 **📚 Lesson** | Subqueries

— Subqueries

PROMPT: The ability to perform subqueries adds a large amount of versatility to what you can do with SQL, and can make some queries more efficient and easier to read. Practice what you’ve learned with these practice questions.

1. Using the **initech\_customers** table, write a query that returns the names of all customers who have placed an order. You’ll use the **initech\_orders** table, but you should do this without using a **JOIN** clause! If done correctly, your output should have a total of 98 entries.

|  |
| --- |
| select customername  from initech\_customers  where customernumber in  (select customernumber from initech\_orders) |

1. Using the **titanic\_passengers** dataset, write a query that returns the sex of the passenger, whether they survived or not, and a calculated field determining whether or not they are older than the average age of all passengers (1 if older than the average, 0 if not). HINT: first write a query to find the average age of all passengers. You’ll need this for your calculated field!

|  |
| --- |
| SELECT  sex,  survived,  CASE  When age > (SELECT AVG(age) From titanic\_passengers)  Then 1  Else 0  END AS is\_above\_avg  FROM titanic\_passengers |

1. Expand your query from the previous question to determine the average survival rate based on the sex of the passenger and whether or not they are above the average age. Your output should have four rows of data: one for each combination of sex and age group. If done correctly, you should see that the survival rate for males younger than the average age is about 18%; survival rate for females younger than the average age is about 81%.

|  |
| --- |
| SELECT  sex,  is\_above\_avg,  AVG(survived)  FROM (SELECT  name,  sex,  survived,  CASE  WHEN age > (SELECT AVG(age) FROM titanic\_passengers)  Then 1  Else 0  END AS is\_above\_avg  FROM titanic\_passengers) AS TEMP  GROUP BY  sex,  is\_above\_avg |
|  |