

PERFORMANCE TEST 01 – WASTE REPORTING:

It shows data filtering, grouping, JOIN performance and the benefit of composite index.

PERFORMANCE TEST 02 – SALARY LOOKUP:

This shows worker based filtering, monthly salary lookup and the benefit of composite index (worker_id, year, month).

PERFORMANCE TEST 03 – SCHEDULING QUERY:

This test shows us multi table JOIN and data filtering. The index improves lookup on worker schedule.

PERFORMANCE OPTIMIZATION STRATEGY:

The system implements indexing on the frequently filtered columns in each table, the foreign key columns, reporting columns and composite columns used in WHERE clauses.

OBSERVATIONS FROM “EXPLAIN ANALYZE”:

The results obtained from EXPLAIN ANALYZE clearly indicate the impact of indexing on query performance. Before indexes were applied, the queries performed full table scans (type: ALL). This resulted in higher execution costs and a larger number of rows were being examined. This increased the processing time especially for queries involving JOINS and filtering conditions.

After creating appropriate single column and composite indexes, the execution changed to index based scans (type: ref). This reduced the number of rows scanned and overall lowered query cost. The most noticeable improvements were observed in salary lookups, waste collection reporting and worker scheduling queries. The composite queries along with filtering and JOIN conditions provided optimal performance gains.

This confirms that proper indexing plays a crucial role in achieving database efficiency.