```sql
-- write a query to create a table books with column author, subject , year and price
create table books(author varchar(20), subject varchar(20), year int, price int);
-- insert into the table
insert into books values('Mayank','database',2021, 350);
insert into books values('Rahul','alzebra',2011, 300);
insert into books values('durga','database',2010, 380);
-- write a query to create a table article with column author, articlename
create table article(author varchar(20), articlename varchar(20));
insert into article values('Mayank','my first');
insert into article values('Rahul','my second');
-- set operations:
-- write a query to find names of the authors who have either written a book or an article or both.

select author from books union select author from article;
select * from books ;
select * from article;
-- retrieve all the details with authors of either book or  article or both  including duplicate value
select author from books union all select author from article;

-- write a query to retrieve the authors who are written both books and article
select author from books intersect select author from article;
-- write a query to retrieve authors  who are written book only
select author from books minus select author from article --( set difference )
-- Cartesian product
select * from books, article;
select * from books cross join article;
-- if one table has 3  tuples and 2nd table has 5 tuples , how many tuples in case of cross join in the resultant table
a 12
B 15
c  8
d 10

select * from books, article where books.author=article.author;
operators:
arithmetic operator  + -,/, *
select (price*2) as multiprice from books;
Logical  operators : all, and , any ,  in , like, not , or , is null , unique
--like _, %
-- write a query to retrieve name of all the authors which starts with M
select author from books where author like 'M%';
-- write a query to retrieve name of all the authors which end with k

select author from books where author like '%k';
-- write a query retrieve author name containing "an"
select author from books where author like '%an%';
select author from books where author like '_a___';
-- in  used to compare values to a list of literals values that have been specified.
select * from books where price in(300, 350);
select * from books where price not in(300, 350);
insert into article(author) values('Ravi');
```

```
select * from article where articlename is NULL;
select * from article where articlename is not NULL;



select * from books;
select * from article;
delete from books where subject='Abiral';
insert into article(author) values('Ravi');
-- set operations:
-- write a query to find names of the authors who have either written a
book or an article or both.
select author from books union select author from article;
-Union all
-- retrieve all the details with authors of either book or  article or
both  including duplicate value
select author from books union all select author from article;
select * from books union select * from article ; ORA-01789: query block
has incorrect number of result columns
-- write a query to retrieve the authors who are written both books and
article
select author from books intersect select author from article;
-- write a query to retrieve authors  who are written book only
select author from books minus select author from article;
-- write a query to retrieve authors  who are written article only
select author from article minus select author from books;
select * from books, article;
select * from books cross join article;
select * from books, article where books.author=article.author and
books.author='Mayank';
select * from books natural join article;
alter table books rename column author to author1;
select * from books natural join article;-- no common column then act as
cross join or cartesian product
```