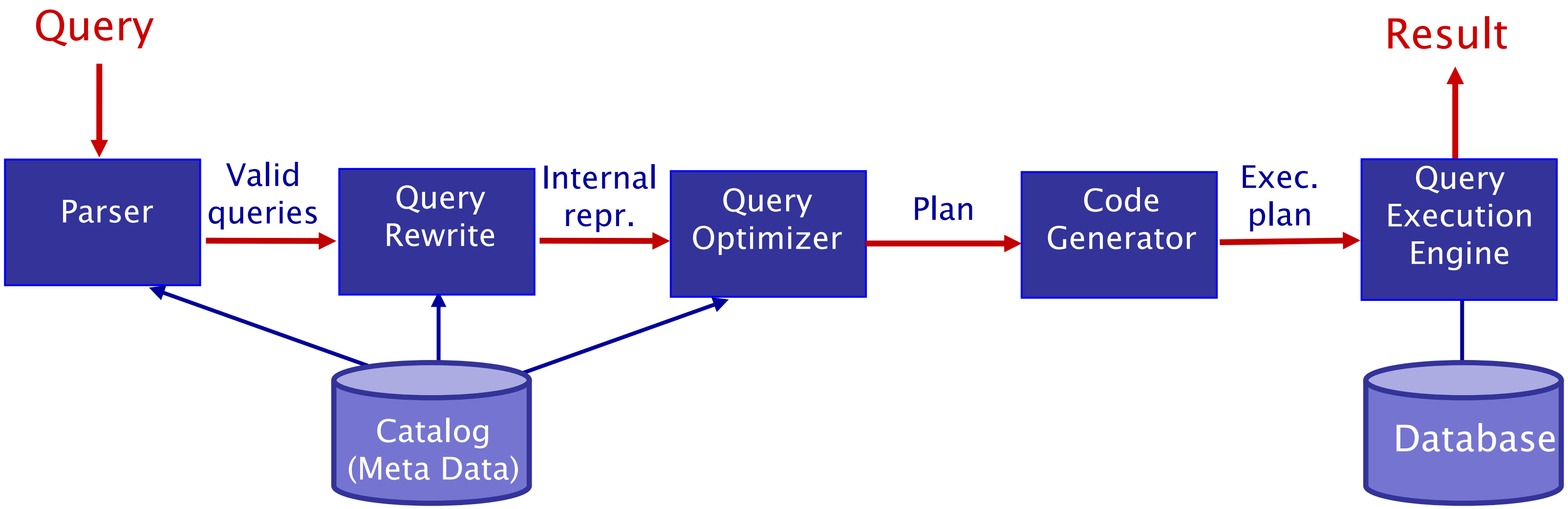
Lecture 7

Strategies for Query Processing

Query Processing

Deals with developing algorithms that analyze queries to generate a *good* execution plan that defines a sequence of steps for query evaluation, each step corresponds to one relational operation and its evaluation method.



Typical steps when processing a high-level query

Basic Steps in Query Processing

- Parsing and translation
  - Parser checks syntax, verifies relations
  - translate the query into its internal form.
    - Query Graph
    - Query Tree

- Query optimization
  - the process of choosing a suitable execution strategy for processing a query.

- Execution
  - The query-execution engine takes a query-evaluation plan, executes that plan, and returns the query answers to the user.

Translating SQL Queries into Relational Algebra and Other Operators

- SQL
  - Query language used in most RDBMSs
- Query decomposed into query blocks
  - Basic units that can be translated into the algebraic operators
  - Contains single SELECT-FROM-WHERE expression
    - May contain GROUP BY and HAVING clauses

Translating SQL Queries (cont'd.)

- Example:

```
SELECT Lname, Fname
FROM EMPLOYEE
WHERE Salary > ( SELECT MAX (Salary)
                 FROM   EMPLOYEE
                 WHERE  Dno=5 );
```

  - Inner block

```
( SELECT MAX (Salary)
  FROM   EMPLOYEE
  WHERE  Dno=5 )
```

  - Outer block

```
SELECT  Lname, Fname
FROM    EMPLOYEE
WHERE   Salary > c
```

Translating SQL Queries (cont'd.)

- Example (cont'd.)
  - Inner block translated into:

$$\Im_{\text{MAX Salary}}(\sigma_{\text{Dno}=5}(\text{EMPLOYEE}))$$

  - Outer block translated into:

$$\pi_{\text{Lname,Fname}}(\sigma_{\text{Salary}>c}(\text{EMPLOYEE}))$$

- Query optimizer chooses execution plan for each query block

Algorithms for SELECT Operation

- SELECT operation
  - Search operation to locate records in a disk file that satisfy a certain condition
  - File scan or index scan (if search involves an index)
- Search methods for simple selection
  - S1: Linear search (brute force algorithm)
    - Linear search can be applied regardless of
      - selection condition or
      - ordering of records in the file, or
      - availability of indices.

Algorithms for SELECT Operation

- S2: Binary Search
  - Applicable if selection is an equality comparison on the attribute on which file is ordered.
- S3: Using Primary Index
  - Selection condition must be on search-key of index.
  - Retrieve a single record that satisfies the corresponding equality condition.

Algorithms for SELECT Operation

- S4: Using a secondary index to retrieve multiple records
  - Retrieve a single record if the search-key is a candidate key
  - Retrieve multiple records if search-key is not a candidate key
- S5: Using a clustering index to retrieve multiple records
- S6: Using a hash key
- S7: Using a secondary (B+ -tree) index on an equality comparison.

Algorithms for SELECT Operation

- Selectivity
  - Ratio of the number of records (tuples) that satisfy the condition to the total number of records (tuples) in the file
  - Number between zero (no records satisfy condition) and one (all records satisfy condition)
- Query optimizer receives input from system catalog to estimate selectivity

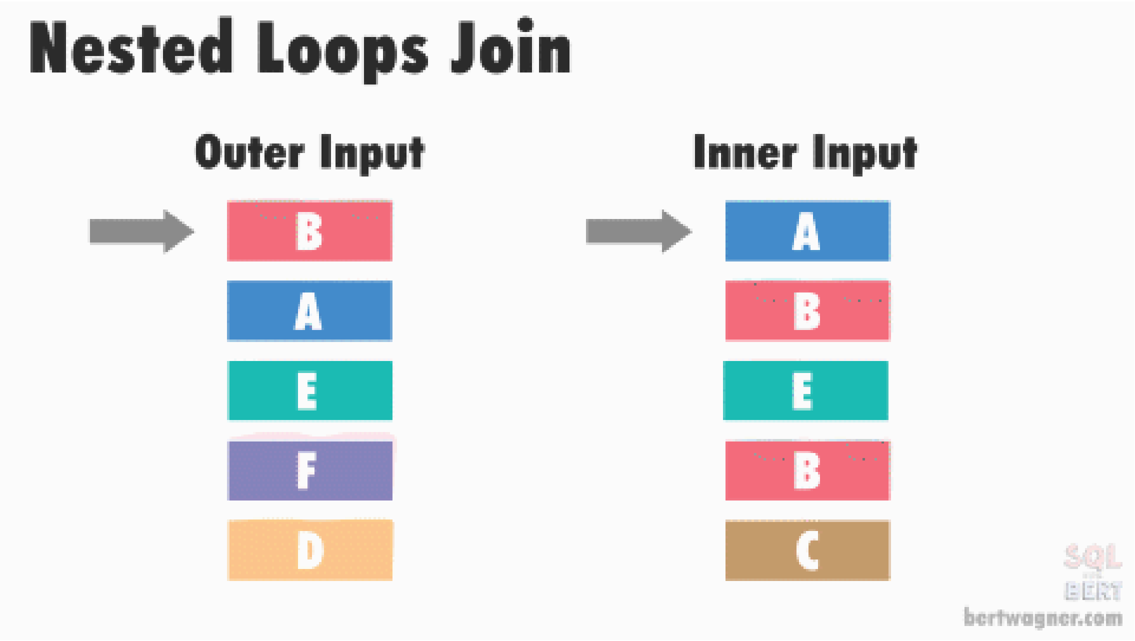
Implementing the JOIN Operation

- JOIN operation
  - One of the most time consuming in query processing
  - EQUIJOIN (NATURAL JOIN)
  - Two-way or multiway joins
- Methods for implementing joins
  - J1: Nested-loop join (nested-block join)
  - J2: Index-based nested-loop join
  - J3: Sort-merge join
  - J4: Hash Join

Implementing the JOIN Operation

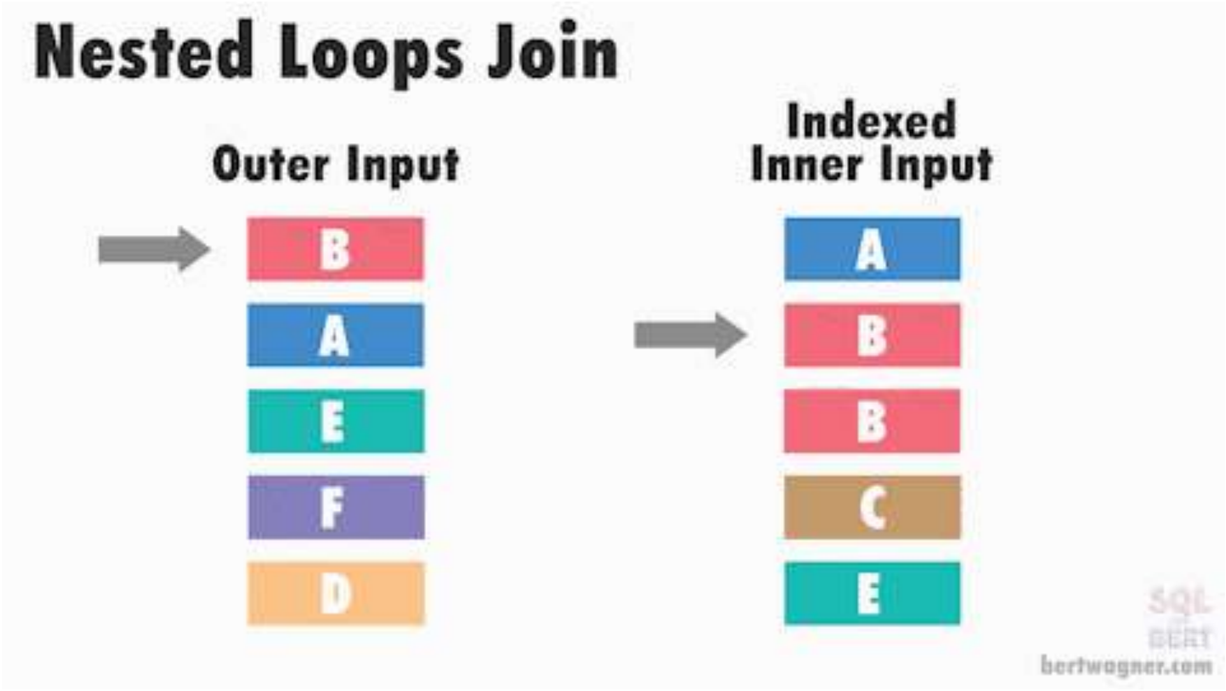
- Available buffer space has important effect on some JOIN algorithms
- Nested-loop approach
  - Read as many blocks as possible at a time into memory from the file whose records are used for the outer loop
  - Advantageous to use the file with fewer blocks as the outer-loop file

J1: Nested-loop join (nested-block join)



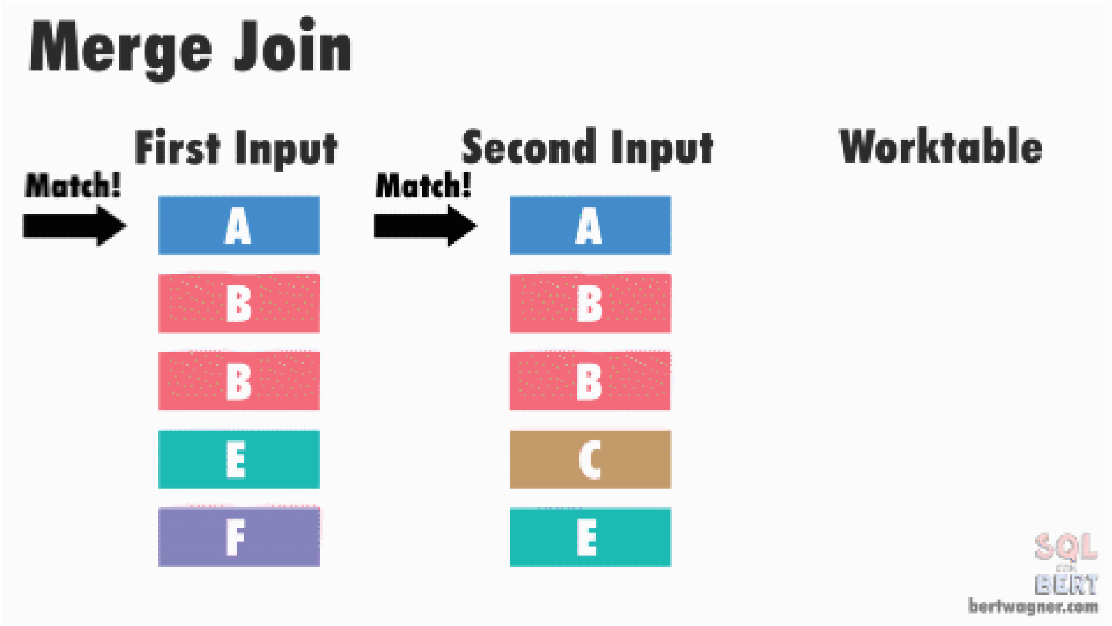
https://bertwagner.com/posts/visualizing-nested-loops-joins-and-understanding-their-implications/

J2: Index-based nested-loop join



https://bertwagner.com/posts/visualizing-nested-loops-joins-and-understanding-their-implications/

Sort-merge join

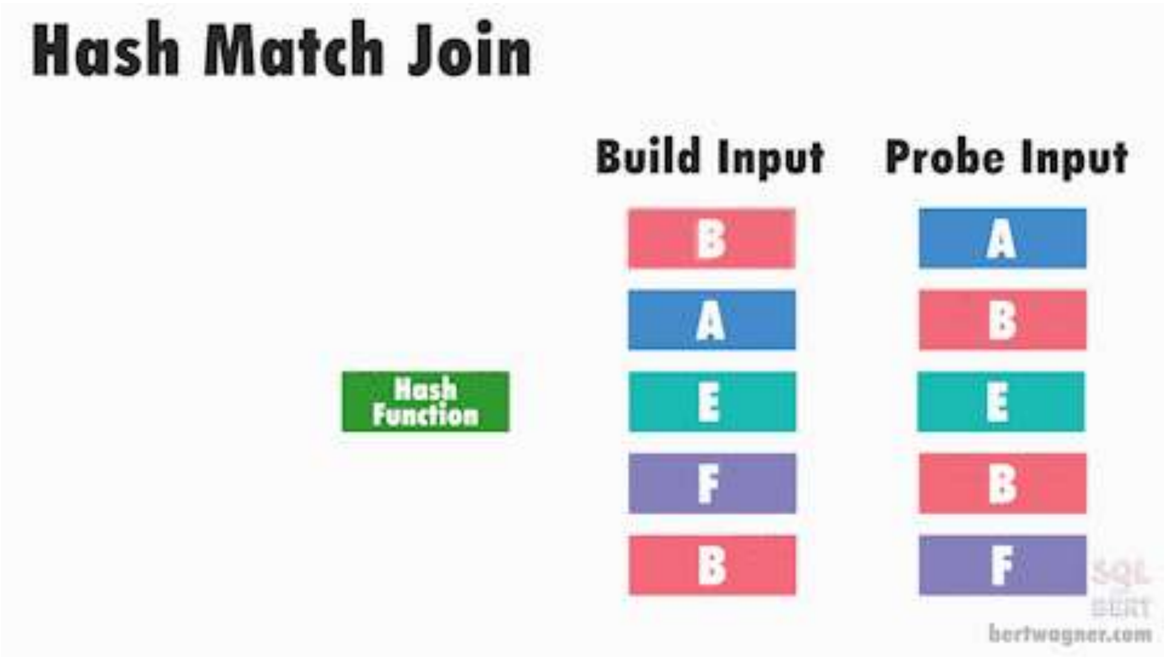


https://bertwagner.com/posts/visualizing-nested-loops-joins-and-understanding-their-implications/

Hash Join

- Build Phase:
  - For each row in table 1, calculate hash value of equi join columns.
  - Store row in a hash table, use calculated hash as key.
- Probe Phase:
  - For each row in table 2, calculate hash value of equi join columns.
  - Check if hash match in hash table, if so, check actual column
  - Output hash match
- Partition-hash join
  - Each file is partitioned into *M* partitions using the same partitioning hash function on the join attributes.
  - Each pair of corresponding partitions is joined.

Hash Join



https://bertwagner.com/posts/visualizing-nested-loops-joins-and-understanding-their-implications/

Implementing the JOIN Operation (cont'd.)

- Join selection factor
  - Fraction of records in one file that will be joined with records in another file
  - Depends on the equijoin condition with another file

Algorithms for PROJECT and Set Operations

- PROJECT operation
  - After projecting *R* on only the columns in the list of attributes, any duplicates are removed by treating the result strictly as a set of tuples
- Default for SQL queries
  - No elimination of duplicates from the query result
    - Duplicates eliminated only if the keyword DISTINCT is included

Algorithms for PROJECT and Set Operations

- Set operations
  - UNION
  - INTERSECTION
  - SET DIFFERENCE
  - CARTESIAN PRODUCT
- Set operations sometimes expensive to implement
  - Sort-merge technique
  - Hashing

Implementing Aggregate Operations and Different Types of JOINs

- Aggregate operators
  - MIN, MAX, COUNT, AVERAGE, SUM
  - Can be computed by a table scan or using an appropriate index
- Example:

```
SELECT   MAX(Salary)
FROM     EMPLOYEE;
```

  - If an (ascending) B_+ -tree index on Salary exists:
    - Optimizer can use the Salary index to search for the largest Salary value
    - Follow the rightmost pointer in each index node from the root to the rightmost leaf