Nikki Jackson-Hebson
GEOG 574: Lab 4 – Data Retrieval and Manipulation Using SQL

Section 2.1 Simple Selection

```
1. SELECT * FROM COUNTRY
   WHERE REGION LIKE 'Ocean%'

2. SELECT * FROM COUNTRY
   WHERE AREA = '0'

3. SELECT DISTINCT region
   FROM country;

4. SELECT name, population
   FROM country
   WHERE population > '10000000'

5. SELECT name, region, population
   FROM country
   WHERE population > 10000000
   AND region LIKE ('%Asia%')
   OR region LIKE ('%Europe%')
   AND population > 10000000

   SELECT name, region, population
   FROM country
   WHERE REGION IN ('Asia', 'Europe')
   AND population > 10000000
```

Section 2.2 Summarization and Aggregation

```
1. SELECT name, area
   FROM country
   WHERE area BETWEEN 500000 AND 1000000
```

21 countries returned from SQL query above.

```
2. SELECT region, count(name), sum(area), avg(area)
   FROM country
  GROUP BY region
```

```
3.  SELECT region, count(*)
    FROM country
    WHERE population > 10000000
    GROUP BY region
    HAVING count(*) >= 5
```

```
4.  SELECT name, population, area, gdp, population/area AS popdensity
    FROM country
    WHERE area != 0
    ORDER by gdp
```

```
5.  SELECT region, avg(population) as pop_AVG, avg(gdp) as gdp_AVG, avg(area) as area_AVG,
    avg(population)/avg(area) as avg_popdensity
    FROM country
    WHERE POPULATION > 10000000
    GROUP by region
    ORDER by gdp_avg DESC
```

Section 2.3 Nested Queries

```
1.  SELECT name, gdp
    FROM country
    WHERE gdp > (
        SELECT sum(gdp)
         FROM country
         WHERE region = 'Africa');
```

```
2.  SELECT region, SUM(gdp) AS gdp_sum
    FROM country
    GROUP BY REGION
    HAVING SUM(gdp) >= ALL (SELECT SUM(gdp)
                            FROM country
                            GROUP BY REGION);
```

```
3.  SELECT name
    FROM country
    WHERE population >
    (
    SELECT MAX(population)
    FROM country
    WHERE region = 'North America'
    ) AND region LIKE '%Asia%'
```

Section 2.4 Insert, Update, and Delete

1.
   ```
   SELECT *
   FROM country
   WHERE area != 0 AND gdp = 0 AND population = 0

   DELETE FROM country
   WHERE area != 0 AND gdp = 0 AND population = 0
   ```

2.
   ```
   SELECT region
   FROM country
   WHERE name = 'Mauritius'

   UPDATE country
   SET region = 'Africa'
   WHERE name = 'Mauritius'
   ```

3.
   ```
   SELECT gdp
   FROM country
   WHERE name = 'United States'

   UPDATE country
   SET gdp = gdp * POWER(1.05,24)
   WHERE name = 'United States'
   ```

4.
   ```
   UPDATE country
   SET population = population + (SELECT population FROM country
                                  WHERE name = 'Hong Kong'),
           area =   area + (SELECT area FROM country
                            WHERE name = 'Hong Kong'),
           gdp = gdp + (SELECT gdp FROM country
                        WHERE name = 'Hong Kong')
   WHERE name = 'China'

   DELETE from country
   WHERE name = 'Hong Kong'
   ```

5.

```
INSERT INTO country VALUES
(
'Korea',
              'Asia',
(SELECT SUM(area) FROM country
                                  WHERE name = 'Korea, North' OR name = 'Korea, South'),
(SELECT SUM(population) FROM country
                                  WHERE name = 'Korea, North' OR name = 'Korea, South'),
(SELECT SUM(gdp) FROM country
                                  WHERE name = 'Korea, North' OR name = 'Korea, South')
)

DELETE from country
WHERE name = 'Korea, South' OR name = 'Korea, North'
```

2.5 Overall Practice

1.

```
CREATE TABLE Planet
  (Name varchar(10),
    NumMoons int,
   Type varchar(5),
   LengthOfYear decimal(5,2),
   primary key (Name)
   );

INSERT INTO Planet VALUES ('Mercury',0,'Rocky',0.24);
INSERT INTO Planet VALUES ('Venus',0,'Rocky',0.62);
INSERT INTO Planet VALUES ('Earth',1,'Rocky',1);
INSERT INTO Planet VALUES ('Mars',2,'Rocky',1.88);
INSERT INTO Planet VALUES ('Jupiter',16,'Gas',11.86);
INSERT INTO Planet VALUES ('Saturn',18,'Gas',29.46);
INSERT INTO Planet VALUES ('Pluto',1,'Rocky',247.7);

COMMIT;
```

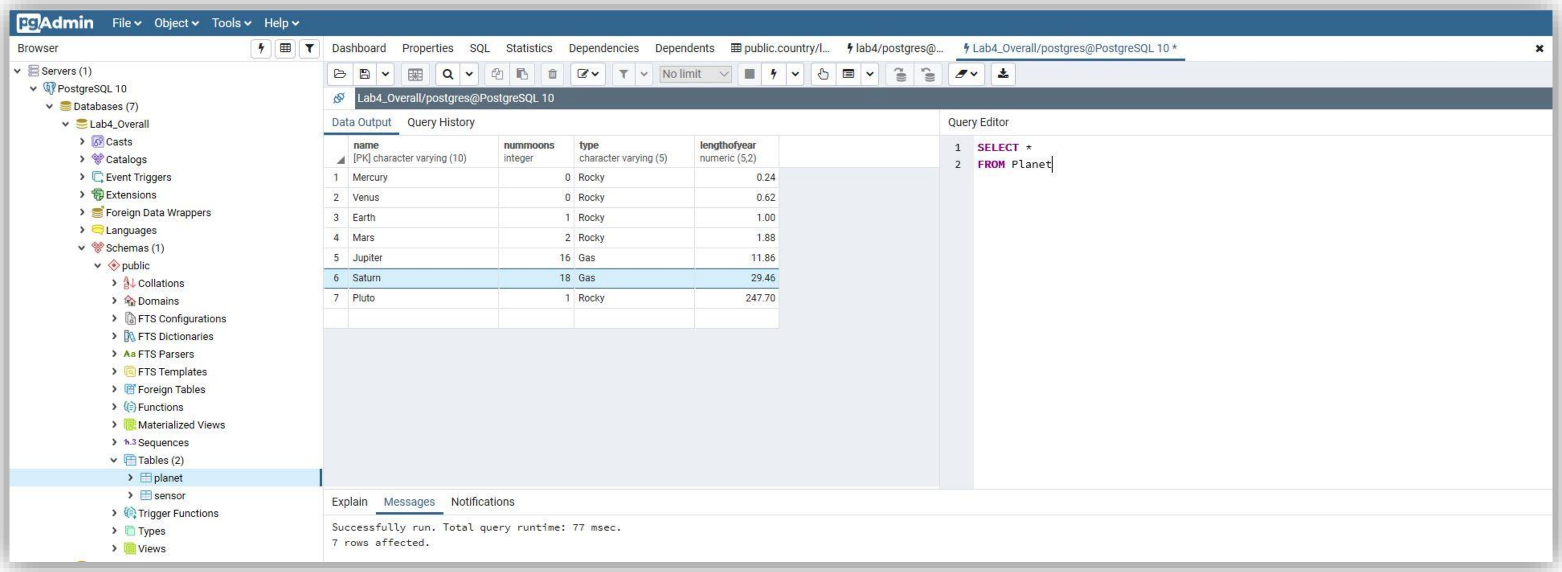
2.

```
CREATE TABLE Sensor
 (name varchar(10),
 year int,
 dest varchar(10),
primary key (name)
 );

INSERT INTO Sensor VALUES ('Pioneer 5',1960,'sun');
INSERT INTO Sensor VALUES ('Mariner 10',1974,'Mercury');
INSERT INTO Sensor VALUES ('Messenger',2008,'Mercury');
INSERT INTO Sensor VALUES ('Zond',1964,'Venus');
INSERT INTO Sensor VALUES ('Viking',1976,'Mars');
INSERT INTO Sensor VALUES ('Cassini',2000,'Jupiter');
INSERT INTO Sensor VALUES ('Galileo',1995,'Jupiter');

COMMIT;
```

3. **Result for Planet: Table name of Planet is a table of 7 planet names, number of moons per planet, type of planet, and length of year for each planet.**

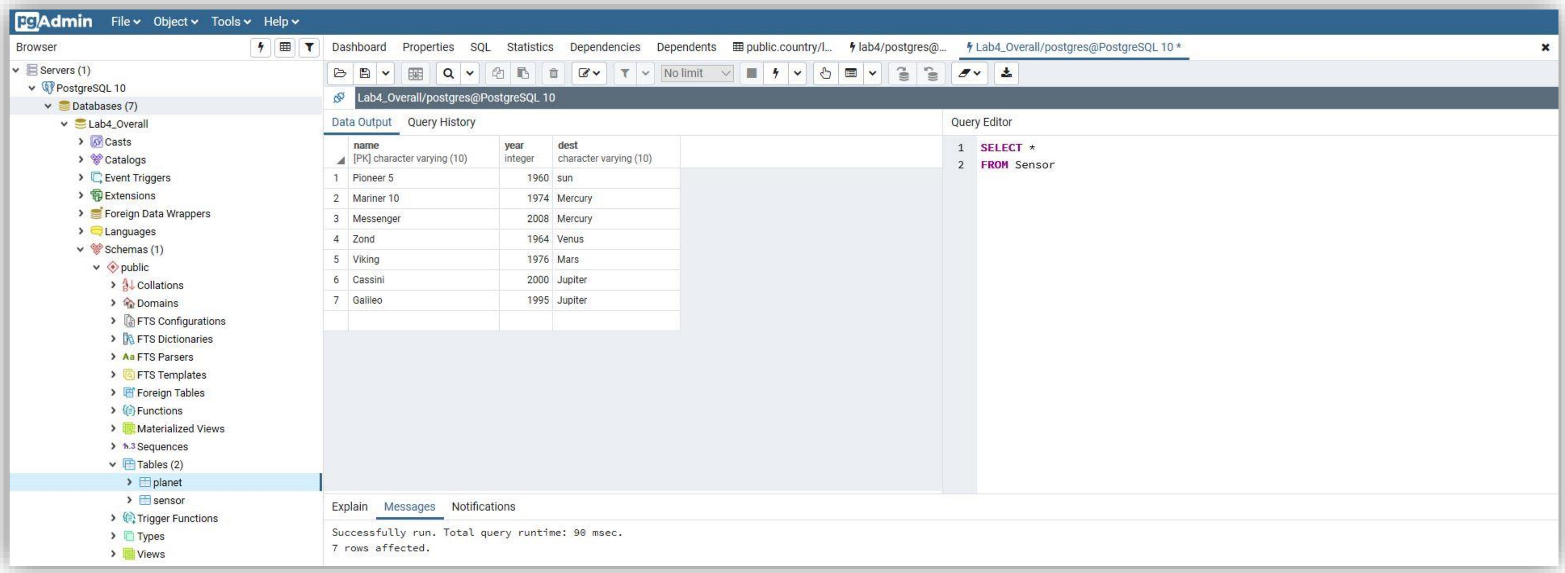


	name [PK] character varying (10)	nummoons integer	type character varying (5)	lengthofyear numeric (5,2)
1	Mercury	0	Rocky	0.24
2	Venus	0	Rocky	0.62
3	Earth	1	Rocky	1.00
4	Mars	2	Rocky	1.88
5	Jupiter	16	Gas	11.86
6	Saturn	18	Gas	29.46
7	Pluto	1	Rocky	247.70

```
SELECT *
FROM Planet
```

```
Successfully run. Total query runtime: 77 msec.
7 rows affected.
```

Result for Sensor: Table name of Sensor is a table of 7 sensor names, the year the sensor was deployed, and the destination planet of the sensor.



	name [PK] character varying (10)	year integer	dest character varying (10)
1	Pioneer 5	1960	sun
2	Mariner 10	1974	Mercury
3	Messenger	2008	Mercury
4	Zond	1964	Venus
5	Viking	1976	Mars
6	Cassini	2000	Jupiter
7	Galileo	1995	Jupiter

```
SELECT *
FROM Sensor
```

```
Successfully run. Total query runtime: 90 msec.
7 rows affected.
```

4.
   ```
   SELECT name
   FROM Planet
   ```

5.
   ```
   SELECT sum(nummoons)
   FROM Planet
   ```

6.
   ```
   SELECT name
   FROM Sensor
   WHERE dest = 'Mars'
   ```

7.
   ```
   SELECT E.name
   FROM Sensor E LEFT OUTER JOIN Planet D
   ON E.dest = D.name
   WHERE nummoons > 0
   ```

8.
   ```
   SELECT E.name
   FROM Sensor E LEFT OUTER JOIN Planet D
   ON E.dest = D.name
   WHERE TYPE = 'Rocky'
   ```

9.
   ```
   SELECT e.name
   FROM Sensor E LEFT OUTER JOIN Planet D
   ON E.dest = D.name
   WHERE d.lengthofyear = (SELECT
                          MIN(d.lengthofyear)
                          FROM Planet D)
   ```

10.
    ```
    SELECT name
    FROM planet
    WHERE lengthofyear = (SELECT MAX(lengthofyear)
                                     FROM planet)
    ```

11.
    ```
    SELECT name
    FROM Planet
    WHERE type = 'Gas'
    ```

12.
    ```
    SELECT name
    FROM Planet
    WHERE nummoons >= 1
    ```

13.
    ```
    SELECT E.name, D.name
    FROM Sensor E LEFT OUTER JOIN Planet D
    ON E.dest = D.name
    ```