```
select * from customer_table ;

alter table customer_table add test varchar(255)

alter table customer_table drop test ;

alter table customer_table drop column test ;

alter table customer_table alter column age type varchar(255) ;

alter table customer_table rename column first_nam to first_name ;

alter table customer_table alter column cust_id set not null ;

insert into customer_table(first_name, last_name, age, email_id) values
('aa', 'bb', '25' , 'abc@xyz.com') ;

alter table customer_table alter column cust_id drop not null ;

DELETE FROM customer_table WHERE cust_id IS NULL;

alter table customer_table add constraint cust_id check (cust_id>0) ;

alter table customer_table add primary key (cust_id) ;

UPDATE Science_class SET science_marks = 45 WHERE enrollment_no = 1;


select * from customer ;

select * from product ;
select * from sales ;

select * from customer where city in ('Philadelphia', 'Seattle') ;

select * from customer where city = 'Philadelphia' OR city ='Seattle' ;

select * from customer where age between 20 and 30 ;

select * from customer where age >= 20 and age <= 30 ;

select * from customer where age not between 20 and 30 ;

/*
multi line comments
*/
--single line comments
/* Like Comments*/

/*
Here % and _ is called as wild card
% represents n number of charachters
_ represents one charachter
*/

select * from customer where customer_name like 'J%';

select * from customer where customer_name like '%Nelson%';

select * from customer where customer_name like '____ %';
/*customer has 4 words in firtst name and with 'n' number of words as second name */
```

```
select * from customer where city not like 'S%';


select * from customer where customer_name like 'G\%';
/*Here the \ is know as the escape charachter which treats % as a charachter and not as wild card*/


/*exercise*/
select * from customer order by customer_name asc ;


select distinct city from customer where region in ('Central', 'East');


select * from sales where sales between 100 and 500 ;


select distinct customer_name from customer where customer_name like '% ____';


/* oder by */


select * from customer where state = 'California' order by customer_name;


select * from customer where state = 'California' order by customer_name asc;


select * from customer where state = 'California' order by customer_name desc;


select * from customer order by city asc, customer_name desc;


select * from customer where state = 'California' order by city asc, customer_name desc;
/*
ORDER BY 2 DESC;
where 2 indicates the column number without specifying te name of the column
*/
```

```sql
select * from customer order by 2 asc;

select * from customer order by age desc;

/*Limiting number of return outputs*/

select * from customer where age >= 25 order by age desc limit 8;

select * from customer where age > 25 order by age asc limit 10;

/* exercise */

select * from sales limit 5 ;

select * from sales where discount > 0 order by discount desc ;

select * from sales where discount > 0 order by discount desc limit 10;

/* AS alias
provides second name for the column name or table name
*/

select customer_id as "Serial Number", customer_name as "Name", age as "Customer_age" from customer ;

select customer_id as "Serial Number", customer_name as Name, age as Customer_age from customer ;

/* "" is used for names with spaces or to retain the capitial initials
if not used then the names are named as small letters like Name will be changed to name
run this querry for better understanding
*/
```

```
/* COUNT */

select count(*) from sales ;

select count(order_line) as "Number Of Products Ordered", count (distinct order_id) as "Number of
Orders" from sales where customer_id = 'CG-12520';

/* SUM */

select sum(profit) as "Total Profit" from sales ;

select sum(quantity) as  "Total Quantity" from sales where product_id = 'FUR-TA-10000577' ;

/* AVERAGE */

select avg(age) as "Average Customer Age" from customer ;

select avg(sales * 0.10) as "Average Commision Value" from sales ;

/* MIN MAX */

select min(sales) as "Minimum Sales Value June" from sales where order_date between '2015-06-01'
and '2015-06-30' ;

select sales from sales where order_date between '2015-06-01' and '2015-06-30' order by sales asc ;

select max(sales) as "Maximun Sales Value June" from sales where order_date between '2015-06-01'
and '2015-06-30' ;



/* exercise */
```

```sql
select sum(sales) as "Total Sales" from sales ;

select count(distinct customer_id) from customer where age between 20 and 30 ;

select avg(age) as "Average Age of Customers In East Region" from customer where region in ('East')
;

select min(age) as "Minimum Age of Customer", max(age) as "Maximum Age Of Customer" from
customer where city in ('Philadelphia');

select min(age) as "Minimum Age of Customer", max(age) as "Maximum Age Of Customer" from
customer where city like ('P%a');

/* GROUP BY */

select region, count(customer_id) as "Customer Count" from customer group by region ;

select product_id, sum(quantity) as "Quantity Sold" from sales group by product_id order by
"Quantity Sold" desc ;

/* ALL QUEREY LEARNED SO FAR */

select customer_id, min(sales) as "Minimun Sales", max(sales) as "Maximun Sales", avg(sales) as
"Average Sales", sum(sales) as "Total Sales" from sales group by customer_id order by "Total Sales"
desc limit 5 ;

/* HAVING */

select region, count(customer_id) as "Customer Count" from customer group by region having
count(customer_id)>200;

select region, count(customer_id) as "Customer Count" from customer where customer_name like
'A%' group by region ;
```

```
select region, count(customer_id) as "Customer Count" from customer where customer_name like
'A%' group by region having count(customer_id) >15;


/* exercise */

select * from sales limit 1 ;


select * from customer limit 1 ;


select sum(sales) as "Total sales", sum(quantity) as "Total quantity", count(order_id) as "Number of
Orders", max(sales) as "Max Sales Value", min(sales) as "Min Sales Value" , avg(sales) as "Average
Sales Value" from sales ;


select product_id, count(product_id) as "List Of Product IDs" from sales group by product_id having
count(quantity) > 10 ;


/* CASE EXPRESSIONS */


SELECT *,

        CASE WHEN age < 30 THEN 'Young'

                WHEN age > 60 THEN 'Citizen'

                ELSE 'Middle Aged'

                END AS Age_Category

        FROM customer ;


/* JOINS */


/*Creating sales table of year 2015 */

/* Creating table with customre age between 20 and 30 */




/*Creating sales table of year 2015*/
```

```
Create table sales_2015 as select * from sales where ship_date between '2015-01-01' and '2015-12-31';

select count(*) from sales_2015; --2131

select count(distinct customer_id) from sales_2015;--578


/* Customers with age between 20 and 60 */

create table customer_20_60 as select * from customer where age between 20 and 60;

select count (*) from customer_20_60;--597


/* INNER JOIN || (A n B)

Gives the Intersection of two tables

*/


select

        a.order_line,

        a.product_id,

        a.customer_id,

        a.sales,

        b.customer_name,

        b.age

from sales_2015 as a

inner join customer_20_60 as b

on a.customer_id = b.customer_id

order by customer_id ;


/* LEFT JOIN || (A U B')


outputs all of A and intersection of A and B


*/
```

```
select
        a.order_line,
        a.product_id,
        a.customer_id,
        a.sales,
        b.customer_name,
        b.age
from sales_2015 as a
left join customer_20_60 as b
on a.customer_id = b.customer_id
order by customer_id ;

/* RIGHT JOIN || (A' U B)

outputs all of B and intersection of A and B
to get all the values of right table, always remember to select the common coulmn
name from the right table rather than from the left table in right join

*/

select
        a.order_line,
        a.product_id,
        b.customer_id, --selecting the common coulmn from the right join table
        a.sales,
        b.customer_name,
        b.age
from sales_2015 as a
right join customer_20_60 as b
on a.customer_id = b.customer_id
```

```
order by customer_id ;

/* FULL JOIN || (A U B) */

select
	a.order_line,
	a.product_id,
	a.customer_id,
	a.sales,
	b.customer_name,
	b.age,
	b.customer_id
from sales_2015 as a
full join customer_20_60 as b
on a.customer_id = b.customer_id
order by a.customer_id, b.customer_id ;

/* CROSS JOIN
creates cartesian product between two sets of data
*/

create table month_values (MM integer) ;
create table year_values (YYYY integer) ;

insert into month_values values (1), (2), (3),(4), (5), (6),(7), (8), (9), (10), (11),(12);
insert into year_values values (2011), (2012), (2013),(2014), (2015), (2016),(2017), (2018), (2019);

select * from month_values ;
select * from year_values ;

select a.YYYY , b.MM
```

```
        from year_values as a, month_values as b
        order by a.YYYY, b.MM ;


/* EXCEPT || (A n B')
output contains values of A exculded of the values common to A and B
*/

select customer_id
        from sales_2015
        except select customer_id from customer_20_60
        order by customer_id ;

/* UNION || (A U B) */

select customer_id
        from sales_2015
        union select customer_id from customer_20_60
        order by customer_id;

/* exercise */

select * from sales _2015 limit 1;

select * from customer_20_60 limit 1;

select b.state , sum(sales) as total_sales
from sales_2015 as a left join customer_20_60 as b
on
a.customer_id = b.customer_id
group by
```

```
b.state ;
```