Assignment No: 4

**Date:24/11/2024**

create table employee4 (emp\_id int primary key, emp\_name varchar(50),

address varchar(100));

CREATE TABLE

create table investment4 (inv\_no int primary key, inv\_name varchar(50),inv\_date date, inv\_amount decimal(10,2) check (inv\_amount > 0), emp\_id int, foreign key (emp\_id) references employee4(emp\_id));

CREATE TABLE

select \* from employee4;

emp\_id | emp\_name | address

--------+-------------+------------

1 | MR.PATIL | PUNE

2 | MS.SHARMA | MUMBAI

3 | MR.DESHMUKH | NASHIK

4 | MS.JOSHI | NAGPUR

5 | MR.KHAN | AURANGABAD

(5 rows)

select \* from investment4;

inv\_no | inv\_name | inv\_date | inv\_amount | emp\_id

--------+-------------+------------+------------+--------

101 | MUTUAL FUND | 2024-01-15 | 50000.00 | 1

102 | STOCK | 2024-02-10 | 100000.00 | 1

103 | MUTUAL FUND | 2024-03-05 | 75000.00 | 2

104 | BOND | 2024-04-20 | 25000.00 | 2

105 | MUTUAL FUND | 2024-05-01 | 65000.00 | 3

106 | MUTUAL FUND | 2024-06-15 | 80000.00 | 1

107 | STOCK | 2024-07-01 | 120000.00 | 5

(7 rows)

Function -1

CREATE OR REPLACE FUNCTION get\_mutual\_fund\_invest()

RETURNS TABLE (emp\_id INT, emp\_name VARCHAR(50), inv\_no INT, inv\_name

VARCHAR(50), inv\_date DATE, inv\_amount DECIMAL)

AS

$$

BEGIN

RETURN QUERY

SELECT e.emp\_id, e.emp\_name, i.inv\_no, i.inv\_name, i.inv\_date,i.inv\_amount

FROM employee4 e JOIN investment4 i ON e.emp\_id = i.emp\_id

WHERE i.inv\_name = 'MUTUAL FUND';

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_mutual\_fund\_invest();

emp\_id | emp\_name | inv\_no | inv\_name | inv\_date | inv\_amount

--------+-------------+--------+-------------+------------+------------

1 | MR.PATIL | 106 | MUTUAL FUND | 2024-06-15 | 80000.00

1 | MR.PATIL | 101 | MUTUAL FUND | 2024-01-15 | 50000.00

2 | MS.SHARMA | 103 | MUTUAL FUND | 2024-03-05 | 75000.00

3 | MR.DESHMUKH | 105 | MUTUAL FUND | 2024-05-01 | 65000.00

(4 rows)

Function -2

CREATE OR REPLACE FUNCTION get\_mr\_patil\_investments()

RETURNS TABLE (inv\_no INT, inv\_name VARCHAR(50), inv\_date DATE,

inv\_amount DECIMAL)

AS

$$

BEGIN

RETURN QUERY

SELECT i.inv\_no, i.inv\_name, i.inv\_date, i.inv\_amount

FROM investment4 i

JOIN employee4 e ON i.emp\_id = e.emp\_id

WHERE e.emp\_name = 'MR.PATIL';

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_mr\_patil\_investments();

inv\_no | inv\_name | inv\_date | inv\_amount

--------+-------------+------------+------------

101 | MUTUAL FUND | 2024-01-15 | 50000.00

102 | STOCK | 2024-02-10 | 100000.00

106 | MUTUAL FUND | 2024-06-15 | 80000.00

(3 rows)

Function -3

CREATE OR REPLACE FUNCTION get\_date\_wise\_investments()

RETURNS TABLE (inv\_date DATE, emp\_name VARCHAR(50), inv\_no INT,

inv\_name VARCHAR(50), inv\_amount DECIMAL)

AS

$$

BEGIN

RETURN QUERY

SELECT i.inv\_date, e.emp\_name, i.inv\_no, i.inv\_name, i.inv\_amount FROM investment4 i

JOIN employee4 e ON i.emp\_id = e.emp\_id

ORDER BY i.inv\_date;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_date\_wise\_investments();

inv\_date | emp\_name | inv\_no | inv\_name | inv\_amount

------------+-------------+--------+-------------+------------

2024-01-15 | MR.PATIL | 101 | MUTUAL FUND | 50000.00

2024-02-10 | MR.PATIL | 102 | STOCK | 100000.00

2024-03-05 | MS.SHARMA | 103 | MUTUAL FUND | 75000.00

2024-04-20 | MS.SHARMA | 104 | BOND | 25000.00

2024-05-01 | MR.DESHMUKH | 105 | MUTUAL FUND | 65000.00

2024-06-15 | MR.PATIL | 106 | MUTUAL FUND | 80000.00

2024-07-01 | MR.KHAN | 107 | STOCK | 120000.00

(7 rows)

Q2)

create table train (train\_no int primary key, train\_name varchar(20), depart\_time

time, arrival\_time time, source\_stn varchar(20), dest\_stn varchar(20),no\_of\_res\_bogies

int, bogie\_capacity int);

CREATE TABLE

create table passenger (passenger\_id int primary key, passenger\_name

varchar(20), address varchar(30), age int, gender char(1) check (gender in ('m', 'f'))

female);

CREATE TABLE

create table ticket (train\_no int, passenger\_id int, ticket\_no int primary key,

bogie\_no int, no\_of\_berths int, tdate date, ticket\_amt decimal(7,2), status char(1) check

(status in ('w', 'c')), foreign key (train\_no) references train(train\_no), foreign key

(passenger\_id) references passenger(passenger\_id));

CREATE TABLE

select \* from train;

train\_no | train\_name | depart\_time | arrival\_time | source\_stn | dest\_stn |

no\_of\_res\_bogies | bogie\_capacity

----------+---------------+-------------+--------------+------------+----------+------------------+-------

---------

101 | EXPRESS TRAIN | 08:00:00 | 10:00:00 | CITY\_A | CITY\_B | 5 |

50

102 | LOCAL TRAIN | 09:30:00 | 11:30:00 | CITY\_C | CITY\_D | 3 | 30

(2 rows)

select \* from passenger;

passenger\_id | passenger\_name | address | age | gender

--------------+----------------+-------------+-----+--------

1 | JOHN DOE | 123 MAIN ST | 28 | m

2 | JANE SMITH | 456 ELM ST | 22 | f

3 | MR.PATIL | 789 OAK ST | 30 | m

(3 rows)

select \* from ticket;

train\_no | passenger\_id | ticket\_no | bogie\_no | no\_of\_berths | tdate | ticket\_amt |

status

----------+--------------+-----------+----------+--------------+------------+------------+--------

101 | 1 | 1001 | 1 | 1 | 2024-10-15 | 150.00 | c

101 | 2 | 1002 | 2 | 2 | 2024-10-15 | 150.00 | c

102 | 3 | 1003 | 1 | 1 | 2024-10-16 | 100.00 | w

102 | 1 | 1004 | 2 | 2 | 2024-10-16 | 200.00 | c

(4 rows)

Function -1

CREATE OR REPLACE FUNCTION get\_confirmed\_bookings\_on\_date(input\_date

DATE)

RETURNS TABLE (train\_name VARCHAR(20), passenger\_name VARCHAR(20),

bogie\_no INT, no\_of\_berths INT, ticket\_amt DECIMAL)

AS

$$

BEGIN

IF input\_date IS NULL THEN

RAISE EXCEPTION 'Invalid date provided';

END IF;

RETURN QUERY

SELECT t.train\_name, p.passenger\_name, tk.bogie\_no, tk.no\_of\_berths,tk.ticket\_amt

FROM ticket tk

JOIN train t ON tk.train\_no = t.train\_no

JOIN passenger p ON tk.passenger\_id = p.passenger\_id

WHERE tk.status = 'c' AND tk.tdate = input\_date

ORDER BY t.train\_name;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_confirmed\_bookings\_on\_date('2024-10-15');

train\_name | passenger\_name | bogie\_no | no\_of\_berths | ticket\_amt

---------------+----------------+----------+--------------+------------

EXPRESS TRAIN | JOHN DOE | 1 | 1 | 150.00

EXPRESS TRAIN | JANE SMITH | 2 | 2 | 150.00

(2 rows)

Function -2

CREATE OR REPLACE FUNCTION get\_berths\_amt(input\_date DATE,

passenger\_name\_input VARCHAR(20))

RETURNS TABLE (no\_of\_berths INT, ticket\_amt DECIMAL)

AS

$$

BEGIN

IF NOT EXISTS (SELECT 1 FROM passenger WHERE passenger\_name =passenger\_name\_input) THEN

RAISE EXCEPTION 'Invalid passenger name';

END IF;

RETURN QUERY

SELECT tk.no\_of\_berths, tk.ticket\_amt

FROM ticket tk

JOIN passenger p ON tk.passenger\_id = p.passenger\_id

WHERE tk.tdate = input\_date AND p.passenger\_name =passenger\_name\_input;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_berths\_amt('2024-10-15', 'JOHN DOE');

no\_of\_berths | ticket\_amt

--------------+------------

1 | 150.00

(1 row)

Function -3

CREATE OR REPLACE FUNCTION get\_ticket\_details(train\_name\_input

VARCHAR(20))

RETURNS TABLE (ticket\_no INT, passenger\_name VARCHAR(20), no\_of\_berths INT, ticket\_amt DECIMAL)

AS

$$

BEGIN

IF NOT EXISTS (SELECT 1 FROM train WHERE train\_name = train\_name\_input)

THEN

RAISE EXCEPTION 'Invalid train name';

END IF;

RETURN QUERY

SELECT tk.ticket\_no, p.passenger\_name, tk.no\_of\_berths, tk.ticket\_amt

FROM ticket tk

JOIN train t ON tk.train\_no = t.train\_no

JOIN passenger p ON tk.passenger\_id = p.passenger\_id

WHERE t.train\_name = train\_name\_input;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_ticket\_details('EXPRESS TRAIN');

ticket\_no | passenger\_name | no\_of\_berths | ticket\_amt

-----------+----------------+--------------+------------

1001 | JOHN DOE | 1 | 150.00

1002 | JANE SMITH | 2 | 150.00

(2 rows)

Q3)

select \* from branch;

bid | brname | brcity

-----+--------------------------------+----------------------

1 | Main Branch | Nagar

2 | Karve Nagar Branch | Karve Nagar

3 | ShivajiNagar Branch | ShivajiNagar

(3 rows)

select \* from customer;

cno | cname | caddr | city

-----+--------------------------------+-------------------------------------+----------------------

107 | Sneha Deshmukh | 34 Vasant Vihar | Nagar

108 | Amit Kothari | 56 Rajendra Nagar | Karve Nagar

109 | Priya Kulkarni | 78 Shivaji Nagar | Karve Nagar

110 | Nikhil Sharma | 90 Laxmi Road | ShivajiNagar

106 | Ravi Patil | 12 Ganesh Lane | Nagar

(5 rows)

select \* from loan\_application;

lno | l\_amt\_require | l\_amt\_approved | l\_date

------+---------------+----------------+------------

1001 | $150,000.00 | $140,000.00 | 2024-07-15

1002 | $200,000.00 | $180,000.00 | 2024-08-20

1003 | $250,000.00 | $220,000.00 | 2024-09-10

1004 | $300,000.00 | $290,000.00 | 2024-07-25

1005 | $50,000.00 | $45,000.00 | 2024-08-15

(5 rows)

select \* from ternary;

bid | cno | lno

-----+-----+------

1 | 106 | 1001

1 | 107 | 1004

2 | 108 | 1002

2 | 109 | 1005

3 | 110 | 1003

(5 rows)

Function -1

CREATE OR REPLACE FUNCTION unapproved(branch\_name\_input

VARCHAR(50))

RETURNS TABLE (cname VARCHAR(30), caddr VARCHAR(100), l\_amt\_require DECIMAL, l\_amt\_approved DECIMAL)

AS

$$

BEGIN

RETURN QUERY

SELECT c.cname::VARCHAR(30), c.caddr::VARCHAR(100),la.l\_amt\_require::DECIMAL, la.l\_amt\_approved::DECIMAL

FROM customer c

JOIN ternary t ON c.cno = t.cno

JOIN loan\_application la ON t.lno = la.lno

JOIN branch b ON t.bid = b.bid

WHERE b.brname = branch\_name\_input AND la.l\_amt\_require >

la.l\_amt\_approved;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM unapproved('Main Branch');

cname | caddr | l\_amt\_require | l\_amt\_approved

----------------+-----------------+---------------+----------------

Ravi Patil | 12 Ganesh Lane | 150000.00 | 140000.00

Sneha Deshmukh | 34 Vasant Vihar | 300000.00 | 290000.00

(2 rows)

Function -2

CREATE OR REPLACE FUNCTION approved(branch\_name\_input VARCHAR(50))

RETURNS TABLE (cname VARCHAR(30), lno INT, INT, l\_amt\_approved DECIMAL)

AS

$$

BEGIN

RETURN QUERY

SELECT c.cname::VARCHAR(30), t.lno, la.l\_amt\_approved::DECIMAL

FROM customer c

JOIN ternary t ON c.cno = t.cno

JOIN loan\_application la ON t.lno = la.lno

JOIN branch b ON t.bid = b.bid

WHERE b.brname = branch\_name\_input AND la.l\_date >= '2019-06-01';

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM approved('Main Branch');

cname | lno | l\_amt\_approved

----------------+------+----------------

Ravi Patil | 1001 | 140000.00

Sneha Deshmukh | 1004 | 290000.00

(2 rows)

Function -3

CREATE OR REPLACE FUNCTION total\_loans() RETURNS DECIMAL AS

$$

DECLARE

total\_approved DECIMAL;

BEGIN

SELECT SUM(l\_amt\_approved) INTO total\_approved

FROM loan\_application

WHERE l\_date > '2019-05-30';

RETURN COALESCE(total\_approved, 0);

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT total\_loans();

total\_loans

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

875000.00

(1 row)

SET B

Q1)

select \* from movie;

m\_name | release\_year | budget

--------------------------+--------------+-----------------

Sholey | 1975 | $6,000,000.00

Dil Chahta Hai | 2001 | $30,000,000.00

Zindagi Na Milegi Dobara | 2011 | $60,000,000.00

War | 2019 | $200,000,000.00

(4 rows)

select \* from actor;

a\_name | role | charges | a\_address

------------------+----------------------+----------------+-----------

Amitabh Bachchan | Lead Actor | $10,000,000.00 | Mumbai

Aamir Khan | Lead Actor | $12,000,000.00 | Mumbai

Farhan Akhtar | Supporting Actor | $8,000,000.00 | Mumbai

Hrithik Roshan | Lead Actor | $15,000,000.00 | Mumbai

(4 rows)

select \* from producer;

producer\_id | p\_name | p\_address

-------------+--------------------------------+-----------

1 | Mr. Subhash Ghai | Mumbai

2 | Excel Entertainment | Mumbai

3 | Zee Studios | Mumbai

4 | Yash Raj Films | Mumbai

(4 rows)

select \* from movie\_actor;

m\_name | a\_name

--------------------------+------------------

Sholey | Amitabh Bachchan

Dil Chahta Hai | Aamir Khan

Zindagi Na Milegi Dobara | Farhan Akhtar

War | Hrithik Roshan

Zindagi Na Milegi Dobara | Aamir Khan

Dil Chahta Hai | Farhan Akhtar

(6 rows)

select \* from movie\_producer;

m\_name | producer\_id

--------------------------+-------------

Sholey | 1

Sholey | 2

Dil Chahta Hai | 2

Dil Chahta Hai | 3

Zindagi Na Milegi Dobara | 2

Zindagi Na Milegi Dobara | 3

(6 rows)

FUNCTION-1

CREATE OR REPLACE FUNCTION get\_actors\_by\_movie(movie\_name\_input VARCHAR)

RETURNS TABLE (a\_name VARCHAR, role VARCHAR, charges DECIMAL,a\_address VARCHAR)

AS

$$

BEGIN

IF NOT EXISTS (SELECT 1 FROM movie WHERE m\_name = movie\_name\_input) THEN

RAISE EXCEPTION 'Invalid movie name: %', movie\_name\_input;

END IF;

RETURN QUERY

SELECT a.a\_name,a.role::VARCHAR,a.charges::DECIMAL,a.a\_address::VARCHAR

FROM actor a

JOIN movie\_actor ma ON a.a\_name = ma.a\_name

WHERE ma.m\_name = movie\_name\_input

ORDER BY a.charges DESC;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_actors\_by\_movie('Dil Chahta Hai');

a\_name | role | charges | a\_address

---------------+------------------+-------------+-----------

Aamir Khan | Lead Actor | 12000000.00 | Mumbai

Farhan Akhtar | Supporting Actor | 8000000.00 | Mumbai

(2 rows)

FUNCTION-2

CREATE OR REPLACE FUNCTION get\_movies\_by\_actor(actor\_name\_input VARCHAR)

RETURNS TABLE (m\_name VARCHAR)

AS

$$

BEGIN

IF NOT EXISTS (SELECT 1 FROM actor WHERE a\_name = actor\_name\_input)

THEN

RAISE EXCEPTION 'Invalid actor name: %', actor\_name\_input;

END IF;

RETURN QUERY

SELECT ma.m\_name

FROM movie\_actor ma

WHERE ma.a\_name = actor\_name\_input;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=# SELECT \* FROM get\_movies\_by\_actor('Aamir Khan');

m\_name

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

Dil Chahta Hai

Zindagi Na Milegi Dobara

(2 rows)

FUNCTION-3

CREATE OR REPLACE FUNCTION

count\_movies\_by\_producer(producer\_name\_input VARCHAR)

RETURNS INTEGER

AS

$$

DECLARE

movie\_count INTEGER;

BEGIN

IF NOT EXISTS (SELECT 1 FROM producer WHERE p\_name = producer\_name\_input) THEN

RAISE EXCEPTION 'Invalid producer name: %', producer\_name\_input;

END IF;

SELECT COUNT(\*) INTO movie\_count

FROM movie\_producer mp

JOIN producer p ON mp.producer\_id = p.producer\_id

WHERE p.p\_name = producer\_name\_input;

RETURN movie\_count;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT count\_movies\_by\_producer('Excel Entertainment');

count\_movies\_by\_producer

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

3

(1 row)

Q2)

create table wholesaler (w\_no int primary key, w\_name varchar(50), address varchar(100), city varchar(50));

CREATE TABLE

create table product (product\_no int primary key, product\_name varchar(50),

rate decimal(10, 2) check (rate > 0));

CREATE TABLE

create table wholesaler\_product (w\_no int, product\_no int, quantity int,

primary key (w\_no, product\_no), foreign key (w\_no) references wholesaler (w\_no),

foreign key (product\_no) references product (product\_no));

CREATE TABLE

select \* from wholesaler;

w\_no | w\_name | address | city

------+------------------+---------------------+--------

1 | ABC Traders | 123 Market St | Pune

2 | XYZ Supplies | 456 Commerce Rd | Mumbai

3 | PQR Distributors | 789 Industrial Area | Nagpur

4 | LMN Wholesales | 101 Business Park | Nashik

(4 rows)

select \* from product;

product\_no | product\_name | rate

------------+--------------+----------

101 | Laptop | 50000.00

102 | Mobile | 20000.00

103 | Tablet | 30000.00

104 | Smartwatch | 15000.00

(4 rows)

mydb=# select \* from wholesaler\_product;

w\_no | product\_no | quantity

------+------------+----------

1 | 101 | 100

2 | 102 | 150

3 | 103 | 80

4 | 104 | 120

(4 rows)

FUNCTION-1

CREATE OR REPLACE FUNCTION add\_quantity(input\_quantity INT, input\_w\_no INT, input\_product\_no INT)

RETURNS VOID AS

$$

BEGIN

-- Check if the quantity is within the valid range

IF input\_quantity < 50 OR input\_quantity > 200 THEN

RAISE EXCEPTION 'Quantity must be between 50 and 200.'; -- Raise an

exception if out of range

ELSE

-- Insert the record into wholesaler\_product table

INSERT INTO wholesaler\_product (w\_no, product\_no, quantity)

VALUES (input\_w\_no, input\_product\_no, input\_quantity);

END IF;

END;

$$

LANGUAGE plpgsql;

CREATE FUNCTION

SELECT add\_quantity(300, 1, 101);

ERROR: Quantity must be between 50 and 200.

CONTEXT: PL/pgSQL function add\_quantity(integer,integer,integer) line 5 at RAISE

FUNCTION-2

CREATE OR REPLACE FUNCTION validate\_rate(rate DECIMAL)

RETURNS VOID AS

$$

BEGIN

IF rate <= 0 THEN

RAISE EXCEPTION 'Invalid rate value. Rate must be greater than zero.';

ELSE

RAISE NOTICE 'Correct Input';

END IF;

END;

$$

LANGUAGE plpgsql;

CREATE FUNCTION

SELECT validate\_rate(-10);

ERROR: Invalid rate value. Rate must be greater than zero.

SET C

Q1)

create table lab (labno int primary key, labname varchar, capacity int not null,equipment varchar);

CREATE TABLE

create table student4 (rollno int primary key, sname varchar, class varchar,timetable varchar, mobileno varchar, labno int references lab(labno));

CREATE TABLE

select \* from lab;

labno | labname | capacity | equipment

-------+---------------+----------+--------------------------

1 | physics lab | 30 | projector, computers

2 | chemistry lab | 35 | microscopes, test tubes

3 | biology lab | 50 | dissection tools, slides

(3 rows)

select \* from student4;

rollno | sname | class | timetable | mobileno | labno

--------+---------------+-------+-------------+------------+-------

1 | johndoe | 10th | 9:00-10:00 | 1234567890 | 1

2 | jane smith | 10th | 10:00-11:00 | 0987654321 | 1

3 | alice johnson | 9th | 11:00-12:00 | 1029384756 | 2

(3 rows)

FUNCTION-1

CREATE OR REPLACE FUNCTION get\_students\_by\_lab(input\_lab\_no INT)

RETURNS TABLE (rollno INT, sname VARCHAR, class VARCHAR, timetable VARCHAR, mobileno VARCHAR) AS $$

BEGIN

IF NOT EXISTS (SELECT 1 FROM lab WHERE labno = input\_lab\_no) THEN

RAISE EXCEPTION 'Invalid Lab Number.';

ELSE

RETURN QUERY SELECT s.rollno, s.sname, s.class, s.timetable, s.mobileno

FROM student4 s WHERE s.labno = input\_lab\_no; -- Corrected

column name

END IF;

END;

$$

LANGUAGE plpgsql;

CREATE FUNCTION

SELECT \* FROM get\_students\_by\_lab(1);

rollno | sname | class | timetable | mobileno

--------+------------+-------+-------------+------------

1 | johndoe | 10th | 9:00-10:00 | 1234567890

2 | jane smith | 10th | 10:00-11:00 | 0987654321

(2 rows)

FUNCTION-2

CREATE OR REPLACE FUNCTION validate\_mobile\_number(input\_mobile\_no VARCHAR)

RETURNS VOID AS $$

BEGIN

IF LENGTH(input\_mobile\_no) <> 10 OR NOT input\_mobile\_no ~ '^[0-9]+$'

THEN

RAISE EXCEPTION 'Invalid Mobile Number.';

ELSE

RAISE NOTICE 'Correct input...!';

END IF;

END;

$$ LANGUAGE plpgsql;

CREATE FUNCTION

SELECT validate\_mobile\_number('1234567890');

NOTICE: Correct input...!

validate\_mobile\_number

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

(1 row)

SELECT validate\_mobile\_number('12345');

ERROR: Invalid Mobile Number.

CONTEXT: PL/pgSQL function validate\_mobile\_number(character varying) line 4 at

RAISE

FUNCTION-3

CREATE OR REPLACE FUNCTION add\_lab\_details(input\_lab\_no INT,

input\_lab\_name VARCHAR, input\_capacity INT, input\_equipment VARCHAR)

RETURNS VOID AS $$

BEGIN

IF input\_capacity > 40 THEN

RAISE EXCEPTION 'Invalid\_Capacity\_Range';

ELSE

INSERT INTO lab (LabNo, LabName, Capacity, Equipment) VALUES (input\_lab\_no, input\_lab\_name, input\_capacity, input\_equipment);

END IF;

END;

$$

LANGUAGE plpgsql;

CREATE FUNCTION

SELECT add\_lab\_details(4, 'Chemistry Lab', 35, 'Beakers, Bunsen Burners');

add\_lab\_details

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

(1 row)

SELECT add\_lab\_details(2, 'Physics Lab', 45, 'Oscilloscope, Multimeter');

ERROR: Invalid\_Capacity\_Range

CONTEXT: PL/pgSQL function add\_lab\_details(integer,character varying,integer,character varying) line 4 at RAISE