

/*

QUESTION- 5

By submitting this assignment for grading, I state or affirm that the work submitted is my own work and I did not receive assistance nor provide assistance from another student (or external agent). Febin Varghese*/

```
with sq as (select c.course_number,pr.prereq,c.course_name
from eb.course c,eb.prereq pr
where c.course_number=pr.course_number(+))
select course_number "Course",
case when sys_Connect_by_path(prereq,'/')='/'
then '-'
else '/'||ltrim(sys_Connect_by_path(prereq,'/'),'/') end
"Prerequisite(s)",
course_name "Course Name",level "Level"
from sq
start with prereq is null
connect by prior course_number=prereq
order siblings by "Course Name";
```

/*

With clause is used to join(left outer) course with prereq so as to get courses with prereq and courses without any prereq(ie prereq would be null).

Hierarchical query is build with Prereq is null (specified as the root node from which hierarchy chain needs to be built).

Child records are built by specifying the condition prior course number equal to prereq.

Sys_connect_by_path function is used with delimiter '/' so as to display the hierarchy path with the formatting necessary.

Pseudo column Level is used to display the hierarchy level for each courses displayed.

Case when is used to determine courses without any prerequisites.

Order siblings by clause is used to sort the sibling courses following the parent in alphabetic order.

*/

/*ASSUMPTION- Displaying all courses. i.e courses from course table that are not in prereq table would be displayed in result but with level 1 and no paths.

If such courses need to be ignored need to put where filter(given below) before start with clause to remove such courses.

```
where course_number not in
(select course_number from (select course_number from eb.course
where course_number not in (select course_number from eb.prereq) )
where course_number not in (select prereq from eb.prereq))
```

*/