Meta Asked This Question

Every Candidate Struggles Until They See This

REVERSE JOIN Pattern

"Find all mutual friendships between users"

How do you match A→B with B→A?

One pattern solves it all

📊 Sample Data

friendships table

user_id (int)	friend_id (int)
1	2
2	1
3	4
1	3
5	6
6	5

💡 **Notice: (1,2) and (2,1) are the SAME friendship!**

❓ WHAT

Reverse Join Pattern

Match two rows that represent the same pair but in opposite order

🤔 WHY

SQL treats A-B and B-A as different

Reverse join helps treat them as one logical pair

⚙️ HOW

T1.col1 = T2.col2 AND T1.col2 = T2.col1

It finds the mirror version of a row

🎯 The Formula

Switch columns and match them back

The Pattern:

```
T1.col1 = T2.col2 AND T1.col2 = T2.col1
```

💡 **It finds the same pair but in reverse order**

Normal Row	Reverse Row	Result
user_id = 1, friend_id = 2	user_id = 2, friend_id = 1	These are MUTUAL friends!

✅ The Solution

Finding Mutual Friendships

```sql
SELECT
  f1.user_id,
  f1.friend_id

FROM friendships f1

JOIN friendships f2
  -- The Reverse Join Pattern!
  ON f1.user_id = f2.friend_id
 AND f1.friend_id = f2.user_id

 -- Remove duplicates (keep only 1 < 2, not 2 < 1)
WHERE f1.user_id < f1.friend_id;
```

🔄 **Reverse Join = Mutual Check**

🎯 **WHERE removes duplicates**

📈 Expected Output

Mutual friendships only

user_id	friend_id
1	2
5	6

←

→

User 1 ↔ User 2
Both rows exist: (1,2) and (2,1)

MUTUAL ✅

User 5 ↔ User 6
Both rows exist: (5,6) and (6,5)

MUTUAL ✅

User 3 ↔ User 4
Only (3,4) exists, no (4,3)

NOT MUTUAL ❌

📍 Where This Pattern Appears

⚽

Home/Away Matches

Team A vs B + Team B vs A

💰

Transaction Pairs

Sender→Receiver + Receiver→Sender

🤝

User Friendships

user1–user2 + user2–user1

📨

Message Threads

A sends to B + B replies to A

🔄

Two-Way Interactions

Any A→B and B→A relationship

🎯

Bidirectional Flows

Import/Export, Give/Receive

🔥 3 Common Real Examples

⚽ Football Matches

Team A vs Team B

- Home: A vs B
- Away: B vs A

💡 Pair home and away legs to calculate total goals

🤝 Friendships

Social Network

- user1 = A, user2 = B
- user1 = B, user2 = A

💡 Detect mutual connections between users

💸 Transactions

Payment System

- Sender = A, Receiver = B
- Sender = B, Receiver = A

💡 Check if money was returned or reversed

⭐ The Value

✨ **Makes logic cleaner**
No complex CASE statements needed

🎯 **Avoids missing pairs**
Catches both A→B and B→A

🚀 **Interview gold**
Meta, Google, Amazon ask this

💡 **Real-world power**
Used in production systems everywhere

🚀 One pattern. Infinite applications.