DBS311 – Advanced Database Services



**TERM– WINTER 2022**

**Course & Section Code:**

**Course Name:**

**Q&A / Virtual Office Hour:**

**Instructor Name & Email:**

**DBS311NCC**

**Advance Database Services**

**Assigment1**

**6:30 PM – 9:30 PM**

**Slavica O’Connor: slavica.oconnor@senecacollege.ca**

NAME-ANUVANSHIKA

ID-153693205

[EMAIL-aanuvanshika@myseneca.ca](mailto:EMAIL-aanuvanshika@myseneca.ca)

**Assignment Due**: Saturday, February 26th, 2022, 11:59pm.

**Submission**

Your submission can be a single Word-based with appropriate header and commenting. Please ensure your file has SQL commands and also as well as screen shots executed in SQL Developer.

TASKS

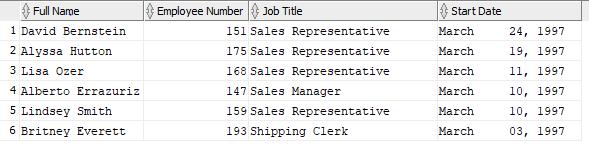
* Each task is worth 10 mark total 100.
* For each question, the columns’ title and the format of the output result must match the sample result given in that question.
* For below tasks you need to use all three schemas that you created in class. o DBS311\_Schema\_Buld\_2.sql

o HumanResources(HR)\_Database\_Build.sql o JustLee\_Book\_Order\_DatabaseBuild\_2.sql

* You can find this file inside **Assigment1 folder**

DBS311 – Advanced Database Services

1. Display the full employee name, employee number, job title, and hire date of all employees hired in March 1997 with the most recently hired employees displayed first. Use HR schema (HR\_ table(s)).



SELECT (first\_name ||' '|| last\_name) AS "Full Name",employee\_id AS "Employee Number",

j.JOB\_TITLE,

TO\_CHAR(hire\_date,'Month DD, YYYY') AS "Start Date"

FROM hr\_employees e

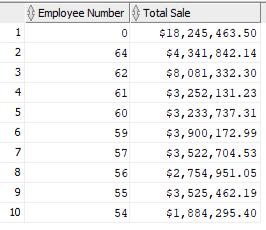
INNER JOIN HR\_JOBS j

ON e.job\_id=j.job\_id

WHERE EXTRACT(MONTH FROM hire\_date)=3 AND EXTRACT(YEAR FROM hire\_date)=1997

ORDER BY "Start Date" DESC;

1. The company wants to see the total sale amount per sales person (salesman) for all orders. Assume that online orders do not have any sales representative. For online orders (orders with no salesman ID), consider the salesman ID as 0. Display the salesman ID and the total sale amount for each employee. Sort the result according to the total sales from the highest to the lowest.



SELECT NVL(O.SALESMAN\_ID, 0) AS "Employee Number",TO\_CHAR(SUM(OD.QUANTITY \* OD.UNIT\_PRICE),'$99,999,999.00') AS "Total Price"

FROM ORDERS O

RIGHT JOIN ORDER\_ITEMS OD

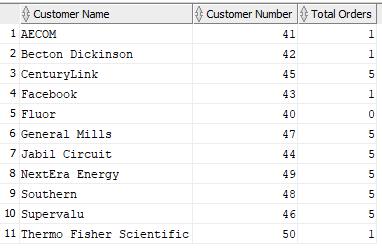
ON O.ORDER\_ID = OD.ORDER\_ID

GROUP BY O.SALESMAN\_ID

ORDER BY SALESMAN\_ID DESC;

DBS311 – Advanced Database Services

1. Display the customer name, customer id, and total number of orders for customers that the value of their customer ID is in values from 40 to 50. Include the customers with no orders in your report if their customer ID falls in the range 40 and 50. Sort the result by the customer name.



SELECT cu.NAME,cu.CUSTOMER\_ID,COUNT(o.ORDER\_ID) AS "Total Orders"

FROM CUSTOMERS cu

LEFT JOIN orders o

ON cu.CUSTOMER\_ID = o.CUSTOMER\_ID

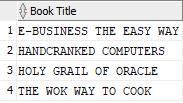
WHERE cu.CUSTOMER\_ID BETWEEN 40 AND 50

GROUP BY cu.CUSTOMER\_ID,cu.name

ORDER BY cu.NAME

1. List the title of all books in the same category as books previously purchased by customers 1018, 1019, 1020. Don’t include books those three customers have already purchased. Use

JustLee Book schema 2 (B\_ table(s)).



SELECT TITLE

FROM b\_books

WHERE

ISBN NOT IN

(SELECT oi.ISBN

FROM b\_orders o JOIN b\_orderitems oi ON o.ORDER#=oi.ORDER#

WHERE CUSTOMER# IN (1018,1019,1020))

AND CATEGORY IN

(SELECT DISTINCT o.CATEGORY

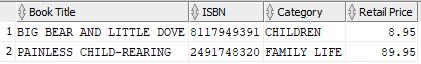
FROM b\_books o JOIN b\_orderitems oi ON o.ISBN=oi.ISBN

JOIN b\_orders os ON oi.ORDER#=os.ORDER#

WHERE os.CUSTOMER# IN(1018,1019,1020))

ORDER BY TITLE

1. Display the least and the most expensive book (retail). Display the book title, ISBN, category, and the price. Use JustLee Book schema 2 (B\_ table(s)).



SELECT TITLE AS "BOOK TITLE",ISBN,CATEGORY,RETAIL AS "RETAIL PRICE"

FROM B\_BOOKS

WHERE RETAIL =(SELECT MIN(RETAIL)

FROM B\_BOOKS)

UNION

SELECT TITLE AS "BOOK TITLE",ISBN,CATEGORY,RETAIL AS "RETAIL PRICE"

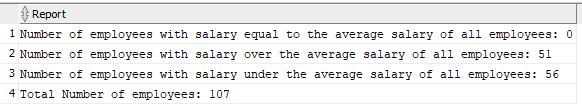
FROM B\_BOOKS

WHERE RETAIL =(SELECT MAX(RETAIL)

FROM B\_BOOKS)

DBS311 – Advanced Database Services

1. Write an SQL query to display the number of employees with salary equal to the average salary of all employees, the number of employees with salary over the average salary of all employees, the number of employees with salary under the average salary of all employees, and the total number of employees. See the format of the following result. Use HR schema (HR\_ table(s)).



SELECT 'Number of employees with salary equal to the average salary of all employees: '|| COUNT(\*) AS "REPORT"

FROM HR\_EMPLOYEES

WHERE SALARY = (SELECT AVG(SALARY)

FROM HR\_EMPLOYEES)

UNION

SELECT 'Number of employees with salary over to the average salary of all employees: '|| COUNT(\*) AS "REPORT"

FROM HR\_EMPLOYEES

WHERE SALARY > (SELECT AVG(SALARY)

FROM HR\_EMPLOYEES)

UNION

SELECT 'Number of employees with salary under to the average salary of all employees: '|| COUNT(\*) AS "REPORT" FROM HR\_EMPLOYEES

WHERE SALARY < (SELECT AVG(SALARY)

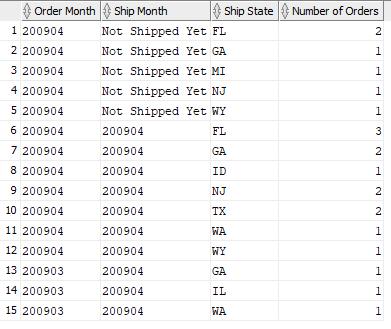
FROM HR\_EMPLOYEES)

UNION

SELECT 'Total Number of employees: '|| COUNT(\*) AS "REPORT"

FROM HR\_EMPLOYEES

1. Display the total number of book orders per month of order, month of shipment, and state. Sort the result from the most recent date to the latest, then state in the alphabetical order. Use JustLee Book schema 2 (B\_ table(s)).



DBS311 – Advanced Database Services

select

to\_char(orderdate, 'YYYYMM') as "order month",

(

case

when

shipdate is not null

then

to\_char(shipdate, 'YYYYMM')

else

'Not Shipped Yet'

end

)

shipstate count(order # ) as "ship month", as "ship state", as "number of orders"

from

z\_orders

group by

to\_char(orderdate, 'YYYYMM'),

(

case

when

shipdate is not null

then

to\_char(shipdate, 'YYYYMM')

else

'Not Shipped Yet'

end

)

, shipstate

order by

"order month" desc, "ship month" desc;

1. Display products id, product name, list price of all products that have not been ordered by any customers. Sort the result according to list price from highest value to the lowest.

SELECT PRODUCT\_ID,PRODUCT\_NAME,LIST\_PRICE AS PRICE

FROM PRODUCTS

MINUS

SELECT DISTINCT PRO.PRODUCT\_ID,PRODUCT\_NAME,LIST\_PRICE AS PRICE

FROM PRODUCTS PRO

INNER JOIN ORDER\_ITEMS OI

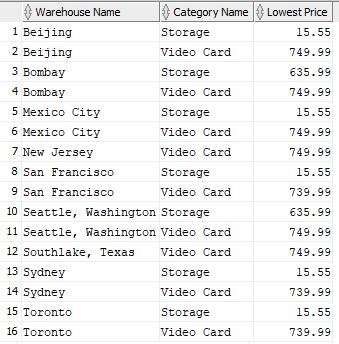
ON PRO.PRODUCT\_ID=OI.PRODUCT\_ID

ORDER BY PRICE DESC;



DBS311 – Advanced Database Services

1. Display warehouse name, product category name, and the lowest list price for this combination.
   * In your result, include the rows that the lowest list price is less than $100.
   * Also, include the rows that the lowest list price is more than $600.
   * Sort the output according to warehouse name and then product category name.



SELECT ws.warehouse\_name "Warehouse Name",pc.category\_name "Category Name", TO\_CHAR(MIN(p.list\_price)) "Lowest Cost"

FROM warehouses ws

JOIN inventories j

ON ws.warehouse\_id=j.warehouse\_id

JOIN products p

ON j.product\_id=p.product\_id

JOIN product\_categories pc

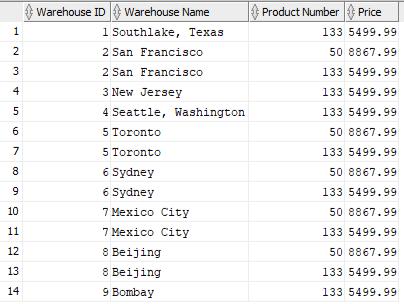
ON p.category\_id=pc.category\_id

GROUP BY ws.warehouse\_name, pc.category\_name

HAVING MIN(p.list\_price) NOT BETWEEN 100 AND 600

ORDER BY 1, 2, 3;

1. Find warehouses with the top two most expensive products. Display warehouse id, warehouse name, product id, and list price. Sort the output according to Warehouse id, then list price from the highest to the lowest price.



select

w.warehouse\_name "warehouse name",

z.category\_name "category name",

to\_char(min(p.list\_price), '$999,999,999.99') "lowest cost"

from

warehouses w

join

inventories x

on w.warehouse\_id = x.warehouse\_id

join

products a

on x.product\_id = p.product\_id

join

product\_categories z

on a.category\_id = c.category\_id

group by

w.warehouse\_name,

z.category\_id,

z.category\_name

having

min(p.list\_price) < 100

or min(p.list\_price) > 600

order by

w.warehouse\_name,

z.category\_name;