The Transaction & Data Control language

- Jayendra Khatod

Objectives

After completing this lesson, you should be able to understand the following:

- **Transaction Control Statements**
  - **Rollback, Commit and Save point**
- **The Data Control Language**
- **Controlling User Access to the database**
- **Grant**
- **Revoke**
- **Role**
- **System privileges**

Database Transactions

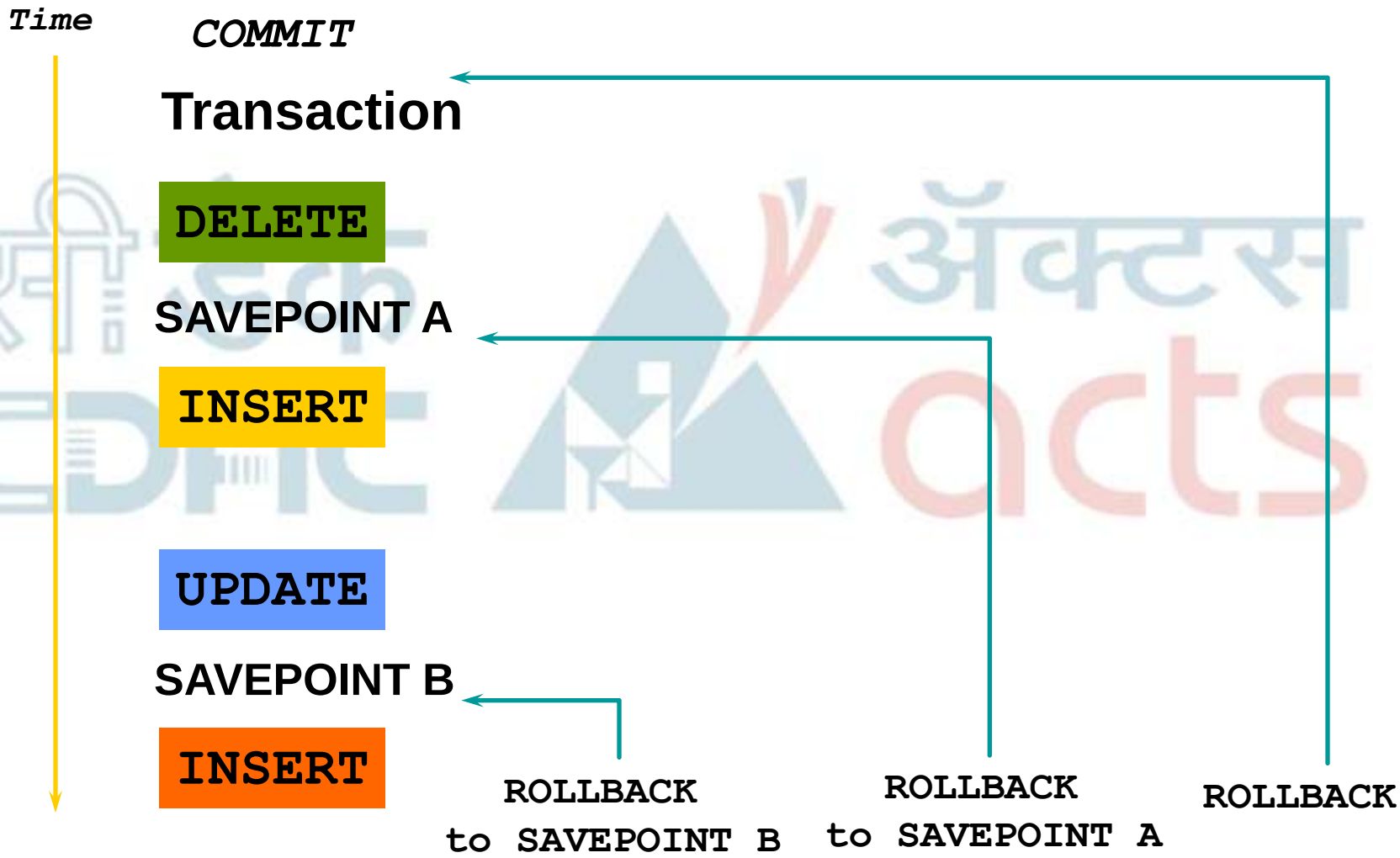
- **Begin when the first DML SQL statement is executed**
- **End with one of the following events:**
  - **A `COMMIT` or `ROLLBACK` statement is issued**
  - **A DDL or DCL statement executes (automatic commit)**
  - **The user exits *i*SQL*Plus**
  - **The system crashes**

Advantages of COMMIT and ROLLBACK Statements

With COMMIT and ROLLBACK statements, you can:

- **Ensure data consistency**
- **Preview data changes before making changes permanent**
- **Group logically related operations**

Controlling Transactions



Rolling Back Changes to a Marker

- **Create a marker in a current transaction by using the `SAVEPOINT` statement.**
- **Roll back to that marker by using the `ROLLBACK TO SAVEPOINT` statement.**

```
UPDATE...
SAVEPOINT update_done;
Savepoint created.
INSERT...
ROLLBACK TO update_done;
Rollback complete.
```

Implicit Transaction Processing

- **An automatic commit occurs under the following circumstances:**
  - **DDL statement is issued**
  - **DCL statement is issued**
  - **Normal exit from SQL*Plus, without explicitly issuing `COMMIT` or `ROLLBACK` statements**
- **An automatic rollback occurs under an abnormal termination of *i*SQL*Plus or a system failure.**

State of the Data Before COMMIT or ROLLBACK

- **The previous state of the data can be recovered.**
- **The current user can review the results of the DML operations by using the `SELECT` statement.**
- **Other users *cannot* view the results of the DML statements by the current user.**
- **The affected rows are *locked*; other users cannot change the data within the affected rows.**

State of the Data after COMMIT

- **Data changes are made permanent in the database.**
- **The previous state of the data is permanently lost.**
- **All users can view the results.**
- **Locks on the affected rows are released; those rows are available for other users to manipulate.**
- **All savepoints are erased.**

- **Make the changes.**

```
DELETE FROM employees
WHERE  employee_id = 99999;
1 row deleted.

INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
1 row inserted.
```

- **Commit the changes.**

```
COMMIT;
Commit complete.
```

State of the Data After ROLLBACK

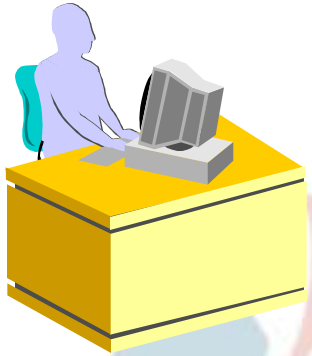
Discard all pending changes by using the ROLLBACK statement:

- **Data changes are undone.**
- **Previous state of the data is restored.**
- **Locks on the affected rows are released.**

```
DELETE FROM copy_emp;
22 rows deleted.
ROLLBACK;
Rollback complete.
```

Controlling User Access

Database administrator



Username and password
Privileges

Users

Privileges

- **Database security:**
  - **System security**
  - **Data security**
- **System privileges: Gaining access to the database**
- **Object privileges: Manipulating the content of the database objects**
- **Schemas: Collections of objects, such as tables, views, and sequences**

System Privileges

- More than 100 privileges are available.
- The database administrator has high-level system privileges for tasks such as:
  - Creating new users
  - Removing users
  - Removing tables
  - Backing up tables

The DBA creates users by using the `CREATE USER` statement.

```
CREATE USER user
IDENTIFIED BY   password;
```

```
CREATE USER  scott
IDENTIFIED BY   tiger;
User created.
```

User System Privileges

- **Once a user is created, the DBA can grant specific system privileges to a user.**

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

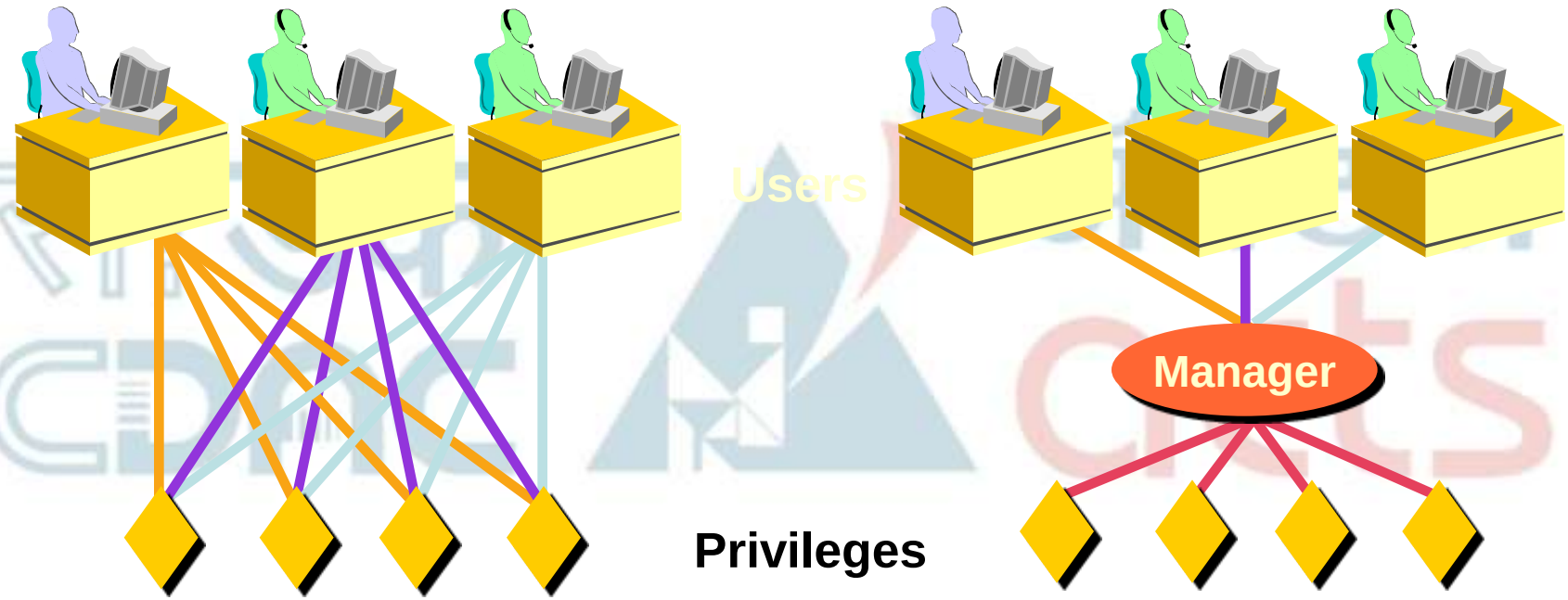
- **An application developer, for example, may have the following system privileges:**
  - **CREATE SESSION**
  - **CREATE TABLE**
  - **CREATE SEQUENCE**
  - **CREATE VIEW**
  - **CREATE PROCEDURE**

Granting System Privileges

The DBA can grant a user specific system privileges.

```
GRANT   create session, create table,
        create sequence, create view
TO      scott;
Grant succeeded.
```

What is a Role?



Allocating privileges without a role

Allocating privileges with a role

Creating and Granting Privileges to a Role

- **Create a role**

```
CREATE ROLE manager;
Role created.
```

- **Grant privileges to a role**

```
GRANT create table, create view
TO manager;
Grant succeeded.
```

- **Grant a role to users**

```
GRANT manager TO DEHAAN, KOCHHAR;
Grant succeeded.
```

Changing Your Password

- **The DBA creates your user account and initializes your password.**
- **You can change your password by using the `ALTER USER` statement.**

```
ALTER USER scott
IDENTIFIED BY lion;
User altered.
```

Object Privileges

Object Privilege	Table	View	Sequence	Procedure
ALTER	√		√	
DELETE	√	√		
EXECUTE				√
INDEX	√			
INSERT	√	√		
REFERENCES	√	√		
SELECT	√	√	√	
UPDATE	√	√		

Object Privileges

- **Object privileges vary from object to object.**
- **An owner has all the privileges on the object.**
- **An owner can give specific privileges on that owner's object.**

```
GRANT      object_priv [(columns)]
ON         object
TO         {user|role|PUBLIC}
[WITH GRANT OPTION];
```

Granting Object Privileges

- **Grant query privileges on the EMPLOYEES table.**

```
GRANT   select
ON      employees
TO      sue, rich;
Grant succeeded.
```

- **Grant privileges to update specific columns to users and roles.**

```
GRANT   update (department_name, location_id)
ON      departments
TO      scott, manager;
Grant succeeded.
```

WITH GRANT OPTION and PUBLIC Keywords

- **Give a user authority to pass along privileges.**

```
GRANT   select, insert
ON      departments
TO      scott
WITH    GRANT OPTION;
Grant succeeded.
```

- **Allow all users on the system to query data from Alice's DEPARTMENTS table.**

```
GRANT   select
ON      alice.departments
TO      PUBLIC;
Grant succeeded.
```

How to Revoke Object Privileges

- **You use the `REVOKE` statement to revoke privileges granted to other users.**
- **Privileges granted to others through the `WITH GRANT OPTION` clause are also revoked.**

```
REVOKE {privilege [, privilege...]|ALL}
ON     object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

Revoking Object Privileges

As user Alice, revoke the SELECT and INSERT privileges given to user Scott on the DEPARTMENTS table.

```
REVOKE   select, insert
ON       departments
FROM     scott;
Revoke succeeded.
```

In this lesson, you should have learned how to use control transactions using following commands:

Statement	Description
`COMMIT`	Makes all pending changes permanent
`SAVEPOINT`	Is used to rollback to the savepoint marker
`ROLLBACK`	Discards all pending data changes

Summary (Continued..)

In this lesson, you should have also learned about DCL statements that control access to the database and database objects:

Statement	Action
`CREATE USER`	Creates a user (usually performed by a DBA)
`GRANT`	Gives other users privileges to access the your objects
`CREATE ROLE`	Creates a collection of privileges (usually performed by a DBA)
`ALTER USER`	Changes a user's password
`REVOKE`	Removes privileges on an object from users

Thank You !