EXISTS

1日目	2日目	3日目	4日目	5日目	6日目	7日目
8日目	9日目	10日目	11日目	12日目	13日目	14日目
15日目	16日目	17日目	18日目	19日目	20日目	21日目

EXISTS

EXISTSとは

ほかのテーブルに**値の存在する行のみ抽出する**SQL。
サブクエリ内でメインクエリの表や列を利用する**相関副問い合わせ**の1つ

```
SELECT
    *
FROM
    a_table
WHERE
    [NOT] EXISTS(subquery);
```

主クエリ
(SELECT * FROM a_table)

サブクエリ(EXISTS)

サブクエリの中でメインクエリの結果を利用

EXISTSの構文

EXISTSでは、EXISTSの後のサブクエリが**何らかの値を返すレコードだけをとりだす**（NOT EXISTSの場合は、**値を返さないレコード**だけを返す）

```
SELECT
    *
FROM
    customers AS ct
WHERE
    EXISTS(
        SELECT
            *
        FROM
            orders AS or
        WHERE
            ct.customer_id = or.customer_id
    );
```

サブクエリの返り値が存在するレコードだけ返す（*でなくてカラム名でも1でもよい）

ct(customers)のcustomer_idを用いて、orのレコードを絞り込む

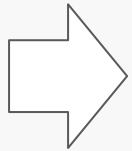
```
SELECT
  *
FROM
    customers AS ct
WHERE
    EXISTS(
        SELECT
            *
        FROM
            orders AS or
        WHERE
            ct.customer_id = or.customer_id );
```

customersテーブル

customer_id
1
5
8
10

ordersテーブル

customer_id
3
4
5
8

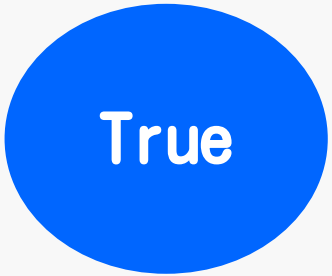


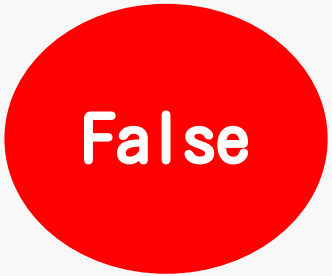
customersテーブルから
customer_idが5, 8のものだけを
取り出す
(1, 10はordersテーブルに存在
しない)

EXISTSとNULL

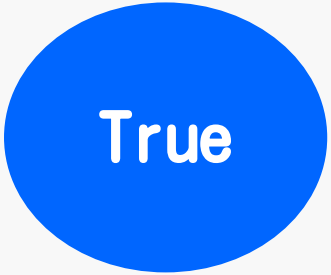
3値論理について

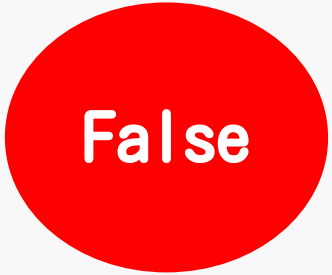
一般的な論理体系では、真か偽か2の論理体系で表される。





DBでは、真か偽が以外に、不明(unknown)であることを表すNULLという値が存在する。この3値の論理値で表す体系を3値論理という。



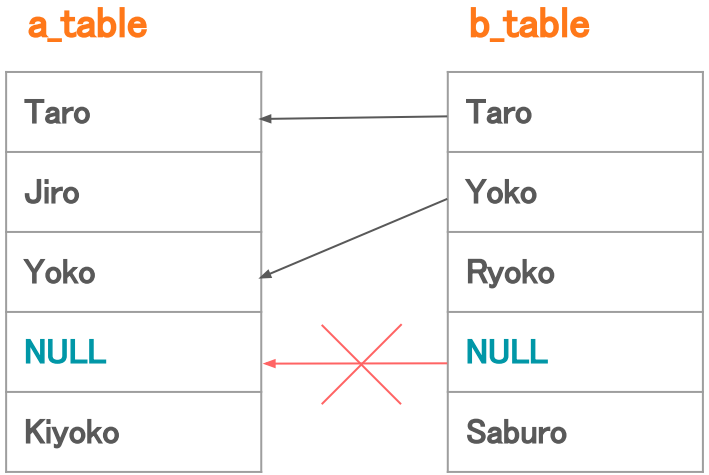


EXISTSで利用されるNULLについて

NULLは、EXISTS内で使用する場合にも複雑な挙動をするので注意が必要
(**特にNOT EXISTSの場合**)

EXISTS句では、サブクエリで **値が返される**レコードのみを取得する
(NULLのものは返されない)

```
SELECT * FROM b_table  WHERE EXISTS(SELECT * FROM a_table WHERE b_table.name = a_table.name) ;
```

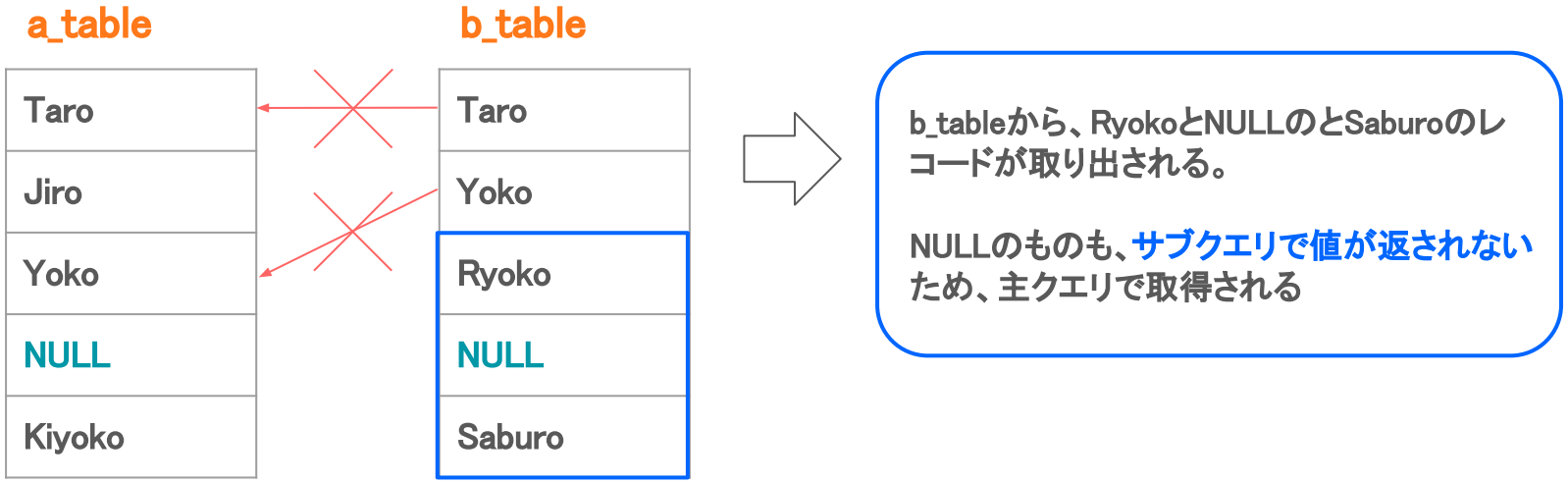


b_tableから、TaroとYokoのレコードが取り出される。
NULLは、SELECT * FROM b_table WHERE NULL=NULLで**値が返されない**ため、取得されない

SELECT * FROM b_table WHERE name IN (SELECT name FROM a_table)と**同じ結果になる**

NOT EXISTS句では、サブクエリで **値が返されない**レコードを取得する
（NULLのものも返される）

```
SELECT * FROM b_table  WHERE NOT EXISTS(SELECT * FROM a_table WHERE b_table.name = a_table.name);
```

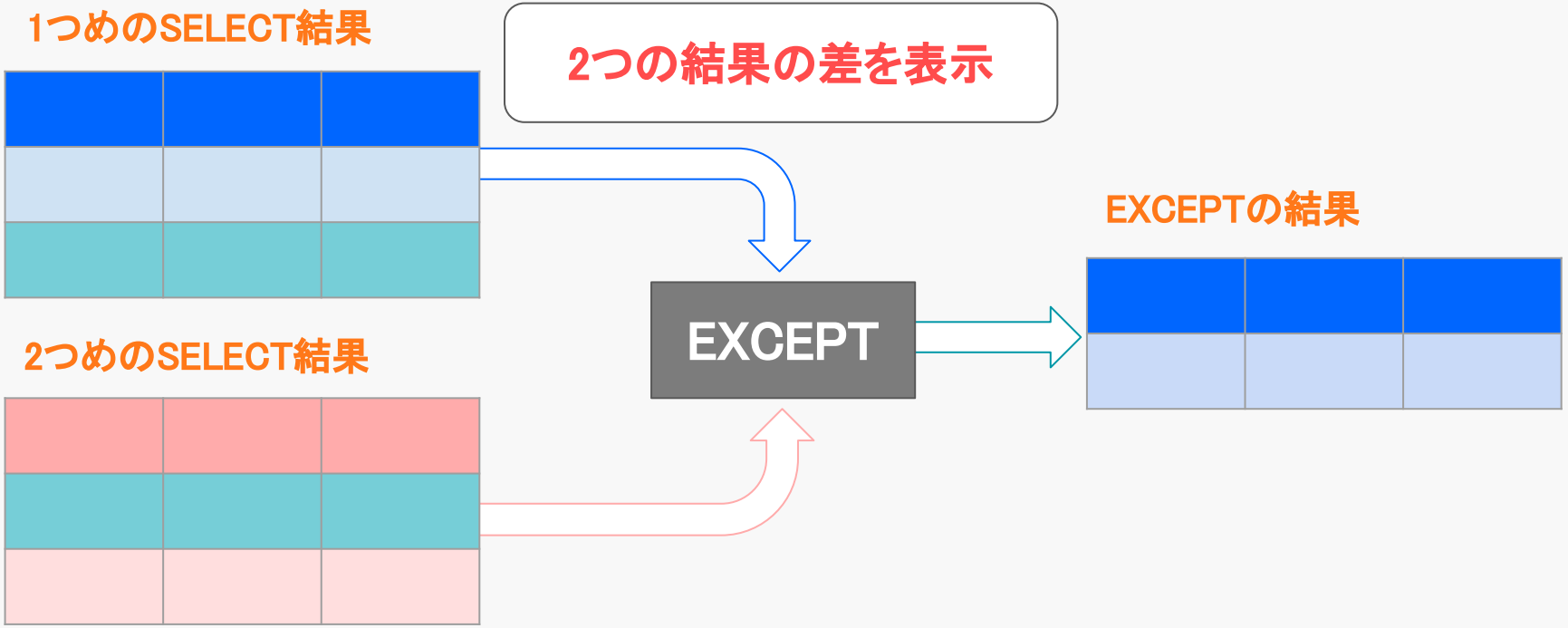


SELECT * FROM b_table WHERE name NOT IN (SELECT name FROM a_table)とは、**結果が異なる**（NOT INは比較の結果、**偽になる**レコードを取り出す。NULLはNOT INの結果、**不明となり偽とならないため**取り出されず、RyokoとSaburoだけが取り出される）

EXISTSでEXCEPTとINTERSECTを実現する

EXCEPT(MINUS)

ある集合と別の集合の差を求めるSQL文
SQL1とSQL2の結果を比較して、SQL1の結果のうちSQL2の結果に存在するものを差し引く



EXCEPT(MINUS)をEXISTSで記述する

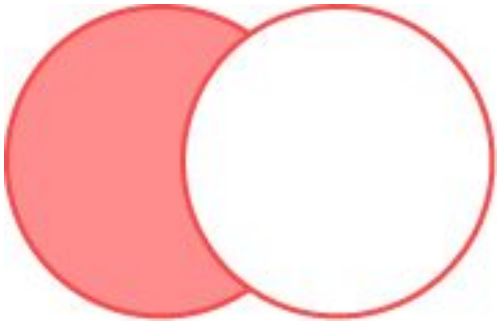
MySQL(ver 8)では、EXCEPT文が存在しないためNOT EXISTSで実装する

NOT EXISTS句を用いて各カラムを比較し、特定のテーブルから別のテーブルに存在するレコードを取り除いて取得する

・EXCEPT(MINUS)を実現する

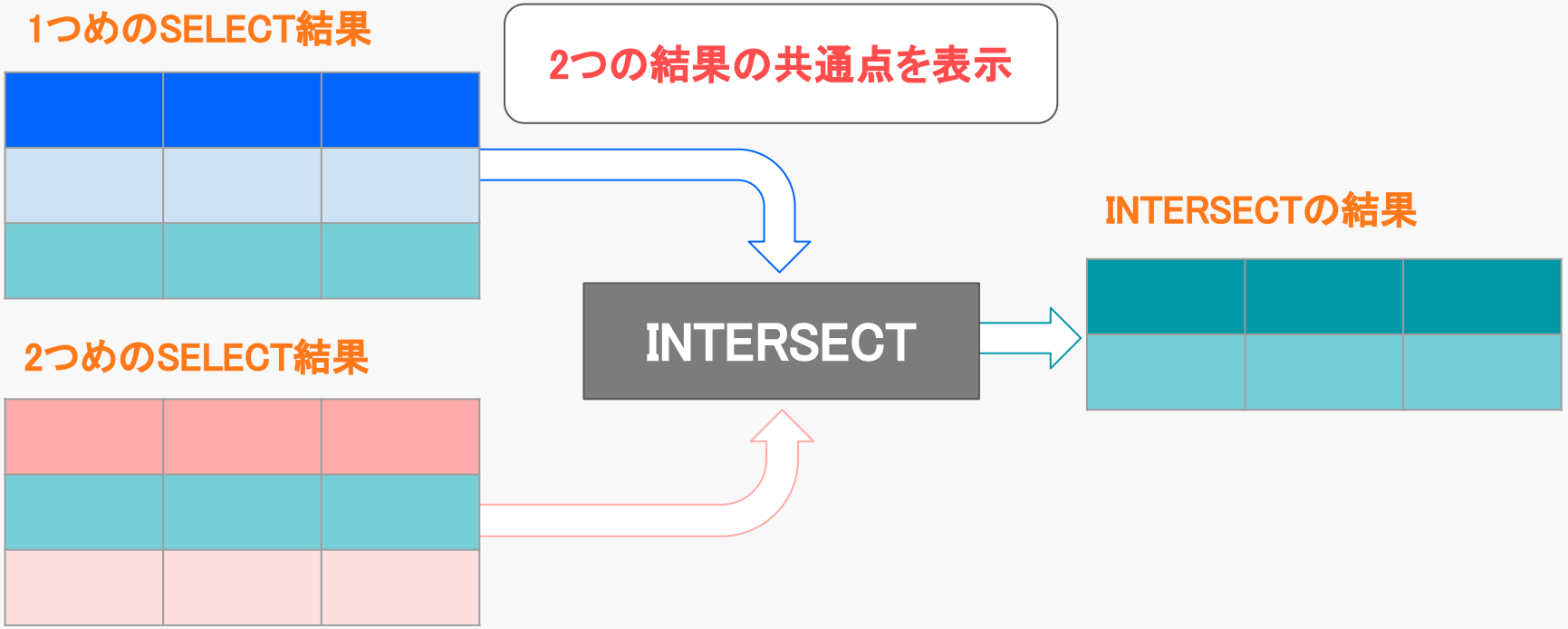
```
SELECT * FROM b_table  WHERE
NOT EXISTS(
SELECT * FROM a_table
WHERE
a_table.column_1 = b_table.column_1 AND
a_table.column_2 = b_table.column_2 AND
:
);
```

EXCEPT*)
(MINUS)



INTERSECT

ある集合と別の集合の積集合を求めるSQL文
SQL1とSQL2の結果を比較して、2つの結果に共通する行を表示する

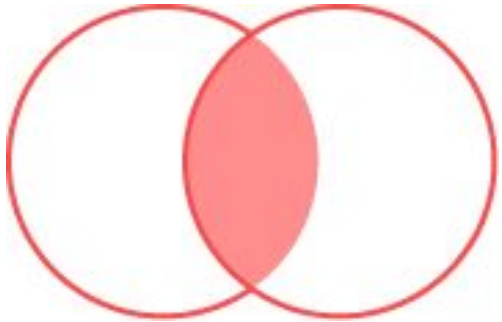


INTERSECTをEXISTSで記述する

MySQL(ver 8)では、INTERSECT文が存在しないためEXISTSで実装する

EXISTS句を用いて各カラムを比較して、特定のテーブルから別のテーブルに存在するレコードのみ取得する。この際、NULLを含むカラムには注意が必要

INTERSECT*)



・INTERSECTを実現する

```
SELECT * FROM b_table  WHERE
EXISTS(
SELECT * FROM a_table
WHERE
a_table.column_1 = b_table.column_1 AND
(a_table.column_2 = b_table.column_2 OR
(a_table.column_2 IS NULL AND b_table.column_2 IS NULL
) AND
:
);
```