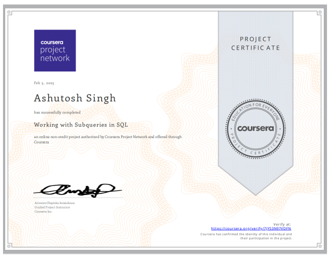
Project: Mastering SQL Subqueries



coursera project network

PROJECT CERTIFICATE

Feb 5, 2025

Ashutosh Singh

has successfully completed

Working with Subqueries in SQL

an online non-credit project authorized by Coursera Project Network and offered through Coursera

Arimoro Olayinka Imisioluwa
Guided Project Instructor
Coursera Inc.

Verify at:
https://coursera.org/verify/7JYS3N97VDFN
Coursera has confirmed the identity of this individual and their participation in the project.

Objectives

In this course, we are going to focus on **three** learning objectives:

1. Use subqueries in the **WHERE** clause
2. Use subqueries in the **FROM** clause
3. Use subqueries in the **SELECT** clause

Task One:

Getting Started

-- In this task, we will retrieve data **FROM** the tables in the employees database

-- **1.1**: Retrieve all the data FROM tables in the employees database

```
SELECT * FROM employees;

SELECT * FROM departments;

SELECT * FROM dept_emp;

SELECT * FROM dept_manager;

SELECT * FROM salaries;

SELECT * FROM customers;

SELECT * FROM sales;
```

Task Two:

Subquery in the **WHERE** clause

-- In this task, we will learn how to use a subquery in the WHERE clause

-- **2.1**: Retrieve a list of all employees that are not managers

```
SELECT  * FROM   employees

WHERE  emp_no IN (SELECT  emp_no FROM   dept_manager);
```

-- **2.2**: Retrieve all columns in the sales table for customers above 60 years old

-- Returns the count of customers

```
SELECT  customer_id, COUNT(*)

FROM   sales

GROUP BY customer_id

ORDER BY COUNT(*) DESC;
```

-- **Solution**

```
SELECT  * FROM   sales

WHERE  customer_id IN (SELECT  customer_id FROM   customers WHERE  age > 60)
```

-- **2.3**: Retrieve a list of all manager's employees number, first and last names

-- Returns all the data FROM   the dept_manager table

```
SELECT  * FROM   dept_manager;
```

-- **Solution**

```
SELECT  emp_no, first_name, last_name FROM   employees

WHERE  emp_no IN (SELECT  emp_no FROM   dept_manager)
```

-- **Exercise 2.1**: Write a JOIN statement to get the result of 2.3:

```
SELECT  e.emp_no, e.first_name, e.last_name FROM   employees e

JOIN dept_manager dm

ON e.emp_no = dm.emp_no;
```

-- **Exercise 2.2**: Retrieve a list of all managers that were employed between 1st January, 1990 and 1st January, 1995

```
SELECT  * FROM   dept_manager

        WHERE  emp_no IN (SELECT  emp_no FROM   employees WHERE  hire_date
        between '1990-01-01' and '1995-01-01' )
```

Task Three:

Subquery in the FROM clause

-- In this task, we will learn how to use a subquery in the **FROM** clause

-- **3.1**: Retrieve a list of all customers living in the southern region

```
SELECT a.customer_id, a.customer_region, b.category
FROM
        (SELECT customer_id,region customer_region FROM customers

         WHERE region = 'South') a, (SELECT customer_id, category FROM sales) b
```

-- **3.2**: Retrieve a list of managers and their department names

-- Returns all the data FROM the dept_manager table

```
SELECT * FROM dept_manager;
```

-- **Solution**

```
SELECT dm.*, d.department_name

FROM dept_manager dm,

         (SELECT dept_no, dept_name department_name FROM departments) d;
```

-- **Exercise 3.1**: Retrieve a list of managers, their first, last, and their department names

-- Returns data FROM the employees table

```
SELECT * FROM employees;
```

-- **Solution**

```
SELECT dm.emp_no, dm.dept_no, e.first_name, e.last_name, d.dept_name

FROM dept_manager dm,

         (SELECT emp_no, first_name, last_name FROM employees) e,

         (SELECT dept_no, dept_name FROM departments) d

WHERE dm.dept_no= d.dept_no AND dm.emp_no = e.emp_no;
```

Task Four:

Subquery in the SELECT clause

-- In this task, we will learn how to use a subquery in the SELECT clause

-- **4.1**: Retrieve the first name, last name and average salary of all employees

```
SELECT  first_name, last_name,

           (SELECT  round(AVG(salary),2) FROM   salaries) average_salary

FROM   employees;
```

-- **Exercise 4.1**: Retrieve a list of customer_id, product_id, order_line and the name of the customer

-- Returns data FROM   the sales and customers tables

```
SELECT  * FROM   sales

ORDER BY customer_id;


SELECT  * FROM   customers;
```

-- **Solution**

```
SELECT  customer_id, product_id, order_line,

           (SELECT  customer_name

            FROM   customers c

            WHERE  s.customer_id = c.customer_id )

FROM   sales s

ORDER BY customer_id;
```

Task Five:

Subquery Exercises - Part 1

-- In this task, we will try our hands on more exercises on subqueries

-- **Exercise 5.1**: Return a list of all employees who are in Customer Service department:

-- Returns data FROM the dept_emp and departments tables:

```
SELECT * FROM dept_emp;

SELECT * FROM departments;
```

-- **Solution**

```
SELECT * FROM dept_emp

WHERE dept_no IN (SELECT dept_no FROM departments

                  WHERE dept_name = 'Customer Service');
```

-- **Exercise 5.2**: Include the employee number, first and last names

```
SELECT de.emp_no, e.first_name, e.last_name FROM employees e

JOIN

(SELECT * FROM dept_emp

WHERE dept_no IN (SELECT dept_no FROM departments

                  WHERE dept_name = 'Customer Service')) de

ON e.emp_no = de.emp_no

ORDER BY emp_no;
```

-- **Exercise 5.3**: Retrieve a list of all managers who became managers after the 1st of January, 1985 and are in the Finance or HR department:

-- Returns data FROM the departments and dept_manager tables

```
SELECT  * FROM   departments;

SELECT  * FROM   dept_manager

WHERE  FROM  _date > '1985-01-01';
```

-- **Solution**

```
SELECT  * FROM   dept_manager

WHERE  FROM  _date > '1985-01-01'
AND
dept_no IN (SELECT  dept_no FROM   departments WHERE  dept_name = 'Finance' OR

                  dept_name = 'Human Resources');
```

-- **Exercise 5.4**: Retrieve a list of all employees that earn above 120,000 and are in the Finance or HR departments

-- Retrieve a list of all employees that earn above 120,000

```
SELECT  emp_no, salary FROM   salaries

WHERE  salary > 120000;
```

-- **Solution**

```
SELECT  emp_no, salary FROM   salaries

WHERE  salary > 120000

AND

emp_no IN (SELECT  emp_no FROM   dept_emp

                   WHERE  dept_no ='d002' OR dept_no ='d003');
```

-- **Alternative Solution**

```
SELECT  emp_no, salary FROM   salaries

WHERE  salary > 120000

AND

emp_no IN (SELECT  emp_no FROM   dept_emp

                   WHERE  dept_no IN ('d002','d003'));
```

-- **Exercise 5.5:** Retrieve the average salary of these employees:

```
SELECT  emp_no, ROUND(AVG(salary),2) as avg_salary FROM   salaries

WHERE  salary > 120000

AND

emp_no IN (SELECT  emp_no FROM   dept_emp

                   WHERE  dept_no IN ('d002','d003'))

GROUP BY emp_no

ORDER BY avg_salary DESC;
```

Task Six:

Subquery Exercises - Part Two

-- In this task, we will try our hands on more

-- exercises on subqueries

-- **Exercise 6.1**: Return a list of all employees number, first and last name.

-- Also, return the average salary of all the employees and average salary of each employee

-- Retrieve all the records in the salaries table

```
SELECT  * FROM   salaries;
```

-- Return the employee number, first and last names and average salary of all employees

```
SELECT  e.emp_no, e.first_name, e.last_name,

(SELECT  ROUND(AVG(salary), 2) FROM   salaries) avg_salary

FROM   employees e

JOIN salaries s

ON e.emp_no = s.emp_no

ORDER BY e.emp_no;
```

-- Returns the employee number and average salary of each employee

```
SELECT  emp_no, ROUND(AVG(salary), 2) AS emp_avg_salary

FROM   salaries

GROUP BY emp_no

ORDER BY emp_no;
```

-- **Solution**

```sql
SELECT  e.emp_no, e.first_name, e.last_name,
(SELECT  ROUND(AVG(salary), 2) FROM   salaries) avg_salary, s.emp_avg_salary
FROM   employees e
 JOIN   (SELECT  emp_no, ROUND(AVG(salary), 2) AS emp_avg_salary
                FROM   salaries
                GROUP BY emp_no
                ORDER BY emp_no) s
ON e.emp_no = s.emp_no
ORDER BY e.emp_no;
```

-- **Exercise 6.2**: Find the difference between an employee's average salary

-- and the average salary of all employees

```sql
SELECT  e.emp_no,
(SELECT  ROUND(AVG(salary),2) FROM   salaries) as avg_salary_all,
(SELECT  ROUND(AVG(salary),2) FROM   salaries) - b.avg_salary_emp salary_diff
FROM   employees e
JOIN (SELECT  emp_no, ROUND(AVG(salary),2) avg_salary_emp  FROM   salaries
                GROUP BY emp_no) b
ON e.emp_no = b.emp_no
ORDER BY e.emp_no
```

-- **Exercise 6.3:** Find the difference between the maximum salary of employees

-- in the Finance or HR department and the maximum salary of all employees

```sql
SELECT  e.emp_no, e.first_name, e.last_name, a.emp_max_salary,
(SELECT  MAX(salary) max_salary FROM   salaries),
(SELECT  MAX(salary) max_salary FROM   salaries) - a.emp_max_salary salary_diff
```

```
FROM   employees e
JOIN (SELECT  s.emp_no, MAX(salary) AS emp_max_salary
                                      FROM   salaries s
                                      GROUP BY s.emp_no
                                      ORDER BY s.emp_no) a
ON e.emp_no = a.emp_no
WHERE  e.emp_no IN (SELECT  emp_no FROM   dept_emp WHERE  dept_no IN ('d002', 'd003'))
ORDER BY emp_no;
```

Task Seven:

Subquery Exercises - Part Three

-- In this task, we will try our hands on more

-- exercises on subqueries

-- **Exercise 7.1**: Retrieve the salary that occured most

-- Returns a list of the count of salaries

```
SELECT  salary, COUNT(*)
FROM   salaries
GROUP BY salary;
```

-- **Solution**

```
SELECT  a.salary
FROM
(SELECT  salary, COUNT(*)
FROM   salaries
GROUP BY salary
ORDER BY count(*) DESC
LIMIT 1) a;
```

-- **Exercise 7.2**: Find the average salary excluding the highest and the lowest salaries

-- Returns the average salary of all employees

```
SELECT  ROUND(AVG(salary), 2) avg_salary
FROM   salaries
```

-- **Solution**

```
SELECT  ROUND(AVG(salary), 2) AS avg_salary

FROM   salaries

WHERE  salary NOT IN((SELECT  MIN(salary) FROM   salaries),(SELECT  MAX(salary)
FROM   salaries));
```

-- Exercise 7.3: Retrieve a list of customers id, name that have bought the most FROM the store

-- Returns a list of customer counts

```
SELECT  customer_id, COUNT(*) AS cust_count

FROM   sales

GROUP BY customer_id

ORDER BY cust_count DESC;
```

-- **Solution**

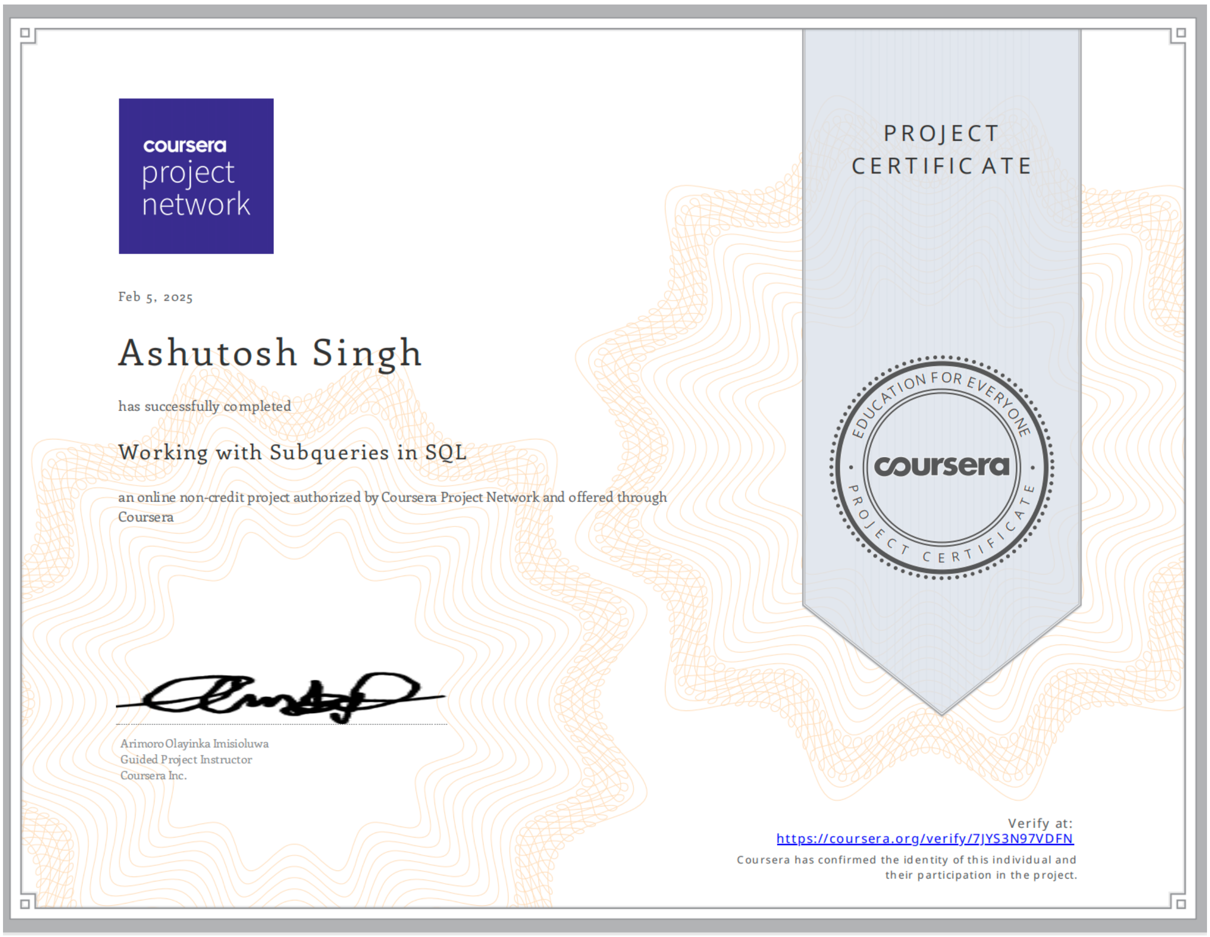
```
SELECT  c.customer_id,c.customer_name, a.cust_count

FROM   customers c,

        (SELECT  customer_id, COUNT(*) AS cust_count

        FROM   sales

        GROUP BY customer_id

        ORDER BY cust_count DESC) a

WHERE  c.customer_id = a.customer_id

ORDER BY a.cust_count DESC;
```

-- **Exercise 7.4:** Retrieve a list of the customer name and segment of those customers that bought the most FROM   the store and had the highest total sales

```
-- Returns a list of customer counts and total sales

SELECT  customer_id, COUNT(*) AS cust_count, SUM(sales) total_sales

FROM   sales

GROUP BY customer_id

ORDER BY total_sales DESC, cust_count DESC;
```

-- Solution

```
SELECT  c.customer_id, c.customer_name, c.segment, a.cust_count, a.total_sales

FROM   customers c,

        (SELECT  customer_id, COUNT(*) AS cust_count, SUM(sales) total_sales

        FROM   sales

        GROUP BY customer_id

        ORDER BY total_sales DESC, cust_count DESC) a

WHERE  c.customer_id = a.customer_id

ORDER BY a.total_sales DESC, a.cust_count DESC;
```



coursera project network

Feb 5, 2025

Ashutosh Singh

has successfully completed

Working with Subqueries in SQL

an online non-credit project authorized by Coursera Project Network and offered through Coursera

Arimoro Olayinka Imisioluwa
Guided Project Instructor
Coursera Inc.

PROJECT CERTIFICATE

EDUCATION FOR EVERYONE
coursera
PROJECT CERTIFICATE

Verify at:
https://coursera.org/verify/7JYS3N97VDFN
Coursera has confirmed the identity of this individual and their participation in the project.