LAB 9 – DML

TASK 1 – Interloop Database

1. Open Oracle 11g.
2. Create database "Your_Roll#"
3. Create table "Employee"
4. Add the following fields as specified:
   Note: You may add extra relevant fields if needed.

Field Name Data Type	
Employee ID (Primary key)	varchar(7) Nulls not allowed e.g. f228079
First Name	varchar Nulls not allowed
Last Name	varchar Nulls not allowed
Branch	nchar(5) Nulls not allowed
Age	INT/Numeric(2,0) Nulls not allowed
NIC	nchar(15) Nulls not allowed
Scale	INT/ Numeric(1,0) Nulls not allowed
Salary	Int Nulls not allowed
Joining Date	Date Nulls not allowed
Overtime (In Hours)	Float/decimal Nulls are allowed

5. Add 4 records in Employee table appropriately. **Note:** Insert two tuples with column list and other two without column list method.

6. You must use appropriate column alias for all the queries in the assignment.

7. Write and execute queries to generate the following outputs one by one:

1) Write an SQL query that will return the **age of all employees** and name age column as updated_age.

2) Output max salary grouped by branch.

3) Output branches whose average salary is greater than 30k.

4) Output sum of overtime in each branch.

5) Output max of overtime in each branch.

6) Output min of overtime in each branch.

7) Output count of employee who have overtime hours greater than 5.6.

8) Output all employee who started working after "1st January 2015".

9) Output all employee whose name starts with (A or M or E).

10) Output all employee whose name contain a vowel.

11) Output all employee whose names are not in the following list {'Ahmed', 'Tahir', 'Hiba','Amna'}.

12) Output standard deviation of overtime hour in each branch.

13) Output variance of salary in each branch.

14) Output the number of employee in each branch.

15) Output the number of employee's having warnings in each branch.

16) Output all employee's whose employee ID starts from '16f'.

17) Output all employee whose overtime hour is greater than the average (Overtime hour).

18) Write an SQL query that will return all records of employees without duplication.

19) Write an SQL query that will sort with respect to age (descending) and return all records of employee.

20) Update the employee table with age increase of five years.

21) Display Distinct values based on First_Name.