REPORT

I Have completed my SQL lab with help of notes provided during the regular online classes as well as reference book shared in class that is “book **BeginningSQLqueries** by Clare Churcher”.

I got required the output (correct format) after rectifying silly error in almost each queries. I Refer website such as https://stackoverflow.com/, https://www.geeksforgeeks.org/ https://www.w3schools.com/ , to get know different ways of writing queies.

Department of Data Science, Bishop Heber College Tiruchirappalli
NoSQL Database Management Lab

Lab7. University Course Enrollment Data Anaytics

In this lab, you will use the dataset that contains 7 course enrolment data files of a university (course.data, dept.data, enroll.data, major.data, prof.data, section.data and student.data) that are given to you.

Please open these files in MS Excel and look at the record values. Understand the relationships between each table.

Write SQL queries for the following statements, execute them and obtain results. Compare the query results by manually checking the records and ensure your SQL query gives you correct result as you expected.

Write SQL queries for the following problems

Question1. Print the names of professors who work in departments that have fewer than 50 PhD students.

```
Select d.dname, d.num_phd, P.pname from prof P, dept d where
d.num_phd <50 order by num_phd;
```

Question 2. Print the names of the students with the lowest GPA.

```
Select sname, gpa from student where gpa = (select min(gpa)
from student);
```

Question3. For each Computer Sciences class, print the class number, section number, and the average gpa of the students enrolled in the class section.

```
Select e.cno, e.sect_no, avg(S.gpa) from enroll e, student S
where dname = 'computer sciences' and S.sid = e.sid group
by dname, cno, sect_no;
```

Question4. Print the names and section numbers of all sections with more than six students enrolled in them.

```
Select C.cname, C.cno, e.sec_no, count(e.sid) as
Student_count from course c left Join enroll e on e.cno=
C.cno group by C.cname, C.cno, e.sec_no having count(e.sid)>
6;
```

Question5. Print the name(s) and sid(s) of the student(s) enrolled in the most sections.

```
Select sname, sid from student where sid in (select sid from
enroll group by sid having count(*) >= all (select count(*)
from enroll group by sid));
```

Question6. Print the names of departments that have one or more majors who are under 18 years old.

```
Select distinct m.dname from major m, student s where
m.sid = s.sid and s.age < 18;
```

Question7. Print the names and majors of students who are taking one of the College Geometry courses.

```
Select m.sid, m.dname from major m inner join enroll e
on e.sid = m.sid where e.cno in (461, 462);
```

Question8. For those departments that have no major taking a College Geometry course print the department name and the number of PhD students in the department.

```
Select dept.dname, dept.num_phds from dept where
not exists (Select 1 from course where course.dname =
dept.dname and course.cname like '%College Geometry%');
```

Question9. Print the names of students who are taking both a Computer Sciences course and a Mathematics course.

```
Select S.sname from student s inner join enroll e on e.sid =
s.sid where e.dname = 'computer Sciences' and e.dname =
'Mathematics';
```

Question10. Print the age difference between the oldest and the youngest Computer Sciences major.

```
Select max(S.age) - min(S.age) as "age dif" from students
inner join major m on m.sid = S.sid where m.dname =
'computer Sciences';
```

Question11. For each department that has one or more majors with a GPA under 1.0, print the name of the department and the average GPA of its majors.

```
Select dname, cno, avg(grade) from enroll enroll group by
dname, cno;
```

Question12. Print the ids, names and GPAs of the students who are currently taking all the Civil Engineering courses.

```
Select e.sid, S.Sname, S.gpa from Student S right outer
Join enroll e on S.sid = e.Sid where e.dname = 'Civil Engineering'
order by gpa;
```

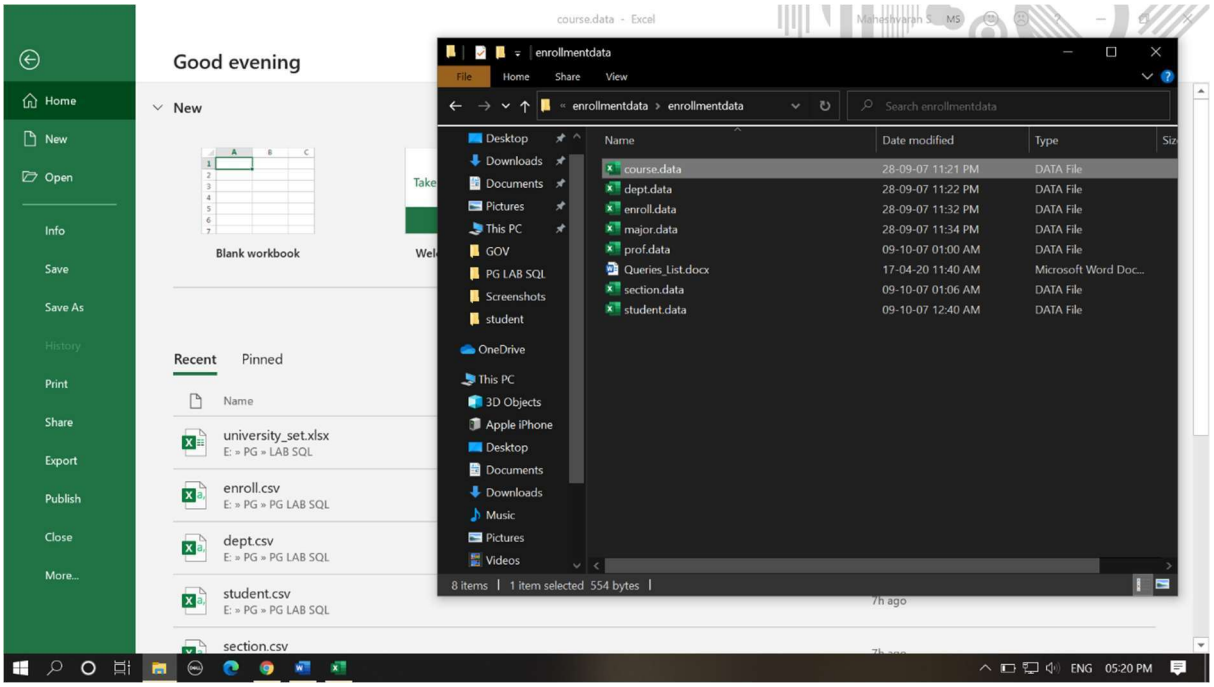
LAB7. UNIVERSITY COURSE ENROLLMENT DATA ANALYTICS PART-1

In this lab, you will use the dataset that contains 7 course enrolment data files of a university (Course.data, dept.data, enroll.data, major.data, prof.data, section.data and student.data) that are given to you.

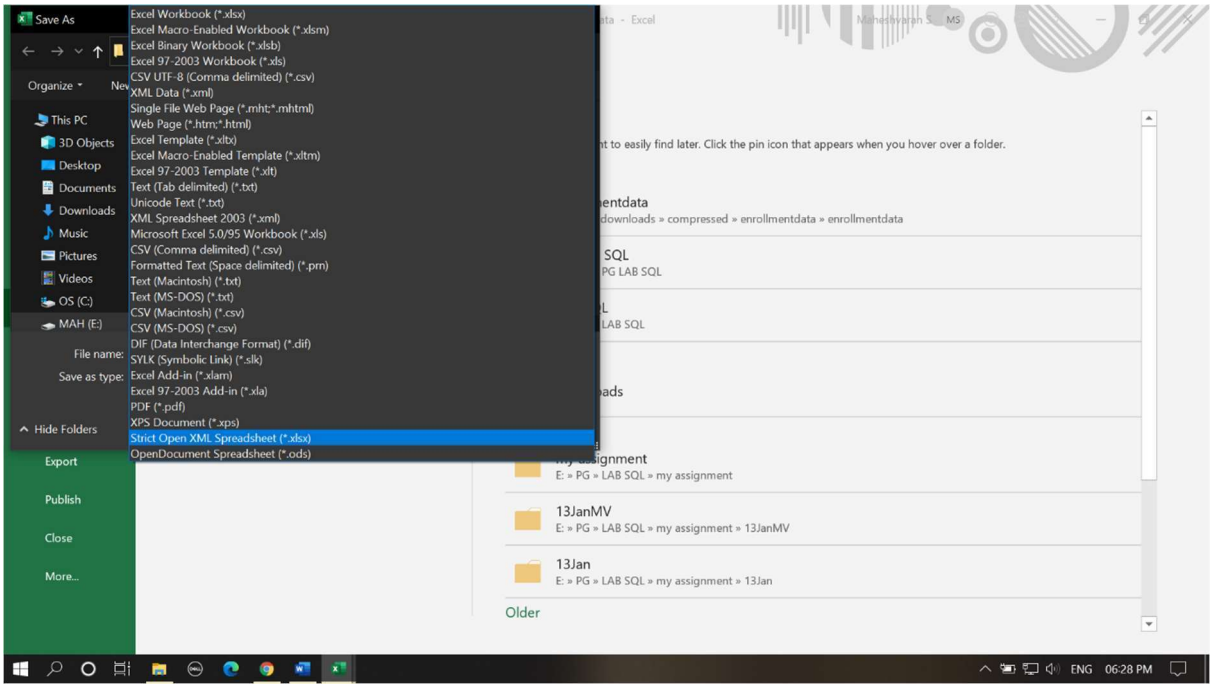
STEP 1:

We need to open these given ".data" format file in a excel. So just open ms excel or any other application which handle xls and csv that may be wps office or anything which related to this. Then just drag and drop the course.data and other ".data" files one by one you can save it as csv or xls file using save as option. Then you can work flawlessly.

Eg1:



Eg2:



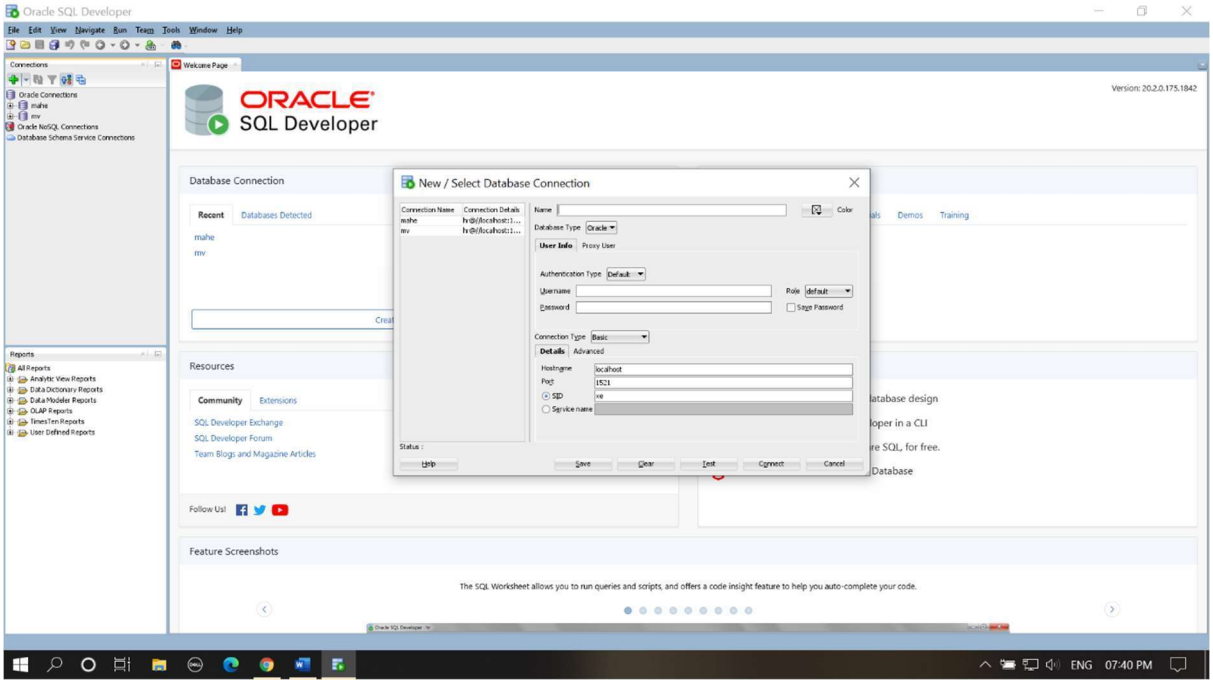
The above eg1, eg2 screenshot shows that you can save the data file into several format like xlsx, xls, csv and many so you can choose one of these and save the file with the name you want.

STEP 2:

After converting all these ".data" files into the excel file format we are ready to import our data to the database through sql developer.

The following steps involve to import the data:

Eg1:



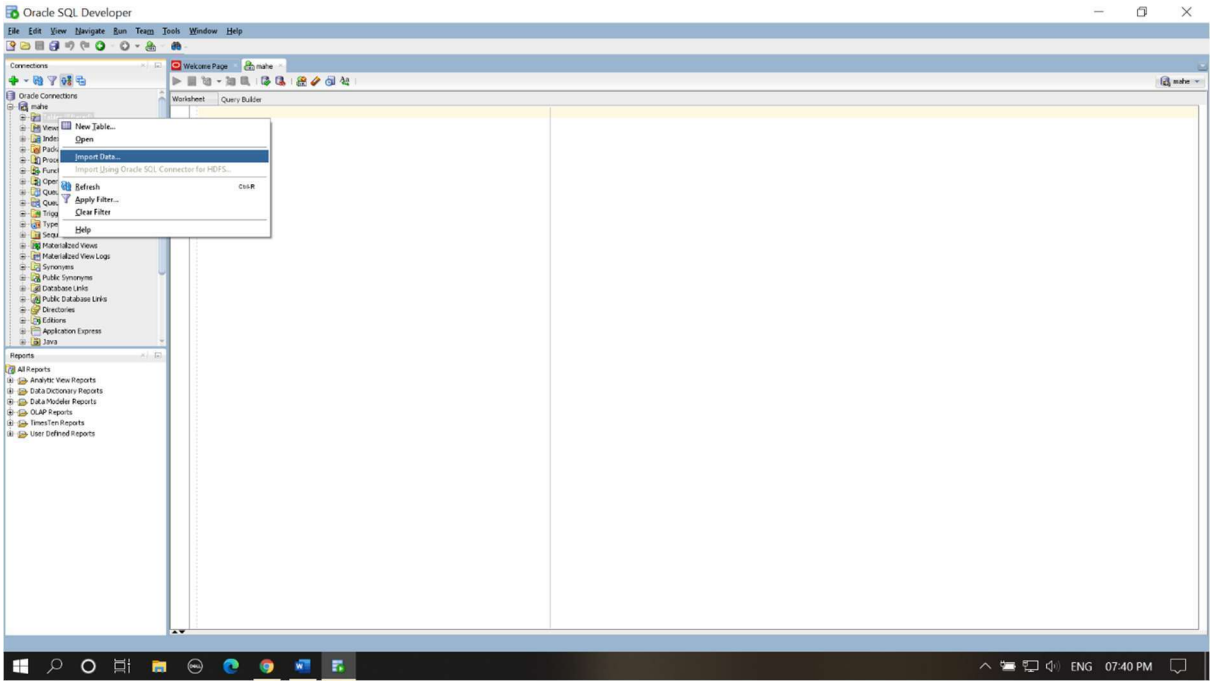
1. You need to download **SQL** developer from the following link and extract the file: https://www.filehorse.com/download-oracle-sql-developer-64/download/

2. You no need to install sql developer like any other software just click on the application icon with sql developer named file. The your sql developer starts load

3. The above eg1 file shows that you need to create a new connection then enter the user name and password for the schema you need to access.

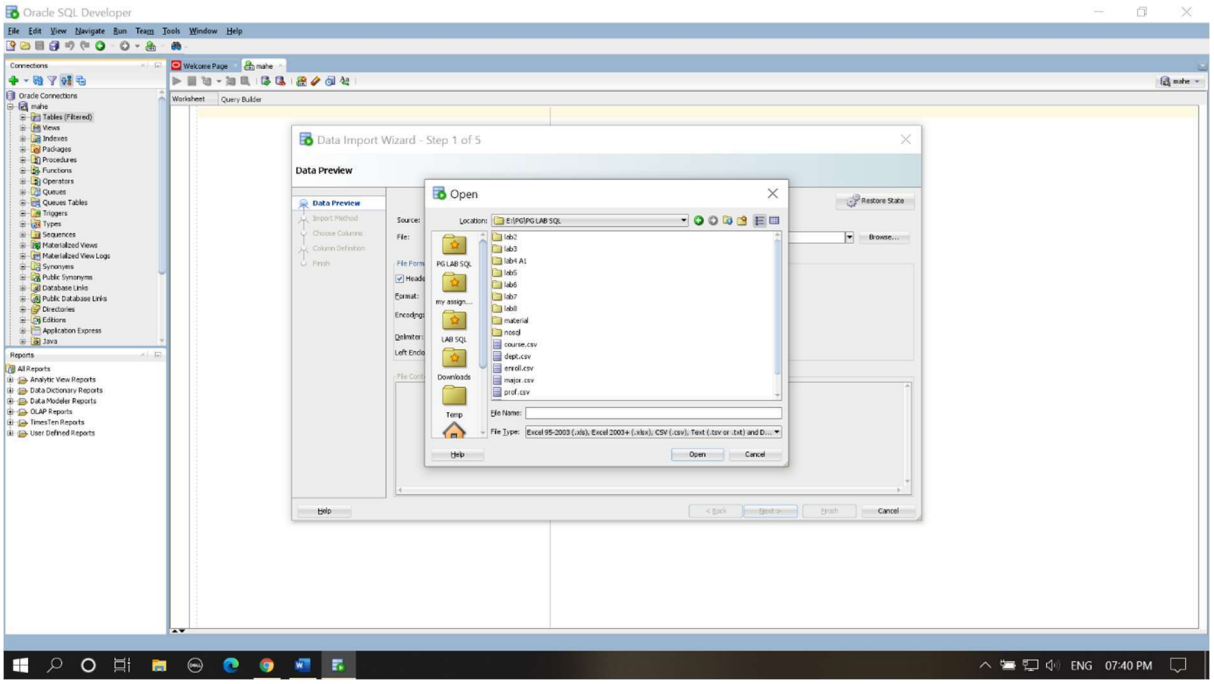
4. After completing all these steps you can test the connection using test button when it shows success then click on connect.

Eg2:

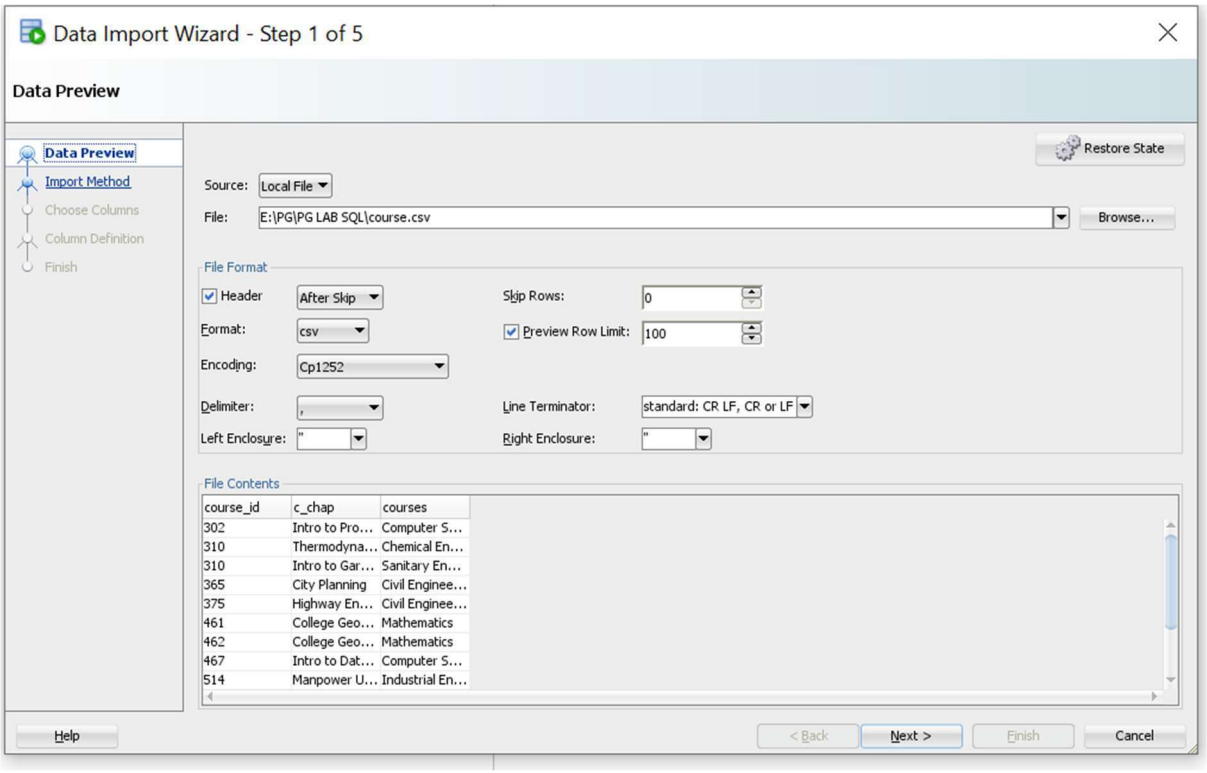


1. By clicking “+” icon on the left under the connections menu you can see the connection name what you gave. Just right click on table click import data then you can see browse button just click on that you will be prompt to open the file which you want to import to the database. Which shows below as like eg3:

Eg3:



STEP 3:



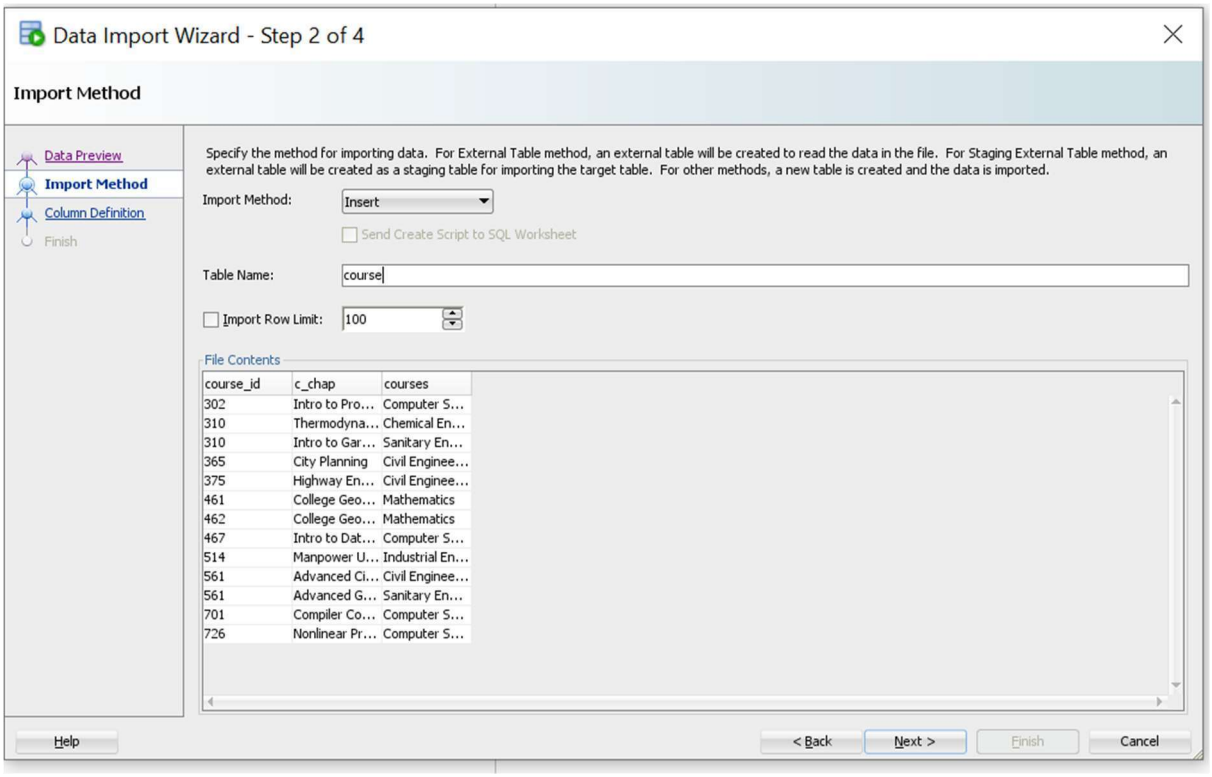
1. By following all the above steps you are ready to choose the file which you want to import. Now this is the first step to import data into the database.

2. You can preview the row limit by specifying 100 or 200 anything it depends upon the data you have.

3. You can simply have a look on file contents where the data in the table will be displayed.

4. You can check the data and click on next.

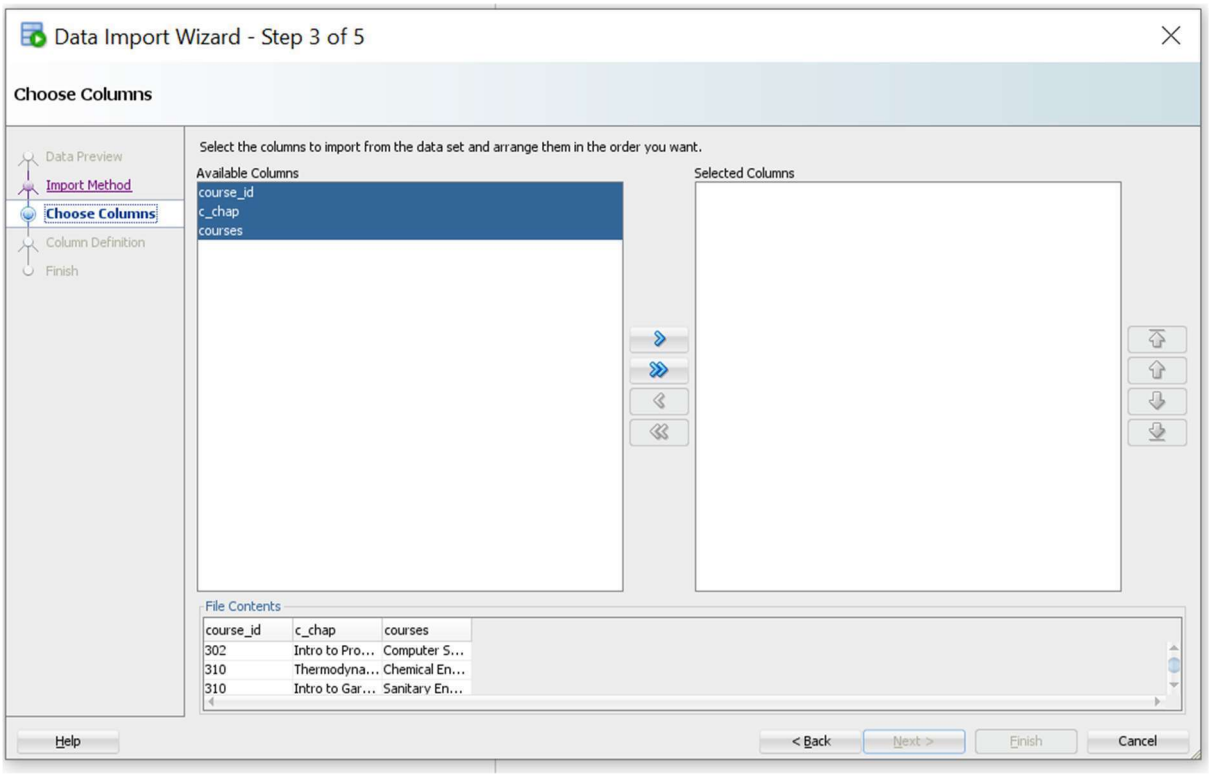
STEP 4:



1. In this case I have imported course data set. So I specify the table name with course as easy to work on it. But you can create a table name whatever you want.

2. You can specify the row limit to import then click on next.

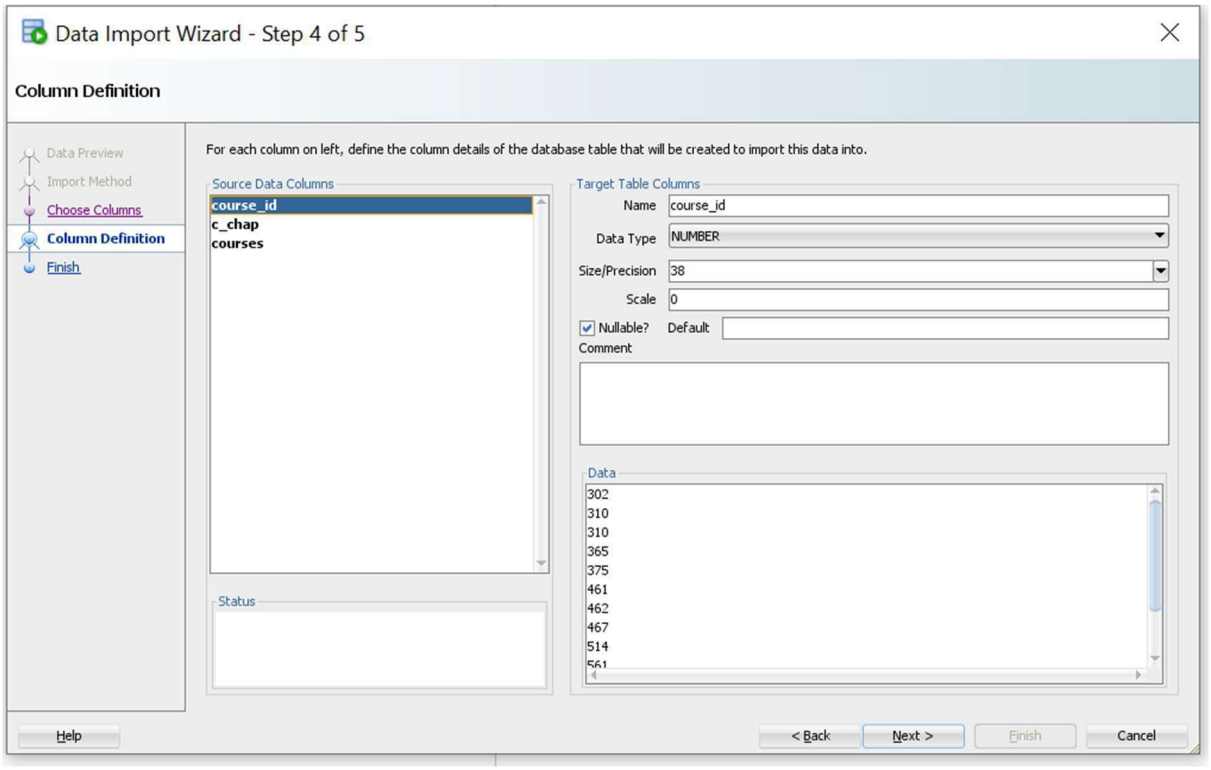
STEP 5:



1. You can select the column names that you want to import into the database by clicking the >> right arow.

2. After selecting all these columns then finally click next.

STEP 6:

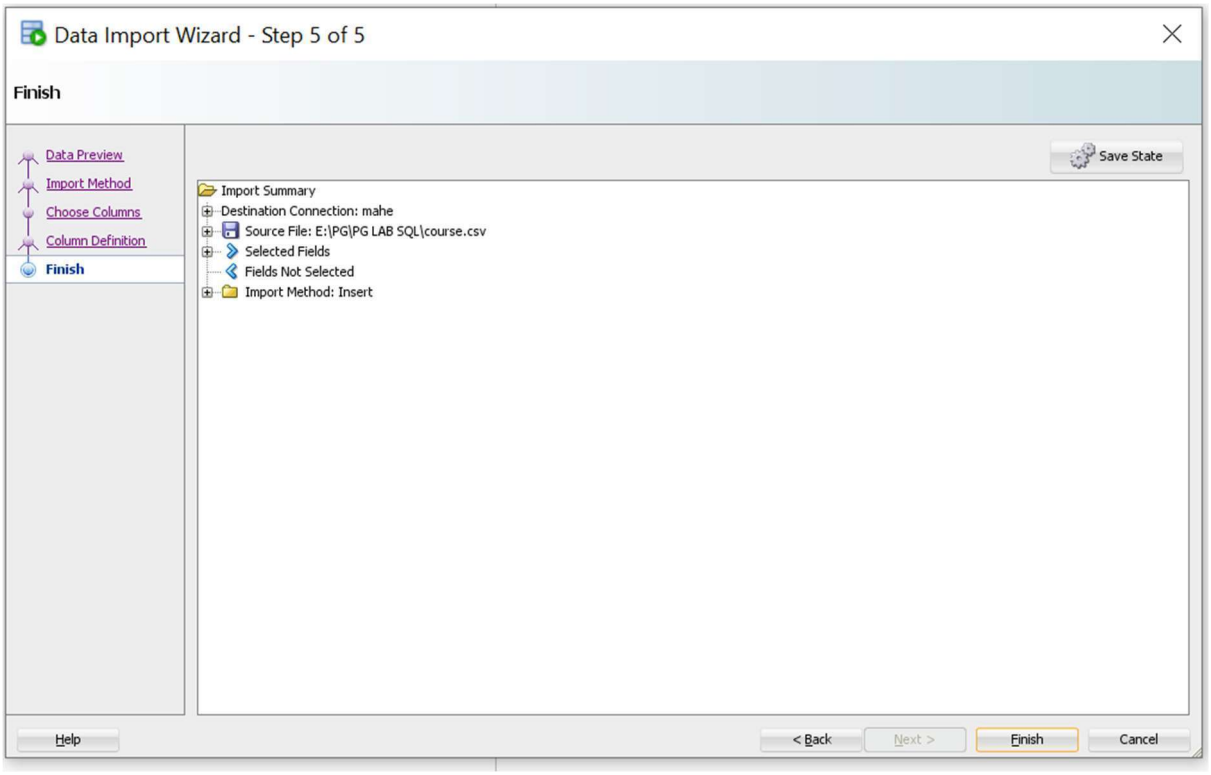


1. You need to check the column one by one. On the right side you have several options to choose the data type, name and nullable.

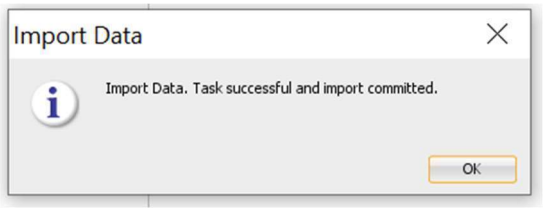
2. Data type will be in dropdown list so you can simply choose the data type that relates the value. By default, it automatically senses the type of data in many cases you need to choose for data format data it may take as date format or you can change in to varchar.

3. Do the same process for all the column name then click next.

STEP 7:



1. The above image shows the import summary in a quick view then by clicking finish button you will be prompted as import committed successfully.



2. All the above steps are followed to insert other tables similarly (dept, enroll, major, prof, section and student)

LAB-7. UNIVERSITY COURSE ENROLLMENT DATA ANALYTICS PART-2

Question1:

Print the names of professors who work in departments that have fewer than 50 PhD students.

SQL> select d.dname, d.num_phd, p.pname from prof p, dept d where d.num_phd<50 order by num_phd;

DNAME	NUM_PHD	PNAME
Sanitary Engineering	3	Jones, J.
Sanitary Engineering	3	Smith, S.
Sanitary Engineering	3	Brown, S.
Sanitary Engineering	3	Brian, C.
Sanitary Engineering	3	Randolph, B.
Sanitary Engineering	3	Bucket, T.
Sanitary Engineering	3	Robinson, T.
Sanitary Engineering	3	Clark, E.
Sanitary Engineering	3	Walter, A.
Sanitary Engineering	3	Edison, L.
Chemical Engineering	32	Walter, A.

DNAME	NUM_PHD	PNAME
Chemical Engineering	32	Clark, E.
Chemical Engineering	32	Robinson, T.
Chemical Engineering	32	Bucket, T.
Chemical Engineering	32	Edison, L.
Chemical Engineering	32	Brian, C.

Chemical Engineering	32	Brown, S.
Chemical Engineering	32	Smith, S.
Chemical Engineering	32	Jones, J.
Chemical Engineering	32	Randolph, B.
Industrial Engineering	41	Jones, J.
Industrial Engineering	41	Smith, S.

DNAME	NUM_PHD	PNAME
Industrial Engineering	41	Randolph, B.
Industrial Engineering	41	Walter, A.
Industrial Engineering	41	Clark, E.
Industrial Engineering	41	Robinson, T.
Industrial Engineering	41	Bucket, T.
Industrial Engineering	41	Edison, L.
Industrial Engineering	41	Brian, C.
Industrial Engineering	41	Brown, S.
Computer Sciences	47	Walter, A.
Computer Sciences	47	Jones, J.
Computer Sciences	47	Smith, S.

DNAME	NUM_PHD	PNAME
Computer Sciences	47	Brown, S.
Computer Sciences	47	Randolph, B.
Computer Sciences	47	Edison, L.
Computer Sciences	47	Bucket, T.
Computer Sciences	47	Robinson, T.
Computer Sciences	47	Clark, E.

Computer Sciences 47 Brian, C.

40 rows selected.

Question2:

Print the names of the students with the lowest GPA.

SQL> select sname, gpa from student where gpa=(select min(gpa) from student);

SNAME GPA

--- ---

Jetplane, Leaving O. 0

Question3:

For each computer Science class, print the class number, section number, and the average GPA of the students enrolled in the class section.

SQL> select e.cno, e.sect_no, avg(s.gpa) from enroll e, student s where dname = 'Computer Sciences' and s.sid = e.sid group by dname,cno,sect_no;

CNO	SECT_NO	AVG(S.GPA)
302	1	3
726	1	2.64117648
467	1	2.98000002
302	2	3.07499999
701	1	3.28333333

Question4:

Print the names and section numbers of all sections with more than six students enrolled in them.

SQL> select c.cname,c.cno,e.sect_no,count(e.sid) as student_count from course c left join enroll e on e.cno=c.cno group by c.cname,c.cno,e.sect_no having count(e.sid)>6;

CNAME

CNO SECT_NO STUDENT_COUNT

---------- ---------- -------------

Intro to Garbage

310 1 7

Manpower Utilization

514 1 9

Nonlinear Programming

726 1 17

CNAME

CNO SECT_NO STUDENT_COUNT

---------- ---------- -------------

Advanced City Planning

561 1 13

Intro to Data Structures

467 1 10

Compiler Construction

701 1 12

CNAME

CNO SECT_NO STUDENT_COUNT

---------- ---------- -------------

Thermodynamics

310 1 7

Highway Engineering

375 1 9

Intro to Programming

302 1 10

CNAME

CNO SECT_NO STUDENT_COUNT

---------- ---------- -------------

Intro to Programming

302 2 8

City Planning

365 1 8

College Geometry 1

461 1 9

CNAME

CNO SECT_NO STUDENT_COUNT

---------- ---------- -------------

College Geometry 2

462 1 9

Advanced Garbage Collection

561 1 13

14 rows selected.

Question5:

Print the name(s) and sid(s) of the student(s) enrolled in the most sections.

SQL> select sname, sid from student where sid in (select sid from enroll group by sid having count(*)>=all(select count(*) from enroll group by sid));

SNAME SID

-------------------------- ----------

Hamilton, S. 29

Question6:

Print the names of departments that have one or more majors who are under 18 years old.

SQL> select distinct m.dname from major m,student s where m.sid=s.sid and s.age<18;

DNAME

Mathematics

Industrial Engineering

Question7:

Print the names and majors of students who are taking one of college Geometry courses.

SQL> select m.sid, m.dname from major m inner join enroll e on e.sid = m.sid where e.cno in(461,462);

SID DNAME

---------- ---

4 Computer Sciences

14 Computer Sciences

17 Computer Sciences

18 Computer Sciences

19 Computer Sciences

26 Chemical Engineering

28 Chemical Engineering

35 Chemical Engineering

37 Civil Engineering

40 Civil Engineering

53 Civil Engineering

SID DNAME

---------- -------------------------

55 Civil Engineering

59 Civil Engineering

90 Mathematics

91 Mathematics

94 Mathematics

101 Mathematics

102 Mathematics

18 rows selected.

Question8:

For those departments that have no major taking a college Geometry course print the department name and the number of PhD students in the department.

SQL> select dept.dname, dept.num_phd from dept where not exists(select 1 from course where course.dname=dept.dname and course.cname like '% college geometry%');

DNAME NUM_PHD

------------------------- ----------

Industrial Engineering 41

Chemical Engineering 32

Mathematics 129

Computer Sciences 47

Sanitary Engineering 3

Civil Engineering 88

6 rows selected.

Question9:

Print the names of students who are taking both a Computer Sciences course and a Mathematics course.

SQL> select s.sname from student s inner join enroll e on e.sid = s.sid where e.dname ='Computer Sciences' and e.dname ='Mathematics';

no rows selected

Question10:

Print the age difference between the oldest and the youngest Computer Sciences major.

SQL> select max(s.age) - min(s.age) as "age dif" from student s inner join major m on m.sid = s.sid where m.dname='Computer Science';

age dif

38

Question11:

For each department that has one or more majors with a gpa under 1.0, print the name of the department and the average GPA of its majors.

SQL> select dname, cno, avg(grade) from enroll enroll group by dname, cno;

DNAME CNO AVG(GRADE)

--- --- ---

Mathematics 462 3.5

Civil Engineering	365	2.6875
Computer Sciences	302	3.01666667
Computer Sciences	701	3.20833333
Civil Engineering	561	2.54166667
Sanitary Engineering	561	2.6
Industrial Engineering	514	3.16666667
Computer Sciences	726	3.23529412
Chemical Engineering	310	3
Mathematics	461	3.11111111
Computer Sciences	467	3.1

DNAME	CNO	AVG(GRADE)
Civil Engineering	375	3.11111111

12 rows selected.

Question12:

Print the ids, names and GPAs of the students who are currently taking all the Civil Engineering courses.

SQL> select e.sid, s.sname, s.gpa from student s right outer join enroll e on s.sid = e.sid where e.dname = 'Civil Engineering' order by gpa;

SID	SNAME	GPA
81	Smith, Ike Z.	1.10000002
18	Gooch	1.39999998
47	Roger, Blotter N.	1.89999998
47	Roger, Blotter N.	1.89999998

9	Smith, Joyce A.	2
61	Kennedy, Ed	2.29999995
34	Kasten, Norman L.	2.5
60	Calcmity, J.	2.59999991
66	Altenhaus, Stuart	2.79999995
29	Hamilton, S.	2.79999995
29	Hamilton, S.	2.79999995

SID	SNAME	GPA
66	Altenhaus, Stuart	2.79999995
29	Hamilton, S.	2.79999995
36	Burroughs, Susan S.	3
54	Maximillian	3
76	Zorhoff, C.	3
70	Caucutt, B.	3
23	Bomber, C.	3.20000005
96	Birch, M.	3.5
85	Mayer, N.	3.5
96	Birch, M.	3.5
33	Chao, Tsechih	3.59999991

SID	SNAME	GPA
74	Andrus, J.	3.70000005
32	Liu, Huihusan	3.9000001
79	Evert, Chris	3.9000001
3	Zeene, Ben N.	3.9000001

64	Fred, Edwin B.	4
48	Natividad, A.	4
73	Quarnty, G.	4

29 rows selected.

REPORT

I Have completed my SQL lab with help of notes provided during the regular online classes as well as reference book shared in class that is “book **BeginningSQLqueries** by Clare Churcher”.

I got required the output (correct format) after rectifying silly error in almost each queries. I Refer website such as https://stackoverflow.com/, https://www.geeksforgeeks.org/ https://www.w3schools.com/ , to get know different ways of writing queies.

Department of Data Science, Bishop Heber College Tiruchirappalli
NoSQL Database Management Lab

Lab8. Retail Sales Analytics Part-IV

Objectives:

In this lab, you will continue the exploration of Retail Sales dataset using multiple tables. You will learn and apply nested queries or subqueries.

Tasks To Be Done

Question1: Develop 4 nested queries that satisfy these requirements:

- At least one subquery per query
- At least 2 of the queries must use an aggregate function
- At least 2 of the queries must use a join (either inner or outer)

Question2: Create 2 Data Visualizations

- Create a chart in Google Data Studio that visualizes the output of two most interesting nested queries
- Add the charts to your existing Data Studio dashboard.

Question:1

i) SQL> Select store_id, fuel_price from feature_data where fuel_price in (select MAX(fuel_price) from feature_data);

ii) SQL> Select sales_id, MIN(weekly_sales) from sales where sales_id<10 group by sales_id;

iii) SQL> Select sales.isholiday, feature_date from sales inner join feature_data on sales.isholiday = feature_data.isholiday where sales_id<5 and feature_id<5;

iv) SQL> Select sales.dept from sales full outer join feature_data on sales.sales_id = feature_data.feature_id where sales_id<7 and feature_id<7;

RETAIL SALES ANALYTICS PART-IV

Question 1: Develop 4 nested queries that satisfy these requirements:

- **At least one subquery per query**
- **At least 2 of the queries must use an aggregate function**
- **At least 2 of the queries must use a join (either inner of outer)**

i) SQL> select store_id,fuel_price from feature_data where fuel_price in (select MAX(fuel_price) from feature_data);

```
 STORE_ID FUEL_PRICE
---------- ----------
        12      4.468
        10      4.468
        28      4.468
        33      4.468
        42      4.468
        38      4.468

6 rows selected.
```

ii) SQL> select sales_id,MIN(weekly_sales)from sales where sales_id<10 group by sales_id;

```
 SALES_ID MIN(WEEKLY_SALES)
---------- -----------------
         1          24924.5
         2         46039.49
         3         41595.55
         4         19403.54
```

5	21827.9
6	21043.39
7	22136.64
8	26229.21
9	57258.43

9 rows selected.

iii) SQL> select sales.isholiday,feature_date from sales inner join feature_data on sales.isholiday = feature_data.isholiday where sales_id<5 and feature_id<5;

ISHOLIDAY	FEATURE_
FALSE	05-02-10
FALSE	05-02-10
FALSE	05-02-10
TRUE	12-02-10
FALSE	19-02-10
FALSE	19-02-10
FALSE	19-02-10
FALSE	26-02-10
FALSE	26-02-10
FALSE	26-02-10

10 rows selected.

iv) SQL> select sales.dept from sales full outer join feature_data on sales.sales_id = feature_data.feature_id where sales_id<7 and feature_id<7;

```
    DEPT
----------
         1
         1
         1
         1
         1
         1

6 rows selected.
```