Hands-on Lab: Built-in functions - Aggregate, Scalar, String, Date and Time Functions

Now let us practice using sub-queries and working with multiple tables. Use the PETRESCUE-CREATE.sql file provided to create the table and execute the queries in the last two videos.

ID	ANIMAL	QUANTITY	COST	RESCUEDATE
1	Cat	9	450.09	5/29/2018
2	Dog	3	666.66	6/1/2018
3	Dog	1	100	6/4/2018
4	Parrot	2	50	6/4/2018
5	Dog	1	75.75	6/10/2018
6	Hamster	6	60.6	6/11/2018
7	Cat	1	44.44	6/11/2018
8	Goldfish	24	48.48	6/14/2018
9	Dog	2	222.22	6/15/2018

Objectives

After completing this lab, you will be able to:

1. Compose and run sub-queries with multiple tables
2. Check query results and view log files

Compose and run the following queries. Check that the results are what you expect and remember to view the logs in the Results section for errors and warnings.

Note: The solutions are provided at the end of this lab, but please try to compose the queries on your own before checking the solutions.

NOTE : Make sure that you are using the CSV file and datasets from the same instruction file.

Exercise 1: Create the Pet Rescue table

Rather than create the table manually by typing the DDL commands in the SQL editor, you will execute a script containing the create table command.

1. Download the script file PETRESCUE-CREATE.sql

   Note: To download, just right-click on the link above and click on **Save As..** or **Save Link As…** depending on your browser. Save the file as a .sql file and not HTML.

2. Login to IBM Cloud and go to the Resources Dashboard: https://cloud.ibm.com/resources where you can find the Db2 service that you created in a previous Lab. Click on the **Db2-xx** service. Next, open the Db2 Console by clicking on **Open Console** button. Go to the **Run SQL** page. The Run SQL tool enables you to run DDL and SQL statements.

3. Click on the + (Add New Script) icon.

4. Click on **From File**.

5. Locate the file **PETRESCUE-CREATE.sql** that you downloaded to your computer earlier and open it.

6. Once the statements are in the SQL Editor tool, you can run the queries against the database by selecting the **Run all** button.

7. On the right side of the SQL editor window you will see a **Result** section. Clicking on a query in the Result section will show the execution details of the job - whether it ran successfully or had any errors or warnings. Ensure your queries ran successfully and created all the tables.

8. Now you can look at the tables you created. Navigate to the three-bar menu icon, select **Explore**, then click on **Tables**.

9. Select the Schema corresponding to your Db2 userid. Then on the right side of the screen you should see the newly created **PETRESCUE** table listed (plus any other tables you may have created in previous labs e.g. INSTRUCTOR, TEST, etc.).

10. Click on any of the tables and you will see its SCHEMA definition (that is list of columns, their data types, and so on). You can also click **View Data** to view the content of the table.

Exercise 2: Aggregate Functions

Query A1: Enter a function that calculates the total cost of all animal rescues in the PETRESCUE table.

Query A2: Enter a function that displays the total cost of all animal rescues in the PETRESCUE table in a column called SUM_OF_COST.

Query A3: Enter a function that displays the maximum quantity of animals rescued.

Query A4: Enter a function that displays the average cost of animals rescued.

Query A5: Enter a function that displays the average cost of rescuing a dog.

Exercise 3: Scalar and String Functions

Query B1: Enter a function that displays the rounded cost of each rescue.

Query B2: Enter a function that displays the length of each animal name.

Query B3: Enter a function that displays the animal name in each rescue in uppercase.

Query B4: Enter a function that displays the animal name in each rescue in uppercase without duplications.

Query B5: Enter a query that displays all the columns from the PETRESCUE table, where the animal(s) rescued are cats. Use **cat** in lower case in the query.

Exercise 4: Date and Time Functions

Query C1: Enter a function that displays the day of the month when cats have been rescued.

Query C2: Enter a function that displays the number of rescues on the 5^{th} month.

Query C3: Enter a function that displays the number of rescues on the 14^{th} day of the month.

Query C4: Animals rescued should see the vet within three days of arrivals. Enter a function that displays the third day from each rescue.

Query C5: Enter a function that displays the length of time the animals have been rescued; the difference between todays date and the rescue date.

Lab Solutions

Exercise 2 Solutions: Aggregate Functions

Query A1: Enter a function that calculates the total cost of all animal rescues in the PETRESCUE table.

```
select SUM(COST) from PETRESCUE;
```

Query A2: Enter a function that displays the total cost of all animal rescues in the PETRESCUE table in a column called SUM_OF_COST.

```
select SUM(COST) AS SUM_OF_COST from PETRESCUE;
```

Query A3: Enter a function that displays the maximum quantity of animals rescued.

```
select MAX(QUANTITY) from PETRESCUE;
```

Query A4: Enter a function that displays the average cost of animals rescued.

```
select AVG(COST) from PETRESCUE;
```

Query A5: Enter a function that displays the average cost of rescuing a dog.

Hint - Bear the cost of rescuing one dog on one day, which is different from another day. So you will have to use an average of averages.

```
select AVG(COST/QUANTITY) from PETRESCUE where ANIMAL = 'Dog';
```

Exercise 3 Solutions: Scalar and String Functions

Query B1: Enter a function that displays the rounded cost of each rescue.

```
select ROUND(COST) from PETRESCUE;
```

Query B2: Enter a function that displays the length of each animal name.

```
select LENGTH(ANIMAL) from PETRESCUE;
```

Query B3: Enter a function that displays the animal name in each rescue in uppercase.

```
select UCASE(ANIMAL) from PETRESCUE;
```

Query B4: Enter a function that displays the animal name in each rescue in uppercase without duplications.

`select DISTINCT(UCASE(ANIMAL)) from PETRESCUE;

Query B5: Enter a query that displays all the columns from the PETRESCUE table, where the animal(s) rescued are cats. Use **cat** in lower case in the query.

```
select * from PETRESCUE where LCASE(ANIMAL) = 'cat';
```

Exercise 4 Solutions: Date and Time Functions

Query C1: Enter a function that displays the day of the month when cats have been rescued.

```
select DAY(RESCUEDATE) from PETRESCUE where ANIMAL = 'Cat';
```

Query C2: Enter a function that displays the number of rescues on the 5th month.

```
select SUM(QUANTITY) from PETRESCUE where MONTH(RESCUEDATE)='05';
```

Query C3: Enter a function that displays the number of rescues on the 14th day of the month.

```
select SUM(QUANTITY) from PETRESCUE where DAY(RESCUEDATE)='14';
```

Query C4: Animals rescued should see the vet within three days of arrivals. Enter a function that displays the third day from each rescue.

```
select (RESCUEDATE + 3 DAYS) from PETRESCUE;
```

Query C5: Enter a function that displays the length of time the animals have been rescued; the difference between todays date and the rescue date.

```
select (CURRENT DATE - RESCUEDATE) from PETRESCUE;
```

Summary

You can now compose and run queries, check results and view the logs. You will use these skills in later labs.

Author(s)

Rav Ahuja