SELECT CONCAT(FIRST\_NAME, ' ', LAST\_NAME) As Worker\_Name, Salary

FROM worker

WHERE Salary BETWEEN 50000 AND 100000;

SELECT worker\_ref\_id, COUNT(\*)

FROM bonus

GROUP BY worker\_ref\_id

HAVING COUNT(\*) > 1;

INSERT INTO WORKER (worker\_id,first\_name,salary,joining\_date,dept) VALUES(9,'Vivek',100000,'2015-05-29 09:00:00','HR');

select \* from worker;

Select RTRIM(FIRST\_NAME) from Worker;

select distinct dept,length(dept) from worker;

Select REPLACE(FIRST\_NAME,'a','ABCD') from Worker;

Select CONCAT(FIRST\_NAME, ' ', LAST\_NAME) AS 'COMPLETE\_NAME' from Worker;

Select \* from Worker order by FIRST\_NAME desc;

Select \* from Worker order by FIRST\_NAME ,DEPt desc;

Select \* from Worker where year(JOINING\_DATE) = 2014 and month(JOINING\_DATE) = 2 ;

select worker\_ref\_id,count(\*) from bonus group by worker\_ref\_id having count(\*) >1

insert into bonus

values(1,'2016-02-20 00:00:00',5000),

(2,'2016-06-11 00:00:00',3000),

(3,'2016-02-20 00:00:00',4000),

(1,'2016-02-20 00:00:00',4500),

(2,'2016-06-11 00:00:00',3500);

select \* from worker t1 inner join

(select worker\_ref\_id,sum(bonus\_amount) as total from bonus group by worker\_ref\_id) t2

on t1.worker\_id=t2.worker\_ref\_id;

select W.\*,WORKER\_TITLE

from worker W

inner join title T

on W.worker\_id=T.worker\_id ;

select \* from worker t1

left join

(select worker\_ref\_id,sum(bonus\_amount) as total from bonus group by worker\_ref\_id) t2

on t1.worker\_id=t2.worker\_ref\_id;

select w.worker\_id,First\_name,WORKER\_TITLE from worker w

inner join

title t

on w.worker\_id=t.worker\_id order by w.worker\_id;

#left join

select \* from worker t2

left join bonus t1

on t2.worker\_id=t1.worker\_ref\_id;

#demo of ifnull function

select t1.\*,ifnull(total,0) from worker t1

left join

(select worker\_ref\_id,sum(bonus\_amount) total from bonus group by worker\_ref\_id) t2

on t1.worker\_id=t2.worker\_ref\_id;

#2nd Maximun

select max(salary) from worker

where salary < (select max(salary) from worker);

select \* from org.worker;

select \* from org1.customer;

create table org.title\_clone as select \* from org.title;

create table org.bonus\_clone like table/as select \* from org.bonus where 1=0;

select \* from org.bonus where 1=0;

desc org.bonus\_clone;

select \* from org.bonus\_clone;

use org;

select \* from title

union all

select \* from title;

select \* from worker full join bonus on worker\_ref\_id=worker\_id;

2/4/2020 Queries

use org;

create table emp(eid int , ename varchar(20) not null,salary int not null,cid int, primary key(eid));

create table city(cid int primary key,cname varchar(15));

alter table emp add foreign key(cid) references city(cid);

desc emp;

insert into emp values (1,'A',10000,1);

select \* from emp;

insert into city values(1,'Udaipur');

select \* from city;

select \* from emp left join city on emp.cid=city.cid

union

select \* from emp right join city on emp.cid=city.cid;

select \* from city;

select distinct \* from worker t2

left join bonus t1

on t2.worker\_id=t1.worker\_ref\_id;

select \* from emp e1

where 6-1=(select count(\*) from emp e2

where e2.salary>e1.salary);

select \* from worker w1

where salary > (select avg(salary) from worker w2

where w1.dept=w2.dept);

select \* from worker;

use org;

select \*,

case

when cname ='Udaipur' then 'Rajasthan'

when cname ='Jaipur' then 'Rajasthan'

when cname ='Kota' then 'Rajasthan'

when cname ='Jodhpur' then 'Rajasthan'

else NULL

end as city\_name

from city;

Alter table city add column state varchar(30);

DESC CITY;

select CID,CNAME from city;

update city

set state=case cname

when 'Udaipur' then 'Rajasthan'

when 'Jaipur' then 'Rajasthan'

when 'Kota' then 'Rajasthan'

when 'Jodhpur' then 'Rajasthan'

else NULL

end;

UPDATE CITY

SET STATE=CASE CNAME

when 'NOIDA' then 'UP'

when 'MUMBAI' then 'MAHARASTRA'

END;

SELECT \* FROM CITY;

CREATE or replace VIEW view\_demo AS SELECT \* from worker where dept='hr';

select \* from view\_demo;

insert into view\_demo values(13,'Ajay','Sharma',100000,'2014-02-20 09:00:00','admin');

update worker

set last\_name='XYZ'

where worker\_id=11;

update view\_demo

set salary =10000

where first\_name='Monika';

insert into worker values(12,'Ram','Sharma',10000,'2014-02-20 09:00:00','HR');

delete from view\_demo where WORKER\_ID=12;

select \* from worker;

drop view View\_demo;

show full tables from mysql;

create table item(item\_id int auto\_increment primary key,item\_name varchar(20),stock int,price int);

alter table item auto\_increment=101;

insert into item (item\_name,stock,price) values('Laptop',100,30000);

insert into item (item\_name,stock,price) values('DESKTOP',100,20000);

insert into item (item\_name,stock,price) values('UPS',100,15000);

insert into item (item\_name,stock,price) values('Monitor',100,8000);

insert into item (item\_name,stock,price) values('CPU',100,16000);

insert into item (item\_name,stock,price) values('Mobile',100,12000);

select \* from item;

create table sales(sale\_id int,item\_id int,price int,quantity int,total int);

insert into sales values(1,101,30000,5,150000);

select \* from sales;

drop trigger t1;

create trigger t1

before insert on sales

for each row

set new.total=new.quantity\*new.price;

create trigger t2

after insert on sales

for each row

update item

set stock=stock-new.quantity where item\_id=new.item\_id;

insert into sales values(3,102,20000,50,0);

create table account(account\_id int auto\_increment primary key,name varchar(20),balance int);

alter table account auto\_increment=101;

select \* from account;

create table deposite(DID int auto\_increment primary key,accountid int,

depsoite\_amount int,foreign key (accountid) references account(account\_id));

select \* from deposite;

create trigger t3

after insert on deposite

for each row

update account

set balance=old.balance+new.deposite\_amount

where account\_id=new.accountid;

show tables from information\_schema;

select \* from information\_schema.KEY\_COLUMN\_USAGE

where table\_name='bonus';

use org;

alter table bonus drop foreign key bonus\_fk;

alter table bonus add constraint bonus\_fk foreign key(worker\_ref\_id)

references worker(worker\_id) on delete cascade;

alter table bonus drop foreign key bonus\_fk;

alter table bonus add constraint bonus\_fk foreign key(worker\_ref\_id)

references worker(worker\_id) on delete set NULL;

use org;

drop trigger checkage\_bu;

DELIMITER $$

CREATE TRIGGER checkage\_bi BEFORE insert ON account FOR EACH ROW

BEGIN

DECLARE dummy,B int;

SET B = 0;

if(NEW.balance >= 1000) THEN

SET B = 1;

END IF;

IF(B = 0) THEN

SELECT CONCAT('Cannot Insert This Because balance ',NEW.balance,' is Invalid')

INTO dummy FROM information\_schema.tables;

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER checkage\_bu BEFORE update ON account FOR EACH ROW

BEGIN

DECLARE dummy,b int;

SET b = 0;

if(NEW.balance >= 1000) THEN

SET b = 1;

END IF;

IF(b = 0) THEN

SELECT CONCAT('Cannot Insert This Because balance ',NEW.balance,' is Invalid')

INTO dummy FROM information\_schema.tables;

END IF;

END$$

DELIMITER ;

select \* from account;

insert into account values(103,'ABCH',1000);

update account set balance=4500 where account\_id=102;

select \* from account;

drop procedure sp\_fail;

DELIMITER $$

CREATE PROCEDURE `sp\_fail`()

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK; -- rollback any error in the transaction

END;

START TRANSACTION;

update account set balance=balance-500 where account\_id=102;

update account set balance=balance+500 where account\_id=101;

COMMIT; -- this will not be executed

END$$

DELIMITER ;

call sp\_fail();

MYSQL CMD-> mysql 'username'@'IP' identified by 'password'

check user list: select user from mysql.user;

CMD - >mysql -u TEST -p

use org;

#Grant select/insert/delete/update/create/drop/all privileges on org.\* to 'TEST'@'localhost';

flush privileges;

revoke select on org.\* from 'TEST'@'localhost';

select \* from account;

grant all privileges on \*.\* to 'TEST'@'localhost';

select \* from mysql.user where user in('root','TEST');

revoke all privileges on \*.\* from 'TEST'@'localhost';