COMM1822

Term 2 2022

Introduction to Databases for Business Analytics

Week 9 Subqueries and Merge Statement

Lecturer-in-Charge: Kam-Fung (Henry) Cheung
Email: kf.cheung@unsw.edu.au
Tutors: Theresa Tran
                  Liam Li Chen
                  Kathy Xu
PASS Leader: Srilekha Chandrashekara Kolaki

WARNING

This material has been reproduced and communicated to you by or on behalf of the University of New South Wales in accordance with section 113P(1) of the Copyright Act 1968 (Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

Copyright

- There are some file-sharing websites that specialise in buying and selling academic work to and from university students.

- If you upload your original work to these websites, and if another student downloads and presents it as their own either wholly or partially, **you might be found guilty of collusion — even years after graduation.**

- These file-sharing websites may also accept purchase of course materials, **such as copies of <u>lecture slides</u> and <u>tutorial handouts</u>. By law, the copyright on course materials, developed by UNSW staff in the course of their employment, belongs to UNSW. It constitutes copyright infringement, if not academic misconduct, to trade these materials.**

Acknowledgement of Country

UNSW Business School acknowledges the Bidjigal (Kensington campus) and Gadigal (City campus) the traditional custodians of the lands where each campus is located.

We acknowledge all Aboriginal and Torres Strait Islander Elders, past and present and their communities who have shared and practiced their teachings over thousands of years including business practices.

We recognise Aboriginal and Torres Strait Islander people's ongoing leadership and contributions, including to business, education and industry.
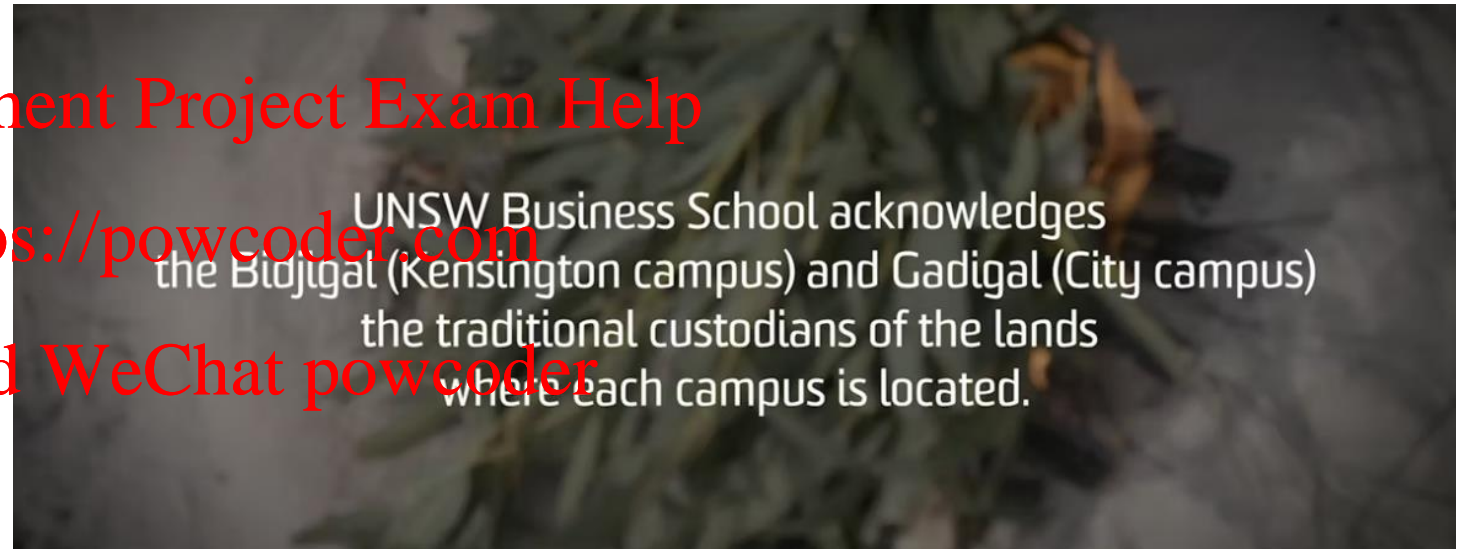
UNSW Business School. (2022, May 7). *Acknowledgement of Country* [online video]. Retrieved from https://vimeo.com/369229957/d995d8087f

At UNSW you are free to...

UNSW SYDNEY

- Respectfully disagree about anything
- Express different opinions
- Write your beliefs
- Show your beliefs
- Leave any club or organisation

It's not acceptable to...

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Attempt to censor opinions
- Use hate speech
- Make threats or instil fear
- Make false accusations
- Access or share others private information without consent

We are here to help...

- Tell a teacher
- Tell UNSW Psychology and Wellness
- Report to UNSW Complaints
- Report to UNSW Security
- Report a crime to police

Find out more

MyExperience

Our aim is **80%** response rate!

Assignment Project Exam Help

Please fill in the MyExperience survey now!

https://powcoder.com

https://web.microsoftstream.com/video/48ac8258-0c17-4b84-89c7-80dff76e6db5

Add WeChat powcoder

Thank you for being so patient with us and we look forward to your feedback.

myExperience

Tell us about your experience. Shape the future of education at UNSW.

Visit **Moodle** to complete the myExperience survey or accessing **myexperience.unsw.edu.au** directly.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

UNSW
S Y D N E Y

Subqueries and Merge Statements

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Subqueries and Merge Statements

❑A subquery is a nested query – one complete query inside another query.

❑The results of the subquery (i.e., the inner query) will be passed to the outer query (also called the parent query).

Subqueries and Merge Statements

With the **exception** of the ORDER BY clause, you can virtually nest a subquery in any of the clauses in a SELECT statement

```
SELECT
        <SubQuery>
FROM
        <SubQuery>
JOIN
        <SubQuery>
WHERE
        <SubQuery>
HAVING
        <SubQuery>
ORDER BY
        <Column>
```

Subqueries and Merge Statements

Subquery	Description
Single-row subquery	Returns to the outer query one row of results that consists of one column
Multiple-row subquery	Returns to the outer query more than one row of results
Multiple-column subquery	Returns to the outer query more than one column of results
Correlated subquery	References a column in the outer query, and executes the subquery once for every row in the outer query
Uncorrelated subquery	Executes the subquery first and passes the value to the outer query

Multiple-Row Subqueries

Operator	Description
> ALL	More than the highest value returned by the subquery
< ALL	Less than the lowest value returned by the subquery
< ANY	Less than the highest value returned by the subquery
> ANY	More than the lowest value returned by the subquery
= ANY	Equal to any value returned by the subquery (same as IN)

Multiple-Row Subqueries

❑Multiple Row Subqueries return more than one row of results.

❑Operators such as IN, ANY, ALL, or EXISTS.

❑ANY and ALL operators combine with other comparison operators to treat a subquery's results as a set of values instead of single values.

Multiple-Column Subqueries

A multiple column subquery returns more than one column to the parent query and can be listed in the parent query's FROM, WHERE or HAVING clause.
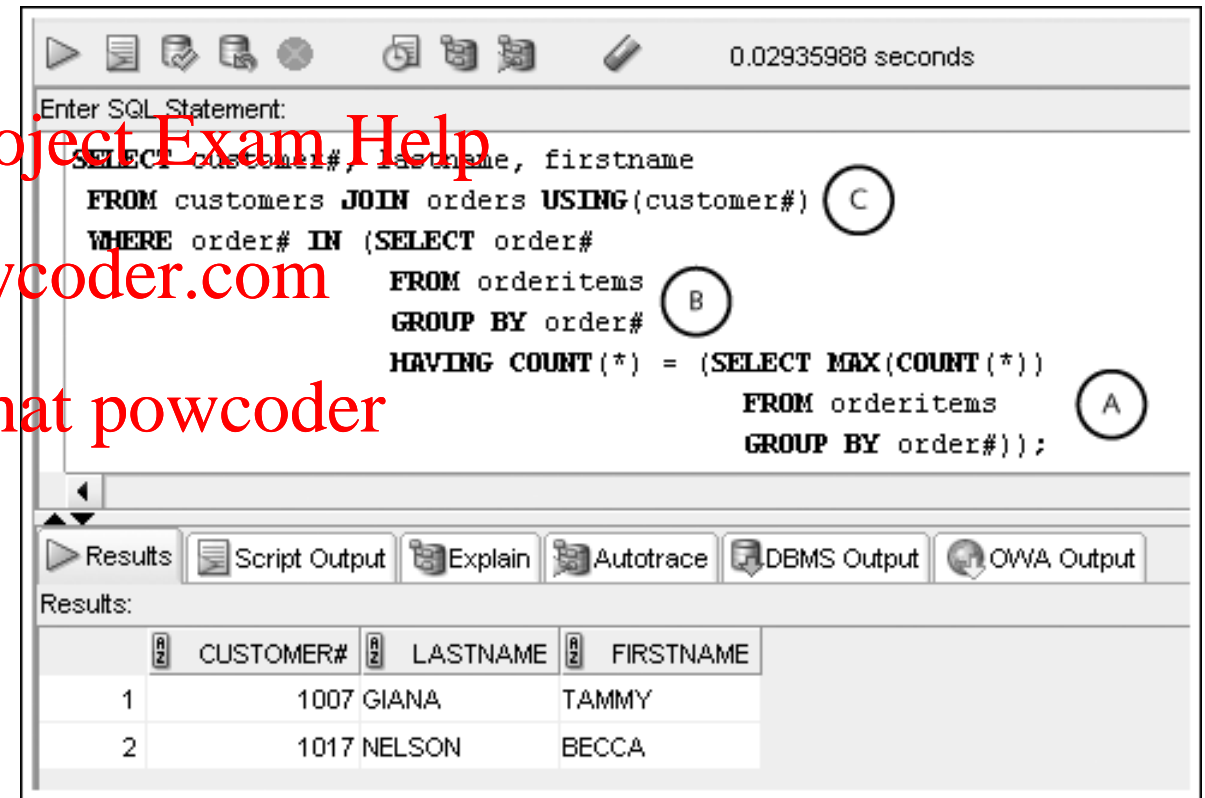
Nested Subqueries

The innermost subquery is executed first, then the next highest level subquery is executed, and so on, until the outermost query is reached.

Correlated Subqueries vs Uncorrelated Subqueries

❑The sequences of **uncorrelated subqueries** process are as follows:
1) Inner query is executed first
2) The result is then passed from the inner query to the parent query
3) Parent query is executed

❑ On the other hand, the process sequence for **correlated subqueries** is different:
1) Inner query is executed once for each row processed by the parent query
2) Inner query references the row contained in the parent query

Exercise

UNSW
SYDNEY

Logistics Database

Consider the **Logistics Database** from https://www.w3schools.com/sql/sql_any_all.asp

Tables in the database:

Customers (CustomerID, CustomerName, ContactName, Address, City, PostalCode, Country)
Category (CategoryID, CategoryName, Description)
Employees (EmployeeID, LastName, FirstName, BirthDate, Photo, Notes)
Suppliers (SupplierID, SupplierName, ContactName, Address, City, PostalCode, Country, Phone)
Products (ProductID, ProductName, SupplierID, CategoryID, Unit, Price)
Shippers (ShipperID, ShipperName, Phone)
Orders (OrderID, CustomerID, EmployeeID, OrderDate, ShipperID)
OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)

Exercise 1

List the product name if it finds any records in the OrderDetails table has quantity equal to 3.

```
SELECT      ProductName
FROM        Products
WHERE       ProductID = ANY
(   SELECT  ProductID
    FROM    OrderDetails
    WHERE   Quantity = 3
);
```

Result:

Number of Records: 3

ProductName
Scottish Longbreads
Fløtemysost
Mozzarella di Giovanni

Exercise 2

List the customer's name who has ordered a product with quantity of at least 80.

```sql
SELECT      CustomerName
FROM        Customers
WHERE       CustomerID = ANY
(   SELECT      CustomerID
    FROM        Orders
    WHERE       OrderID = ANY
    (   SELECT      OrderID
        FROM        OrderDetails
        WHERE       Quantity >= 80
    )
);
```

1. Find OrderID with Quantity >= 80 in OrderDetails table;
2. Find CustomerID in Orders table using the OrderID in Step 1;
3. List all the eligible customers' names in Customers table.

Customers (CustomerID, CustomerName, …)

Orders (OrderID, CustomerID, …)

OrderDetails (OrderDetailID, OrderID, …, Quantity)

Result:

Number of Records: 5

CustomerName
Ernst Handel
Hungry Owl All-Night Grocers
QUICK-Stop
Save-a-lot Markets
Seven Seas Imports

UNSW
SYDNEY

Exercise 3

List the product name if ALL the records in the OrderDetails table has quantity equal to 3.

```
SELECT        ProductName
FROM          Products
WHERE         ProductID = ALL
(     SELECT  ProductID
      FROM    OrderDetails
      WHERE   Quantity = 3
);
```

Result:

Number of Records: 0

ProductName

Exercise 4

List the customer's name who has ordered the same products as customer 3.
(Please use this link https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_join)

```sql
SELECT  DISTINCT CustomerName
FROM            Customers JOIN Orders USING (CustomerID) JOIN OrderDetails USING (OrderID)
WHERE           ProductID IN
(       SELECT    ProductID
        FROM      OrderDetails JOIN Orders USING (OrderID)
        WHERE     CustomerID = 3
);
```
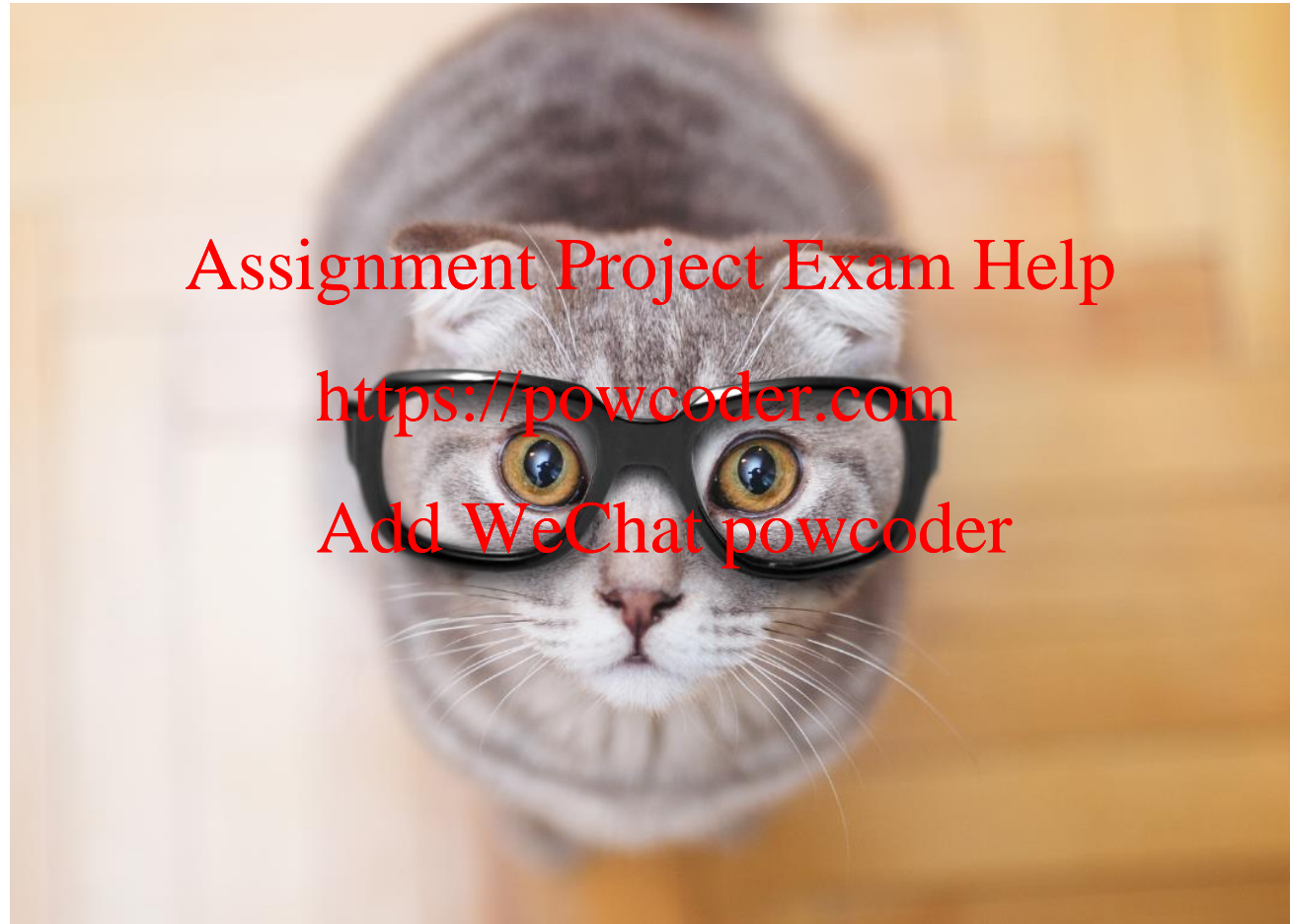
Result:

Number of Records: 8

CustomerName
Wilman Kala
LILA-Supermercado
Folk och fä HB
Piccolo und mehr
Antonio Moreno Taquería
Ottilies Käseladen
Ernst Handel
Reggiani Caseifici

UNSW
SYDNEY

Questions

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Source: stacker.com