```sql
CREATE DATABASE school;
use school;
Create table temp (sid int primary key  , sname varchar(20), addredd varchar
(20));
alter table temp add column contact int;
alter table temp rename column sname to student_name;
select * from temp;
alter table temp rename column addredd to address;
alter table temp add column class varchar(10) after student_name;
alter table temp modify column contact bigint;
insert into temp value('1',"snehal","fymca","bhandup","123456");
insert into temp
values('2',"shraddha","fymca","bhandup","123456"),('3',"sanika","fymca","bhandup"
,"123456");
update temp set student_name="shreya" where student_name="Shraddha";
 set sql_safe_updates=0;
 SET SQL_SAFE_UPDATES=0;
 update temp set class="symca" where student_name="shreya";
 delete from temp where student_name="snehal";
 rollback;
commit;

create table temp2(id int , name varchar(20));
select * from temp2;
alter table temp2 add constraint pri_key primary key (id);
alter table temp2 add column sid int;
alter table temp2 add foreign key(sid)references temp(sid);
create user RMAN;
GRANT ALL privileges ON school.temp2 to 'RMAN'@'localhost';

CREATE USER 'RMAN'@'localhost' IDENTIFIED BY 'root';
GRANT ALL PRIVILEGES ON temp2.* TO 'RMAN'@'localhost';
FLUSH privileges;
grant insert on school.temp to 'RMAN'@'localhost';
revoke privileges on school.temp from 'RMAN';

create table horse (horse_name varchar(20) , age int not null, weight int ,
trainer varchar(20));
alter table horse add constraint prim_key primary key (horse_name);
create table Rides (horse_name varchar(20) references horse(horse_name) ,
race_name varchar(20) references Race(race_name), jockey varchar(20) , position
int ) ;

create table Race (race_name varchar(20) primary key , course varchar(20), length
int);
```

```sql
insert into horse values ("rani" ,10, 56,"snehal"), ("raja" ,20,
66,"snehal2"),("priya" ,30, 67,"snehal3"),("priti" ,55, 89,"snehal4"),("sameer"
,56, 34,"naina");
insert into Race values ("abc1","xyz1", 10),("abc2","xyz2", 20),("abc3","xyz3",
30),("abc4","xyz4", 40),("abc5","xyz5", 50);
insert into Rides values("rani","xyz1", "malhar1",1),("raja","xyz2",
"malhar2",2),("priya","xyz3", "malhar3",3),("priti","xyz4",
"malhar4",4),("sameer","xyz5", "malhar5",5);
insert into Race values ("abc6","mumbai", 67);
SELECT
    horse_name,
    SUBSTRING(horse_name, 2) AS extracted_substring,
    LENGTH(SUBSTRING(horse_name, 2)) AS length_of_substring
FROM
    horse;

    update Race set length=length+20 where course="mumbai";

    SELECT DISTINCT horse.horse_name
FROM horse
JOIN Rides ON horse.horse_name = Rides.horse_name
WHERE Rides.jockey = 'malhar1'
ORDER BY horse.horse_name;

-- CREATE DEFINER=`root`@`localhost` PROCEDURE `trainer_table`()
-- BEGIN
-- declare continue handler for 1146
-- select 'trainer table is not available' message;
-- select * from trainer;
-- select * from horse;
-- END

call trainer_table();

SELECT MAX(position) AS highest_position
FROM Rides r
JOIN Race ra ON r.race_name = ra.race_name
WHERE ra.course = 'mumbai' AND r.jockey = 'malhar1';


-- next practical

create database SBI;
```

```sql
CREATE table branch (branch_name varchar(20) primary key , branch_city
varchar(30), assets int);
create table customer (customer_name varchar (20) primary key , customer_street
varchar(20), customer_city varchar(20));
create table account (account_number int, customer_name varchar (20) references
customer (customer_name) ,branch_name varchar (20) references branch
(branch_name), balance int);

INSERT INTO branch (branch_name, branch_city, assets) VALUES
    ('B001', 'New York', 1000000),
    ('B002', 'London', 800000),
    ('B003', 'Tokyo', 1200000),
    ('B004', 'Paris', 950000),
    ('B005', 'Sydney', 700000);

INSERT INTO customer (customer_name, customer_street, customer_city) VALUES
    ('Alice', '123 Main St', 'New York'),
    ('Bob', '456 Oak St', 'London'),
    ('Charlie', '789 Pine St', 'Tokyo'),
    ('David', '101 Maple St', 'Paris'),
    ('Eve', '202 Cedar St', 'Sydney');

INSERT INTO account (account_number, customer_name, branch_name, balance) VALUES
    (1, 'Alice', 'B001', 5000),
    (2, 'Bob', 'B002', 8000),
    (3, 'Charlie', 'B003', 12000),
    (4, 'David', 'B004', 9500),
    (5, 'Eve', 'B005', 7000);

SELECT
    customer_name,
    customer_street,
    customer_city
FROM
    customer
WHERE
    customer_city IN (SELECT branch_city FROM branch);

    SELECT
    branch_name,
    AVG(balance) AS average_balance
FROM
    account
GROUP BY
    branch_name;
```

```sql
TRUNCATE TABLE branch;
TRUNCATE TABLE customer;
TRUNCATE TABLE account;


CREATE USER 'RMAN2'@'localhost' IDENTIFIED BY 'root';

call grant_revoke('grant');
call grant_revoke('revoke');
rollback;

grant rollback starts from here--
-- CREATE DEFINER=`root`@`localhost` PROCEDURE `random`()
-- BEGIN
-- CREATE USER 'RMAN3'@'localhost' IDENTIFIED BY 'root';
-- grant all privileges on SBI.account to 'RMAN3'@'localhost';
-- END

call random();

-- CREATE DEFINER=`root`@`localhost` PROCEDURE `random2`()
-- BEGIN
-- revoke  privileges on SBI.account from 'RMAN3'@'localhost';
-- END
call random2();


SELECT *
FROM
    customer c
LEFT JOIN
    account a ON c.customer_name = a.customer_name
LEFT JOIN
    branch b ON a.branch_name = b.branch_name;
```