

# Examples of Transformations and SQL Query Rewrites

## Cartesian Product Elimination

Detects Cartesian Joins and propose corrections based on analysis of statement, for example suggesting `dept.deptno = emp.deptno` if emp and dept had no join criteria.

## Expression Transformation

Identifies actions on predicates that might suppress index usage such as "`where empid + 1 = 1`", should be "`where empid=0`"

## Invalid Outer Join

Identifies invalid outer joins and suggests more efficient alternatives.

Before	After
<pre>SELECT * FROM employee e, customer c WHERE e.employee_id = c.salesperson_id ( + ) AND c.state = 'CA'</pre>	<pre>SELECT * FROM employee e, customer c WHERE e.employee_id = c.salesperson_id ( + ) AND c.state( + ) = 'CA'</pre>

## Transitivity

Before	After
<pre>SELECT * FROM item i, product p, price pr WHERE i.product_id = p.product_id AND p.product_id = pr.product_id</pre>	<pre>SELECT * FROM item i, product p, price pr WHERE i.product_id = p.product_id AND p.product_id = pr.product_id AND i.product_id = pr.product_id</pre>

## Move Expression to WHERE Clause

Before	After
<pre>SELECT col_a, SUM(col_b) FROM table_a GROUP BY col_a HAVING col_a &gt; 100</pre>	<pre>SELECT col_a, SUM(col_b) FROM table_a WHERE col_a &gt; 100 GROUP BY col_a</pre>

## NULL Column

Before	After
<pre>SELECT * FROM employee WHERE manager_id != NULL</pre>	<pre>SELECT * FROM employee WHERE manager_id IS NULL</pre>

## Push Subquery

Before	After
<pre>SELECT * FROM employee WHERE employee_id = (SELECT MAX(salary) FROM employee)</pre>	<pre>SELECT employee.* FROM employee, (SELECT DISTINCT MAX(salary) col1 FROM employee) t1 WHERE employee_id = t1.col1</pre>

## Mismatched column types

Identify joins type mismatch such as `number = character` which might suppress use of Index.