Views

- Virtual table based on the result-set of SQL statement and that is Stored in the database with some name.
- A view can contain all rows of a table or select rows from a table.
- A view can be created from one or many tables which depends on the written SQL query to create a view.
- Restrict access to the data such that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.
- If table structure is changed then the view definition also has to be changed.

Views

- **Simple View**
  - A view based on only a single table, which doesn't contain GROUP BY clause and any functions.
- **Complex View**
  - A view based on multiple tables, which contain GROUP BY clause and functions.
- **Inline View**
  - A view based on a subquery in FROM Clause, that subquery creates a temporary table and simplifies the complex query.
- **Materialized View**
  - A view that stores the definition as well as data. It creates replicas of data by storing it physically.

Features	Simple View	Complex View
No Of Tables	One	One or More
Contain function	NO	Yes
Contain group of data	No	Yes

Views - Creation

67 **Create a view for Employee-Age**

```
create view empage(empcode,age)
as (select empcode, timestampdiff(year,birthdate,curdate())
from emp);
```

Views - Creation

68 **Create a view for Employee-Pay**

```
CREATE VIEW EMPPAY (EmpCode, NetPay,
                                SalMonth) AS
    ( SELECT     EmpCode,
                 (Basic + Allow - Deduct), SalMonth
      FROM    Salary );
```

Use of Views

69

List employees and their ages

```
SELECT *
FROM          EMPAGE;
```

Views - Complex

69

Create a view for displaying the number of employees in each department

```
CREATE VIEW DeptEmpCount
      (DeptCode, DeptEmpCount) AS
(SELECT     DeptCode, count(*)
FROM        Emp
GROUP BY    DeptCode);
```

Use of Views

69

Create a view for display the total number of employees of the organization

```
CREATE VIEW EmpCount (EmpCount) AS
(SELECT     count(*) as EmpCount
FROM        Emp);
```

Use of Views

69 **Display the percentage of employees in each department**

```
select deptcode,(deptempcount/empcount)*100
from deptempcount,empcount
order by deptcode;
```

Views - Updation

70 **Update the view for Employee-Pay definition**

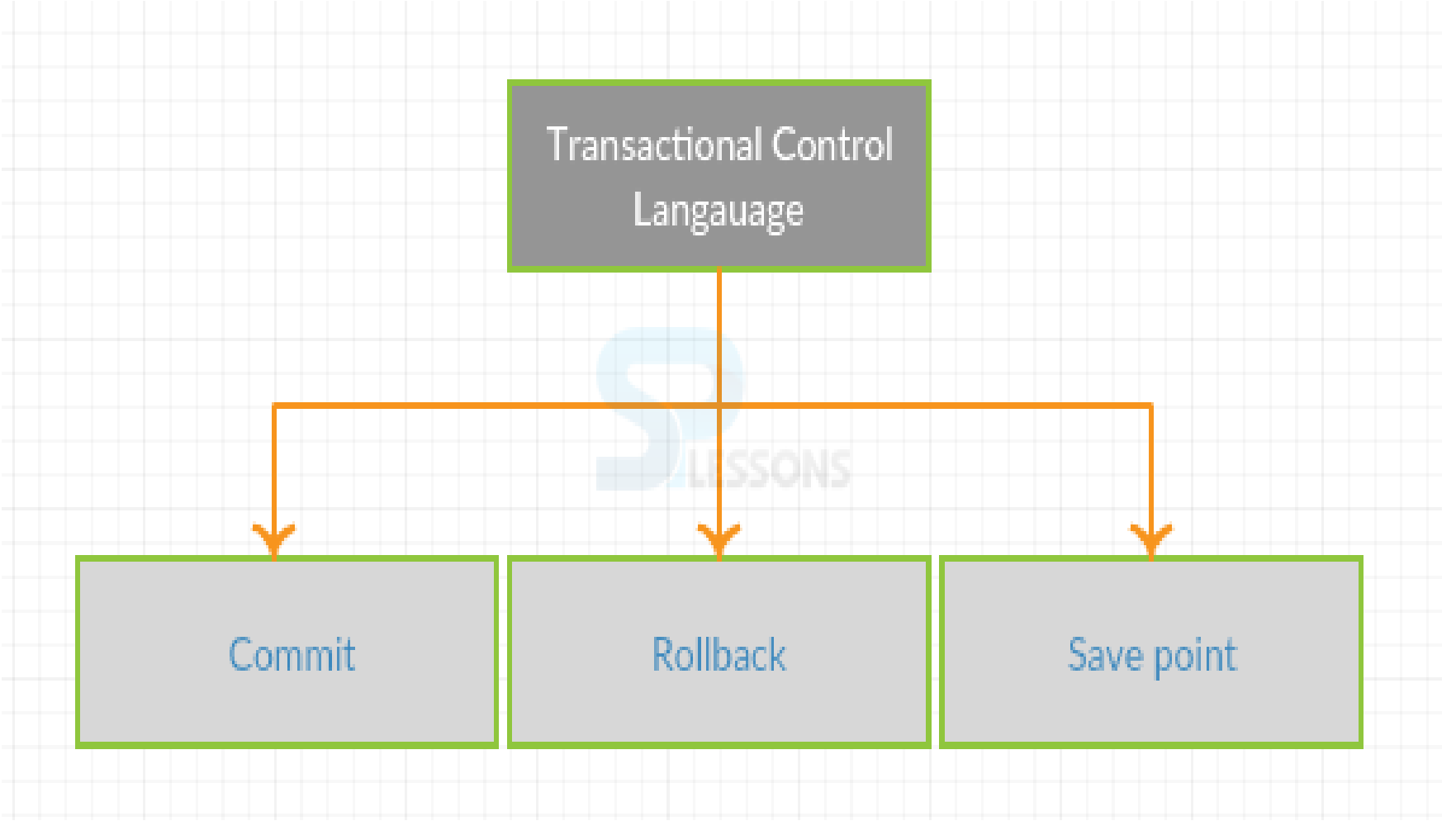
```
alter view emppay(empcode,netpay,salmonth)
as (select empcode, (basic+allow-deduct)+1000,salmonth from salary);
```

Views - Deletion

71 **Delete the view Employee-Pay**

```
DROP VIEW EMPPAY;
```

TCL Commands (Commit/Rollback/Savepoint)



TCL Commands

- **Commit:**
  - Commit command is used to save all the transactions to the database.
- **Rollback:**
  - Rollback command is used to undo transactions that have not already been saved to the database.
- **SAVEPOINT:**
  - It is used to roll the transaction back to a certain point without rolling back the entire transaction.

- Example

DCL Commands (GRANT/REVOKE/GRANT OPTION)

- The GRANT statement enables system administrators to grant privileges and roles, which can be granted to user accounts and roles.
- GRANT cannot mix granting both privileges and roles in the same statement. A given GRANT statement must grant either privileges or roles.