Chapter 2

Query Optimization

By Fkee

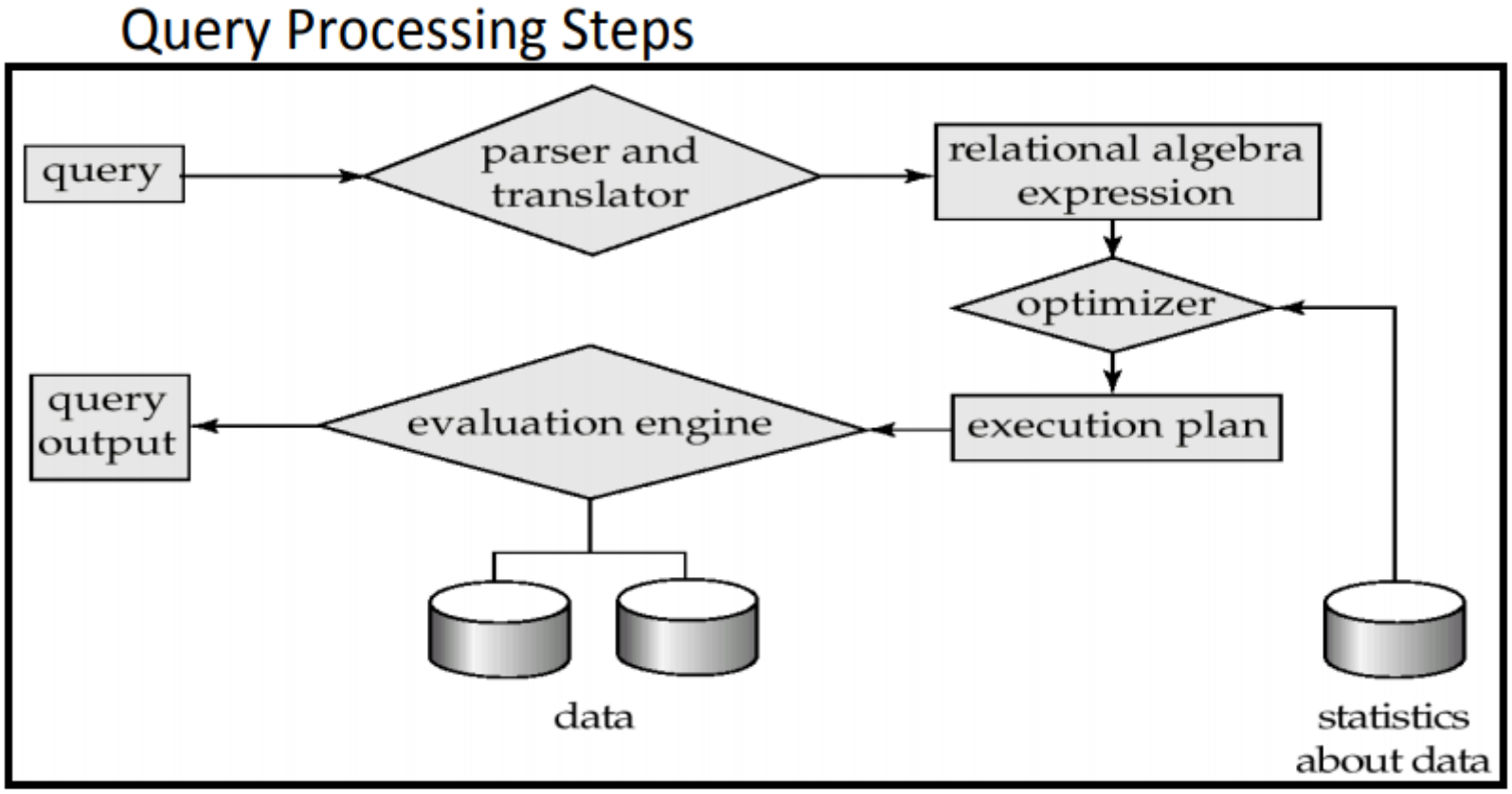
Query processing

- **Query processing:** activities involved in retrieving data from the database.
- Its about extracting data from a database.
- Activities of QP:
  - transform query written in high-level language (e.g. SQL), into correct and efficient execution strategy expressed in low-level language (implementing RA);
  - Query optimization (identifying best query)
  - Actual evaluation of queries(execute the query)

By Fkee

Query processing

- Query processing goes through various phases:



By Fkee

Query Analysis

- • During the query analysis phase, the query is syntactically analyzed using the programming language compiler (parser).
- • A syntactically legal query is then validated, using the system catalog, to ensure that all data objects (relations and attributes) referred to by the query are defined in the database.
- •Example: SELECT emp_nm FROM EMPLOYEE WHERE emp_desg>100
- This query will be rejected because the comparison “>100” is incompatible with the data type of emp_desg which is a variable character string

By Fkee

Query processing…..

- Internal query representation is created in either of the two internal representations of a query
  - **Query Tree**
  - **Query Graph**
- The DBMS must then plan an execution strategy and must choose a suitable execution strategy from many possible strategies. The process is known as **query optimization**.
- In general, QP has four main phases:
  - decomposition (consisting of scanning, parsing and validation)
  - optimization
  - code generation, and
  - execution.

By Fkee

Query Tree

- A Query Tree is a tree data structure that corresponds expression.
- • A Query Tree is also called a relational algebra tree.
- – Leaf node of the tree, representing the base input relations of the query.
- – Internal nodes result of applying an operation in the algebra. –
- Root of the tree representing a result of the query.

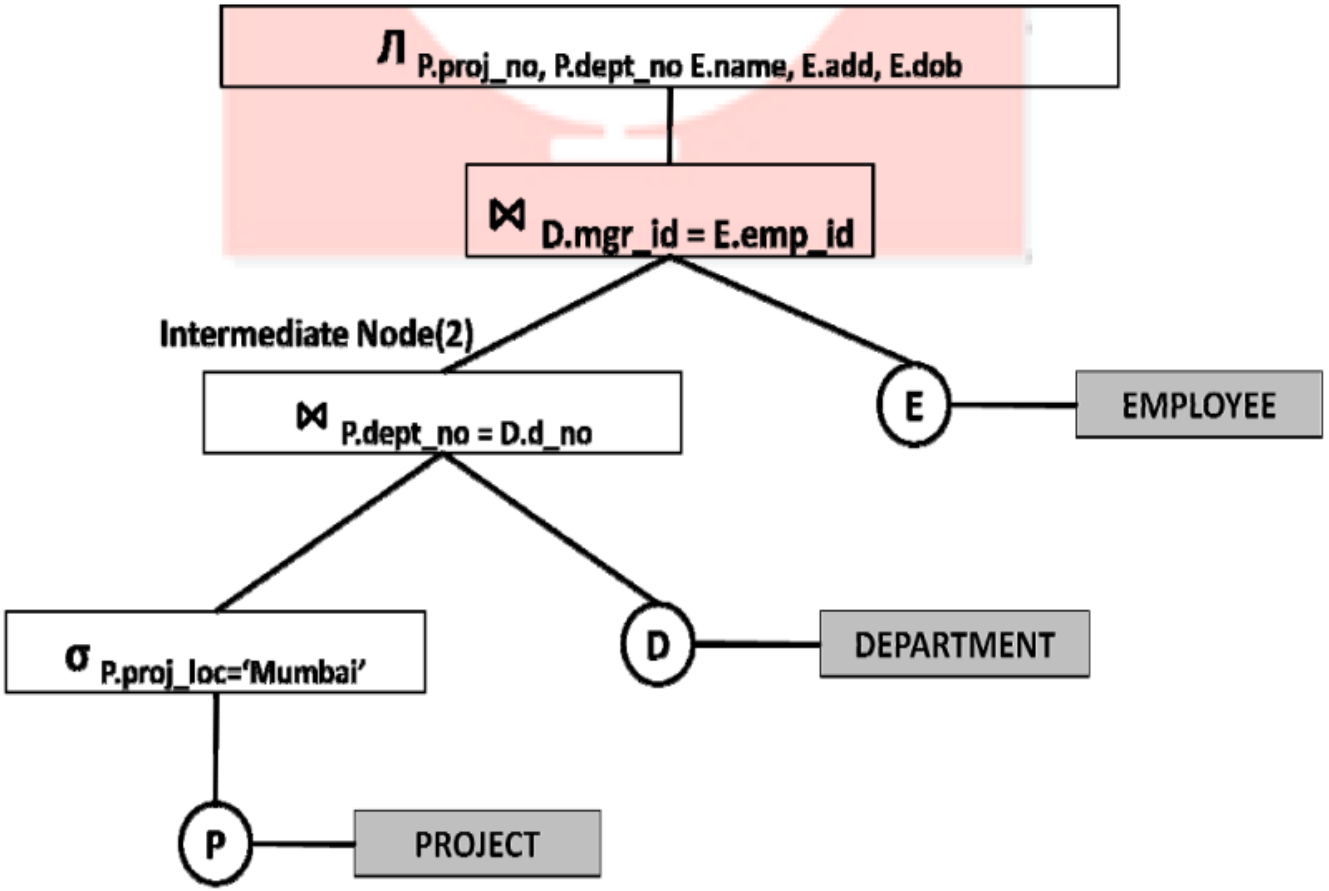
By Fkee

Query Tree

- Example:-

- SELECT (P.proj_no, P.dept_no, E.name, E.add, E.dob) • FROM PROJECT P, DEPARTMENT D, EMPLOYEE E • WHERE P.dept_no = D.d_no AND D.mgr_id = E.emp_id AND P.proj_loc = ‘Mumbai’ ;

Query Tree



By Fkee

Translating SQL Queries into Relational Algebra(RA)

- **Query block:** the basic unit that can be translated into the algebraic operators and optimized.
- A query block contains a single SELECT-FROM-WHERE expression, as well as GROUP BY and HAVING clause if these are part of the block.
- **Nested queries** within a query are identified as separate query blocks.
- **Aggregate operators**(such as: sum, min, average, max) in SQL must be included in the extended algebra.

By Fkee

Translating SQL…..

```
SELECT    LNAME, FNAME
FROM      EMPLOYEE
WHERE     SALARY > (  SELECT    MAX (SALARY)
                      FROM      EMPLOYEE
                      WHERE     DNO = 5);
```

```
SELECT    LNAME, FNAME
FROM      EMPLOYEE
WHERE     SALARY > C
```

$\pi_{\text{LNAME, FNAME}} (\sigma_{\text{SALARY>C}}(\text{EMPLOYEE}))$

```
SELECT    MAX (SALARY)
FROM      EMPLOYEE
WHERE     DNO = 5
```

$\pi_{\text{MAX SALARY}} (\sigma_{\text{DNO=5}} (\text{EMPLOYEE}))$

- The query optimizer would then choose an execution plan for each block.

Query Decomposition

- Aims are to transform high-level query into RA query and check that query is syntactically and semantically correct.
- Typical stages are:
  - Analysis:
    - ✓ Analyze query syntactically using compiler techniques.
    - ✓ Verify relations and attributes exist.
    - ✓ Verify operations are appropriate for object type.
  - Normalization:
    - ✓ Converts query into a normalized form for easier manipulation.

By Fkee

Query Optimization

- **Query Optimization:** is the process of choosing a suitable execution strategy for processing a query.
- It is the activity of choosing an efficient execution strategy for processing a query.
- As there are many equivalent transformations of same high-level query, the aim of QO is to choose one that minimizes resource usage.
  - reduce total execution time of query.
  - may also reduce response time of query.
- Eg:- Find all Managers who work at a London branch.

```
SELECT *
FROM Staff s, Branch b
WHERE s.branchNo = b.branchNo AND (s.position = 'Manager' AND b.city = 'London')
```

By Fkee

Query optimization…..

- Three equivalent RA queries are:

  1) σ(position='Manager') $\wedge$ (city='London') $\wedge$ (Staff.branchNo=Branch.branchNo) **(Staff X Branch)**

  2) σ(position='Manager') $\wedge$ (city='London')(**Staff** $\bowtie$ Staff.branchNo=Branch.branchNo **Branch)**

  3) (σposition='Manager'**(Staff)**) $\bowtie$ Staff.branchNo=Branch.branchNo (σcity='London' **(Branch))**

- Two main techniques for implementing query optimization:
  - heuristic rules that order operations in a query execution strategy;
  - comparing different strategies based on relative costs, and selecting one that minimizes resource usage.

By Fkee

Using Heuristics in Query Optimization

- Process for heuristics optimization

1. The parser of a high-level query generates an initial internal representation;
2. Apply heuristics rules to optimize the internal representation.
3. A query execution plan is generated to execute groups of operations based on the access paths available on the files involved in the query.

Steps in converting query tree in heuristic optimization

- 1.initial (canonical)query tree for sql query
- 2. moving selection operation down query tree
- Applying more restrictive select operation first
- Replacing Cartesian product and select with join
- Moving project operation down query tree

By Fkee

Using Heuristics…..

- **Query tree**:
  - A tree data structure that corresponds to a **relational algebra** expression. It represents the input **relations** of the query as **leaf nodes** of the **tree**, and represents the relational algebra **operations** as **internal nodes.**
- An execution of the query tree consists of executing an **internal node operation** whenever its operands are available and then replacing that internal node by the relation that results from executing the operation.
- **Query graph**:
  - A graph data structure that corresponds to a **relational calculus** expression. It does ***not*** indicate an order on which operations to perform first. There is only a ***single*** graph corresponding to each query.

By Fkee

Using Heuristics…..

- Notation for Query Trees and Query Graphs
- Example:
  - Consider three tables course(CNO,CNAME,CREDITS), STUDENT(ROLLNO,NAME,ADRESS) and ENROLLMENT(CNO,ROLLNO,GRADE).
  - Write Algebric expression and query tree for the following SQL expression using heuristics.

SELECT S.NAME, S.ADDRESS, E.GRADE FROM COURSE C,STUDENT S, ENROLLMENT E, WHERE S.ROLLNO =E.ROLLNO AND C.CNO =E.CNO AND CNAME='DB'

COURSE(CNO,CNAME,CREDIT)

STUDENT(ROLLNO,NAME,ADDRESS,SEM)

ENROLLMENT(CNO,ROLLNO,GRADE)

By Fkee

Using Heuristics…..

- SELECT S.NAME,S.ADDRESS,E.GRADE
- FROM COURSE C, STUDENT S, ENROLLMENT E

- WHERE S.ROLLNO = E.ROLLNO AND C.CNO=E.CNO AND CNAME = ‘DB’

By Fkee

Using Heuristics…..

- Notation for Query Trees and Query Graphs
- Example:
  - For every project located in ‘Stafford’, retrieve the project number, the controlling department number and the department manager’s last name, address and birth date.

- Relation algebra:

$$\pi_{\text{PNUMBER, DNUM, LNAME, ADDRESS, BDATE}}(((\sigma_{\text{PLOCATION='STAFFORD'}}(\text{PROJECT})) \bowtie_{\text{DNUM=DNUMBER}} (\text{DEPARTMENT})) \bowtie_{\text{MGRSSN=SSN}} (\text{EMPLOYEE}))$$

- SQL query:

```
Q2: SELECT   P.PNUMBER, P.DNUM, E.LNAME, E.ADDRESS, E.BDATE
    FROM     PROJECT AS P, DEPARTMENT AS D,  EMPLOYEE AS E
    WHERE    P.DNUM=D.DNUMBER AND D.MGRSSN=E.SSN AND
                                P.PLOCATION='STAFFORD';
```

By Fkee

Using Heuristics…..

- Notation for Query Trees and Query Graphs…..

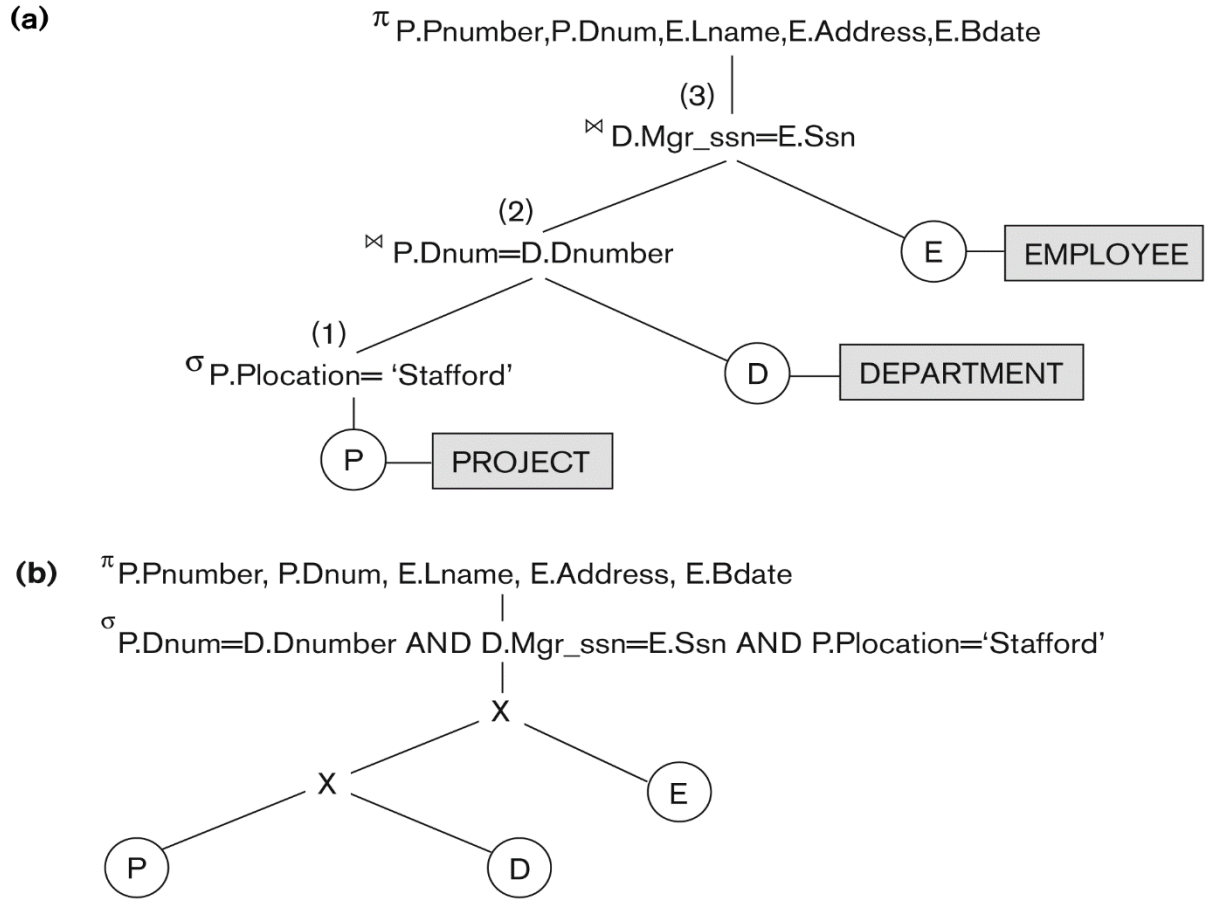
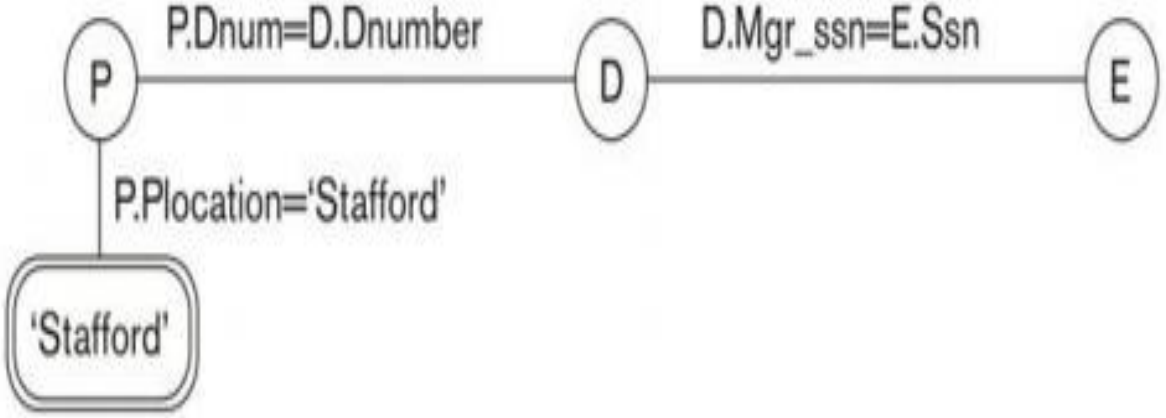


Figure 15.4
Two query trees for the query Q2. (a) Query tree corresponding to the relational algebra expression for Q2. (b) Initial (canonical) query tree for SQL query Q2. (c) Query graph for Q2.

(c) [P.Pnumber, P.Dnum] [E.Lname, E.Address, E.Bdate]



By Fkee

Using Heuristics…..

- Heuristic Optimization of Query Trees:
  - The same query could correspond to many different **relational algebra** expressions, and hence many different query trees.
  - A query tree can be transformed step by step into another query tree that is more efficient to execute.
  - The task of heuristic optimization of query trees is to find a **final query tree** that is efficient to execute.
  - Example: "Find the last names of employees born after 1957 who work on a project named 'Aquarius'."

Consider table

- Employee(fname, lname, ssn, bdate,dno)
- Project (pname, pnumber, plocation)
- works_on(essn,pno,hours)

```
Q:  SELECT      LNAME
    FROM   EMPLOYEE, WORKS_ON, PROJECT
    WHERE       PNAME = 'AQUARIUS' AND
                  PNMUBER=PNO AND ESSN=SSN
                        AND BDATE > '1957-12-31';
```

By Fkee

Using Heuristics…..

- The main heuristic is to apply first the operations that reduce the size of intermediate results.
  - E.g: Apply SELECT and PROJECT operations before applying the JOIN or other binary operations.
- Perform **select** operations as early as possible to reduce the number of **tuples** and perform **project** operations as early as possible to reduce the number of **attributes.** (This is done by moving select and project operations as far down the tree as possible.)
- The **select and join operations** that are most restrictive should be executed before other similar operations. (This is done by reordering the leaf nodes of the tree among themselves and adjusting the rest of the tree appropriately.)

By Fkee

Using Heuristics….

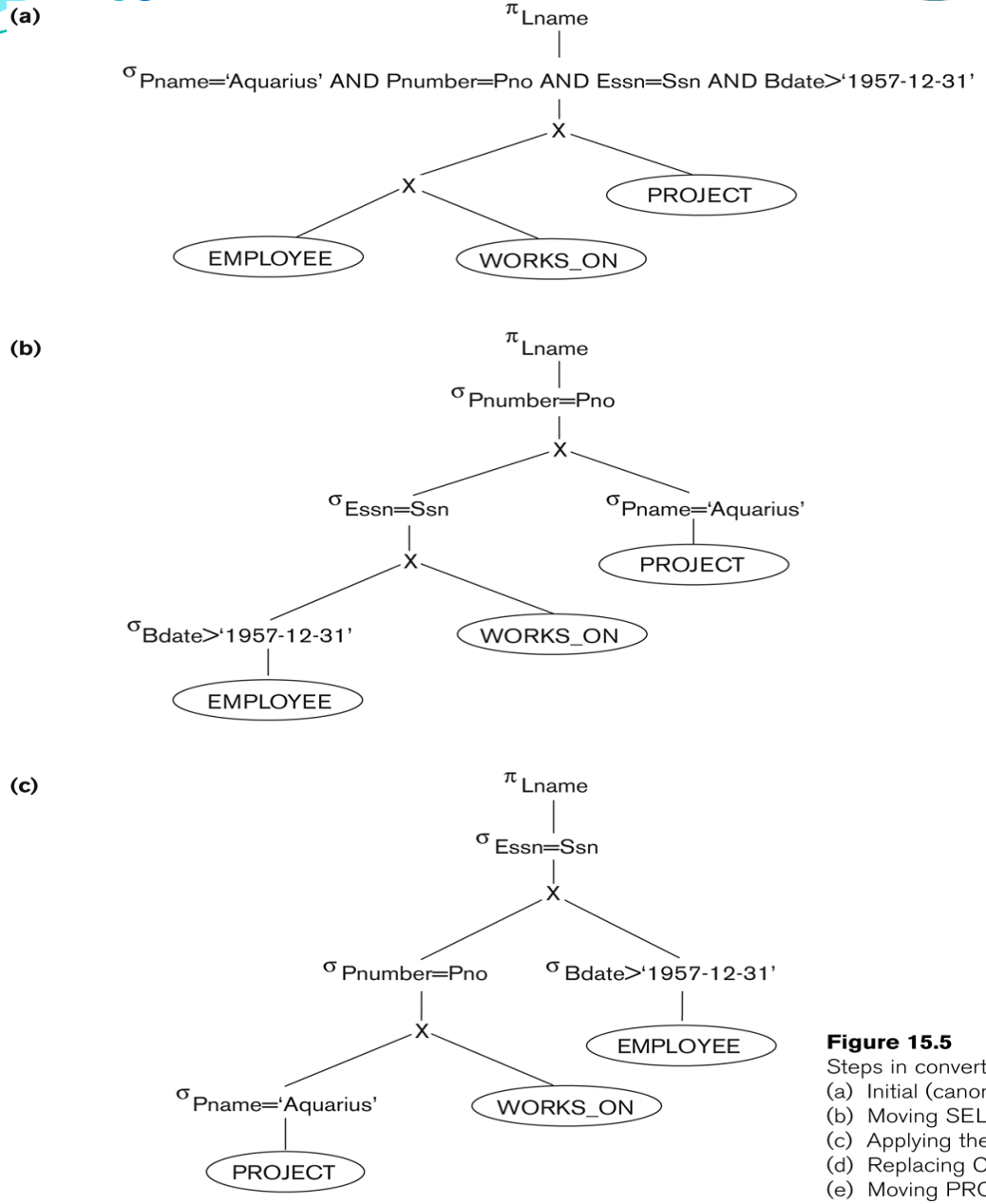
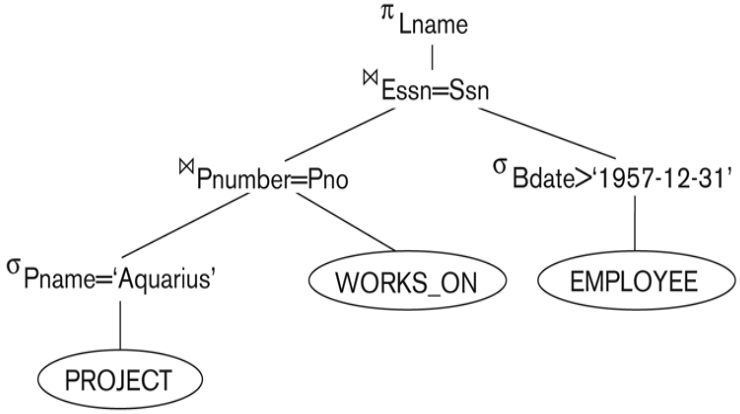


Figure 15.5
Steps in converting a query tree during heuristic optimization.
(a) Initial (canonical) query tree for SQL query Q.
(b) Moving SELECT operations down the query tree.
(c) Applying the more restrictive SELECT operation first.
(d) Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.
(e) Moving PROJECT operations down the query tree.

(d)



(e)

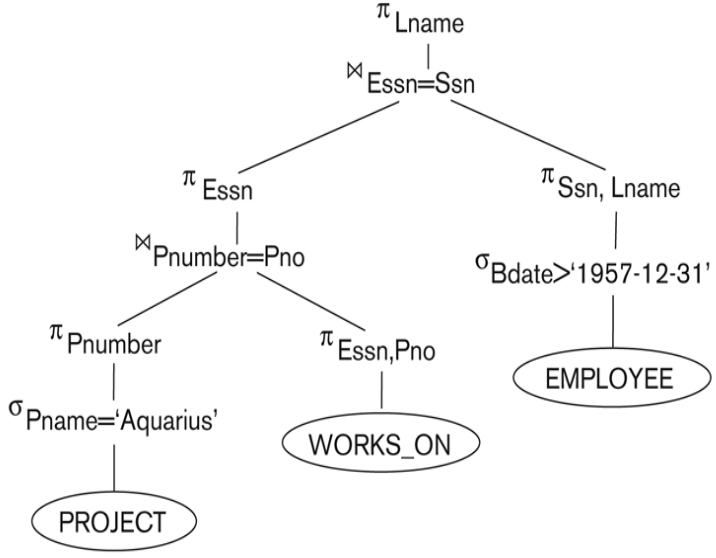


Figure 15.5
Steps in converting a query tree during heuristic optimization.
(a) Initial (canonical) query tree for SQL query Q.
(b) Moving SELECT operations down the query tree.
(c) Applying the more restrictive SELECT operation first.
(d) Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.
(e) Moving PROJECT operations down the query tree.

By Fkee

Measure of query cost

- The query evaluation cost is dependent on different recourses. A),Disk Access , B),CPU time to excute the query code. C), cost of data transmission.
- A),Disk Acess: the cost to acess data from disk is the most important for large database since the disk acess are slow when we compare to main memory acess.
- B) CPU time: time to excute the query code. Generally disk access time must be > cpu time, hence we ignore cpu time and consider disk access cost to measure the query evaluation cost.

By Fkee

Measure of query cost

- Disk access: to measure the cost of the data access from disk we consider
- Number of block transfer and number of disk seek from the disk

Con't

- Disk Access: to measure of the cost the data acces from the disk we consider
- Number of block transfer and number of disk seek from the disk
- tT: Avg time to transfer a single block data(in sec)
- tS: Avg block access time (disk seek) + rotation latency(in sec)
- E.g let no of block be b1 and no of seeks be se, then
- The cost =(b1 x tT + se x tS)sec

Evaluation plan

- Evaluation plan is needed to define exactly what algorithm should be used for each operation. And how the execution of the operation should be coordinated

By Fkee

Cost Based Query Optimization

- A query optimizer should not depend solely on **heuristic rules**; it should also estimate and compare the **costs** of executing a query using different execution strategies and should choose the strategy with the ***lowest cost estimate***. This approach is known as **cost-based query optimization.**
- Issues to be considered
  - Cost function
  - Number of execution strategies

By Fkee

Cost Based…..

- Cost Components for Query Execution

1. Access cost to secondary storage
   - ✓ is the cost of searching for, reading, and writing data blocks that reside on secondary storage, mainly on disk.
2. Storage cost
   - ✓ is the cost of storing any intermediate files that are generated by an execution strategy for the query.
3. Computation cost
   - ✓ is the cost of performing in-memory operations on the data buffers during query execution(such as operations searching, sorting, merging,…).
4. Memory usage cost
   - ✓ is the cost pertaining to the number of memory buffers needed during query execution.
5. Communication cost
   - ✓ is the cost of shipping the query and its results from the database site to the site or terminal where the query originated.

Cost Based…..

- Different database systems may focus on different cost components.
- In large databases,
  - the main emphasis is on minimizing the access cost to secondary storage.
  - comparing different query execution strategies in terms of the number of block transfers between disk and main memory.
- In smaller databases,
  - most of the data in the files involved in the query can be completely stored in memory.
  - the emphasis is on minimizing computation cost.
- In distributed databases,
  - where many sites are involved, communication cost must be minimized.

Cost Based…..

- Catalog Information Used in Cost Functions
  - Information about the size of a file
    - number of records (tuples) (r),
    - record size (R),
    - number of blocks (b)
    - blocking factor (bfr)
  - Information about indexes and indexing attributes of a file
    - Number of levels (x) of each multilevel index (Primary, secondary, or clustering)
    - Number of first-level index blocks (bI1)
    - Number of distinct values (d) of an attribute
    - Selectivity (sl) of an attribute
    - Selection cardinality (s) of an attribute. (s = sl * r)

By Fkee

Materialization and Pipelining

- An execution plan for a relational algebra query consists of a combination of the relational algebra query tree and information about the access methods to be used for each relation as well as the methods to be used in computing the relational operators stored in the tree.
- **Materialization:** it store intermediate result into the disk
- the result of an operation is stored as a temporary relation for processing by next (that is, the result is *physically materialized).*

By Fkee

Materialization and Pipelining

- Example : evaluation of expression
- (r ⋈ s) n (T)
- t1 ⟵ r ⋈ s
- t2 ⟵ t n t1

T1:- store intermediate value to the disk.

It store the temporary relation on secondary memory

By Fkee

Materialization and Pipelining…..

- **Pipelining**: as the result of an operator is produced, it is forwarded to the next operator without storing in temporary relation.
  - The results of one operation could also be pipelined to another without creating temporary relation.
  - It is also known as on-the-fly processing.
- The advantage of pipelining is the **cost savings** in **not having to write** the intermediate results to disk and **not having to read** them **back** for the next operation.
- It store to buffer and passing as input to next operation

By Fkee

Rule Based Optimization

- **Rule-based query optimization:** the optimizer chooses execution plans based on heuristically ranked operations.
  - Currently, the rule-based approach is being phased out.
- **Cost-based query optimization:** the optimizer examines alternative access paths and operator algorithms and chooses the execution plan with lowest estimated cost.
  - The query cost is calculated based on the estimated usage of resources such as I/O, CPU and memory needed.
- Application developers could also specify hints to the some DBMS query optimizers.
- The idea is that an application developer might know more information about the data.

By Fkee

Semantic Query Optimization

- A different query optimization which have been suggested.
- It uses constraints specified on the database schema in order to modify one query into another query that is more efficient to execute.
- Consider the following SQL query,

```
SELECT    E.LNAME, M.LNAME
FROM      EMPLOYEE E M
WHERE     E.SUPERSSN=M.SSN AND E.SALARY>M.SALARY
```

- Suppose that we had a constraint on the database schema that stated that no employee can earn more than his or her direct supervisor. If the semantic query optimizer checks for the existence of this constraint, it need not execute the query at all because it knows that the result of the query will be empty.

By Fkee

Thank You

Hope you understand?

Please refer extra books for more…

By Fkee