# SɅRɅSWɅTI

## Department of Computer Engineering

**College of Engineering**

SɅRɅSWɅTI

### Experiment 7

**Aim: Perform TCL and DCL commands Resources required:** P-IV and above, Oracle **Theory:**

Commit, Rollback and Savepoint SQL commands

Transaction Control Language(TCL) commands are used to manage transactions in the database. These are used to manage the changes made to the data in a table by DML statements. It also allows statements to be grouped together into logical transactions.

COMMIT command

COMMIT command is used to permanently save any transaction into the database.

When we use any DML command like INSERT, UPDATE or DELETE, the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use the COMMIT command to mark the changes as permanent. Following is commit command's syntax,

Commit;

ROLLBACK command

This command restores the database to last committed state. It is also used with SAVEPOINT command to jump to a savepoint in an ongoing transaction.

If we have used the UPDATE command to make some changes into the database, and realise that those changes were not required, then we can use the ROLLBACK command to rollback those changes, if they were not commited using the COMMIT command.

Following is rollback command's syntax, Rollback;

SAVEPOINT command

SAVEPOINT command is used to temporarily save a transaction so that you can rollback to that point whenever required.

# SɅRɅSWɅTI

## Department of Computer Engineering

**College of Engineering**

SɅRɅSWɅTI

Following is savepoint command's syntax, savepoint

savepoint a2

#### DCL (Data Control Language)

As always, begin by connecting to your server where Oracle is hosted, then connect to Oracle itself as the SYSTEM account.

The SYSTEM account is one of a handful of predefined administrative accounts generated automatically when Oracle is installed. SYSTEM is capable of most administrative tasks, but the task we’re particularly interested in is account management.

###### Creating a User

Once connected as SYSTEM, simply issue the [CREATE USER](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8003.htm) command to generate a new account.

###### create user <username> identified by <password>;

Here we’re simply creating a rina account that is IDENTIFIED or authenticated by the specified password.

###### The Grant Statement

With our new account created, we can now begin adding privileges to the account using

the [GRANT](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9013.htm) statement. GRANT is a very powerful statement with many possible options, but the core functionality is to manage the privileges of both users and roles throughout the database.

###### Providing Roles

Typically, you’ll first want to assign privileges to the user through attaching the account to various roles, starting with the CONNECT role:

###### GRANT CONNECT TO <username>;

In some cases to create a more powerful user, you may also consider adding the RESOURCE role (allowing the user to create named types for custom schemas) or even the DBA role, which allows the user to not only create custom named types but alter and destroy them as well.

###### GRANT CONNECT, RESOURCE, DBA TO <username>;

**SɅRɅSWɅTI**

**Department of Computer Engineering**

**College of Engineering**

SɅRɅSWɅTI

1. **Assigning Privileges**

Next you’ll want to ensure the user has privileges to actually connect to the database and create a session using GRANT CREATE SESSION. We’ll also combine that with all privileges using GRANT ANY PRIVILEGES.

GRANT CREATE SESSION TO **<username>**;

We also need to ensure our new user has disk space allocated in the system to actually create or modify tables and data, so we’ll GRANT TABLESPACE like so:

GRANT UNLIMITED TABLESPACE TO **<username>**; GRANT RESOURCE TO RINA;

GRANT SELECT, INSERT, UPDATE, DELETE ON demo TO **<username>**;

**GRANT ALL ON demo TO <username>;**

connect rina/abc;

select \* from system.demo;

delete from system.demo where rno=1; connect system/oracle;

revoke delete on demo from **<username>**;

###### Conclusion: We performed TCL and DCL command successfully in MySQL Command Line.

###### Code and Output:

###### DCL command:

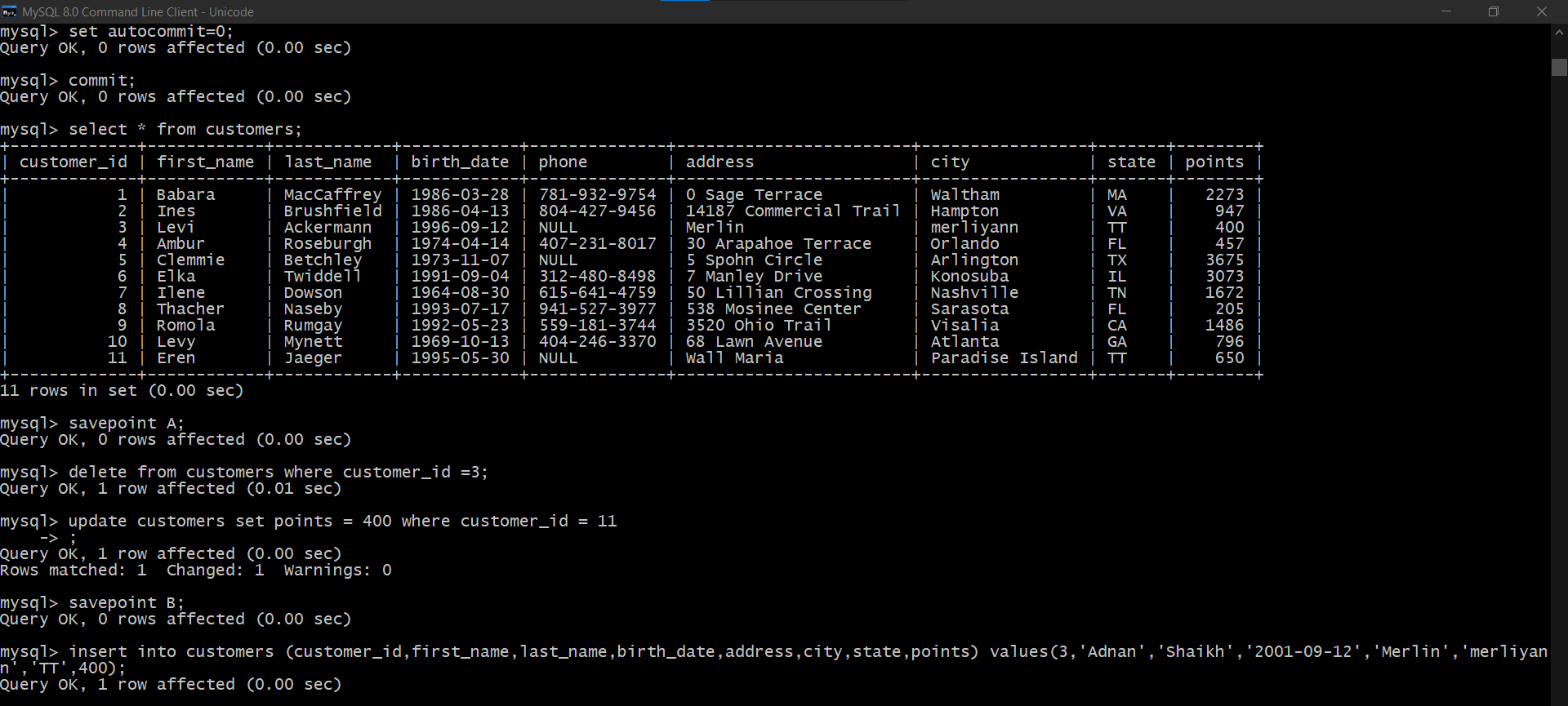
###### 

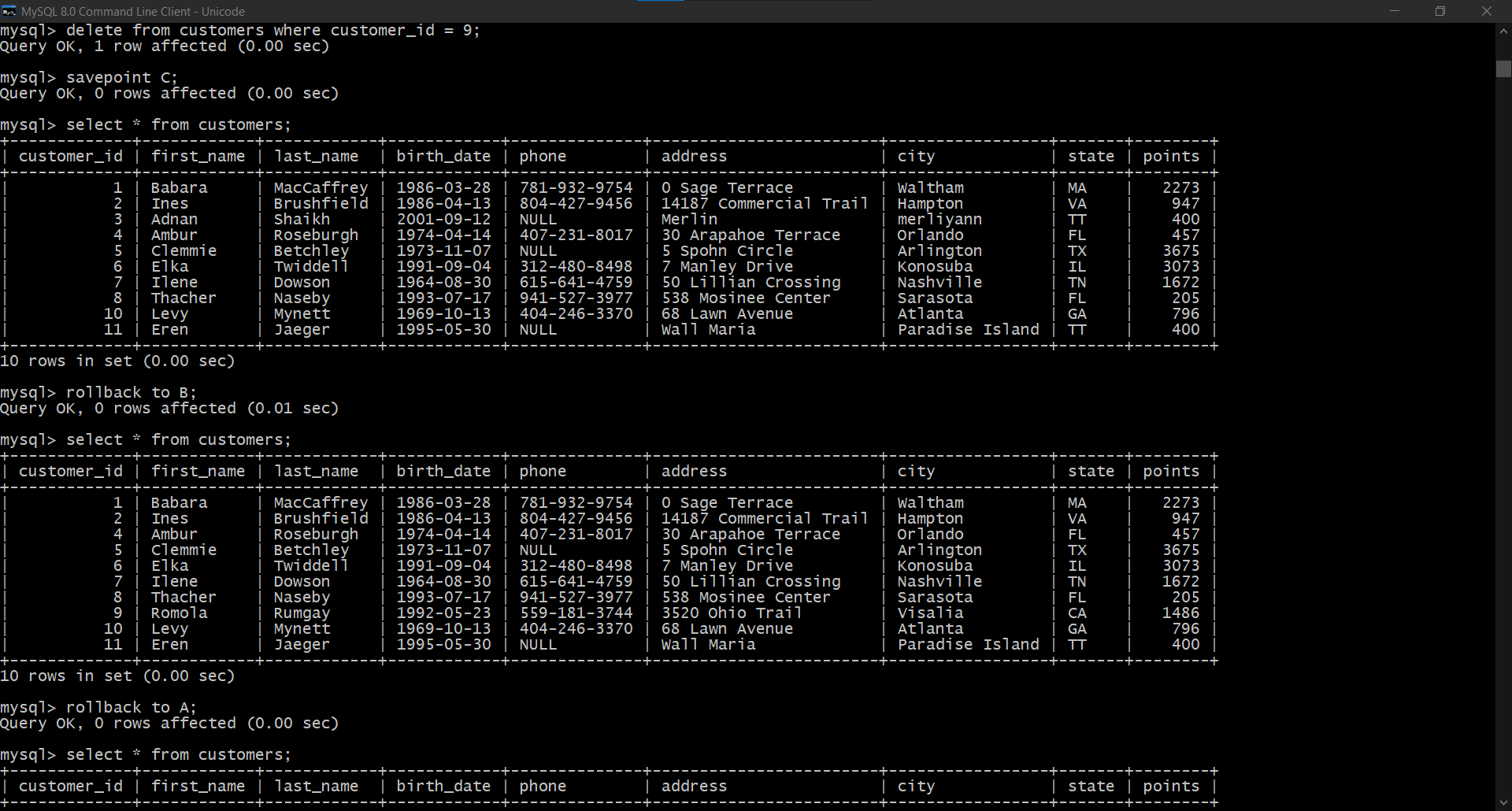
###### 

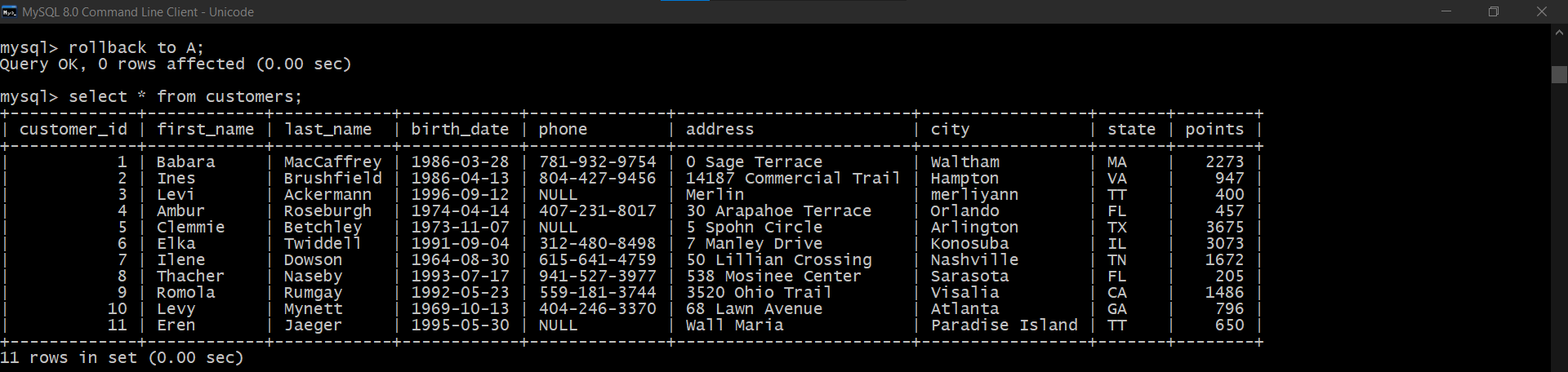
###### 

###### 

## TCL command:







SɅRɅSWɅTI