NESTED LOOPS pseudo-code *(high-level, simplified perspective)*

NESTED LOOPS
 CHILD_ROW_SOURCE_1 ← driving row source (or "outer" row source) alias: r_1 columns: $(c_1, c_2, \dots, c_n)$
 CHILD_ROW_SOURCE_2 ← inner row source (or "inner" row source) alias: r_2 columns: $(c_1, c_2, \dots, c_m)$

with join conditions defined on columns $(c_{h_1}, c_{h_2}, \dots, c_{h_p})$ of r_1, and $(c_{j_1}, c_{j_2}, \dots, c_{j_q})$ of r_2

```
Start CHILD_ROW_SOURCE_1
For each row r1 = (c1, c2, ... , cn) from CHILD_ROW_SOURCE_1 Loop -- outer loop
    Start CHILD_ROW_SOURCE_2, given (r1.ch1, r1.ch2, ... , r1.chp)
    /*
      CHILD_ROW_SOURCE_2 uses the values of columns from the
      current row r1 in join access/filter conditions in order
      to find all rows r2 matching r1
     */
    For each row r2 = (c1, c2, ... , cm) from CHILD_ROW_SOURCE_2 Loop -- inner loop
        /*
          Rows from CHILD_ROW_SOURCE_2 are joined to the
          current row from CHILD_ROW_SOURCE_1
         */
        Yield the combined row rj = (r1.c1, ... , r1.cn, r2.c1, ... , r2.cm) to the parent operation (*)
    End Loop
End loop
```

(*) Actually, only projected columns are passed to the parent operation

Key points:

- CHILD_ROW_SOURCE_1 is started once per start of its parent
- CHILD_ROW_SOURCE_2 is started as many times as CHILD_ROW_SOURCE_1 supplies a row to be joined with
- CHILD_ROW_SOURCE_2 uses join columns from the "outer row" as input
- Join access/filter conditions are processed by CHILD_ROW_SOURCE_2