

| Business Analyst Career Program (ChatGPT Included) |

||Gourav Kinker||gouravkinker111@gmail.com||

SQL [Major]

Q1) Create a table "STATION" to store information about weather observation stations.

```
Solution ->
create table STATION(
ID int primary key,
CITY char(20),
STATE char(2),
LAT_N int,
LONG_W int);
Input -
        -- Q1) Create a table "STATION" to store information about weather observation stations.
  5 • ⊖ create table STATION(
  6
        ID int primary key,
        CITY char(20),
        STATE char(2),
        LAT_N int,
       LONG_W int );
 10
Output -
 Result Grid
                                                          Edit:
                     Filter Rows:
                                         LONG_W
                                LAT N
                     NULL
                                        NULL
    NULL
             NULL
                              NULL
station 3 ×
```

Q2) Insert the following records into the table

Solution ->

```
insert into STATION (ID ,CITY ,STATE ,LAT_N , LONG_W)
Values (13 , 'PHOENIX' , 'AZ' , 33 , 112),
(44 , 'DENVER' , 'CO' ,40 ,105 ),
(66 , 'CARIBOU' , 'ME' , 47 , 68 );
```

Input -

```
12 -- Q2) Insert the following records into the table
13
14 • insert into STATION (ID ,CITY ,STATE ,LAT_N , LONG_W)
15 Values (13 , 'PHOENIX' , 'AZ' , 33 , 112),
16 (44 , 'DENVER' , 'CO' ,40 ,105 ),
17 (66 , 'CARIBOU' , 'ME' , 47 , 68 );
18
```

```
12
        -- Q2) Insert the following records into the table
 13
       insert into STATION (ID ,CITY ,STATE ,LAT_N , LONG_W)
 14 •
        Values (13 , 'PHOENIX' , 'AZ' , 33 , 112),
 15
        (44 , 'DENVER' , 'CO' ,40 ,105 ),
        (66 , 'CARIBOU' , 'ME' , 47 , 68 );
 17
                                      | Edit: 🕍 📆 | Export/Import: 📳 🐻 | Wrap Cell Content: 🏗
ID
        CITY
                 STATE LAT_N LONG_W
                       33
  13
        PHOENIX
                AZ
                             112
   44
        DENVER
                CO
                       40
                             105
        CARIBOU
  66
                ME
                       47
                             68
· NULL
                      NULL
                             NULL
```

Q3) Execute a query to look at table STATION in undefined order.

Solution ->

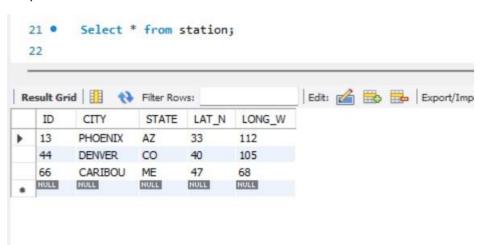
Select * from station;

```
Input –
```

```
-- Q3) Execute a query to look at table STATION in undefined order.

20
21 • Select * from station;

22
```



Q4) Execute a query to select Northern stations (Northern latitude>39.7)

Solution ->

Select * from STATION

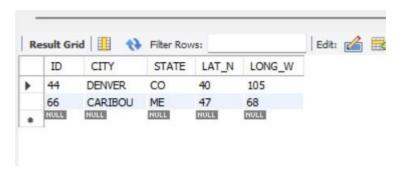
Where LAT_N > 39.7;

Input -

```
-- Q4) Execute a query to select Northern stations (Northernlatitude>39.7)

Select * from STATION

Where LAT_N > 39.7;
```



Q5) Create another table, 'STATS', to store normalized temperature and precipitation data.

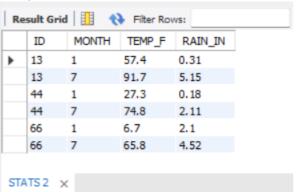
```
Solution->
create table STATS (
ID int,
MONTH int,
TEMP_F float check (TEMP_F >= -80 and TEMP_F <= 150),
RAIN_IN float check (RAIN_IN >= 0 and RAIN_IN <= 100)
);
Input -
        -- Q5) Create another table, 'STATS', to store normalized temperature and precipitation data.
 28
 29
 30 • ⊖ create table STATS (
 31
        ID int,
 32
        MONTH int,
        TEMP_F float check (TEMP_F >= -80 and TEMP_F <= 150),
 33
        RAIN_IN float check (RAIN_IN >= 0 and RAIN_IN <= 100)
       );
 35
 36
Output -
Expor
                 TEMP_F RAIN_IN
```

STATS 1 ×

Q6) Populate the table STATS with some statistics for January and July

Solution ->

```
INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_IN)
VALUES
(13, 1, 57.4, .31),
(13, 7, 91.7, 5.15),
(44, 1, 27.3, .18),
(44, 7, 74.8, 2.11),
(66, 1, 6.7, 2.1),
(66, 7, 65.8, 4.52);
Input -
 39 •
         INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_IN)
 40
         VALUES
 41
         (13, 1, 57.4, .31),
         (13, 7, 91.7, 5.15),
 42
         (44, 1, 27.3, .18),
 43
         (44, 7, 74.8, 2.11),
 45
         (66, 1, 6.7, 2.1),
         (66, 7, 65.8, 4.52);
 46
 47
```

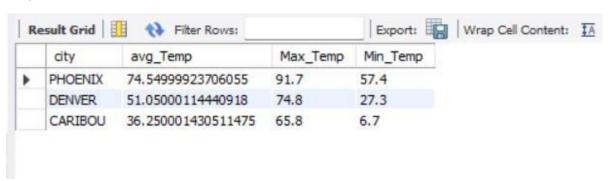


Q7) Execute a query to display temperature stats (from the STATS table) for each city (from the STATION table).

Solution ->

```
select st.city , avg(temp_f) As avg_Temp ,
max(temp_f) As Max_Temp,
min(temp_f) As Min_Temp from STATS As s
Join STATION As st
on s.id = st.id
Group By st.city;
```

Input-

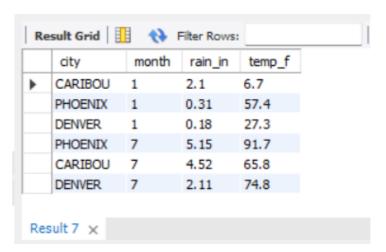


Q8) Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged. It should also show the corresponding cities.

```
select st.city , s.month , s.rain_in ,s.temp_f from stats as s
join station as st
on st.id = s.id
order by s.month , s.rain_in desc ;
```

input-

```
/*Q8) Execute a query to look at the table STATS, ordered by month and greatest rainfall,
with columns rearranged. It should also show the corresponding cities.*/
select st.city , s.month , s.rain_in ,s.temp_f from stats as s
join station as st
on st.id = s.id
order by s.month , s.rain_in desc ;
```

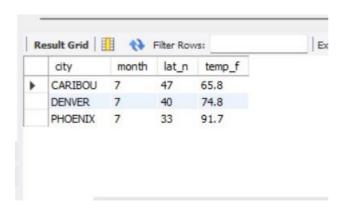


Q9) Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude.

Solution->

```
select st.city , s.month , st.lat_n , s.temp_f from stats as s
join station as st
on st.id = s.id
where s.month = 7
order by s.temp_f asc;
```

input-



Q10) Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.

Solution->

```
select st.city , max(s.temp_f) as Max_Temp, min(s.temp_f) as Min_Temp, Avg(s.rain_in) from stats As s join station as st on st.id =s.id group by st.city;
```

input-

```
-- Q10) Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.

75

76 • select st.city , max(s.temp_f) as Max_Temp, min(s.temp_f) as Min_Temp, Avg(s.rain_in) from stats As s

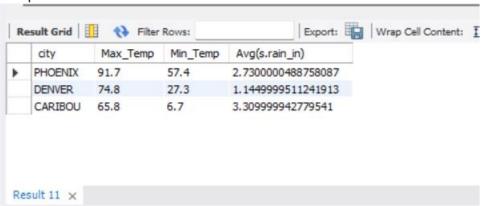
77    join station as st

78    on st.id =s.id

79    group by st.city;

80
```

output-



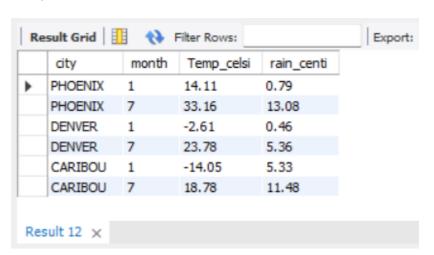
Q11) Execute a query to display each city's monthly temperature in Celsius and rainfall in Centimeter.

Solution->

```
select st.city , s.month ,round((s.temp_f -32 ) * 5/9 ,2) as Temp_celsi , round(s.rain_in * 2.54 , 2) as rain_centi from stats as s join station as st on st.id =s.id;
```

Input*

```
81 -- Q11) Execute a query to display each city's monthly temperature in Celsius and rainfall in Centimeter.
82
83
84 • select st.city , s.month ,round((s.temp_f -32 ) * 0.5555 ,2) as Temp_celsi ,
85     round(s.rain_in * 2.54 , 2) as rain_centi from stats as s
86     join station as st
87     on st.id =s.id;
88
```



Q12) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches slow.

Solution-<

```
update stats rain_in
set rain_in = rain_in + 0.01
```

input-

```
-- Q12) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches slow.

90

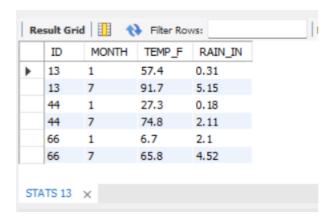
91 • update stats rain_in

92 set rain_in = rain_in + 0.01

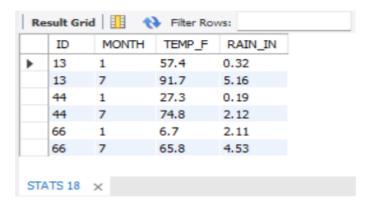
93
```

Output-

Before Add rain_in +0.001



After Add rain_in +0.001



Q13) Update Denver's July temperature reading as 74.9.

Solution->

```
update STATS set TEMP_F = 74.9
where id = 44 and month = 7;
```

input-

```
96 -- Q13) Update Denver's July temperature reading as 74.9.
97
98 • update STATS set TEMP_F = 74.9
99 where id = 44 and month = 7;
```

Output-

Before Update Denver's July TEMP_F = 74.8



After Update Denver's July TEMP_F = 74.9

