SQL Major Assignment

Name- Suyog Suhas Samant

Mobile- 9970292030

Email- suyogsamant62@gmail.com

1. **Create a table "Station" to store information about weather observation stations:**

ID	Number	Primary key
CITY	CHAR(20)	
STATE	CHAR(20)	
LAT_N	Number	
LONG_W	Number	

Ans:

Create table Station

(ID numeric primary key,

CITY char(20),

STATE char(20),

LAT_N numeric,

LONG_W numeric

);

```
create table Station
(ID numeric primary key,
CITY char(20),
STATE char(20),
LAT_N numeric,
LONG_W numeric
);
```

2. **Insert the following records into the table:**

ID	CITY	STATE	LAT_N	LONG_W
13	PHOENIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

Ans:

Insert into Station

values (13, 'PHOENIX', 'AZ', 33, 112),

(44, 'DENVER', 'CO', 40, 105),

(66, 'CARIBOU', 'ME', 47, 68)

;

```
Insert into Station
values (13, 'PHOENIX', 'AZ', 33, 112),
(44, 'DENVER', 'CO', 40, 105),
(66, 'CARIBOU', 'ME', 47, 68)
```

3. **Execute a query to look at table STATION in undefined order:**

Ans:

Select * from Table

```
Select * from Station
```

ID	CITY	STATE	LAT_N	LONG_W
13	PHOENIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

4. **Execute a query to select Northern stations (Northern latitude > 39.7):**

Ans:

Select * from Station

Where LAT_N > 39.7

```
Select * from Station
Where LAT_N > 39.7
```

ID	CITY	STATE	LAT_N	LONG_W
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

5. **Create another table, 'STATS', to store normalized temperature and precipitation data:**

Column	Data type	Remark
ID	Number	Must match some STATION table ID (so name and location will be known)
MONTH	Number	Range between 1 and 12
TEMP_F	Number	In Fahrenheit degrees, Range between -80 and 150
RAIN_I	Number	In inches, Range between 0 and 100

There will be no Duplicate ID and MONTH combination.

Ans:

Create table STATS

(ID int references station(ID),

MONTH int check (MONTH between 1 and 12),

TEMP_F real check (TEMP_F between -80 and 150),

RAIN_I real check (RAIN_I between 0 and 100),

primary key (id, month))

;

```
Create table STATS
(ID int references station(ID),
MONTH int check (MONTH between 1 and 12),
TEMP_F real check (TEMP_F between -80 and 150),
RAIN_I real check (RAIN_I between 0 and 100),
primary key (id, month))
;
```

6. Populate the table STATS with some statistics for January and July:

ID	MONTH	TEMP_F	RAIN_I
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

Ans:

Insert into STATS

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)

```
Insert into STATS
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)
```

7. **Execute a query to display temperature stats (from STATS table) for each city (from Station table):**

Ans:

Select STATS.ID, STATS.TEMP_F, Station.CITY

from STATS

join Station

on Station.ID = STATS.ID

;

```
Select STATS.ID, STATS.TEMP_F, Station.CITY
from STATS
join Station
on Station.ID = STATS.ID
;
```

ID	TEMP_F	CITY
13	57.4	PHOENIX
13	91.7	PHOENIX
44	27.3	DENVER
44	74.8	DENVER
66	6.7	CARIBOU
66	65.8	CARIBOU

8. **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:**

Ans:

Select STATS.MONTH, STATS.RAIN_I, Station.CITY

From STATS

Join STATION

ON STATS.ID = Station.ID

Order by MONTH, RAIN_I Desc

;

```
Select STATS.MONTH, STATS.RAIN_I, Station.CITY
From STATS
Join STATION
ON STATS.ID = Station.ID
Order by MONTH, RAIN_I Desc
;
```

MONTH	RAIN_I	CITY
1	2.1	CARIBOU
1	0.31	PHOENIX
1	0.18	DENVER
7	5.15	PHOENIX
7	4.52	CARIBOU
7	2.11	DENVER

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

Ans:

Select STATS.TEMP_F, Station.CITY, Station.LAT_N

From STATS

Join Station

Where MONTH = 7

And STATS.ID = Station.ID

Order by TEMP_F

;

```
Select STATS.TEMP_F, Station.CITY, Station.LAT_N
From STATS
Join Station
Where MONTH = 7
And STATS.ID = Station.ID
Order by TEMP_F
;
```

TEMP_F	CITY	LAT_N
65.8	CARIBOU	47
74.8	DENVER	40
91.7	PHOENIX	33

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

Ans:

Select Station.CITY,

MAX(STATS.TEMP_F) as MAX_Temperature, MIN(STATS.TEMP_F) as MIN_Temperature, round(avg(STATS.RAIN_I),2) as AVG_Rain

From STATS

Join STATION

on STATS.ID =Station.Id

Group by CITY

;

```
Select Station.CITY,
MAX(STATS.TEMP_F) as MAX_Temperature, MIN(STATS.TEMP_F) as MIN_Temperature, round(avg(STATS.RAIN_I),2) as AVG_Rain
From STATS
Join STATION
on STATS.ID =Station.Id
Group by CITY
;
```

CITY	MAX_Temperature	MIN_Temperature	AVG_Rain
PHOENIX	91.7	57.4	2.73
DENVER	74.8	27.3	1.14
CARIBOU	65.8	6.7	3.31

11. Execute a query to display each city's monthly temperature in Celcius and rainfall in Centimeter:

Ans:

Select Station.CITY, MONTH, Round(((temp_F-32)*5/9), 2) as Temp_celsius, Round((Rain_I-2.54), 2) as Rainfall_in_Centimeter

From STATS

Join STATION

On STATS.ID = Station.Id

;

```
Select Station.CITY, MONTH, Round(((temp_F-32)*5/9), 2) as Temp_celsius, Round((Rain_I-2.54), 2) as Rainfall_in_Centimeter
From STATS
Join STATION
On STATS.ID = Station.Id
;
```

CITY	MONTH	Temp_celsius	Rainfall_in_Centimeter
PHOENIX	1	14.11	-2.23
PHOENIX	7	33.17	2.61
DENVER	1	-2.61	-2.36
DENVER	7	23.78	-0.43
CARIBOU	1	-14.06	-0.44
CARIBOU	7	18.78	1.98

12. Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low:

Ans:

Update STATS set RAIN_I = round((RAIN_I + 0.01), 2);

Select * from STATS;

```
Update STATS set RAIN_I = round((RAIN_I + 0.01), 2);
Select * from STATS;
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.34
13	7	91.7	5.18
44	1	27.3	0.21
44	7	74.8	2.14
66	1	6.7	2.13
66	7	65.8	4.55

13. Update Denver's July temperature reading as 74.9:

Ans:

Update STATS set TEMP_F = 74.9

Where ID = 44

And MONTH = 7;

Select * from STATS;

```
Update STATS set TEMP_F = 74.9
Where ID = 44
And MONTH = 7;
Select * from STATS;
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.34
13	7	91.7	5.18
44	1	27.3	0.21
44	7	74.9	2.14
66	1	6.7	2.13
66	7	65.8	4.55