

## CST 8215 - Lab 3A - Queries on Part\_T Table

---

### Objective:

1. Work on a table titled Part T to reinforce basic SQL syntax
2. Match expected output with results from SQL
3. Use column names
4. Use simple aggregate functions
5. Use string functions
6. Use INSERT statements to add data to table

### Reference:

1. Section 2.7 Aggregate Functions in postgresql-9.5-US.pdf
2. References in Lab02

**Submission:** Complete a short online quiz on SQL, make sure you understand all queries.

**Background:** We are using a table with four columns. The four columns are Part, Material, Size and Cost. The DDL and DML script are provided. Run the DDL script first, it will create the table. Then run the DML script, it will add data to the table. The table created is called Part\_T.

*Note:* There is a difference between Part and Part\_T. Part is a column name, Part T is the name of the table. Type in each query, do not cut and paste. Characters in this .PDF document are not compatible with the postgresSQL editor.

---

### Procedure:

**Query 1.** Choosing columns List all rows and columns from the table. Type in the query shown below, make sure you type in the semicolon at the end of the query.

Sample 1a: **SELECT \* FROM Part\_T;**

Type in

Sample 1b: **SELECT Part, Material, Size, Cost FROM Part \_T;**

In the first query the symbol \* represents all the columns, in the second query you have listed all the column names explicitly.

Practice 1c: **Write a query to list only Part, Material and Cost from the Part\_T table.**

**Query 2.** Using comparison operator

In this query we want to filter rows, i.e. list only those rows that meet a certain criteria. Type in the query shown below, it will list all parts that cost less than \$1.00. Note: If we want to include parts that cost \$1.00 or less than \$1.00 then we would use the <= operator.

Sample 2a: **SELECT Part, Material, Cost FROM Part\_T WHERE Cost < 1.00;**

Practice 2b: **Write a query to list Part, Size and Cost for Parts that cost greater than \$2.10.**

**Query 3.** Using an aggregate function

Here is a query to find the average cost of all parts made from BRASS.

Sample 3a: **SELECT AVG( Cost ) FROM Part\_T WHERE Material = 'BRASS';**

Practice 3b: **Write a query to find the average cost of all screws.**

**Query 4.** - Using substrings

Refer section 9.4 String Functions and Operators in postgresQL-9.5-US.pdf.

The following query will list the first three letters of Parts. Notice the keyword FROM inside the function SUBSTRING, it has a different purpose from the second FROM, which is used to choose table names.

Sample 4a: **SELECT SUBSTRING( Part FROM 1 FOR 3 ) FROM Part T;**

Sample 4b: **Write a query to list the first five letters of all Material.**

**Query 5.** - Adding a row to the Part T table Type in the following DML statement. It will add a column to the PART T table. Note: At this stage you may run this statement more than once, later we shall learn about CONSTRAINTS that will prevent the user from adding duplicate rows to tables.

Sample 5a: **INSERT INTO Part\_T( Part, Material, Size, Cost ) VALUES ( 'NAIL', 'STEEL', 'SMALL', 0.15 );**

Now type in a SQL statement to check your results.

Practice 5b:

**Write a query to add a row to the Part T table that contains the following data values.**

**Part -> HAMMER**

**Material -> STEEL**

**Size -> SMALL**

**Cost -> 7.15**

---