Vanshita Gupta
vanshitagupta13@gmail.com

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

ID	Number	Primary key
CITY	CHAR(20)	
STATE	CHAR(2)	
LAT_N	Number	
LONG_W	Number	

Solution) Query for table creation:

```
CREATE table STATION (ID INT(2) PRIMARY KEY, CITY CHAR(20), STATE CHAR(2), LAT_N INT(4), LONG_W INT(4));
SELECT * FROM STATION;
```

Result:

ID	CITY	STATE	LAT_N	LONG_W
NULL	NULL	NULL	NULL	NULL

Q2) 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

Solution)

Query to insert the record:

```
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)
```

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

Solution)

Query to look at table station in undefined order:

```
select * from station;
```

Result:

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

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

Solution)

Query to select Northern stations:

```
SELECT * FROM STATION
where LAT_N > 39.7;
```

Result:

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

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

Column	Data type	Remark
ID	Number	**ID** must match with some **ID** from the **STATION** table(so name & location will be known).
MONTH	Number	The range of months is between (**1 and 12**)
TEMP_F	Number	Temperature is in Fahrenheit degrees, Ranging between (**-80 and 150**)
RAIN_I	Number	Rain is in inches, Ranging between (**0 and 100**)

Solution:

Query to create table 'STATS':

```
CREATE TABLE stats (
    ID INT REFERENCES Station(ID),
    MONTH INT CHECK (MONTH BETWEEN 1 AND 12),
    TEMP_F float CHECK (TEMP_F BETWEEN -80 AND 150),
    RAIN_I float CHECK (RAIN_I BETWEEN 0 AND 100),
    PRIMARY KEY (ID, MONTH)
);
select * from stats;
```

Result:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ID	MONTH	TEMP_F	RAIN_I

Q6) 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

Solution)

Query to insert data in STATS table :

```
SELECT*FROM STATS
insert into STATS (ID, MONTH, TEMP_F, RAIN_I) 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)
```

Result:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.31
13	7	91.7	5.15
44	1	27.3	0.18
44	7	74.8	2.11
66	1	6.7	2.1
66	7	65.8	4.52

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

Solution)

Query to display temperature stats for each city from station table:

```
select ST.CITY, SA.TEMP_F
FROM station AS ST LEFT JOIN stats AS SA
ON ST.ID = SA.ID;
```

Result:

CITY	TEMP_F
PHOENIX	91.7
PHOENIX	57.4
DENVER	74.8
DENVER	27.3
CARIBOU	65.8
CARIBOU	6.7

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.

Solution)
Query to look at the table STATS, ordered by month and greatest rainfall:

```
select stats.*, CITY
from stats join station
on stats.ID = station.ID
order by MONTH, RAIN_I desc;
```

Result:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ID	MONTH	TEMP_F	RAIN_I	CITY
66	1	6.7	2.1	CARIBOU
13	1	57.4	0.31	PHOENIX
44	1	27.3	0.18	DENVER
13	7	91.7	5.15	PHOENIX
66	7	65.8	4.52	CARIBOU
44	7	74.8	2.11	DENVER

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

Solution)
Query to look at temperatures for July from table STATS, , lowest temperatures first, picking up city name and latitude.

assignment

Limit to 50000 rows

```
select CITY, LAT_N, TEMP_F, MONTH from station join stats
where Month = 7
and  station.ID = stats.ID
order by TEMP_F
```

Result:

CITY	LAT_N	TEMP_F	MONTH
CARIBOU	47	65.8	7
DENVER	40	74.8	7
PHOENIX	33	91.7	7

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

Solution)
Query to show MAX and MIN temperature and average rainfall for each city.

```
select CITY, max(TEMP_F) as 'MAX_TEMP', min(TEMP_F) as 'MIN_TEMP', avg(RAIN_I) as 'AVG_RAIN'
from station join stats
on station.ID = stats.ID
group by CITY
```

Result:

CITY	MAX_TEMP	MIN_TEMP	AVG_RAIN
PHOENIX	91.7	57.4	2.7300000488758087
DENVER	74.8	27.3	1.1449999511241913
CARIBOU	65.8	6.7	3.309999942779541

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

Solution)
Query to display each city's monthly temperature and rainfall

```
select CITY, MONTH, ROUND((TEMP_F - 32) * 5/9,4) AS "TEMP_Celsius",
 ROUND((RAIN_I * 0.39370079),4) AS "RAIN_Centimeters"
from station join stats
on station.ID = stats.ID
```

Result:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CITY	MONTH	TEMP_Celsius	RAIN_Centimeters
PHOENIX	1	14.1111	0.122
PHOENIX	7	33.1667	2.0276
DENVER	1	-2.6111	0.0709
DENVER	7	23.7778	0.8307
CARIBOU	1	-14.0556	0.8268
CARIBOU	7	18.7778	1.7795

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

Solution)
Query to update STATS table and add 0.01 to RAIN_I to compensate for faulty gauges:

```
--Query to add 0.01 to RAIN_I column:

update STATS
set RAIN_I = RAIN_I + 0.01 ;

-- Showing the results
select * from STATS
```

Result:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.32
13	7	91.7	5.16
44	1	27.3	0.19
44	7	74.9	2.12
66	1	6.7	2.11
66	7	65.8	4.53
NULL	NULL	NULL	NULL

STATS 29

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

Solution) Query Update Denver's July temperature reading

```
--STEP 1

CREATE VIEW UPDATE_STATS AS (SELECT CITY, MONTH, TEMP_F
from station join stats
WHERE MONTH = 7
AND station.ID = stats.ID)

--STEP 2
UPDATE UPDATE_STATS
SET TEMP_F = 74.9
WHERE CITY = 'DENVER';

--STEP 3

SELECT * FROM UPDATE_STATS;
```

Result:

CITY	MONTH	TEMP_F
PHOENIX	7	91.7
DENVER	7	74.9
CARIBOU	7	65.8