

# Advance Database Management System Lab

## Ens Term Lab Test

Aryan Mohan

500092142

Batch- 2

### Experiment – 9

1. Write a PL/SQL code to accept the value of A, B & C display which is greater.
2. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQLProgramming” 20 times.
3. Write a PL/SQL code block to find the factorial of a number.
4. Write a PL/SQL program to generate Fibonacci series.
5. Write a PL/SQL code to fund the sum of first N numbers

--1) Write a PL/SQL code to accept the value of A, B & C display which is greater.

```
BEGIN
DECLARE @A INTEGER;
SET @A =45;
DECLARE @B INTEGER;
SET @B = 65;
DECLARE @C INTEGER;
SET @C =25;
IF @A>@B AND @A>@C
PRINT 'GREATEST IS A';
ELSE IF @B>@C AND @B>@A
PRINT 'GREATEST IS B';
ELSE
PRINT 'GREATEST IS C';
END;
```

Output:

---

GREATEST IS B

Completion time: 2023-04-27T14:42:12.6708994+05:30

--2) Using PL/SQL Statements create a simple loop that display message "Welcome to PL/SQL Programming" 20 times

```
DECLARE @i integer;
set @i=1;
while @i<=20
BEGIN
PRINT 'Welcome to PL/SQL Programming';
set @i=@i+1;
END
```

Output:

```
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
```

-- 3) Write a PL/SQL code block to find the factorial of a number.

```
DECLARE @fact integer, @n integer;
set @fact=1;
set @n=9;
while @n > 0
begin
set @fact=@n*@fact
set @n=@n-1
end
print @fact
```

Output:

```
362880
```

```
Completion time: 2023-04-27T14:41:44.2719532+05:30
```

--4) Write a PL/SQL program to generate Fibonacci series.

```

declare @f1 INTEGER=10, @f2 INTEGER=12,@f3 INTEGER,@i
INTEGER=3,@len INTEGER;
print 'First two number'
print @f1;
print @f2;
set @len=15;
print 'fibonacci series is';
while(@i<=@len)
begin
set @f3=@f1+@f2;
print @f3
set @f1=@f2;
set @f2=@f3;
set @i=@i+1;
end;

```

Output:

```

First two number
10
12
fibonacci series is
22
34
56
90
146
236
382
618
1000
1618
2618
4236
6854

```

Completion time: 2023-04-27T14:41:26.7003765+05:30

--5) Write a PL/SQL code to find the sum of first N numbers

```

declare @n integer, @i integer, @sum integer = 0;
set @i = 1;
set @n=25;
while (@i <= @n)
begin
set @sum=@sum+@i
set @i=@i+1
end
print 'Sum of first N=10 numbers:'
print @sum

```

Output:

```
Sum of first N=10 numbers:  
325
```

```
Completion time: 2023-04-27T14:41:07.3180330+05:30
```

## EXPERIMENT-4 C

The following relations keep track of airline flight information:

**Flights**(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price:integer)

**Aircraft**(aid: integer, aname: string, cruisingrange: integer)

**Certified**(eid: integer, aid: integer)  
**Employees**(eid: integer, ename: string, salary: integer)

1. Find the names of aircraft such that all pilots certified to operate them earn more than 80,000.
2. For each pilot who is certified for more than three aircraft, find the eid and the maximumcruisingrange of the aircraft that he (or she) is certified for.
3. Find the names of pilots whose salary is less than the price of the cheapest route from LosAngeles to Honolulu.
4. For all aircraft with cruisingrange over 1,000 miles, find the name of the aircraft and theaverage salary of all pilots certified for this aircraft.5. Find the names of pilots certified for some Boeing aircraft

```
create database EndSemLabExam;  
use EndSemLabExam;
```

```
CREATE TABLE Flights (  
    flno INTEGER,  
    from_ VARCHAR(255),  
    to_ VARCHAR(255),  
    distance INTEGER,  
    departs TIME,  
    arrives TIME,  
    price INTEGER,  
    PRIMARY KEY (flno)  
);
```

```
CREATE TABLE Aircraft (  
    aid INTEGER,  
    aname VARCHAR(255),  
    cruisingrange INTEGER,  
    PRIMARY KEY (aid)  
);
```

```
CREATE TABLE Certified (  
    eid INTEGER,  
    aid INTEGER,  
    PRIMARY KEY (eid, aid),  
    FOREIGN KEY (eid) REFERENCES Employees(eid),  
    FOREIGN KEY (aid) REFERENCES Aircraft(aid)  
);
```

```
CREATE TABLE Employees (  
    eid INTEGER,  
    ename VARCHAR(255),  
    salary INTEGER,  
    PRIMARY KEY (eid)  
);
```

-- Insert 5 entries into Flights table

```
INSERT INTO Flights VALUES(1, 'New York', 'Los Angeles',  
2475, '08:00:00', '11:30:00', 500);
```

```
INSERT INTO Flights VALUES(2, 'Los Angeles', 'Chicago',  
1745, '13:00:00', '17:00:00', 350);
```

```
INSERT INTO Flights VALUES(3, 'Chicago', 'Miami', 1250,  
'10:00:00', '14:00:00', 400);
```

```
INSERT INTO Flights VALUES(4, 'Miami', 'Houston', 970,  
'08:30:00', '10:30:00', 200);
```

```
INSERT INTO Flights VALUES(5, 'Houston', 'San Francisco',  
1635, '12:00:00', '16:30:00', 450);
```

-- Insert 5 entries into Aircraft table

```
INSERT INTO Aircraft VALUES(1, 'Airbus A320', 3500);
```

```
INSERT INTO Aircraft VALUES(1, 'Airbus A320', 3500);
```

```
INSERT INTO Aircraft VALUES(2, 'Boeing 747', 8000);
```

```
INSERT INTO Aircraft VALUES(3, 'Bombardier CRJ200', 1700);
```

```
INSERT INTO Aircraft VALUES(4, 'Embraer E175', 2400);
```

```
INSERT INTO Aircraft VALUES(5, 'Boeing 737', 5500);
```

```

-- Insert 5 entries into Certified table
INSERT INTO Certified VALUES(1, 1);
INSERT INTO Certified VALUES(2, 2);
INSERT INTO Certified VALUES(3, 2);
INSERT INTO Certified VALUES(4, 3);
INSERT INTO Certified VALUES(5, 4);

-- Insert 5 entries into Employees table
INSERT INTO Employees VALUES(1, 'John Doe', 75000);
INSERT INTO Employees VALUES(2, 'Jane Smith', 90000);
INSERT INTO Employees VALUES(3, 'Bob Johnson', 80000);
INSERT INTO Employees VALUES(4, 'Sara Lee', 85000);
INSERT INTO Employees VALUES(5, 'Mike Smith', 95000);

SELECT aname
FROM Aircraft
WHERE cruisingrange > ALL (
    SELECT cruisingrange
    FROM Aircraft
    JOIN Certified ON Aircraft.aid = Certified.aid
    JOIN Employees ON Certified.eid = Employees.eid
    WHERE salary <= 80000
);

```

Output:

	aname
1	Airbus A320
2	Boeing 747
3	Bombardier CRJ200
4	Embraer E175
5	Boeing 737

```

SELECT eid, MAX(cruisingrange)
FROM Certified
JOIN Aircraft ON Certified.aid = Aircraft.aid
GROUP BY eid
HAVING COUNT(*) > 3;

```

Output:

eid	(No column name)
-----	------------------

```

SELECT ename
FROM Employees
WHERE salary < (
    SELECT MIN(price)
    FROM Flights
    WHERE from_ = 'Los Angeles' AND to_ = 'Honolulu'
);

```

Output:

ename
-------

```

SELECT aname, AVG(salary)
FROM Aircraft
JOIN Certified ON Aircraft.aid = Certified.aid
JOIN Employees ON Certified.eid = Employees.eid
WHERE cruisingrange > 1000
GROUP BY Aircraft.aid;

```

Output:

```

SELECT ename
FROM Employees
JOIN Certified ON Employees.eid = Certified.eid
JOIN Aircraft ON Certified.aid = Aircraft.aid
WHERE aname LIKE 'Boeing%';

```

Output:

ename
-------