**44-560 Advanced Topics in Database Systems**

**Assignment-04: Transaction Management**

Are you excited to know what happens when two concurrent transactions are being executed on the database?

Please follow the below steps to know by yourself.

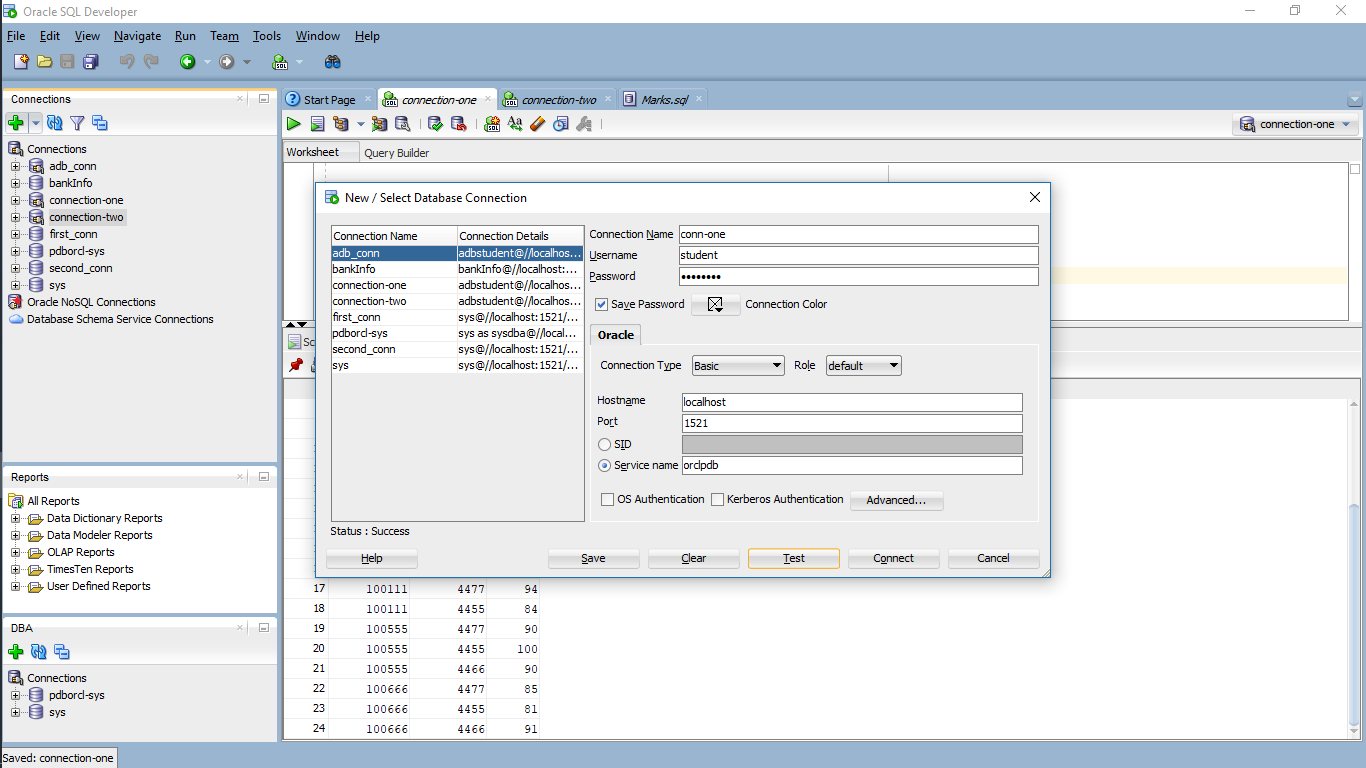
1. Create a user and grant Connect, resource and DBA permissions to the user in SQL plus. Refer to the SQL help worksheet provided earlier.

**For creating a user:** CREATE USER <username> IDENTIFIED BY <Password>;

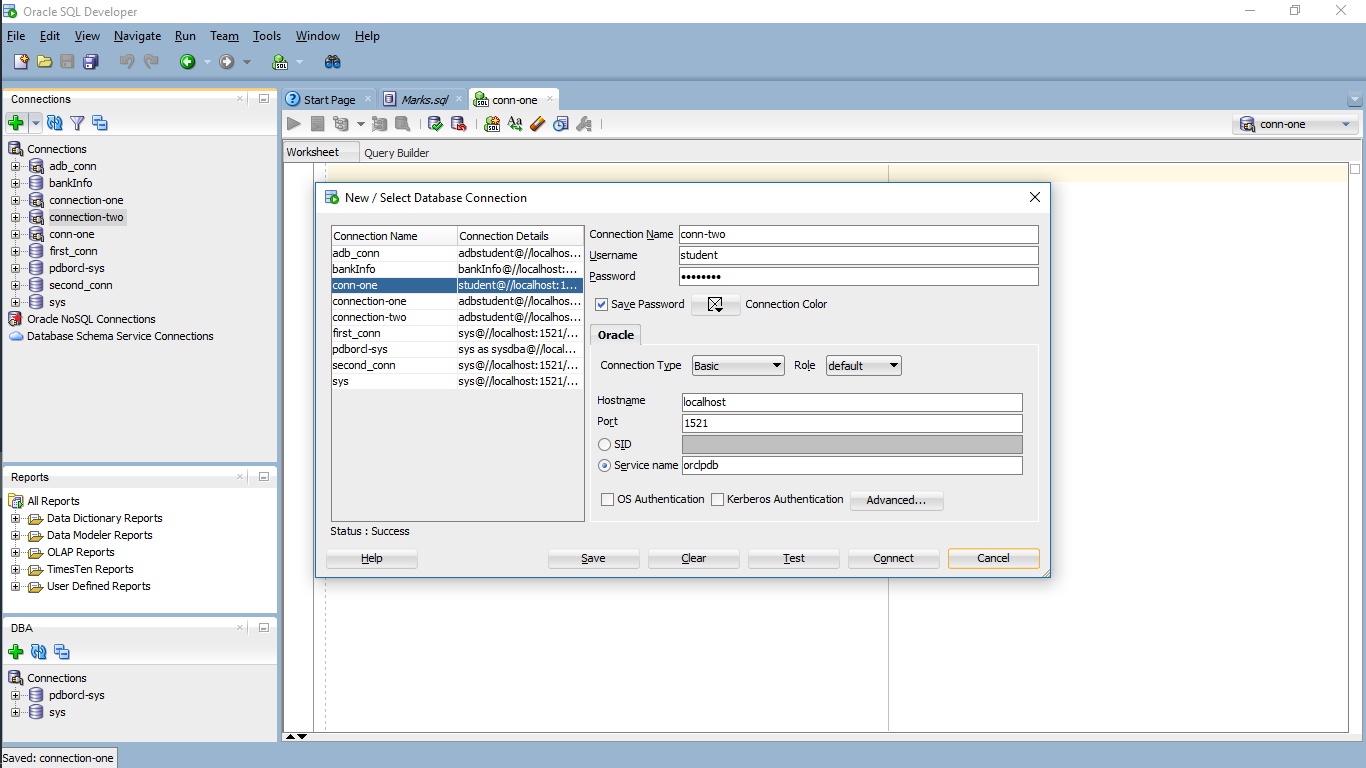
**For granting permissions:** GRANT CONNECT, RESOURCE, DBA TO <username>;

1. Create two different connections in SQL developer with the same user and same service name i.e., orclpdb or pdborcl (whatever you have given while installing the Oracle).

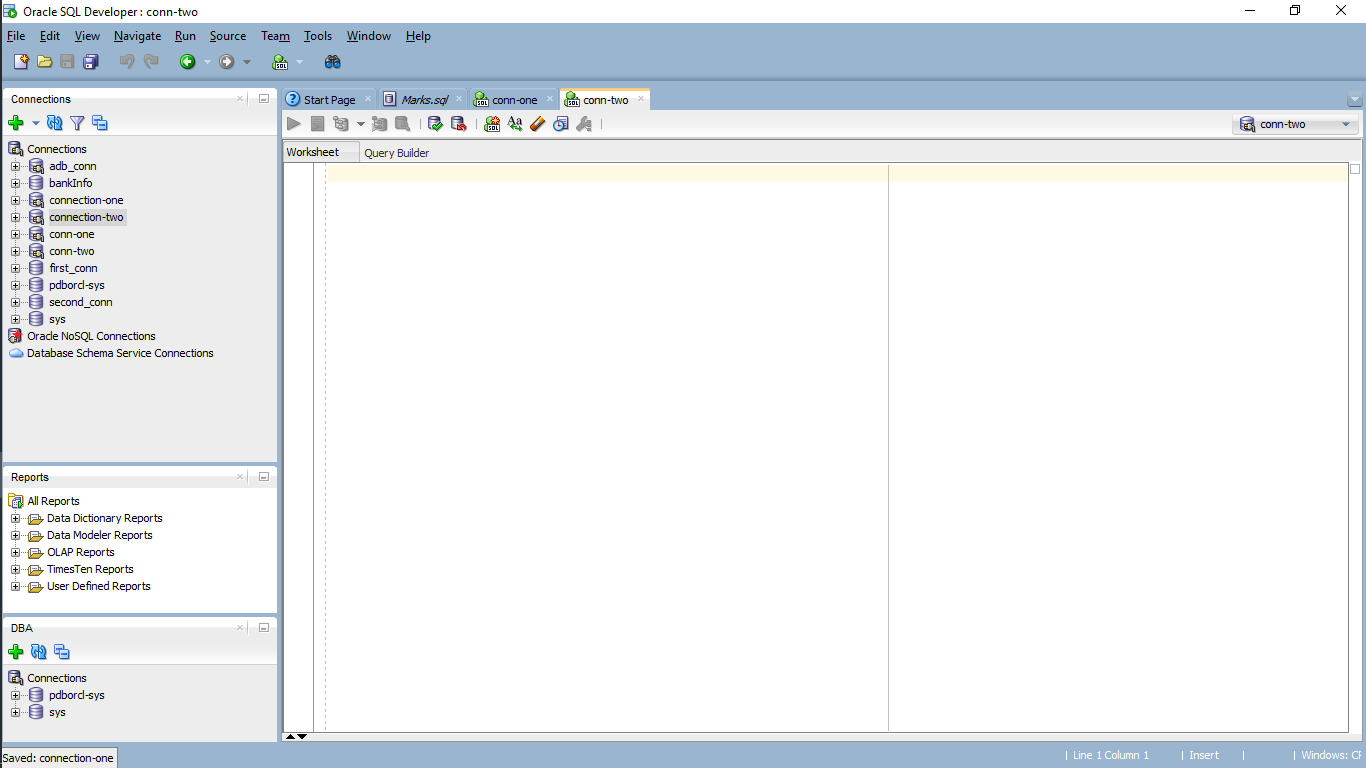
First Connection name: connection-one



Second Connection name: connection-two



1. Open the SQL worksheets of both the connections by right clicking the connections.

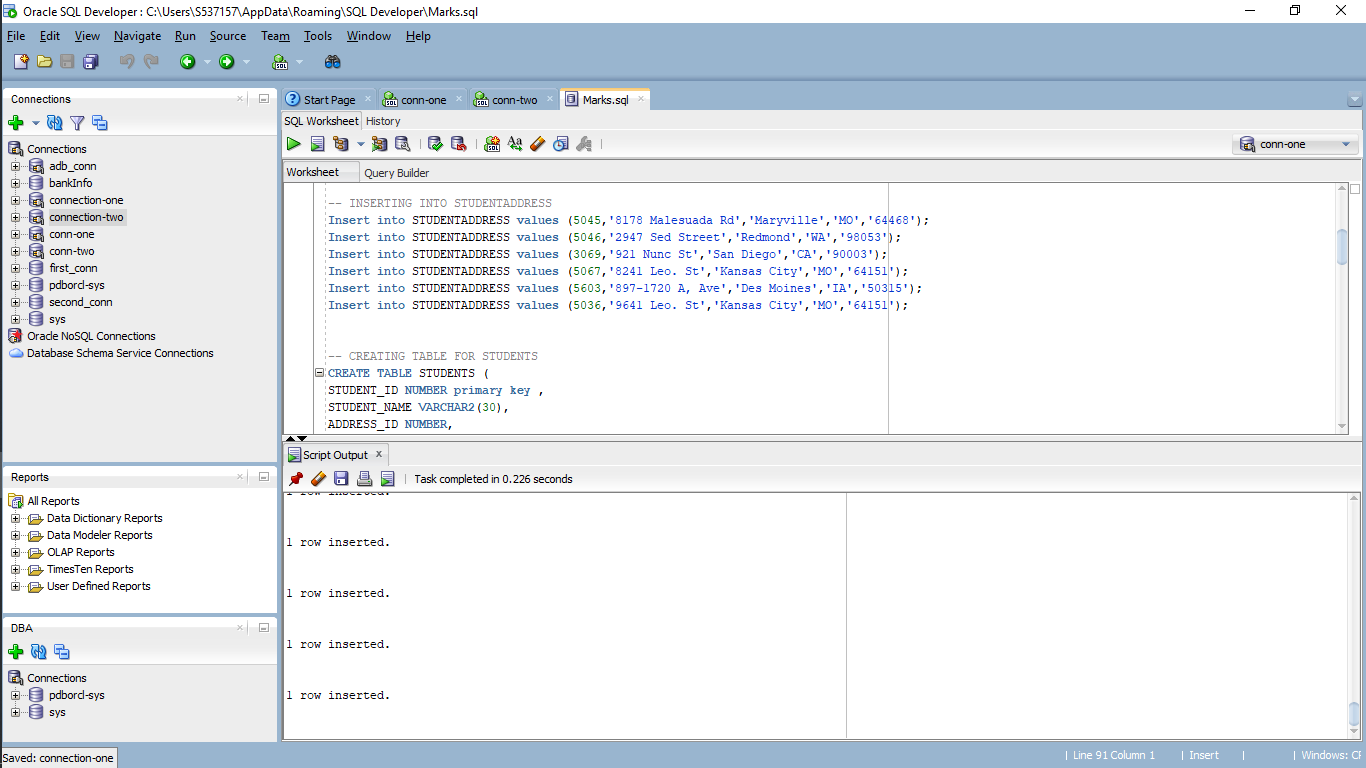


1. Open the Marks.sql file provided below and run the file line by line. Select any of the two connections that we created while running the script file.

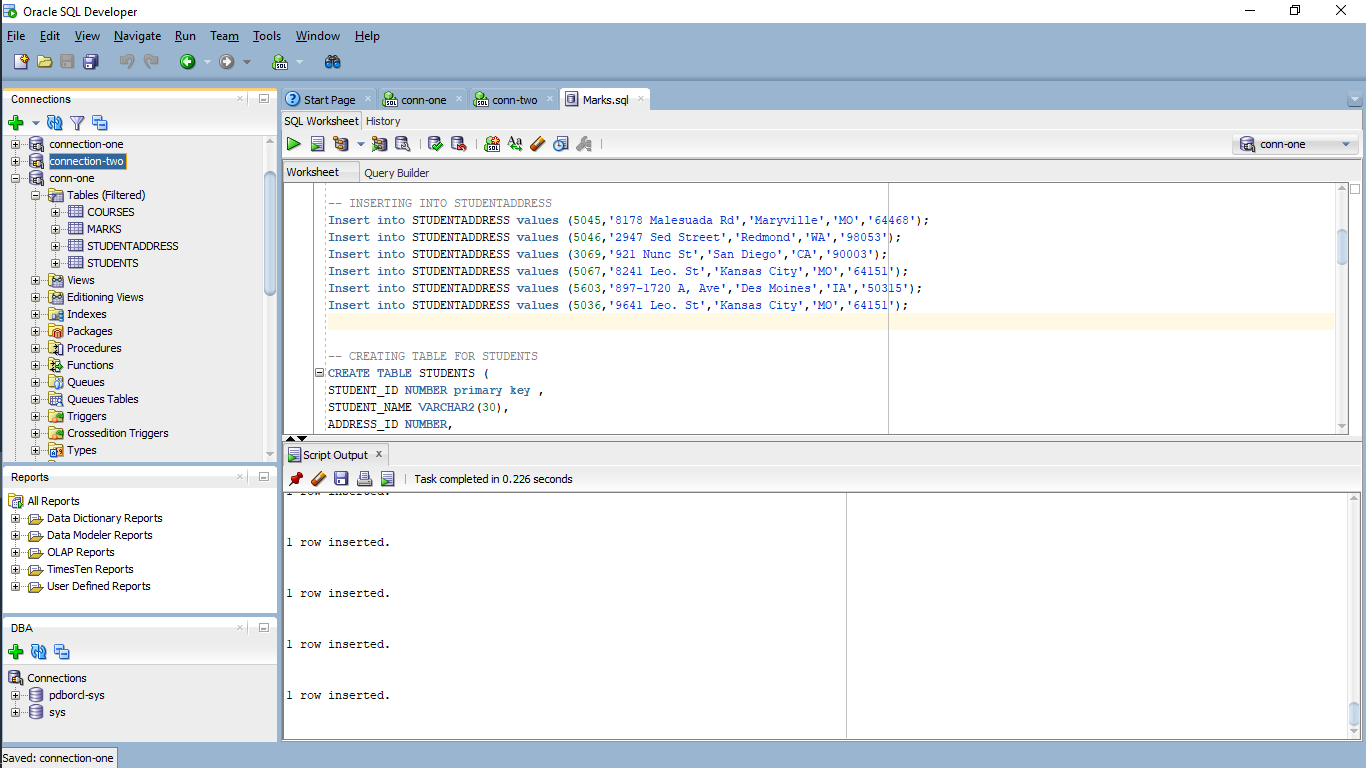
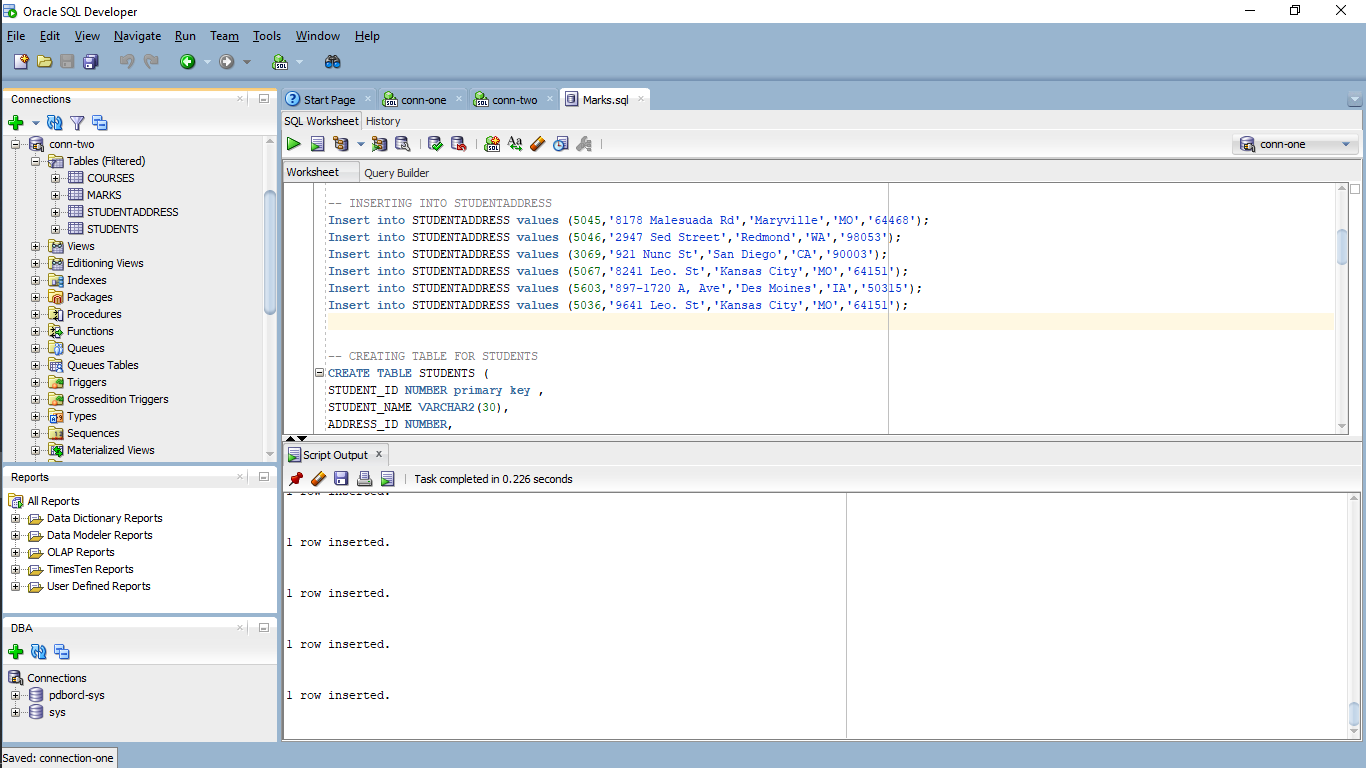
**SQL file:**



This process results as the following screen.



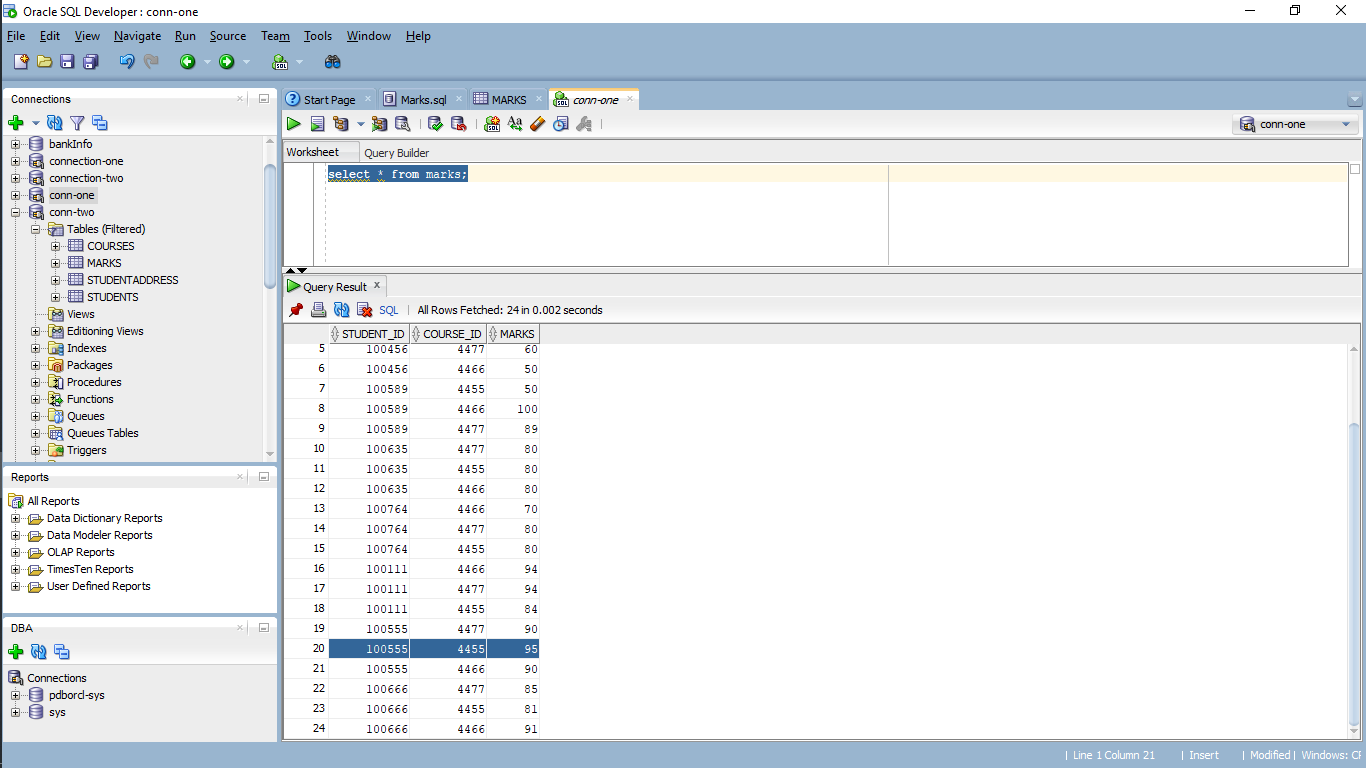
1. After running the script, you should observe same tables created in both the connections.

1. Now let us test by executing concurrent transactions on the same database from the two connections that we have created. We will update the marks of student with STUDENT\_ID **‘100555‘** in subject with SUBJECT\_ID **‘4455’** in both the connections.

In the first connection run the below queries in the order specified:

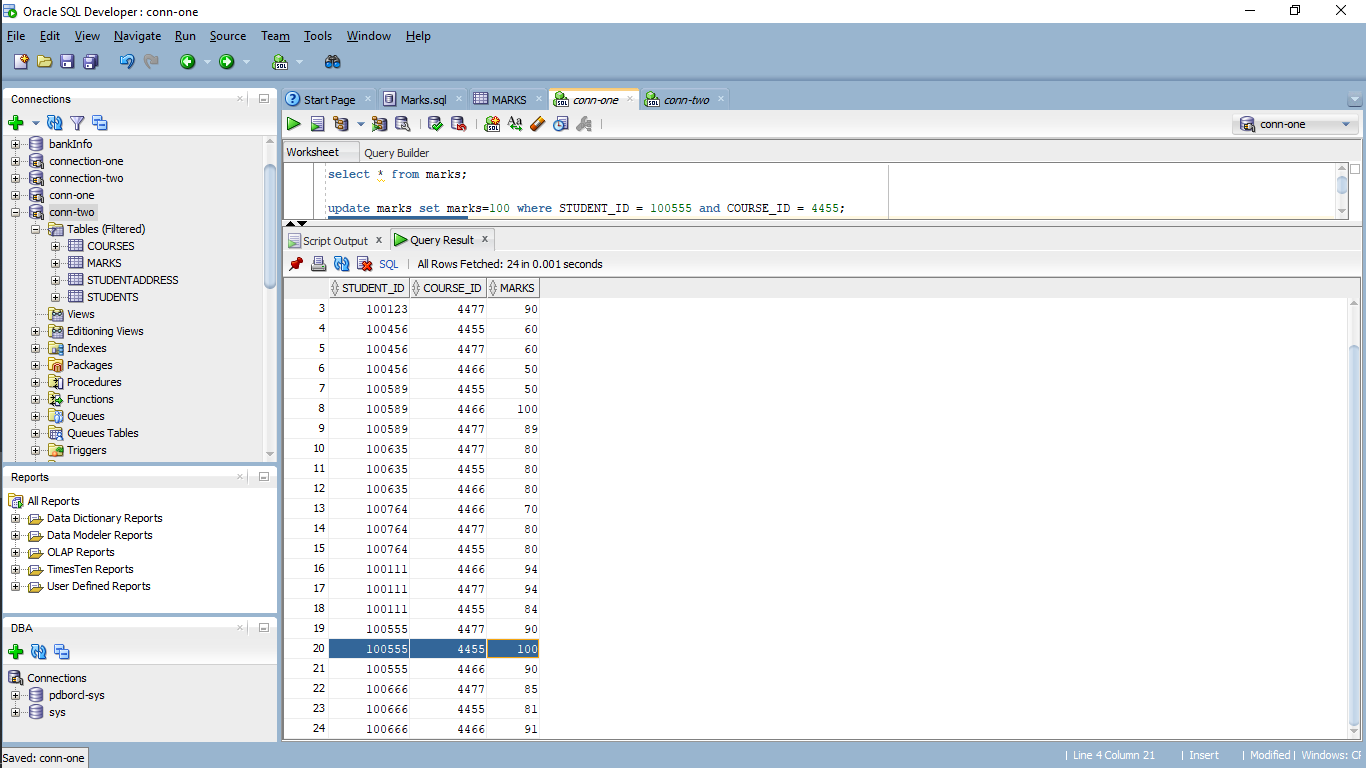
1. select \* from marks;



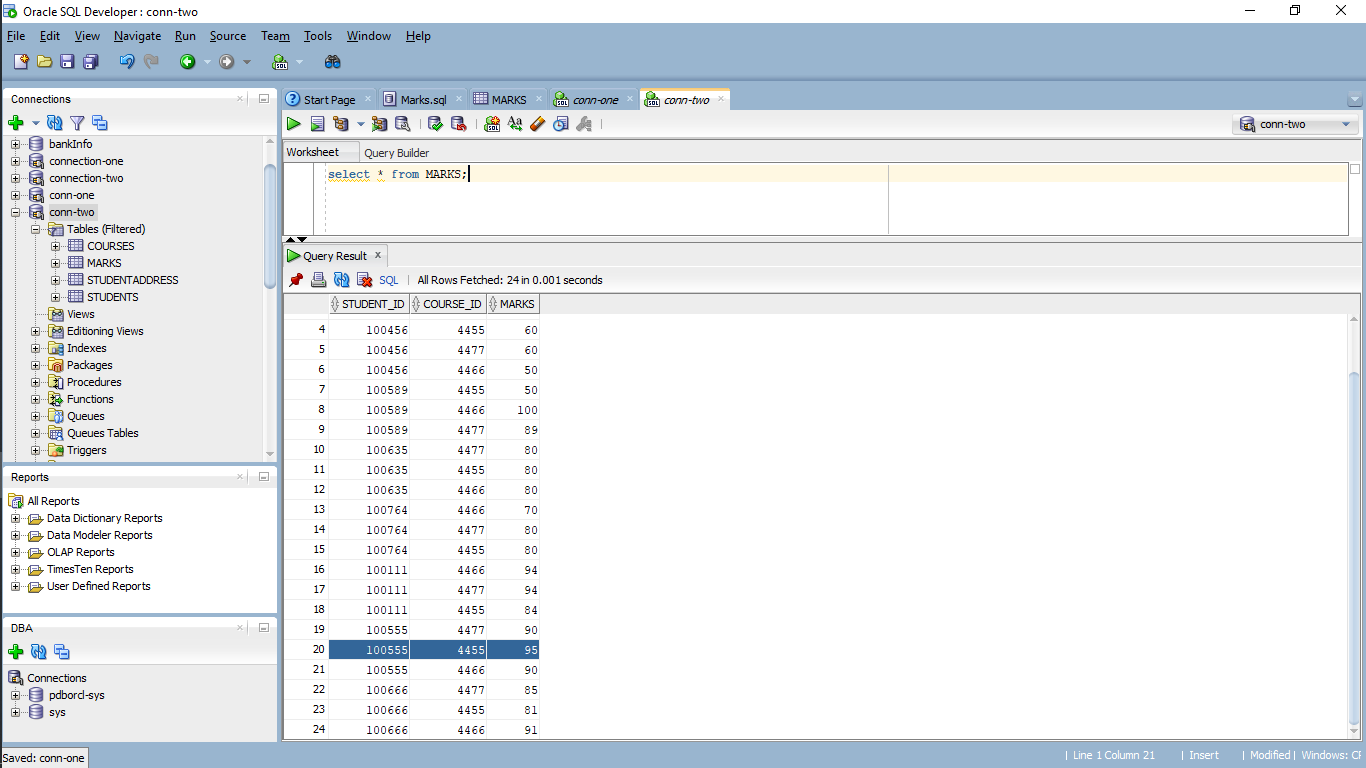
1. Now update the marks to 100 and check the description:

Use the below query to change it.

update marks set marks=100 where STUDENT\_ID = 100555 and COURSE\_ID = 4455; and check the data once again.



1. In the second connection run the below queries in the order specified:
2. Select \* from marks;



Here you can observe that the mark is still 95.

**Answer the below questions:**

1. Why the marks is still 95 even though we have updated it in First connection?

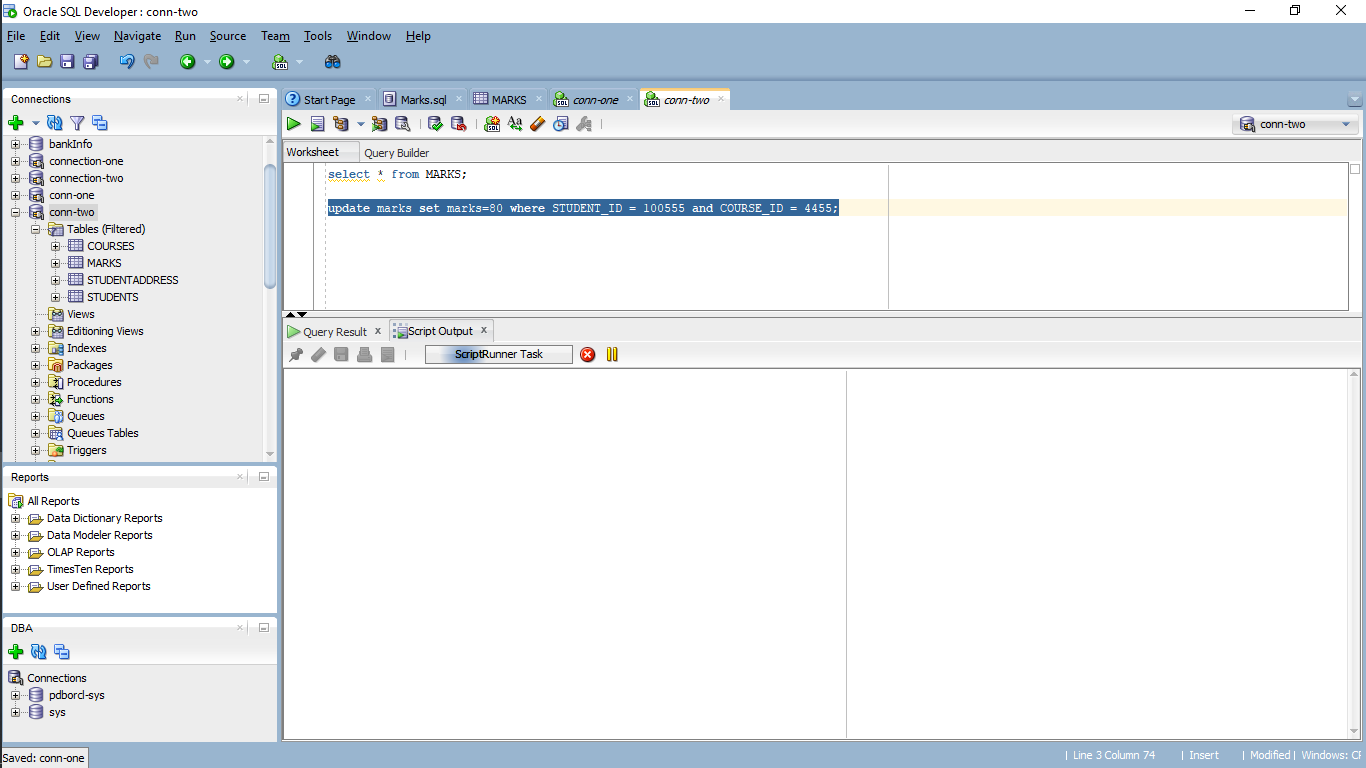
*Ans:* Because, Inconsistent retrievals in both connection-one and connection-two which are connected to one database, this issue occur because here we updated the data of one row using connection-one and in another one we display the data using connection-two but still data is not updated. Here we just update the data, but we didn’t make any commit to make changes in database it shows still 95 even though we have updated it in First connection.

b. Update the marks to 80 in the second connect.

update marks set marks=80 where STUDENT\_ID = 100555 and COURSE\_ID = 4455;

You can see that the task is still executing and spinning script output.

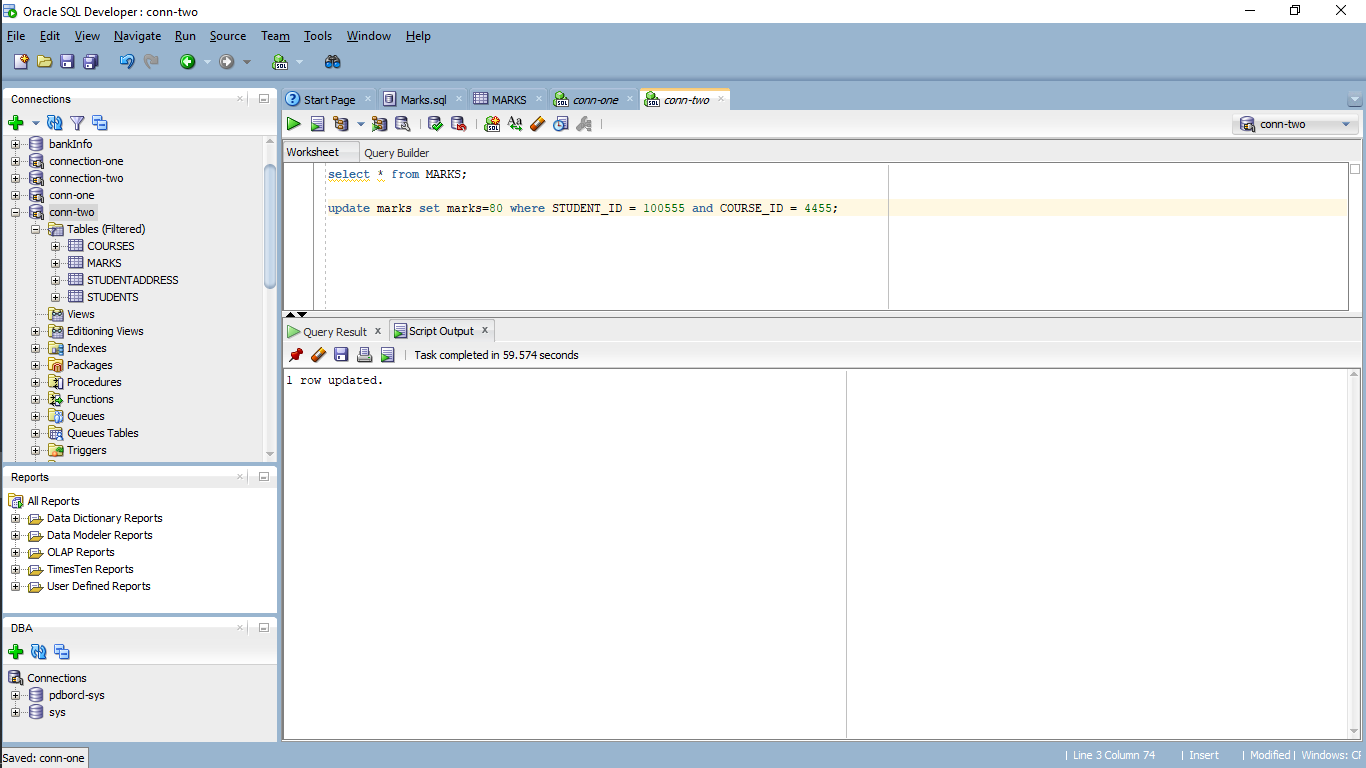
*Ans:* Yes, after the update of the marks to 80 in the second connection we can see the task and it was running in spinning manner it means(Loop).



1. Commit the update we performed in first connection.

commit;

You can see that row is updated and the output in the second connection stopped spinning.



**Answer the below questions:**

1. Why was the output in second connection continuously spinning and not completing before?

*Ans:* The data required for performing the given query in the second connection is locked by the first transaction to avoid inconsistent retrieval. An exclusive lock must be used by a transaction that wants to write to an object; with an exclusive lock, no other transaction can access the object. Till the locks are released the second one keeps spinning.

1. Why did the update complete in the second connection, once we committed in the first connection?

*Ans:* After committing in connection-one, I received a notification in connection-two stating that one row had been modified. Because we are in the next waiting state (time stamping) in the queue, after committing in connection-one row level locking will be disabled, allowing us to access the database.

1. What are the of marks for student with STUDENT\_ID = 100555 and COURSE\_ID = 4455 on each connection? Is the value the same or different in both connections? Why? (don’t commit in the second connection)

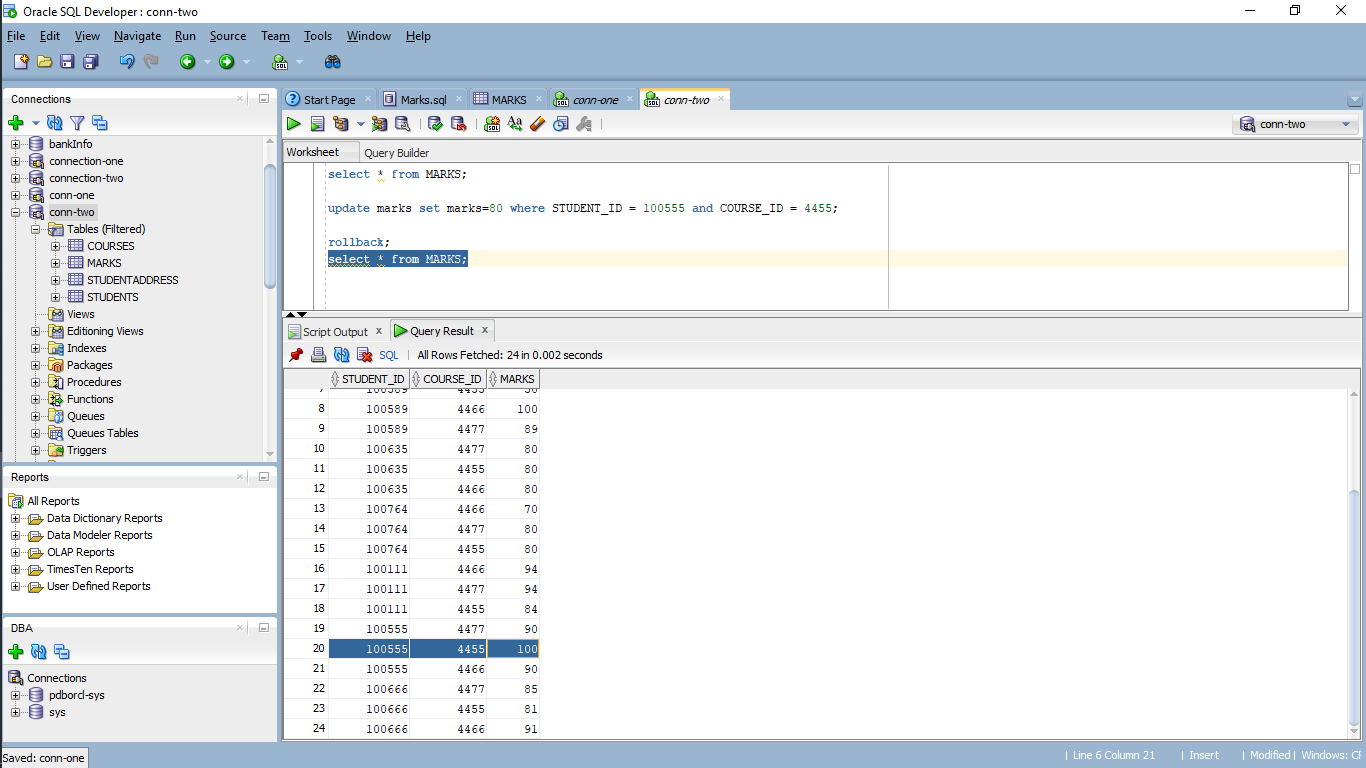
*Ans:* In connection-one I got marks value 100. In connection-two I got marks value 80. Because we updated marks data in connection-one as 100 and connection-two as 80 before, the values of both the connections are different for each other. In addition, we committed the data in the connection-one but not in connection-two. However, due of uncommitted data, we are still getting value 80 in connection-two.

1. Now, in the second connection, rollback the changes and check the marks value in that connection.

rollback;

select \* from marks;

You can see that the description value as “100”



**Answer the below question:**

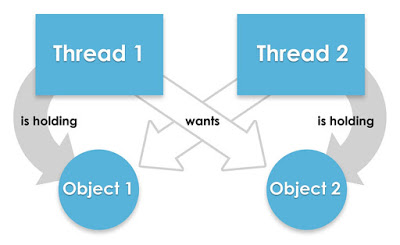
1. What happened immediately after we rollback in second connection?

*Ans:* Uncommitted data is erased after a rollback. As a result, we get the

marks value of 100 because, it commits the data after updating the value in connection-one. So, the database was updated, but in connection-two, we only updated the row, not the data, which is why the data was not reflected in the database.

**Questions about Transaction Management:**

1. In terms of a database, what is the picture below showing?



*Ans*: The picture displayed is Deadlock. Each thread is holding on to the

objects which are wanted by the other threads.

1. Consider the following two transactions made within a bank database:
   1. **T1:** The bank is preparing a report and runs a query that shows the sum of the balances of all its accounts.
   2. **T2:** A balance transfer of $100 is executed from account X to account Y in the following manner:

|  |  |  |
| --- | --- | --- |
| 1 | Read the balance of X | X: 400 |
| 2 | Compute X-100 |  |
| 3 | Write new balance of X | X: 300 |
| 4 | Read the balance of Y | Y: 300 |
| 5 | Compute Y+100 |  |
| 6 | Write new balance of Y | Y: 400 |

Suppose transaction T1 occurs after Line 3 of T2, but before Line 6 of T2. In that case, the sum computed in T1 will not have included the correct balance for account Y. What type of concurrency control problem is this?

*Ans:* The type of concurrency is Inconsistent for the retrieval as the sum of data values is displayed in T1 and the data is updating in T2.

**Submission:**

**Rename this given file with your Lastname\_Assignment04 where the last name is your last name. Please highlight your answers in yellow color and submit your word doc file.**