Assignment 1

Due date: Oct 18, 2017

7% of final mark

Total Marks 100

Your group of two or three people have to solve the following Queries. You have to explain your work during lab on Oct 18.

Assignment must be done in a group. Penalty of 10% applies if done individually.

You must submit the Submissions Form (below) with your assignment (one per group) on the blackboard.

Student Assignment Submission Form

I/we declare that the attached assignment is my/our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students

.

Name(s) Student ID(s) Signature

1 Elena Khoroshun 101908168

2 Lazaros Mitro 115350167

Create the following table and insert values in it according to following table.

Note: Employees is work table for these queries

Table: Employees

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Employee First\_Name Last\_Name Dept Hire\_Date Credit Phone Manager\_id

Id Code Limit Ext

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

201 Susan Brown Exe 01-Jun-1998 $30.00 3484 (null)

202 Jim Kern Sal 16-Aug-1999 $25.00 8722 201

203 Martha Woods Shp 02-Feb-2004 $25.00 7591 201

204 Ellen Owens Sal 01-Jul-2003 $15.00 6830 202

205 Henry Perkins Sal 01-Mar-2000 $25.00 5286 202

206 Carol Rose Act null null null (null)

207 Dan Smith Shp 01-Dec-2004 $25.00 2259 203

208 Fred Campbell Shp 01-Apr-2003 $30.00 1752 203

209 Paula Jacobs Mkt 17-Mar-1999 $15.00 3357 201

210 Nancy Hoffman Sal 16-Feb-2004 $25.00 2974 203

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CREATE TABLE employees**

**( employee\_id NUMBER(6)**

**, first\_name VARCHAR2(20)**

**, last\_name VARCHAR2(25)**

**CONSTRAINT emp\_last\_name\_nn NOT NULL**

**, dept\_code CHAR(3)**

**, hire\_date DATE**

**, credit\_limit NUMBER(4,2)**

**, phone\_ext NUMBER(4)**

**, manager\_id NUMBER(3)**

**) ;**

>> Write the query for the following tasks:

#1. List the following columns of the Employees table in this order:

first name, last name, hire\_date, phone\_ext and department code

Change the name of the hire\_date column to Date\_of\_Joining within the result table. List the employees from the Sales and Marketing department only.

Sort the rows of the result table by the last\_name column in descending order.

**select first\_name as "first name", last\_name as "last name",**

**hire\_date as "Date\_of\_Joining", phone\_ext as "phone ext", dept\_code**

**as "department code"**

**from employees**

**where dept\_code IN ('Sal', 'Mkt')**

**order by last\_name desc;**

#2. Write a SQL statement to add a new employee Rick Adam to the Employees table. Rick joined the Accounting department.

His date of joining is 18th Sept, 2011. He works for manager with id: 203. He is allotted any phone extension - 3757.

His credit limit is set to $35.00

-----

SQL QUery:

**. INSERT INTO**

**employees VALUES**

**(206,**

**'Rick'**

**, 'Adam'**

**, 'Act'**

**, TO\_DATE('18-Sep-2011', 'dd-MON-yyyy')**

**, 35**

**, 3757**

**, 203**

**);**

#3. List the employees that have a phone number starting with 7 or whose names are starting with letter - 'N'.

Show the employee\_id, first\_name, last\_name and phone\_ext. Sort the rows by employee\_id.

-----

SQL QUery:

**select employee\_id, first\_name, last\_name, phone\_ext**

**from employees**

**where first\_name like 'N%' OR phone\_ext like '7%'**

**order by employee\_id;**

#4. Using the Employees table, list the following columns:

dept\_code, credit\_limit, last\_name, first\_name

Place the columns in that order. Sort the rows by :

dept\_code in ascending order

credit\_limit in descending order

last\_name in ascending order.

And get only those employees who are having credit limits between 20 and 40.

------

Sql Query:

**select dept\_code, credit\_limit, last\_name, first\_name**

**from employees**

**where credit\_limit between 20 and 40**

**order by dept\_code asc, credit\_limit desc, last\_name asc;**

#5. From the Employees table, list the employee\_id, first\_name, last\_name and credit\_limit columns for employees

with the first names: Martha, Carol, Nancy

Sort the rows by the last name.

------

SQl Query:

**select employee\_id, first\_name, last\_name, credit\_limit**

**from employees**

**where first\_name in ('Martha', 'Carol', 'Nancy')**

**order by last\_name;**

#6. From the Employees table, get the department and credit limits for the employees 'Jim Kern'.

------

SQl Query:

**select dept\_code, credit\_limit**

**from employees**

**where first\_name in ('Jim') AND last\_name in ('Kern');**

#7. List the department and phone extension for all the employees who don't have a manager.

Sort the rows by the employee\_id.

----

Sql Query:

**select dept\_code, phone\_ext**

**from employees**

**where manager\_id is NULL**

**order by employee\_id;**

#8. Write a query to display distinct department codes credit\_limits.

**select distinct dept\_code, credit\_limit**

**from employees;**

#9. Delete all the employees who work for Shipping Department or who have manager\_id null.

----

SQl Query:

**delete from employees**

**where dept\_code = 'Shp' or manager\_id is NULL;**

#10. Increase the credit limits of all the employees by $15.00 who work for Accounting department.

----

SQl Query:

**update employees**

**set credit\_limit = credit\_limit + 15**

**where dept\_code = 'Act';**

#11. Create sequence to uniquely identify each row in the employees table. Employee\_id for the employee table:

Start with 220; do not allow caching of the values. Name the sequence Employee\_id Seq. Verify the existence of sequence in the Data Dictionary.

----

SQL Query:

**. create sequence employee\_id\_seq**

**start with 220**

**nocache;**

#12. Write a SQL statement to add a new employee John Adam to the Employees table. John joined the Accounting department.

His date of joining is 18th Sept, 2011. He works for manager with id: 203. He is allotted any phone extension - 3758.

His credit limit is set to $40.00. Use the sequence to uniquely identify the new employee.

-----

SQL QUery:

**INSERT INTO**

**employees VALUES**

**(employee\_id\_seq.nextval,**

**'John'**

**, 'Adam'**

**, 'Act'**

**, TO\_DATE('18-Sep-2011', 'dd-MON-yyyy')**

**, 40**

**, 3758**

**, 203**

**);**

Task#1:

For each manager\_id, list the number of employees ( except the Shipping Department ) each one manages.

Also list the range of their employees's credit limits by showing the minimum and maximum.

Omit the employee having first name: Susan and last name: Brown

>> Sql Query:.......

**. select count(manager\_id), manager\_id, max(credit\_limit), min(credit\_limit)**

**from employees**

**where dept\_code not in 'Shp' and**

**first\_name not in 'Susan' and last\_name not in 'Brown'**

**group by manager\_id;**

Task#2:

For each department, find the sum of the credit\_limits. Display only those departments

whose total credit limits are greater than $50.00 but less then $100.00.

>> Sql Query:

**select sum(credit\_limit), dept\_code**

**from employees**

**having sum(credit\_limit) between 50 and 100**

**group by dept\_code;**

Task#3: For each manager\_id between 202and 203, show the number of employees each manager supervises.

Show only those managers who manage more then 2 employees.

>> SQL Query:

**select count(manager\_id), last\_name**

**from employees**

**where manager\_id between 202 and 203**

**having count(manager\_id) > 2**

**group by last\_name;**

Task#4: Find how many employees have joined between the year 2002 and 2005 in the accounting department.

**select hire\_date from employees where hire\_date between '02-01-01' and '05-12-31'**

**and dept\_code in 'Act'**

Task#5: List all the employees, their hire dates and the number of days each person will have worked

for the company as of January 1, 2005.

**. select last\_name, hire\_date, (sysdate - hire\_date) as "day worked"**

**from employees**

**where hire\_date > '05-01-01';**

Task#6: Find out in which month the employee - Carol Rose joined.

Sql Query:

**select first\_name, last\_name, to\_char(hire\_date, 'dd-MON-yyyy')**

**from employees**

**where first\_name like 'Carol%' and**

**last\_name like 'Rose%';**

Task#7: Write the Sql statement for the following:

a) Create a table named : Foods which has the following columns:

menu\_item as numeric of size 2,

supplier\_id as character type of 3,

product\_code as character type of 2,

description varchar2(20),

price as numeric with scale and precision of 4, 2,

price\_increase as numeric with scale and precision of 4, 2

Sql Query:

**CREATE TABLE foods**

**( menu\_item NUMBER(2)**

**, supplier\_id VARCHAR2(3)**

**, product\_code VARCHAR2(2)**

**, description VARCHAR(20)**

**, price NUMBER(4,2)**

**, price\_increase NUMBER(4,2)**

**) ;**

b) Rename the product\_code to product\_id.

Sql Query:

**alter table foods**

**rename column product\_code to product\_id;**

c) Add a check constraint to price column ensuring that price of all the items are less then $10.00

Sql Query:

**alter table foods**

**add constraint price\_check check (price < 10);**

d) Add a unique constraint to description column.

Sql Query:

**. alter table foods**

**add constraint desc\_un unique (description);**