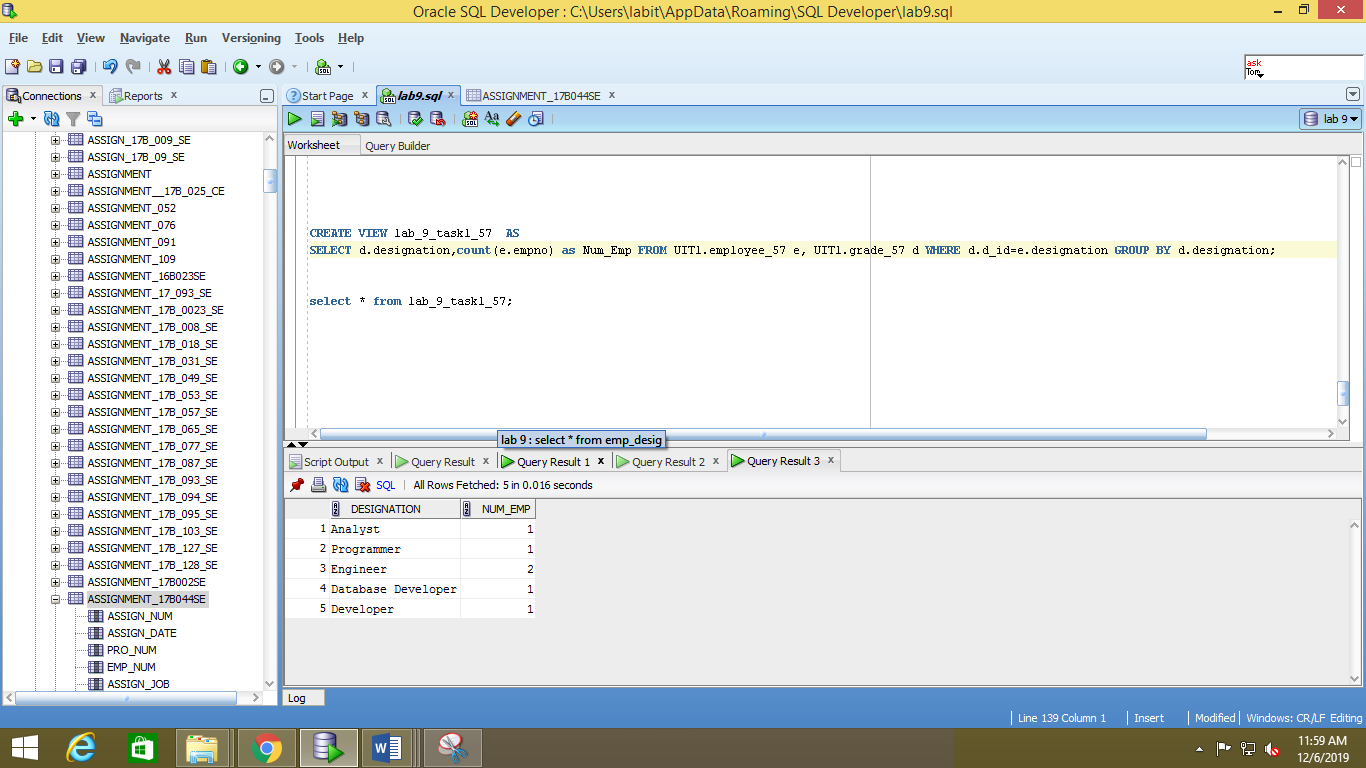
**TASK1**

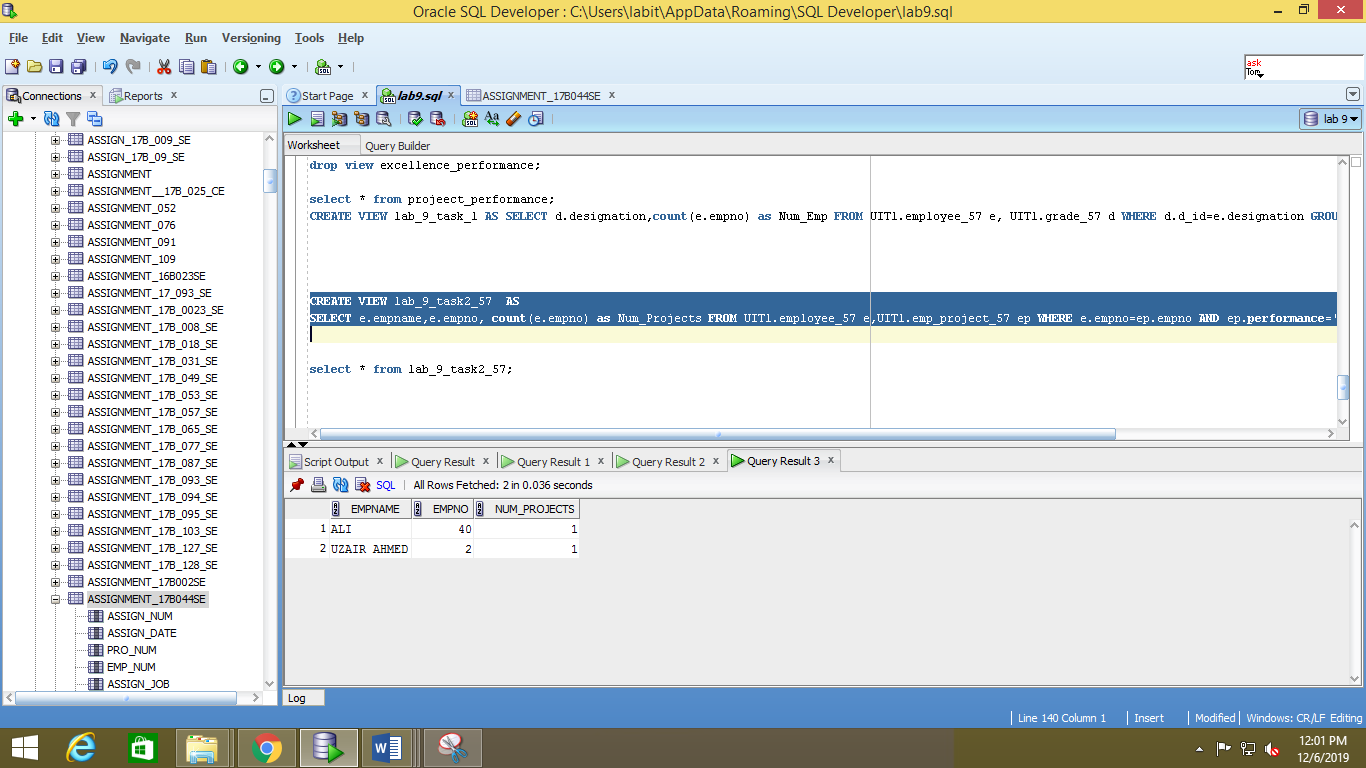
CREATE VIEW lab\_9\_task1\_33 AS SELECT d.designation,count(e.empno) as Num\_Emp FROM UIT1.employee\_33 e, UIT1.grade\_33 d WHERE d.d\_id=e.designation GROUP BY d.designation;



**TASK 2**

CREATE VIEW lab\_9\_task2\_33 AS

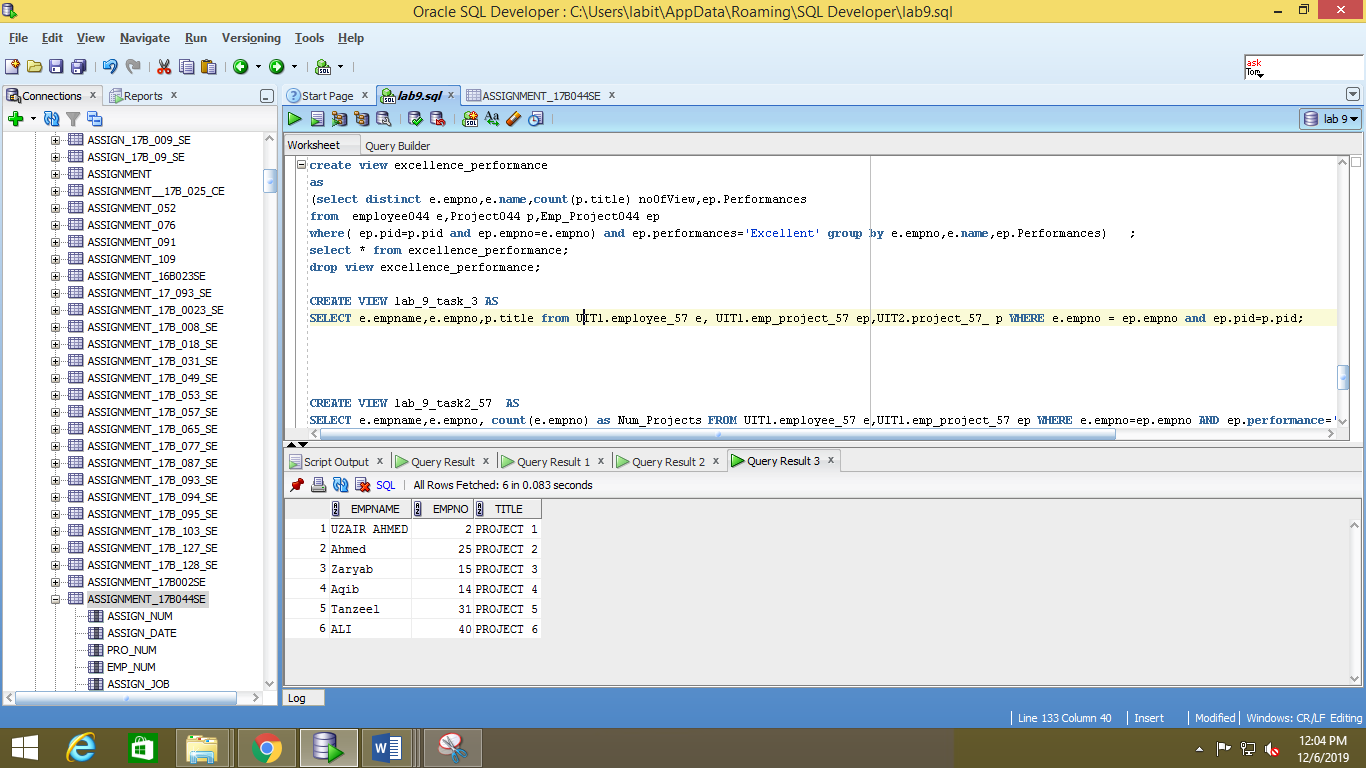
SELECT e.empname,e.empno, count(e.empno) as Num\_Projects FROM UIT1.employee\_33 e,UIT1.emp\_project\_33 ep WHERE e.empno=ep.empno AND ep.performance='Excellent' GROUP BY e.empname,e.empno;



**TASK 3**

CREATE VIEW lab\_9\_task\_3 AS

SELECT e.empname,e.empno,p.title from UIT1.employee\_33 e, UIT1.emp\_project\_33 ep,UIT2.project\_33\_ p WHERE e.empno = ep.empno and ep.pid=p.pid;



**TASK 4**

What are the different uses of views? Differentiate between simple and complex views.

A **View** in SQL as a logical subset of data from one or more tables. Views are used to restrict data access. A View contains no data of its own but its like window through which data from tables can be viewed or changed. The table on which a View is based are called BASE Tables.

There are 2 types of Views in SQL:

1. Simple View
2. Complex View.

|  |  |
| --- | --- |
| **SIMPLE VIEW** | **COMPLEX VIEW** |
| 1. Contains only one single base table or is created from only one table. | 1. Contains more than one base tables or is created from more than one tables. |
| 1. We cannot use group functions like MAX(), COUNT(), etc. | 1. We can use group functions. |
| 1. Does not contain groups of data. | 1. It can contain groups of data. |
| 1. DML operations could be performed through a simple view. | 1. DML operations could not always be performed through a complex view. |
| 1. INSERT, DELETE and UPDATE are directly possible on a simple view. | 1. We cannot apply INSERT, DELETE and UPDATE on complex view directly. |