEY (lid) REFERENCES Ilibrary (lid)
);
```

```
INSERT INTO ISSUE (Issueid, memid, bid, lid, issuedate, returndate)
VALUES
```

```
(1, 1, 1, 1, '2024-01-10', '2024-02-10'),
(2, 2, 2, 2, '2024-01-11', '2024-02-11'),
(3, 3, 3, 3, '2024-01-12', '2024-02-12'),
(4, 4, 4, 4, '2024-01-13', '2024-02-13'),
(5, 5, 5, 5, '2024-01-14', '2024-02-14');
```

	Issueid	memid	bid	lid	issuedate	returndate
▶	1	1	1	1	2024-01-10	2024-02-10
	2	2	2	2	2024-01-11	2024-02-11
	3	3	3	3	2024-01-12	2024-02-12
	4	4	4	4	2024-01-13	2024-02-13
	5	5	5	5	2024-01-14	2024-02-14
●	NULL	NULL	NULL	NULL	NULL	NULL

14.SELLS (sid,bid,pid)

```

CREATE TABLE SELLS (
    sid INT,
    bid INT,
    pid INT,
    PRIMARY KEY (sid, bid, pid),
    FOREIGN KEY (sid) REFERENCES SELLER (Sid),
    FOREIGN KEY (bid) REFERENCES BOOKS (Bid),
    FOREIGN KEY (pid) REFERENCES PUBLISHER (Pid)
);

```

INSERT INTO SELLS (sid, bid, pid)

VALUES

```

(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5);

```

	sid	bid	pid
▶	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
●	NULL	NULL	NULL

15.Employee(eid,empname,email,salary,lid)

```
CREATE TABLE Employee (
    eid INT PRIMARY KEY NOT NULL,
    empname VARCHAR(50),
    email VARCHAR(50),
    salary NUMERIC,
    lid INT,
    FOREIGN KEY (lid) REFERENCES Ilibrary (lid)
);
```

```
INSERT INTO Employee (eid, empname, email, salary, lid)
VALUES
```

```
(1, 'Emma', 'emma@example.com', 30000, 1),
(2, 'Liam', 'liam@example.com', 32000, 2),
(3, 'Olivia', 'olivia@example.com', 34000, 3),
(4, 'Noah', 'noah@example.com', 36000, 4),
(5, 'Ava', 'ava@example.com', 38000, 5);
```

	eid	empname	email	salary	lid
▶	1	Emma	emma@example.com	30000	1
	2	Liam	liam@example.com	32000	2
	3	Olivia	olivia@example.com	34000	3
	4	Noah	noah@example.com	36000	4
	5	Ava	ava@example.com	38000	5
•	NULL	NULL	NULL	NULL	NULL

16.A_specialization(spec_id,spec_name,Aid)

```
CREATE TABLE A_specialization (  
    spec_id INT PRIMARY KEY NOT NULL,  
    spec_name VARCHAR(50),  
    Aid INT,  
    FOREIGN KEY (Aid) REFERENCES AUTHOR (Aid)  
);
```

```
INSERT INTO A_specialization (spec_id, spec_name, Aid)  
VALUES  
    (1, 'Fiction', 1),  
    (2, 'Science', 2),  
    (3, 'History', 3),  
    (4, 'Technology', 4),  
    (5, 'Mathematics', 5);
```

	spec_id	spec_name	Aid
▶	1	Fiction	1
	2	Science	2
	3	History	3
	4	Technology	4
	5	Mathematics	5
✱	NULL	NULL	NULL

17. **Member**(memid,lid)

```
CREATE TABLE Member (
    memid INT PRIMARY KEY NOT NULL,
    lid INT,
    FOREIGN KEY (lid) REFERENCES Ilibrary (lid)
);
```

```
INSERT INTO Member (memid, lid)
VALUES
```

```
(101, 1),
(102, 2),
(103, 3),
(104, 4),
(105, 5);
```

Result Grid			Filter
	memid	lid	
▶	101	1	
	102	2	
	103	3	
	104	4	
	105	5	
✱	NULL	NULL	

Section- 2 DML Command execution

Q-1. Explain DML commands and their syntax.

DML

Page No.
 Date

→ Data Manipulation language

↓
Use to make change or enter data into a schema or DB

- ① INSERT → to insert data
Syntax
INSERT tablename (col1, col2)
Value (val1, val2);
- ② UPDATE → to update data
UPDATE table_name;
SET col1 = val1, col2 = val2
- ③ DELETE FROM tablename WHERE condition
- ④ MERGE
MERGE INTO target USING SOURCE
ON (target.common col) = source.common col)
WHEN MATCHED THEN
UPDATE SET col1 = val1,

Q-2. Insert 5 tuples in each of the created tables.
ALREADY DONE ABOVE




Q-3 Execute following queries on the library database

1. Which institute libraries are located in pune city?

```

1 • use siulibrary;
2
3 • SELECT * FROM Ilibrary department
4 WHERE city = 'Pune';
5
6 • SELECT Iname
7 FROM DEPARTMENT
8 WHERE city = 'Pune';

```

Result Grid   Filter Rows: <input type="text"/> Edit: 					
	lid	Iname	city	area	slid
▶	1	Main Branch	Pune	Kothrud	1
✱	NULL	NULL	NULL	NULL	NULL

2. To which institute CS department belongs to?

```

6 • SELECT Iname
7 FROM DEPARTMENT
8 WHERE deptname = 'Computer Science';
9
10 • SELECT * FROM BOOKS

```

Result Grid		Filter Rows:	Export:
	Iname		
1	Central Library		

3. Find all the books whose price is between 800 to 12000?

```

9
10 • SELECT * FROM BOOKS
11 WHERE price BETWEEN 800 AND 12000;
12
13 • SELECT * FROM Employee

```

Result Grid					Filter Rows:	Edit:
	Bid	Bname	Price	Lid		
1	5	AI Fundamentals	800	5		
2	NULL	NULL	NULL	NULL		

4. Find out such employees who's salaries are not greater than 50,000/-

```

13 • SELECT * FROM Employee
14 WHERE salary <= 50000;

```

```

15

```

```

16

```

```


17 • SELECT * FROM SELLER

```

```

18 WHERE 1 = 1 AND 1 = 1

```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: 					
	eid	empname	email	salary	lid
▶	1	Emma	emma@example.com	30000	1
	2	Liam	liam@example.com	32000	2
	3	Olivia	olivia@example.com	34000	3
	4	Noah	noah@example.com	36000	4
	5	Ava	ava@example.com	38000	5
•	NULL	NULL	NULL	NULL	NULL

5. Find out such sellers who's name end with "ta"

16

17 • `SELECT * FROM SELLER`18 `WHERE slname LIKE '%ta';`

19

20

21 • `SELECT * FROM Ilibrary`22 `WHERE area IS NULL;`

Result Grid			
	Sid	slname	city
	3	Pagesta	Bangalore
	NULL	NULL	NULL

6. Find out such institute libraries where their area information is missing.

20

21 • `SELECT * FROM Ilibrary`22 `WHERE area IS NULL;`

23

24

25 • `SELECT * FROM STAFF`

Result Grid					
	lid	lname	city	area	slid
*	NULL	NULL	NULL	NULL	NULL

7. Find out such staff members who's name doesn't starts with "A"

24

25 • `SELECT * FROM STAFF`

26 `WHERE sname NOT LIKE 'A%';`

27

28

Result Grid					
		Filter Rows:		Edit:	
	Stid	sname	email	deptid	memid
▶	1	Mr. Smith	smith@example.com	1	201
	2	Ms. Johnson	johnson@example.com	2	202
	3	Mr. Lee	lee@example.com	3	203
	4	Ms. Brown	brown@example.com	4	204
	5	Mr. Davis	davis@example.com	5	205
•	NULL	NULL	NULL	NULL	NULL

8. Find out such SIU libraries which have institute libraries located in Bangalore.

28

29 • `SELECT * FROM Ilibrary`

30 `WHERE city = 'Bangalore';`

31

32

Result Grid					
		Filter Rows:		Edit:	
	lid	Iname	city	area	slid
▶	5	Branch D	Bangalore	Indiranagar	5
•	NULL	NULL	NULL	NULL	NULL

9. Which students belong to civil department?

```

32
33 • SELECT Sname
34 FROM STUDENT
35 WHERE deptid = (SELECT Deptid FROM DEPARTMENT WHERE deptname = 'Civil');
36
37

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Sname
▶	Eva

10. Find out books which are written by “shruti” and published by McGraw hill

```

38 • SELECT b.*
39 FROM BOOKS b
40 JOIN writes w ON b.Bid = w.Bid
41 JOIN AUTHOR a ON w.Aid = a.Aid
42 JOIN SELLS s ON b.Bid = s.bid
43 JOIN PUBLISHER p ON s.pid = p.Pid
44 WHERE a.Aname = 'shruti' AND p.Pname = 'McGraw Hill';
45

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Bid	Bname	Price	Lid
▶	5	AI Fundamentals	800	5

