

SQL Training, Session 2

This lesson assumed that you have completed the w3 schools SQL tutorial up to LEFT JOIN. We will go over some problems together with a project assigned at the end of the lesson. As before, you have two weeks to upload your .sql files into your personal github.

This is an excellent visual explanation of joins:

<https://stackoverflow.com/questions/38549/what-is-the-difference-between-inner-join-and-outer-join>

For the most part we can skip RIGHT and FULL joins as they are not native to BigQuery.

Recall the Table from Last Session. We are going to add some additional variables and change the variables names slightly.

emp_id	emp_name	age	annual_salary	tenure	department
101	Sally Johnson	29	50000	1	IT
102	Tom Brooks	45	80000	4	analytics
103	Tom Felton	27	50000	1	IT
104	Jack Kelvin	55	85000	5	Sales
105	Beth George	35	70000	3	Sales
106	Rusty	65	120000	4	executive

- 1) Write a query that generates all variables for any employee whose in the IT department or analytics
- 2) This is not in the W3 tutorial but is an essential method to optimize queries that require multiple levels of data retrieval. Let's compute the average salary of all employees who are in the IT or analytics department using a sub query.

Subquery: A SQL query within another SQL query, be it in the WHERE clause, FROM statement, or JOIN statement.

For our problem, we want to design our subquery in two layers, the first (inner) layer is what we wrote for problem 1 and the outer layer will compute the average annual salary. Of course there are multiple ways to do this.

Let's say we now have another table that was created by HR. This table contains the employee Id and an employee rating out of five assigned after the annual review. We also have a variable with the ID of the HR rep who conducted the evaluation. Let's call this table "evaluations"

emp_id	Emp_rating	Hr_evaluator_id
101	4.5	Hr1
102	5	Hr2
103	4.6	Hr2
104	4.1	Hr1
105	4	Hr1
106	4.9	Hr2
		Hr3

In this example, the emp_id variable is known as our primary key. By definition, a primary key uniquely identifies each record in a table. Primary keys must be UNIQUE values and cannot contain NULL values. A table can only have ONE primary key. In our example, the primary key uniquely identifies each employee. Since the employee id is found both in the employees table and evaluations, we can do some joins on that id.

- 3) Generate a table that gives the employee id, employee name, employee rating, and the HR rep assigned to that employee using JOIN.
- 4) Generate a table that gives the employee id, employee name, employee rating, and all HR reps, even if they were not to an employee using LEFT join.