



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF ELECTRICAL ENGINEERING**

**SMARTBRIDGE EXTERNSHIP**

**Modern Application Development(Java Spring Boot)**

**ASSIGNMENT - 2**

Reg No : 20BEE0344

Name : Anusha Y

Programme : Btech.EEE

## **Create , Update , Delete commands in mysql:**

SHOW DATABASES;

use logicfirst;

```
CREATE TABLE company(  
com_id INT PRIMARY KEY,  
com_name VARCHAR(30),  
job_desc VARCHAR (30),  
salary INT(15)  
);
```

DESCRIBE company;

INSERT INTO company VALUES (1,"anusha" ,"engineer",230000);

INSERT INTO company VALUES (2,"monisha" ,"manager",200000);

INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);

INSERT INTO company VALUES (4,"viji" ,"admin",320000);

INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);

SELECT \* FROM company;

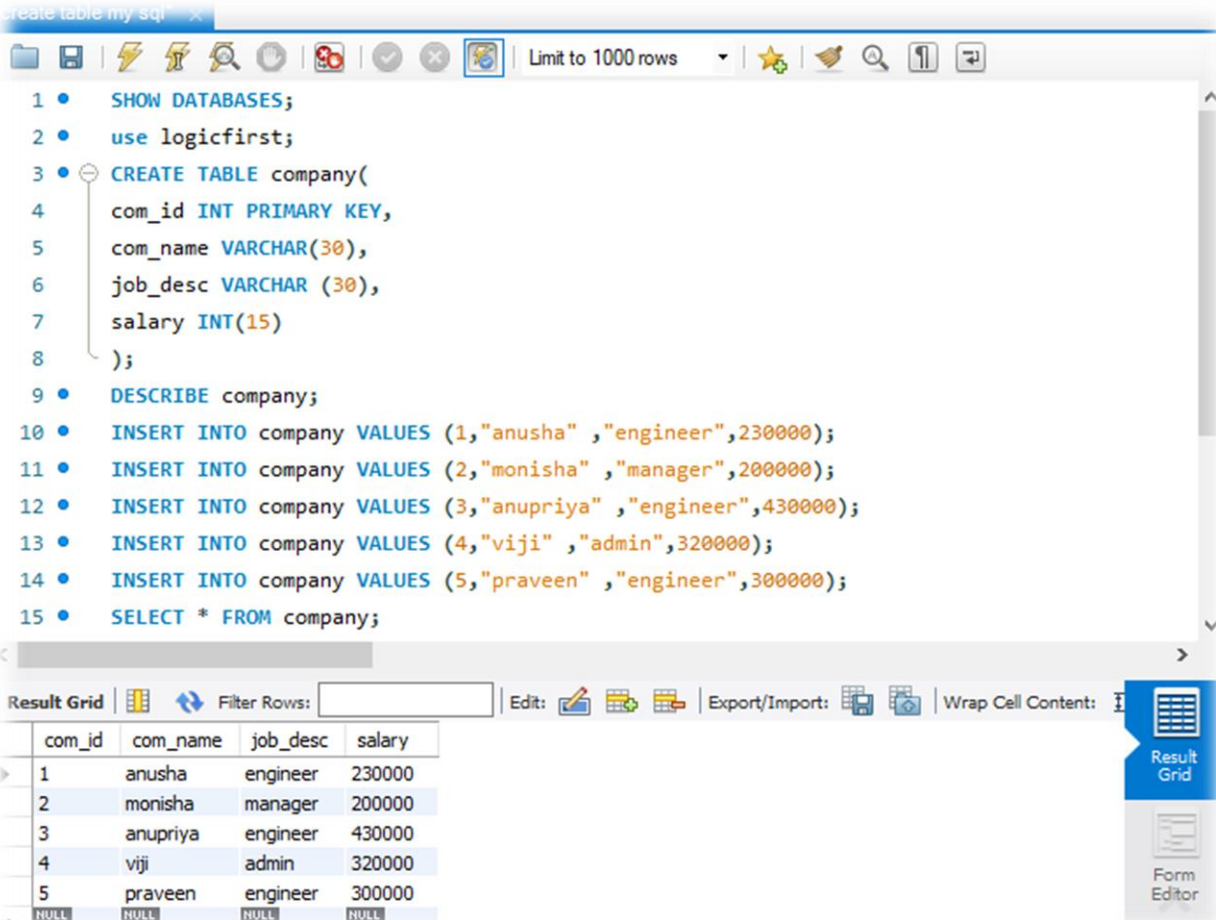
UPDATE company SET em\_name ="anusha" WHERE em\_id=1;

DELETE FROM company WHERE em\_id=1;

SELECT \* FROM company;

## Output:

## Create Table :



The screenshot displays a MySQL IDE window titled "create table my sql". The SQL editor contains the following code:

```
1 • SHOW DATABASES;
2 • use logicfirst;
3 • CREATE TABLE company(
4   com_id INT PRIMARY KEY,
5   com_name VARCHAR(30),
6   job_desc VARCHAR (30),
7   salary INT(15)
8 );
9 • DESCRIBE company;
10 • INSERT INTO company VALUES (1,"anusha" ,"engineer",230000);
11 • INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
12 • INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
13 • INSERT INTO company VALUES (4,"viji" ,"admin",320000);
14 • INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
15 • SELECT * FROM company;
```

Below the editor, the "Result Grid" shows the output of the last query. It includes a toolbar with options like "Filter Rows", "Edit", "Export/Import", and "Wrap Cell Content". The data is presented in a table with 5 rows and 4 columns: com\_id, com\_name, job\_desc, and salary.

com_id	com_name	job_desc	salary
1	anusha	engineer	230000
2	monisha	manager	200000
3	anupriya	engineer	430000
4	viji	admin	320000
5	praveen	engineer	300000
NULL	NULL	NULL	NULL

On the right side of the IDE, there are buttons for "Result Grid" and "Form Editor".

## Update :

create table my sql\* x

Limit to 1000 rows

```
11 • INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
12 • INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
13 • INSERT INTO company VALUES (4,"viji" ,"admin",320000);
14 • INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
15 • SELECT * FROM company;
16 • UPDATE company SET com_name ="lakshmi" WHERE com_id=2;
17 • DELETE FROM company WHERE com_id=1;
18 • SELECT * FROM company;
```

Result Grid

	com_id	com_name	job_desc	salary
▶	2	lakshmi	manager	200000
	3	anupriya	engineer	430000
	4	viji	admin	320000
	5	praveen	engineer	300000
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

## Delete :

create table my sql\* x

Limit to 1000 rows

```
11 • INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
12 • INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
13 • INSERT INTO company VALUES (4,"viji" ,"admin",320000);
14 • INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
15 • SELECT * FROM company;
16 • UPDATE company SET com_name ="lakshmi" WHERE com_id=2;
17 • DELETE FROM company WHERE com_id=1;
18 • SELECT * FROM company;
```

Result Grid

	com_id	com_name	job_desc	salary
▶	2	lakshmi	manager	200000
	3	anupriya	engineer	430000
	4	viji	admin	320000
	5	praveen	engineer	300000
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

## **Joins in Mysql :**

- Inner Join
- Left Join
- Right Join
- Cross Join

## **Inner Join :**

```
SHOW DATABASES;
```

```
use logicfirst;
```

```
CREATE TABLE company(  
com_id INT PRIMARY KEY,  
com_name VARCHAR(30),  
job_desc VARCHAR (30),  
salary INT(15)  
);
```

```
DESCRIBE company;
```

```
INSERT INTO company VALUES (1,"anusha" ,"engineer",230000);
```

```
INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
```

```
INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
```

```
INSERT INTO company VALUES (4,"viji" ,"admin",320000);
```

```
INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
```

```
CREATE TABLE branch(  
  br_id INT PRIMARY KEY,  
  br_name VARCHAR(30),  
  address VARCHAR(50)  
);  
  
DESCRIBE branch;  
  
INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");  
INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");  
INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");  
INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");  
INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");  
  
SELECT * FROM company;  
  
SELECT * FROM branch;  
  
  
SELECT company.com_id,company.com_name,company.job_desc,branch.br_name  
FROM company  
INNER JOIN branch  
ON company.branch_id =branch.branch_id
```

## Output :

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
19 address VARCHAR(50)
20 );
21 • DESCRIBE branch;
22 • INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
23 • INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
24 • INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
25 • INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
26 • INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
27 • SELECT * FROM company;
28 • SELECT * FROM branch;
29
30 • SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
31 FROM company
32 INNER JOIN branch
33 ON company.branch_id =branch.branch_id
```

The result grid displays the output of the query, showing a table with 3 columns: br\_id, br\_name, and address. The table contains 5 rows of data, corresponding to the 5 branches inserted into the branch table. The first row is highlighted in blue.

br_id	br_name	address
1	Delhi	123,B-block road
2	Mumbai	Ab mel street
3	pune	vivekanandha circle g-block
4	gujarath	main road 123 kamarajar nagar
5	Bopal	32 12th street
* NULL	NULL	NULL

## Left Join :

SHOW DATABASES;

use logicfirst;

CREATE TABLE company(

com\_id INT PRIMARY KEY,

com\_name VARCHAR(30),

job\_desc VARCHAR (30),

```
salary INT(15)
```

```
);
```

```
DESCRIBE company;
```

```
INSERT INTO company VALUES (1,"anusha" ,"engineer",230000);
```

```
INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
```

```
INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
```

```
INSERT INTO company VALUES (4,"viji" ,"admin",320000);
```

```
INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
```

```
CREATE TABLE branch(
```

```
br_id INT PRIMARY KEY,
```

```
br_name VARCHAR(30),
```

```
address VARCHAR(50)
```

```
);
```

```
DESCRIBE branch;
```

```
INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
```

```
INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
```

```
INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
```

```
INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
```

```
INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
```

```
SELECT * FROM company;
```

```
SELECT * FROM branch;
```

```
SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
```

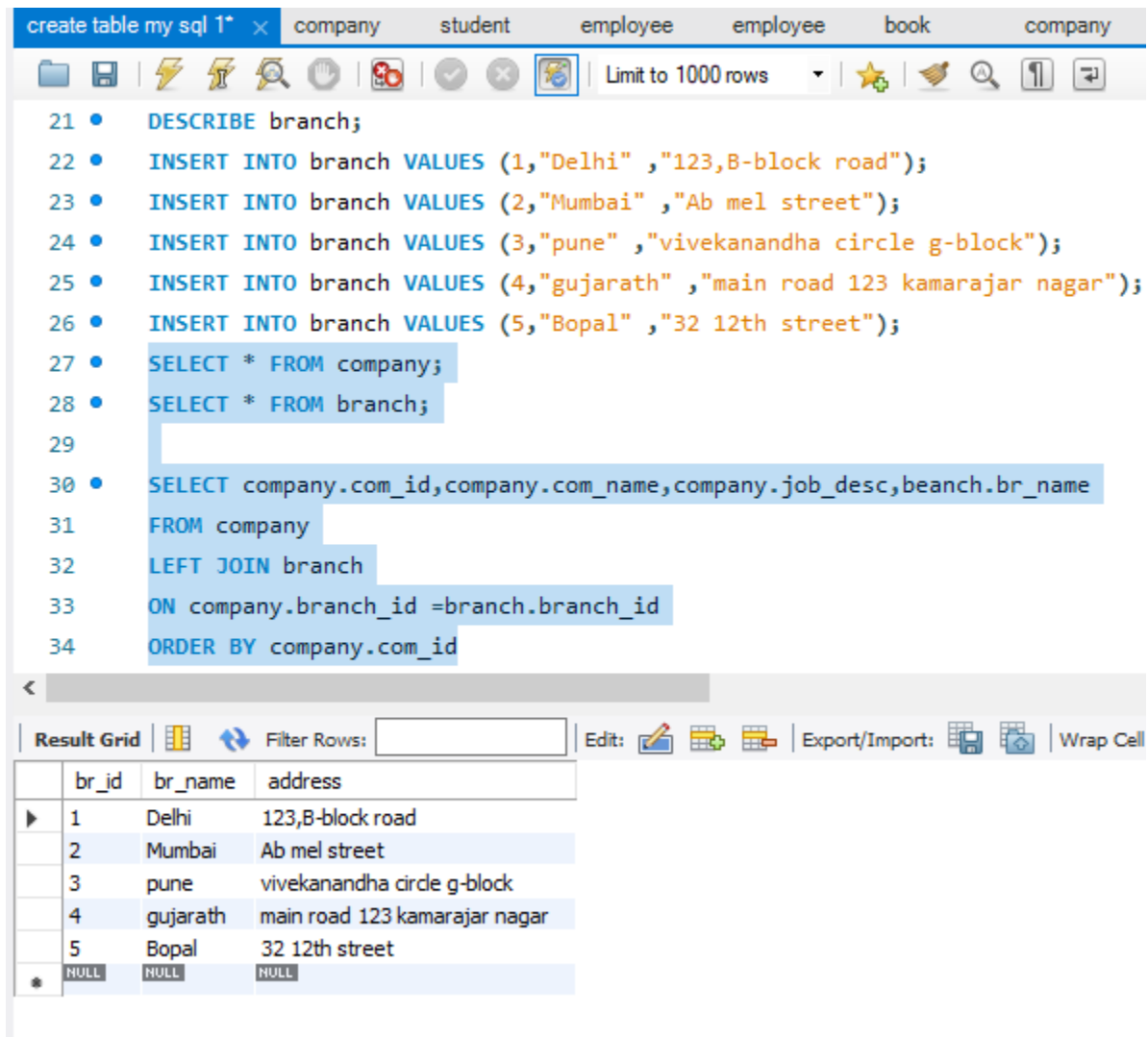
```
FROM company
```

```
LEFT JOIN branch
```



ON company.branch\_id =branch.branch\_id

## Output :



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
21 • DESCRIBE branch;
22 • INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
23 • INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
24 • INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
25 • INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
26 • INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
27 • SELECT * FROM company;
28 • SELECT * FROM branch;
29
30 • SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
31 FROM company
32 LEFT JOIN branch
33 ON company.branch_id =branch.branch_id
34 ORDER BY company.com_id
```

The results grid displays the output of the query. It has columns for br\_id, br\_name, and address. The data is as follows:

br_id	br_name	address
1	Delhi	123,B-block road
2	Mumbai	Ab mel street
3	pune	vivekanandha circle g-block
4	gujarath	main road 123 kamarajar nagar
5	Bopal	32 12th street
NULL	NULL	NULL

## Right Join:

SHOW DATABASES;

use logicfirst;

CREATE TABLE company(

com\_id INT PRIMARY KEY,

```
com_name VARCHAR(30),
```

```
job_desc VARCHAR (30),
```

```
salary INT(15)
```

```
);
```

```
DESCRIBE company;
```

```
INSERT INTO company VALUES (1,"anusha" ,"engineer",230000);
```

```
INSERT INTO company VALUES (2,"monisha" ,"manager",200000);
```

```
INSERT INTO company VALUES (3,"anupriya" ,"engineer",430000);
```

```
INSERT INTO company VALUES (4,"viji" ,"admin",320000);
```

```
INSERT INTO company VALUES (5,"praveen" ,"engineer",300000);
```

```
CREATE TABLE branch(
```

```
br_id INT PRIMARY KEY,
```

```
br_name VARCHAR(30),
```

```
address VARCHAR(50)
```

```
);
```

```
DESCRIBE branch;
```

```
INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
```

```
INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
```

```
INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
```

```
INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
```

```
INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
```

```
SELECT * FROM company;
```

```
SELECT * FROM branch;
```

```

SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
FROM company
RIGHT JOIN branch
ON company.branch_id =branch.branch_id
ORDER BY company.com_id

```

## Output:

The screenshot shows a SQL IDE interface with a script editor and a result grid. The script editor contains the following SQL statements:

```

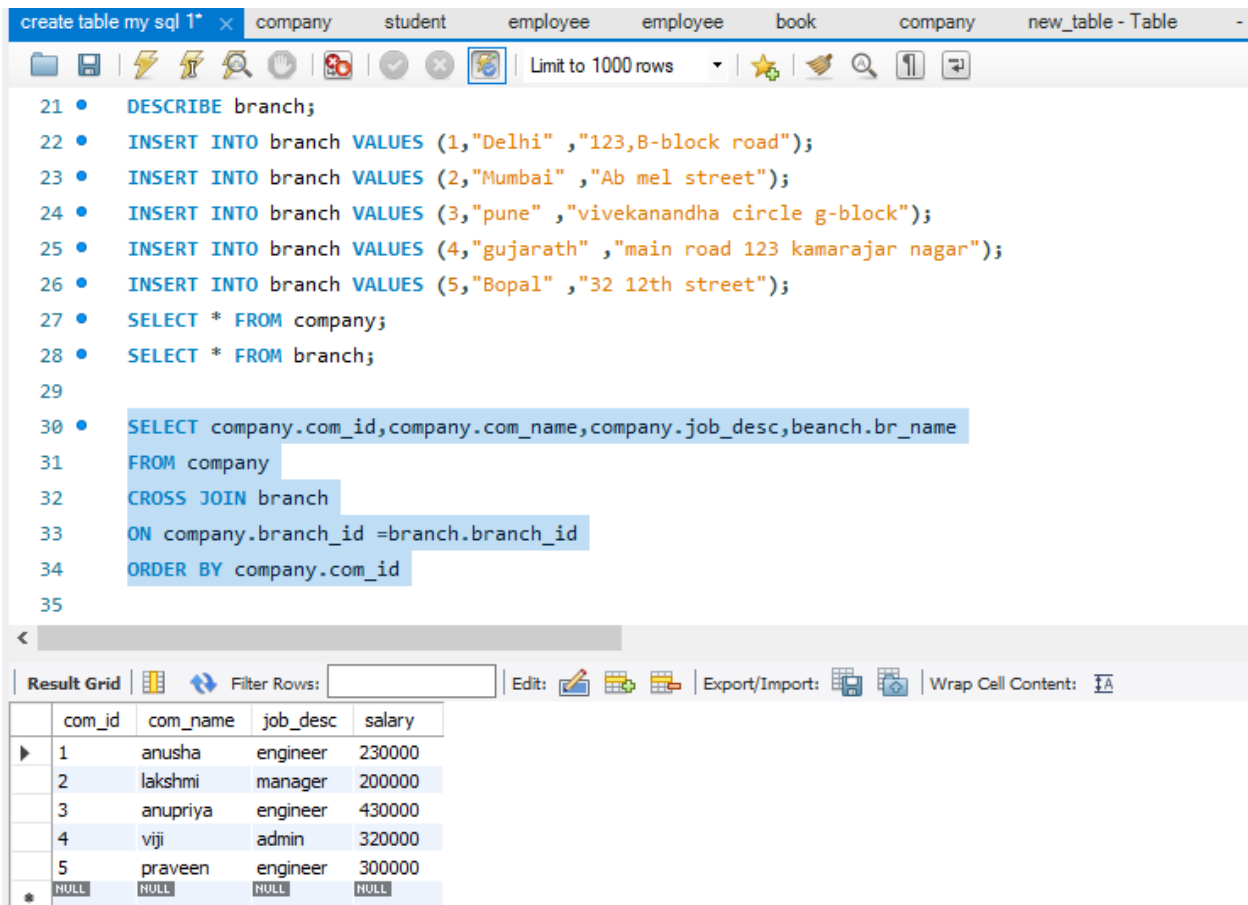
20 );
21 • DESCRIBE branch;
22 • INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
23 • INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
24 • INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
25 • INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
26 • INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
27 • SELECT * FROM company;
28 • SELECT * FROM branch;
29
30 • SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
31 FROM company
32 RIGHT JOIN branch
33 ON company.branch_id =branch.branch_id
34 ORDER BY company.com_id

```

The result grid displays the output of the SELECT statement, showing a table with 5 rows of employee data:

	com_id	com_name	job_desc	salary
▶	1	anusha	engineer	230000
	2	lakshmi	manager	200000
	3	anupriya	engineer	430000
	4	viji	admin	320000
	5	praveen	engineer	300000
*	NULL	NULL	NULL	NULL

## Cross Join :



The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
21 • DESCRIBE branch;
22 • INSERT INTO branch VALUES (1,"Delhi" ,"123,B-block road");
23 • INSERT INTO branch VALUES (2,"Mumbai" ,"Ab mel street");
24 • INSERT INTO branch VALUES (3,"pune" ,"vivekanandha circle g-block");
25 • INSERT INTO branch VALUES (4,"gujarath" ,"main road 123 kamarajar nagar");
26 • INSERT INTO branch VALUES (5,"Bopal" ,"32 12th street");
27 • SELECT * FROM company;
28 • SELECT * FROM branch;
29
30 • SELECT company.com_id,company.com_name,company.job_desc,branch.br_name
31 FROM company
32 CROSS JOIN branch
33 ON company.branch_id =branch.branch_id
34 ORDER BY company.com_id
35
```

The result grid displays the output of the query, showing a cross join between the 'company' and 'branch' tables. The columns are 'com\_id', 'com\_name', 'job\_desc', and 'salary'. The data is as follows:

	com_id	com_name	job_desc	salary
▶	1	anusha	engineer	230000
	2	lakshmi	manager	200000
	3	anupriya	engineer	430000
	4	viji	admin	320000
	5	praveen	engineer	300000
*	NULL	NULL	NULL	NULL

## Create,update,delete commands in mongodb:

## Update :

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'PracticeDB' with collections 'Employee', 'SoftwareEngineers', 'Sample', 'faculty', and 'student'. The 'student' collection is selected. The main panel shows the 'Sample.student' document with fields: '\_id: ObjectId('647457fd2d9b683aede954e6')', 'studentName: "xyz"', and 'studentaddress: "blg"'. Below this, the 'MONGOSH' terminal shows the following commands and results:

```
> db.student.updateOne({studentName:"abc"},{$set:{studentaddress:"chennai"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.student.deleteOne({studentName:"abc"})
< {
  acknowledged: true,
```

## Create :

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'PracticeDB' with collections 'Employee', 'SoftwareEngineers', 'Sample', 'faculty', and 'student'. The 'student' collection is selected. The main panel shows the 'Sample.student' document with fields: '\_id: ObjectId('647457fd2d9b683aede954e6')', 'studentName: "xyz"', and 'studentaddress: "blg"'. Below this, the 'MONGOSH' terminal shows the following commands and results:

```
> db.student.insertOne({'studentName':'abc'})
< {
  acknowledged: true,
  insertedId: ObjectId("6474571a2d9b683aede994e7")
}
> db.student.countDocuments()
< 1
> db.student.insertOne({'studentName':'xyz','age':20})
< {
  acknowledged: true,
  insertedId: ObjectId("647457fd2d9b683aede994e8")
}
```

## Delete:

The screenshot displays the MongoDB Compass web interface. On the left sidebar, the 'Databases' section is expanded, showing a tree structure with 'PracticeDB' containing 'Employee' and 'SoftwareEngineers', and 'Sample' containing 'faculty' and 'student'. The 'student' collection is selected and highlighted in green. The main panel is titled 'Sample.student' and has tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The 'Documents' tab is active, showing a single document with the following fields: `_id` (ObjectId), `studentName` (xyz), and `studentaddress` (blg). Below the document, there are buttons for 'ADD DATA' and 'EXPORT DATA'. At the bottom, a dark terminal window shows the execution of a MongoDB command: `> db.student.deleteOne({studentName:"abc"})`. The output shows the document was successfully deleted, with `acknowledged: true` and `deletedCount: 1`.

```
>_MONGOSH
{
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.student.deleteOne({studentName:"abc"})
< {
  acknowledged: true,
  deletedCount: 1
}
Sample > |
```