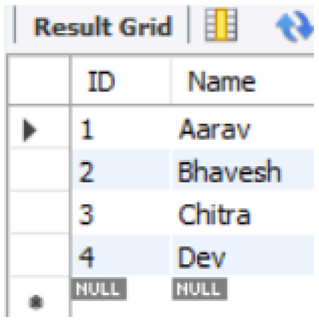
PL/SQL Lab

Student Name: Bhawesh Rawat **UID:**24MCA20121
Branch: MCA(General) **Section/Group:** 2B
Semester: 1st **Date of Performance:**26 Sep 2024
Subject Name: PL/SQL Lab **Subject Code:**24CAP-602

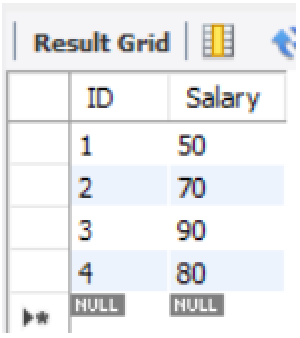
1. **Aim/Overview of the practical:**

You are given 3 tables students, friend and packages. Students contain 2 columns Id and Name, friend contains 2 columns Id and Friend-id(id of only bestfriend), packages contained 2 columns Id and Salary. Write a query to outputs the names of those students whose bestfriend got offered a higher salary than them names must be ordered by the salary amount offered to the best friend it is garneted that no 2 students got same salary offer**.**

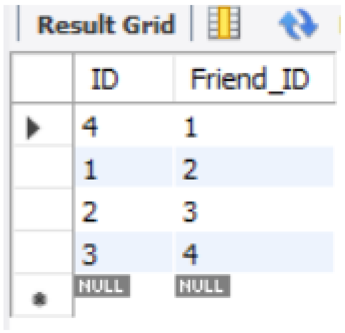
- **Table:**



ID	Name
1	Aarav
2	Bhavesh
3	Chitra
4	Dev
NULL	NULL



ID	Salary
1	50
2	70
3	90
4	80
NULL	NULL



ID	Friend_ID
4	1
1	2
2	3
3	4
NULL	NULL

- **Join:**

  INNER JOIN:

  Retrieves records that have matching values in both tables. If there is no match, the record is excluded.

  LEFT JOIN

  Returns all records from the left table, and the matched records from the right table. If there is no match, it will return NULL for the right table columns.

  RIGHT JOIN

  It returns all records from the right table and the matched records from the left table.

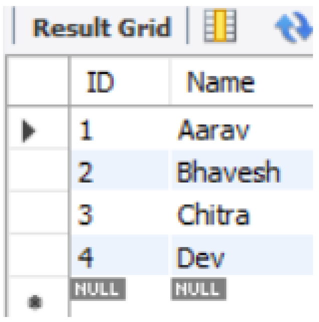
  FULL JOIN

  Returns all records when there is a match in either left or right table. If no match exists, NULL values are returned for non-matching rows.

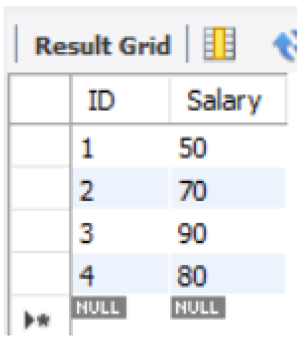
2. Procedure:

```
 CREATE TABLE Students (
    ID INT PRIMARY KEY,
    Name VARCHAR(50)
 );

 INSERT INTO Students (ID, Name) VALUES
 (1, 'Aarav'),
 (2, 'Bhavesh'),
 (3, 'Chitra'),
 (4, 'Dev');

CREATE TABLE Friends (
   ID INT,
   Friend_ID INT,
   PRIMARY KEY (ID, Friend_ID),
   FOREIGN KEY (ID) REFERENCES Students(ID),
   FOREIGN KEY (Friend_ID) REFERENCES Students(ID)
);
```

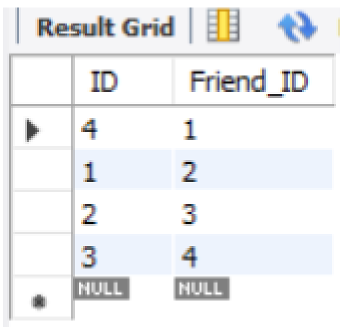
```
INSERT INTO Friends (ID, Friend_ID) VALUES
(1, 2),
(2, 3),
(3, 4),
(4, 1);

CREATE TABLE Packages (
   ID INT PRIMARY KEY,
   Salary INT,
   FOREIGN KEY (ID) REFERENCES Students(ID)
);

INSERT INTO Packages (ID, Salary) VALUES
(1, 50),
(2, 70),
(3, 90),
(4, 80);

select * from students;
select * from friends;
select * from packages;
```



ID	Name
1	Aarav
2	Bhavesh
3	Chitra
4	Dev
NULL	NULL



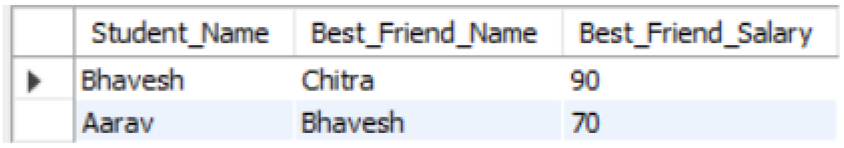
ID	Salary
1	50
2	70
3	90
4	80
NULL	NULL



ID	Friend_ID
4	1
1	2
2	3
3	4
NULL	NULL

1. Sorting the output in descending order of the best friend's salary, showing students whose best friends received the highest salaries first.

```
SELECT S1.Name AS Student_Name, S2.Name AS Best_Friend_Name,  P2.Salary AS
Best_Friend_Salary FROM Students S1  JOIN Friends F ON S1.ID = F.ID JOIN Students S2 ON
F.Friend_ID = S2.ID JOIN Packages P1 ON S1.ID = P1.ID JOIN Packages P2 ON S2.ID = P2.ID
WHERE P2.Salary > P1.Salary ORDER BY P2.Salary DESC;
```



Student_Name	Best_Friend_Name	Best_Friend_Salary
Bhavesh	Chitra	90
Aarav	Bhavesh	70

Learning outcomes (What I have learnt):

1. Learn about Join.
2. Learn How to sort data in a order.
3. Learn How to use multiple queries in one query.
4. Learn how to only show limited information.

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance		12
2.	Worksheet		8
3.	Viva		10