

### Task 3:

**Goal:** Write a query to output all such symmetric pairs in ascending order by the value of X.

**Table:** Functions with two columns: X (Integer) and Y (Integer).

**Definition of Symmetric Pair:** Two pairs (X, Y) and (Y, X) are said to be symmetric pairs IF  $X \neq Y$ , AND  $X = Y$ . **Sample Input (Functions):** | X | Y | |----|----| | 20 | 20 | | 20 | 21 | | 20 | 22 | | 21 | 20 | | 22 | 23 | | 23 | 22 | | 21 | 20 |

**Sample Output:** 20 20 20 21 22 23

Let's analyze the sample output against the input:

- **20 20:** This is a pair where  $X = Y$ .
- **20 21:** The symmetric counterpart (21, 20) also exists in the input. Here  $X \neq Y$ .
- **22 23:** The symmetric counterpart (23, 22) also exists in the input. Here  $X \neq Y$ .

This implies the true conditions for a symmetric pair are:

1. Pairs where  $X = Y$ .
2. Pairs where  $X \neq Y$ , AND the pair (Y, X) also exists in the table. And for the second case ( $X \neq Y$ ), we should only output one of the pairs, specifically the one where  $X < Y$  to avoid duplicates (e.g., if we output 20 21, we don't output 21 20). The sample output confirms this: 20 21 is output, not 21 20. 22 23 is output, not 23 22.

So, the refined definition of a symmetric pair for outputting, based on the sample output, is: A pair (X, Y) is considered for output if:

- $X = Y$  (e.g., (20, 20)).
- $X < Y$  AND (Y, X) also exists in the Functions table (e.g., (20, 21) and (21, 20) are both in Functions, so we output (20, 21)).

### SQL Query Strategy:

1. Handle the  $X = Y$  case directly.
2. Handle the  $X \neq Y$  case by self-joining the Functions table to find pairs (F1.X, F1.Y) where  $F1.X < F1.Y$  and (F1.Y, F1.X) exists as (F2.X, F2.Y).
3. Combine these two sets of results using UNION.
4. Order the final result by X in ascending order.

```
SELECT
    f1.X,
    f1.Y
FROM
    Functions f1
JOIN
    Functions f2 ON f1.X = f2.Y AND f1.Y = f2.X
WHERE
    f1.X < f1.Y -- To get only one of the (X,Y) or (Y,X) symmetric
pairs where X != Y
UNION
SELECT
    X,
    Y
FROM
    Functions
WHERE
    X = Y -- To include pairs where X = Y
ORDER BY
```

X;

Let's test this logic with the sample input:

**1. WHERE X = Y part:**

- (20, 20) -> This will be selected.

**2. Self-join part (f1.X < f1.Y):**

- Consider (20, 21) (f1). Does (21, 20) (f2) exist? Yes. And 20 < 21. So (20, 21) is selected.
- Consider (20, 22) (f1). Does (22, 20) (f2) exist? No. So (20, 22) is NOT selected from this part.
- Consider (22, 23) (f1). Does (23, 22) (f2) exist? Yes. And 22 < 23. So (22, 23) is selected.
- Consider (21, 20) (f1). 21 < 20 is false. So this is not selected. (We already got 20,21).
- Consider (23, 22) (f1). 23 < 22 is false. So this is not selected. (We already got 22,23).

**Combined and Ordered:**

1. (20, 20)
2. (20, 21)
3. (22, 23)

This exactly matches the Sample Output.

Therefore, the SQL query is:

```
SELECT
    f1.X,
    f1.Y
FROM
    Functions f1
JOIN
    Functions f2 ON f1.X = f2.Y AND f1.Y = f2.X
WHERE
    f1.X < f1.Y
UNION
SELECT
    X,
    Y
FROM
    Functions
WHERE
    X = Y
ORDER BY
    X;
```