-- create database bookdb;

use bookdb;

-- create table book(id int auto\_increment,name varchar(200) not null,author varchar (200) not null,price varchar (200) not null, primary key(id));

-- insert into book(name,author,price) values("abcd","cef","888");

insert into book(name,author,price) values("cd","f","8");

insert into book(name,author,price) values("acd","ce","88");

insert into book(name,author,price) values("app","hjn",9999);

show tables;

desc book;

use bookdb;

select name from book;

-- print all books,800

-- --select field name from table\_name where condition

select \* from book where price<800;

-- print all books available in range 80 to 900

select \* from book where price >80 and price < 900;

-- print all books exclude book abcd

select \* from book where name <> "abcd";

-- like

-- print all book name begining whith character a

select \* from book where name like "a%";

-- -- print all book name ending whith character d

select \* from book where name like "%d";

-- ordred by general sorting done by accending order(asc) desc for decending

select \* from book order by price desc;

select \* from book order by price asc;

-- max() min() sum() avg() count()

-- total number of book objects

select count(\*) from book;

select sum(price) from book;

select min(price) from book;

select max(price) from book;

select avg(price) from book;

-- sub queries

--- select name of largest price

select name from book where price=(select max(price) from book);

-- second largest one

select max(price) from book where price < (select max(price) from book);

select max(price) from book where price <> (select max(price) from book);

-- select name of second largest one

select name from book where price =( select max(price) from book where price < (select max(price) from book));

-- update

-- update table\_name set colomn\_name =new\_value where condition;

update book set author="fds" where id=2;

-- delete

-- delete from book where id=3;

-- create another table reviews and its relationship 1:m foregin key

use bookdb;

create table reviews(

id int auto\_increment primary key,

comment varchar(200) not null,

user varchar(200) not null,

book\_id int not null,

rating int not null,

foreign key (book\_id) references book(id) on delete cascade);

show tables;

-- drop table revie;

insert into reviews(comment,user,book\_id,rating) values("good","hari",1,2);

insert into reviews(comment,user,book\_id,rating) values("bad","gpi",2,2);

insert into reviews(comment,user,book\_id,rating) values("good","amm",1,2);

insert into reviews(comment,user,book\_id,rating) values("must read","hari",3,2);

insert into reviews(comment,user,book\_id,rating) values("good","appu",3,2);

insert into reviews(comment,user,book\_id,rating) values("worst","hari",2,2);

insert into reviews(comment,user,book\_id,rating) values("good","hari",1,5);

insert into reviews(comment,user,book\_id,rating) values("bad","gp",2,1);

insert into reviews(comment,user,book\_id,rating) values("ver good","paru",3,4);

select \* from reviews;

-- select perticular book review

select \* from reviews where book\_id=(select id from book where name ="abcd");

-- select perticular review using aliasing method

select \* from reviews as r ,book as b where r.book\_id=b.id and b.name="abcd";

-- select perticualar method to print other reviews expect a pertucular book

select \* from reviews as r ,book as b where r.book\_id=b.id and b.name <> "abcd";

-- find avg rating of abcd along with book name

select b.name,avg(r.rating) as avg\_rating from book as b,reviews as r where b.id=r.book\_id and b.name="abcd";

-- select highest rating in abcd

select b.name,max(r.rating) as max\_rating from book as b,reviews as r where b.id=r.book\_id and b.name="abcd";

-- sort it based on avg rating

select b.name,avg(r.rating) as avg\_rating from book as b,reviews as r where b.id=r.book\_id group by b.name order by avg\_rating desc;

-- inner join reviews ulladh mathram kittum

select b.name,r.comment from book as b inner join reviews as r on b.id=r.book\_id;

-- left join left side full plus right side coomon

select b.name,r.comment from book as b left join reviews as r on b.id=r.book\_id;

-- right join right side full and left side coomon

select b.name,r.comment from book as b right join reviews as r on b.id=r.book\_id;

create table entrol(id int auto\_increment not null primary key,sid int not null,cid int not null);

insert into entrol (sid,cid) values (2,1),(1,2),(2,3),(1,3),(3,2);

-- drop table entroll;

select distinct e1.sid from entrol as e1,entrol as e2 where e1.sid=e2.sid and e1.cid<>e2.cid;