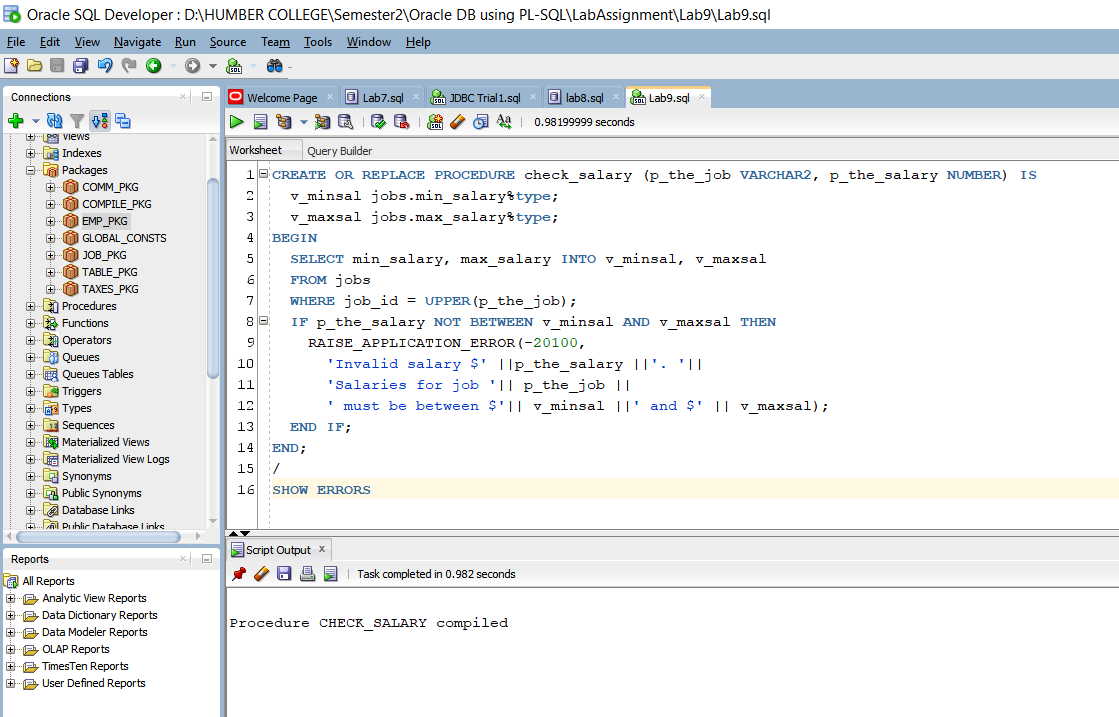
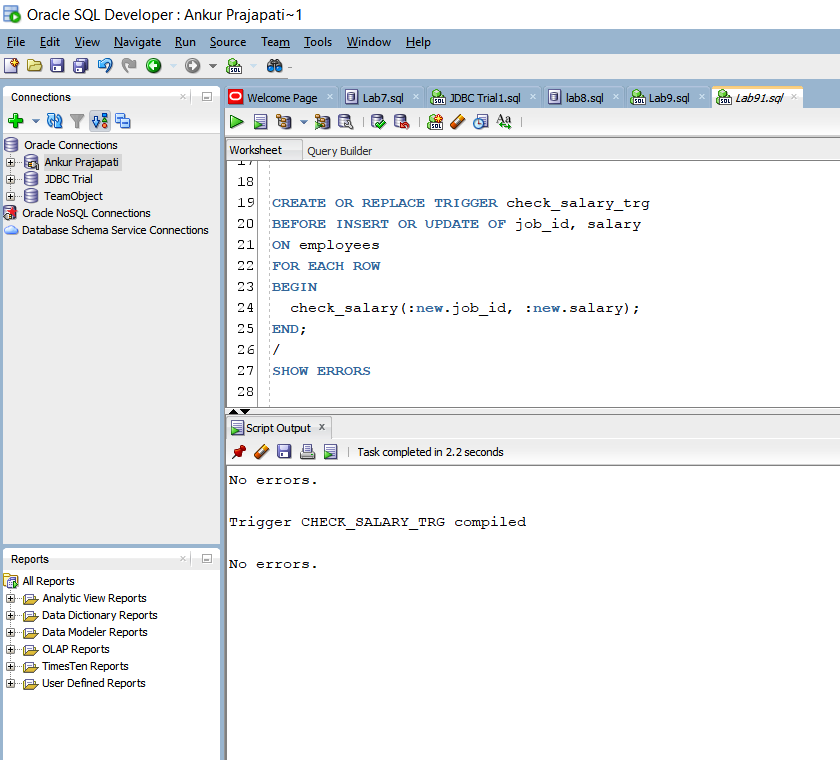
Practice – 1 - 1:

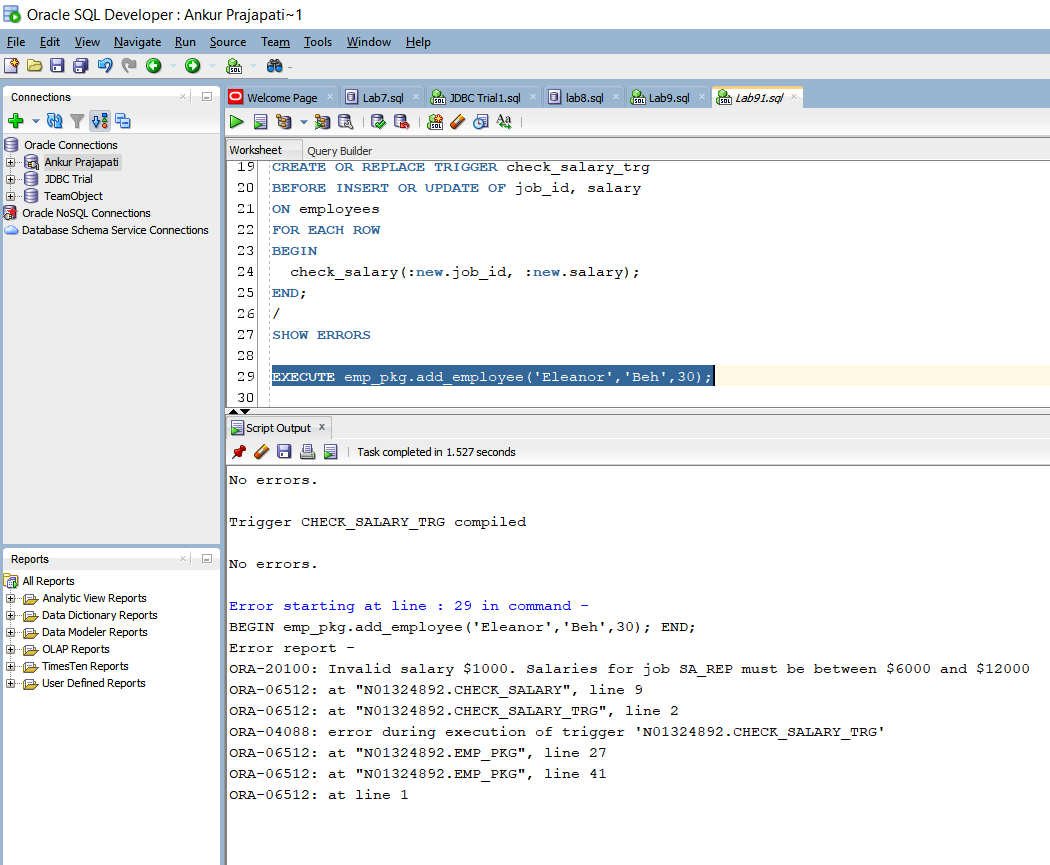
* Creation of procedure check\_salary with parameters p\_the\_job with type varchar2 and p\_the\_salary with type number and we are declaring two variables v\_minsal and v\_maxsal with type of jobs.min\_salary. After that we are selecting minimum and maximum salary from jobs where job\_id matches with the parameters p\_the\_job. If the salary is not between min and max salary it will raise an error notifying that it is invalid salary.
* Creation of trigger check\_salary\_trg which will be triggered when an insert or update happens for job\_id and salary for employees table.
* Note that we are calling this for each and every row for job\_id. Trigger will pass the new job\_id and salary to the procedure parameters.

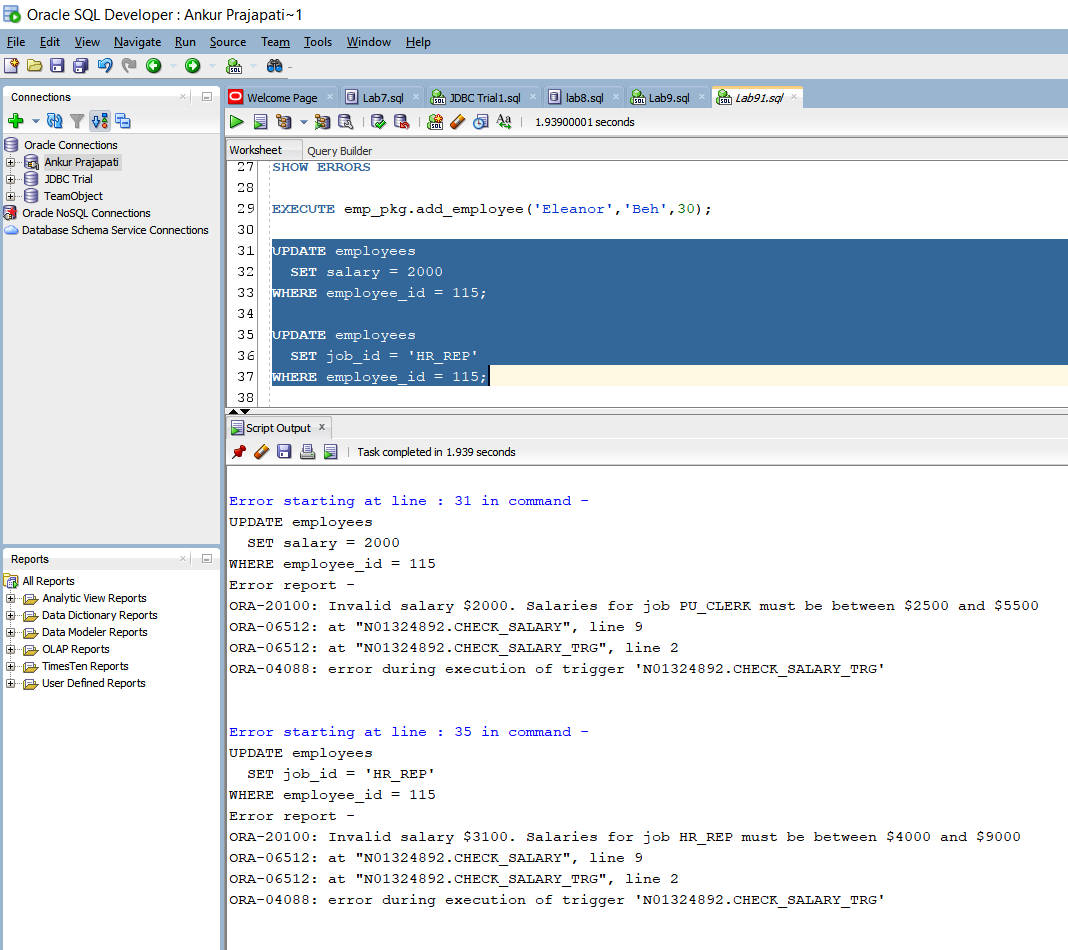


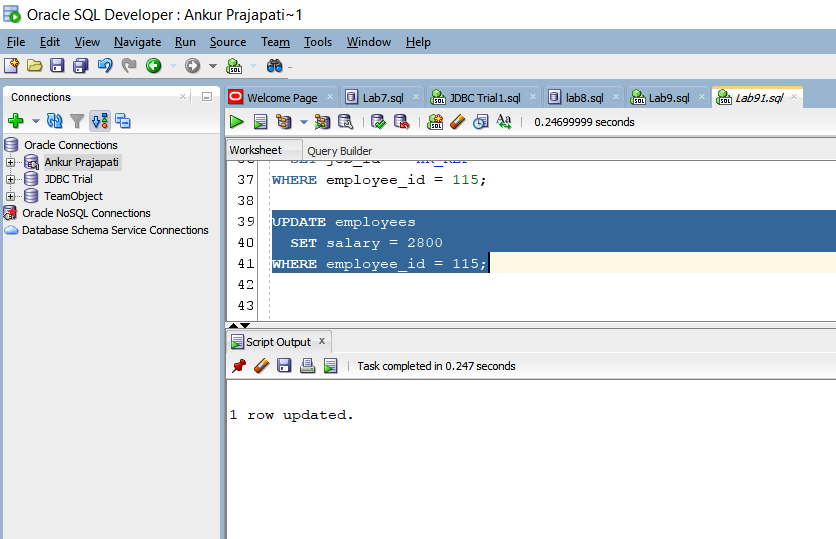


Practice – 1 - 2:

* We are executing emp\_pkg.add\_employee with parameters ‘Eleanor’, ‘Beh’, 30. Note that this salary ($1000) is invalid because it should be between min salary $6000 and max salary $12000. Due to check\_salary\_trg trigger we are actually getting this error. Here, procedure invokes an overloaded version of itself that uses the default salary of $1000 and default job\_id of ‘SA\_REP’
* Here, first update statement fails to set the salary to $2000 where employee\_id = 115. Because min salary of job\_id ‘PU\_CLERK’ is $2500. That’s why an application error is invoke during updating data. In Second update statement same happens, here min salary of HR\_REP is $4000 and current employee’s salary is less than that salary. That’s why it fails.
* Where in third update statement it gets updated because the new value of salary, we are assigning is greater than the min salary of employee whose id is 115.

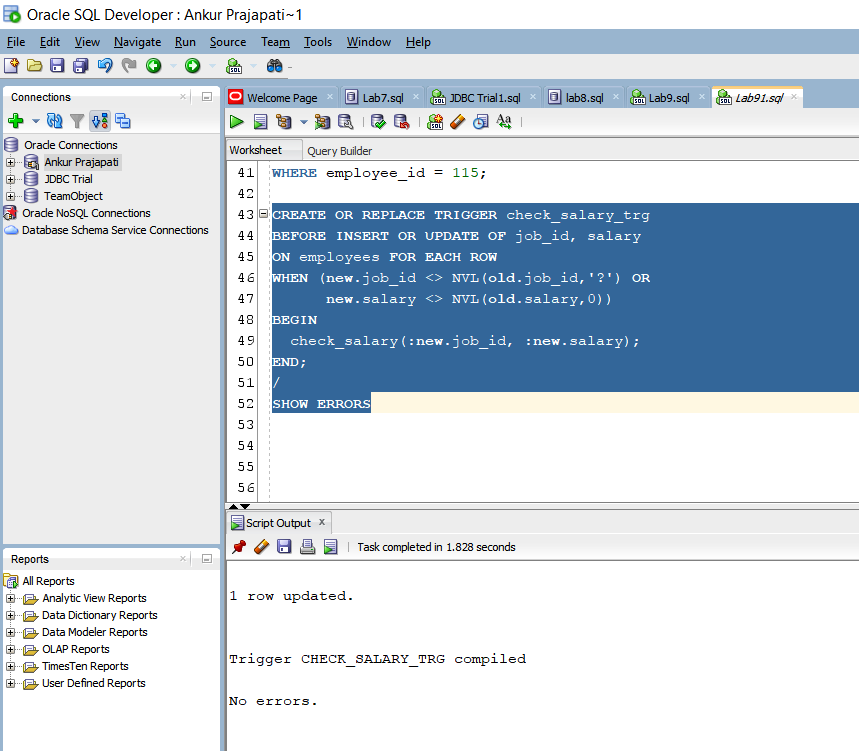


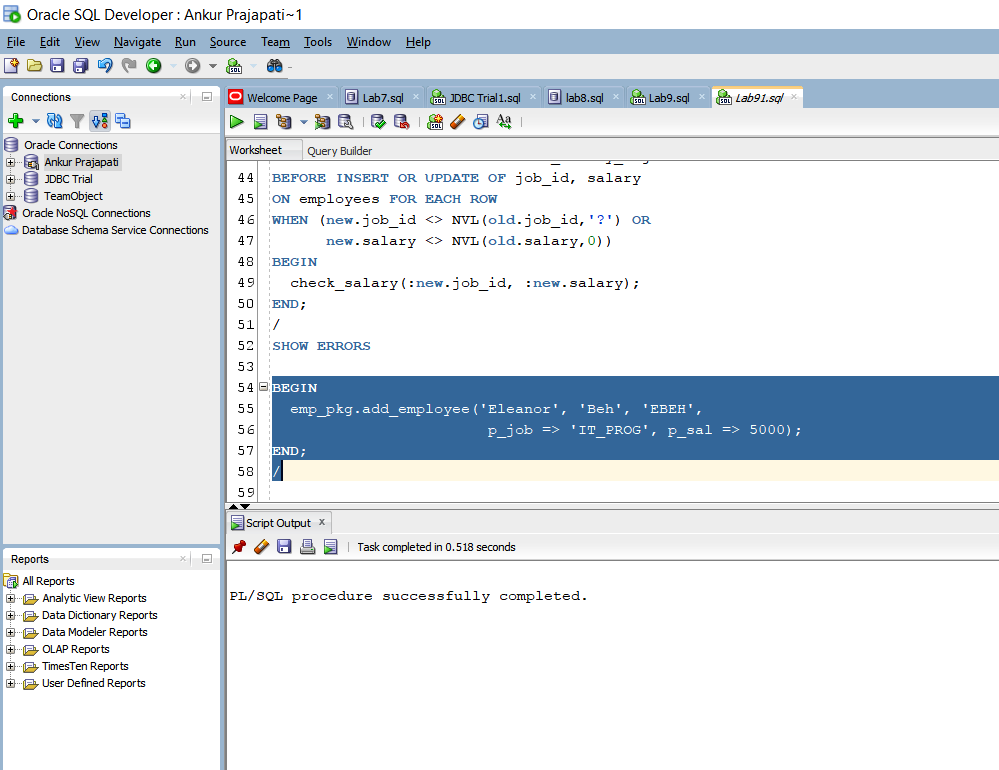


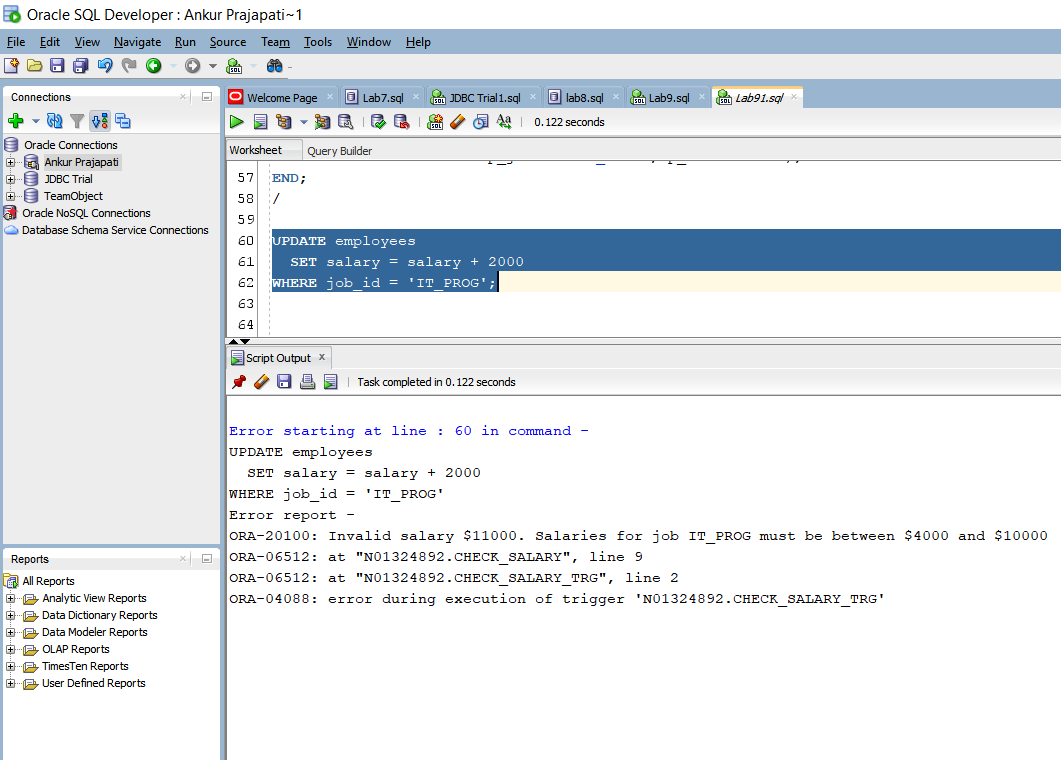


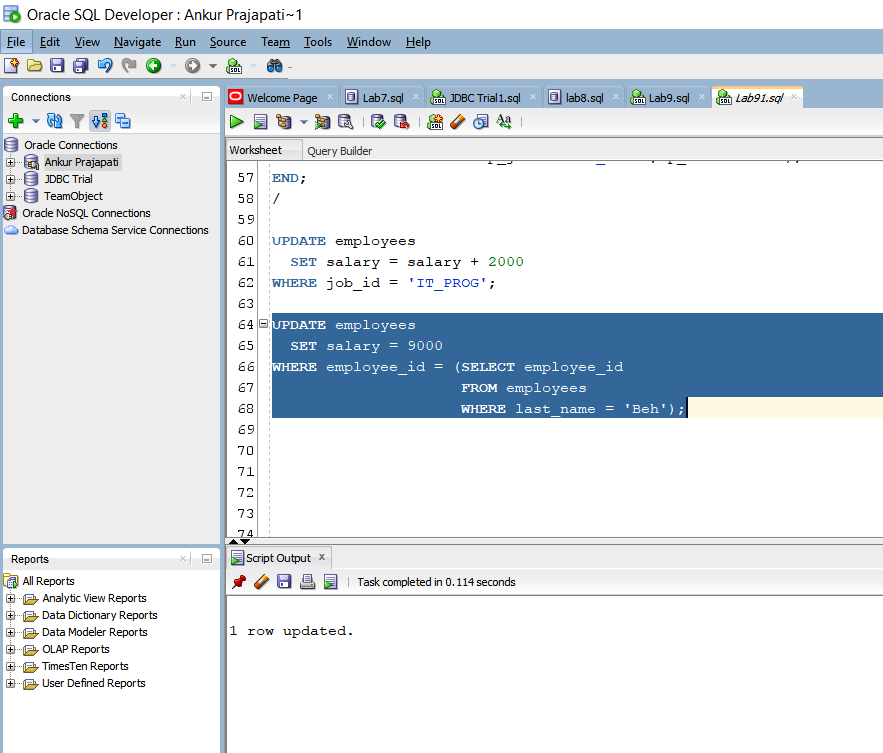
Practice – 1 – 3:

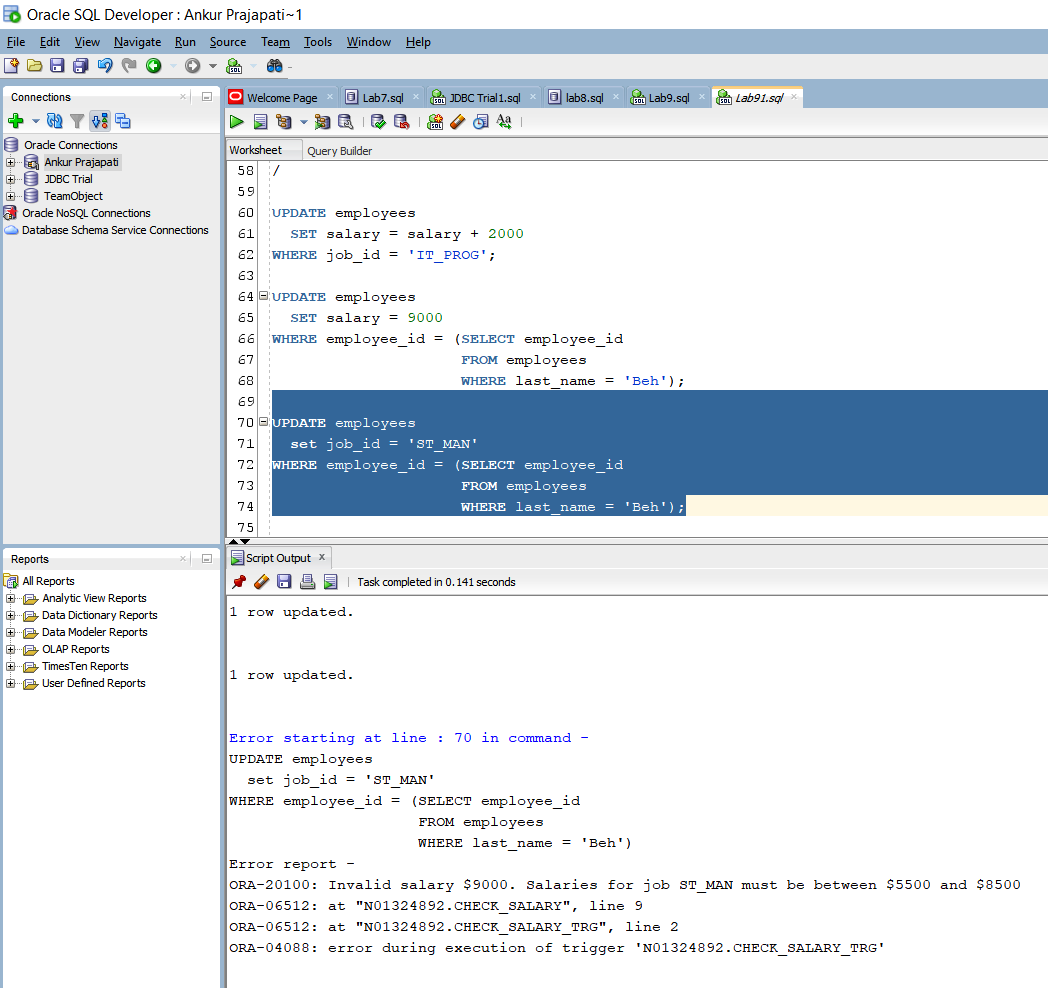
* We have created trigger check\_salary\_trg but here we are updating it by adding new business rule to that trigger. Basically, we are checking that whether job\_id or salary values have changed or not using WHEN clause.
* After that we are executing emp\_pkg.add\_employee to test the trigger created. On executing it gets executed successfully.
* Here, we are updating salary of employees by $ 2000 where job\_id is ‘IT\_PROG’. But we are not able to do that because of the trigger we created. Suppose we do that then it exceeds maximum salary of IT\_PROG. It raised an application error notifying that salaries for IT\_PROG must be between 4000 to 10000.
* Here we are updating salary for employee Beh by selecting employee\_id of Beh. We are setting salary of employee Beh to $ 9000. And it gets executed successfully. Because 9000 is between min salary and max salary.
* Here we are changing job of Eleanor Beh to ST\_MAN using update statement with another subquery but it dosen’t get updated because that employee’s salary is between 5500 and 8500. Where Beh’s salary is more than maximum salary. That’s why we are not able to change job.





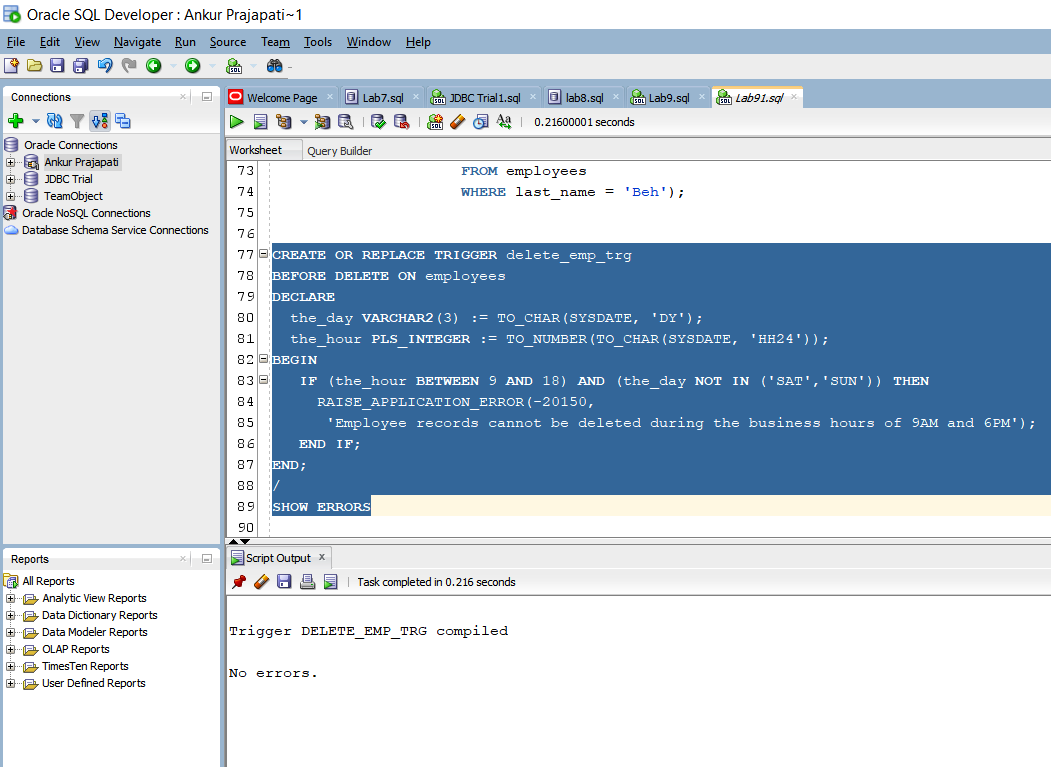


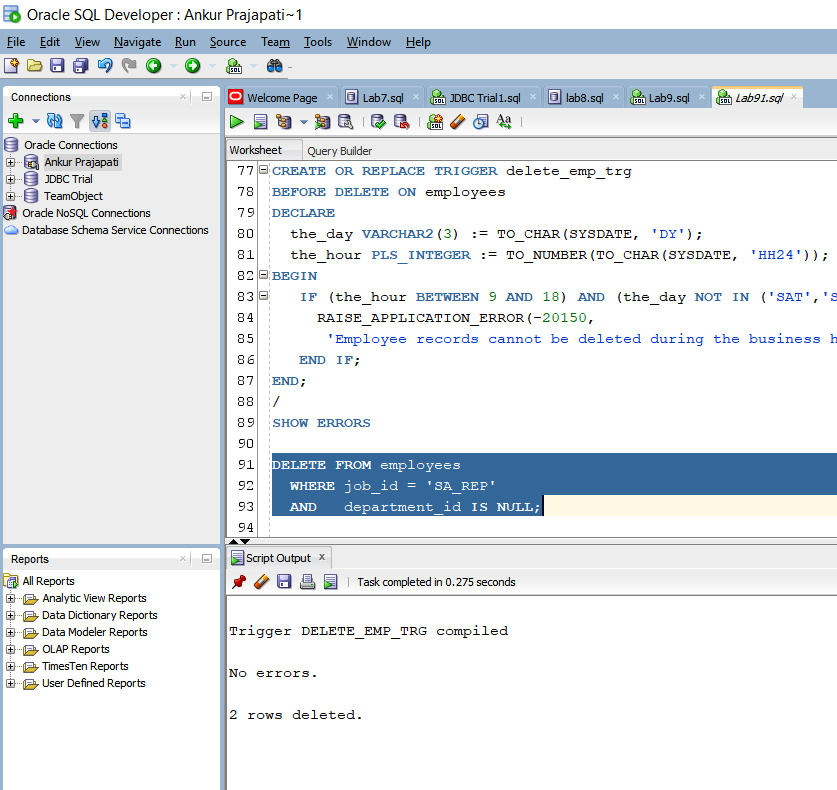




Practice - 1 – 4:

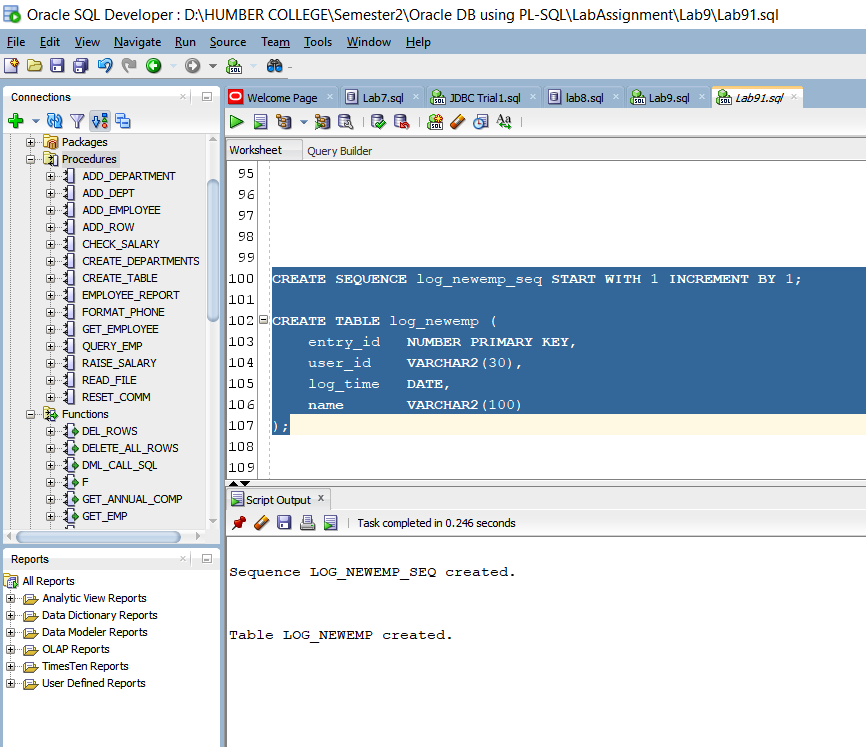
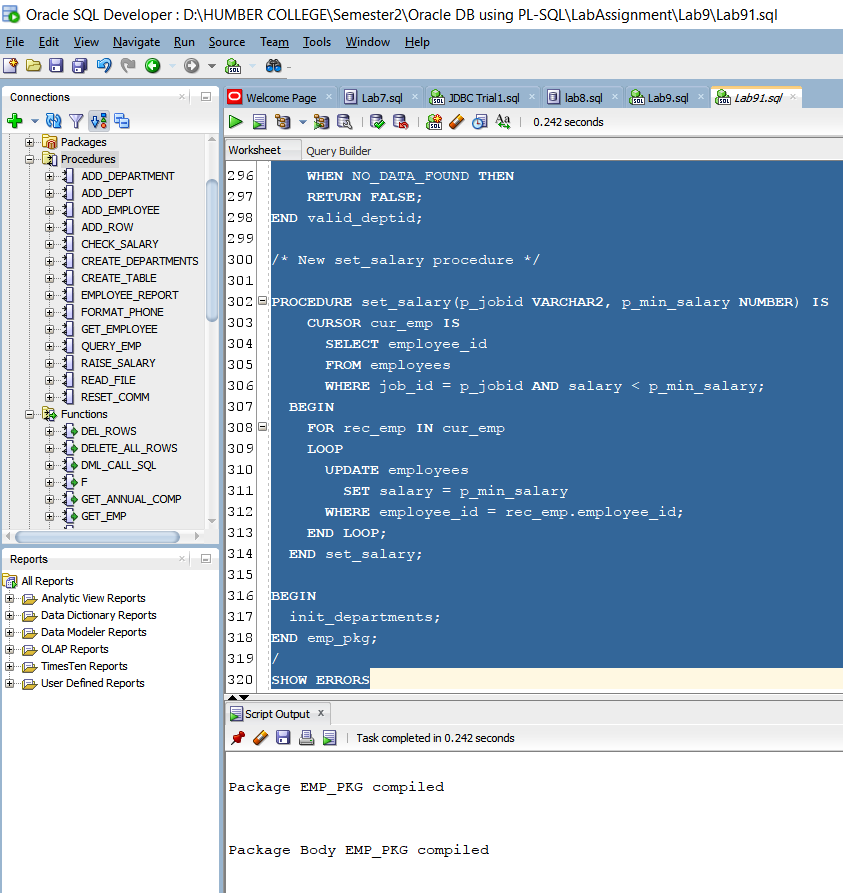
* We are writing a trigger called delete\_emp\_trg on the employees table to prevent an employee from being deleted during weekday business hours in between 9:00 AM to 6:00 PM. Trigger gets complied successfully.
* After that we are trying to delete from employees table. Note that it gets deleted because current user is trying to delete from employees table after 6:00 PM. That’s why it displays 2 rows deleted.

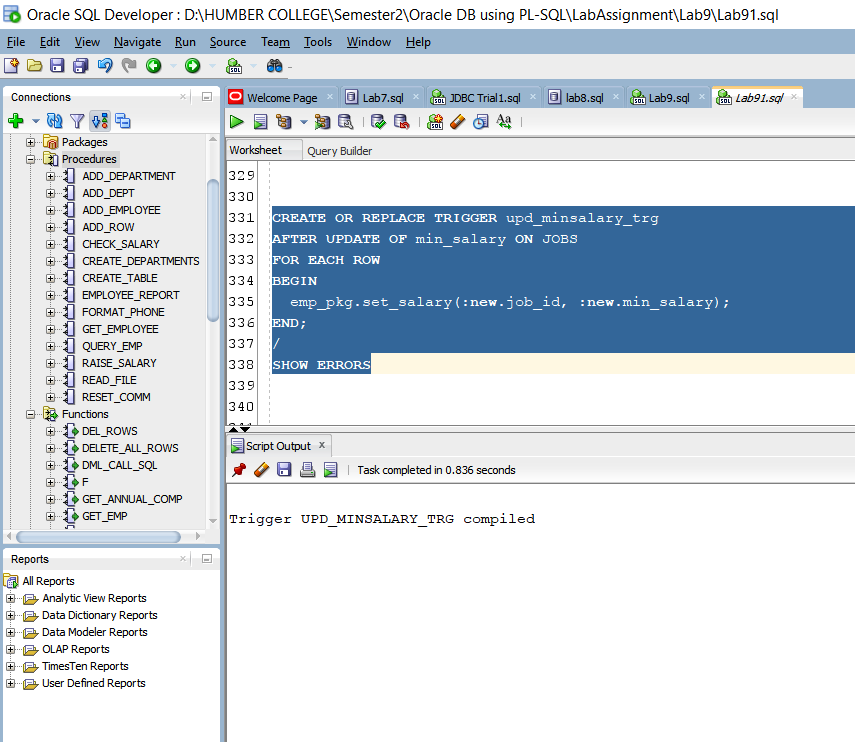


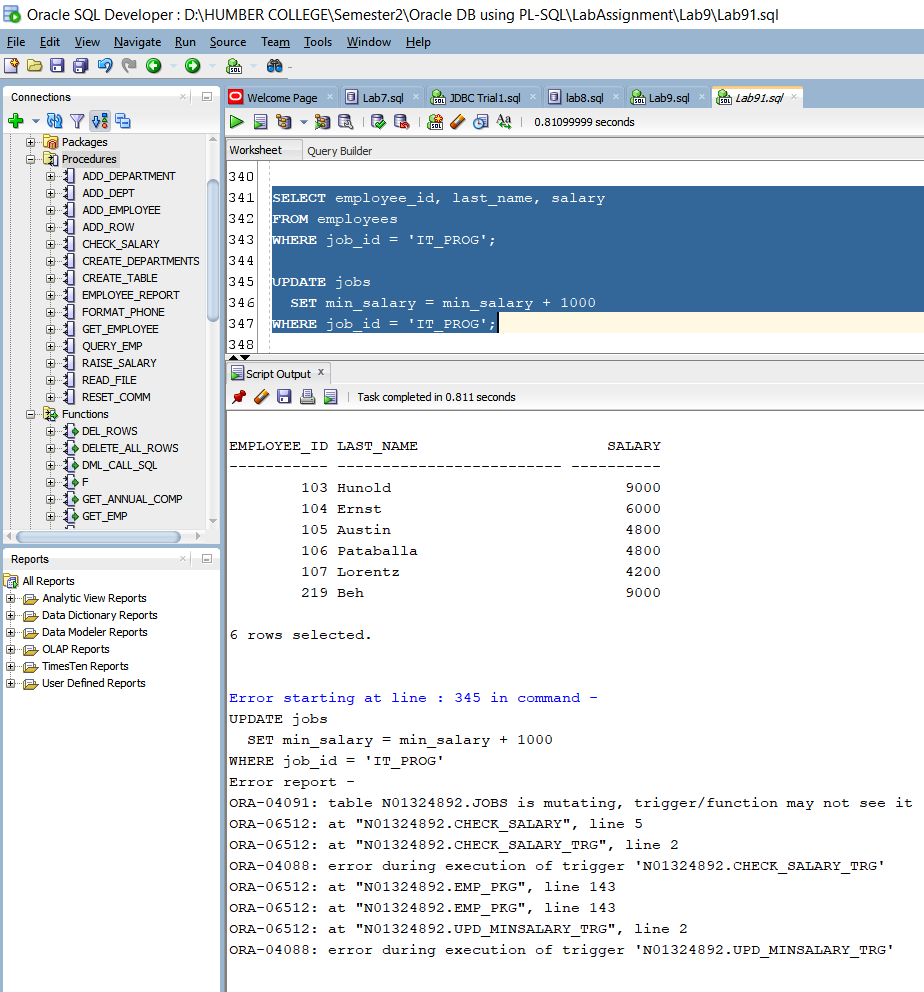


Practice – 2 – 1:

* We will implement a business rule to ensure data integrity of employee’s salaries with respect to the valid salary range for their jobs. For that employees receive an automatic increase in salary if the minimum salary for a job is increased to a value larger than their salaries. To implement this, we have to update emp\_pkg. On updating data, it will call the check\_salary trigger to read the salaries from jobs table. We have added procedure called SET\_SALARY that updates the employee’s salaries.
* Note that we weren’t able to run that package emp\_pkg because we need to create table called log\_newemp\_seq and to create that table we have created sequence that starts with 1 and increments by 1. After creating this table when we run this emp\_pkg gets complied successfully.
* After that we are creating a trigger named upd\_minsalary\_trg on the JOBS table that will invoke emp\_pkg.set\_salary procedure when minimum salary in the JOBS table is updated for a specified job ID.
* In next part we are writing a query to display employeeID, last name, job ID, current salary and minimum salary for employees whose job\_id is ‘IT-PROG’. After that we are updating minimum salary by $ 1000. Note that we were successful to fetch the 6 rows using select query. But we cannot update salary because upd\_minsalary\_trg calls setSalary, which invokes check\_salary\_trg which calls check\_salary which selects the minimum salary from jobs which puts implicit lock and so mutating error is thrown.

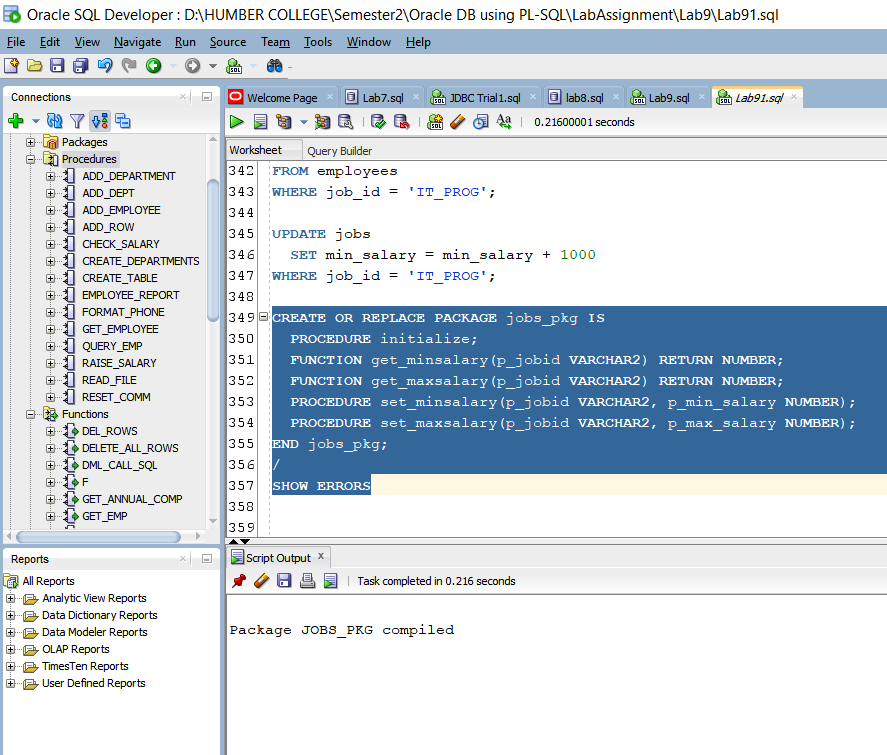


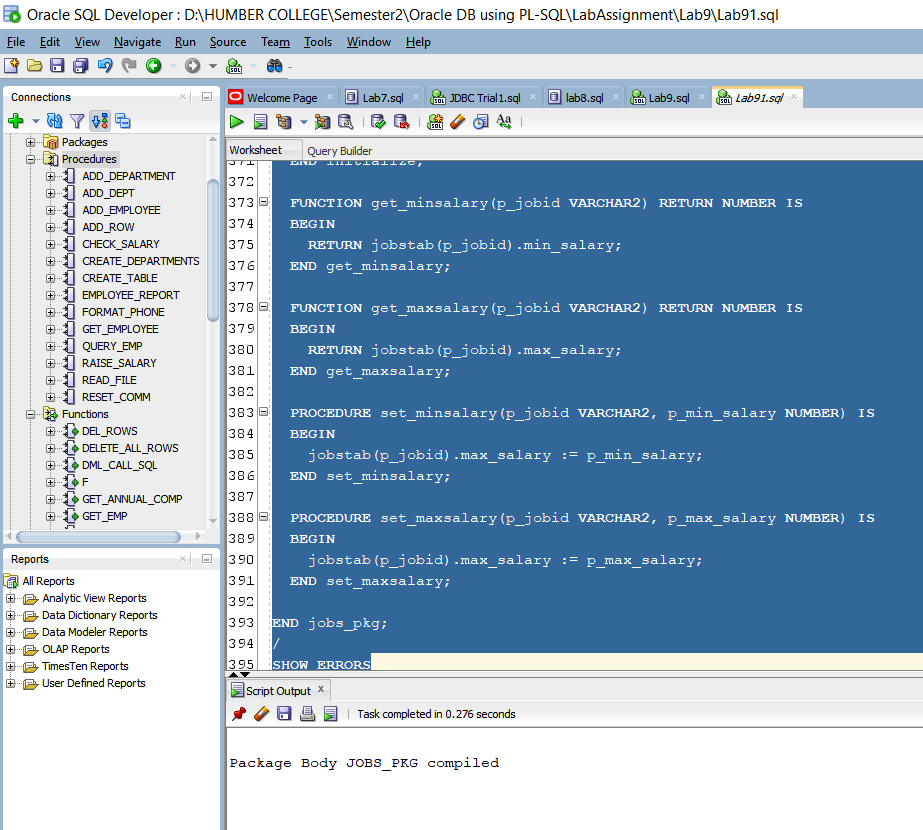


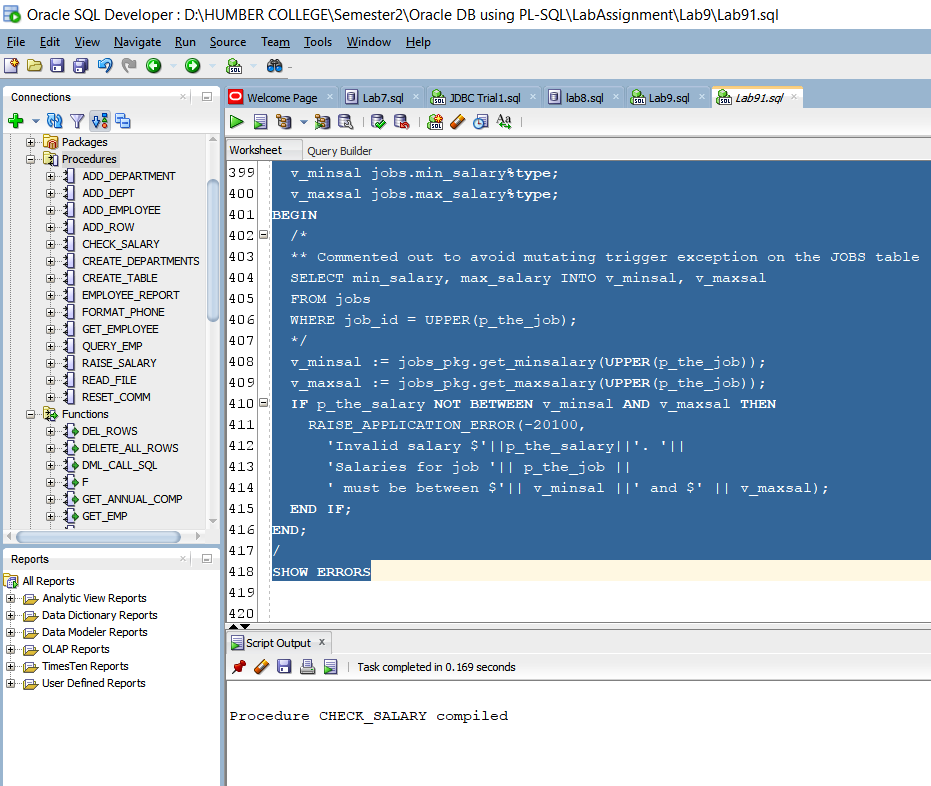


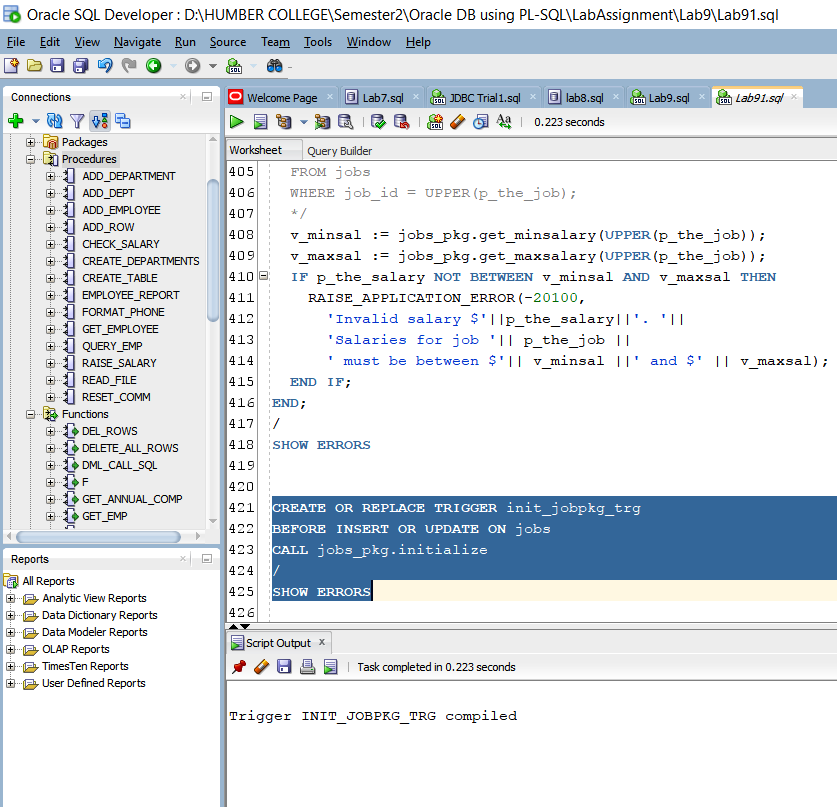
Practice – 2 – 2:

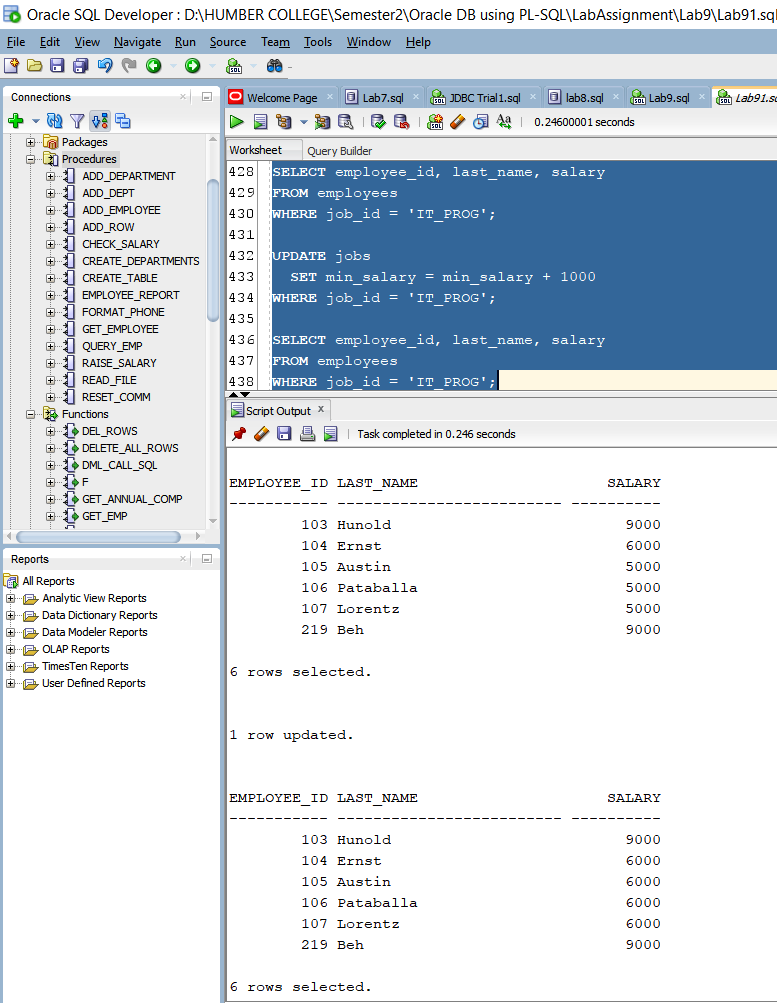
* We are creating jobs\_pkg with procedure initialize and functions named get\_minsalary and get\_maxsalary with two other procedures set\_minsalary and set\_maxsalary. It gets complied successfully. Note that we are using getter and setter method to get the min and max salary and after that to set the min and max salary.
* Declaring a private PL/SQL index by table called jobs\_tab\_type that is indexed by a string type based on the jobs.job\_id%type.the get\_minsalary function uses a p\_jobid and returns min\_salary, max\_salary.
* After that we are modifying the code by replacing the query on the jobs table with statements to set the local minsal and maxsal varribles with values from the jobs\_pkg data by calling appropriate get\_\*salary functions.
* Now we will implement BEFORE INSERT OR UPDATE statement in trigger INIT\_JOBPKG\_TRG that uses the CALL syntax to invoke the JOBS\_PKG.INITIALIZE procedure to ensure that the package state is current before the DML operations are performed.
* In next query employees with name Austin, Pataballa and Lorentz have had all their salaries updated. No exceptions occurred during this process and we have implemented a solution for the mutating table trigger exceptions.











Practice - 2- 3:

* Now we will check whether this trigger works. To test this we will call emp\_pkg.add\_employee with parameters (Steave, Morse, Smorse, p\_sal => 6500). It gets executed successfully.
* Now we are creating BEFORE INSERT OR UPDATE statement trigger called employee\_initjobs\_trg on the employees table that calls that procedure. We have also implemented CALL syntax in the trigger body.
* Now we are again adding this Steave Morse name and after calling select query we can fetch this but we can not add that name because it is already there.

