1. Print the type of each node given in the binary tree.

```
        8
      /   \
    5       9
   / \     / \
  1   4   2   3
```

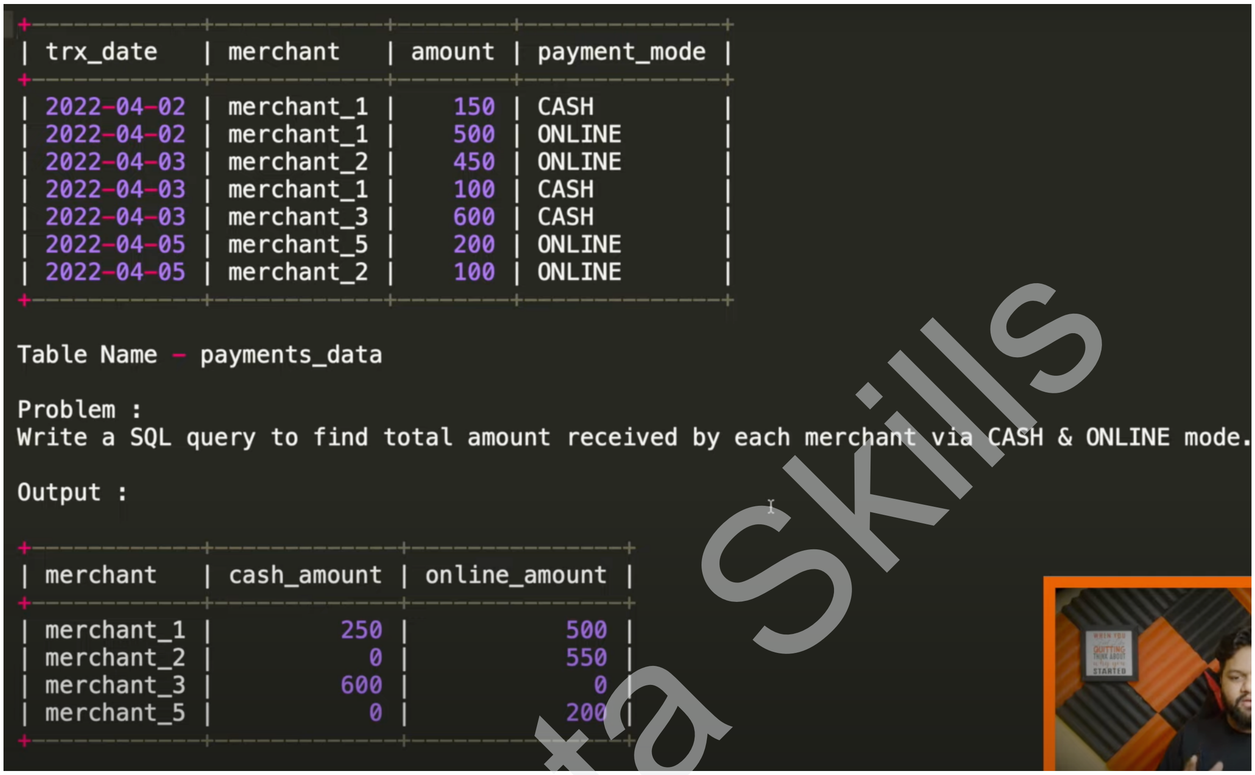
Input Table : tree(node, parent)

Node	Parent
5	8
9	8
4	5
2	9
1	5
3	9
8	Null

Output :

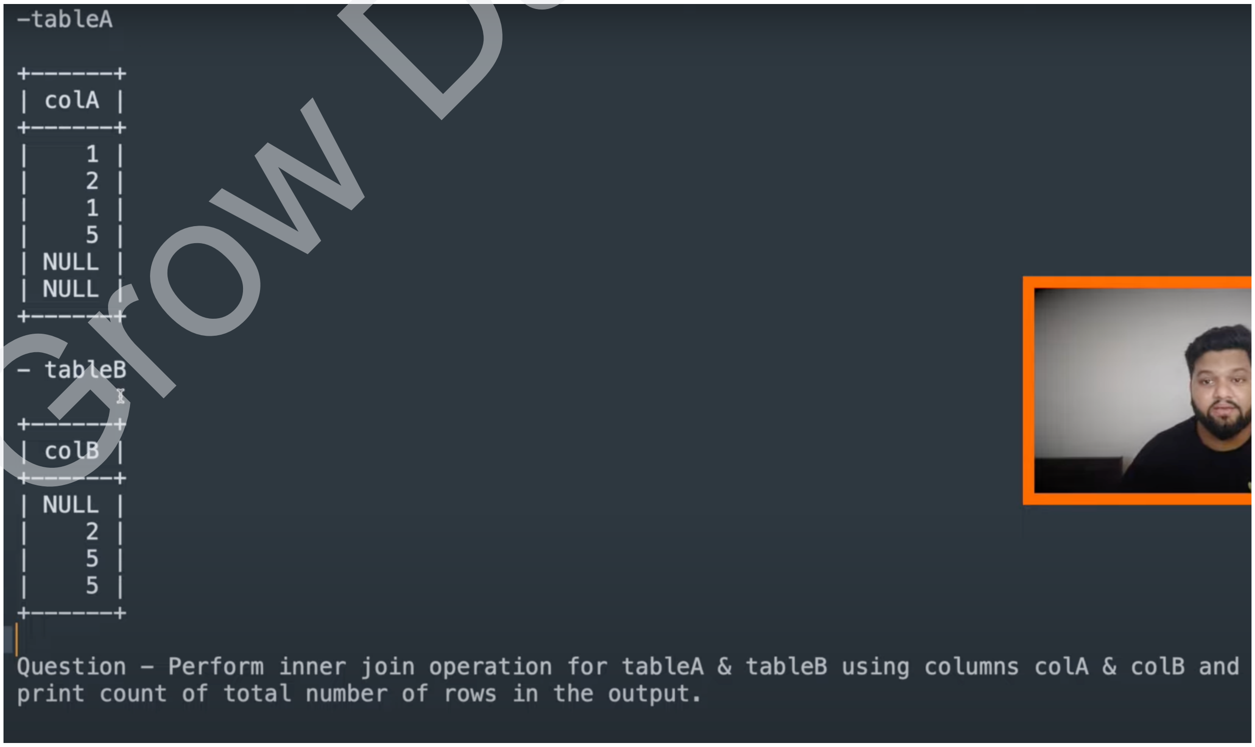
Node	Type
5	Inner
9	Inner
4	Leaf
2	Leaf
1	Leaf
3	Leaf
8	Root

2.



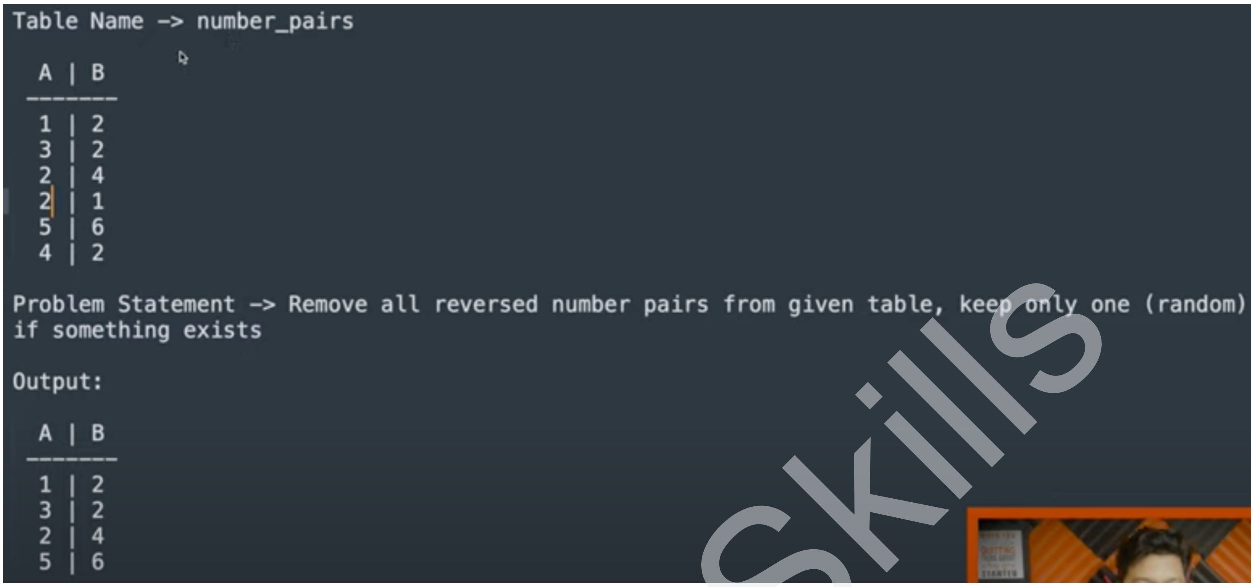
```
+------------+------------+--------+--------------+
| trx_date   | merchant   | amount | payment_mode |
+------------+------------+--------+--------------+
| 2022-04-02 | merchant_1 |    150 | CASH         |
| 2022-04-02 | merchant_1 |    500 | ONLINE       |
| 2022-04-03 | merchant_2 |    450 | ONLINE       |
| 2022-04-03 | merchant_1 |    100 | CASH         |
| 2022-04-03 | merchant_3 |    600 | CASH         |
| 2022-04-05 | merchant_5 |    200 | ONLINE       |
| 2022-04-05 | merchant_2 |    100 | ONLINE       |
+------------+------------+--------+--------------+

Table Name - payments_data

Problem :
Write a SQL query to find total amount received by each merchant via CASH & ONLINE mode.

Output :

+------------+-------------+---------------+
| merchant   | cash_amount | online_amount |
+------------+-------------+---------------+
| merchant_1 |         250 |           500 |
| merchant_2 |           0 |           550 |
| merchant_3 |         600 |             0 |
| merchant_5 |           0 |           200 |
+------------+-------------+---------------+
```

3.



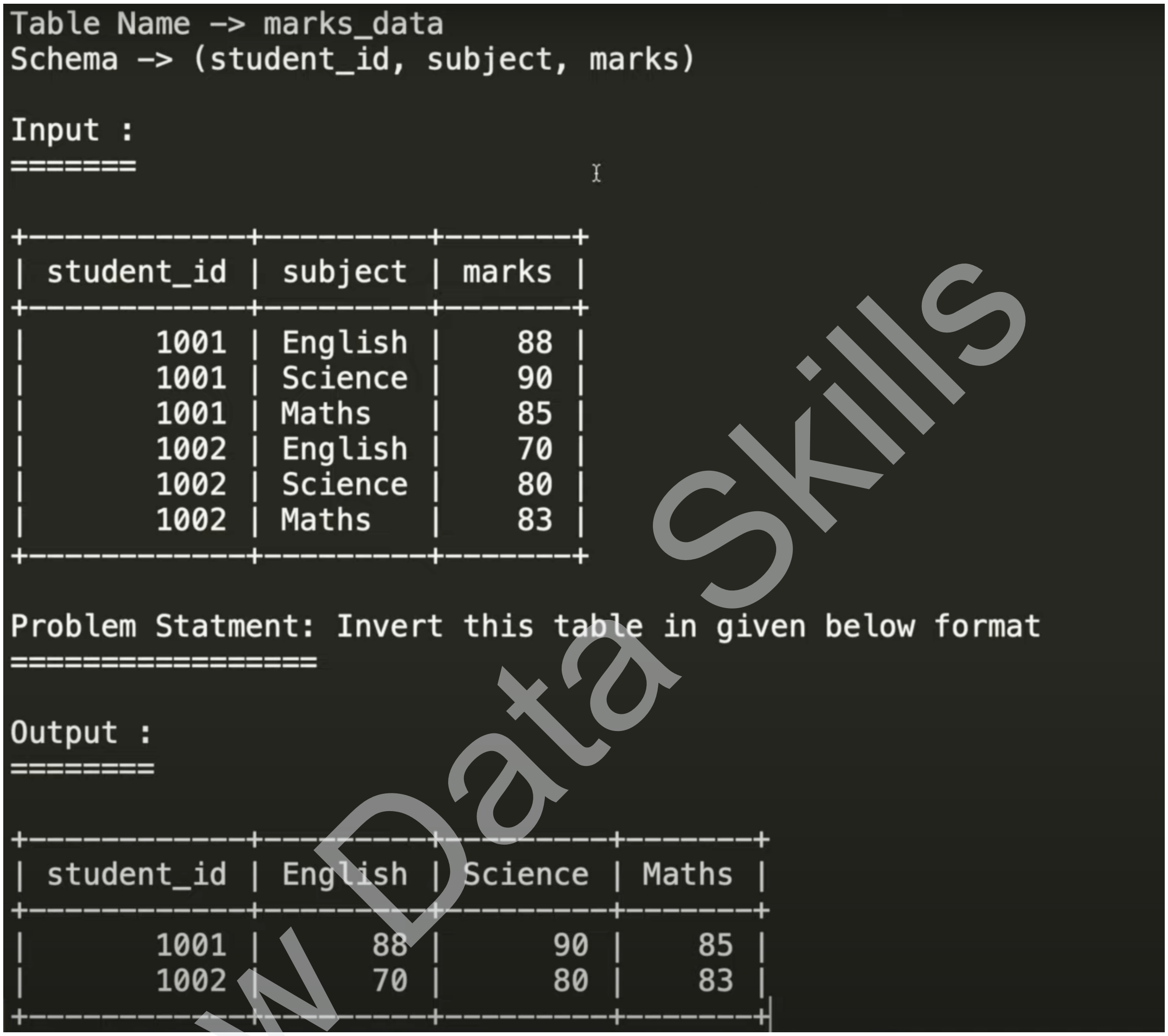
```
-tableA

+-------+
| colA  |
+-------+
|     1 |
|     2 |
|     1 |
|     5 |
| NULL  |
| NULL  |
+-------+

- tableB

+-------+
| colB  |
+-------+
| NULL  |
|     2 |
|     5 |
|     5 |
+-------+

Question - Perform inner join operation for tableA & tableB using columns colA & colB and
print count of total number of rows in the output.
```

4.



```
Table Name -> number_pairs

  A | B
 -------
  1 | 2
  3 | 2
  2 | 4
  2 | 1
  5 | 6
  4 | 2

Problem Statement -> Remove all reversed number pairs from given table, keep only one (random)
if something exists

Output:

  A | B
 -------
  1 | 2
  3 | 2
  2 | 4
  5 | 6
```

5.



```
Table Name -> marks_data
Schema -> (student_id, subject, marks)

Input :
=======

+------------+---------+-------+
| student_id | subject | marks |
+------------+---------+-------+
|       1001 | English |    88 |
|       1001 | Science |    90 |
|       1001 | Maths   |    85 |
|       1002 | English |    70 |
|       1002 | Science |    80 |
|       1002 | Maths   |    83 |
+------------+---------+-------+

Problem Statment: Invert this table in given below format
=================

Output :
========

+------------+---------+---------+-------+
| student_id | English | Science | Maths |
+------------+---------+---------+-------+
|       1001 |      88 |      90 |    85 |
|       1002 |      70 |      80 |    83 |
+------------+---------+---------+-------+
```

6.

Table -> Travel_Data

City1	City2	Price	
A	B	200	
A	C	300	
C	D	200	
A	B	200	——--> Direct duplicate
B	A	200	——---> Logical duplicate

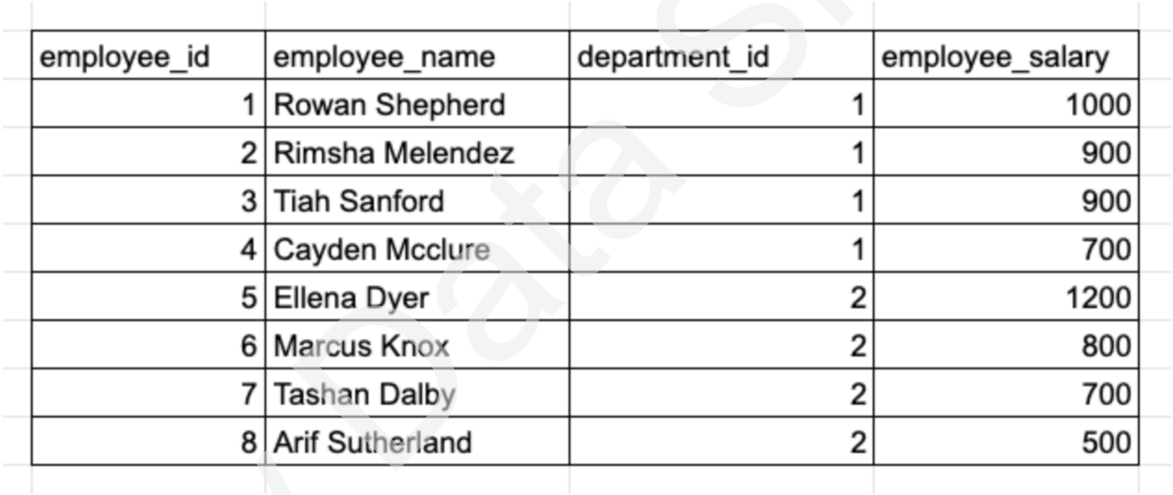
Problem Statement -> Remove all logical & direct duplicates from input table

Output:

```
City1, City2, Price
  A       B      200
  A       C     300
  C       D      200
```

7.

Find the top 3 employees who have the highest salary in each department.

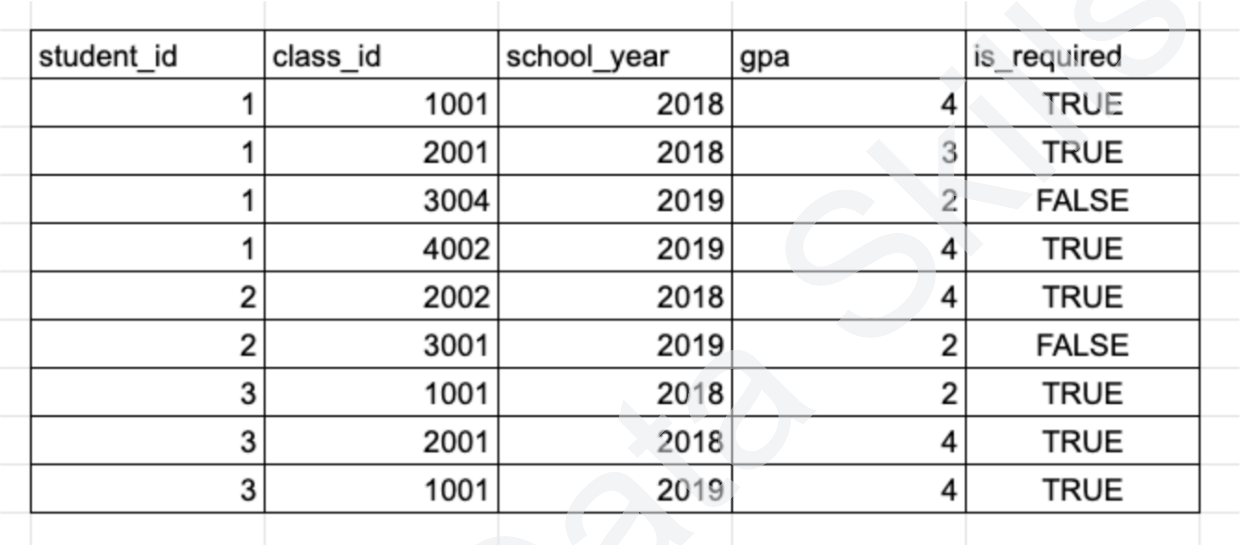


employee_id	employee_name	department_id	employee_salary
1	Rowan Shepherd	1	1000
2	Rimsha Melendez	1	900
3	Tiah Sanford	1	900
4	Cayden Mcclure	1	700
5	Ellena Dyer	2	1200
6	Marcus Knox	2	800
7	Tashan Dalby	2	700
8	Arif Sutherland	2	500

Another sample employee_salary table.

8.

Calculate the average required course GPA in each school year for each student and find students who are qualified for the Dean's List (GPA ≥ 3.5) in each semester.

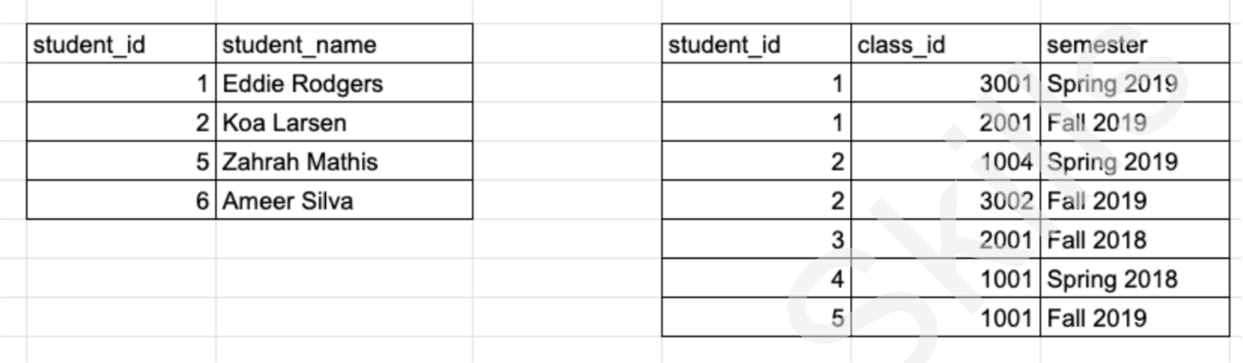


student_id	class_id	school_year	gpa	is_required
1	1001	2018	4	TRUE
1	2001	2018	3	TRUE
1	3004	2019	2	FALSE
1	4002	2019	4	TRUE
2	2002	2018	4	TRUE
2	3001	2019	2	FALSE
3	1001	2018	2	TRUE
3	2001	2018	4	TRUE
3	1001	2019	4	TRUE

The sample gpa_history table.

9.

Find the total number of classes taken by each student. (Provide student id, name and number of classes taken.)



student_id	student_name
1	Eddie Rodgers
2	Koa Larsen
5	Zahrah Mathis
6	Ameer Silva

student_id	class_id	semester
1	3001	Spring 2019
1	2001	Fall 2019
2	1004	Spring 2019
2	3002	Fall 2019
3	2001	Fall 2018
4	1001	Spring 2018
5	1001	Fall 2019

The sample student and class_history tables.

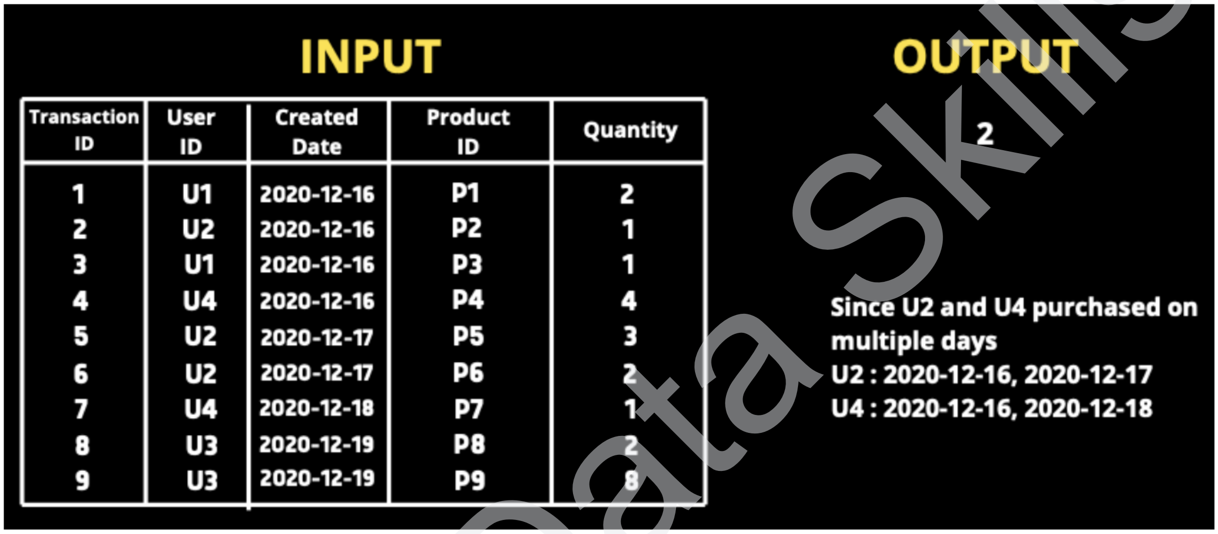
10.

For the following relation schema:
employee(employee-name, street, city)
works(employee-name, company-name, salary)
company(company-name, city)
manages(employee-name, manager-name)

a. Find the names of all employees in the database who live in the same cities as the companies for which they work.
b. Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'. Assume that all people work for at most one company.

11.

We are given a transaction table that consists of **transaction_id, user_id, transaction_date, product_id, and quantity**. We need to query the number of users who purchased products on multiple days(Note that a given user can purchase multiple products on a single day).



INPUT

Transaction ID	User ID	Created Date	Product ID	Quantity
1	U1	2020-12-16	P1	2
2	U2	2020-12-16	P2	1
3	U1	2020-12-16	P3	1
4	U4	2020-12-16	P4	4
5	U2	2020-12-17	P5	3
6	U2	2020-12-17	P6	2
7	U4	2020-12-18	P7	1
8	U3	2020-12-19	P8	2
9	U3	2020-12-19	P9	8

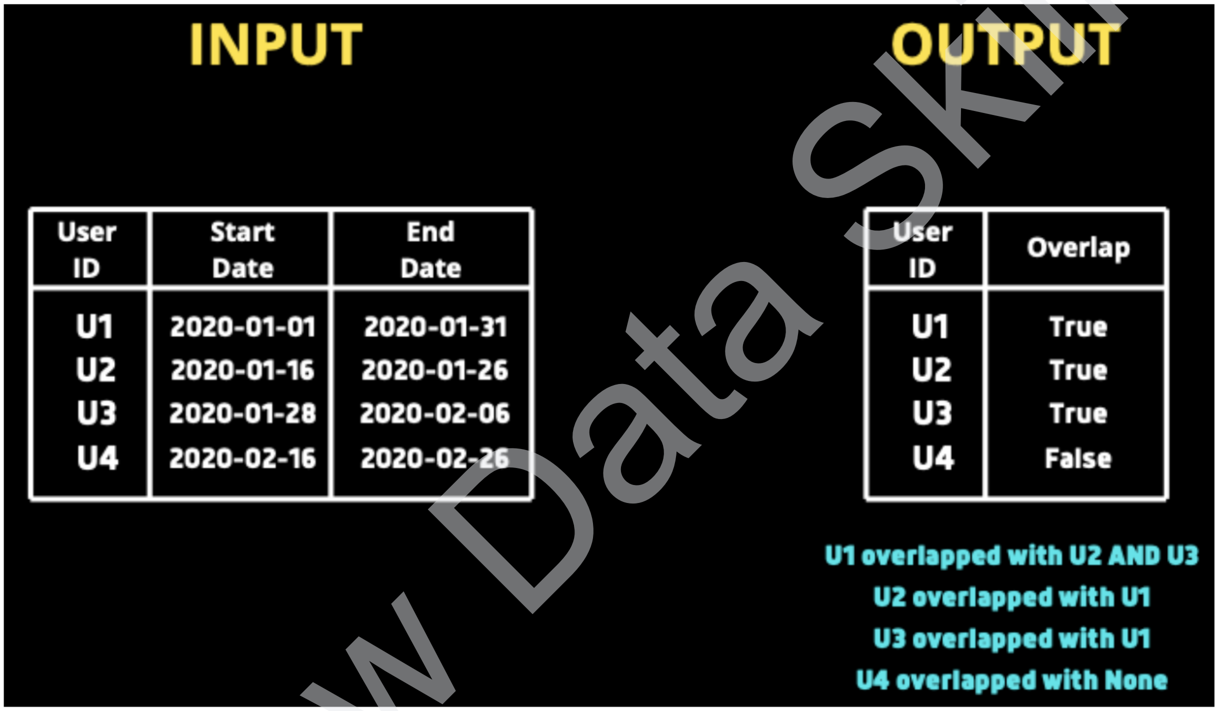
OUTPUT

2

Since U2 and U4 purchased on multiple days
U2 : 2020-12-16, 2020-12-17
U4 : 2020-12-16, 2020-12-18

12.

We are given a subscription table which consists of subscription start and end date for each user. We need to write a query that returns true/false for each user based on the overlapping of dates with other users. For instance, If user1's subscription period overlaps with any other user the query must return **True** for user1.



INPUT

User ID	Start Date	End Date
U1	2020-01-01	2020-01-31
U2	2020-01-16	2020-01-26
U3	2020-01-28	2020-02-06
U4	2020-02-16	2020-02-26

OUTPUT

User ID	Overlap
U1	True
U2	True
U3	True
U4	False

U1 overlapped with U2 AND U3
U2 overlapped with U1
U3 overlapped with U1
U4 overlapped with None