Fighting Bad PL/SQL & SQL with Custom Rules

Philipp Salvisberg
2 October 2026



WARNING

This presentation contains explicit SQL and PL/SQL,
strong opinions on code quality,
and prolonged scenes of Java development.

Viewers will be exposed to Maven, ANTLR parse trees,
and debugging in multiple IDEs.

Some rules may conflict with deeply held coding beliefs.
Side effects may include raised eyebrows,
involuntary refactoring, and an uncontrollable urge
to automate code reviews.

Viewer discretion is advised.
Your code will receive none.

Welcome



Philipp Salvisberg

Grisselbav Founder, CEO, Consultant, Developer

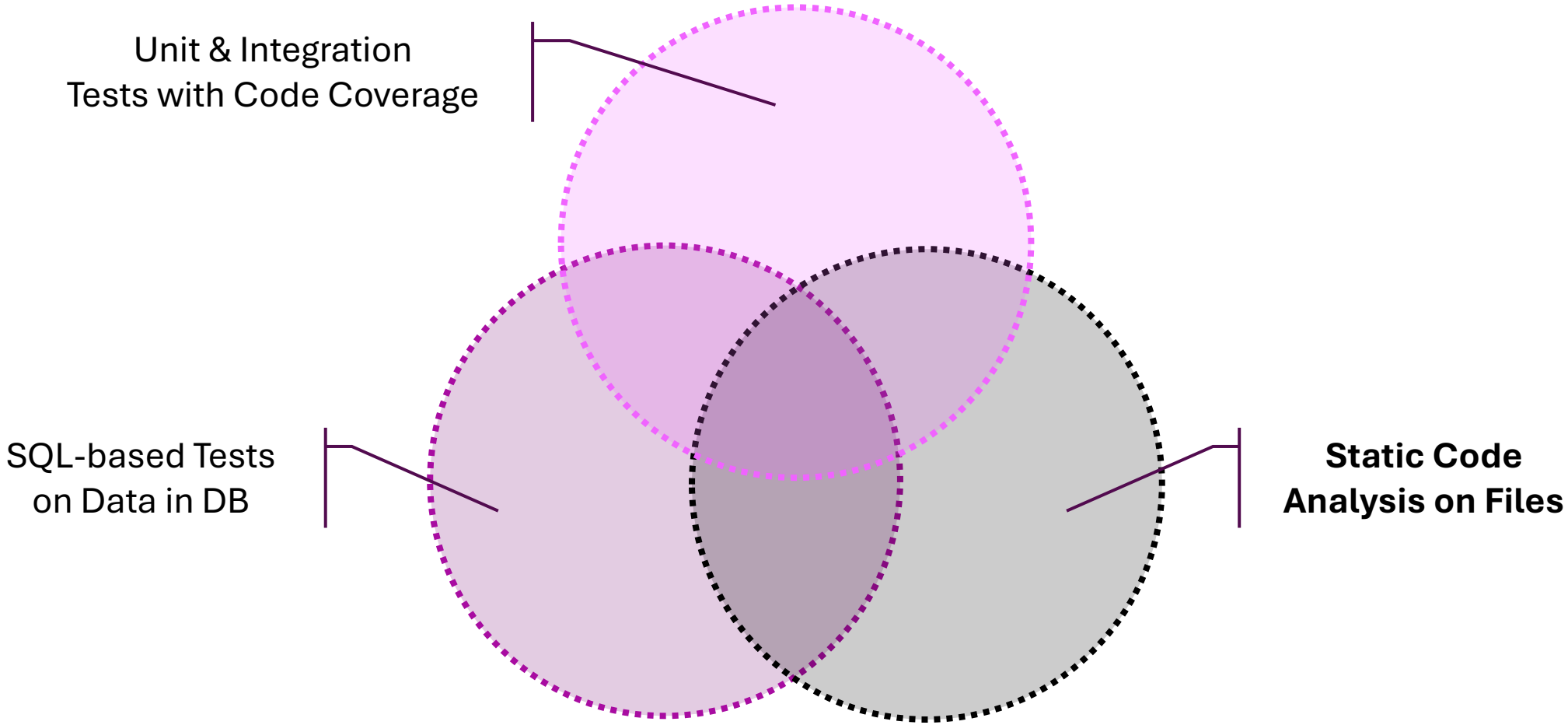
- Database-centric development
- Enforcing high-quality database standards
- Simple & accurate solutions

philipp.salvisberg@grisselbav.com

https://www.salvis.com/blog/

Introduction

Quality Assessment of Code in the DB



Rules – The Basis for Static Code Analysis

Core G-3130 FREE

Try to use ANSI SQL-92 join syntax.

SQL • Check

Rule Details

Severity Level
Major

Quality Area
Maintainability

Fixing times
10 mins
Quick Fix | Local-high

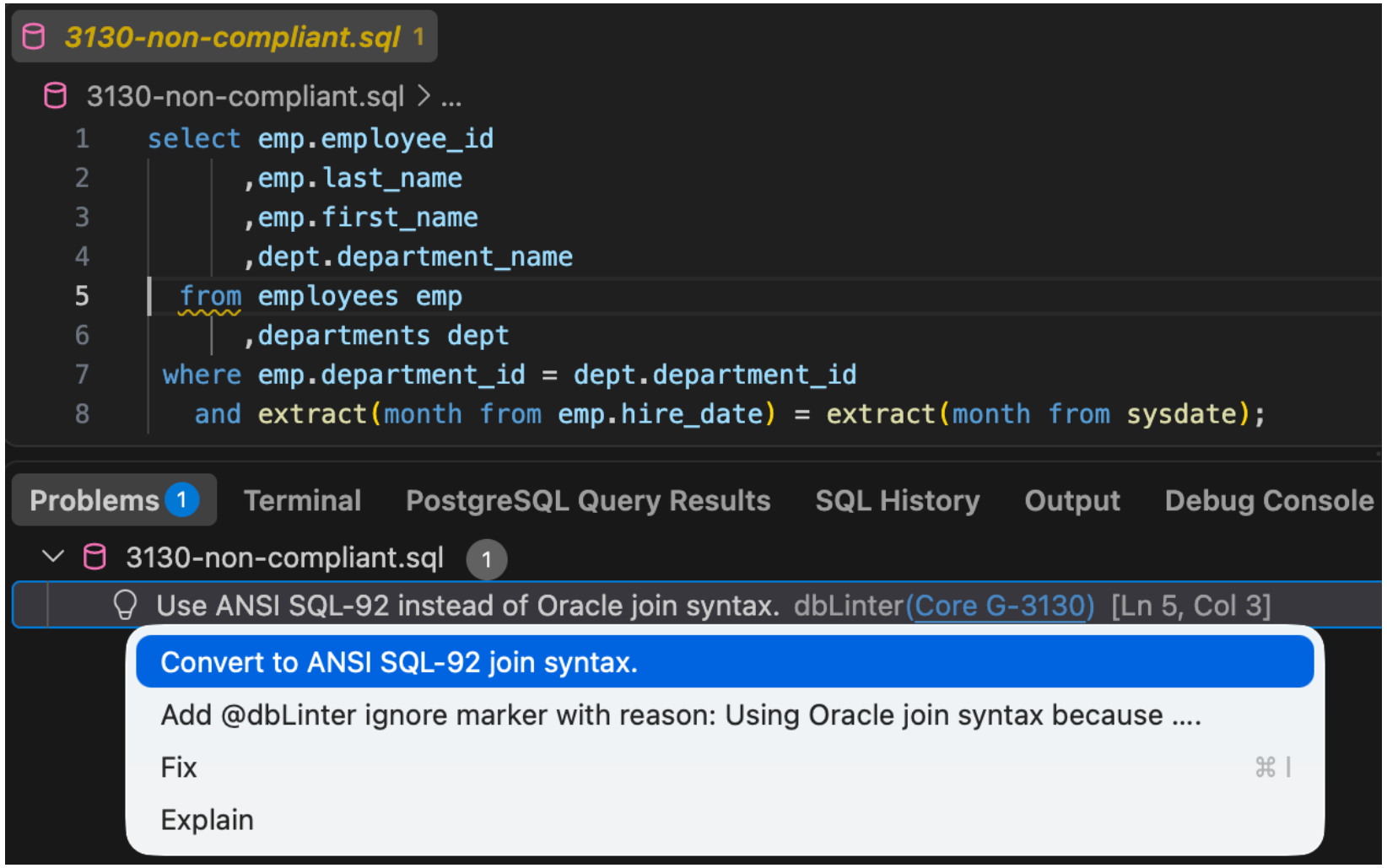
Database Support
OracleDB (v9.0.1 +)
PostgreSQL (v7.0.3 +)

Reason

ANSI SQL-92 join syntax supports the full outer join. A further advantage of the ANSI SQL-92 join syntax is the separation of the join condition from the query filters.

Source: https://dblinter.app/ords/r/dblinter/dblinter-console/rules#P1000_SHOW_RULE=core%20g-3130

Integration: IDE,CLI, SW Quality Platforms



New Core Or Custom Rule?

Core Rules

- Avoid defining variables that are not used
- Never reference an unknown table/alias in hints
- Never use ROWNUM at the same query level as ORDER BY
- Try to use ANSI SQL-92 join syntax

Custom Rules

- Always restrict rows in views using "tenant_id = current_tenant_id()"
- Always guard soft-delete tables by "deleted_at is null", unless ...
- Never use insert, update, delete, merge outside of PL/SQL packages
- Never use ANSI SQL-92 join syntax

Principles for rules: https://grisselbav.github.io/dbLinter/reference/principles/#principles-for-rules

Let's Implement This Custom Rule

Demo R-3131 FREE

Never use ANSI SQL-92 join syntax.

SQL • Check

Rule Details

Severity Level
Blocker

Quality Area
Reliability

Fixing times
10 mins
Local-high

Database Support
OracleDB (v7.3.4 +)

Reason

ANSI SQL-92 join syntax is not supported by all Oracle Database features. For example, SQL Assertions in Oracle AI Database 26ai require join conditions to be specified in the `WHERE` clause.

ANSI SQL-92 joins have also been associated with optimizer bugs in older Oracle Database releases. Avoiding them can therefore improve compatibility when code must support older database versions.

Register Custom Rule

dbLinter

Tenant: Demo | philipp42

Rules | Configuration | Administration

Administration \ Rules \ Edit Rule

Delete | Preview Rule | Save

Show All | Rule | Parameters | References | Example Sets | DBMS Scopes | Tests

Rule

* Rule Name: R-3131

* Rule Title: Never use ANSI SQL-92 join syntax.

Always..., Never..., Avoid..., Try...

* Rule Reason:

ANSI SQL-92 join syntax is not supported by all Oracle Database features. For example, SQL Assertions in Oracle AI Database 26ai require join conditions to be specified in the `WHERE` clause.

ANSI SQL-92 joins have also been associated with optimizer bugs in older Oracle Database releases. Avoiding them can therefore improve compatibility when code must support older database versions.

* Software Quality Area: Reliability

* Severity Level: Blocker

* Severity: Error

* Fixing Time: Local-high

* Profile: Strict

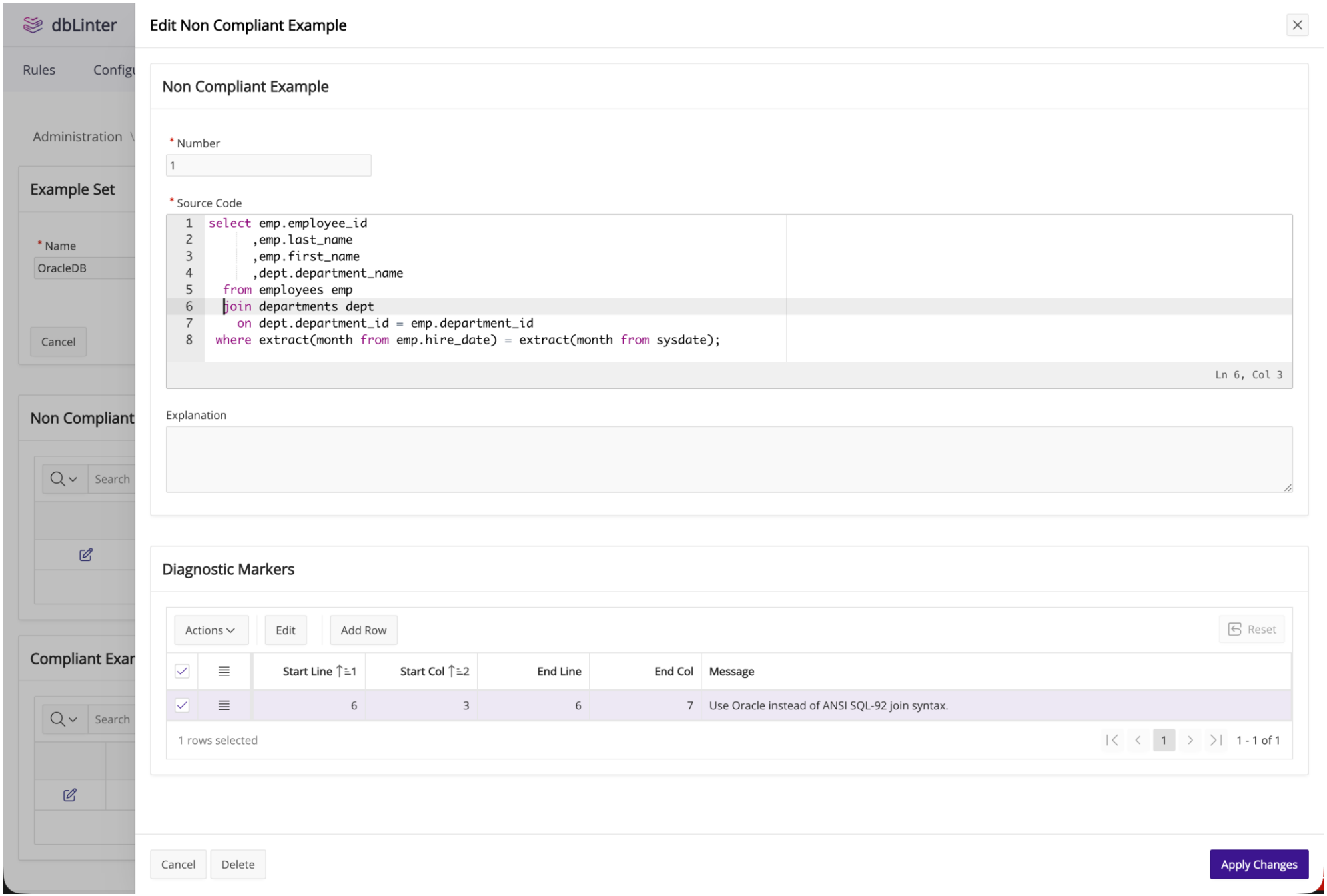
Tags: SQL (Core) 1

Create tag

* Rule Has Check: on

* Rule Has Quick Fix: off

* Rule Check Is Free: on


dbLinter
Rules
Configu
Administration
Example Set
Name
OracleDB
Cancel
Non Compliant
Search
Compliant Exar
Search
Edit Non Compliant Example
Non Compliant Example
Number
1
Source Code
1 select emp.employee_id
2 ,emp.last_name
3 ,emp.first_name
4 ,dept.department_name
5 from employees emp
6 join departments dept
7 on dept.department_id = emp.department_id
8 where extract(month from emp.hire_date) = extract(month from sysdate);
Ln 6, Col 3
Explanation
Diagnostic Markers
Actions
Edit
Add Row
Reset
Start Line
Start Col
End Line
End Col
Message
6
3
6
7
Use Oracle instead of ANSI SQL-92 join syntax.
1 rows selected
1
1 - 1 of 1
Cancel
Delete
Apply Changes

Generate Java Validator Project

Generate Java Validator Project

```
dblinter gen-java \
  --repoUrl=https://api.dblinter.app \
  --tenantName=Demo \
  --userName=philipp.salvisberg+42@gmail.com \
  --accessToken=$DBLINTER_DEMO_ACCESS_TOKEN \
  --configName=Default \
  --indent=4 \
  --outputName=dbLinter-Demo-Custom-Validator \
  --tenantFilter=Demo \
  --ruleFilter=.+ \
  --groupId=com.grisselbav \
  --packageName=com.grisselbav.demo.validator
```

Result of "dblinter gen-java"

```
find dbLinter-Demo-Custom-Validator -type f | sort

dbLinter-Demo-Custom-Validator/.gitignore
dbLinter-Demo-Custom-Validator/.mvn/wrapper/maven-wrapper.properties
dbLinter-Demo-Custom-Validator/mvnw
dbLinter-Demo-Custom-Validator/mvnw.cmd
dbLinter-Demo-Custom-Validator/pom.xml
dbLinter-Demo-Custom-Validator/README.md
dbLinter-Demo-Custom-Validator/src/main/java/com/grisselbav/demo/validator/DemoR3131.java
dbLinter-Demo-Custom-Validator/src/test/java/com/grisselbav/demo/validator/AbstractApxTest.java
dbLinter-Demo-Custom-Validator/src/test/java/com/grisselbav/demo/validator/AbstractJdbcTest.java
dbLinter-Demo-Custom-Validator/src/test/java/com/grisselbav/demo/validator/AbstractTest.java
dbLinter-Demo-Custom-Validator/src/test/java/com/grisselbav/demo/validator/DemoR3131Test.java
dbLinter-Demo-Custom-Validator/src/test/resources/application.properties
```

README.md

dbLinter Custom Validator for Demo

Project Structure

- `src/main/java/com/grisselbav/demo/validator` : A class for each dbLinter rule.
- `src/test/java/com/grisselbav/demo/validator` : A test class for each dbLinter rule.
- `src/test/resources/application.properties` : Optional JDBC configuration for rule checks that may improve results when using read-only database access.

Prerequisites

- JDK 17 or newer.
- Ensure the environment variable `JAVA_HOME` points to that JDK.

The Maven wrapper script (`mvnw`) simplifies the build process, so no Maven installation is required.

Build and Test

Build the JAR with tests:

```
./mvnw clean package
```

Build the JAR without tests:

```
./mvnw clean package -DskipTests=true
```

Run a single test:

```
./mvnw -Dtest=DemoR3131Test test
```

DemoR3131Test.java

```java
public class DemoR3131Test extends AbstractTest {
    @Nested
    public class OracleDB {

        @Test
        public void non_compliant_1() {
            parse("""
                    select emp.employee_id
                         ,emp.last_name
                         ,emp.first_name
                         ,dept.department_name
                      from employees emp
                      join departments dept
                        on dept.department_id = emp.department_id
                     where extract(month from emp.hire_date) = extract(month from sysdate);
                    """);
            check("Demo", "R-3131");
            Assertions.assertEquals(1, issues.size());
            // issue 1
            var issue1 = issues.get(0);
            Assertions.assertEquals("""
                    Use Oracle instead of ANSI SQL-92 join syntax.""", issue1.getMessage());
            Assertions.assertEquals(6, issue1.getRange().getStartLine());
            Assertions.assertEquals(3, issue1.getRange().getStartCol());
            Assertions.assertEquals(6, issue1.getRange().getEndLine());
            Assertions.assertEquals(7, issue1.getRange().getEndCol());
        }
```

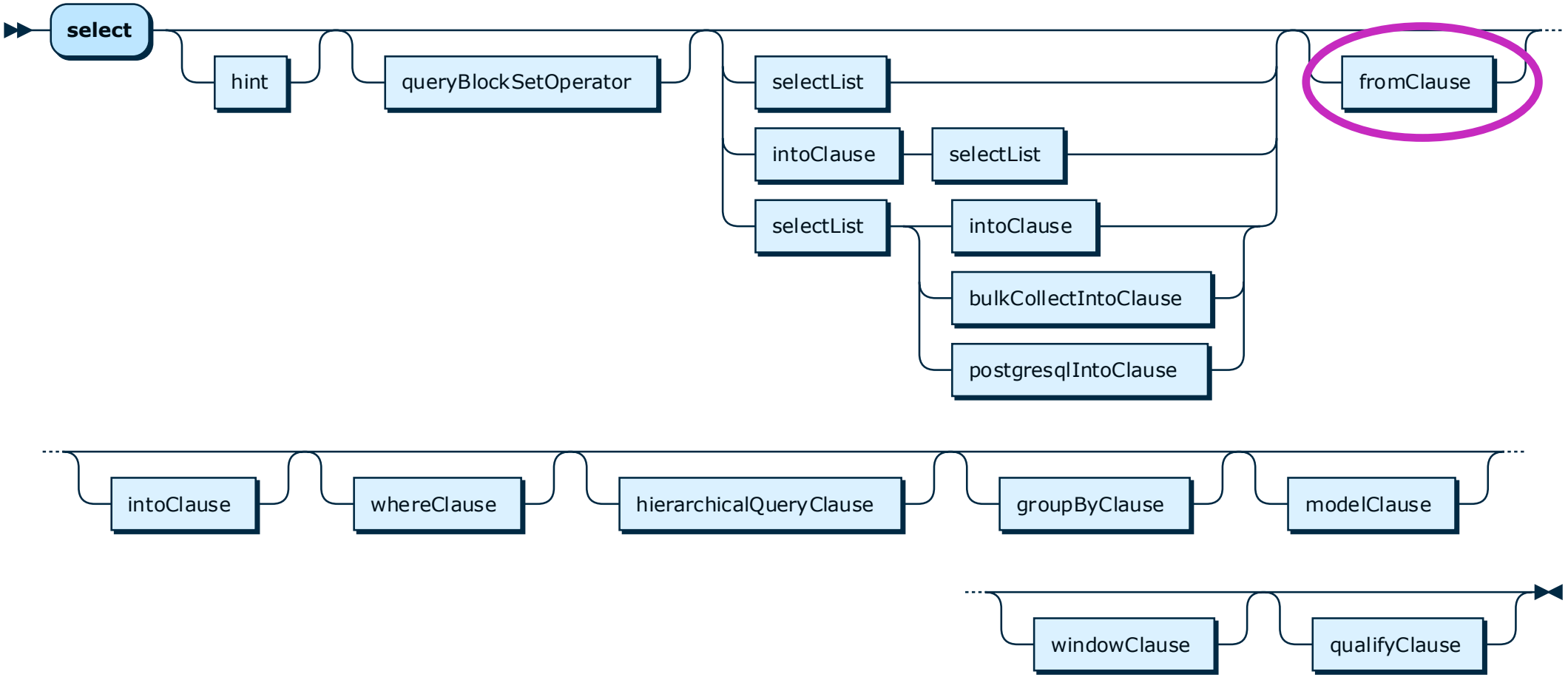
DemoR3131.java



IslandSQL
Grammar & Parse Tree

queryBlock

```
select *
  from emp
  join dept using (deptno);
```

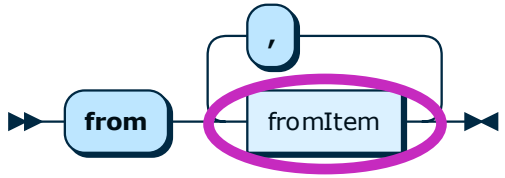


Source: https://islandsql.github.io/IslandSQL/grammar.html#queryBlock

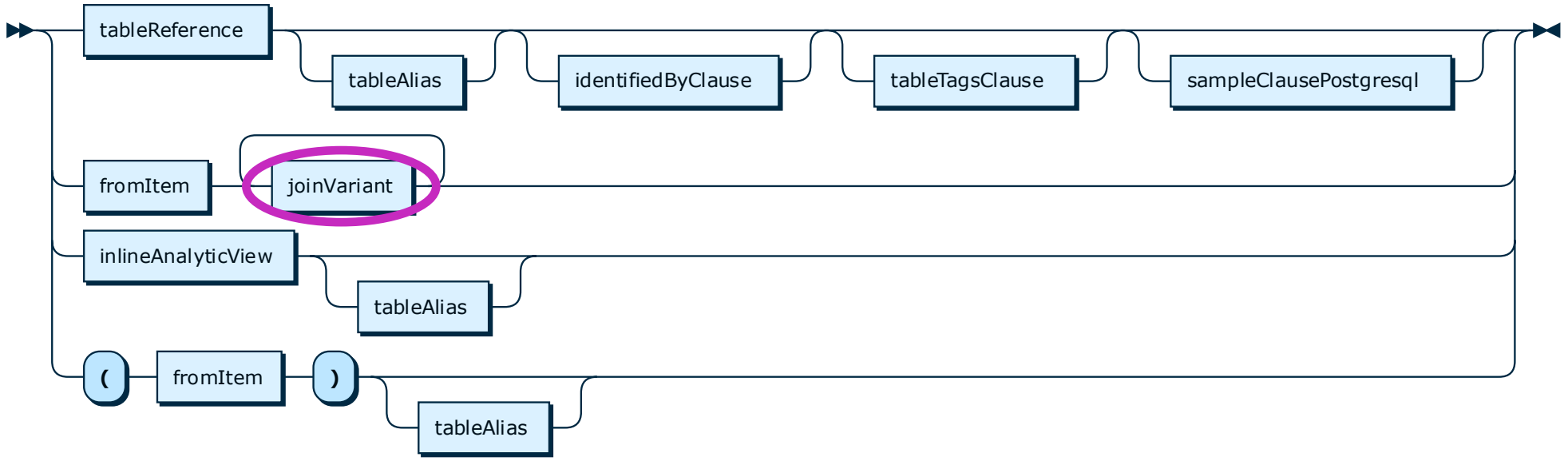
fromClause

```
select *
  from emp
  join dept using (deptno);
```

fromClause:



fromItem:

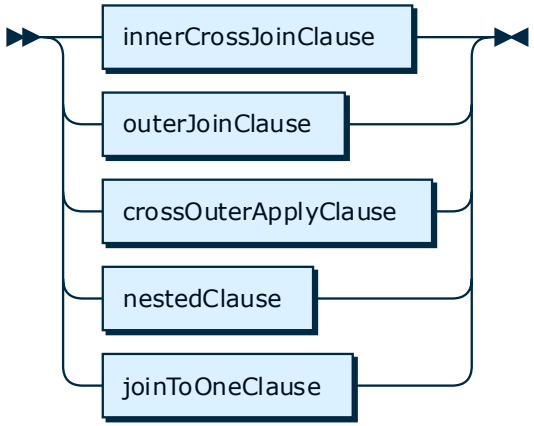


Source: https://islandsql.github.io/IslandSQL/grammar.html#fromClause

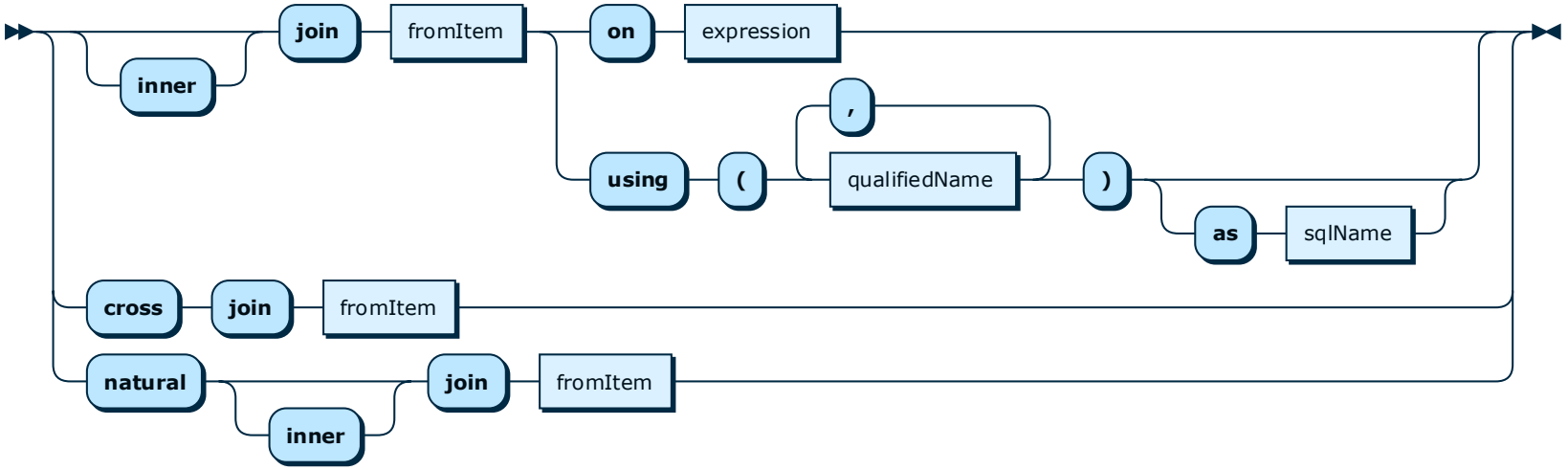
joinVariant

```
select *
  from emp
  join dept using (deptno);
```

joinVariant:



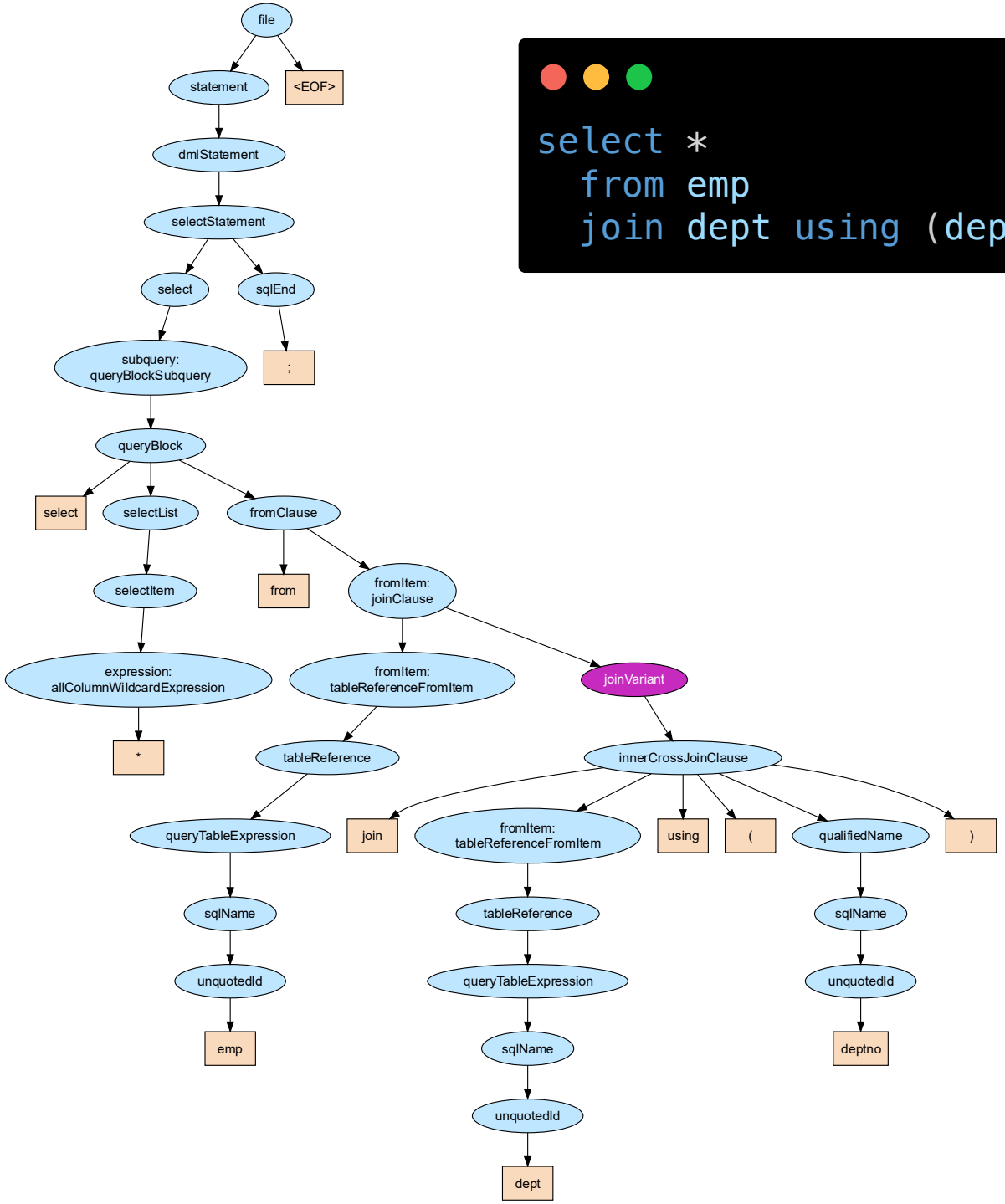
innerCrossJoinClause:



Source: https://islandsql.github.io/IslandSQL/grammar.html#joinVariant

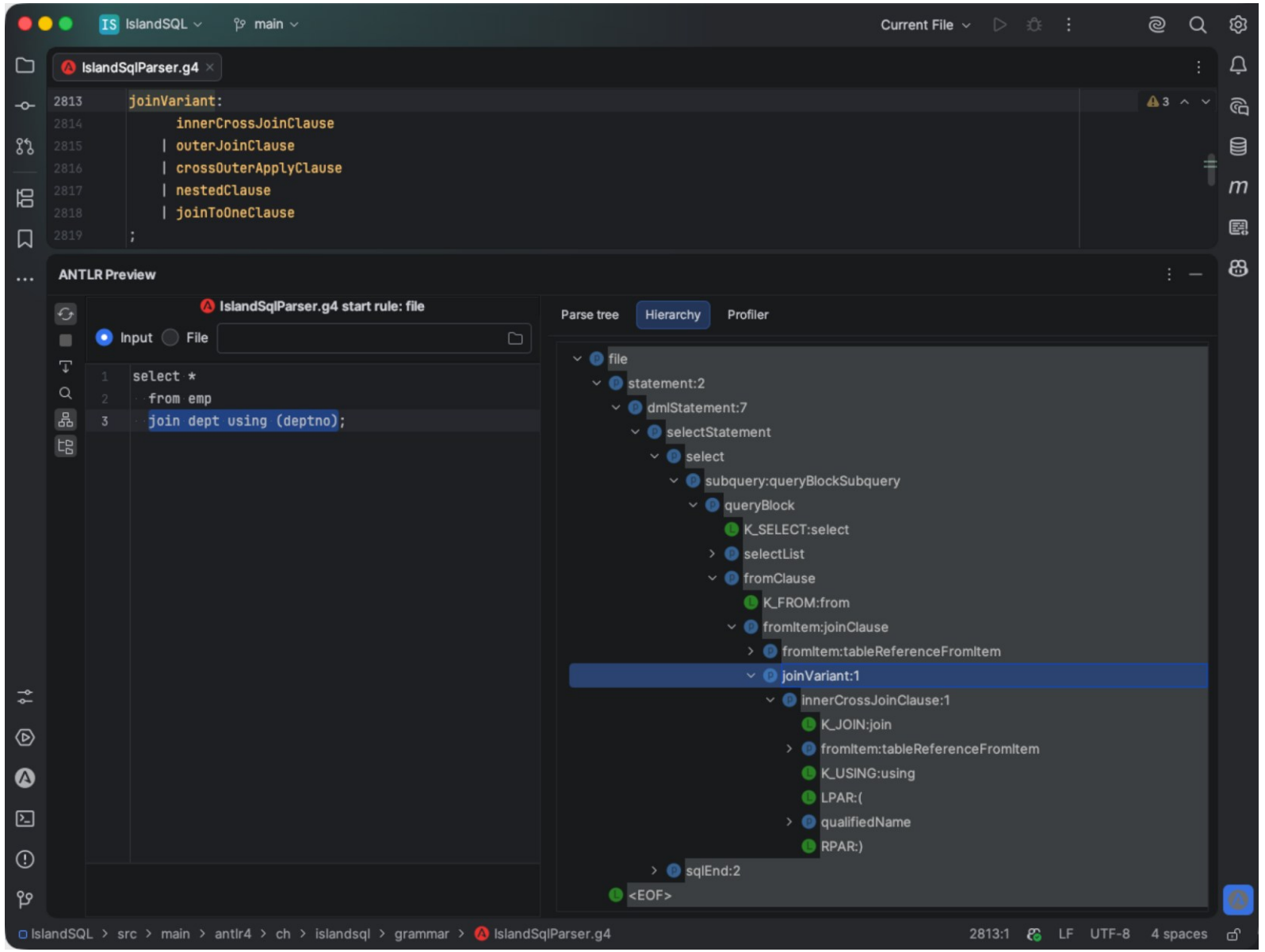
Parse Tree

```
file
  statement
    dmlStatement
      selectStatement
        select
          subquery:queryBlockSubquery
            queryBlock
              K_SELECT:select
              selectList
                selectItem
                  expression:allColumnWildcardExpression
                    AST:*
              fromClause
                K_FROM:from
                fromItem:joinClause
                  fromItem:tableReferenceFromItem
                    tableReference
                      queryTableExpression
                        sqlName
                          unquotedId
                            ID:emp
                  joinVariant
                    innerCrossJoinClause
                      K_JOIN:join
                      fromItem:tableReferenceFromItem
                        tableReference
                          queryTableExpression
                            sqlName
                              unquotedId
                                ID:dept
                      K_USING:using
                      LPAR:(
                      qualifiedName
                        sqlName
                          unquotedId
                            ID:deptno
                      RPAR:)
        sqlEnd
          SEMI:;
  <EOF>
```




```
select *
  from emp
  join dept using (deptno);
```

IntelliJ IDEA ANTLR4 Plugin



Complete Project and Test (Demo)

https://github.com/Grisselbav/dbLinter-Demo-Custom-Validator

Register Issue

DemoR3131.java

```java
/**
 * Demo R-3131: Never use ANSI SQL-92 join syntax.
 */
public class DemoR3131 extends AbstractCheck {
    @Check(tenant = "Demo", rule = "R-3131")
    public List<CheckIssue> checkPlsqlStatement(IslandSqlParser.JoinVariantContext ctx) {
        addIssue()
                .setMessage("Use Oracle instead of ANSI SQL-92 join syntax.")
                .setRange(ctx.start);
        return checkIssues;
    }
}
```

Register Issue With Quick Fixes

DemoR3131.java

```java
/**
 * Demo R-3131: Never use ANSI SQL-92 join syntax.
 */
public class DemoR3131 extends AbstractCheck {
    @Check(tenant = "Demo", rule = "R-3131")
    public List<CheckIssue> checkPlsqlStatement(IslandSqlParser.JoinVariantContext ctx) {
        addIssue()
                .setMessage("Use Oracle instead of ANSI SQL-92 join syntax.")
                .setRange(ctx.start)
                .addQuickFix("Convert to Oracle join syntax", () ->
                        List.of(new Replacement(new Range(ctx),
                                ctx.start.getText() +
                                        " /* TODO: convert to Oracle join */")), true)
                .addDbLinterIgnoreQuickFix(ctx.start, "Keep ANSI SQL-92 join because ...")
                .addFalsePositiveQuickFix(ctx.start);
        return checkIssues;
    }
}
```

Build Project

```
./mvnw clean package

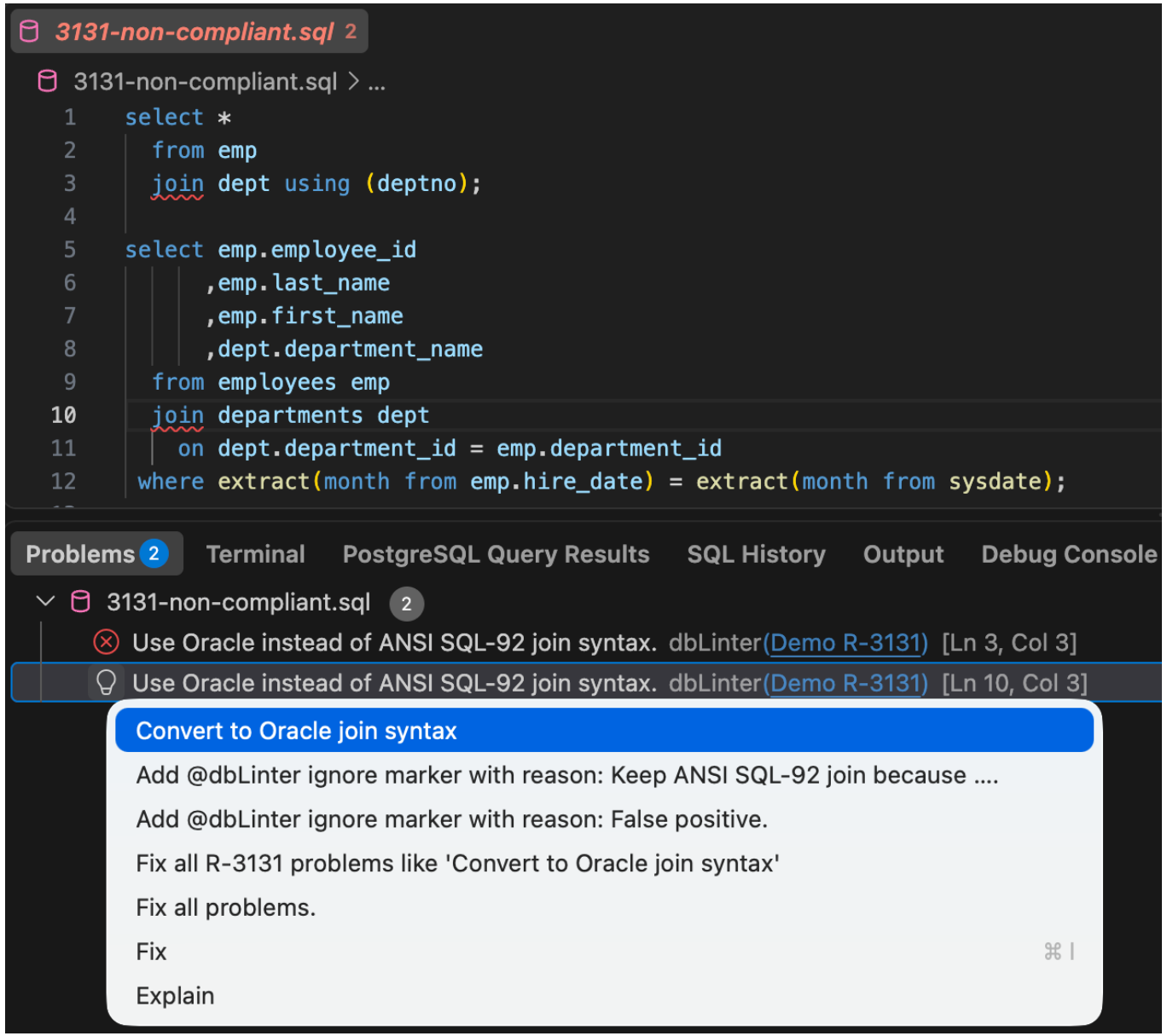
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------< com.grisselbav:custom-validator >--------------------
[INFO] Building dbLinter Validator 1.0.0-SNAPSHOT
[INFO] from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] ...
[INFO] --- surefire:3.5.4:test (default-test) @ custom-validator ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -------------------------------------------------------
[INFO] T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.grisselbav.demo.validator.DemoR3131Test
[INFO] Running com.grisselbav.demo.validator.DemoR3131Test$OracleDB
[INFO] Running com.grisselbav.demo.validator.DemoR3131Test$OracleDB$Solution
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.051 s -- in com.grisselbav.demo.validator.DemoR3131Test$OracleDB$Solution
[INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.781 s -- in com.grisselbav.demo.validator.DemoR3131Test$OracleDB
[INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.787 s -- in com.grisselbav.demo.validator.DemoR3131Test
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.5.0:jar (default-jar) @ custom-validator ---
[INFO] Building jar: dbLinter-Demo-Custom-Validator/target/custom-validator-1.0.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.420 s
[INFO] ------------------------------------------------------------------------
```

Publish Validator

```
./mvnw deploy

[INFO] Scanning for projects...
[INFO]
[INFO] -------------------< com.grisselbav:custom-validator >-------------------
[INFO] Building dbLinter Validator 1.0.0-SNAPSHOT
[INFO] from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] ...
[INFO]
[INFO] --- exec:3.6.3:exec (import-validator) @ custom-validator ---
Importing validator ...
Validator imported successfully from dbLinter-Demo-Custom-Validator/target/custom-validator-1.0.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.543 s
[INFO] ------------------------------------------------------------------------
```

Test in VS Code



Core Messages

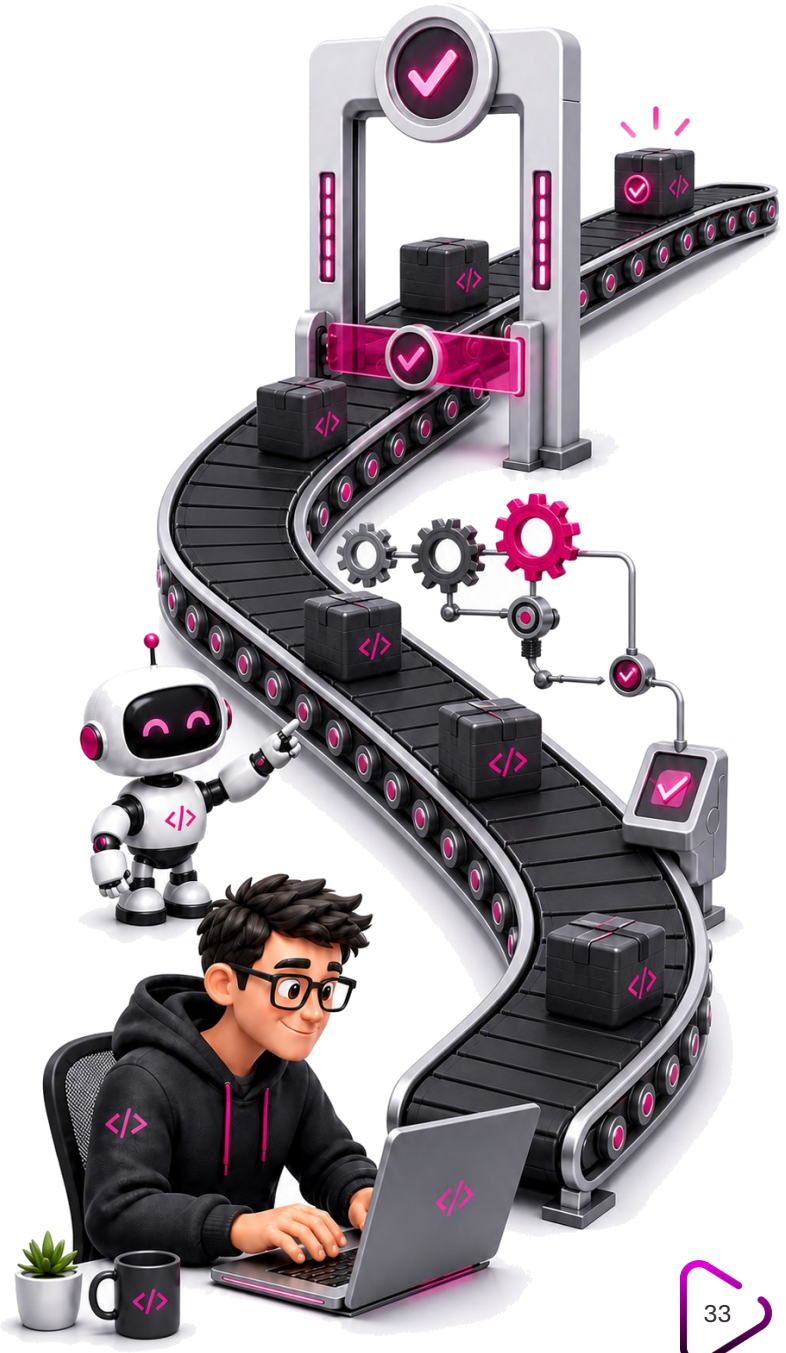
The Right Home for Every Rule

- Propose broadly useful rules in the dbLinter GitHub repository
- Suggest improvements to existing rules
- Define custom rules for project-specific or conflicting requirements



Make Your Rules Executable

- Give developers feedback while they are writing code
- Apply the same standard in the IDE and build pipelines
- Enforce those rules through quality gates
- Implementing custom rules is easier than you might think



Thank you!

