JFS-24 — Join Filter Safety Standard

24-Cell Operational Safety Architecture

Version 1.0 — 2025 Author: Usman Zafar

1. Purpose

JFS-24 defines the **operational semantics** of filters in relational joins. It establishes the rules governing ON clause matching, WHERE clause elimination, safe vs unsafe filter placement, and the 24-cell join filter safety matrix.

This standard ensures that SQL joins remain:

- semantically correct
- structurally preserved
- free from silent collapse
- operationally predictable

JFS-24 is the operational counterpart to **SFS-24**, which defines structural semantics.

2. Scope

JFS-24 applies to:

- analytical SQL
- ETL/ELT pipelines
- BI semantic layers
- dimensional modeling
- fact table construction
- SQL governance and code review

It governs filter behavior in:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN
- SELF JOIN

3. Execution Phase Semantics

3.1 ON Clause (Matching Phase)

The ON clause defines **matching**, not elimination.

- Executes **before** row preservation
- Determines which rows match
- Does **not** eliminate NULL extended rows
- Safe for filters on either side (with exceptions for CROSS JOIN)

3.2 WHERE Clause (Elimination Phase)

The WHERE clause defines **elimination**, not matching.

- Executes **after** row preservation
- Eliminates rows that fail the condition
- Eliminates NULL extended rows
- Can collapse outer joins

3.3 Fundamental Law of JFS-24

WHERE kills NULLs. ON controls matching.

This distinction is the foundation of filter safety.

4. Filter Side Semantics

4.1 Preserved Side

Filtering the preserved side is **safe**.

Examples:

- LEFT JOIN → filter LEFT side
- RIGHT JOIN → filter RIGHT side
- FULL JOIN → filter both sides (with caution)

4.2 NULL Producing Side

Filtering the NULL producing side in the WHERE clause is **unsafe**.

Examples:

- LEFT JOIN → filtering RIGHT side in WHERE collapses the join
- RIGHT JOIN → filtering LEFT side in WHERE collapses the join
- FULL JOIN → filtering either side in WHERE collapses that side

4.3 ON Clause Filters

ON clause filters are generally **safe**, because they occur before preservation.

Exceptions:

- CROSS JOIN + ON filter → **invalid**
- FULL JOIN + ON filter → **safe but must be intentional**

5. Unsafe Join Filter Patterns

JFS-24 identifies **six collapse patterns** that must be prohibited.

5.1 LEFT JOIN Collapse

Code
```
LEFT JOIN … WHERE right_column = …
```

5.2 RIGHT JOIN Collapse

Code
```
RIGHT JOIN … WHERE left_column = …
```

5.3 FULL JOIN Collapse (Left Side)

Code
```
FULL JOIN … WHERE left_column = …
```

5.4 FULL JOIN Collapse (Right Side)

Code
```
FULL JOIN … WHERE right_column = …
```

5.5 CROSS JOIN with ON (Left Side)

Code
```
CROSS JOIN … ON left_column = …
```

5.6 CROSS JOIN with ON (Right Side)

Code
```
CROSS JOIN … ON right_column = …
```

These patterns violate join semantics and must be rejected in code review.

6. The 24-Cell Join-Filter Safety Matrix

JFS-24 classifies join-side × filter-side × phase interactions into:

- **SAFE**
- **UNSAFE (Collapse)**
- **INVALID**
- **NEUTRAL**

```
JOIN TYPE | LEFT WHERE | LEFT ON | RIGHT WHERE | RIGHT ON
-------------------------------------------------------
INNER     | SAFE       | SAFE    | SAFE        | SAFE
LEFT      | SAFE       | SAFE    | COLLAPSE    | SAFE
RIGHT     | COLLAPSE   | SAFE    | SAFE        | SAFE
FULL      | COLLAPSE   | SAFE    | COLLAPSE    | SAFE
CROSS     | SAFE       | INVALID | SAFE        | INVALID
SELF      | SAFE       | SAFE    | SAFE        | SAFE
```

This matrix is the **operational topology** of join filter interactions.

7. Filter Type Semantics

7.1 Equality Filters (=)

- Safe in ON
- Unsafe in WHERE on NULL producing side

7.2 Inequality Filters (>, <, >=, <=)

- Safe in ON
- Unsafe in WHERE on NULL producing side

7.3 NULL Sensitive Filters

- `IS NOT NULL` → **always unsafe** on NULL producing side
- `IS NULL` → safe in ON, unsafe in WHERE

7.4 Non Deterministic Filters

- Must not be used in ON
- Allowed in WHERE only on preserved side

8. Operational Doctrine

JFS-24 establishes the following doctrine:

- Filters must preserve join contracts
- WHERE must not eliminate NULL extended rows
- ON must not be used to simulate INNER JOIN behavior
- CROSS JOIN must not contain ON conditions

- Filter placement must be intentional and documented

This doctrine elevates filter reasoning to the level of **enterprise governance**.

9. Compliance Requirements

To comply with JFS-24:

- Do not filter the NULL producing side in WHERE
- Do not introduce ON conditions into CROSS JOINs
- Do not rely on implicit join behavior
- Validate all filters against the 24-cell matrix
- Document filter intent in design artifacts

10. Relationship to SFS-24

SFS-24 defines **what the join must structurally preserve**. JFS-24 defines **how filters must behave to preserve it**.

Together, they form the **Enterprise Join Semantics Architecture Suite**.

11. Versioning

JFS-24 follows semantic versioning:

- **Major** — changes to collapse rules or matrix
- **Minor** — new filter classifications
- **Patch** — clarifications or examples

12. Copyright