Repetition & Highlights:
SQL from the outside and the inside

SQL (for a user)

Details: https://www.postgresql.org/docs/current/sql.html

SQL is the inter-galactic data-speak with several sub-speaks:

- Data Definition Language (**DDL**):
  CREATE-queries, creating schemas
- Data Query Language (**DQL**):
  SELECT-queries
- Data Manipulation Language (**DML**):
  INSERT/UPDATE, write-queries
- Data Control Language (**DCL**):
  GRANT/REVOKE, access & user management

SQL (for a user)

- Works on **tables** (or **relations**):
  - **Relation name**
  - **Relation schema**: set of attribute names with associated datatypes
  - Set(!) of **Relation records:** tuples of elements conforming the schema

  - Example:

Tutorials

ID	site	tutorial	topic
1	w3schools	SQL_2003STD	Database
2	w3schools	HTML_5	WebDev
3	w3schools	CSS_3	WebDev
4	w3resource	SQL_2003STD	Database
5	w3resource	MySQL	Database

SQL (for a user)

SQL is the inter-galactic data-speak with several sub-speaks:

- Data Definition Language (**DDL**):
  CREATE-queries, creating schemas
- Data Query Language (**DQL**):
  SELECT-queries
- Data Manipulation Language (**DML**):
  INSERT/UPDATE, write-queries
- Data Control Language (**DCL**):
  GRANT/REVOKE, access & user management

SQL (for a user)

- Example (Select-project-join-query)

```
SELECT p.Name, m.Id FROM Person p JOIN Movie m ON
    p.Id = m.ActorId
WHERE m.DirectorId = 1234;
```

- Many other operators (**`LEFT/RIGHT JOIN`**, **`UNION`**, **`GROUP BY`**, subqueries, etc.)

- Always to keep in mind: bag semantics by default, NULL may behave strangely, rich datatypes

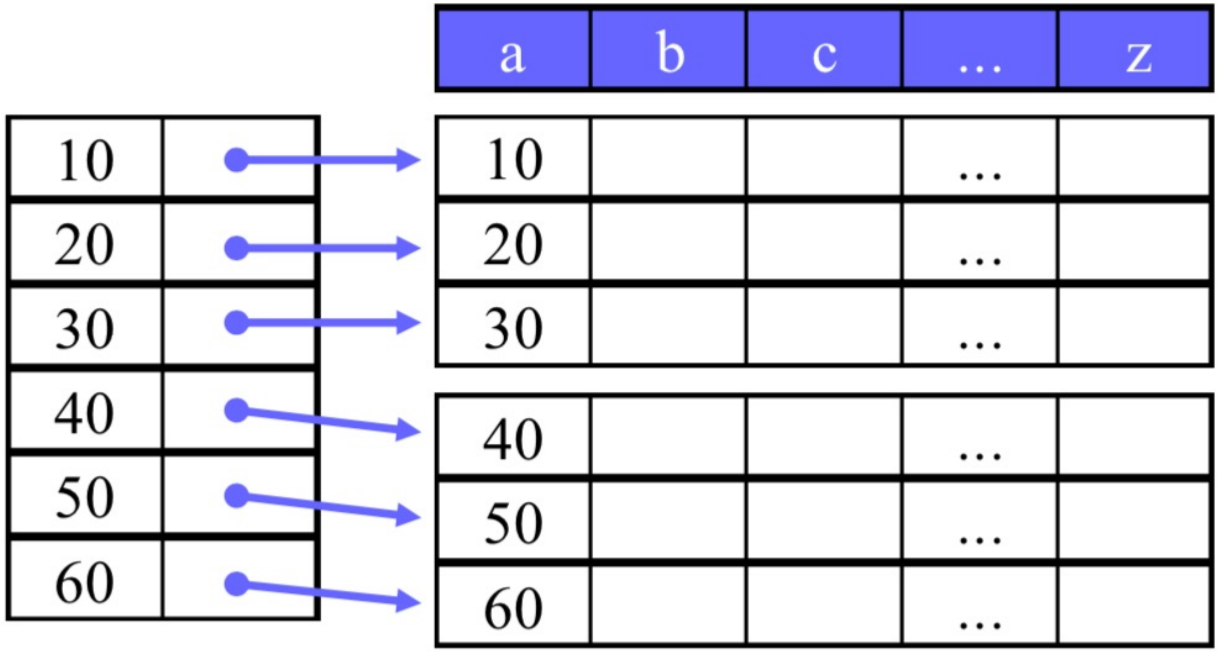
Inside SQL: Indices

- An **index** on an attribute A is a data structure that facilitates finding the elements with a certain values in A (A is called the **search key**)
- The index is **organised** (e.g., sorted) on the search key
- For each value of the search key, the index has a list of pointers to the corresponding records
- More than one index in the same table (or file) means
  - Faster search
  - More complexity (changes will lead to updated indices as well)
  - Increased storage requirement, larger files

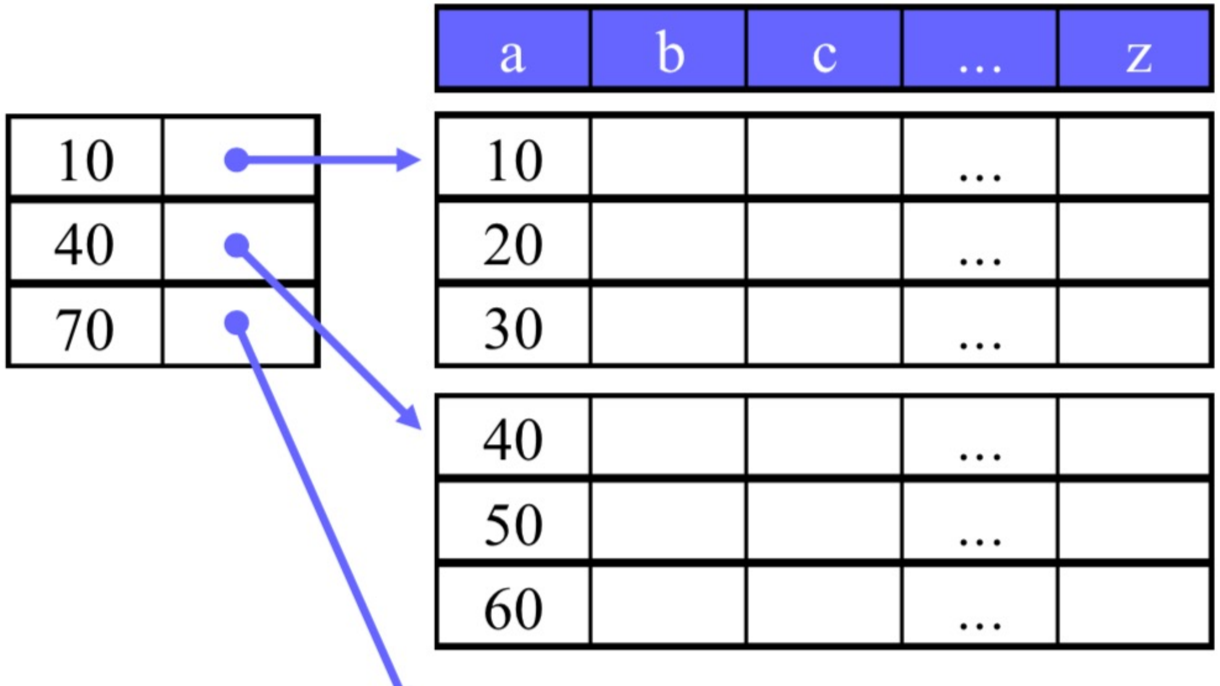
Primary indices

Dense vs. Sparse

A **dense index** has one lookup for each value of the search key



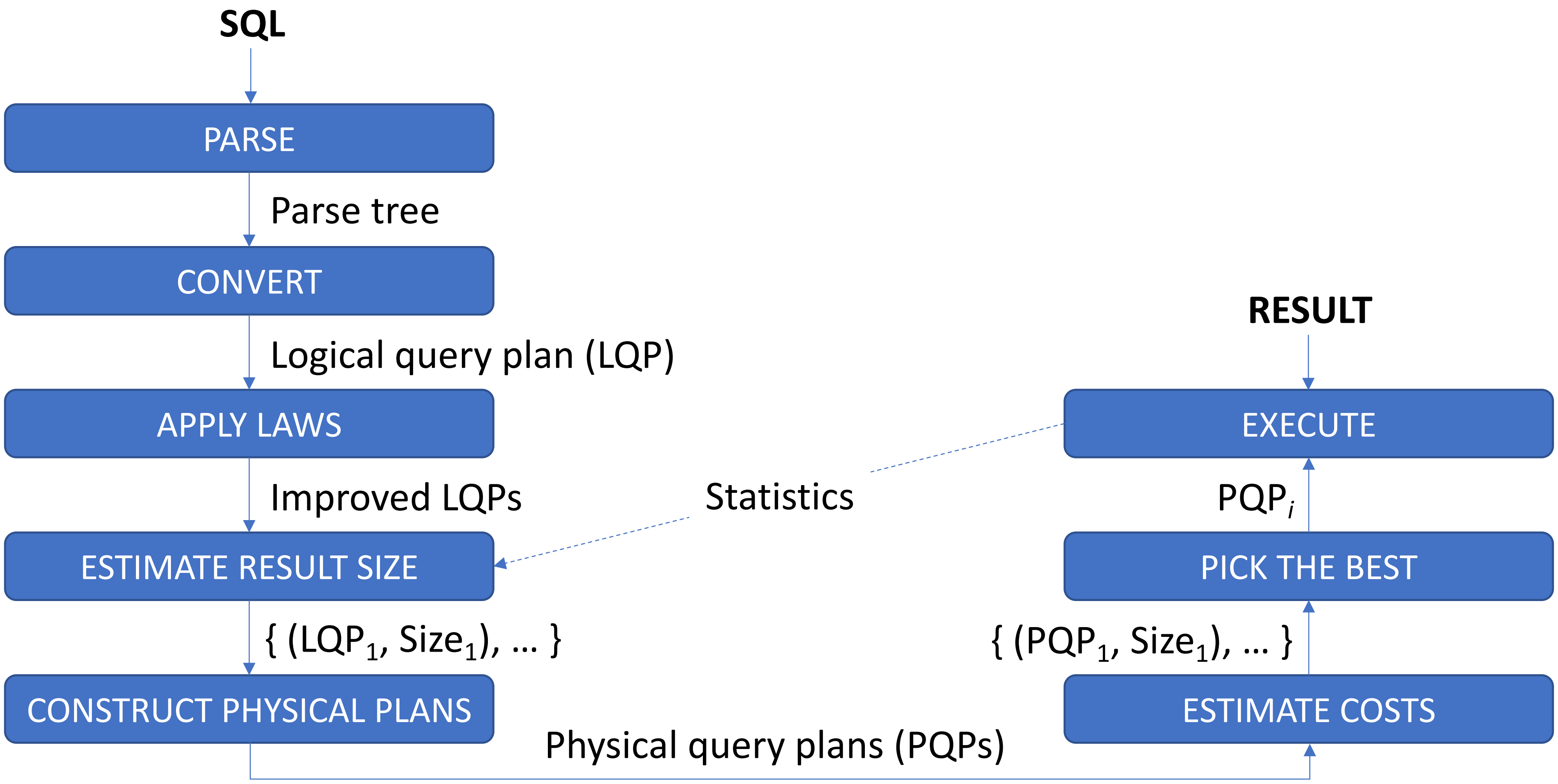
A **sparse index** has one lookup for each data block



Types of indices

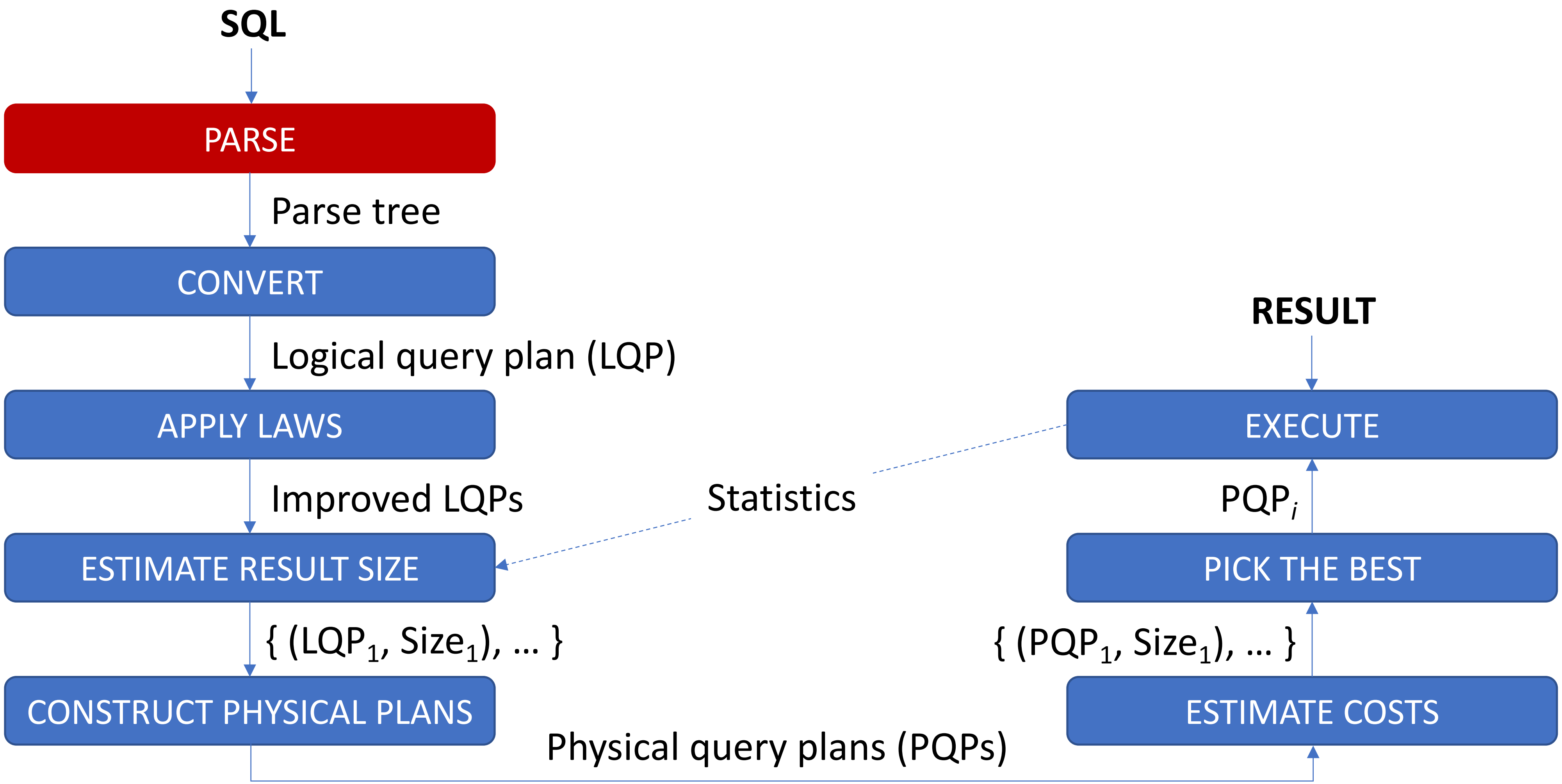
- Dense/sparse
- Primary/Cluster/Secondary
  - Determines how main data file is sorted
- Multilevel (indices on indices)
- Multidimentional (simultaneously on many attributes)
- Organisation: List/B+/hash/…

Inside SQL: The (Typical) Journey of a Query

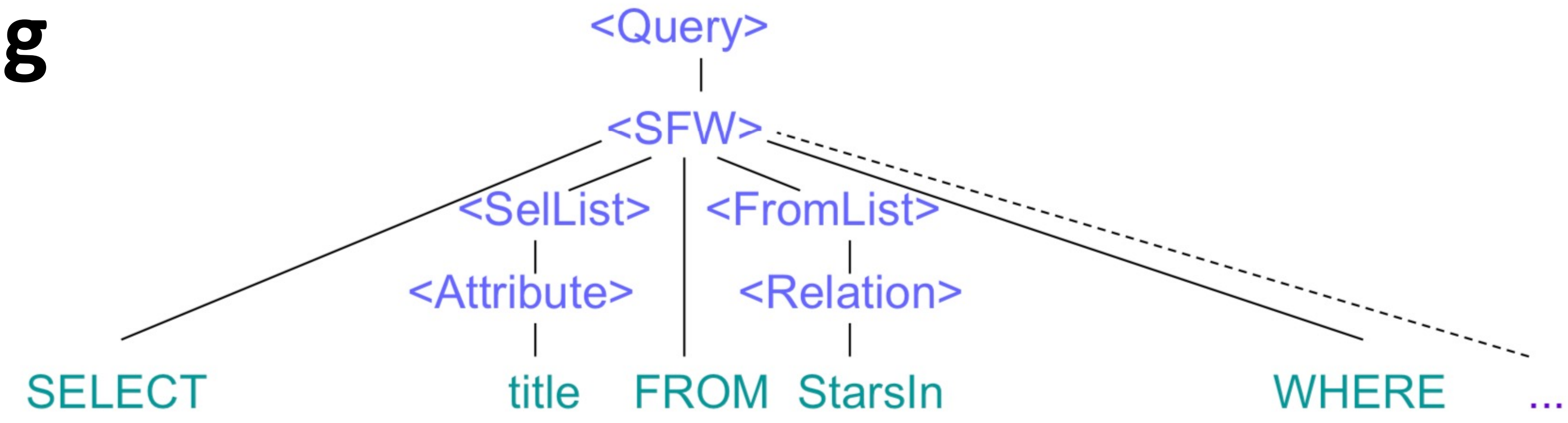


Parsing

The goal is to convert an SQL-query to a parse tree

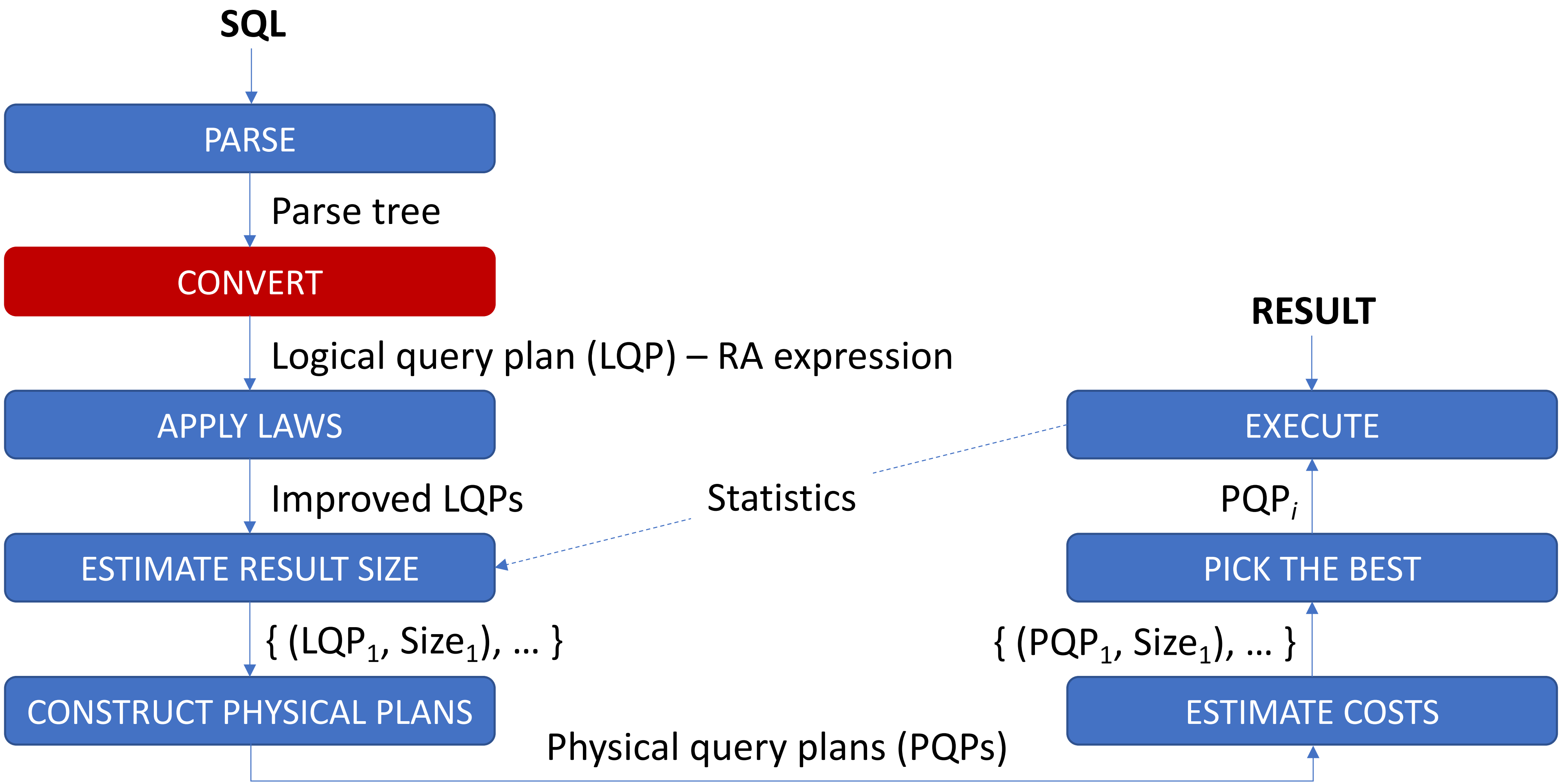


Parsing



- Each node in a parse tree is
- an *atom (primitive)* – i.e., a lexical element like a keyword, name, constant, parentheses or operators ... (leaf node)
- a *syntactic* category – part of the query ... (inner node)

Generating the logical query plan

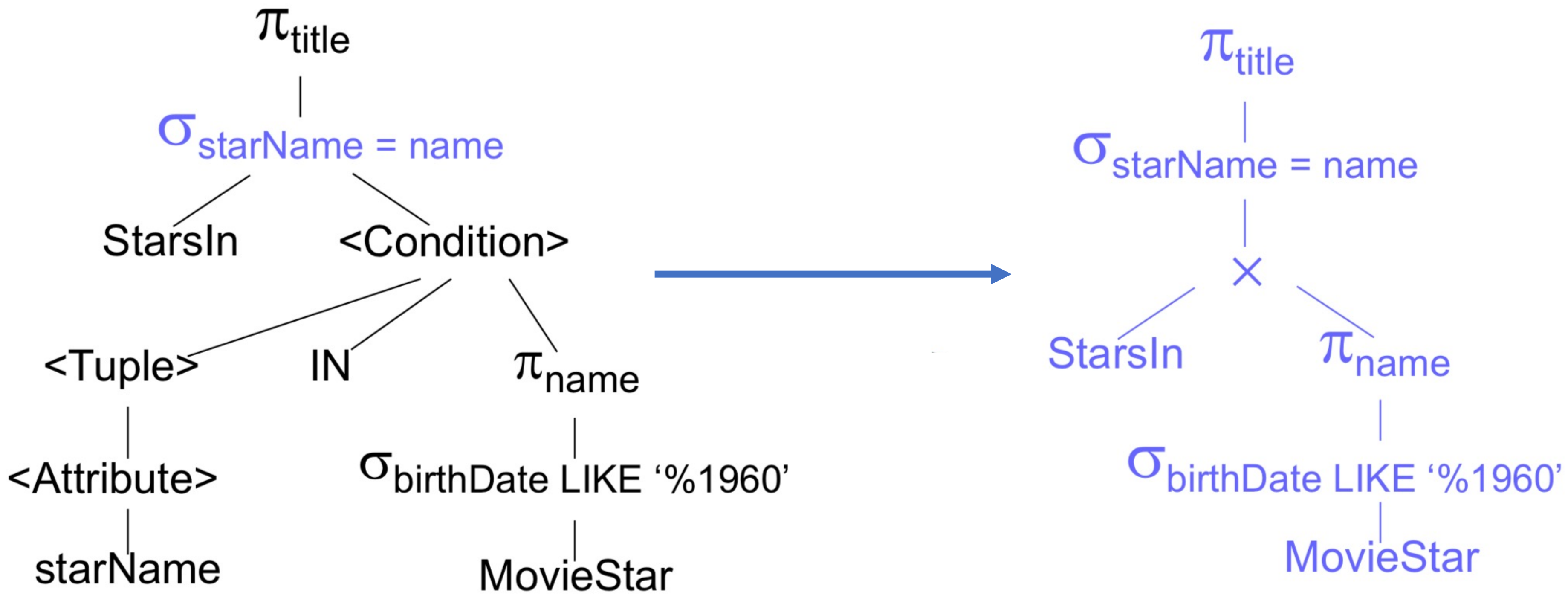


Relational Algebra

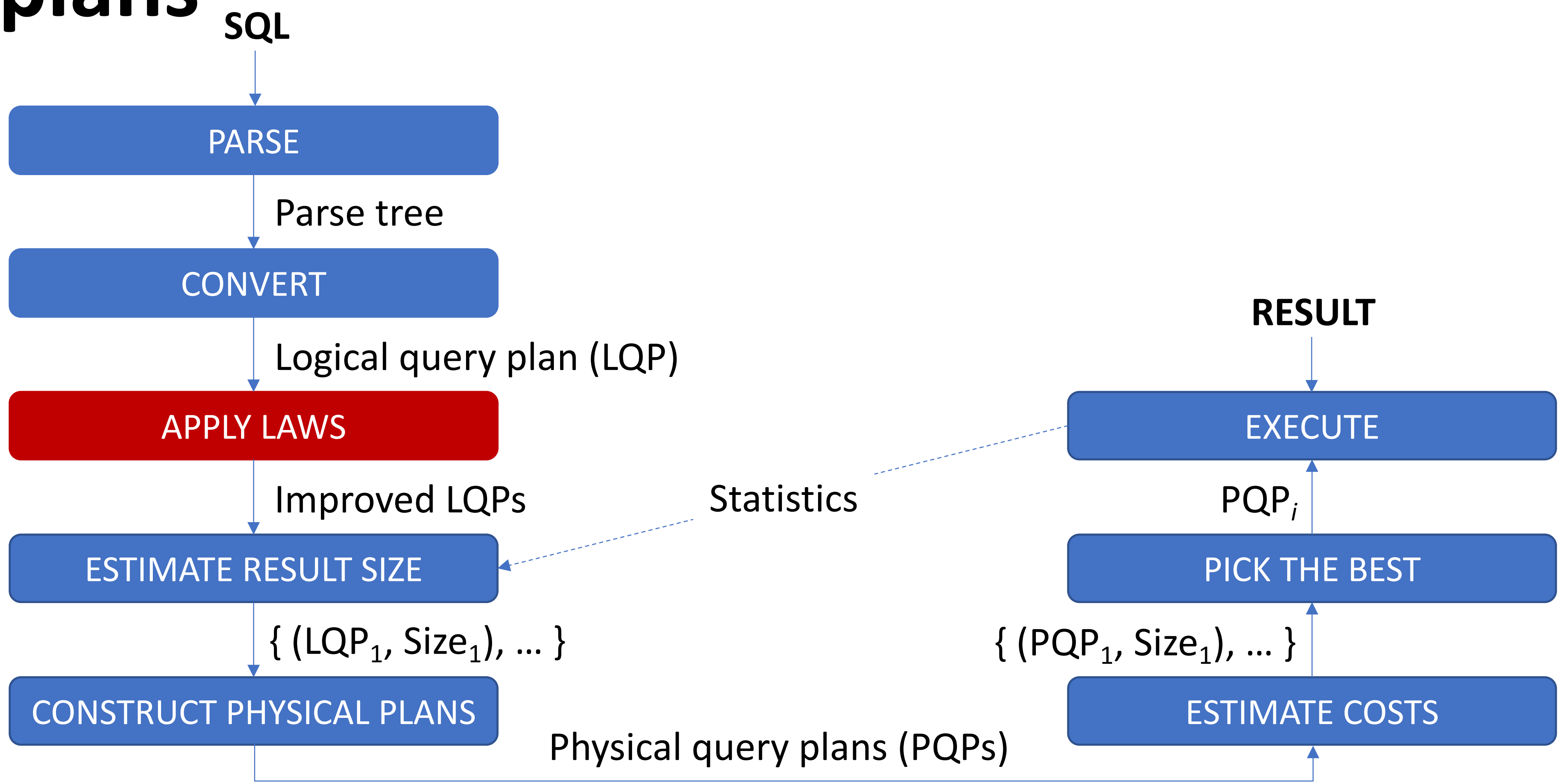
- **Domain:** relations (tables)
- **Set operators:**
  1. (Set) Union $R \cup S$
  2. (Set) Difference $R - S$
  3. Projection $\pi_L(R)$
  4. Selection $\sigma_C(R)$
  5. Cartesian product (a.k.a., Cross Product and Cross Join) $R \times S$
  6. Renaming $\rho_{S(A1,A2,\ldots,An)}(R)$
  7. + Expressible operators (Intersection, Other joins, Division, etc.)
- **Bag operators:**
  1. Versions of the set operators
  2. Duplicate elimination
  3. Aggregation
- **Beyond bags**:
  1. Sorting, etc.

From SQL to RA Example

```
SELECT title FROM StarsIn WHERE starName IN
( SELECT name
  FROM MovieStar
  WHERE birthDate LIKE '%1960'
);
```



Algebraic laws for improving logical query plans

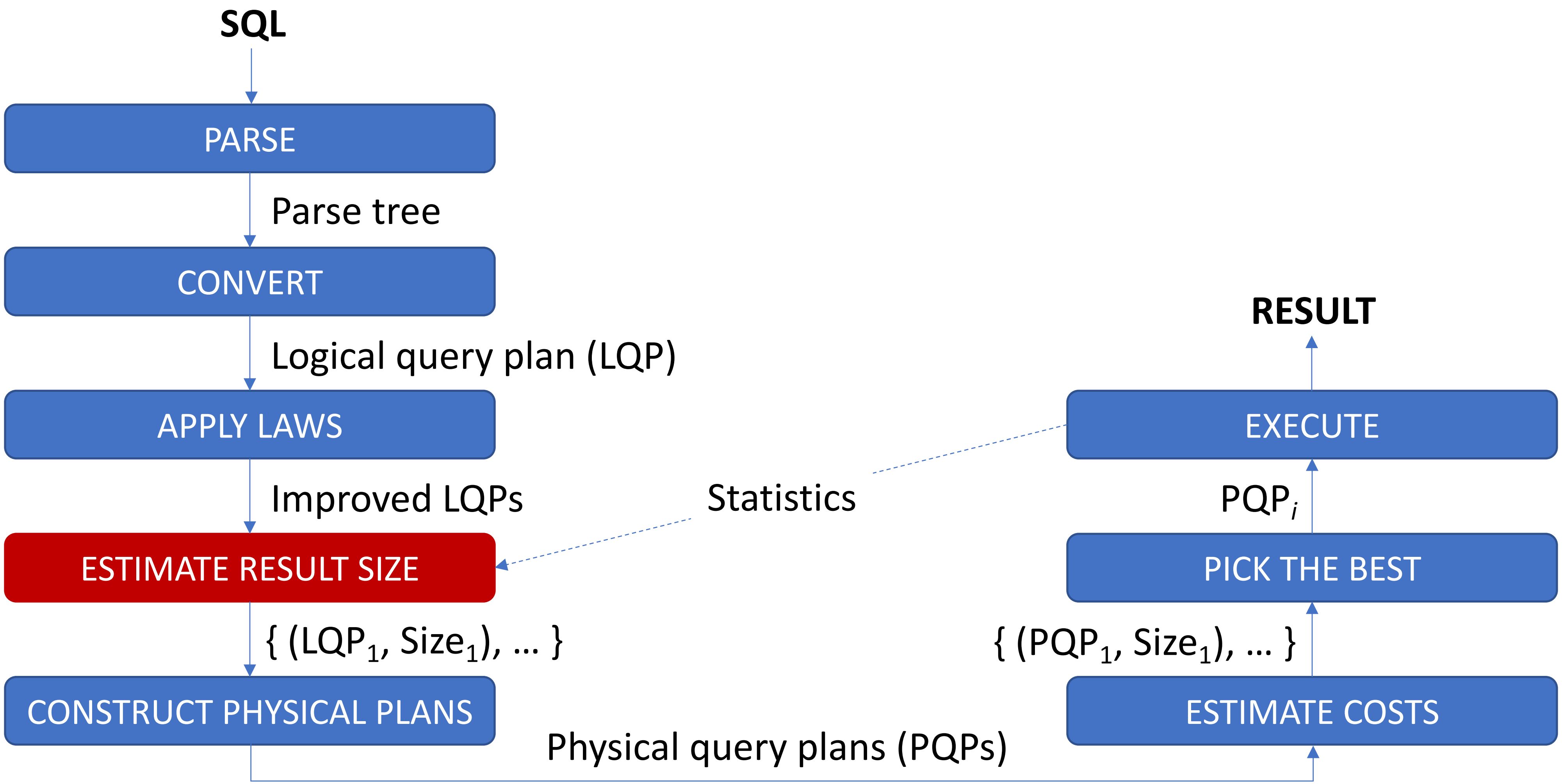


Query Optimisation: Equivalent Plans

- There are many RA lows preserving query results:
  - associativity and commutativity of many operators
    - e.g., $R \bowtie S = S \bowtie R$, $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$, etc.
  - special rules for selection
    - e.g., $\sigma_a(R \bowtie S) = \sigma_a(R) \bowtie \sigma_a(S)$, $\sigma_{a \text{ AND } b}(R) = \sigma_a(\sigma_b(R))$
  - etc.

- From one LQP we can get many equivalent ones

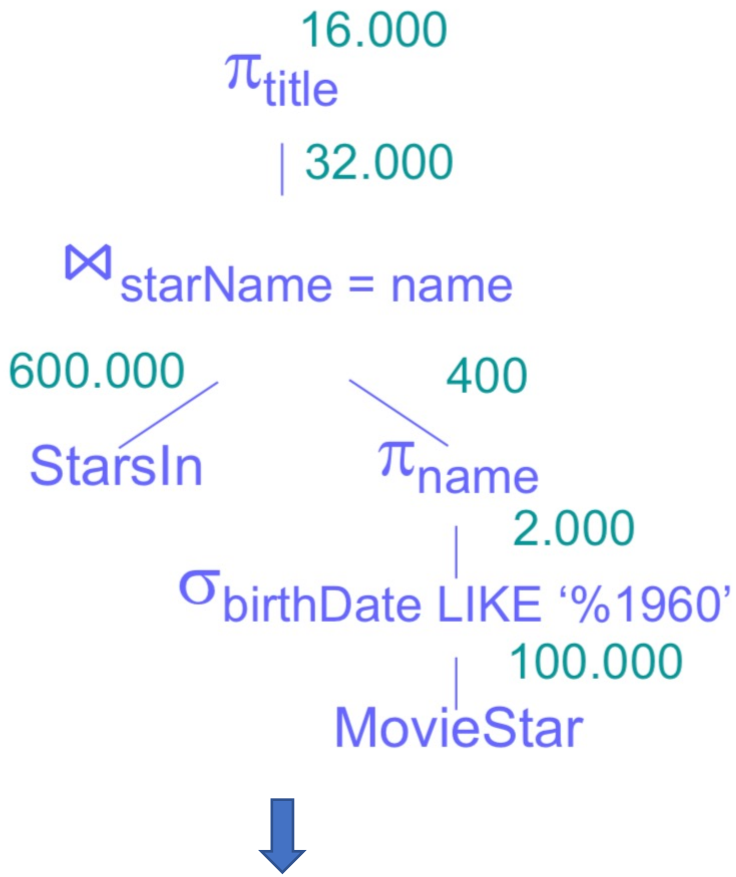
Query Plan Cost Estimation

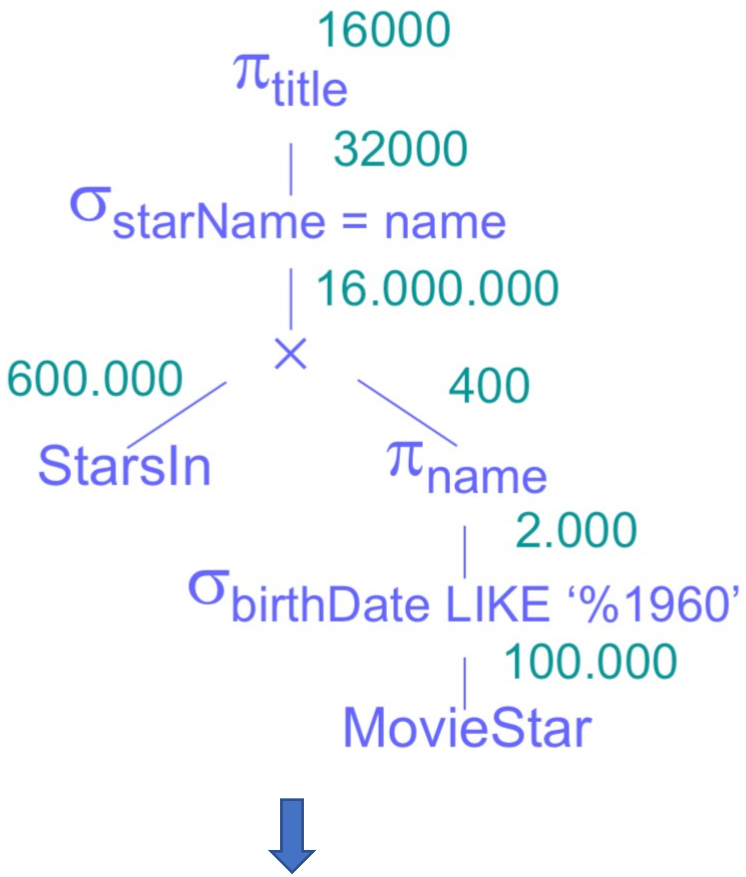


Estimating LQPs Costs

- To assess LQPs, we need some way to calculate cost
- DBMS estimates the costs
- We use **size of temporary relations** (#tuples x TupleSize)
  - Estimate the result of each operator
  - Add costs to the tree
  - The cost of the plan is the sum of all the costs in the tree (except the leaves and the root)

Comparing Logical Query Plans - Example

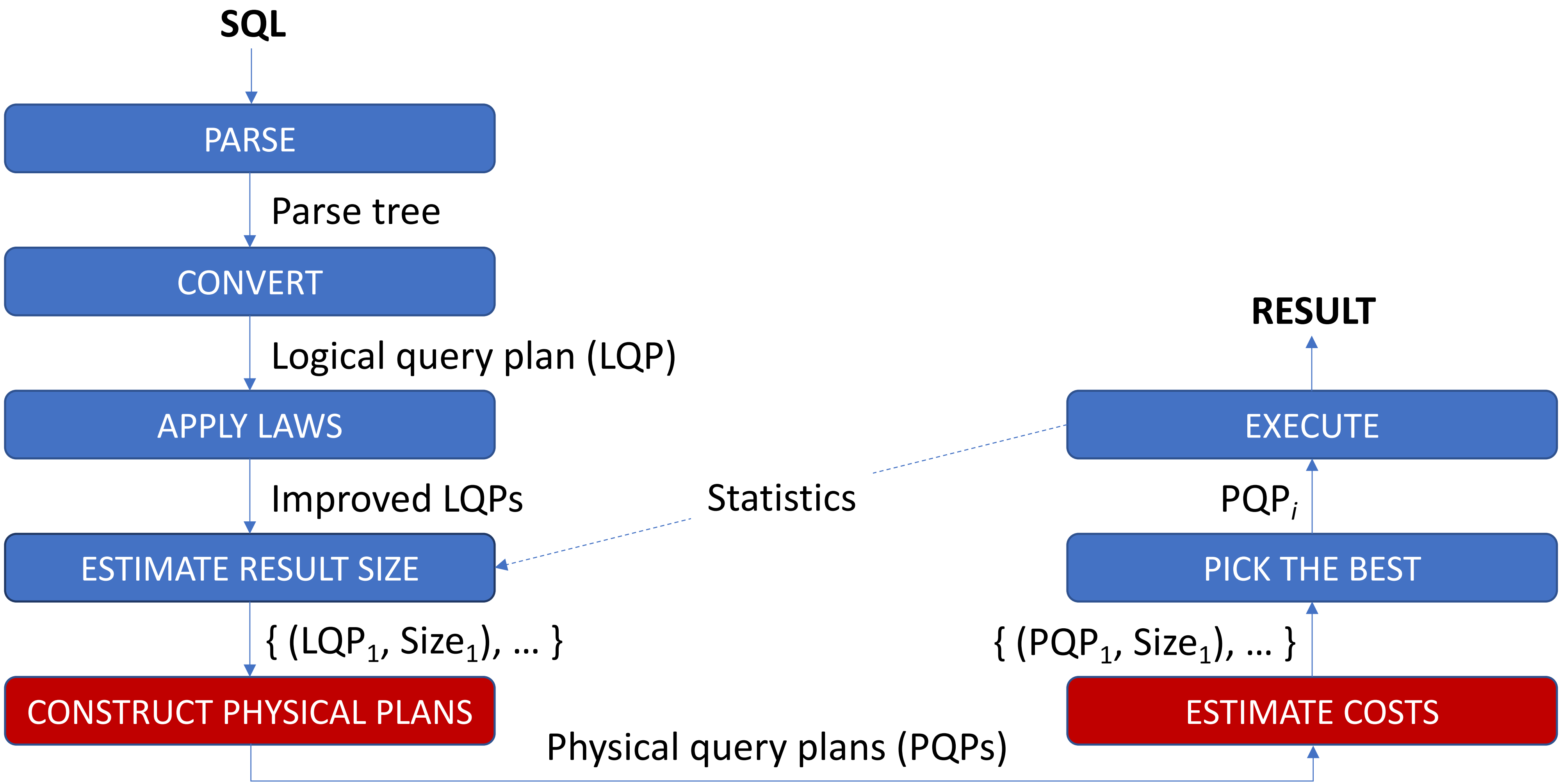




Total: 2000 + 400 + 32,000 = 34,400

Total: 2000 + 400 + 20,000,000 + 40,000 = 20,042,400

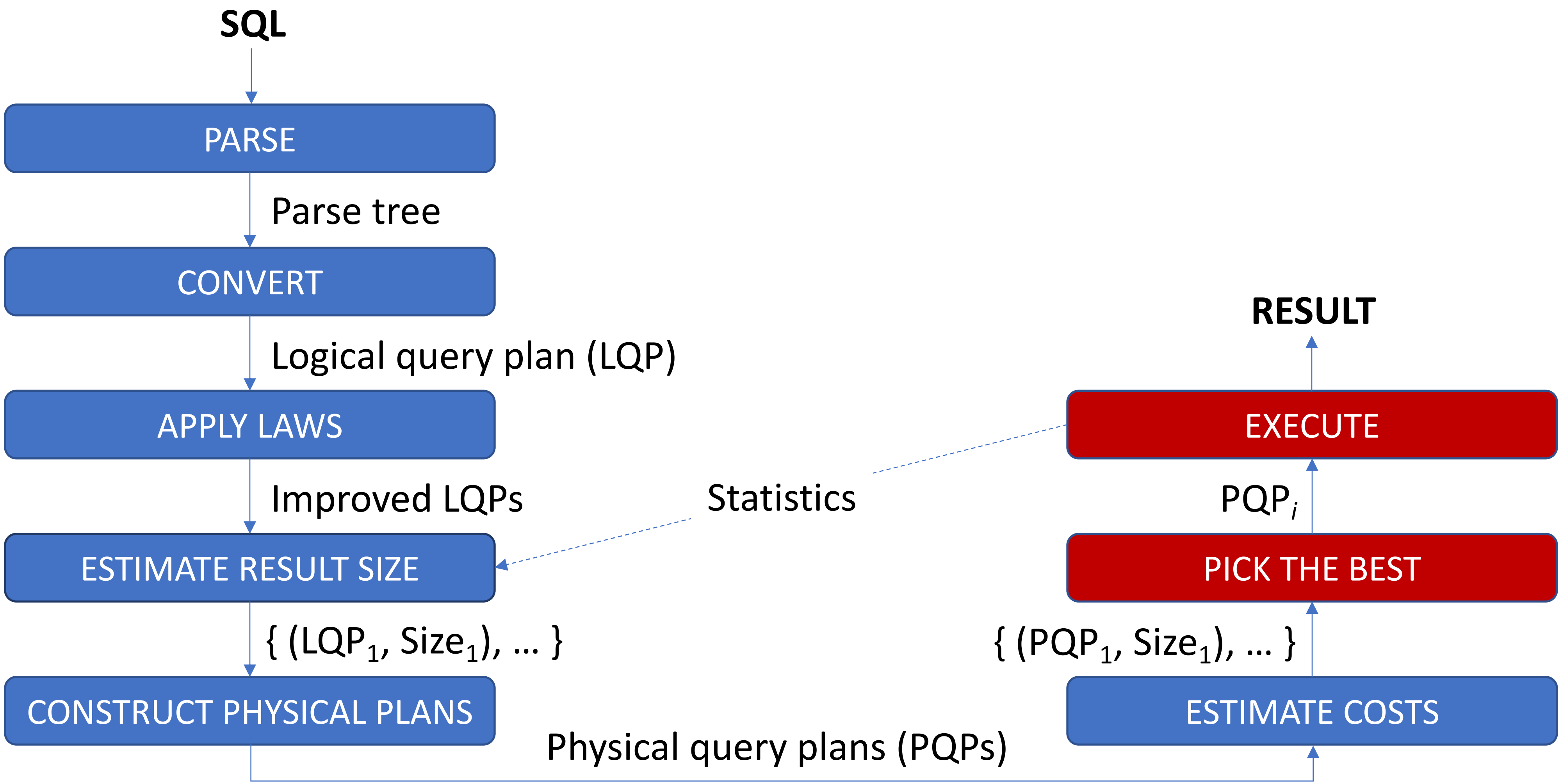
Physical Plans



Physical operators

- A query can be expressed in terms of a relational algebra expression
  - A **physical query plan (PQP)** is implemented by algorithms for relational algebra operators
  - They are based on basic algorithms for reading (scanning) a relation, sorting a relation, etc.
    - Two-Phase Multiway Merge Sort (TPMMS), etc.
- To choose a good PQP, we need to estimate the cost of algorithms
  - We can use the **number of disk IOs** as the cost

Physical Plans



Repetition & Highlights: Graph DBs

(Labelled) graph databases

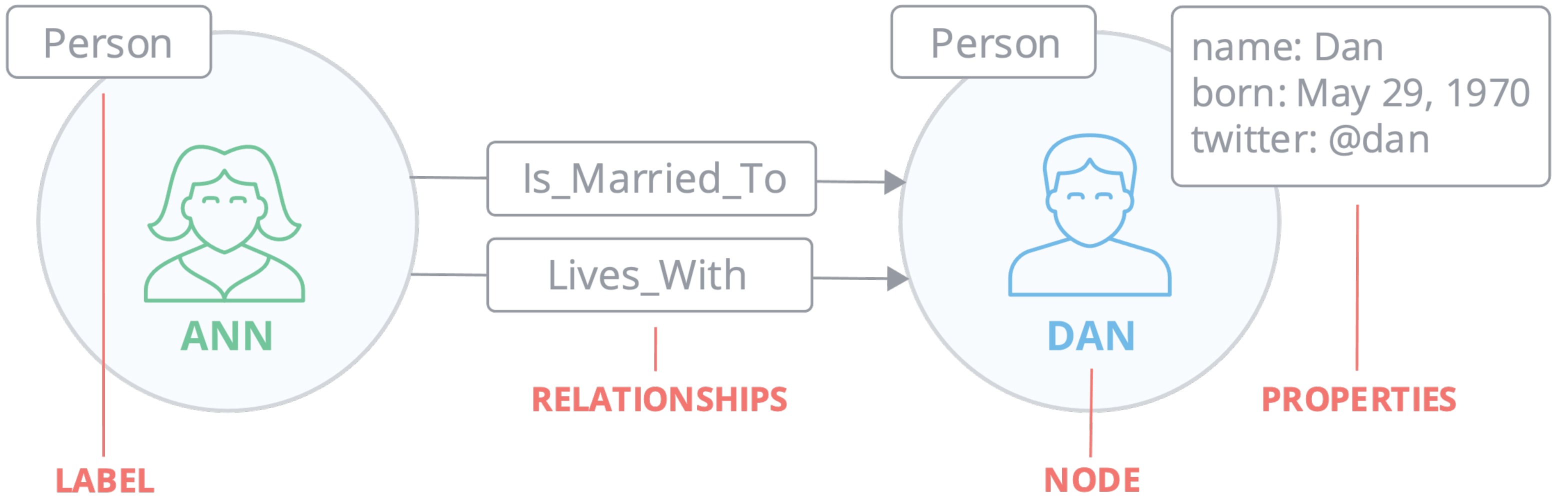
- For data that is natural to describe and traverse as graphs
  - Each node has an inner structure describing its properties
  - The edges indicate relationships between the nodes
  - The edges can carry information in the same way as the nodes
- Can be schema-free
  - New nodes and edges (with new inner structures) can be introduced dynamically
  - Existing nodes and edges can be expanded with new properties
- Search: Specify how the graph should be navigated
  - The graph is traversed directly via pointers to neighboring nodes (the traverse requires no indexes and no join operations)

Linked Data, Semantic Knowledge Graphs

- Method for **publishing data on the Web**
- **Self-describing** data and relations
- **Interlinking**
- Accessed using **semantic queries**

- A set of open standards developed by W3C
  - Data format: RDF
  - Knowledge representation: RDFS/OWL
  - Query language: SPARQL

Labeled Property Graphs (e.g., Neo4j)



- Node, Relationship, Property, Label, Relationship type
- Cypher query language