

<i>Student Name</i>	Lin Rui
<i>Maynooth ID</i>	21124264

<i>Student Name</i>	林锐
<i>FZU ID</i>	832103316

CS130 Databases

Lab 9 Report

TASK 1: Given the following dataset called *Lab9_Station* – write an appropriate *CREATE* statement and then *INSERT* statements to represent this dataset in a Postgre SQL table. You are advised to use the column names as shown below. You should decide on the Primary Key, the Primary Key, the column data types, etc.

EIONetCode	Station	Status
IE0145A	Mayo Claremorris	Rural-Remote
IE0028A	Dublin Rathmines	Urban
IE0125A	Galway City	Suburban
IE0111A	Laois Emo Court	Rural-regional
IE0090A	Monaghan Kilkitt	Rural-regional
IE0147A	Kilkenny Seville Lodge	Suburban

SQL Language:

```
CREATE TABLE Lab9_Station(
    EIONetCode VARCHAR NOT NULL PRIMARY KEY,
    Station TEXT NOT NULL,
    Status TEXT NOT NULL
);
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0145A','Mayo Claremorris','Rural-Remote');
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0028A','Dublin Rathmines','Urban');
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0125A','Galway City','Suburban');
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0111A','Laois Emo Court','Rural-regional');
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0090A','Monaghan Kilkitt','Rural-regional');
INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0147A','Kilkenny Seville Lodge','Suburban');
SELECT * FROM Lab9_Station;
```

Running result: 6 rows affected.

Query Editor

Query History

```
1 CREATE TABLE Lab9_Station(  
2     EIONetCode VARCHAR NOT NULL PRIMARY KEY,  
3     Station TEXT NOT NULL,  
4     Status TEXT NOT NULL  
5 );  
6  
7 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0145A','Mayo Claremorris','Rural-Remote');  
8 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0028A','Dublin Rathmines','Urban');  
9 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0125A','Galway City','Suburban');  
10 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0111A','Laois Emo Court','Rural-regional');  
11 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0090A','Monaghan Kilkitt','Rural-regional');  
12 INSERT INTO Lab9_Station(EIONetCode,Station,Status) VALUES ('IE0147A','Kilkenny Seville Lodge','Suburban');  
13  
14 SELECT * FROM Lab9_Station;
```

Data Output

Explain

Notifications

Messages

eionetcode

character varying

station

text

status

text

1IE0145AMayo ClaremorrisRural-Remote

2IE0028ADublin RathminesUrban

3IE0125AGalway CitySuburban

4IE0111ALaois Emo CourtRural-regional

5IE0090AMonaghan KilkittRural-regional

6IE0147AKilkenny Seville LodgeSuburban

Successfully run. Total query runtime: 56 msec.
6 rows affected.

TASK 2: Given the following dataset called *Lab9_Pollutant* – write an appropriate *CREATE* statement and then *INSERT* statements to represent this dataset in a PostgreSQL table. You are advised to use the column names as shown below. You should decide on the Primary Key, the column data types, etc.

EEAPol	Pollutant	Notation
1	Sulphur dioxide (air)	SO2
10	Carbon monoxide (air)	CO
7	Ozone (air)	O3
8	Nitrogen dioxide (air)	NO2
5	Particulate matter < 10 (aersol)	PM10
6001	Particulate matter < 2.5 (aersol)	PM2.5

SQL Language:

```
CREATE TABLE Lab9_Pollutant(
    EEAPol VARCHAR NOT NULL PRIMARY KEY,
    Pollutant TEXT NOT NULL,
    Notation TEXT NOT NULL
);

INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('1','Sulphur dioxide(air)','SO2');
INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('10','Carbon monoxide(air)','CO');
INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('7','Ozone(air)','O3');
INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('8','Nitrogen dioxide(air)','NO2');
INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('5','Particulate matter<10(aersol)','PM10');
INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('6001','Particulate matter<2.5(aersol)','PM2.5');
SELECT * FROM Lab9_Pollutant;
```

Running result: 6 rows affected.

[Query Editor](#) [Query History](#)

```
1 CREATE TABLE Lab9_Pollutant(
2     EEAPol VARCHAR NOT NULL PRIMARY KEY,
3     Pollutant TEXT NOT NULL,
4     Notation TEXT NOT NULL
5 );
6
7 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('1','Sulphur dioxide(air)','SO2');
8 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('10','Carbon monoxide(air)','CO');
9 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('7','Ozone(air)','O3');
10 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('8','Nitrogen dioxide(air)','NO2');
11 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('5','Particulate matter<10(aersol)','PM10');
12 INSERT INTO Lab9_Pollutant(EEAPol,Pollutant,Notation) VALUES ('6001','Particulate matter<2.5(aersol)','PM2.5');
13
14 SELECT * FROM Lab9_Pollutant;
```

Data Output	Explain	Notifications	Messages
<div> <div>eeapol</div> <div>character varying</div> </div>		<div> <div>pollutant</div> <div>text</div> </div>	Successfully run. Total query runtime: 72 msec. 6 rows affected.
1	1	Sulphur dioxide(air)	
2	10	Carbon monoxide(air)	
3	7	Ozone(air)	
4	8	Nitrogen dioxide(air)	
5	5	Particulate matter<10(aersol)	
6	6001	Particulate matter<2.5(aersol)	

TASK 3: Each station can monitor multiple air pollutants. All stations monitor at least one pollutant. Dublin Rathmines station monitors all of the pollutants, stations with a status containing the term 'Rural' monitor all pollutants except SO2 and CO, while suburban stations monitor all pollutants except Ozone and Carbon Monoxide.

Write an appropriate CREATE statement and then INSERT statements to represent these relationships in a PostgreSQL table. You should call the table relationships in a PostgreSQL table. You should call the table Lab9_Monitors. This table should enforce Referential Integrity on all UPDATES and DELETES.

SQL Language:

```
CREATE TABLE Lab9_Monitors (  
    EIONetCode VARCHAR NOT NULL REFERENCES Lab9_Station (EIONetCode) ON DELETE CASCADE ON UPDATE  
    CASCADE,  
    EEAPoI VARCHAR NOT NULL REFERENCES Lab9_Pollutant(EEAPoI) ON DELETE CASCADE ON UPDATE  
    CASCADE  
);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',1);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',10);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',7);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0028A',6001);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0145A',7);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0145A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0145A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0145A',6001);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0111A',7);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0111A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0111A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0111A',6001);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0090A',7);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0090A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0090A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0090A',6001);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0125A',1);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0125A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0125A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0125A',6001);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0147A',1);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0147A',8);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0147A',5);  
INSERT INTO Lab9_Monitors(EIONetCode,EEAPoI) VALUES ('IE0147A',6001);  
SELECT * FROM Lab9_Monitors
```

Running result: 26 rows affected.

```

1 CREATE TABLE Lab9_Monitors (
2     EIONetCode VARCHAR NOT NULL REFERENCES Lab9_Station (EIONetCode) ON DELETE CASCADE ON UPDATE CASCADE,
3     EEAPoL VARCHAR NOT NULL REFERENCES Lab9_Pollutant(EEAPoL) ON DELETE CASCADE ON UPDATE CASCADE
4 );
5
6 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',1);
7 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',10);
8 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',7);
9 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',8);
10 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',5);
11 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0028A',6001);
12 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0145A',7);
13 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0145A',8);
14 INSERT INTO Lab9_Monitors(EIONetCode,EEAPoL) VALUES ('IE0145A',5);

```

Data Output Explain Notifications Messages

	eionetcode character varying	eeapol character varying
1	IE0028A	1
2	IE0028A	10
3	IE0028A	7
4	IE0028A	8
5	IE0028A	5

Successfully run. Total query runtime: 66 msec.
26 rows affected.

QUESTION 1: Write a JOIN query which shows all of the relations between the Lab9_Station and Lab9_Pollutant tables. You are asked to indicate how many rows are returned by this query.

SQL Language:

```

SELECT * FROM lab9_station AS a1,Lab9_pollutant AS a2,Lab9_monitors AS a3
WHERE (a1.EIONetCode=a3.EIONetCode) AND (a2.EEAPoL=a3.EEAPoL)

```

Running result: 26 rows affected

```

1 SELECT * FROM lab9_station AS a1,Lab9_pollutant AS a2,Lab9_monitors AS a3
2 WHERE (a1.EIONetCode=a3.EIONetCode) AND (a2.EEAPoL=a3.EEAPoL)

```

Messages

Successfully run. Total query runtime: 53 msec.
26 rows affected.

Data Output Explain Notifications

	eionetcode character varying	station text	status text	eeapol character varying	pollutant text	notation text	eionetcode character varying	eeapol character varying
1	IE0028A	Dublin Rathmines	Urban	1	Sulphur dioxide(air)	SO2	IE0028A	1
2	IE0028A	Dublin Rathmines	Urban	10	Carbon monoxide(air)	CO	IE0028A	10
3	IE0028A	Dublin Rathmines	Urban	7	Ozone(air)	O3	IE0028A	7
4	IE0028A	Dublin Rathmines	Urban	8	Nitrogen dioxide(air)	NO2	IE0028A	8
5	IE0028A	Dublin Rathmines	Urban	5	Particulate matter<10(aersol)	PM10	IE0028A	5

QUESTION 2: Write an ALTER table statement to change the structure of the Lab9_Monitors table to include a text column called Frequency. Do not include referential integrity constraints for this new column. Write an appropriate number of UPDATE statements to assign the monitoring and reporting frequency for each relationship in the Lab9_Monitors table, based on the information provided.

SQL Language:

```
ALTER TABLE Lab9_Monitors ADD COLUMN Frequency TEXT;
```

```
UPDATE Lab9_Monitors
```

```
SET Frequency = 'Hourly'
```

```
WHERE EEAPol = '1' OR EEAPol = '10' OR EEAPol = '7' OR EEAPol = '8';
```

```
UPDATE Lab9_Monitors
```

```
SET Frequency = 'Daily'
```

```
WHERE EEAPol = '5' OR EEAPol = '6001';
```

```
SELECT * FROM Lab9_Monitors
```

Running result: 26 rows affected

[Query Editor](#) [Query History](#)

```

1 ALTER TABLE Lab9_Monitors ADD COLUMN Frequency TEXT;
2
3 UPDATE Lab9_Monitors
4 SET Frequency = 'Hourly'
5 WHERE EEAPol = '1' OR EEAPol = '10' OR EEAPol = '7' OR EEAPol = '8';
6
7 UPDATE Lab9_Monitors
8 SET Frequency = 'Daily'
9 WHERE EEAPol = '5' OR EEAPol = '6001';
10
11 SELECT * FROM Lab9_Monitors

```

[Data Output](#) [Explain](#) [Notifications](#)

[Messages](#)

	eionetcode character varying	eeapol character varying	frequency text
1	IE0028A	1	Hourly
2	IE0028A	10	Hourly
3	IE0028A	7	Hourly
4	IE0028A	8	Hourly
5	IE0145A	7	Hourly

Successfully run. Total query runtime: 50 msec.
26 rows affected.