Charotar University of Science and Technology [CHARUSAT]
Faculty of Technology and Engineering
Devang Patel Institute of Advance Technology & Research (DEPSTAR)
Department Of Information Technology
IT260 Database Management System

Practical List [A.Y. 2024-25]

Practical-1

Aim: Design and implement the schema as per the given information.

- **Create a schema for the bank that wants to keep the records of different Job profiles along with associated employee and their related information. Bank also wants to keep the records of their account and loan-related customer separately in their database.**

Constraints –

- Not Null Constraints: Critical fields must not be null to ensure data completeness.
- Unique Constraints: Certain fields must have unique values to avoid duplicates (e.g., Account Number).
- Check Constraints: Enforce domain integrity by limiting the values that can be placed in a column.

Tasks:-

1) Create Table Job (job_id, job_title, min_sal, max_sal)

COLUMN NAME	DATA TYPE
job_id	Varchar2(15)
job_title	Varchar2(30)
min_sal	Number(7,2)
max_sal	Number(7,2)

2) Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no)

COLUMN NAME	DATA TYPE
emp_no	Number(3)
emp_name	Varchar2(30)
emp_sal	Number(8,2)
emp_comm	Number(6,1)
dept_no	Number(3)

3) Create table deposit (a_no, name,bname, amount,a_date).

COLUMN NAME	DATA TYPE
a_no	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Number(7,2)
a_date	Date

4) Create table borrow (loan no,cname,bname,amount).

COLUMN NAME	DATA TYPE
loanno	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Varchar2(7,2)

5) Insert the following values in the table Employee.

emp_no	emp_name	emp_sal	emp_comm	dept _no
101	Smith	800	455	20
102	Snehal	1600	0	25
103	Adama	1100	425	20
104	Aman	3000		15
105	Anita	5000	50,000	10
106	Anamika	2975		30

6) Insert the following values in the table JOB.

job_id	job_title	min_sal	max_sal
IT_PROG	Programmer	4000	10000
MK_MGR	Marketing manager	9000	15000
FI_MGR	Finance manager	8200	12000
FI_ACC	Account	4200	9000
LEC	Lecturer	6000	17000
COMP_OP	Computer Operator	1500	13000

7) Insert the following values in the table deposit.

a_no	cname	bname	amount	a_date
101	Anil	andheri	7000	01-jan-06
102	sunil	virar	5000	15-jul-06
103	jay	villeparle	6500	12-mar-06
104	vijay	andheri	8000	17-sep-06
105	keyur	dadar	7500	19-nov-06
106	mayur	borivali	5500	21-dec-06

8) Insert the following values in the table borrow.

loanno	cname	bname	amount
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

9) Describe the table Job, employee, deposit, borrow.

Practical-2

Aim: Perform Data Definition Language (DDL) commands and change the existing schema as per the given information.

Constraints –

- Not Null Constraints: Ensure critical fields are not null.
- Unique Constraints: Ensure data integrity by limiting column values.
- Check Constraints: Ensure columns like Account Number have unique values.

Tasks:-

1) Create a table supplier from an employee with all the columns and verify.

Test case: Verify suppler table consists of all columns of employee table.

2) Create table sup1 from an employee with the first two columns and verify.

Test case: Verify sup1 consists of first two columns of employee table.

3) Create table sup2 from employee with no data and verify.

Test case: Verify sup2 table contains no data.

4) Insert the data into sup2 from employee whose name is 'Anita' and verify.

Test case: Verify the details of employee whose name is 'Anita' is inserted in sup2 table.

5) Rename the table sup2 and verify.

Test case: Verify the sup2 table name is changed.

6) Destroy table sup1 with all the data and verify.

Test case: Verify table sup1 is destroyed or not.

7) Add one column phone to an employee with size of column is Varchar2(10) and verify.

Test case: Verify phone column is added in employee as per the column size.

8) Modify column phone and change type to char(10) and verify.

Test case: Verify phone column type is changed to char(10).

9) Delete employee_name column from sup2 and verify;

Test case: Verify employee_name column is deleted from sup2.

10) Rename the column salary to new_sal in sup2 and verify.

Test case: Verify the column name of salary is changed to new_sal in sup2 table.

Practical-3

im: Perform Data Definition Language (DDL) commands and change the existing schema per given information.

Constraints –

- Not Null Constraints: Ensure critical fields are not null.
- Unique Constraints: Ensure data integrity by limiting column values.
- Check Constraints: Ensure columns have unique values where required.

Test Cases-

1) Retrieve all data from **employee, jobs and deposit.**

Test Case: Verify all rows are retrieved from each table.

2) Display job title and maximum salary of all jobs.

Test Case: Verify the job title and maximum salary for each job.

3) Write a query to find out all the branches.

Test Case: Verify the list of all unique branches.

4) Display all the account no. into which rupees are between dates **01-01-06 and 25-07-06**.

Test Case: Verify the account numbers with deposits in the specified date range.

5) Display names of all customers whose account is deposited after **09-oct-06**

Test Case: Verify the names of customers with accounts deposited after the specified date.

6) Display name and salary of employee whose department no is 20. Give alias name to name of employee.

Test Case: Verify the names and salaries of employees in department 20 with the alias.

7) Display employee no, name and department details of those employee whose department lies **in(10,20).**

   Test Case: Verify the details of employees in departments 10 and 20.

8) Display employee no, name and department details of those employee whose department **not in(15,30) except 25 .**

   Test Case: Verify the details of employees excluding departments 15, 30, and 25.

9) Display employee no, name and department details of those employee whose department no is **between 15 and 25.**

   Test Case: Verify the details of employees in departments between 15 and 25.

10) Display name of all employee whose emp_comm contains the **non-null** values.

    Test Case: Verify the names of employees with non-null emp_comm values.

11) Combine two columns min_sal and max_sal and display it one column using common alias name.
    Test Case: Verify the combined minimum and maximum salary displayed as a single column.
12) Insert the data into sup2 from employee.
    Test Case: Verify the data from Employee is inserted into Supplier2.

13) Delete all the rows from sup1 as sup.

    Test Case: Verify all rows are deleted from Sup1.

14) Delete the detail of supplier whose emp_no is 103.

    Test Case: Verify the row with emp_no 103 is deleted from Supplier.

15) Update the name of employee to **'Aman'** name whose emp_name is 'Anita'.

    Test Case: Verify the name 'Anita' is updated to 'Aman'.

16) Update the value of employee name whose employee number is 103.
Test Case: Verify the name is updated for the employee with EmployeeID 103.

17) Find out the maximum and minimum salary form job table.
Test Case: Verify the maximum and minimum salary from the JobProfile table.

18) Find out the average salary of employee.
Test Case: Verify the average salary of employees.

19) Count the total no as well as distinct rows in dept_no column with a condition of salary greater than 1000 of employee.
Test Case: Verify the total and distinct count of departments with a salary greater than 1000.

20) Display the detail of all employees in ascending order, descending order of their name and no.
Test Case: Verify the employee details sorted by name in ascending order and by number in descending order.

21) Display the dept_no in ascending order and accordingly display emp_comm in descending order.
Test Case: Verify the dept_no in ascending order and emp_comm in descending order.

22) Update the value of emp_comm to 500 where dept_no is 20.
Test Case: Verify the emp_comm is updated to 500 for department 20.

23) Display the emp_comm in ascending order with null value first and accordingly sort employee salary in descending order.
Test Case: Verify emp_comm in ascending order with null values first and salary in descending order.

24) Display the emp_comm in ascending order with null value last and accordingly sort emp_no in descending order.
Test Case: Verify emp_comm in ascending order with null values last and emp_no in descending order.

Practical-4

Aim: - To execute value-matching and pattern-matching conditions on the bank's schema to retrieve specific data based on given requirements.

Constraints –

- Not Null Constraints: Ensure critical fields are not null.
- Unique Constraints: Ensure data integrity by limiting column values.
- Check Constraints: Ensure columns have unique values where required.

Tasks:–

1) Display all employees whose name start with ‘A’ and the third character is ‘a’.

   Test case: Verify the details of Aman and Adama is displayed.

2) Display the name, number and salary of those employees whose name is 5 characters long and the first three characters are ‘Ani’.

   Test case: Verify the details of Anita is displayed.

3) Display all information of employees whose second character of name is either ‘M’ or ‘N’.

   Test case: Verify the details of Aman, Anamika and Anita is displayed.

4) Find the list of all customer names whose branch is in ‘Andheri’ or ‘Dadar’ or ‘Virar’.

   Test case: Verify customer names such as Anil, Sunil and keyur is displayed.

5) Display the job name whose first three characters in the job ID field are ‘FI_’.

   Test case: Verify FI_MGR and FI_ACC is displayed.

6) Display the title/name of the job whose last three characters are ‘_MGR’ and whose maximum salary is greater than Rs 12000.

   Test case: Verify Marketing manager is displayed.

7) Display the non-null values of employees also employee name's second character should be ‘n’ and the string should be 5 characters long.

Test case: Verify ‘Anita’ details is displayed.

8) Display the null values of the employee and also employee name’s third character should be ‘a’.

   Test case: Verify ‘Aman’ details is displayed.

9) What will be output if you are giving the LIKE predicate as ‘%_%’ ESCAPE ‘\’

   Test Case**:** Verify that the retrieved job IDs contain an underscore character.

Practical-5

Aim: To study data communication using SQL functions, focusing on aggregate, numeric, string, conversion, date, and set operations functions, to manipulate and retrieve data effectively from the bank's database.

Constraints –

- Not Null Constraints: Ensure critical fields are not null.
- Unique Constraints: Ensure data integrity by limiting column values.
- Check Constraints: Ensure columns have unique values where required.

Test Cases –

Aggregate Functions:

- AVG (DISTINCT | ALL | n)
  Test Case: Verify the average salary from the JobProfile table and the distinct average salary.
- MIN (DISTINCT | ALL | expr)
  Test Case: Verify the minimum salary from the JobProfile table.
- COUNT (DISTINCT | ALL | expr)
  Test Case: Verify the total count of employees and the count of distinct departments.
- COUNT (*)
  Test Case: Verify the total number of employees.
- MAX (DISTINCT | ALL | expr)
  Test Case: Verify the maximum salary from the JobProfile table.
- SUM (DISTINCT | ALL | n)
  Test Case: Verify the total sum of salaries and the distinct total salary.

Numeric Functions:

- ABS(n)
  Test Case: Verify the absolute difference between salary and 1000.
- POWER(m, n)
  Test Case: Verify the square of the salary.
- ROUND(n, m)
  Test Case: Verify the salary rounded to 2 decimal places.
- SQRT(n)
  Test Case: Verify the square root of the salary.

String Functions:

- LOWER(char)
  Test Case: Verify the employee first names in lowercase.
- INITCAP(char)
  Test Case: Verify the employee first names with the initial capital letter.
- UPPER(char)
  Test Case: Verify the employee first names in uppercase.
- SUBSTR(char, m [, n])
  Test Case: Verify the first three characters of the employee first names.
- LENGTH(word)
  Test Case: Verify the length of the employee first names.
- LTRIM(char [, set])
  Test Case: Verify the employee first names with leading 'A' removed.
- RTRIM(char [, set])
  Test Case: Verify the employee first names with trailing 'a' removed.
- LPAD(char1, n [, char2])
  Test Case: Verify the employee first names padded to the left with '*' up to a total length of 10.
- RPAD(char1, n [, char2])
  Test Case: Verify the employee first names padded to the right with '*' up to a total length of 10.

Conversion Functions:

- TO_NUMBER(char)
  Test Case: Verify the conversion of a string to a number.
- TO_CHAR(n [, fmt])
  Test Case: Verify the conversion of a number to a formatted string.

Date Conversion Functions:

- TO_DATE (char [, fmt])

Test Case: Verify the conversion of a string to a date.

DATE Functions:

- ADD_MONTHS(d, n)
  Test Case: Verify the date after adding 6 months to the current date.
- LAST_DAY(d)
  Test Case: Verify the last day of the current month.
- MONTHS_BETWEEN(d1, d2)
  Test Case: Verify the number of months between two dates.
- NEXT_DAY(date, char)
  Test Case: Verify the next Monday from the current date.

Set Operations:

- UNION
  Test Case: Verify the union of first names from employees and customers.
- UNION ALL
  Test Case: Verify the union all of first names from employees and customers, including duplicates.
- INTERSECTION
  Test Case: Verify the intersection of first names from employees and customers.
- MINUS
  Test Case: Verify the first names from employees that are not in customers.

Practical-6

Aim: To solve various queries related to grouping and aggregate functions by manipulating data in the product and emp_company tables.

Constraints –

- Not Null Constraints: Ensure critical fields are not null.
- Unique Constraints: Ensure data integrity by limiting column values.
- Check Constraints: Ensure columns have unique values where required.

Test Cases –

1) Insert the following values into the product table.

Detorder _no	Product_no	Qty_order
O19001	P00001	10
O19001	P00002	3
O19002	P00001	4
O19003	P00004	2
O19004	P00003	6
O19005	P00005	2
O19006	P00004	7

Test Case: Verify that the values are inserted correctly into the product table.

2) Retrieve the product numbers and total quantity ordered for each product from the product table.

Test Case: Verify that the sum of quantities ordered for each product number is calculated correctly.

3) Retrieve the product no and the total quantity ordered for product's 'P00001' and 'P00004' from product table.

Test Case: Verify that the sum of quantities ordered for product numbers 'P00001' and 'P00004' is calculated correctly.

4) Insert the following values into emp_company

ENAME	CNAME	SALARY
Anil	ACC	1500
Shankar	TATA	2000
Jay	WIPRO	1800
Sunil	WIPRO	1700
Vijay	TATA	5000
Prakash	TATA	3000
Ajay	ACC	8000

Abhay	ACC	1800

Test Case: Verify that the values are inserted correctly into the emp_company table.

5) List the name of the company and the maximum salary in that company.

   Test Case: Verify that the maximum salary for each company is calculated correctly.

6) Find out the average salary of each company.

   Test Case: Verify that the average salary for each company is calculated correctly.

7) Find out the names of companies having an average salary of more than 1500.

   Test Case: Verify that the companies with an average salary greater than 1500 are listed correctly.

8) Find out the average salary of each company except 'ACC'.

   Test Case: Verify that the average salary for each company, excluding 'ACC', is calculated correctly.

Practical-7

For a given University schema, create tables and generate a Master-Slave relationship along with all the necessary integrity constraints in MS Access Tool.

Practical-8

AIM - To apply the concept of integrity/data constraints while creating or altering tables in a database for managing sales-related data.

1) Create Table Salespeople where Snum number(4) P.K, Sname varchar2(20) NOT NULL, City varchar2(15), Comm number(5,2).

Constraints

1. **Primary Key Constraint**: Ensure Snum uniquely identifies each record.
2. **Not Null Constraint**: Ensure Sname is not null.

Tasks:

Test Case 1: Insert a valid record.

Objective: Verify that a valid record can be inserted into the Salespeople table.

Expected Result: The record should be inserted successfully.

Test Case 2: Insert a record with a duplicate Snum

Objective: Verify that inserting a record with a duplicate Snum results in an error.

Expected Result: The insertion should fail with a primary key constraint violation error.

Test Case 3: Insert a record with a null Sname

Objective: Verify that inserting a record with a null Sname results in an error.

Expected Result: The insertion should fail with a not-null constraint violation error.

Test Case 4: Insert a record with missing City and Comm

Objective: Verify that a record can be inserted with City and Comm as null.

Expected Result: The record should be inserted successfully with City and Comm as null.

Test Case 5: Retrieve all records from the Salespeople table.

Objective: Verify that all records can be retrieved from the table.

2) Create Table customer where Cnum number (4) P.K, Cname varchar2(20)NOT NULL, City varchar2(20), Rating number(3) DEFAULT 10, Snum number(4) F.K.(where snum refer salespeople table).

Constraints

1. **Primary Key Constraint**: Ensure Cnum uniquely identifies each record.
2. **Not Null Constraint**: Ensure Cname is not null.
3. **Foreign Key Constraint**: Ensure Snum references the Snum field in the Salespeople table.
4. **Default Constraint**: Ensure Rating defaults to 10 if no value is provided.

Tasks:

Test Case 1: Insert a valid record

Objective: Verify that a valid record can be inserted into the Customer table.

Expected Result: The record should be inserted successfully.

Test Case 2: Insert a record with a missing Rating

Objective: Verify that a record can be inserted with a Rating defaulting to 10.

Expected Result: The record should be inserted successfully with a Rating of 10.

Test Case 3: Insert a record with a null Cname

Objective: Verify that inserting a record with a null Cname results in an error.

Expected Result: The insertion should fail with a not null constraint violation error.

Test Case 4: Insert a record with a non-existent Snum

Objective: Verify that inserting a record with a non-existent Snum results in an error.

Expected Result: The insertion should fail with a foreign key constraint violation error.

Test Case 5: Insert a record with missing City and Rating

Objective: Verify that a record can be inserted with City and Rating as null/default.

Expected Result: The record should be inserted successfully with City as null and Rating as 10.

Test Case 6: Retrieve all records from the Customer table

Objective: Verify that all records can be retrieved from the table.

Expected Result: The query should return all the inserted records.

3) Create table Order where Order_no number(4) P.K, Amount number(5), Odate varchar2(10), Cnum number(4) F.K, (where cnum refer customer table). Snum number(4) F.K (where snum refers salespeople table).

Constraints

1. **Primary Key Constraint**: Ensure Order_no uniquely identifies each record.
2. **Foreign Key Constraint**: Ensure Cnum references the Cnum field in the Customer table.
3. **Foreign Key Constraint**: Ensure Snum references the Snum field in the Salespeople table.

Tasks:

Test Case 1: Insert a valid record

Objective: Verify that a valid record can be inserted into the Orders table.

Expected Result: The record should be inserted successfully.

Test Case 2: Insert a record with a non-existent Cnum

Objective: Verify that inserting a record with a non-existent Cnum results in an error.

Expected Result: The insertion should fail with a foreign key constraint violation error for Cnum.

Test Case 3: Insert a record with a non-existent Snum

Objective: Verify that inserting a record with a non-existent Snum results in an error.

Expected Result: The insertion should fail with a foreign key constraint violation error for Snum.

Test Case 4: Insert a record with a null Amount

Objective: Verify that a record can be inserted with Amount as null.

Expected Result: The record should be inserted successfully with Amount as null.

Test Case 5: Insert a record with null Odate

Objective: Verify that a record can be inserted with Odate as null.

Expected Result: The record should be inserted successfully with Odate as null.

Test Case 6: Retrieve all records from the Orders table

Objective: Verify that all records can be retrieved from the table.

Expected Result: The query should return all the inserted records.

4. Table: Sales_order

Constraints

- **Primary Key Constraints**: Ensure unique identification of records.
- **Foreign Key Constraints**: Maintain relationships between tables.
- **Not Null Constraints**: Ensure critical fields are not null.
- **Check Constraints**: Ensure data integrity by limiting column values.
- **Unique Constraints**: Ensure columns have unique values where required.
- **Default Constraints**: Assign default values to columns when no value is provided.

Column Name	Data Type	Constraints
Order_no	Varchar2(6)	Primary key/First letter starting with 'O'
Order_date	Date	
Client_no	Varchar2(6)	Foreign Key references client_no of client_master table

Dely_addr	Varchar2(25)	
Salesman_no	Varchar2(6)	Foreign Key references sales_no of Salesman_master table
Dely_type	char	Delivery: part(P)/ full (f) Default 'F'
Order_status	Varchar2	In Process, Fulfilled, Backorder, Cancelled

Test Case: Verify that all constraints, including the primary key, foreign key, check constraints, and default values, are correctly applied.

5. Table: salesman_master

Constraints

- **Primary Key Constraints**: Ensure unique identification of records.
- **Foreign Key Constraints**: Maintain relationships between tables.
- **Not Null Constraints**: Ensure critical fields are not null.
- **Check Constraints**: Ensure data integrity by limiting column values.
- **Unique Constraints**: Ensure columns have unique values where required.
- **Default Constraints**: Assign default values to columns when no value is provided.

Column name	Data type	Constraints
Salesman_no	Varchar2(6)	The primary key/first letter must start with 'S'
Salesman _name	Varchar2(20)	Not null
Address	Varchar2(30)	Not null
City	Varchar2(20)	
Pincode	Varchar2(8)	
State	Varchar2(20)	
Sal_amt	Number(8,2)	Not null, cannot be 0
Ytd_sales	Number(6,2)	Not null, cannot be 0
Tgt_sales	Number(6,2)	Not null

Test Case: Verify that all constraints, including the primary key, not null, and check constraints, are correctly applied.

6. Table: Client_master

Constraints

- **Primary Key Constraints**: Ensure unique identification of records.
- **Foreign Key Constraints**: Maintain relationships between tables.
- **Not Null Constraints**: Ensure critical fields are not null.
- **Check Constraints**: Ensure data integrity by limiting column values.
- **Unique Constraints**: Ensure columns have unique values where required.
- **Default Constraints**: Assign default values to columns when no value is provided.

Column name	Data type	Constraints
Client_no	Varchar2(6)	The primary key/first letter must start with 'C'
Name	Varchar2(20)	Not null
Address	Varchar2(30)	
City	Varchar2(15)	
Pincode	Number(8)	
State	Varchar2(15)	
Bal_due	Number(10,2)	

Test Case: Verify that all constraints, including the primary key, not null, and check constraints, are correctly applied.

Practical-9

Aim: To study and execute various JOIN commands to perform data retrieval and manipulation from Salespeople, Customer, and Order tables based on specific requirements.

Constraints

1. **Primary Key Constraints**: Ensure unique identification of records in each table.
2. **Foreign Key Constraints**: Maintain referential integrity between related tables.

3. **Not Null Constraints**: Ensure essential fields are not left empty.

Salespeople:

snum	sname	city	comm.
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelord	New York	0.10

Customer:

cnum	cname	city	rating	snum
2001	Hoffman	London	100	1001
2002	Giovanne	Rome	200	1003
2003	Liu	San Jose	300	1002
2004	Grass	Berlin	100	1002
2006	Clemens	London	300	1007
2007	Pereira	Rome	100	1004

Order:

onum	amt	odate	cnum	snum

3001	18.96	10.3.94	2002	1002
3003	767.19	10.3.94	2001	1001
3002	1900.10	10.3.94	2007	1003
3005	5160.45	10.3.94	2003	1002
3006	1098.16	10.3.94	**2008**	1002
3009	1713.23	10.4.94	2002	1003
3007	75.75	10.4.94	2004	1002
3008	4723.95	10.5.94	2006	1001
3010	1309.95	10.6.94	2004	1002
3011	9891.00	10.6.94	2006	1001

Test Cases -

1. All customers serviced by Peel or Motika

Expected Result: List of customers serviced by salespeople named Peel or Motika.

2. All orders except those with 0 or null value in the amt field

Expected Result: List of all orders where amt is not 0 or null.

3. Largest order taken by each sales order value of more than 3000

Expected Result: Largest order amount for each salesperson where the order value is more than 3000.

4. All combinations of salespeople and customers who belong to the same city

Expected Result: List of salespeople and customers who are from the same city.

5. List each order with the name of the customer who placed the order

Expected Result: List of orders with the corresponding customer names who placed the orders.

6. Produce a listing of all the customers serviced by salespeople having a commission of more than 12%

Expected Result: List of customers serviced by salespeople with a commission greater than 12%.

7. Produce names and cities of all customers with the same rating as Hoffman

Expected Result: List of customers who have the same rating as Hoffman.

8. Count the customers with ratings above San Jose's average

Expected Result: Number of customers with a rating above the average rating of customers from San Jose.

9. Find the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the order table

Expected Result: Total amount of orders for each salesperson where the total is greater than the largest single order amount in the orders table.

10. Create a union of two queries that shows all customers' names, cities, and ratings. Those with a rating of 200 or greater will have the words 'High Rating' while others will have 'Low Rating'.

Expected Result: List of customers with their names, cities, ratings, and a rating category indicating 'High Rating' or 'Low Rating'.

Practical-10

Aim: To create a PL/SQL procedure that performs the multiplication of two numbers. The procedure will take two input parameters and return the product.

Constraints

1. **Input Parameters**: The procedure will accept two input parameters of type NUMBER.
2. **Output Parameter**: The procedure will have an output parameter to return the result of the multiplication.

3. **Error Handling**: The procedure will handle potential errors such as invalid input types.

Tasks -

Test Case 1: Multiplying two positive numbers

Input: num1 = 10, num2 = 5

Expected Output: The result should be 50.

Test Case 2: Multiplying a positive number and zero

Input: num1 = 15, num2 = 0

Expected Output: The result should be 0.

Test Case 3: Multiplying two negative numbers

Input: num1 = -4, num2 = -6

Expected Output: The result should be 24.

Test Case 4: Multiplying a positive number and a negative number

Input: num1 = 7, num2 = -3

Expected Output: The result should be -21.

Test Case 5: Multiplying two floating-point numbers

Input: num1 = 2.5, num2 = 4.2

Expected Output: The result should be 10.5.

Practical-11

Aim: To create a PL/SQL block that deletes records from a table with age 21. Additionally, it generates a trigger that stores the original record in another table before deletion.

Constraints

1. **Deletion Condition**: Delete records from the main table where the age is 21.
2. **Trigger**: Before deleting a record, store the original record in another table.

Tasks -

Test Case 1: Delete records where the age is 21

Objective: Verify that records with an age of 21 are successfully deleted.

Expected Result: Records with an age of 21 ('Alice' and 'Charlie') should be deleted.

Test Case 2: Verify Trigger Functionality

Objective: Verify that the trigger successfully stores original records in the DeletedRecords table before deletion.

Expected Result: The DeletedRecords table should contain the original records of 'Alice' and 'Charlie' before they were deleted.