

## Students Data:

MySQL Workbench

IVY Practice x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ipl  
ivy\_sql\_internship  
practice  
sakila  
student  
student\_db1  
student\_salary

Tables

friends  
packages  
students

Administration Schemas

Information

Table: friends

Columns:

ID int  
Friends\_ID int

Day\_5\_IVY\_Internship\_Rounak P...

```
3 • use student_Salary;
4 /*Creating tables Students, Packages, Friends and inseting values in them*/
5 • create table Students(ID int, Name varchar(100));
6 • insert into Students(ID,Name)values(1,"Ashley"),(2,"Samantha"),(3,"Julia"),(4,"Scarlet");
7 • create table Packages(ID int, Salary float);
8 • insert into Packages(ID,Salary) values(1,15.20),(2,10.06),(3,11.55),(4,12.12);
9 • Create table Friends(ID int, Friends_ID int);
10 • insert into Friends(ID,Friends_ID) values(1,2),(2,3),(3,4),(4,1);
11 /*Checking the tables*/
12 • select * from students;
13 • select * from Packages;
14 • select * from Friends;
15 /*Write a query to output the names of those students whose friends got offered a higher salary than them.
16 Names must be ordered by the salary amount offered to the friends. It is guaranteed that no two students got same salary offer.*/
17 • select Name from
18 (select S1.ID, S1.Name as Name, P1.Salary, F1.Friends_ID,S2.Name as Friends_name, P2.Salary as Friends_Salary
19 from Students as S1
```

Result Grid

ID	Name
1	Ashley
2	Samantha
3	Julia
4	Scarlet

students 19 x

Output

## Packages Data:

MySQL Workbench

IVY Practice x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ipl  
ivy\_sql\_internship  
practice  
sakila  
student  
student\_db1  
student\_salary

Tables

friends  
packages  
students

Administration Schemas

Information

Table: friends

Columns:

ID int  
Friends\_ID int

Day\_5\_IVY\_Internship\_Rounak P...

```
3 • use student_Salary;
4 /*Creating tables Students, Packages, Friends and inseting values in them*/
5 • create table Students(ID int, Name varchar(100));
6 • insert into Students(ID,Name)values(1,"Ashley"),(2,"Samantha"),(3,"Julia"),(4,"Scarlet");
7 • create table Packages(ID int, Salary float);
8 • insert into Packages(ID,Salary) values(1,15.20),(2,10.06),(3,11.55),(4,12.12);
9 • Create table Friends(ID int, Friends_ID int);
10 • insert into Friends(ID,Friends_ID) values(1,2),(2,3),(3,4),(4,1);
11 /*Checking the tables*/
12 • select * from students;
13 • select * from Packages;
14 • select * from Friends;
15 /*Write a query to output the names of those students whose friends got offered a higher salary than them.
16 Names must be ordered by the salary amount offered to the friends. It is guaranteed that no two students got same salary offer.*/
17 • select Name from
18 (select S1.ID, S1.Name as Name, P1.Salary, F1.Friends_ID,S2.Name as Friends_name, P2.Salary as Friends_Salary
19 from Students as S1
```

Result Grid

ID	Salary
1	15.2
2	10.06
3	11.55
4	12.12

Packages 20 x

Output

## Friends Data:

MySQL Workbench

IVY Practice x

File Edit View Query Database Server Tools Scripting Help

Day\_5\_IVY\_Internship\_Rounak P...

Limit to 50000 rows

**SCHEMAS**

Filter objects

- ipl
- ivy\_sql\_internship
- practice
- sakila
- student
- student\_db1
  - student\_salary**
    - Tables
      - friends
      - packages
      - students

Administration Schemas

Information

**Table: friends**

**Columns:**

- ID int
- Friends\_ID int

```
3 • use student_salary;
4 /*Creating tables Students, Packages, Friends and inseting values in them*/
5 • create table Students(ID int, Name varchar(100));
6 • insert into Students(ID,Name)values(1,"Ashley"),(2,"Samantha"),(3,"Julia"),(4,"Scarlet");
7 • create table Packages(ID int, Salary float);
8 • insert into Packages(ID,Salary) values(1,15.20),(2,10.06),(3,11.55),(4,12.12);
9 • Create table Friends(ID int, Friends_ID int);
10 • insert into Friends(ID,Friends_ID) values(1,2),(2,3),(3,4),(4,1);
11 /*Checking the tables*/
12 • select * from students;
13 • select * from Packages;
14 • select * from Friends;
15 /*Write a query to output the names of those students whose friends got offered a higher salary than them.
16 Names must be ordered by the salary amount offered to the friends. It is guaranteed that no two students got same salary offer.*/
17 • select Name from
18 (select S1.ID, S1.Name as Name, P1.Salary, F1.Friends_ID,S2.Name as Friends_name, P2.Salary as Friends_Salary
19 from Students as S1
```

Result Grid

ID	Friends_ID
1	2
2	3
3	4
4	1

Friends 21 x

Read Only

Output