**TERM : Summer 2025**

**A picture containing logo

Description automatically generated**

| **Course & Section Code:** | **COMP214** |
| --- | --- |
| **Course Name:** | **Advanced Database Concept** |
|  |  |
| **Q&A / Virtual Office Hour:** |  |
| **Instructor Name & Email:** | Ersan Cam  [ecam@my.centennialcollege.ca](mailto:ecam@my.centennialcollege.ca) |

Assigment#1 \_Version2

(Subqueries, Joins , group by)

**Due Date : June 15th, 2025**

Instructions for delivery of this Labs file back to instructor.

**Step 1:** Download this word copy of Lab document.

**Step2:** Work on your question in SQL Developer.

**Step3:** Once you solve the problem copy paste the code under each question and Highlight with RED color

**Step4:** Also go to your Sql Developer and capture screen entire screen with command you execute and result at the bottom. Use Snipping tool in windows to capture screen shot. Below picture shows how to open free windows based snipping screen capture tool

Step5: Drop your finalized & saved word document to respective Lab /Assignment dropbox assignment folder.

**Please note that Instructor keeps the right to call out any students randomly to ask demonstration of their solution, and walk thru their work and justify their answers in one on one breakout room.**

**The tool to capture screen shot. Snipping Tool or equivalent.**

Graphical user interface, text, application

Description automatically generated

Demo Screen shot: Always share Editable version of your code and Screen shot of your SQL developer with your instructor.. Do not share just code or not only Screen shot…

Copy paste screen shot of same command

Graphical user interface, text, application, email

Description automatically generated

**For first 6 questions please use Just Lee Book (JL\_XXX) database**

**Question 1**

A manager at JustLee Books requests a list of the titles of all books and order# of them from orderitems table and books table for the orders spending more than 100 .. (How to calculate spending amount for each order is multiply by quantity \*paideach and then add up (SUM ) for each order#

Basically you will have to group by based on order# in orderitems table and calculate total spending for each order# based on above formula and then after grouping list them only the ones its spending is 100$ or more

Hint: You may need to use group by , having and join together

SELECT b.title, o.order#

FROM JL\_BOOKS b

JOIN JL\_ORDERITEMS o ON b.ISBN = o.ISBN

WHERE o.order# IN (

SELECT order#

FROM JL\_ORDERITEMS

GROUP BY order#

HAVING SUM(quantity \* paideach) > 100

);

A screenshot of a computer

AI-generated content may be incorrect.

**Question 2**

list of customer# and their total order counts from orders table for those orders are having 1 or more days in between ship date - orderdate

PS: when you come up with group by results, only list count results 2 or more

**Hint:**

select customer#,count(\*) from jl\_orders

where shipdate-orderdate >= ???

group by xxxxxx

SELECT customer#, count(\*)

FROM JL\_ORDERS

WHERE shipdate-orderdate >= 1

GROUP BY customer#

HAVING COUNT(\*) >= 2;  
  
  
  
A screenshot of a computer

AI-generated content may be incorrect.

**Question 3**

Display how many times this book (ISBN no 3437212490) has been sold among all orders

SELECT SUM(quantity)

FROM JL\_ORDERITEMS

WHERE ISBN = '3437212490';

A screenshot of a computer

AI-generated content may be incorrect.

**Question 4** .

Create a list the title of all books in the same category as books previously purchased by customer 1005. Don’t include books this customer has already purchased.

Hint: Sub query with NOT IN

SELECT b.title

FROM JL\_BOOKS b

WHERE b.category IN (

SELECT b.category

FROM JL\_BOOKS b

JOIN JL\_ORDERITEMS oi ON b.ISBN = oi.ISBN

JOIN JL\_ORDERS o ON oi.order# = o.order#

WHERE o.customer# = 1005

)

AND b.ISBN NOT IN (

SELECT oi.ISBN

FROM JL\_ORDERITEMS oi

JOIN JL\_ORDERS o ON oi.order# = o.order#

WHERE o.customer# = 1005

);

A screenshot of a computer

AI-generated content may be incorrect.

**Question 5** .

**List the shipping city and state for the order that had the longest shipping delay.**

Hint: Sub query with MAX group function in subquery

SELECT shipcity, shipstate

FROM JL\_ORDERS

WHERE shipdate-orderdate = (

SELECT MAX(shipdate-orderdate)

FROM JL\_ORDERS

);

A screenshot of a computer

AI-generated content may be incorrect.

**Question 6**

Determine which customers placed orders for the least expensive book (in terms of regular retail price) carried by JustLee Books.

Hint: Inner Join and Subquery . Subquery is something like

retail =

(SELECT MIN(retail)

FROM books);

SELECT c.firstname || ' ' || c.lastname as "Customer"

FROM JL\_CUSTOMERS c

JOIN JL\_ORDERS o ON c.customer# = o.customer#

JOIN JL\_ORDERITEMS oi ON o.order# = oi.order#

JOIN JL\_BOOKS b ON oi.ISBN = b.ISBN

WHERE b.retail =

(SELECT MIN(retail)

FROM JL\_BOOKS

);

A screenshot of a computer

AI-generated content may be incorrect.

**Below questions are from HR\_XXXX database**

**Question #7**

Create a report/query for the HR application to produce the address and city and country name details of all departments in the company.

Output should be similar to below. May not be exact same

Table

Description automatically generated

Hint: Join

SELECT DISTINCT l.location\_id, l.street\_address, l.city, c.country\_name

FROM HR\_DEPARTMENTS d

JOIN HR\_LOCATIONS l ON d.location\_id = l.location\_id

JOIN HR\_COUNTRIES c ON l.country\_id = c.country\_id

ORDER BY l.location\_id;

A screenshot of a computer

AI-generated content may be incorrect.

**Question #8**

The HR department needs a report of employees in US country.

Display the last name of employee, salary, job, department\_number , department name , city and country\_id details for those who are in country\_id as US and for those salary is in the range of 3400 and 9500

Sample output should be similar to below. Not exact same

Graphical user interface, text, application, table

Description automatically generated

Hint: Join

I followed the question’s instructions and NOT the picture.

SELECT e.employee\_id, e.last\_name, e.salary, j.job\_title, d.department\_id, d.department\_name, l.city, c.country\_id

FROM HR\_EMPLOYEES e

JOIN HR\_JOBS j ON e.job\_id = j.job\_id

JOIN HR\_DEPARTMENTS d ON e.department\_id = d.department\_id

JOIN HR\_LOCATIONS l ON d.location\_id = l.location\_id

JOIN HR\_COUNTRIES c ON l.country\_id = c.country\_id

WHERE c.country\_id = 'US'

AND e.salary BETWEEN 3400 AND 9500

ORDER BY e.employee\_id;  
  
A screenshot of a computer

AI-generated content may be incorrect.

**Question #9**

Explain Variable Scope and visibility and provide some examples of codes with Nested Block version

**Question #10**

Explain the Scalar variables?

a)What type of data we can store them?

b)Which other data types we can store in them (like CHAR is one of them how about others? ..)

c)How and Why we use %TYPE… Provide some examples.

d) How do you assign initial values to a variable ? give two options… If you do not have any initial value then what is going to be first value for those variable which you haven’t assign anything?