# ORACLE DBMS COMPLETE PRACTICAL PACK (E01–E11)

==============================================

This file contains:

✔ All experiments

✔ All Oracle SQL & PL/SQL code

✔ Sample data

✔ Procedures, functions, triggers, cursors

✔ All calling statements

------------------------------------------------------------

## E01 — ER DIAGRAM → TABLES (Library Management)

------------------------------------------------------------

```
CREATE TABLE Book (
BookID NUMBER PRIMARY KEY,
Title VARCHAR2(100),
Author VARCHAR2(60),
Price NUMBER(6,2)
);

CREATE TABLE Member (
MemberID NUMBER PRIMARY KEY,
Name VARCHAR2(60),
Email VARCHAR2(80),
Phone VARCHAR2(15)
);

CREATE TABLE Loan (
LoanID NUMBER PRIMARY KEY,
BookID NUMBER REFERENCES Book(BookID),
MemberID NUMBER REFERENCES Member(MemberID),
LoanDate DATE,
DueDate DATE
);
```

Sample Data:

```
INSERT INTO Book VALUES (1, 'DBMS Concepts', 'Korth', 550);
INSERT INTO Member VALUES (101, 'Aditya Patil', 'adi@mail.com', '9876543210');
INSERT INTO Loan VALUES (1001, 1, 101, SYSDATE, SYSDATE+14);
```

------------------------------------------------------------

E02 — DDL Commands

------------------------------------------------------------

```
CREATE TABLE Employee (
EmpID NUMBER PRIMARY KEY,
EmpName VARCHAR2(50),
Dept VARCHAR2(50),
Salary NUMBER(10,2)
);

ALTER TABLE Employee ADD HireDate DATE;
ALTER TABLE Employee MODIFY EmpName VARCHAR2(80);
ALTER TABLE Employee DROP COLUMN HireDate;

TRUNCATE TABLE Employee;
DROP TABLE Employee;
```

------------------------------------------------------------

E03 — DML Commands

------------------------------------------------------------

```
CREATE TABLE Students (
Roll NUMBER PRIMARY KEY,
Name VARCHAR2(50),
Marks NUMBER(3)
);

INSERT INTO Students VALUES (1,'Amit',85);
INSERT INTO Students VALUES (2,'Riya',92);
INSERT INTO Students VALUES (3,'Neha',78);
```

```
UPDATE Students SET Marks = 95 WHERE Roll = 3;
DELETE FROM Students WHERE Roll = 1;

SELECT * FROM Students;
SELECT Name FROM Students WHERE Marks > 90;
```

------------------------------------------------------------

E04 — Functions (Oracle)

------------------------------------------------------------

```
SELECT ABS(-10), ROUND(123.456,2), CEIL(12.1), FLOOR(12.9) FROM dual;
SELECT UPPER('hello'), LOWER('WORLD'), SUBSTR('DATABASE',1,4) FROM dual;
SELECT SYSDATE, ADD_MONTHS(SYSDATE,1), LAST_DAY(SYSDATE) FROM dual;
SELECT TO_CHAR(SYSDATE,'DD-MON-YYYY') FROM dual;

CREATE TABLE Sales (Prod VARCHAR2(20), Qty NUMBER, Price NUMBER);
INSERT INTO Sales VALUES ('Pen',10,5);
INSERT INTO Sales VALUES ('Pen',5,5);
INSERT INTO Sales VALUES ('Book',3,50);

SELECT Prod, SUM(Qty), AVG(Price), COUNT(*)
FROM Sales GROUP BY Prod;
```

------------------------------------------------------------

E05 — Group By, Having, Order By, Index

------------------------------------------------------------

```
SELECT Dept, COUNT(*), AVG(Salary)
FROM Employee
GROUP BY Dept
HAVING AVG(Salary) > 35000;

SELECT * FROM Employee ORDER BY Salary DESC;

CREATE INDEX idx_emp_dept ON Employee(Dept);
```

------------------------------------------------------------

E06 — Set Operations & Joins

```
-----------------------------------------------------------
CREATE TABLE A (ID NUMBER);
CREATE TABLE B (ID NUMBER);

INSERT INTO A VALUES (1);
INSERT INTO A VALUES (2);
INSERT INTO A VALUES (3);

INSERT INTO B VALUES (2);
INSERT INTO B VALUES (3);
INSERT INTO B VALUES (4);

SELECT * FROM A UNION SELECT * FROM B;
SELECT * FROM A INTERSECT SELECT * FROM B;
SELECT * FROM A MINUS SELECT * FROM B;

CREATE TABLE Dept (
DeptID NUMBER PRIMARY KEY,
DeptName VARCHAR2(30)
);

INSERT INTO Dept VALUES (10,'IT');
INSERT INTO Dept VALUES (20,'HR');

SELECT e.Name, d.DeptName
FROM Employee e
JOIN Dept d ON e.Dept = d.DeptID;

-----------------------------------------------------------
```

## E07 — Subqueries & Views

```
-----------------------------------------------------------
SELECT Name FROM Employee
WHERE Salary = (SELECT MAX(Salary) FROM Employee);

SELECT Name FROM Employee
WHERE Dept IN (SELECT DeptID FROM Dept);
```

```
SELECT e.Name
FROM Employee e
WHERE Salary > (SELECT AVG(Salary) FROM Employee WHERE Dept = e.Dept);

CREATE OR REPLACE VIEW vw_highsal AS
SELECT Name, Salary FROM Employee WHERE Salary > 40000;
```

------------------------------------------------------------

E08 — Transactions

------------------------------------------------------------

```
CREATE TABLE Account (
AccNo NUMBER PRIMARY KEY,
Name VARCHAR2(50),
Balance NUMBER
);

INSERT INTO Account VALUES (101,'Aditya',5000);
INSERT INTO Account VALUES (102,'Riya',8000);

UPDATE Account SET Balance = Balance - 500 WHERE AccNo = 101;
SAVEPOINT sp1;
UPDATE Account SET Balance = Balance + 500 WHERE AccNo = 102;
ROLLBACK TO sp1;
COMMIT;
```

------------------------------------------------------------

E09 — Procedure & Function + CALLS

------------------------------------------------------------

Procedure:

```
CREATE OR REPLACE PROCEDURE give_bonus(p_emp NUMBER, p_bonus NUMBER) IS
BEGIN
UPDATE Employee SET Salary = Salary + p_bonus WHERE EmpID = p_emp;
DBMS_OUTPUT.PUT_LINE('Bonus Added!');
END;
/
```

Function:

```
CREATE OR REPLACE FUNCTION yearly_salary(p_emp NUMBER)
RETURN NUMBER IS
sal NUMBER;
BEGIN
SELECT Salary INTO sal FROM Employee WHERE EmpID = p_emp;
RETURN sal * 12;
END;
/
```

CALLING PROCEDURE:

```
BEGIN
give_bonus(1,2000);
END;
/
```

CALLING FUNCTION:

```
DECLARE
y NUMBER;
BEGIN
y := yearly_salary(1);
DBMS_OUTPUT.PUT_LINE(y);
END;
/
```

------------------------------------------------------------

E10 — Trigger & Cursor + CALLS

------------------------------------------------------------

Trigger:

```
CREATE TABLE Emp_Audit (
EmpID NUMBER,
Action VARCHAR2(20),
ActDate DATE
);

CREATE OR REPLACE TRIGGER trg_insert_audit
AFTER INSERT ON Employee
FOR EACH ROW
BEGIN
INSERT INTO Emp_Audit VALUES (:NEW.EmpID,'INSERT',SYSDATE);
END;
/
```

Test trigger:

```
INSERT INTO Employee VALUES (10,'Karan','IT',45000);
SELECT * FROM Emp_Audit;
```

Cursor:

```
DECLARE
CURSOR cur IS SELECT Name, Salary FROM Employee;
v_name Employee.Name%TYPE;
v_sal Employee.Salary%TYPE;
BEGIN
OPEN cur;
LOOP
FETCH cur INTO v_name, v_sal;
EXIT WHEN cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_name || ' earns ' || v_sal);
END LOOP;
CLOSE cur;
END;
/
```

------------------------------------------------------------

E11 — JDBC (Oracle)

------------------------------------------------------------

```
import java.sql.*;
```

```
public class DBConnect {
public static void main(String args[]){
try{
Connection con = DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle"
);
PreparedStatement ps = con.prepareStatement(
"INSERT INTO Students VALUES(10,'Kiran',88)"
);
ps.executeUpdate();
ResultSet rs = con.createStatement().executeQuery("SELECT * FROM Students");
while(rs.next()){
System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
con.close();
}catch(Exception e){ e.printStackTrace(); }
}
}
```