# SQL Queries:

1) Write a query to create a table 'electricity\_connection\_type'.

**create table electricity\_connection\_type (id int(11) primary key, connection\_name varchar(20) not null);**

2) Write a query to create a table 'slab'.

mysql> create table slab (id int(11) primary key,

-> connection\_type\_id int(11) not null,

-> from\_unit int(11) not null,

-> to\_unit int(11) not null,

-> rate double not null,

-> foreign key(connection\_type\_id) references electricity\_connection\_type(id));

Query OK, 0 rows affected, 4 warnings (1.69 sec)

mysql> desc slab;

+--------------------+--------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------------+--------+------+-----+---------+-------+

| id | int | NO | PRI | NULL | |

| connection\_type\_id | int | NO | MUL | NULL | |

| from\_unit | int | NO | | NULL | |

| to\_unit | int | NO | | NULL | |

| rate | double | NO | | NULL | |

+--------------------+--------+------+-----+---------+-------+

5 rows in set (0.07 sec)

3) Write a query to create a table 'building\_type'.

**mysql> create table building\_type(id int(11) primary key,**

**-> name varchar(100) not null,**

**-> connection\_type\_id int(11) not null,**

**-> foreign key(connection\_type\_id) references electricity\_connection\_type(id));**

**Query OK, 0 rows affected, 2 warnings (1.43 sec)**

**mysql> desc building\_type;**

**+--------------------+--------------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+--------------------+--------------+------+-----+---------+-------+**

**| id | int | NO | PRI | NULL | |**

**| name | varchar(100) | NO | | NULL | |**

**| connection\_type\_id | int | NO | MUL | NULL | |**

**+--------------------+--------------+------+-----+---------+-------+**

**3 rows in set (0.08 sec)**

4) Write a query to create a table 'building'

**mysql> create table building (**

**-> id int(11) primary key,**

**-> owner\_name varchar(100) not null,**

**-> address varchar(100) not null,**

**-> building\_type\_id int(11) not null,**

**-> email\_address varchar(100),**

**-> foreign key(building\_type\_id) references building\_type(id));**

**Query OK, 0 rows affected, 2 warnings (0.96 sec)**

**mysql> desc building;**

**+------------------+--------------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+------------------+--------------+------+-----+---------+-------+**

**| id | int | NO | PRI | NULL | |**

**| owner\_name | varchar(100) | NO | | NULL | |**

**| address | varchar(100) | NO | | NULL | |**

**| building\_type\_id | int | NO | MUL | NULL | |**

**| email\_address | varchar(100) | YES | | NULL | |**

**+------------------+--------------+------+-----+---------+-------+**

**5 rows in set (0.08 sec)**

5) Write a query to alter the column 'owner\_name' to 'building\_owner\_name' in the table 'building'.

**mysql> alter table building rename column owner\_name to building\_owner\_name;**

**Query OK, 0 rows affected (0.43 sec)**

**Records: 0 Duplicates: 0 Warnings: 0**

**mysql> desc building;**

**+---------------------+--------------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+---------------------+--------------+------+-----+---------+-------+**

**| id | int | NO | PRI | NULL | |**

**| building\_owner\_name | varchar(100) | NO | | NULL | |**

**| address | varchar(100) | NO | | NULL | |**

**| building\_type\_id | int | NO | MUL | NULL | |**

**| email\_address | varchar(100) | YES | | NULL | |**

**+---------------------+--------------+------+-----+---------+-------+**

**5 rows in set (0.10 sec)**

6) Write a query to change the datatype of the column address in the table 'building' to varchar(255).

**mysql> alter table building modify column address varchar(255);**

**Query OK, 0 rows affected (1.91 sec)**

**Records: 0 Duplicates: 0 Warnings: 0**

**mysql> desc building;**

**+---------------------+--------------+------+-----+---------+-------+**

**| Field | Type | Null | Key | Default | Extra |**

**+---------------------+--------------+------+-----+---------+-------+**

**| id | int | NO | PRI | NULL | |**

**| building\_owner\_name | varchar(100) | NO | | NULL | |**

**| address | varchar(255) | YES | | NULL | |**

**| building\_type\_id | int | NO | MUL | NULL | |**

**| email\_address | varchar(100) | YES | | NULL | |**

**+---------------------+--------------+------+-----+---------+-------+**

**5 rows in set (0.07 sec)**

7) Write a query to create a constraint which allows only 'commercial' or 'home' connection\_name in the 'electricity\_connection\_type' table.Specify constraint name as 'check\_connection\_name'.

**alter table electricity\_connection\_type add constraint check\_connection\_type check ((connection\_name = 'home' )or(connection\_name ='commercial'));**

8) Write a query to rename table 'building' to 'building\_details'.

**mysql> alter table building rename to building\_details;**

**Query OK, 0 rows affected (0.66 sec)**

9) Write a query to drop table 'slab'.

**mysql> drop table slab;**

**Query OK, 0 rows affected (1.49 sec)**

10) Write a query to drop table 'building\_details'.

**mysql> drop table building\_details;**

**Query OK, 0 rows affected (0.33 sec)**

11) Write a query to insert any 2 records into the 'electricity\_connection\_type' table.

**insert into electricity\_connection\_type values(1, 'home');**

**insert into electricity\_connection\_type values(2, 'commercial');**

12) Write a query to insert any 3 records into the 'slab' table.

**insert into slab values (1, 1, 11, 10, 12.5);**

**insert into slab values (2, 1, 21, 11, 13.5);**

**insert into slab values (3, 2, 31, 33, 14.5);**

13) Write a query to insert any 5 records into the 'building\_type' table

**insert into building\_type values (1, 'arjun', 1);**

**insert into building\_type values (2, 'sevag', 2);**

**insert into building\_type values (3, 'kiran', 1);**

**insert into building\_type values (4, 'twiti', 2);**

**insert into building\_type values (5, 'sweki', 1);**

14) Write a query to change the from\_unit value from 0 to 1 in the 'slab' table.

**update slab set from\_unit = 1 where from\_unit= 0;**

15) Write a query to change the name 'Shopping Mall' to 'Mall' in the building\_type table.

**update building\_type set name = 'mall' where name like 'shopping mall';**

16) Write a query to delete the entire details of the table 'slab'.

**Delete from slab;**

**Queries:-**

1. Write a query to display the entire contents of the 'electricity\_connection\_type'.Display the records in ascending order based on their connection name.

**select \* from electricity\_connection\_type order by connection\_name asc;**

2. Write a query to display the entire contents of the building\_type table, sorted by name in ascending order.

**select \* from building\_type order by name asc;**

3. Write a query to display the entire contents of the 'building'.Display the records in ascending order based on owner name.

**create table building (**

**id int(11) primary key,owner\_name varchar(100) not null, address varchar(100) not null,building\_type\_id int(11) not null,email\_address varchar(100),foreign key(building\_type\_id) references building\_type(id));**

**insert into building values(1, 'arjun', 'chennai', 5, 'arjun@gmail.com');**

**insert into building values(2, 'kiran', 'chennai', 5, 'kiran@gmail.com');**

**insert into building values(3, 'tarun', 'chennai', 5, 'tarun@gmail.com');**

**insert into building values(4, 'praven', 'chennai', 5, 'praven@gmail.com');**

**select \* from building order by owner\_name asc;**

4. Write a query to display the entire contents of the 'electricity\_reading', Display the records in descending order based on 'total\_units'.

**select \* from electricity\_reading order by total\_units asc;**

5. Write a query to display all 'meter\_number' from meter table.

**select meter\_number from meter;**

6. Write a query to display the owner\_name and contact\_number of all building, Display the records in ascending order based on owner\_name.

**select owner\_name, contact\_number from building order by owner\_name asc;**

7. Write a query to display the total\_units, payable\_amount, fine\_amount of all bills , sorted by total\_units in descending order.

**select total\_units, payable\_amount, fine\_amount from bill order by total\_units desc;**

8. Write a query to display the entire contents of the slab table, sorted by from\_unit in ascending order.

**Select \* from slab order by from\_unit asc;**

9. Write a query to display the details of the building whose owner\_name is 'Nicholas'.

**select owner\_name from building where owner\_name like ‘Nicholas’;**

10. Write a query to display the details of all the bills whose 'total\_units' greater than 10000, sorted by total\_units in descending order.

**select \* from bill where total\_units>10000 order by total\_units desc;**

11. Write a query to display the details of all the bills with the due\_date on '2017-10-01', sorted by payable\_amount in descending order.

**select \* from bill where due\_date = ‘01/10/2017’ order by payable\_amount desc;**

12. Write a query to display the owner\_name, address and contact\_number of the buildings which does not have an email\_address, sorted by owner\_name in ascending order.

**select owner\_name, address , contact\_number from building where email\_address = null, order by owner\_name asc;**

13. Write a query to display the entire details of the building whose owner\_name starts with the letter 'M', sorted by owner\_name in ascending order.

**Select \* from building where owner\_name like ‘M%’;**

14. Write a query to display the entire details of the building whose building\_type\_id is 2, sorted by owner\_name in ascending order.

**Select \* from building where building\_type\_id =2 order by owner\_name asc;**

15. Write a query to display the details of the electricity\_reading whose total\_units per day is between 500 and 1000, sorted by total\_units in ascending order.

**Select \* electricity\_reading where total\_units between 500 and 1000;**

16. Write a query to display the meter\_id and total\_units of electricity\_reading whose 13th hour reading is lesser than the 14th hour reading, sorted by total\_units in descending order.

**Select meter\_id, total\_units from electricity\_reading h13 <h14 order by total\_units desc;**

**Online Course Electricity\_bill(Aggregate functions)**

Query’s:-

1. Write a query to display the average 8th hour unit consumption from electricity reading for all users.Give an alias name to the average 8th hour consumption as 'average\_8th\_hour\_consumption'.

**select avg(h8) as average\_8th\_hour\_consumption from electricity\_reading;**

1. Write a query to display the average unit of electricity  consumption for all the users.Give an alias name to the average units as 'average\_units\_of\_electricity'.

**select avg(h1+h2+h3+h4+h5+h6+h7+h8) as average\_8th\_hour\_consumption from electricity\_reading;**

1. Write a query to display the average payable amount from bill where payable\_amount is greater than 10000.Give an alias name to the average payable amount as 'average\_payable\_amount'.

**select avg(payable\_amount) as average\_payable\_amount from bill where payable\_amount >10000;**

1. Write a query to display the average of fine amount, whose payment date is on the year 2018.Give an alias name to the average fine amount as 'average\_fine\_amount'.

**select avg(fine\_amount) as average\_fine\_amount from bill where year =2018;**

1. Write a query to display the sum of payable amount from bill.Give an alias name to the sum of payable amount as 'sum\_payable\_amount'.

**select sum(payable\_amount) as sum\_payable\_amount from bill;**

1. Write a query to display the sum of payable amount with due date '2017-10-01'.Give an alias name to the sum of payable amount as 'sum\_payable\_amount'.

**select sum(payable\_amount) as sum\_payable\_amount from bill where due\_date = 2017/10/01;**

1. Write a query to display the minimum total units consumed by all users from electricity reading table. Give an alias name to the minimum total units as 'min\_total\_units'.

**select min(total\_units) as min\_total\_units from bill ;**

1. Write a query to display the second minimum fine amount from bill.Give an alias name to the second minimum fine amount as 'second\_min\_fine\_amount'.

**select min(fine\_amount)from bill where fine\_amount > (select min(fine\_amount)from bill);**

Note : Min amount is 360

1. Write a query to display the month and the minimum units of electricity consumed in each month. Give an alias name to the minimum units as 'minimum\_units'.

**select month(payment\_date), min(total\_units) as minimum\_units from bill;**

1. Write a query to display the standard deviation of the fine amount, whose payment date is on the year 2018.Give an alias name to the standard deviation to the fine amount  as 'standard\_deviation\_amount'.

Note:Use stddev()

**select stddev(fine\_amount) as standard\_deviation\_amount from bill where payment\_date between 2018/01/01 and 2018/12/31;**