Queries with Tables & Constraints

1

```
CREATE TABLE People (
   PID INT Primary Key,
   LastName VARCHAR(100),
   FirstName VARCHAR(100),
   Address VARCHAR(100),
   City VARCHAR(100)
);

SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'People'
ORDER BY column_name;
```

2

```
CREATE TABLE Patients (
   Patient_id INT Primary Key,
   Patient_title CHAR(100),
   Patient_name CHAR(100),
   admit_date DATE
);


SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'Patients'
ORDER BY column_name;
```

3

```
CREATE TABLE customer (
   id INT PRIMARY KEY NOT NULL,
   name VARCHAR(50) NOT NULL,
   city VARCHAR(50) NOT NULL
);

DESC customer;

CREATE TABLE contacts (
   ID INT PRIMARY KEY,
   Customer_Id INT,
   Customer_Info VARCHAR(50) NOT NULL,
   Type VARCHAR(50) NOT NULL,
   FOREIGN KEY(Customer_Id) REFERENCES customer(ID)
);

DESC contacts;
```

4

```
CREATE TABLE users (
   id INT Primary Key,
   full_name VARCHAR(50),
   enabled CHAR(50),
   last_login DATE
);

SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'users';


CREATE TABLE addresses (
   user_id INT Primary Key,
   street VARCHAR(50),
   city VARCHAR(50),
   state VARCHAR(50),
   FOREIGN KEY (USER_ID) REFERENCES USERS(ID)
);

SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'addresses';
```

5

```
CREATE TABLE books(
 id INT PRIMARY KEY,
   title VARCHAR(100) NOT NULL,
   author VARCHAR(100) NOT NULL,
   published_date TIMESTAMP,
   isbn CHAR(12) UNIQUE
);
```

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'books';

CREATE TABLE reviews(
 id INT PRIMARY KEY,
   book_id INT NOT NULL,
    reviewer_name VARCHAR(255),
    content VARCHAR(255),
    rating INT,
    published_date TIMESTAMP,
    FOREIGN KEY(book_id) REFERENCES books(id)
);
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'reviews';
```

6

```
ALTER TABLE members
ADD cc_number VARCHAR(50);

   SELECT table_name, column_name, data_type
    FROM information_schema.columns
    WHERE table_name = 'members'
    ORDER BY column_name;
```

7

```
ALTER TABLE Bank
CHANGE COLUMN person_id Pid VARCHAR(50);

DESC Bank;
```

8

```
ALTER TABLE members
DROP COLUMN member_dob;

--DESC members;

 SELECT table_name, column_name, data_type
    FROM information_schema.columns
    WHERE table_name = 'members'
```

```
   ORDER BY column_name;
```

9

```
ALTER TABLE members
DROP COLUMN member_dob;

--DESC members;

 SELECT table_name, column_name, data_type
   FROM information_schema.columns
   WHERE table_name = 'members'
   ORDER BY column_name;
```

10

```
ALTER TABLE customer
DROP COLUMN last_login ;
ALTER TABLE customer
CHANGE COLUMN full_name customer_name VARCHAR(30);

DESC customer;
```

Modifying Data

1

```
INSERT INTO tutorials(tutorial_title, tutorial_author, submission_date) VALUES
('Learn MySQL', 'Balachandra Raju', '01-09-2021'); SELECT * FROM tutorials;
```

2

```
INSERT INTO Students (Roll_no, std_name,Age) VALUES (7,'Shantnu',21); SELECT *
FROM Students;
```

3

```
INSERT INTO BankAccount (Id, Name, cash_balance, Age) VALUES (1,'Ayush',500,21),
(2,NULL,1000,18), (4,'Muthu',NULL,25); SELECT * FROM BankAccount;
```

4

```
UPDATE BankAccount SET Age = 18 WHERE Id = 2; UPDATE BankAccount SET
cash_balance = 2000 WHERE Name = 'Muthu'; SELECT * FROM BankAccount;
```

5

```
UPDATE stud_data
SET Fname = 'Neelabh',
Age = '22'
WHERE roll_no = 17;
SELECT * FROm stud_data;
```

6

```
UPDATE Employee SET salary = 150000 WHERE position IN("SDE", "CA"); SELECT *
FROM Employee;
```

6.2

```
UPDATE Employee
SET salary = salary + 10000
WHERE age IN ('18','21','65');
SELECT * FROM Employee;
```

7

```
UPDATE Bank
SET branch = 'Noida'
WHERE id IN (2,3,4,5);
SELECT * FROM Bank;
```

8

```
DELETE FROM products
WHERE product_id IN ('596','700');
SELECT * FROM products;
```

9

```
DELETE FROM stud_data
WHERE Age<=20;
SELECT * FROM stud_data;
```

10

```
DELETE FROM Employee WHERE name LIKE '%an%' AND salary > 100000; SELECT *
FROM Employee;
```

11

```
REPLACE INTO Student
   SET id = 103,
        Name = 'Lawrence',
        gender = 'M',
        admission_date = "2008-11-27";
SELECT * FROM Student;
```

12

```
UPDATE Insurance SET Name = 'Kev', insurance_amount = 1750000, premium_date =
'2014-09-08' WHERE insurance_id = 305; SELECT * FROM Insurance;
```

13

```
REPLACE INTO cities (id,cname,population) VALUES('4','Phoenix','1768980'); SELECT *
FROM cities;
```

Joining Tables

1

```
SELECT E.EmpID, E.EmpFname, E.EmpLname, P.ProjectID, P.ProjectName
FROM Employee AS E
INNER JOIN Project AS P
ON E.EmpID = P.EmpID;
```

2

```
SELECT e.EmpID,e.EmailID,e.PhoneNo,c.clientFname,c.clientLname
FROM Employee e
INNER JOIN Client_d c ON e.EMPID = c.EMPID
WHERE e.city = 'Delhi' OR c.City='Kolkata';
```

3

```
SELECT
     p.ProjectName,c.ClientEmailID
FROM
     Project as p
 INNER JOIN
     Client_d as c

ON
    p.ClientID  = c.ClientID
```

```
WHERE
   p.ProjectStartDate > '2021-04-27'
ORDER BY
   c.Age DESc
```

4

```
SELECT *
FROM Supplier INNER JOIN SP ON Supplier.Sno = SP.Sno
INNER JOIN PRODUCT ON SP.Pno = Product.Pno;
```

5

```
SELECT Product.Pname, Product.Colour,SP.Qty
   FROM Supplier INNER JOIN SP ON Supplier.Sno = SP.Sno
   INNER JOIN Product ON Product.Pno = SP.Pno
   WHERE Supplier.Sname = 'sahil';
```

6

```
select p.colour,count(sp.qty) as nop
from product p
inner join SP sp
on p.Pno=sp.Pno
group by colour;
```

7

```
select Status,max(Qty) as m
from Supplier s
inner join SP sp
on s.Sno=sp.Sno
inner join Product p
on sp.Pno=p.Pno
group by status;
```

8

```
SELECT Supplier.Sname,Product.Pname,Supplier.Status
FROM Supplier INNER JOIN SP ON Supplier.Sno = SP.Sno
INNER JOIN PRODUCT ON SP.Pno = Product.Pno
order  by SP.Qty;
```

9

```
SELECT e.EmpFname,e.EmpLname,p.ProjectID,p.ProjectName
FROM Employee e
LEFT JOIN Project p ON e.EmpID = p.EmpID;
```

10

```
SELECT
   b.bname,b.bcity,p.pid,p.pdate,p.amount
FROM
   Buyer AS b
LEFT JOIN
   Product AS p
ON
   b.bid = p.bid
ORDER BY
   p.pdate ASC
```

11

```
SELECT
   b.bname,b.bcity,b.budget,s.sname,s.scity
FROM
   Buyer AS b
LEFT JOIN
   Seller AS s
ON
   b.sid = s.sid
WHERE
   b.budget < 3000
ORDER BY
   b.bid ASC
```

12

```
SELECT
   b.bname,b.bcity,p.pid,p.pdate,p.amount,s.sname,s.profit
FROM
   Buyer as b
LEFT JOIN
   Seller as s
ON
   b.sid = s.sid
LEFT JOIN
   Product as p
ON
   b.bid = p.bid;
```

13

```
SELECT p.ProjectID,p.Projectname,e.EmpFname,e.EmpLname,e.EmailID

FROM
    Employee  e
 RIGHT JOIN
    Project  p

ON
   e.EmpID = p.EmpID;
```

14

```
SELECT
   c.ClientID,c.ClientFname,c.ClientLname,c.ClientEmailID,p.ProjectID,p.ProjectName
FROM
   Client_d c
RIGHT JOIN
   Project p
ON
   p.ClientID = c.ClientID
WHERE
   c.Age BETWEEN 25 AND 35
```

15

```
SELECT Buyer.bname,Buyer.bcity,Buyer.budget,Seller.sname,Seller.scity
FROM Buyer
RIGHT JOIN Seller
ON Buyer.sid = Seller.sid
ORDER BY Seller.sid ASC;
```

16

```
SELECT b.bname,b.bcity,b.budget,s.sname,p.pid,p.pdate,p.amount FROM Buyer as b
RIGHT JOIN Seller as s on b.sid = s.sid
 RIGHT JOIN Product as p on b.bid=p.bid;
```

17

```
SELECT e.EmpFname,e.EmpLname,p.ProjectID
FROM  Employee AS e
CROSS JOIN Project AS p ;
```

18

```
SELECT b.bname,b.bcity,p.pid,p.pdate,p.amount FROM  Buyer b  left JOIN Product p
ON b.bid=p.bid where b.budget IS  not NULL
union
SELECT b.bname,b.bcity,p.pid,p.pdate,p.amount FROM  Buyer b  right JOIN Product p
ON b.bid=p.bid where b.budget IS  not NULL
```

19

```
SELECT
   p.ProjectID,p.ProjectName,e.EmpFname,e.EmpLname,e.EmailID
FROM
   Project p
LEFT JOIN
   Employee e
ON
     p.EmpID = e.EmpID
UNION

SELECT
   p.ProjectID,p.ProjectName,e.EmpFname,e.EmpLname,e.EmailID
FROM
   Project p
RIGHT JOIN
   Employee e
ON
     p.EmpID = e.EmpID
```