CS443 -- A3

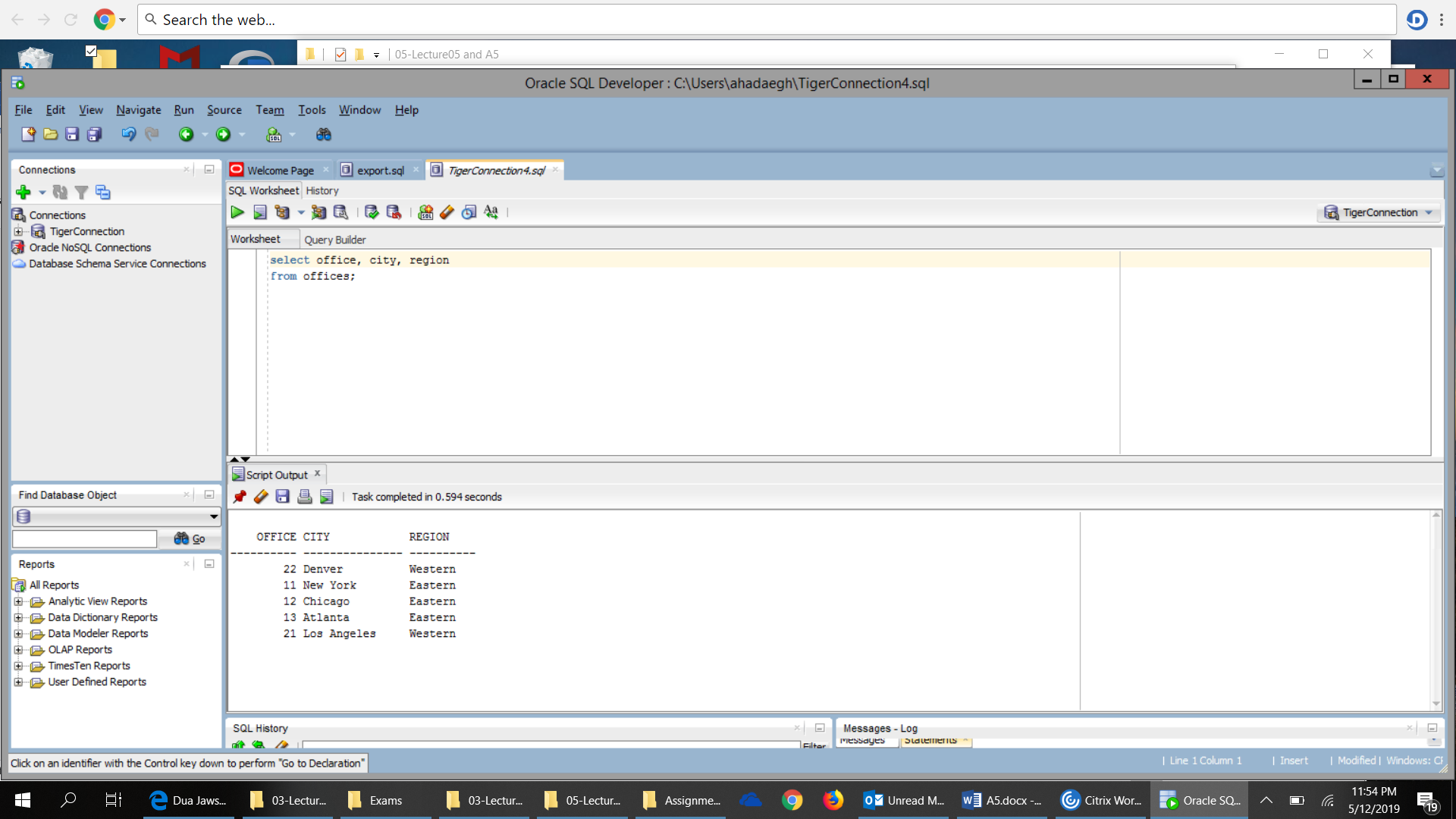
**Write the queries necessary to obtain the required information. Make sure all columns you return have descriptive column headings. You must show the result of the query. For example, if the query is:**

* Show the office id, the city, and the region
* Your query should be:

**select office, city, region**

**from offices;**

* and then you need to show the following on the screen: (snapshot of your result)



In this assignment, some questions require using “Delete”, “Insert”, and “Update”. Thus, if a table is modified, the modification may be cascaded to other questions. To avoid that, if a question changes table X, before doing the next question, drop and recreate table X. This ensures that all questions will be executed on original unchanged tables.

The following are the questions for A3.

1. Return the Minimum and Maximum Target for all offices.

SELECT MIN(offices.target) AS “Min\_Target”, MAX(offices.target) AS “Max\_ Target”

FROM offices;

A picture containing website

Description automatically generated

1. Determine how many orders were made in 2022. Return the number of rows that meet this condition.

SELECT COUNT(\*) AS “ORDERS FROM 2022”

FROM ORDERS

WHERE ORDERS.ORDER\_DATE LIKE ‘%22’;

Shape

Description automatically generated with low confidence

1. How many different titles in the sales reps table.

SELECT COUNT(DISTINCT salesreps.title) AS “Diff\_Title\_of\_Salesreps”

FROM salesreps

WHERE salesreps.title IS NOT NULL;

A picture containing chart

Description automatically generated

1. What is the average sales for salesreps in office 22.

SELECT AVG(sales) AS "Avg\_Sales\_for\_Office\_22"

FROM salesreps

WHERE rep\_office IN (‘22’);



1. What is the average sale amount for each sale reps in each office. Null should be ignored

SELECT AVG(salesreps.sales) AS "Avg\_Sale\_Amount", salesreps.rep\_office

FROM salesreps, offices

WHERE salesreps.rep\_office = offices.office and salesreps.sales is not null

GROUP BY salesreps.rep\_office;

Text

Description automatically generated

1. For each salesrep that has made an order, list the minimum, maximum and average order amount for all their orders. Include only those orders made anytime from 2020-2021. Omit from the list any salesrep that has only made 1 order in this time frame. Sort the results by Empl\_Num.

SELECT salesreps.empl\_num, MIN(ORDERS.AMOUNT) AS "Min\_Amount", MAX(ORDERS.AMOUNT) AS "Max\_Amount", AVG(ORDERS.AMOUNT) AS "Avg\_Amount"

FROM salesreps, ORDERS

WHERE salesreps.empl\_num = ORDERS.REP AND EXTRACT(year FROM ORDERS.ORDER\_DATE) BETWEEN 2020 AND 2021 --include orders made anytime from 2020-2021

GROUP BY salesreps.empl\_num

HAVING COUNT(ORDERS.ORDER\_NUM) > 1 -- Omit from any salesreps that made 1 order

ORDER BY salesreps.empl\_num;

Table

Description automatically generated

1. Use a sub-query to list the Customer number; Name and Credit Limit of any customers who have exceeded their credit limit (amount > credit limit) on any order.

SELECT customers.CUST\_NUM, customers.COMPANY, customers.CREDIT\_LIMIT

FROM customers

WHERE customers.CREDIT\_LIMIT < ANY

(SELECT ORDERS.AMOUNT

FROM ORDERS

WHERE ORDERS.CUST = customers.CUST\_NUM);

A picture containing shape

Description automatically generated

1. Use a subquery and using the “all” keyword to find the customer number, Salesrep id, and CreditLimit of every customer whose CreditLimit is larger than the CreditLimit of all of the customers of sales rep number 109.

SELECT customers.CUST\_NUM, salesreps.empl\_num, customers.CREDIT\_LIMIT

FROM customers, salesreps

WHERE customers.CUST\_REP = salesreps.empl\_num AND customers.CREDIT\_LIMIT > ALL

(SELECT customers.CREDIT\_LIMIT

FROM customers

WHERE customers.CUST\_REP = 109);

Table

Description automatically generated

1. Do question 8, still using the subquery but do not use the “all” keyword.

SELECT customers.CUST\_NUM, salesreps.empl\_num, customers.CREDIT\_LIMIT

FROM customers, salesreps

WHERE customers.CUST\_REP = salesreps.empl\_num AND customers.CREDIT\_LIMIT >

(SELECT MAX(customers.CREDIT\_LIMIT)

FROM customers

WHERE customers.CUST\_REP = 109);

Table

Description automatically generated

1. Use sub query and “in” keyword to print the salesreps (ids) who have taken order for the companies starts with letter ‘Z’ or with letter ‘J’. Duplicate rows are not allowed

SELECT DISTINCT salesreps.empl\_num

FROM salesreps

WHERE salesreps.empl\_num IN

(SELECT customers.cust\_rep

FROM customers

WHERE customers.company LIKE 'Z%' OR customers.company LIKE 'J%');

Table

Description automatically generated

1. Use sub query to find the id and the name of every sales rep that represents at least one customer with a credit limit of greater than $50,000.

SELECT salesreps.empl\_num, salesreps.name

FROM salesreps

WHERE salesreps.empl\_num IN

(SELECT customers.CUST\_REP

FROM customers

WHERE customers.CREDIT\_LIMIT > 50000.00);

Text

Description automatically generated

1. Use sub query and keyword “exists” to list the id and the name of the salesreps in which some customers have orders some products in their hiredate.

SELECT salesreps.empl\_num, salesreps.name

FROM salesreps

WHERE EXISTS

(SELECT ORDERS.REP

FROM ORDERS

WHERE salesreps.hire\_date = ORDERS.ORDER\_DATE);

**Text

Description automatically generated**

1. List all the products (only Product\_ID) that have never been sold.

SELECT PRODUCT\_ID

FROM PRODUCTS

WHERE NOT EXISTS

(SELECT ORDERS.QTY

FROM ORDERS

WHERE ORDERS.PRODUCT = PRODUCTS.PRODUCT\_ID);

Table

Description automatically generated with medium confidence

1. Insert the following information into the OFFICES table:

# Office: 10 City: Miami Region: Southern Manager: 106 Sales: 0

* Target should be Null. Do not use explicit Null for the target in your insert statement.

*Recreate the orders table after doing the insert*

INSERT INTO offices (office, city, region, mgr, sales)

VALUES (10, 'Miami', 'Southern', 106, 0);

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

1. Write an insert statement to add Your Name as Empl\_Num 772. Use the date the insert is done for the hire date (sysdate). Sales is zero.

*Recreate the orders table after doing the insert*

INSERT INTO salesreps (empl\_num, name, hire\_date, sales)

VALUES (772, 'EJ Lilagan', 'sysdate', 0);

Graphical user interface, text, application, email

Description automatically generated

**Table, Excel

Description automatically generated**

1. Use subquery to Delete all orders for employees 'Dan Roberts'.

*Recreate the orders table after doing the delete*

DELETE FROM ORDERS

WHERE ORDERS.REP = 101;

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

1. Lower the price of the products by 10% if they are higher the average price

*Recreate the products table after doing the update*

UPDATE PRODUCTS

SET PRICE = (0.9 \* PRICE)

WHERE PRODUCTS.PRICE >

(Select avg(price)

from products); --avg(price)

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

1. Set the quota of the salesreps to (average of the quota) + 1500 if they are hired in 2021.

*Recreate the salesreps table after doing the update*

(incorrect)

update salesreps

set quota = 300000 + 1500 --set the 300000 to the avg(quota)

where hire\_date LIKE '%21';

(correct)

UPDATE salesreps

SET quota = 1500 +

(SELECT AVG(quota)

FROM salesreps)

WHERE hire\_date LIKE '%21';

Output: 2 rows updated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

1. Increase customers credit limit by 25% for all customers that have 3 or more orders in which each order is more than 500.

*Recreate the customers table after doing this update*

UPDATE customers

SET customers.CREDIT\_LIMIT = (1.25 \* customers.CREDIT\_LIMIT)

WHERE 3 <=

(SELECT COUNT(ORDERS.CUST)

FROM ORDERS

WHERE ORDERS.CUST = customers.CUST\_NUM AND AMOUNT > 500);

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

1. Increase the credit limit of any customer who has any order that exceeds their credit limit. The new credit limit should be set to their maximum order amount plus $1,000. This must be done in 1 SQL statement.

*Recreate the customers table after doing this update*

UPDATE customers

SET customers.credit\_limit =

((SELECT MAX(orders.amount)

FROM orders

WHERE orders.cust = customers.cust\_num) + 1000)

WHERE customers.credit\_limit < ANY

(SELECT orders.amount

FROM orders

WHERE orders.cust = customers.cust\_num);

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated