

NOTE: Difference Between **EXISTS** and **IN** in SQL (Easy Explanation)

★ 1. What does **EXISTS** do?

EXISTS checks only one thing:

“Does the subquery return at least one row?”

It does **not care** about:

- how many columns the subquery returns
- what values the columns have
- what the SELECT list contains

✓ Example (works even with multiple columns):

```
WHERE EXISTS (  
    SELECT b.roomnumber, COUNT(*) AS c  
    FROM bookings b  
    GROUP BY b.roomnumber  
    HAVING COUNT(*) > 5  
);
```

Why it works:

Because **EXISTS** only checks if a row exists — not what the row looks like.

★ 2. What does **IN** do?

IN compares a value to a **list of values**.

So the subquery must return **exactly one column**.

✗ This will give an error:

```
WHERE r.roomnumber IN (
```

```
SELECT b.roomnumber, COUNT(*) AS c
FROM bookings b
GROUP BY b.roomnumber
);
```

Reason:

Subquery returns **two columns**, and **IN** expects **one**.

★ 3. Why **EXISTS** works with multiple columns but **IN** doesn't

✓ **EXISTS:**

- Only checks whether a row exists
- Doesn't use the returned columns
- Works with 1 column, 2 columns, or 10 columns

✗ **IN:**

- Requires the subquery to return **one single column**
 - Used for list comparisons
 - Will throw “**Subquery returns more than 1 column**” error if you return multiple columns
-

★ 4. When to use what

Use Case	Use EXISTS	Use IN
You only need to check if related data exists	✓ Best choice	✗ Not ideal
Subquery has multiple columns	✓ Works	✗ Error
Subquery returns exactly one column	✓ Works	✓ Works

You need to compare actual values

✗ No

✓ Yes

★ 5. Your Example Explanation (Simple)

Your **EXISTS** query:

```
UPDATE rooms r
SET r.status = 'High Demand'
WHERE EXISTS (
    SELECT b.roomnumber, COUNT(*) AS c
    FROM bookings b
    WHERE b.roomnumber = r.roomnumber
    GROUP BY b.roomnumber
    HAVING COUNT(*) > 5
);
```

Why it works:

- **EXISTS** does **not** care that two columns are selected
- The subquery is correlated (**b.roomnumber = r.roomnumber**)
- It returns at least one row if bookings > 5
- Therefore the room gets updated

Why the same subquery would fail in **IN**:

Because it selects **two columns**, and **IN** requires **only one**.

Short Memory Trick

EXISTS = checks rows

IN = checks values

Tuple IN (Very Short Explanation)

`(col1, col2) IN (SELECT col1, col2 ...)`

means you are comparing a pair of columns together.

Both sides must return the same number of columns, and the row is selected only if the entire pair matches.

Memory line:

➡ Tuple IN = match whole pairs (or groups) of columns.

query

`SELECT roomtype, roomrate`

`FROM rooms`

`WHERE (roomnumber, roomtype) IN (`

`SELECT roomnumber, roomtype`

`FROM rooms`

`WHERE roomrate < 200`

`);`