PL/SQL Teaching Resource: Views, Transactions, Indexing, and User Management (Oracle Live SQL)

# Module Overview

This training module is designed for both learners and trainers to master key PL/SQL concepts compatible with Oracle Live SQL:  
- Views (CREATE, UPDATE, Inline Views)  
- Indexes (CREATE, DROP, Clustered vs Non-Clustered simulation)  
- Transactions (COMMIT, ROLLBACK, SAVEPOINT)  
- User Management and Role-based access control simulation (GRANT, REVOKE, ROLES)  
The training is practical and aligned to real-world case study execution using Oracle Live SQL's single-schema simulation.

# Day-Wise Learning Plan

Day 1: Views  
Day 2: Indexes  
Day 3: Transactions  
Day 4: Role-Based Access Control

## Day 1: Views (Trainer & Learner Guide)

Objective: Understand the use of Views for abstraction, reuse, and security.

### Example Tables:

Table: Products

product\_id NUMBER, name VARCHAR2(50), price NUMBER(10,2), stock NUMBER

Table: Orders

order\_id NUMBER, product\_id NUMBER, quantity NUMBER, order\_date DATE

### Step-by-Step Trainer Script

1. Create View:   
CREATE VIEW vw\_product\_summary AS SELECT name, price FROM Products;  
  
2. Update via View:   
UPDATE vw\_product\_summary SET price = price + 10 WHERE name = 'Apple';  
  
3. Inline View:   
SELECT \* FROM (SELECT \* FROM Orders WHERE ROWNUM <= 5);  
  
4. Query View:  
SELECT \* FROM vw\_product\_summary;

### Use Case Scenario:

An inventory dashboard shows only name and price. Updating prices for a promotion is done through the view.

## Day 2: Indexes (Trainer & Learner Guide)

Objective: Understand indexing to improve query performance.

### Trainer Script:

1. Create Index:  
CREATE INDEX idx\_product\_name ON Products(name);  
  
2. Drop Index:  
DROP INDEX idx\_product\_name;  
  
Note: Oracle doesn't support 'CLUSTERED' explicitly. Simulate by indexing heavily-used query columns.

## Day 3: Transactions (Trainer & Learner Guide)

Objective: Understand how COMMIT, ROLLBACK, and SAVEPOINT manage transaction boundaries.

### Trainer Script:

BEGIN  
 UPDATE Products SET stock = stock - 5 WHERE product\_id = 1;  
 SAVEPOINT before\_payment;  
 UPDATE Payments SET payment\_status = 'Failed' WHERE payment\_id = 9001;  
 ROLLBACK TO before\_payment;  
 COMMIT;  
END;

### Real-Life Scenario:

Rollback payment update while retaining product stock update when payment fails.

## Day 4: User & Role Simulation (Single Schema)

Since Oracle Live SQL doesn’t support CREATE USER, simulate users/roles with role\_id.

### Simulated User Table:

user\_id, username, role\_id (1=Admin, 2=Customer, 3=Seller)

### Trainer Script:

-- GRANT Simulation:  
CREATE TABLE Roles (role\_id NUMBER PRIMARY KEY, role\_name VARCHAR2(20));  
CREATE TABLE Users (user\_id NUMBER, username VARCHAR2(50), role\_id NUMBER REFERENCES Roles(role\_id));  
  
-- Role Check in Procedure:  
SELECT role\_name INTO v\_role FROM Users u JOIN Roles r ON u.role\_id = r.role\_id WHERE u.user\_id = :id;  
IF v\_role != 'Admin' THEN RAISE\_APPLICATION\_ERROR(-20001, 'Access Denied'); END IF;

## Deliverables for Learners:

- Create tables and insert sample data  
- Practice each script with small changes  
- Write output interpretation after each action  
- Run each scenario in Oracle Live SQL