# Oracle lab 07
# Transactions

A transaction is a unit of work that is performed against a database. Transactions are units or sequences of work accomplished in a logical order, whether in a manual fashion by a user or automatically by some sort of database program.

A transaction is the propagation of one or more changes to the database. For example, if you are creating a record or updating a record or deleting a record from the table, then you are performing a transaction on that table. It is important to control these transactions to ensure the data integrity and to handle database errors.

Practically, you will gather many SQL queries into a group, and you will execute all of them together as a part of a transaction.

### Transaction Control

The following commands are used to control transactions.

- COMMIT − to save the changes.
- ROLLBACK − to roll back the changes.
- SAVEPOINT − creates points within the groups of transactions in which to ROLLBACK.
- SET TRANSACTION − Places a name on a transaction.

### Transactional Control Commands

Transactional control commands are only used with the DML Commands such as - INSERT, UPDATE and DELETE only. They cannot be used while creating tables or dropping them because these operations are automatically committed in the database.

## Saving DML Changes

To save changes to the database, you must issue a **commit**.

So, to ensure you preserve the row you inserted, commit afterward:

```
Insert into <table_name> (col1, col2, ...)  values (‘value1', 'value2', ...);
```

```
commit;
```

The row is only visible to other users after you commit. Before this point, only your session (database connection) can see the new data. Many tools have an autocommit property (show autocommit to see its state). This will make the **commit** for you. Either after each call to the database. Or when the session ends. LiveSQL commits after each click of the Run button completes.

# Undoing DML

Your code may throw an exception between inserting a row and committing it. So, you may wish to undo the change. You can do this with rollback.

**Rollback** reverts all the changes since your last commit.

```
commit;

Some DML changes;

rollback;
```

When you issue a **rollback**, the database will undo all changes you made since your last **commit**. If you commit between insert and rollback, for example, rollback does nothing. And you need to run delete to remove the row.

**Lab work:**

Open two SQL*Plus windows and connect with DBAINTERVENTION and another user (sys as sysdba, Admin…etc.).

**1) Do the following in DBAINTERVENTION session:**

```
select salary from employee;
update employee set salary=1000000;
```

- Close the window using the X button
- Connect again using DBAINTERVENTION, check if the changes have been made
- What have you noticed?

**2) Update the salaries again and check if the changes were made in the other session (sys, Admin…etc.).**

```
DBAINTERVENTION: update employee set salary=1000000;
SYS: select salary from c##dbaintervention.employee;
```

- What have you noticed?
- Close the window using the exit command
- Check if the changes were made in the other session (sys, Admin…etc.)
- Connect again using DBAINTERVENTION and see if the appropriate changes have been made. What have you noticed?

**3) Make another update on salary, then add a commit statement as follows:**

```
DBAINTERVENTION: update employee set salary=5000000;
DBAINTERVENTION: commit;
```

- Close the window using the X button.
- Check if the changes were made in the other session (sys, Admin…etc.).
- Connect again using DBAINTERVENTION and see if the appropriate changes have been made. What have you noticed?

**4) Rollback:**

```
DBAINTERVENTION: update customer set civ='Mr';
DBAINTERVENTION: select customernumber, civ from customer;

DBAINTERVENTION: rollback;
DBAINTERVENTION: select customernumber, civ from customer;
```

**5) Savepoint:**

```
DBAINTERVENTION: update customer set civ='Mr';
DBAINTERVENTION: select customernumber, civ from customer;
DBAINTERVENTION: savepoint Mr;

DBAINTERVENTION: update customer set fax = '021102040';
DBAINTERVENTION: select customernumber, fax from customer;
```

```
DBAINTERVENTION: savepoint fax;

DBAINTERVENTION: update customer set telpriv = '021102040';
DBAINTERVENTION: select customernumber, telpriv from customer;

DBAINTERVENTION: rollback to fax;
DBAINTERVENTION: select customernumber, telpriv from customer;

DBAINTERVENTION: rollback to Mr;
DBAINTERVENTION: select customernumber, fax from customer;

DBAINTERVENTION: rollback;
DBAINTERVENTION: select customernumber, civ from customer;
DBAINTERVENTION: select customernumber, fax from customer;
DBAINTERVENTION: select customernumber, telpriv from customer;
```

**6) SET TRANSACTION NAME**

```
DBAINTERVENTION: SET TRANSACTION NAME 'some_changes';

DBAINTERVENTION: update customer set civ='Mr';
DBAINTERVENTION: select customernumber, civ from customer;
DBAINTERVENTION: savepoint Mr;

DBAINTERVENTION: update customer set fax = '021102040';
DBAINTERVENTION: select customernumber, fax from customer;
DBAINTERVENTION: savepoint fax;

DBAINTERVENTION: update customer set telpriv = '021102040';
DBAINTERVENTION: select customernumber, telpriv from customer;

DBAINTERVENTION: rollback;
DBAINTERVENTION: select customernumber, civ from customer;
DBAINTERVENTION: select customernumber, fax from customer;
DBAINTERVENTION: select customernumber, telpriv from customer;
```

What have you noticed?

7) Run the following queries :

DBAINTERVENTION: update customer set fax=0000 where customernumber=10;
SYS: update dbaintervention.customer set fax=1111 where customernumber=10;

What have you noticed?

DBAINTERVENTION: update customer set fax=2222 where customernumber=10;
SYS: update dbaintervention.customer set fax=1111 where customernumber=11;

8) Run the following queries :

DBA : SELECT * FROM customer FOR UPDATE;
DBA : select * from customer;
SYS : select * from dbaintervention.customer;

DBA: update customer set FAX = 11111 where customernumber =10;
SYS: update dbaintervention.customer set FAX = 22222 where customernumber =11;

**What have you noticed?**