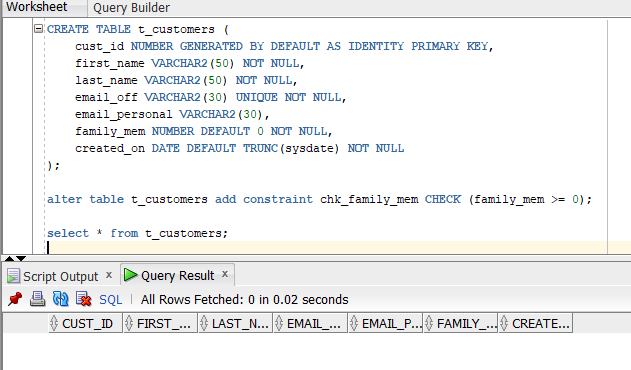
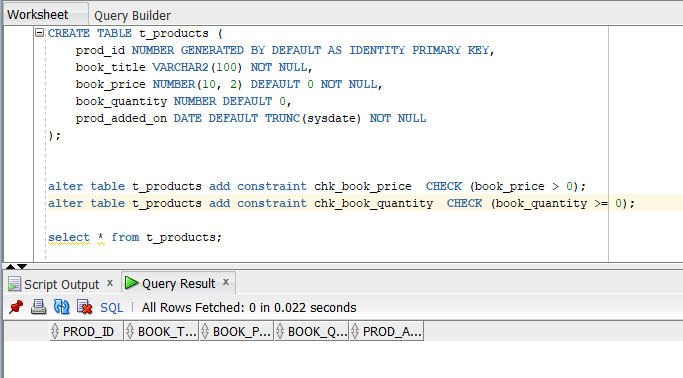
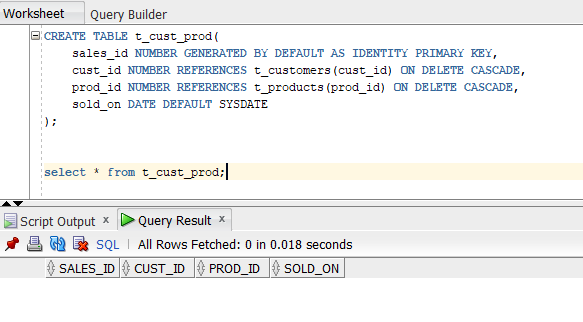
1. t\_customers table creation



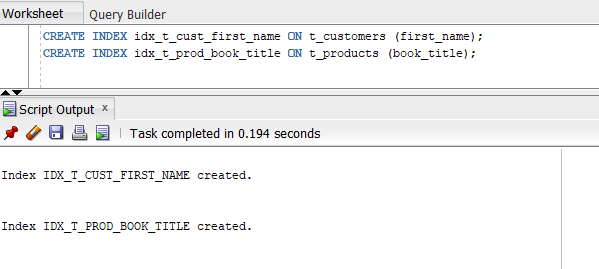
1. t\_products table creation



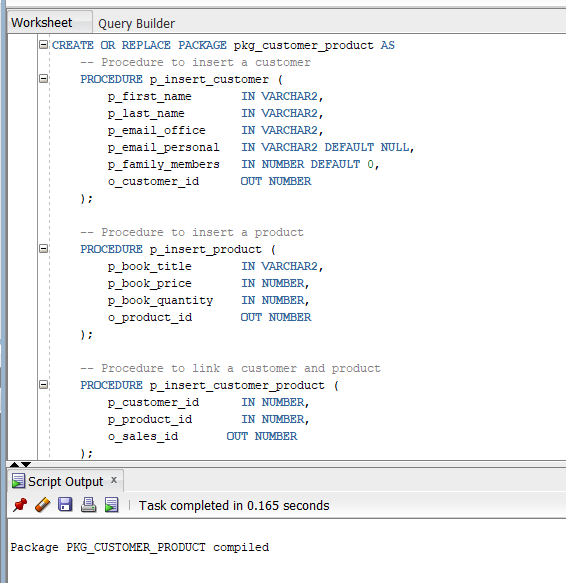
1. t\_cust\_prod table creation with cascade constraint( to maintain the data integrity and avoid integrity constraint error)

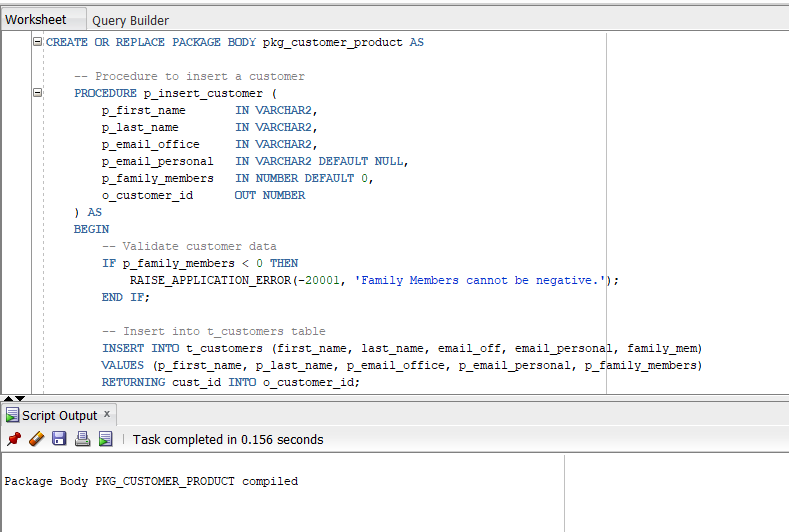


1. Index creation on t\_customers and t\_products tables

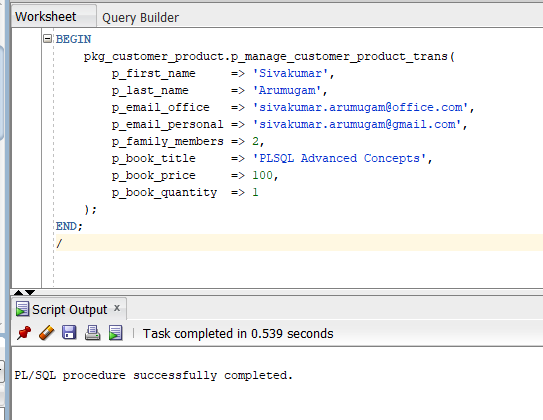


1. Creating package to demonstrate insert into 3 tables using separate procedures.
2. Also create a transactional procedure to demonstrate the transaction management

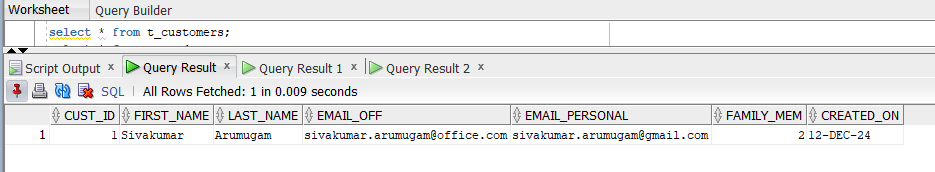


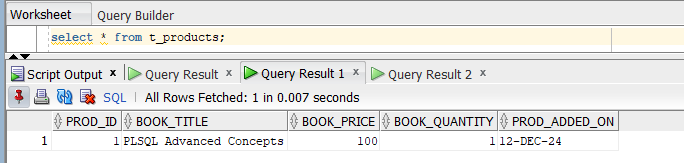


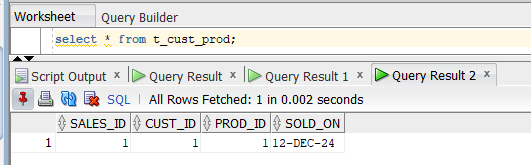
1. Inserting a transaction by creating a customer, product and a sales transaction by calling p\_manage\_customer\_product\_trans procedure.



Results successful

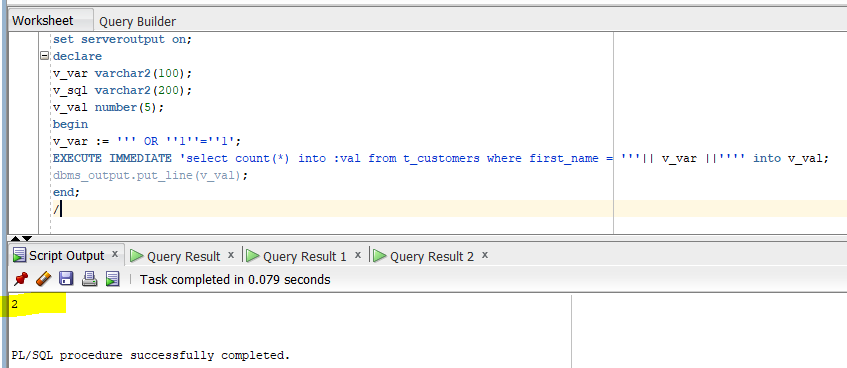






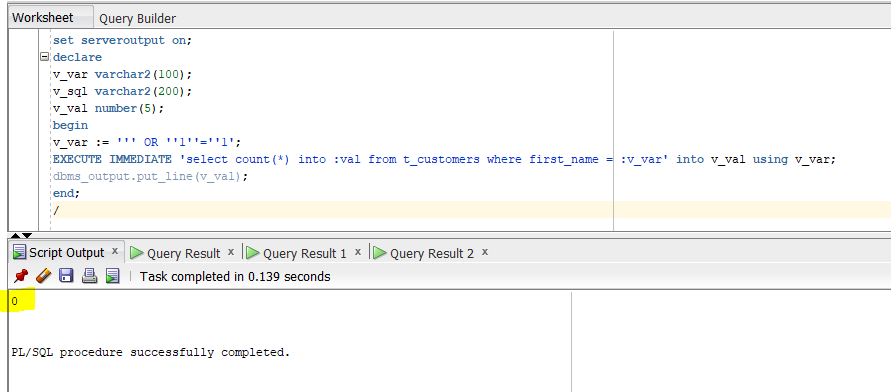
1. safeguarding sensitive data and preventing SQL Injection.

* When string concatenation is used in SQL with parameter RISK of SQL injection attack is possible like below. The table have 2 records in total and all are brought by passing some attack string like **''' OR ''1''=''1'**

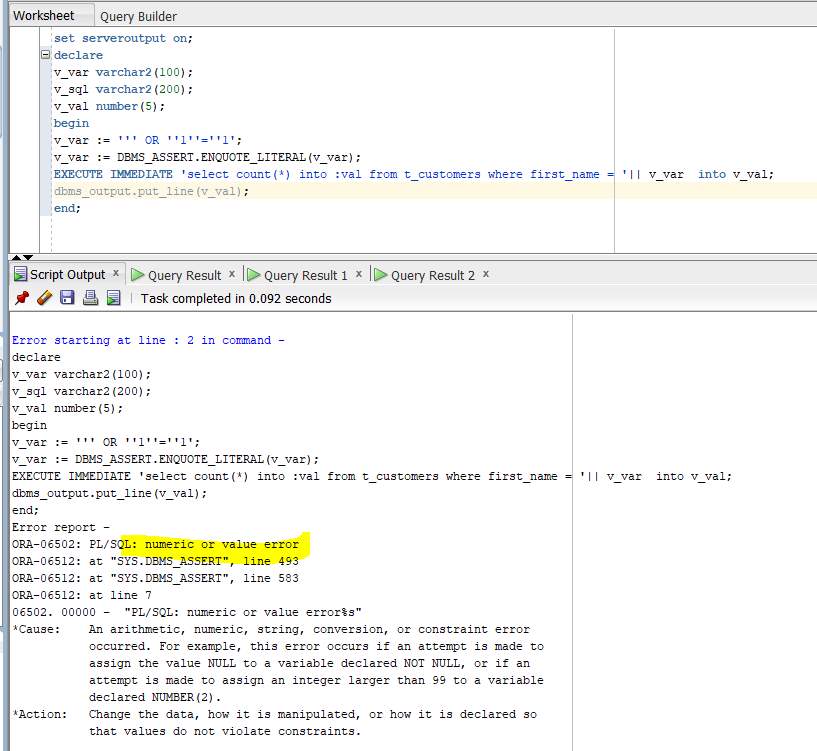


* Can be avoided by using bind variable or DBMS\_ASSERT.ENQUOTE\_LITERAL

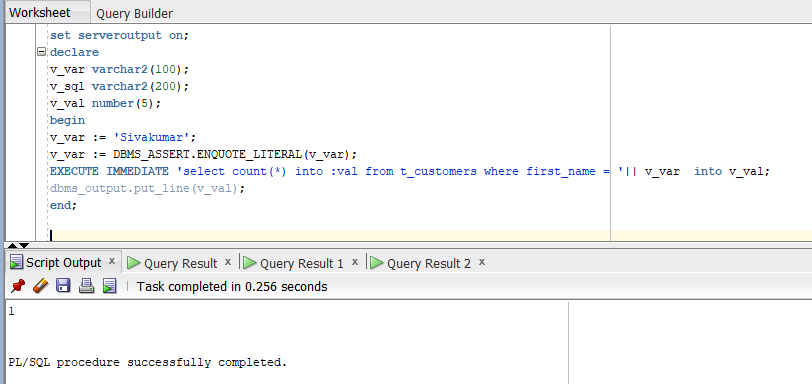
Bind variable



DBMS\_ASSERT.ENQUOTE\_LITERAL gives numeric or value error which can be handled in exception block.

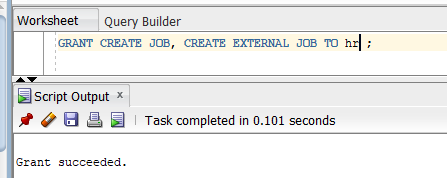


Correct output when proper first\_name is passed

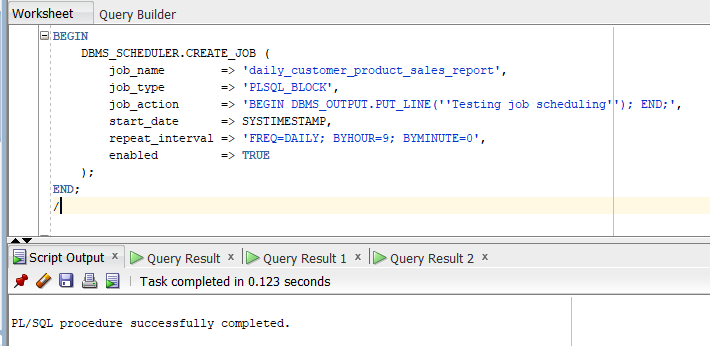


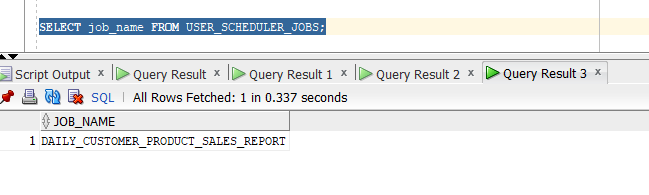
1. Demo for scheduling job.

Grant privilege to user for creating job

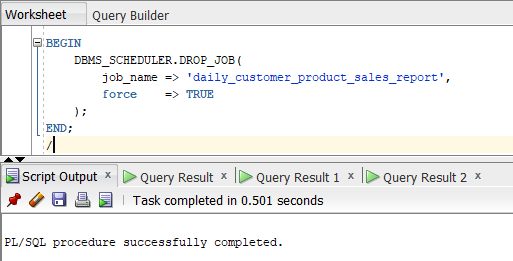


Create job





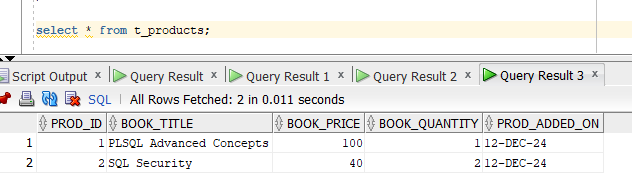
Drop job



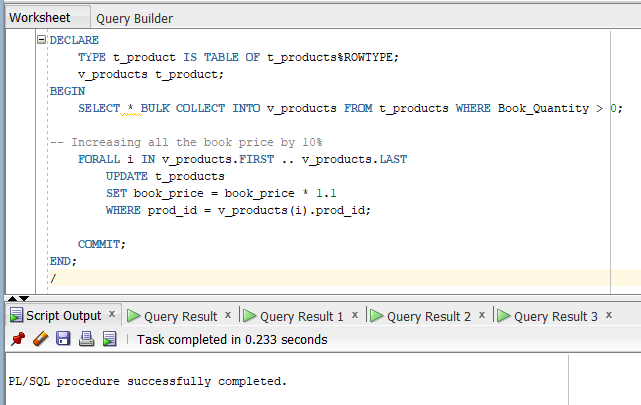
1. Caching and Bulk Processing demo:

Increasing the book\_price for all the product in t\_product table by 10%. It will be helpful when millions of rows are present in real time scenario.

Before update



Bulk collect processing to increase price by 10%



After processing

