

Assignment - SQL [Major]

Name - Mohit Gupta

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

```
CREATE TABLE Station14 (
```

```
    ID_Number INT PRIMARY KEY,
```

```
    CITY CHAR(20),
```

```
    STATE CHAR(2),
```

```
    LAT_N NUMERIC,
```

```
    LONG_W NUMERIC
```

```
);
```

Schema

STATION

Syntax Help

Actions

View All Objects

Show All

Table Attributes

Columns

Indexes

Triggers

Constraints

Table Attributes

Table Name

STATION

Status

VALID

Temporary

No

Nested

No

Owner

SQL_WUIOBKYXSIHRUOXMMMLATGSH

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	ID_NUMBER	NUMBER	22		0	No		
2	CITY	CHAR	20			Yes	Byte	
3	STATE	CHAR	2			Yes	Byte	
4	LAT_N	NUMBER	22		0	Yes		
5	LONG_W	NUMBER	22		0	Yes		

Indexes

2. Insert the following records into the table:

insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)

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

insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)

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

insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)

Live SQL

Feedback Help amitkumar209727@gmail.com

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

Clear Find Actions Save Run

```
1 CREATE TABLE Station14 (  
2   ID_Number INT PRIMARY KEY,  
3   CITY CHAR(20),  
4   STATE CHAR(2),  
5   LAT_N NUMERIC,  
6   LONG_W NUMERIC  
7 );  
8  
9 insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)  
10 values(13,'PHOENIX','AZ',33,112);  
11 select * from station14  
12 insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)  
13 values(44,'DENVER','CO',40,105);  
14 insert into Station14(ID_Number,CITY,STATE,LAT_N,LONG_W)  
15 values(66,'CARIBOU','ME',47,68);
```

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

Download CSV

3 rows selected.

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

select * from station14

Live SQL

SQL Worksheet

```

1 CREATE TABLE Station4 (
2   ID_Number INT PRIMARY KEY,
3   CITY CHAR(20),
4   STATE CHAR(2),
5   LAT_N NUMBER(38,10),
6   LONG_W NUMBER(38,10)
7 );
8
9 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
10 values(15,'PHOENIX','AZ',33,112);
11 select * from station4
12
13 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
14 values(44,'DENVER','CO',40,105);
15
16 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
17 values(66,'CARIBOU','ME',47,68);
18
19 select * from station4
20
21 select * from station4
22 where LAT_N > 39.7;

```

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

Download CSV

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn
Built with ♥ using Oracle APEX - Privacy - Terms of Use

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

WHERE LAT_N > 39.7;

Live SQL

SQL Worksheet

```

1 CREATE TABLE Station4 (
2   ID_Number INT PRIMARY KEY,
3   CITY CHAR(20),
4   STATE CHAR(2),
5   LAT_N NUMBER(38,10),
6   LONG_W NUMBER(38,10)
7 );
8
9 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
10 values(15,'PHOENIX','AZ',33,112);
11 select * from station4
12
13 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
14 values(44,'DENVER','CO',40,105);
15
16 Insert into Station4(ID_Number,CITY,STATE,LAT_N,Long_W)
17 values(66,'CARIBOU','ME',47,68);
18
19 select * from station4
20
21 select * from station4
22 where LAT_N > 39.7;

```

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

Download CSV

2 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn
Built with ♥ using Oracle APEX - Privacy - Terms of Use

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

```
CREATE TABLE STATS (
```

```
  ID INT,
```

```
  MONTH INT CHECK (MONTH BETWEEN 1 AND 12),
```

```
  TEMP_F NUMERIC(5,2) CHECK (TEMP_F BETWEEN -80 AND 150),
```

```
  RAIN_I NUMERIC(5,2) CHECK (RAIN_I BETWEEN 0 AND 100),
```

```
  PRIMARY KEY (ID, MONTH)
```

```
);
```

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Schema \

STATS

Syntax HelpActionsView All Objects

Show AllTable AttributesColumnsIndexesTriggersConstraints

Table Attributes

Table Name	STATS
Status	VALID
Temporary	No
Nested	No
Owner	SQL_HYFOXUOLUBIFIAUEBXDPBYK

Columns

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	ID	NUMBER	22		0	No		
2	MONTH	NUMBER	22		0	No		
3	TEMP_F	NUMBER	22	5	2	Yes		
4	RAIN_I	NUMBER	22	5	2	Yes		

Indexes

Index Name	Index Type	Uniqueness	Status	Columns
SYS_C00113750086	NORMAL	UNIQUE	VALID	ID, MONTH

Triggers

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

```
INSERT INTO STATS VALUES(13,1,57.4,.31);
```

```
INSERT INTO STATS VALUES(13,7,91.7,5.15);
```

```
INSERT INTO STATS VALUES(44,1,27.3,.18);
```

```
INSERT INTO STATS VALUES(44,7,74.8,2.11);
```

```
INSERT INTO STATS VALUES(66,1,6.7,2.1);
```

INSERT INTO STATS VALUES(66,7,65.8,4.52);

The screenshot shows the 'Live SQL' web interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a SQL script. The script creates a table named 'STATS' with columns 'ID', 'MONTH', 'TEMP_F', and 'RAIN_I'. It then inserts several rows of data. Below the script, a table view displays the data that has been inserted. The table has 4 rows selected, as indicated by the 'Download CSV' button and the text '4 rows selected.' at the bottom.

```
1 CREATE TABLE STATS (
2   ID INT,
3   MONTH INT CHECK (MONTH BETWEEN 1 AND 12),
4   TEMP_F NUMBER(10,2) CHECK (TEMP_F BETWEEN -40 AND 100),
5   RAIN_I NUMBER(10,2) CHECK (RAIN_I BETWEEN 0 AND 100),
6   PRIMARY KEY (ID, MONTH)
7 );
8 INSERT INTO STATS VALUES(13,1,57.4,.31);
9 INSERT INTO STATS VALUES(13,7,51.7,5.15);
10 INSERT INTO STATS VALUES(44,1,57.3,.18);
11 INSERT INTO STATS VALUES(44,7,74.8,2.55);
12 INSERT INTO STATS VALUES(66,1,61.7,2.1);
13 INSERT INTO STATS VALUES(66,7,65.8,4.52);
14 SELECT * FROM STATS;
15
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.31
13	7	51.7	5.15
44	1	57.3	.18
44	7	74.8	2.55
66	1	61.7	2.1
66	7	65.8	4.52

Download CSV
4 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.3.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

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

SELECT * FROM STATION14, STATS

WHERE STATION14.ID = STATS.ID;

SELECT MONTH, ID, RAIN_I, TEMP_F

FROM STATS

ORDER BY MONTH, RAIN_I DESC;

The screenshot shows the 'Live SQL' web application. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a text editor with the following SQL query:

```
1 select * from STATS
2 select * from STATION14
3
4 SELECT stationid,city, avg(stats.temperature) AS avg_temp, min(stats.temperature) AS min_temp, max(stats.temperature) AS max_temp
5 FROM STATION14
6 JOIN STATS ON (stationid,station_id = STATS.station_id)
7 GROUP BY STATION14.city;
8
9 SELECT * FROM STATION14, STATS
10 WHERE STATION14.ID = STATS.ID;
11 SELECT MONTH, ID, RAIN_I, TEMP_F
12 FROM STATS
13 ORDER BY MONTH, RAIN_I DESC;
```

Below the editor, the results of the last query are displayed in a table:

MONTH	ID	RAIN_I	TEMP_F
5	66	2.3	6.7
5	13	-13	37.4
5	44	-18	17.3
7	13	9.33	39.7
7	66	4.32	65.8
7	44	2.33	74.8

Below the table is a 'Download CSV' button and a message '0 rows selected.' At the bottom of the interface, a footer contains version and copyright information: '2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 Database Documentation - Ask Tom - Dev Gym. Built with ❤️ using Oracle APEX. Privacy · Terms of Use.'

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.

SELECT LAT_N, CITY, RAIN_I, TEMP_F

FROM STATS, STATION14

WHERE MONTH = 7

Live SQL

Feedback Help amitkumar209727@gmail.com

Home SQL Worksheet Clear Find Actions Save Run

SQL Worksheet

```
4 SELECT station4.city, avg(stats.temperature) AS avg_temp, min(stats.temperature) AS min_temp, max(stats.temperature) AS max_temp
5 FROM station4
6 JOIN stats ON station4.station_id = stats.station_id
7 GROUP BY station4.city;
8
9 SELECT * FROM station4, stats
10 WHERE station4.id = stats.id;
11 SELECT month, id, rain_i, temp_f
12 FROM stats
13 ORDER BY month, rain_i DESC;
14
15
16 SELECT lat_n, city, rain_i, temp_f
17 FROM stats, station4
18 WHERE month = 7
19 AND stats.id = station4.id
20 ORDER BY temp_f;
21
```

lat_n	city	rain_i	temp_f
33	PHOENIX	5.15	91.7
33	PHOENIX	2.11	74.8
33	PHOENIX	4.52	69.8
47	CATERO	5.15	91.7
47	CATERO	2.11	74.8
47	CATERO	4.52	69.8
48	DENVER	5.15	91.7
48	DENVER	2.11	74.8
48	DENVER	4.52	69.8

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

```
SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), ID
```

```
FROM STATS
```

```
GROUP BY ID;
```

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Clear

Find

Actions

Save

```

8 SELECT * FROM STATION14, STATS
9 WHERE STATION14.ID = STATS.ID;
10
11 SELECT MONTH, ID, RAIN_I, TEMP_F
12 FROM STATS
13 ORDER BY MONTH, RAIN_I DESC;
14
15
16 SELECT LAT_N, CITY, RAIN_I, TEMP_F
17 FROM STATS, STATION14
18 WHERE MONTH = 7
19 AND STATS.ID = STATION14.ID
20 ORDER BY TEMP_F;
21
22 SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), ID
23 FROM STATS
24 GROUP BY ID;

```

MAX(TEMP_F)	MIN(TEMP_F)	AVG(RAIN_I)	ID
74.8	27.3	1.145	44
69.8	6.7	3.31	66
91.7	57.4	2.73	13

Download CSV

3 rows selected.

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

SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), City

FROM STATS,STATION14

GROUP BY CITY;

Live SQL

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

```
1 select * from STATION14
2 select * from Stats
3
4
5 SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), City
6 FROM STATS,STATION14
7 GROUP BY CITY;
8
```

MAX(TEMP_F)	MIN(TEMP_F)	AVG(RAIN_I)	CITY
91.7	6.7	2.395	CARIBOU
91.7	6.7	2.395	DENVER
91.7	6.7	2.395	PHOENIX

Download CSV

3 rows selected.

11. Execute a query to display each city's monthly temperature in

Celcius and rainfall in Centimeter.

Select ID,

MONTH,

(TEMP_F - 32) * 5 / 9,

$$\text{RAIN}_I * 0.3937$$

FROM STATS

[illegible]

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

UPDATE STATS

```
SET rain_I = RAIN_I + 0.01;
```

```
SELECT * FROM STATS
```

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActions

15 Select ID,

16 MONTH,

17 (TEMP_F - 32) * 5 / 9,

18 RAIN_I * 0.3937

19 FROM STATS

20

21

22 CREATE VIEW METRIC_STATS (ID, MONTH, TEMP_C, RAIN_C) AS

23 SELECT ID,

24 MONTH,

25 (TEMP_F - 32) * 5 / 9,

26 RAIN_I * 0.3937

27 FROM STATS;

28

29 UPDATE STATS

30 SET rain_i = RAIN_I + 0.01;

31 SELECT * FROM STATS

32

6 row(s) updated.

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.16
44	1	27.3	.19
44	7	74.8	2.12
66	1	6.7	2.11

13. Update Denver's July temperature reading as 74.9

UPDATE STATS

SET temp_F = 74.9

WHERE month = 7

AND ID = 44;

SELECT * FROM STATS

Live SQL

Feedback Help amitkumar209727@gmail.com

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Clear Find Actions Save Run

```
23 SELECT ID,  
24 MONTH,  
25 (TEMP_F - 32) * 5 / 9,  
26 RAIN_I * 0.3937  
27 FROM STATS;  
28  
29 UPDATE STATS  
30 SET RAIN_I = RAIN_I + 0.01;  
31 SELECT * FROM STATS  
32  
33  
34  
35 UPDATE STATS  
36 SET Temp_F = 74.9  
37 WHERE month = 7  
38 AND ID = 44;  
39 SELECT * FROM STATS
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.18
44	1	27.3	.19
44	7	74.9	2.12
68	1	6.7	2.11
68	7	65.5	4.53