

# CSC3170 Introduction to Database Systems

## (Spring 2024) Assignment 2 – SQL PART

Please answer all the questions below and submit your answer to blackboard

### 1. Connection to Oracle DB

You should first connect to the linux host with your CSE account, e.g. linux 1 (images show linux 12)  
The login command is `ssh CSEaccount@linux12.cse.cuhk.edu.hk`

```
C:\Users\qyli21>ssh qyli21@linux12.cse.cuhk.edu.hk
qyli21@linux12.cse.cuhk.edu.hk's password:
Last login: Mon Nov  7 14:51:20 2022 from vpngw2.cse.cuhk.edu.hk

-----
Resources limit applied:

- Max number of logins per user: 10
- Max number of processes per user: 1024
- Max resident memory per session: 64GB
-----

(base) linux12:/research/d4/gds/qyli21> _
```

Before you connect to the database, please first transfer the data folder to the server and enter the data fold. After that, type

- `ssh`
- `source /opt1/oracle122/setup`
- `sqlplus oracle_account@db18.cse.cuhk.edu.hk`

Then input the oracle password. (Please note **the case of the password**)

```
(base) linux12:/research/d4/gds/qyli21> ssh
linux12:/research/d4/gds/qyli21> source /opt1/oracle122/setup
linux12:/research/d4/gds/qyli21> sqlplus h170@db18.cse.cuhk.edu.hk

SQL*Plus: Release 12.2.0.1.0 Production on Mon Nov 7 15:35:18 2022

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Enter password:
Last Successful login time: Mon Nov 07 2022 14:54:30 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

SQL>
```

The SQL part of assignment 2 is meant to run on terminal. Even if you do this assignment based on the Oracle workbench, please submit your final answer by checking it's operation on the terminal, otherwise, some commands may not work.

Once you have successfully connected to the sql server through the terminal, you can then create the tables and add data entries using the files in the data fold. You may SCP the files to the directory, type '@create\_table.sql' in the terminal and then type '@add.sql' in the terminal. After that, the database is created successfully and you can complete your assignment now.

If unable, simply copy and execute the SQL commands from the sample data directly into the terminal.

## **2. Background**

A software company sells and develops software applications. The company has many offices and each office has a unique ID. Each engineer is assigned to one of the offices. An engineer has his own assignments, such as conducting research on a project, or maintaining a software application (it is possible for him to have multiple assignments). Each project belongs to a category and has its own budget and expenditure. Each software applications must be associated with one project.

## **3. Schema**

OFFICE (OFFICEID, ONAME, OADDRESS, OPHONENUM)

ENGINEER (ENGINEERID, ENAME, ESALARY, OFFICEID)

PROJECT (PROJECTID, OFFICEID, PNAME, PCATEGORY, PBUDGET, PEXPENDITURE)

APPLICATION (APPID, ANAME, APRICE, ADATE, PROJECTID)

MAINTAIN (APPID, ENGINEERID) RESEARCH (PROJECTID, ENGINEERID)

## 4. Description

### OFFICE

Field Name	Format	Description
OFFICEID	Integer	The ID of the office. It is the primary key.
ONAME	255 Characters	The name of the office.
OADDRESS	255 Characters	The address of the office.
OPHONENUM	100 Characters	The phone number of the office.

### ENGINEER

Field Name	Format	Description
ENGINEERID	Integer	The ID of the engineer. It is the primary key.
ENAME	255 Characters	The name of the engineer.
ESALARY	Integer	The salary of the engineer.
OFFICEID	Integer	The ID of the office that the engineer belongs to. May treat as foreign key referred to OFFICEID in OFFICE.

### PROJECT

Field Name	Format	Description
PROJECTID	Integer	The ID of the project. It is the primary key.
OFFICEID	Integer	The ID of the office that holds the project. May treat as foreign key referred to OFFICEID in OFFICE.
PNAME	255 Characters	The name of the project.
PCATEGORY	255 Characters	The category of the project.
PBUDGET	Integer	The budget of the project.
PEXPENDITURE	Integer	The expenditure of the project

### APPLICATION

Field Name	Format	Description
APPID	Integer	The ID of the application. It is the primary key
ANAME	255 Characters	The name of the application
APRICE	Integer	The price of the application
ADATE	Integer	The release date of the application. Stored as an 8-digit integer of the form yyyymmdd
PROJECTID	Integer	The ID of the project that this application is based on. May treat as foreign key referred to PROJECTID in PROJECT

### MAINTAIN

Field Name	Format	Description
APPID	Integer	The ID of the application. It is a foreign key referred to APPID in APPLICATION
ENGINEERID	Integer	The ID of the engineer. It is a foreign key referred to ENGINEERID in ENGINEER

### RESEARCH

Field Name	Format	Description
PROJECTID	Integer	The ID of the project. It is a foreign key referred to PROJECTID in PROJECT
ENGINEERID	Integer	The ID of the engineer. It is a foreign key referred to ENGINEERID in ENGINEER

## 5. Assignment 2 Queries

Based on the above information, write the following queries in SQL statements on the provided Oracle database account:

1. Find the **ENAME** of the engineer(s) who have the highest salary at office named "Compatibility Tools Group". The result should be sorted by **ENAME** in ascending order (assume name is unique)
2. Find the **ENAME** of the engineer(s) who have the highest salary at office named "Compatibility Tools Group". The result should be sorted by **ENAME** in ascending order (assume name is non-unique)
3. Find **OFFICEID** and **ONAME** of the office(s) of which the total budget of all the projects **greater than 20000**. The result should be sorted by **OFFICEID** in descending order. The ordering of the columns is shown below:

OFFICEID	ONAME
----------	-------

4. The **HEAD COUNT** of a project is the number of engineers working on it (through the research relation only). Now, for every project category, calculate the average **HEAD COUNT** of the projects for that category. The result should be sorted by the **PCATEGORY** name in descending order. The ordering of the columns is shown below: (Please only count the researchers and ignore the maintainers in this question. You may assume that every project must have at least one engineer working on it and you may need to use **AVG()** in your query. TIP: *GroupBy* needed.

PCATEGORY	HEAD_COUNT
-----------	------------

5. Find the **OFFICENAME**, **ENGINEERID** and **ENAME** of the project engineer who is working on the project(s) with the highest budget surplus (i.e.,  $PBUDGET - PEXPENDITURE$ ). The result should be sorted by **ENGINEERID** in ascending order. The ordering of the columns is shown below: (Please only count the researchers and ignore the maintainers in this question. TIP: You may need to perform arithmetic operation in your query.

OFFICENAME	ENGINEERID	ENAME
------------	------------	-------

6. Find the **APPID** and **ANAME** of application(s) who are maintained by more than or equal to 2 engineers. The result should be sorted by **APPID** in descending order. The ordering of the columns is shown below:

APPID	ANAME
-------	-------

## 6. Submission Procedure

You should **STRICTLY** follow this procedure to submit all your SQL queries or you may receive mark deduction. Assume your name is “Chan Tai Man” and your student ID is 1101234567. The submission procedures are shown as follows:

1. Save a copy of your commands in a single file called **<your\_student\_ID>.sql** (e.g. 1101234567.sql) for all of the above queries and save the query results to the files result1.lst, result2.lst, ..., result5.lst for queries 1, 2, ..., and 6 respectively using the Spool command in Oracle(see the example shown below).

**You should use comment lines to include your name and student ID at the header** of 1101234567.sql. You should also use the Oracle command Spool for each of the queries. Do NOT add any comment lines inside your SQL statements. There is always at least one space between your comment body and /\* (or \*/). Your 1101234567.sql should be in the following format:

```
/*
Student ID: 1101234567 Name: Chan Tai Man */

/* Query 1 */

Spool result1.lst
Select ... from ... ;
Spool off

/* Query 2 */

Spool result2.lst
Select ... from ... ;
Spool off
.....
```

If you would like to create temporary tables, you can do so using the **CREATE OR REPLACE VIEW ... AS ...** statement. However, if you need to create views, DO NOT write the create and drop sql statements inside the body of Spool. The formatting should be (iff you use views):

```
/* Query 5 */
Create OR Replace view ... AS ...

Spool result5.lst
Select ... from ... ;
Spool off
Drop view temp;
```

You should ensure that your submitted file should not contain any special characters (e.g. ^M), which are resulted from transferring your files from Windows to Linux, by using a Linux command dos2unix on *linux* machines.

**You should test your final .sql file (e.g. 1101234567.sql) before submission by typing the command “source <your\_student\_ID>.sql;”** (e.g. source 1101234567.sql;) in your Oracle account on linux machine linux10-15. This should generate the result files result1.lst, result2.lst, ..., result8.lst in your current directory in Unix. **You have to ensure that the content of each result file is correct in order to get score for the query. Please note that we will test your sql file and only the correct results will get scores. The output format will not affect your final score.**

2. Submit your .sql file to the submission box on the blackboard. Please only submit this one file and name it accordingly.