Name : Dev Kshitij Patel

Lab 06 – Transactions and Security

# Objectives:

The purpose of this lab is to introduce the student to both transactions and security. In the real-world, databases tasks often involve multiple steps and if any step in the middle fails, the procedure is a failure. This lap walks the student through a couple transactions and lets them learn how various steps have varying consequences that they need to be aware of.

By the end of this lab, the student will be able to:

* Describe the steps of a transaction, how a transaction begins and ends and walk through live scenarios of a variety of transactions
* Understand and act appropriately on what needs to be done in the case of transaction failure
* Grant and revoke permissions to and from other users and public users from the database

# Submission:

***You must supply screenshot of each statement run with output in sequence in a single file. Answer any questions in the same file.***

You will use following data to complete the given tasks:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **employeeNumber** | **lastname** | **firstname** | **extension** | **email** | **OfficeCode** | **reportsTo** | **jobTitle** |
| 100 | Patel | Ralph | 22333 | rpatel@mail.com | 1 | NULL | Sales Rep |
| 101 | Denis | Betty | 33444 | [bdenis@mail.com](mailto:bdenis@mail.com) | 4 | NULL | Sales Rep |
| 102 | Biri | Ben | 44555 | [bbirir@mail.com](mailto:bbirir@mail.com) | 2 | NULL | Sales Rep |
| 103 | Newman | Chad | 66777 | [cnewman@mail.com](mailto:cnewman@mail.com) | 3 | NULL | Sales Rep |
| 104 | Ropeburn | Audrey | 77888 | aropebur@mail.com | 1 | NULL | Sales Rep |

* ***SET TRANSACTION READ WRITE*** starts a new transaction.
* ***COMMIT*** commits the current transaction, making its changes permanent.
* ***SAVEPOINT <name>***  sets a pointer to a location that can be rolled back to.
* ***ROLLBACK*** rolls back the current transaction, canceling its changes.
* ***SET autocommit*** disables or enables the default ***autocommit*** mode for the current session.

**Tasks:**

It is very important that these tasks be performed in the order presented here for maximum learning.

## PART A - Transactions

1. List the 4 ways that we know that a transaction can be started  
   Ans: The transaction which starts from 4 ways that are listed below:
2. The user begins a new transaction after connecting to the server again and having a blank page available.
3. The user starts a new transaction by using Oracle SQL's BEGIN command.
4. The user issues a COMMIT command, which commits the ongoing transaction and initiates a new one.
5. Any DDL statement is executed by the user. This automatically initiates a new transaction and causes the current transaction to auto-commit.
6. Using SQL, create an **empty** table, that is the same as the employees table, and name it ***newEmployees***.

Ans: A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Execute the following commands.

SET AUTCOMMIT OFF;  
 SET TRANSACTION READ WRITE;

A screenshot of a computer

Description automatically generated

1. Write an INSERT statement to populate the newEmployees table with the rows of sample data. Insert the NULL value for the reportsTo column.

A screenshot of a computer

Description automatically generated

1. Create a query that shows all the inserted rows from the newEmployees table. How many rows are selected?

A screenshot of a computer

Description automatically generated

1. Execute the rollback command. Display all rows and columns from the newEmployees table. How many rows are selected?

Ans) ZERO rows

A screenshot of a computer

Description automatically generated

1. Repeat Task 4. Make the insertion permanent to the table newEmployees. Display all rows and columns from the newEmployee table. How many rows are selected?

Ans) ONE row is selected.

A screenshot of a computer

Description automatically generated

1. Write an update statement to update the value of column jobTitle to ‘unknown’ for all the employees in the newEmployees table.

Ans) A screenshot of a computer

Description automatically generated

1. Make your changes permanent.

Ans) A screenshot of a computer

Description automatically generated

1. Execute the rollback command.
   1. Display all employees from the newEmployees table whose job title is ‘unknown’. How many rows are still updated?
   2. Was the rollback command effective?
   3. What was the difference between the result of the rollback execution from Task 6 and the result of the rollback execution of this task?

Ans) a) One row is fully updated where a change in jobtitle is seen.A screenshot of a computer

Description automatically generated

b) No, it is not effective.

c)There is big difference between the result of ROLLBACK after task 6 and ROLLBACK from this task. Since before task 6, we didn’t COMMIT (Save) the table, so in the task 6, when we do ROLLBACK, it delete all the preceding executed data. On the other hand, We have COMMIT in task 9. So all the preceding data will be saved. By doing ROLLBACK in this task, it jumps back to the saved data.

A screenshot of a computer

Description automatically generated

1. Create a VIEW, called **vwNewEmps**, that queries all the records in the newEmployees table sorted by last name and then by first name.  
   Ans) A screenshot of a computer

   Description automatically generated
2. Rerun the data insertion from Task 4 (copy the code down to Task 14 and run it) with new id.

Ans) A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Set a Savepoint, called ***insertion***, after inserting the data

Ans) A screenshot of a computer

Description automatically generated

1. Rerun the update statement from Task 8 and run a query to view the data (copy the code down and run it again)

Ans) A screenshot of a computer

Description automatically generated

1. Rollback the transaction to the Savepoint created in task 15 above and run a query to view the data.  
   What does the data look like i.e. describe what happened?

Ans) UPDATE should be undone, since it jumps back to the savepoint, and whatever happens after this savepoint wont be saved.

A screenshot of a computer

Description automatically generated

1. Use the basic for of the rollback statement and again view the data. Describe what the results look like and what happened.

Ans) In this, it should rollback all the way to last commit. A screenshot of a computer

Description automatically generated

## Part B – Permissions (You need username of a partner for this part)

1. Write a statement that denies all access to the newemployees table for all public users

Ans) A screenshot of a computer

Description automatically generated

1. Write a statement that allows a classmate (use their database login) read only access to the newemployees table. [Optional Question]

Ans) A screenshot of a computer

Description automatically generated

1. Write a statement the denies all access to the newemployees table for the same classmate. [Optional Question]

Ans) A screenshot of a computer

Description automatically generated

## Part C – Clean up

1. Write statements to permanently remove the view and table created for this lab

Ans) A screenshot of a computer

Description automatically generated