RDBMS, Query Plans and Evaluation: Comprehensive Review

DSC 208R - Data Management for Analytics

June 2025

Overview of RDBMS

A Relational Database Management System (RDBMS) is a software system that implements the relational model, enabling users to store and process relational databases. RDBMS software is a significant industry, with many popular open-source options also available. Examples of RDBMS include ORACLE, Microsoft SQL Server, Amazon Redshift, Google Big Query, SAP, Snowflake, Pivotal, HP, PostgreSQL, MySQL, SQLite, and DuckDB.

Life of an SQL Query

The process of an SQL query within an RDBMS involves several stages:

1. A user submits an SQL Query (e.g., via Web Forms, Application Front Ends, or SQL Interface).
2. The query is sent to the Database Server.
3. **Parser**: Parses the SQL query to check for syntax correctness and converts it into an internal representation, often a syntax tree or Logical Query Plan (LQP).
4. **Optimizer**: Takes the LQP and generates an optimized Physical Query Plan (PQP). It considers various execution strategies to find the most efficient one.
5. **Query Scheduler**: Manages the execution of the PQP, potentially breaking it into segments for parallel or distributed execution.
6. **Execute Operators (Plan Executor / Operator Evaluator)**: Executes the physical operations defined in the PQP. This interacts with various DBMS components like Files and Access Methods, Buffer Manager, and Disk Space Manager.
7. **Result**: The query result is returned to the user.

Other core components of an RDBMS include:

- **Transaction Manager**: Manages transactions, ensuring atomicity, consistency, isolation, and durability (ACID properties).
- **Recovery Manager**: Handles system failures and ensures the database can be restored to a consistent state.
- **Lock Manager / Concurrency Control**: Manages concurrent access to data to prevent conflicts.
- **System Catalog**: Stores metadata about the database schema (tables, columns, indexes, etc.).
- **Index Files and Data Files**: Where the actual data and indexes are stored on disk.

Query Plans and Evaluation

Query plans are structured representations of how an RDBMS will execute an SQL query.

Logical Query Plan (LQP)

- A Directed Acyclic Graph (DAG) with vertices representing ”Logical Operators” from Extended Relational Algebra.
- It tells *what* is computed.
- Each logical operator can have alternate ”physical” implementations.

Physical Query Plan (PQP)

- A DAG with vertices called ”Physical Operators”.
- It specifies the exact algorithm/code to run for each logical operation, including all parameters.
- It tells *how* the query is computed.
- A single logical query can have many possible physical plans.

Example SQL Query (Netflix Schema):

```
SELECT M.Year, COUNT(*) AS NumBest
FROM Ratings R, Movies M
WHERE R.MID = M.MID
AND R.Stars = 5
GROUP BY M.Year
ORDER BY NumBest DESC;
```

Netflix Schema:

- **Ratings** (RatingID, Stars, RateDate, UID, MID)
- **Users** (UID, Name, Age, Join Date)
- **Movies** (MID, Name, Year, Director)

Logical Query Plan for the example query:

```
Result Table
   ^
   |
  SORT (On NumBest)
   ^
   |
  GROUP BY AGGREGATE (M.Year, COUNT(*))
   ^
   |
  JOIN (R.MID=M.MID)
  /    \
 SELECT  SELECT
 R.stars=5 No predicate
  /        \
Ratings Table  Movies Table
```

Physical Query Plan for the example query (assuming B+ Tree Index on Ratings.Stars):

```
Result Table
   ^
   |
  External Merge-Sort (In-mem quicksort; B=50)
   ^
   |
  Hash-based Aggregate
   ^
   |
  Index-Nested Loop Join
  /                  \
 Indexed Access     File Scan
 Use Index on Stars  Read heapfile
  /                  \
Ratings Table       Movies Table
```

RDBMS Logical-Physical Separation

The concept of Logical-Physical Separation (or data independence) is a hallmark of RDBMSs.

- **Benefits**:
  - Increased user productivity.
  - Automated optimization leads to faster and more scalable code.
  - Application portability, as internal system changes do not affect the application.
- This declarativity has influenced other systems like Hadoop/MapReduce, data visualization, graph processing systems, and scalable ML systems.

Indexes and Their Usage

Indexes are data structures that improve the speed of data retrieval operations on a database table. They are crucial for efficient physical query plans.

- Physical operators are chosen based on data and system parameters, as their runtimes can vary significantly.
- Common physical operators for major extended Relational Algebra (RA) logical operators include:
  - **Selection**: Filescan, Index-based.
  - **Join**: Nested loop, Hash join, Sort-merge join.
  - **Aggregation**: Hash-based, Sort-based, Index-based.

Glimpse of Query Optimization

Query optimization is the process of choosing the most efficient execution plan for an SQL query.

‘EXPLAIN [ANALYZE]‘ in SQL

Most RDBMSs allow users to see the logical and physical plans generated for a query using the ‘EXPLAIN‘ command, sometimes with runtimes attached (with ‘ANALYZE‘).

- This command can offer insights when a query runs too slowly.

Example ‘EXPLAIN‘ Output:

```
EXPLAIN SELECT sum(i) FROM foo WHERE i < 10;
```

```
                      QUERY PLAN
------------------------------------------------------
 Aggregate (cost=23.93..23.93 rows=1 width=4)
   -> Index Scan using fi on foo (cost=0.00..23.92 rows=6 width=4)
       Index Cond: (i < 10)
(3 rows)
```

This output shows that the query will perform an ‘Index Scan‘ on ‘foo‘ using index ‘fi‘ with the condition ‘i ¡ 10‘, and then an ‘Aggregate‘ operation.