COSC344
Database Theory and Applications

Lecture 24
Query Optimisation

Overview

- Last Lecture
  - Security
- This Lecture
  - Query Optimisation
  - Source: Chapter 18
- Next Lecture
  - Non-relational data models

DBMS Component Modules
Where are we now?

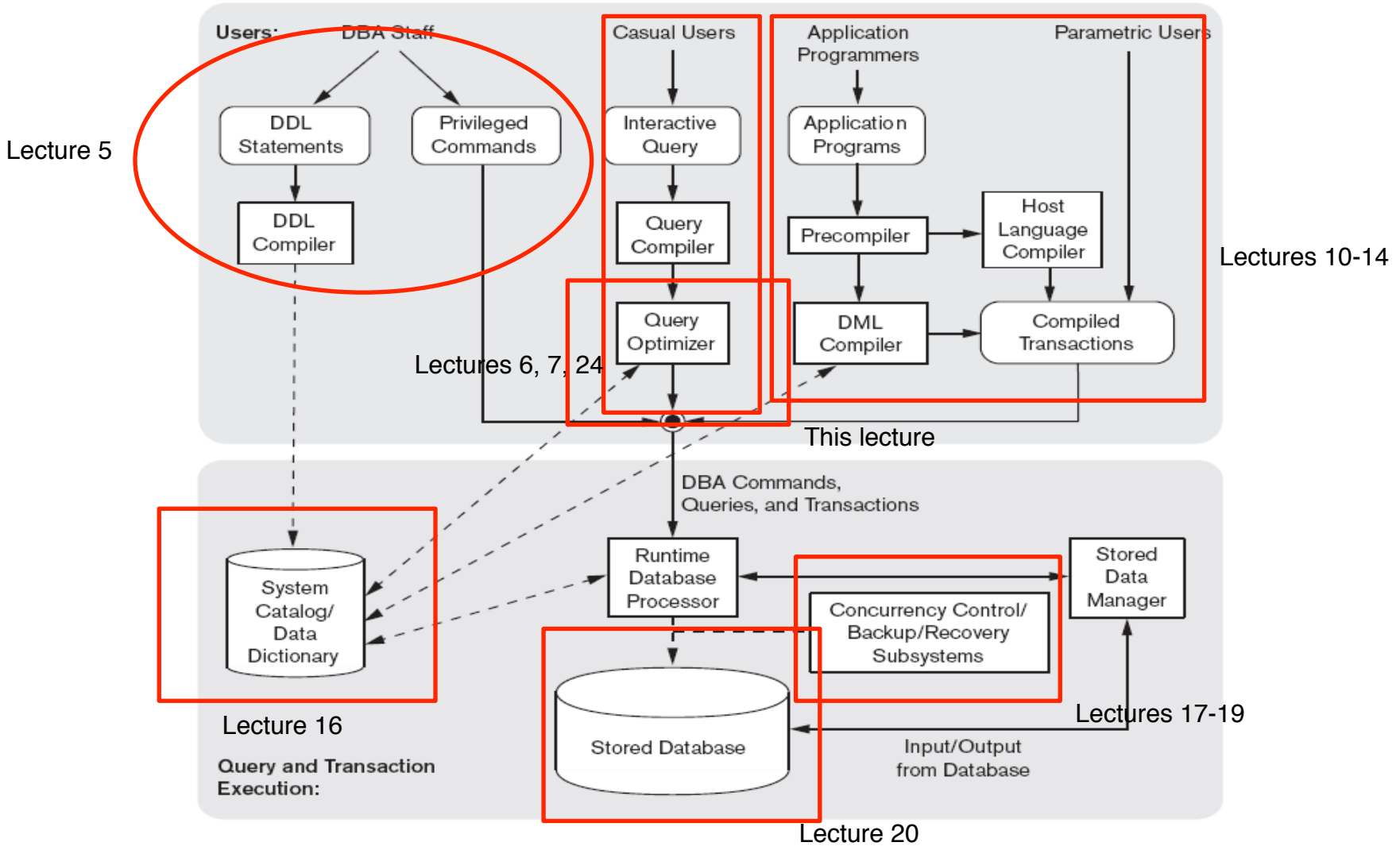


Figure 2.3
Component modules of a DBMS and their interactions.

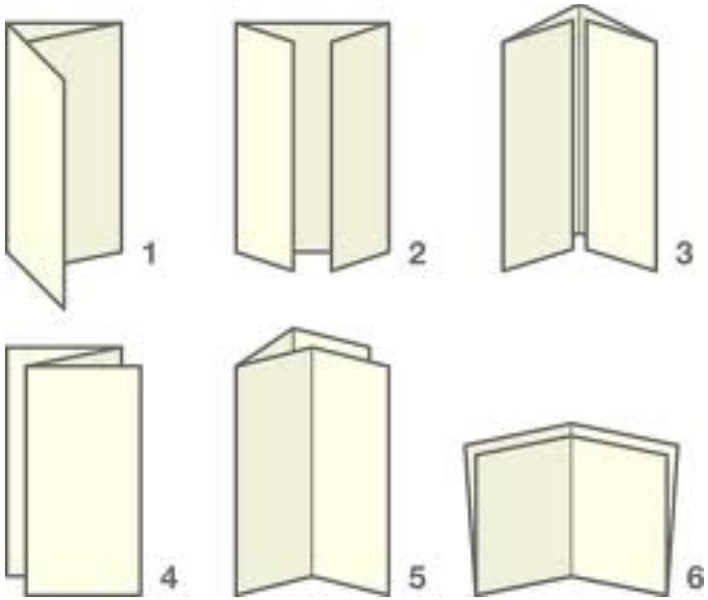
Query Optimization

```
Select  A1,A2,…,An
From    R1,R2, …,Rm
Where   condition
```

← what to return

← relations

← combine filter

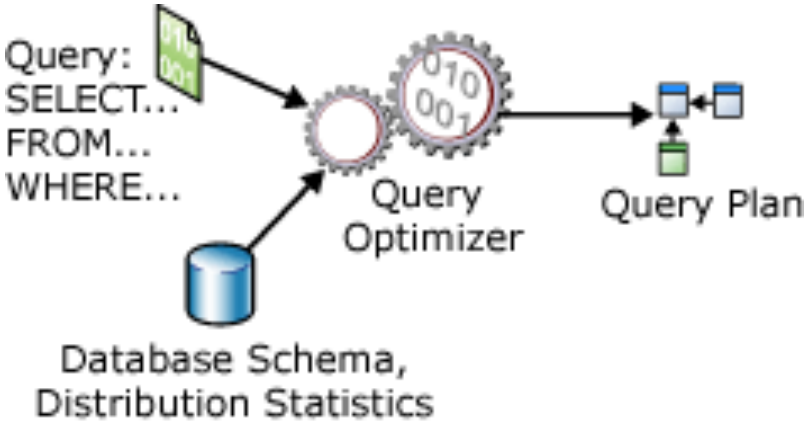


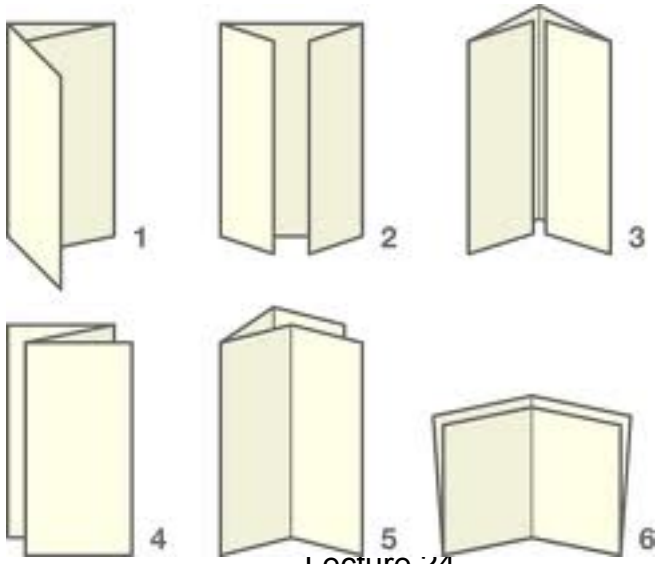
Example Query

Get the salesperson's name, customer's name, order number, and amount where the rating is less than 300 and the amount is greater than 4000.

```
SELECT sname, cname, onum, amt
   FROM salespeople s,
        customers c, orders o
   WHERE  s.snum = c.snum AND
          c.cnum = o.cnum AND
          c.rating < 300 AND
          o.amt > 4000;
```

```
SELECT sname, cname, onum, amt
   FROM salespeople s,
        customers c, orders o
   WHERE  c.rating < 300 AND
          o.amt > 4000 AND
          s.snum = c.snum AND
          c.cnum = o.cnum;
```



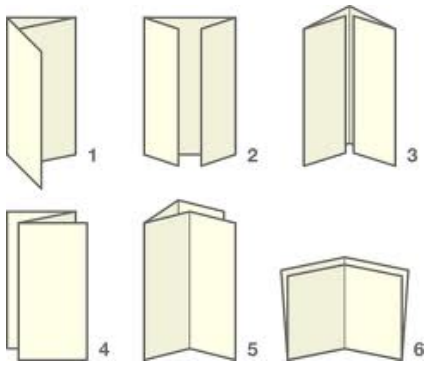


Example Query(cont.)

```
SELECT *
FROM salespeople s, customers c, orders o
WHERE  s.snum = c.snum;
```

```
 SNUM  SNAME    CITY    COMM      CNUM CNAME    CITY   RATING  SNUM     ONUM  AMT     ODATE    CNUM    SNUM
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
 1004  Motika   London  .11      2007 Pereira  Rome    100    1004      3003 767.19 03/10/90 2001    1001
 1004  Motika   London  .11      2007 Pereira  Rome    100    1004      3001 18.69  03/10/90 2008    1007
 . . . . . .

70 rows selected.      (14 columns)
```

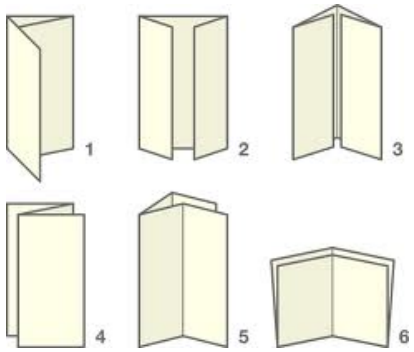



```
SELECT *
FROM salespeople s, customers c, orders o
WHERE  s.snum = c.snum AND
       c.cnum = o.cnum;
```

```
 SNUM  SNAME    CITY    COMM    CNUM  CNAME    CITY      RATING  SNUM     ONUM  AMT     ODATE     CNUM  SNUM
 ------ ---- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
1007  Rifkin  Barcelona  .15  2008 Cisneros  San Jose  300    1007     3001  18.69  03/10/90  2008  1007
1001  Peel    London     .12  2001 Hoffman   London    100    1001     3003  767.19  03/10/90 2001  1001
. . . . . .
10 rows selected.  (still 14 columns)
```

Example Query(cont.)

```
SELECT *
FROM salespeople s, customers c, orders o
WHERE  s.snum = c.snum AND
       c.cnum = o.cnum AND
       c.rating < 300 AND
       o.amt > 4000;
```



SNUM	SNAME	CITY	COMM	CNUM	CNAME	CITY	RATING
SNUM	ONUM	AMT	ODATE	CNUM	SNUM		
1002	Serres	San Jose	.13	2003	Liu	San Jose	200
1002	3005	5160.45	03/10/90	2003	1002		
1001	Peel	London	.12	2006	Clemens	London	100
1001	3008	4723	05/10/90	2006	1001		
1001	Peel	London	.12	2006	Clemens	London	100
1001	3011	9891.88	06/10/90	2006	1001		

3 rows, and 14 columns as before

Example Query(cont.)

```
SELECT sname, cname, onum, amt
FROM salespeople s, customers c, orders o
WHERE  s.snum = c.snum AND
       c.cnum = o.cnum AND
       c.rating < 300 AND
       o.amt > 4000;
```

SNAME	CNAME	ONUM	AMT
Serres	Liu	3005	5160.45
Peel	Clemens	3008	4723
Peel	Clemens	3011	9891.88

3 rows and 4 columns

largest intermediate result table 14 col X 70 rows

Example Query(cont.)

```
SELECT cname, onum, amt
FROM   orders o, customers c
WHERE  o.amt > 4000 AND
       c.rating < 300;
```

CNAME	ONUM	AMT
Hoffman	3005	5160.45
Hoffman	3008	4723
Hoffman	3011	9891.88
Giovanni	3005	5160.45
Giovanni	3008	4723
Giovanni	3011	9891.88
Liu	3005	5160.45
Liu	3008	4723
Liu	3011	9891.88
Clemens	3005	5160.45
Clemens	3008	4723
Clemens	3011	9891.88
Pereira	3005	5160.45
Pereira	3008	4723
Pereira	3011	9891.88

15 rows selected, and 3 columns.

Example Query(cont.)

```
SELECT sname, cname, onum, amt
FROM   salespeople s, orders o, customers c
WHERE  o.amt > 4000 AND
       c.rating < 300 AND
       s.snum = c.snum;
```

SNAME	CNAME	ONUM	AMT
Peel	Hoffman	3005	5160.45
Peel	Hoffman	3008	4723
Peel	Hoffman	3011	9891.88
Peel	Clemens	3005	5160.45
Peel	Clemens	3008	4723
Peel	Clemens	3011	9891.88
Serres	Liu	3005	5160.45
Serres	Liu	3008	4723
Serres	Liu	3011	9891.88
Axelrod	Giovanni	3005	5160.45
Axelrod	Giovanni	3008	4723
Axelrod	Giovanni	3011	9891.88
Motika	Pereira	3005	5160.45
Motika	Pereira	3008	4723
Motika	Pereira	3011	9891.88

15 rows selected.

Example Query(cont.)

```
SELECT sname, cname, onum, amt, c.cnum, o.cnum
FROM  salespeople s, orders o, customers c
WHERE  o.amt > 4000 AND
       c.rating < 300 AND
       s.snum = c.snum;
```

SNAME	CNAME	ONUM	AMT	CNUM	CNUM
Peel	Hoffman	3005	5160.45	2001	2003
Peel	Hoffman	3008	4723	2001	2006
Peel	Hoffman	3011	9891.88	2001	2006
Peel	Clemens	3005	5160.45	2006	2003
Peel	Clemens	3008	4723	2006	2006
Peel	Clemens	3011	9891.88	2006	2006
Serres	Liu	3005	5160.45	2003	2003
Serres	Liu	3008	4723	2003	2006
Serres	Liu	3011	9891.88	2003	2006
Axelrod	Giovanni	3005	5160.45	2002	2003
Axelrod	Giovanni	3008	4723	2002	2006
Axelrod	Giovanni	3011	9891.88	2002	2006
Motika	Pereira	3005	5160.45	2007	2003
Motika	Pereira	3008	4723	2007	2006
Motika	Pereira	3011	9891.88	2007	2006

15 rows selected.

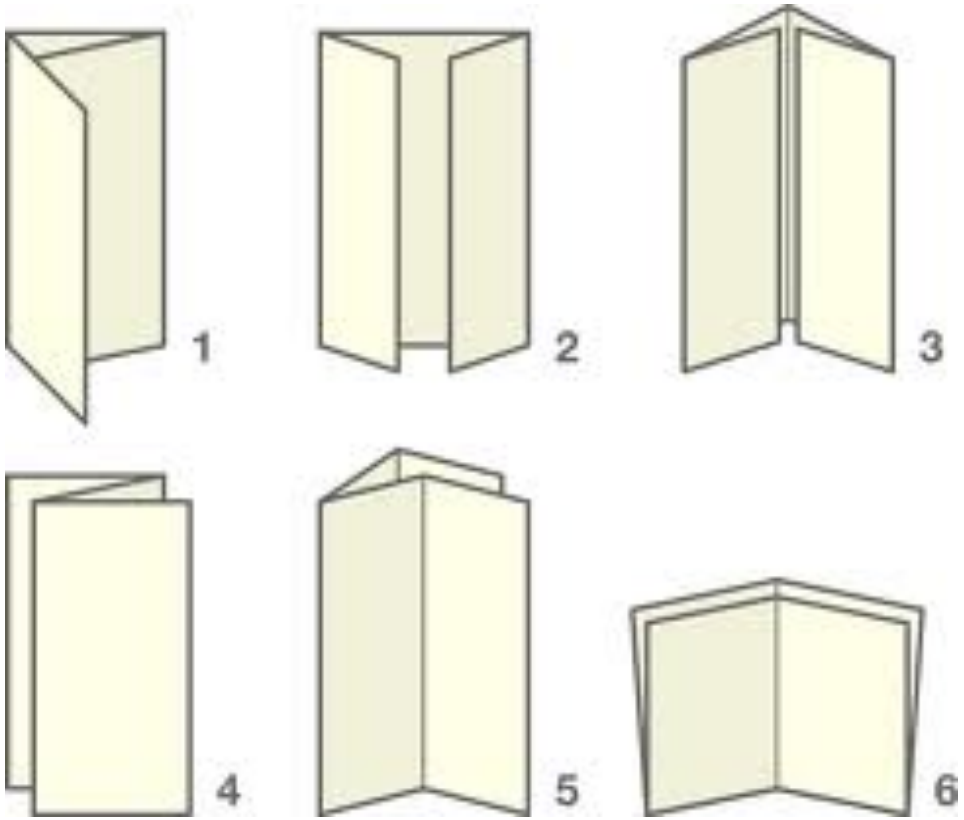
Example Query(cont.)

```
SELECT sname, cname, onum, amt
FROM  salespeople s, orders o, customers c
WHERE  o.amt > 4000 AND
       c.rating < 300 AND
       s.snum = c.snum AND
       c.cnum = o.cnum;
```

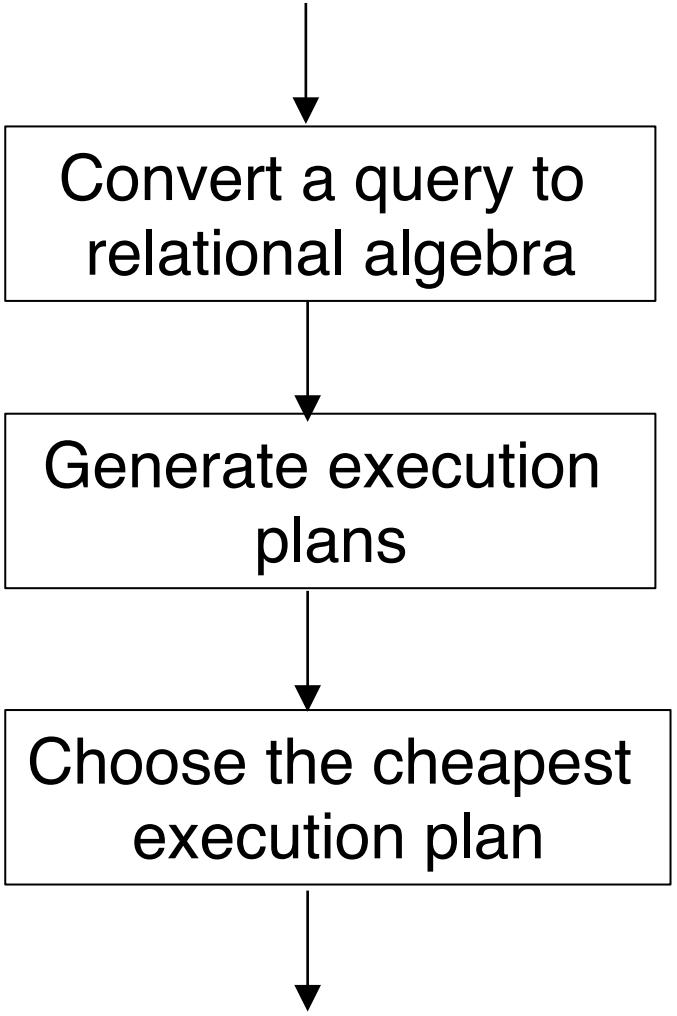
SNAME	CNAME	ONUM	AMT
Serres	Liu	3005	5160.45
Peel	Clemens	3008	4723
Peel	Clemens	3011	9891.88

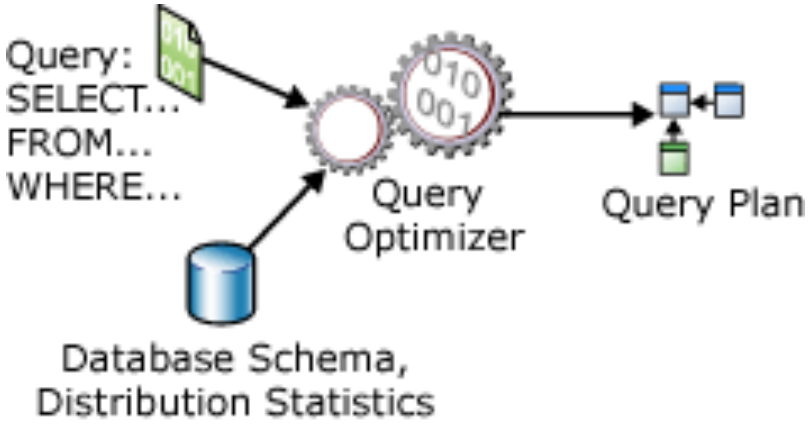
largest intermediate result table 4 (6) col X 15 rows

What is the best plan for query?



Optimisation Process





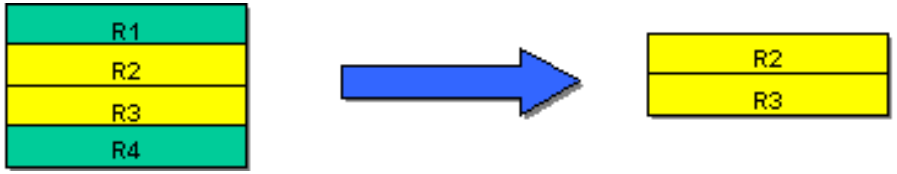
Two main techniques

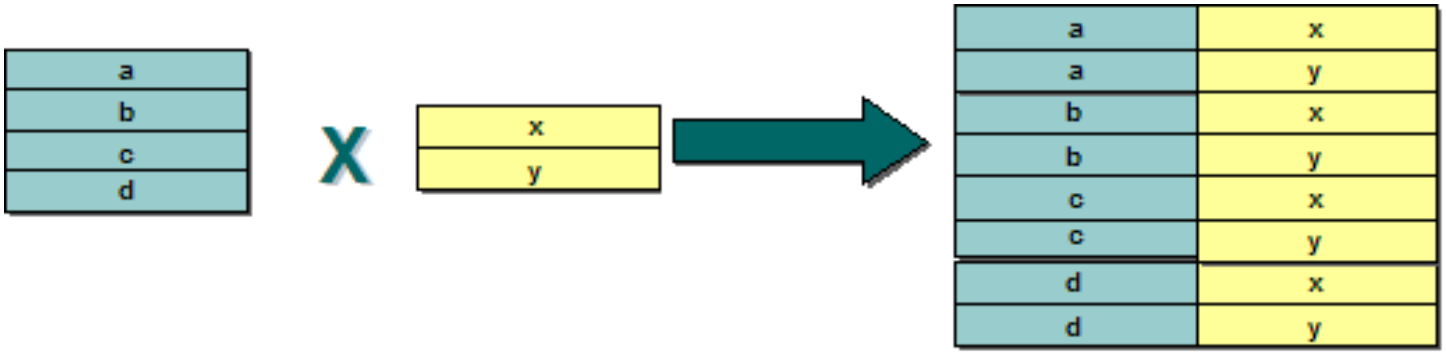
Heuristic rules

Systematic cost estimation

Convert to a Relational Algebra

- Relational Algebra
  - Select $\sigma_{<\text{selection condition}>}(R)$
  - Project $\Pi_{<\text{attribute list}>}(R)$
  - CARTESIAN Product R X S





Convert to a Relational Algebra

- Relational Algebra
  - Join $R \bowtie_{\text{<join condition>}} S$
  - Division $R \div S$

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

*

Enrol

SID	Course
3936	101
1108	113
8507	101

Result

Id	Name	Suburb	Course
1108	Robert	Kew	113
3936	Glen	Bundoora	101
8507	Norman	Bundoora	101

Subject

Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS

÷

Course

Course
BCS
MCS

Result

Name
Database

Convert to a Relational Algebra

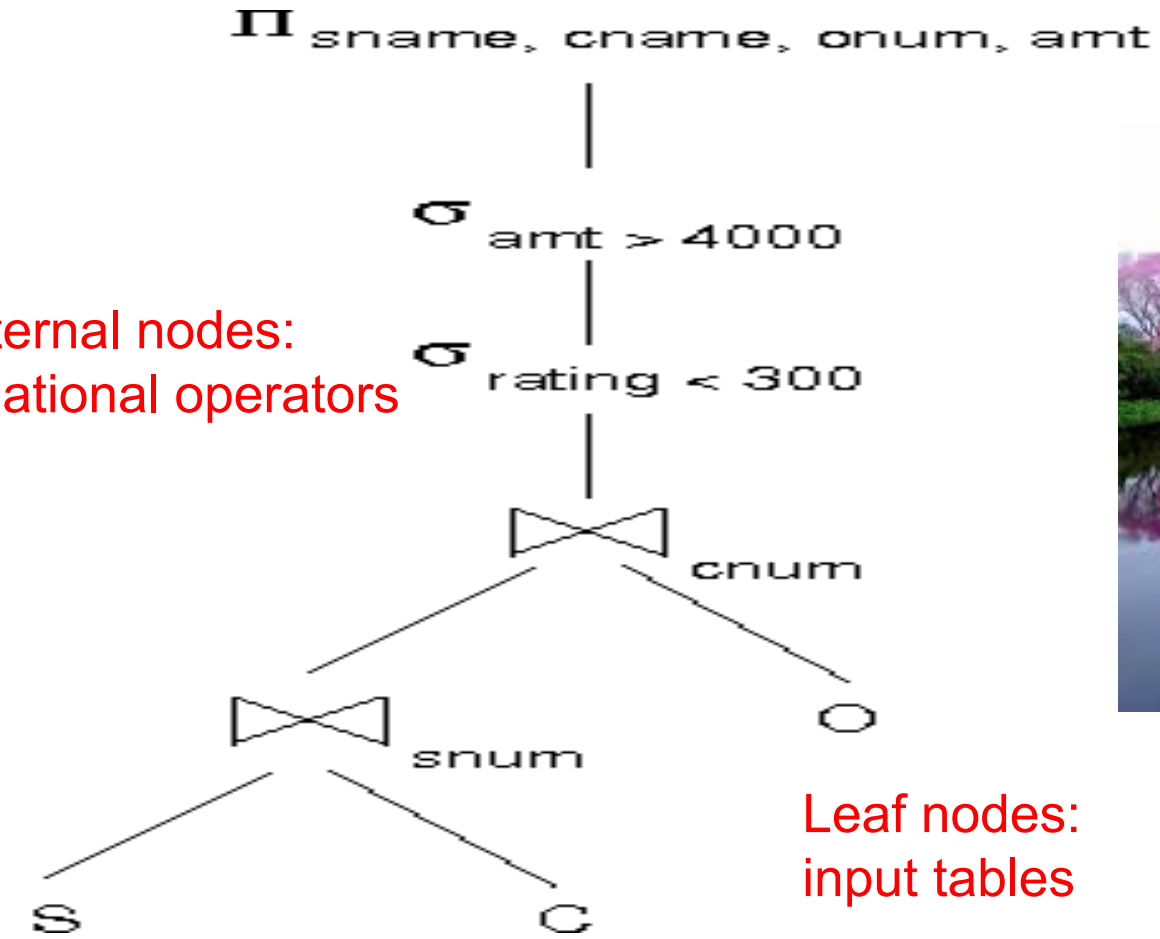
- Relational Algebra
  - Select $\sigma_{\text{<selection condition>}}(R)$
  - Project $\Pi_{\text{<attribute list>}}(R)$
  - CARTESIAN Product R X S
  - Join $R \bowtie_{\text{<join condition>}} S$
  - Division $R \div S$
- Example


```
SELECT sname, cname, onum, amt
FROM salespeople s, customers c,
orders o
WHERE  s.snum = c.snum AND
       c.cnum = o.cnum AND
       c.rating < 300 AND
       o.amt > 4000;
```

$$\Pi_{\text{sname,cname,onum,amt}}(\sigma_{\text{rating<300 and amt>4000}}(S \bowtie_{\text{snum}} C \bowtie_{\text{cnum}} O))$$

Query Tree

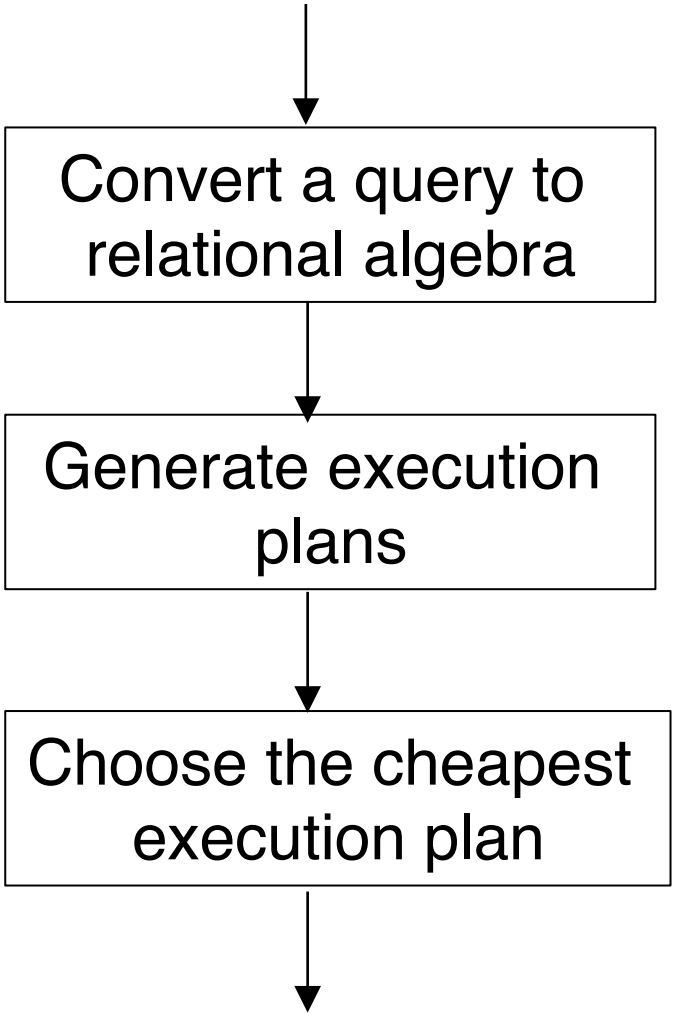
- A tree data structure that corresponds to a relational algebra expression.
- The order of execution starts at the leaf nodes and ends at the root node.

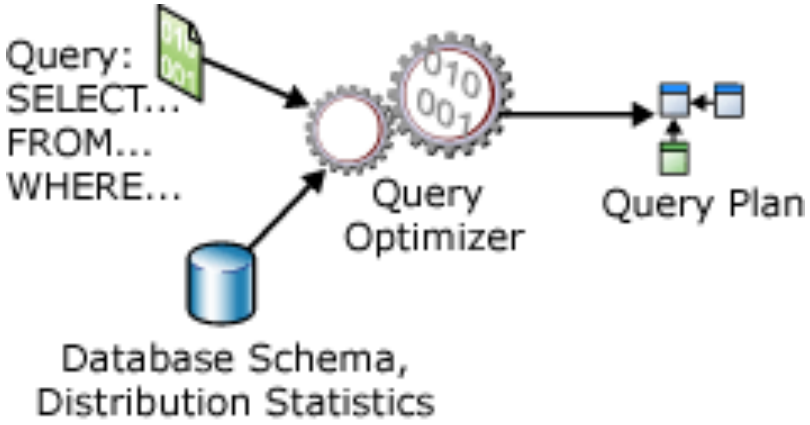


Internal nodes:
relational operators

Leaf nodes:
input tables

Optimisation Process





Two main techniques

Heuristic rules

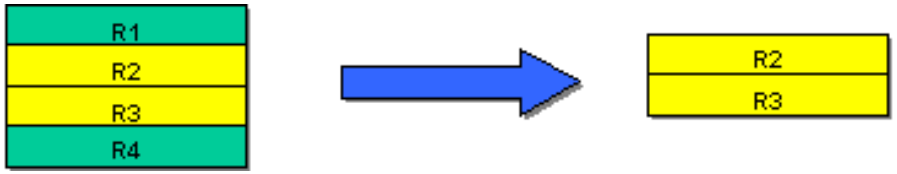
Systematic cost estimation

Implementing SELECT

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

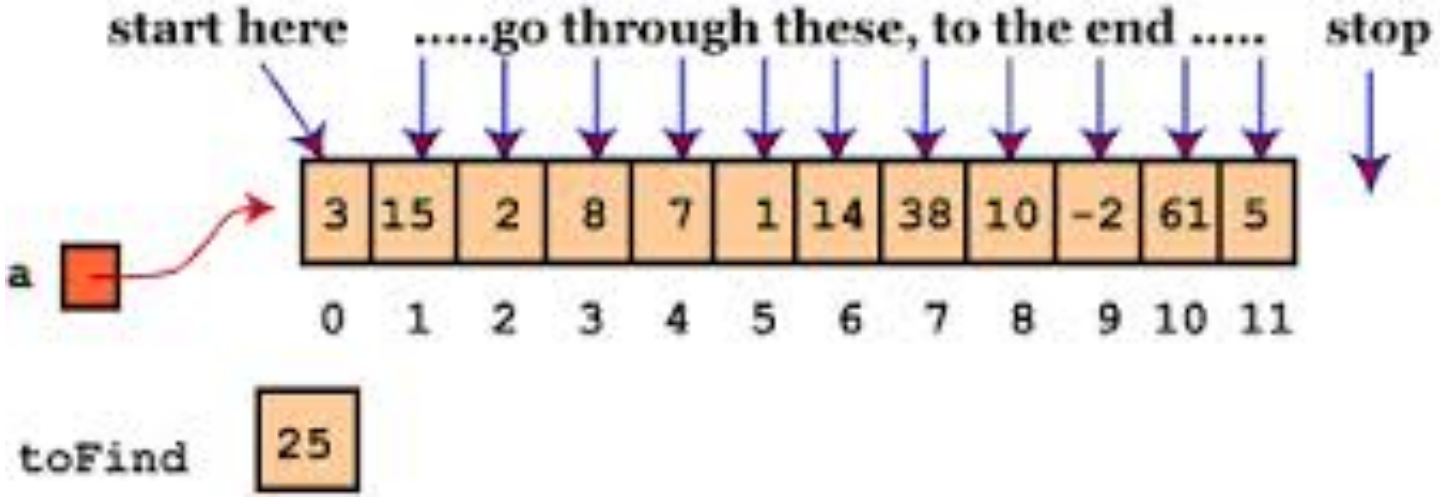
(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



Algorithms for simple selection

1. Linear search

   retrieve every record and test whether it satisfies the selection condition

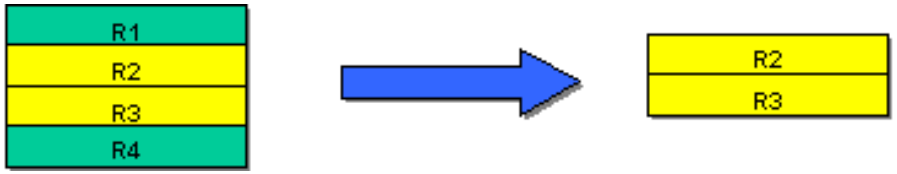


Implementing SELECT

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

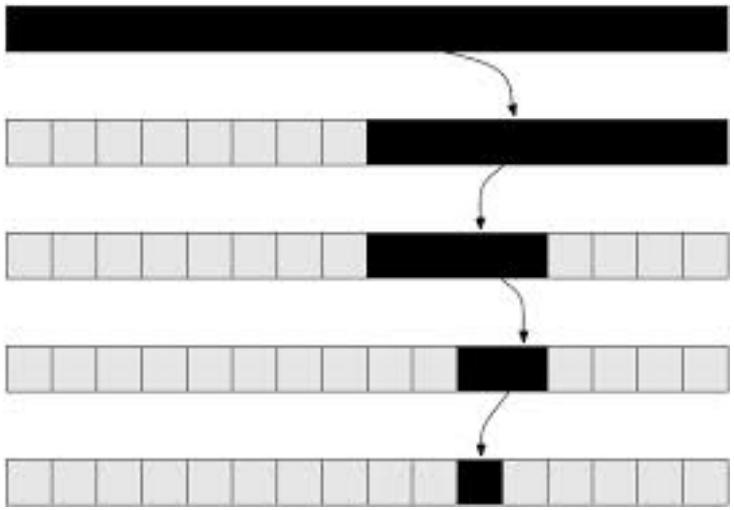
(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



Algorithms for simple selection

2. Binary search

If the search condition involves an equality comparison on a key attribute on which the file is ordered, a binary search can be used. Example is OP1 *if SSN is the ordering attribute.*

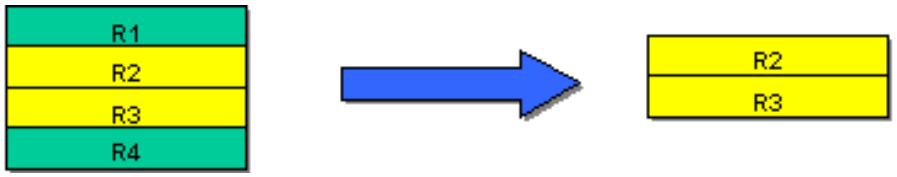


Implementing SELECT (cont.)

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

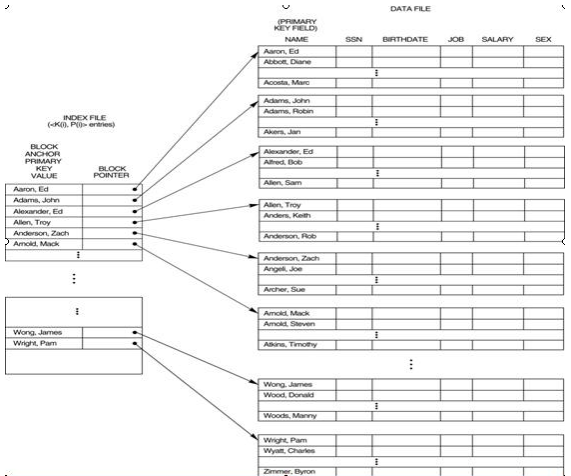
(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

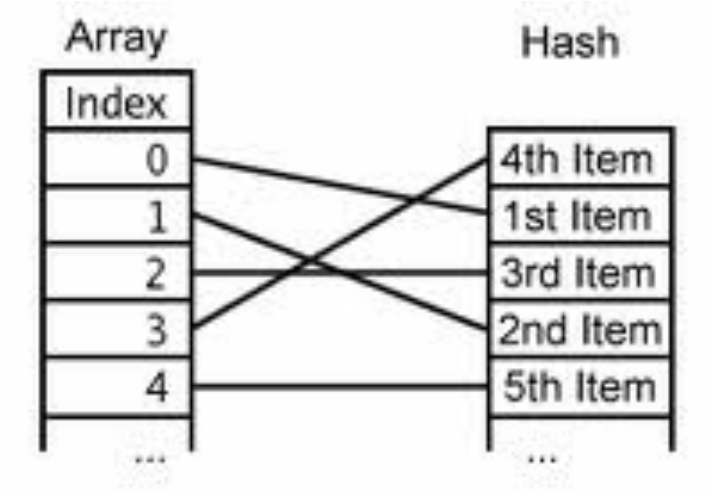
(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



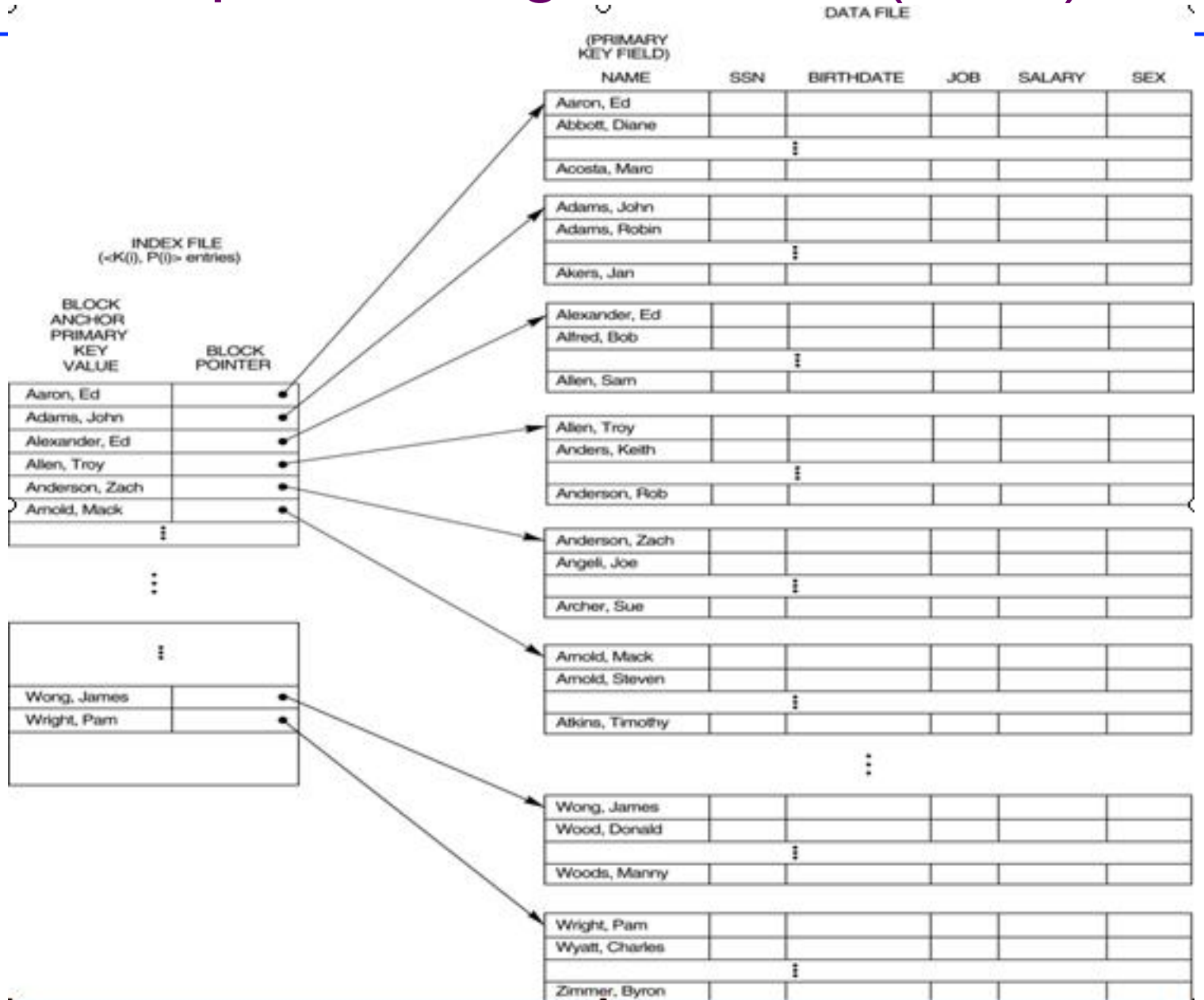
3. Use a primary index or hash key to retrieve a single record

If the selection condition involves an equality comparison on a key attribute with a primary index or hash key, use the primary index or hash key to retrieve the record. Example is OP1





Implementing SELECT (cont.)

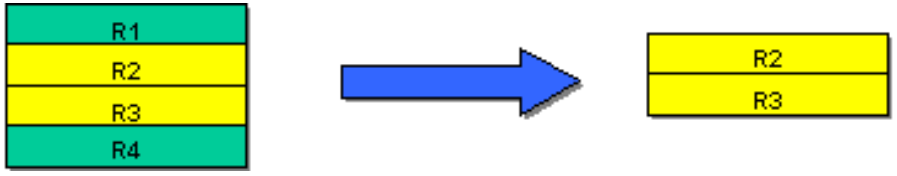


Implementing SELECT (cont.)

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

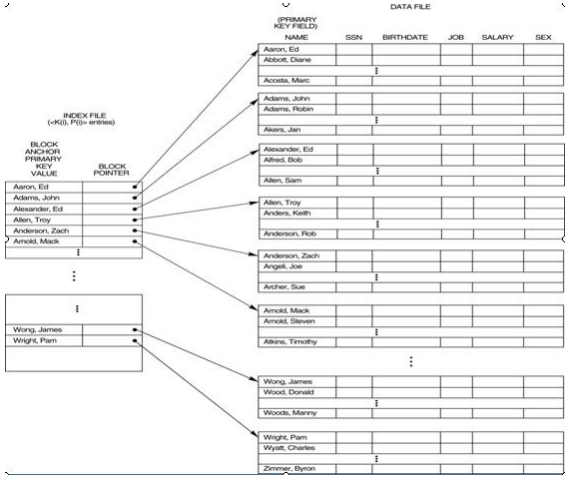
(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

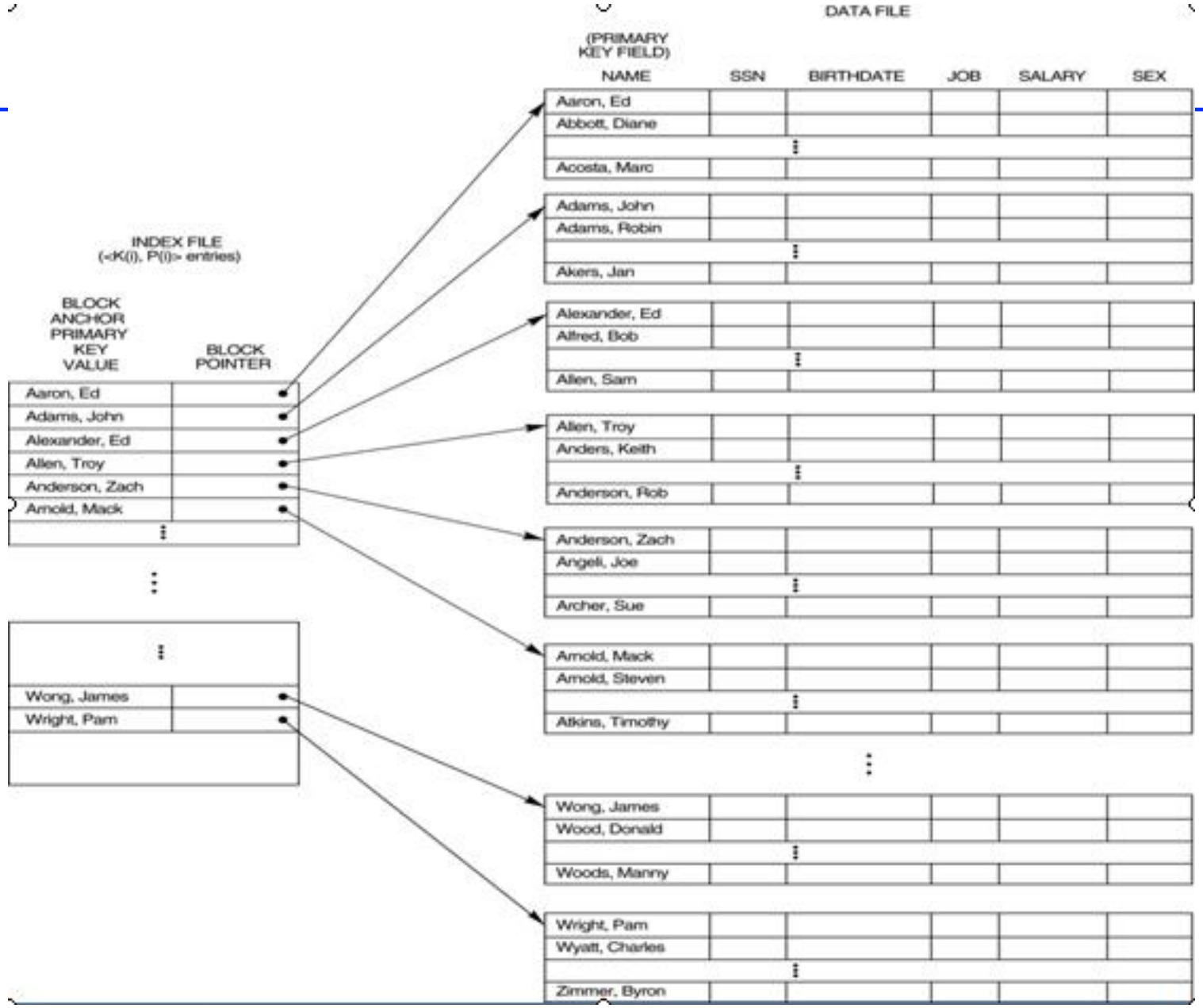
(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



4. Use a primary index to retrieve multiple records

   If the comparison condition is >, >=, <, <= on a key field with a primary key, use the index to find the record satisfying the corresponding equality then retrieve all subsequent records in the file. Example is OP2



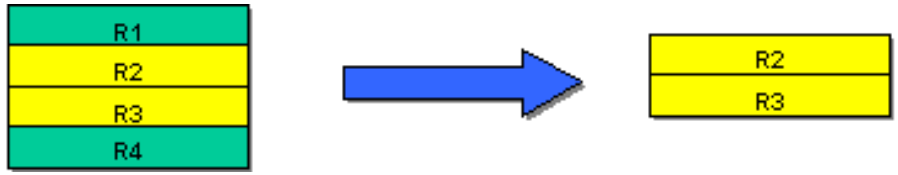

DATA FILE
(PRIMARY KEY FIELD)
NAME
SSN
BIRTHDATE
JOB
SALARY
SEX
Aaron, Ed
Abbott, Diane
Acosta, Marc
Adams, John
Adams, Robin
Akers, Jan
Alexander, Ed
Alfred, Bob
Allen, Sam
Allen, Troy
Anders, Keith
Anderson, Rob
Anderson, Zach
Angeli, Joe
Archer, Sue
Arnold, Mack
Arnold, Steven
Atkins, Timothy
Wong, James
Wood, Donald
Woods, Manny
Wright, Pam
Wyatt, Charles
Zimmer, Byron
INDEX FILE
(<K(i), P(i)> entries)
BLOCK ANCHOR PRIMARY KEY VALUE
BLOCK POINTER
Aaron, Ed
Adams, John
Alexander, Ed
Allen, Troy
Anderson, Zach
Arnold, Mack
Wong, James
Wright, Pam

Implementing SELECT (cont.)

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

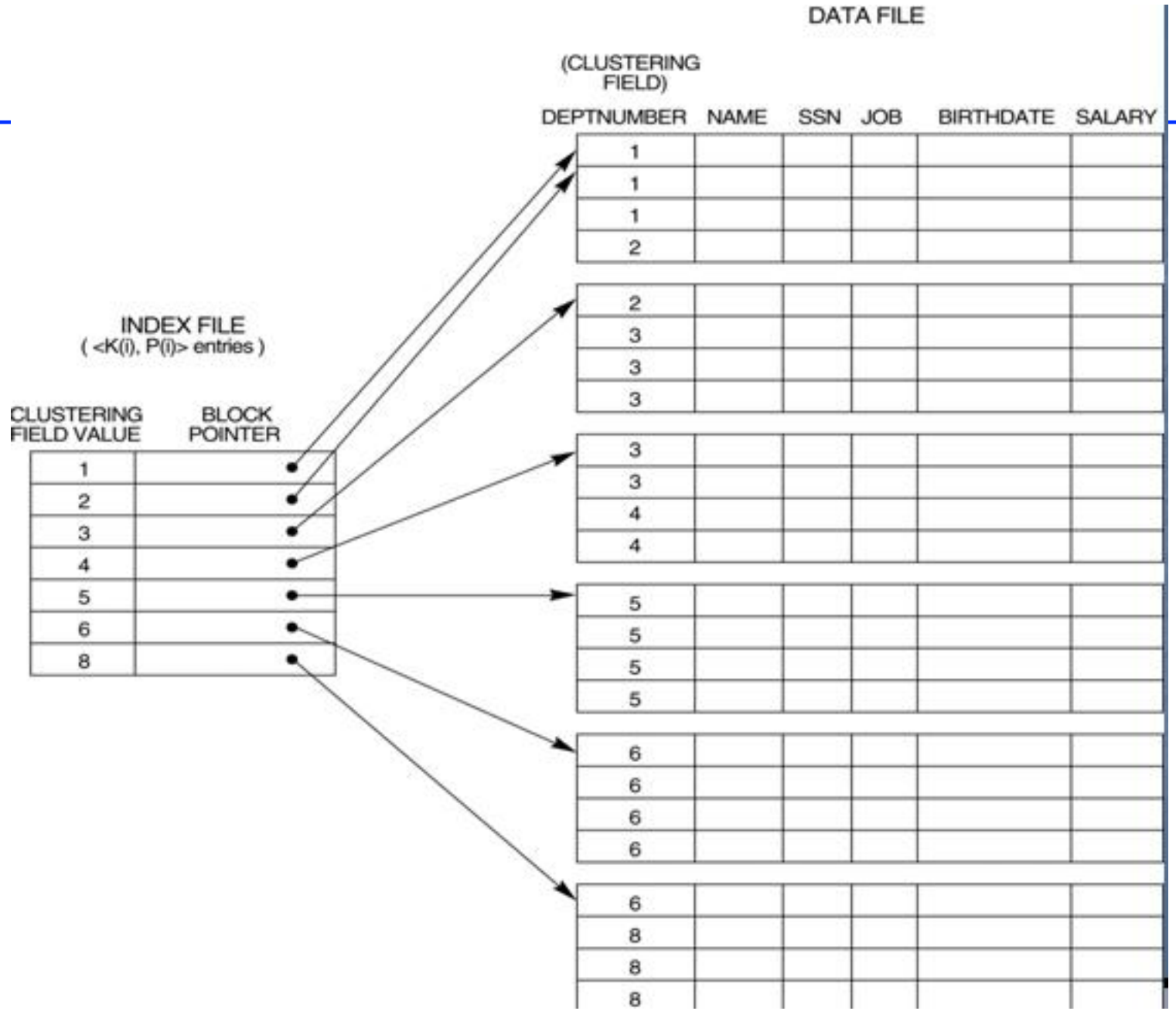
(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



5. Use a clustering index to retrieve multiple records

   If the selection condition involves an equality comparison on a non key attribute with a clustering index, use the clustering index to retrieve all records satisfying the condition. OP3

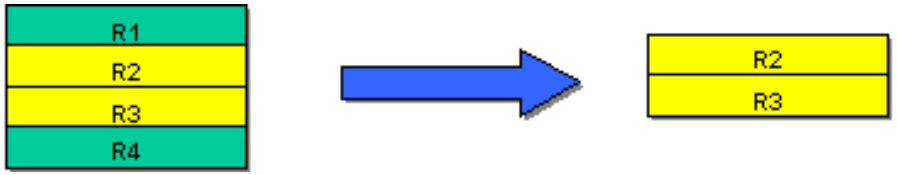

DATA FILE
(CLUSTERING FIELD)
DEPTNUMBER
NAME
SSN
JOB
BIRTHDATE
SALARY
INDEX FILE
(<K(i), P(i)> entries)
CLUSTERING FIELD VALUE
BLOCK POINTER
1
2
3
4
5
6
8

Implementing SELECT (cont.)

(OP1) $\sigma_{SSN=123456789}(EMPLOYEE)$

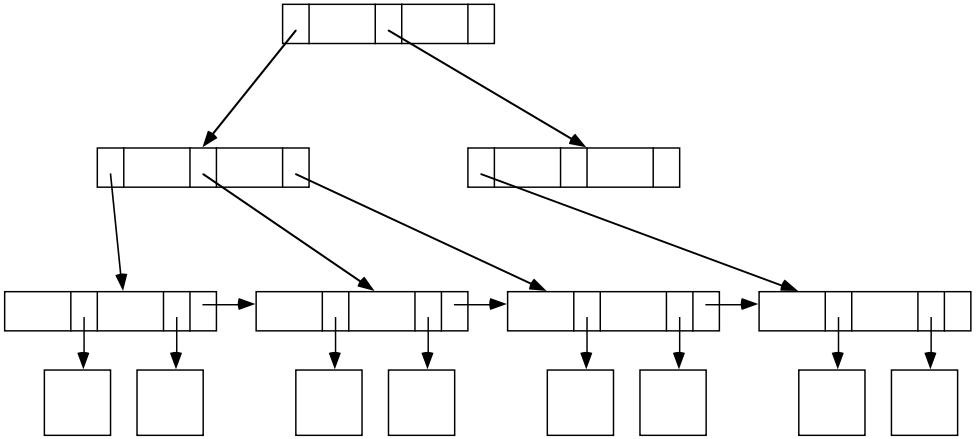
(OP2) $\sigma_{DNUMBER>5}(DEPARTMENT)$

(OP3) $\sigma_{DNO=5}(EMPLOYEE)$



6. B^+-tree

B^+- index - On equality, index can be used to retrieve a single record if the indexing field is unique or multiple records if not a key. Also works on >, >=, <, <=



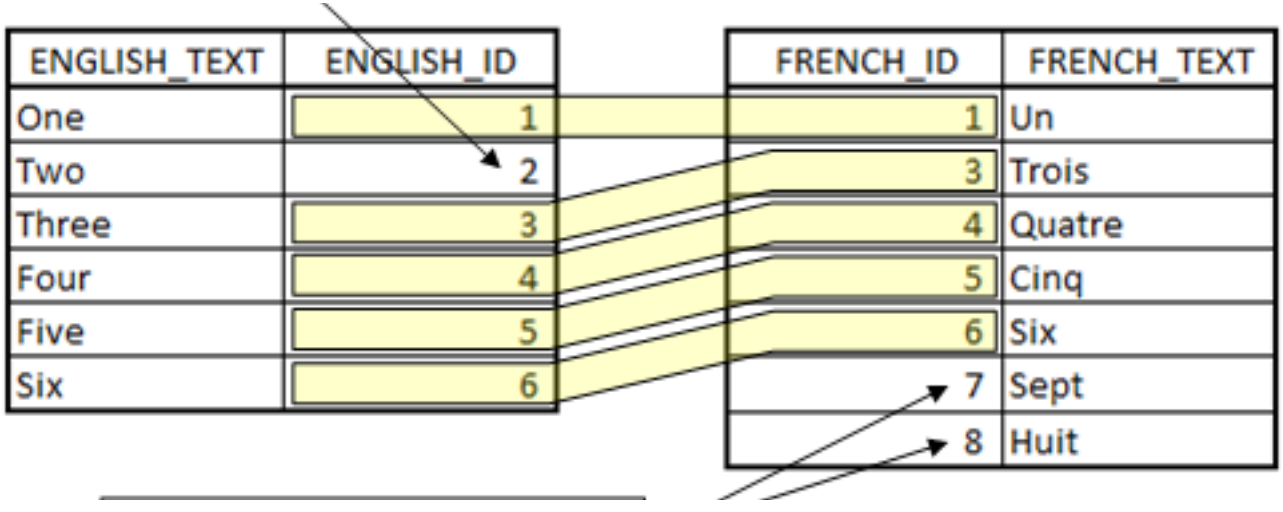
4 & 6 can be used for selection involving ranges of values, called range queries. (Ex, 30000 <= salary <= 35000)

Implementing JOIN

OP6: EMPLOYEE $\bowtie_{\text{DNO=DNUMBER}}$ DEPARTMENT

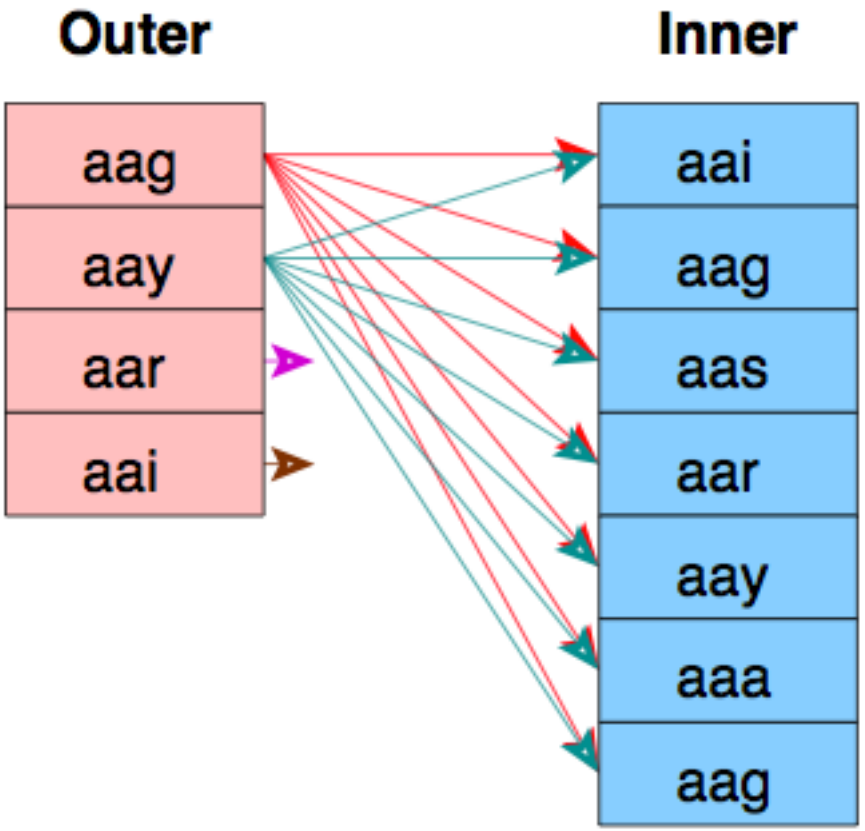
OP7: DEPARTMENT $\bowtie_{\text{MGRSSN=SSN}}$ EMPLOYEE

- Nested-loop join (brute force)
- Single-loop join (using an access structure to retrieve matching records)
- Sort-merge join



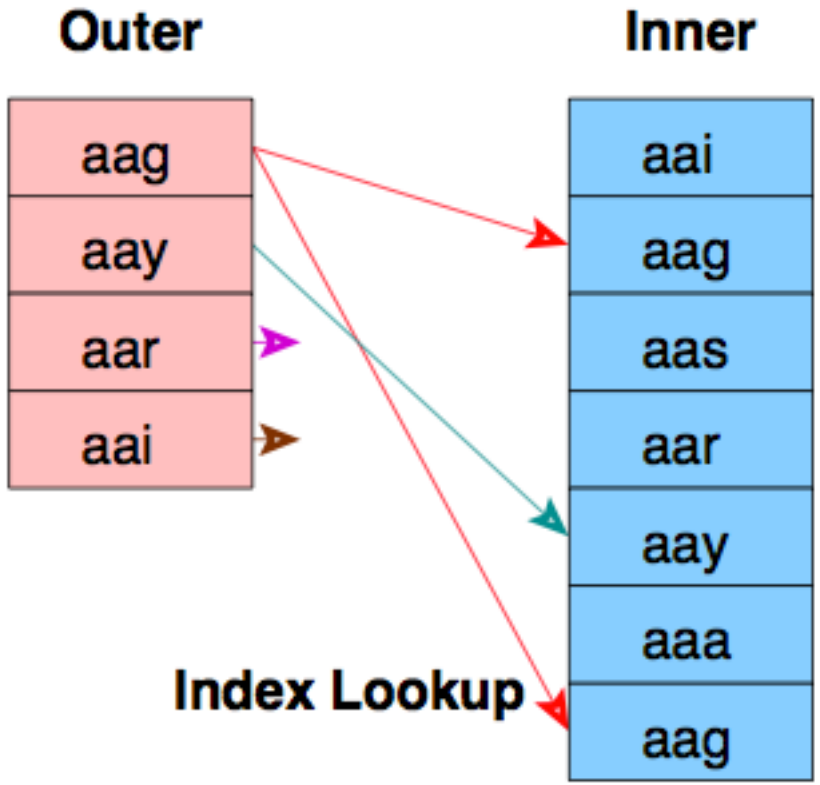
Implementing JOIN

- Nested-loop join (brute force)



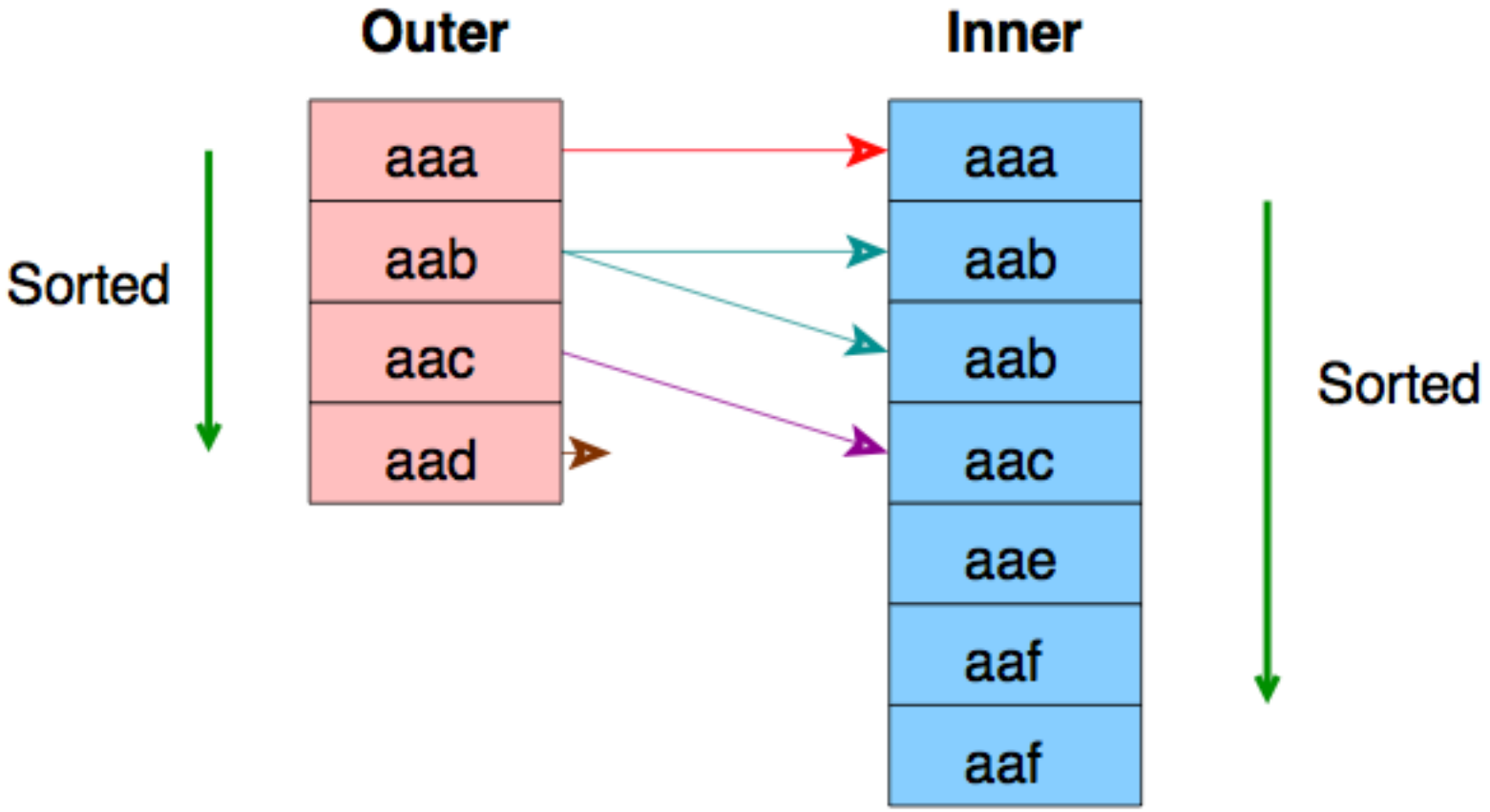
Implementing JOIN

- Single-loop join (using an access structure[index/ hash] to retrieve matching records)



Implementing JOIN

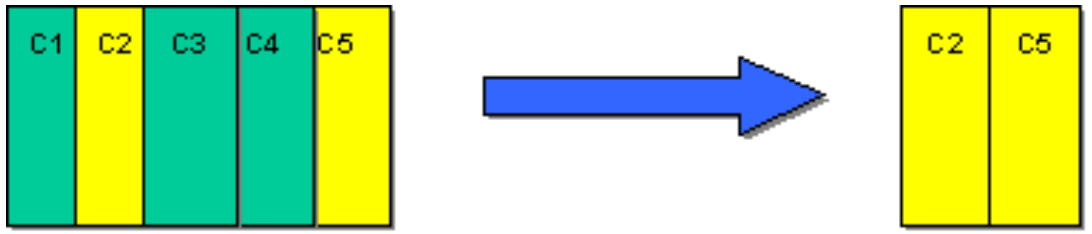
- Sort-merge join



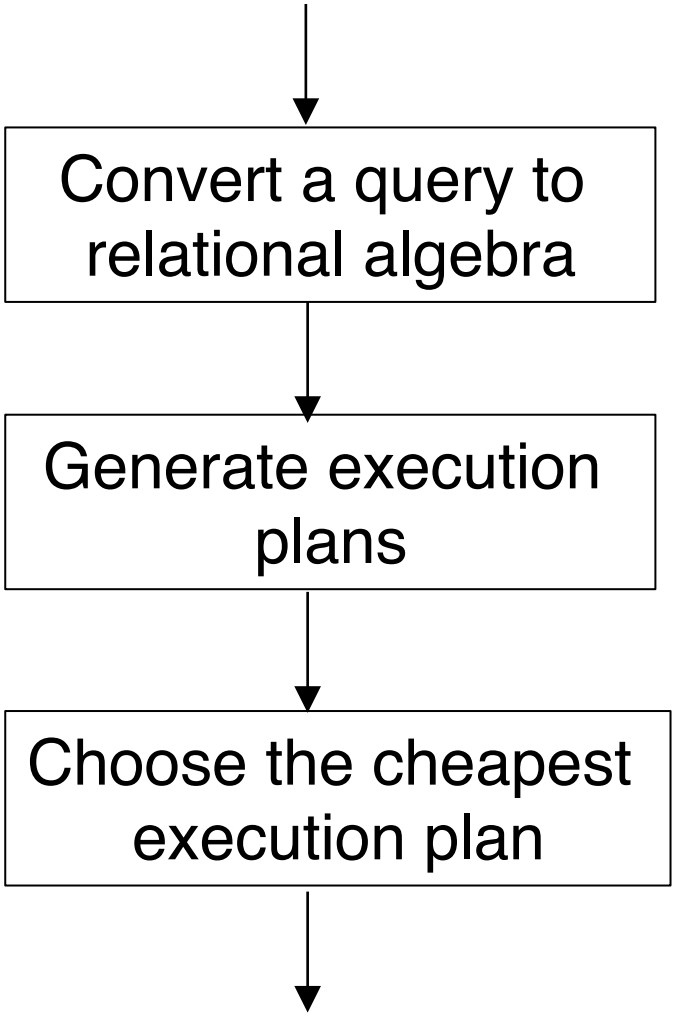
Implementing PROJECT

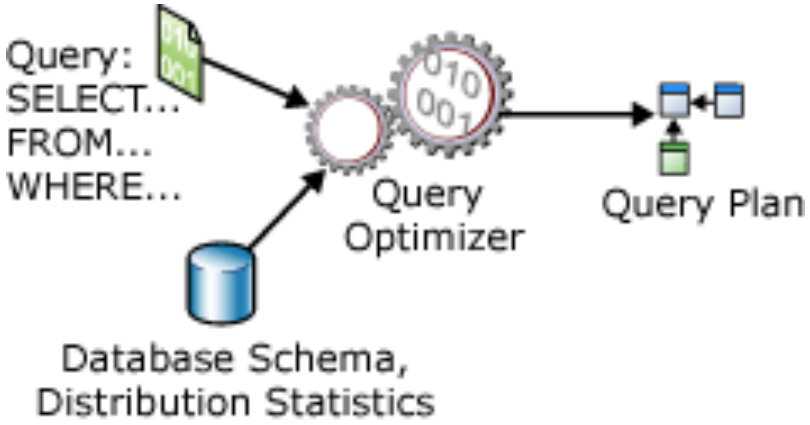
π `<attribute list>` `(R)`

- Straight forward if attribute list includes a key
- Eliminate duplicates (if “DISTINCT” used)



Optimisation Process





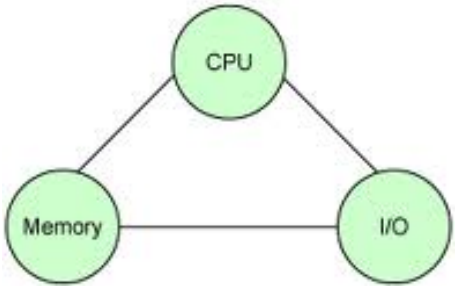
Two main techniques

Heuristic rules

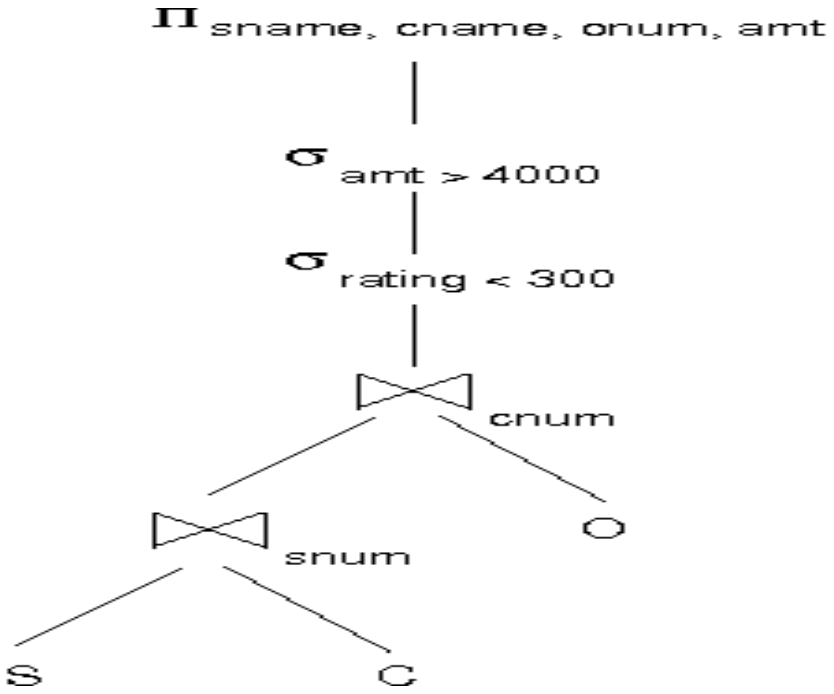
Systematic cost estimation

Generate Query Plans and Choose the Cheapest

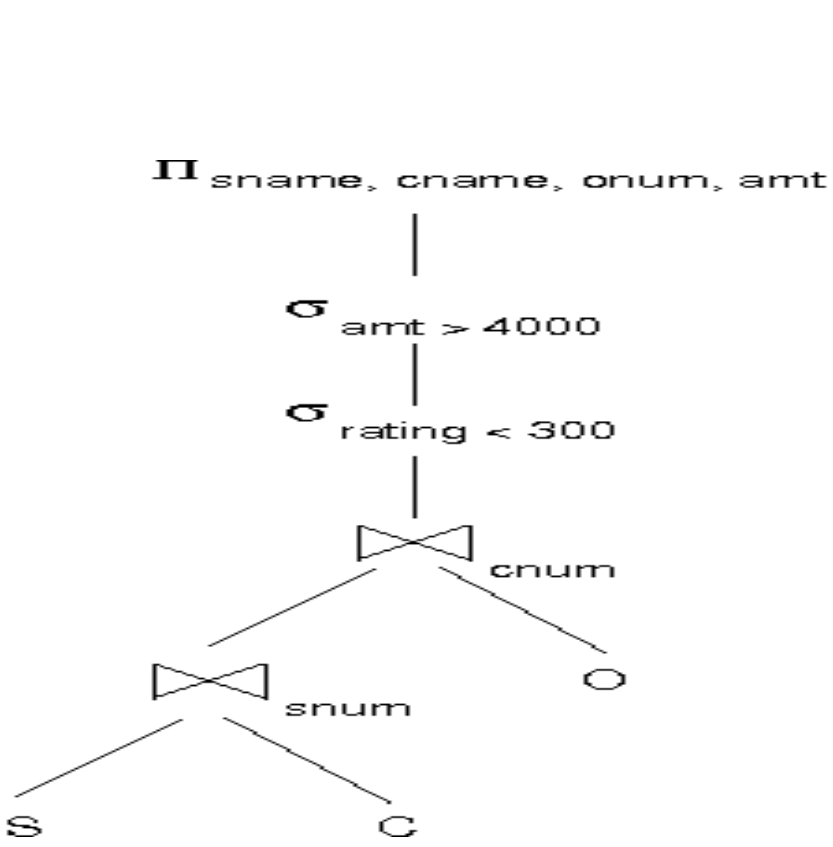
- Construct a set of query execution plans
- Built by combining a set of candidate procedures
  - One for each low-level operation
- Many plans possible
- Cost of creating all plans is prohibitive
  - Heuristics can be used to keep the number of plans within reason
- Assign a cost to each plan
  - need to know the size of data set and estimate size of intermediate tables
- Choose the cheapest

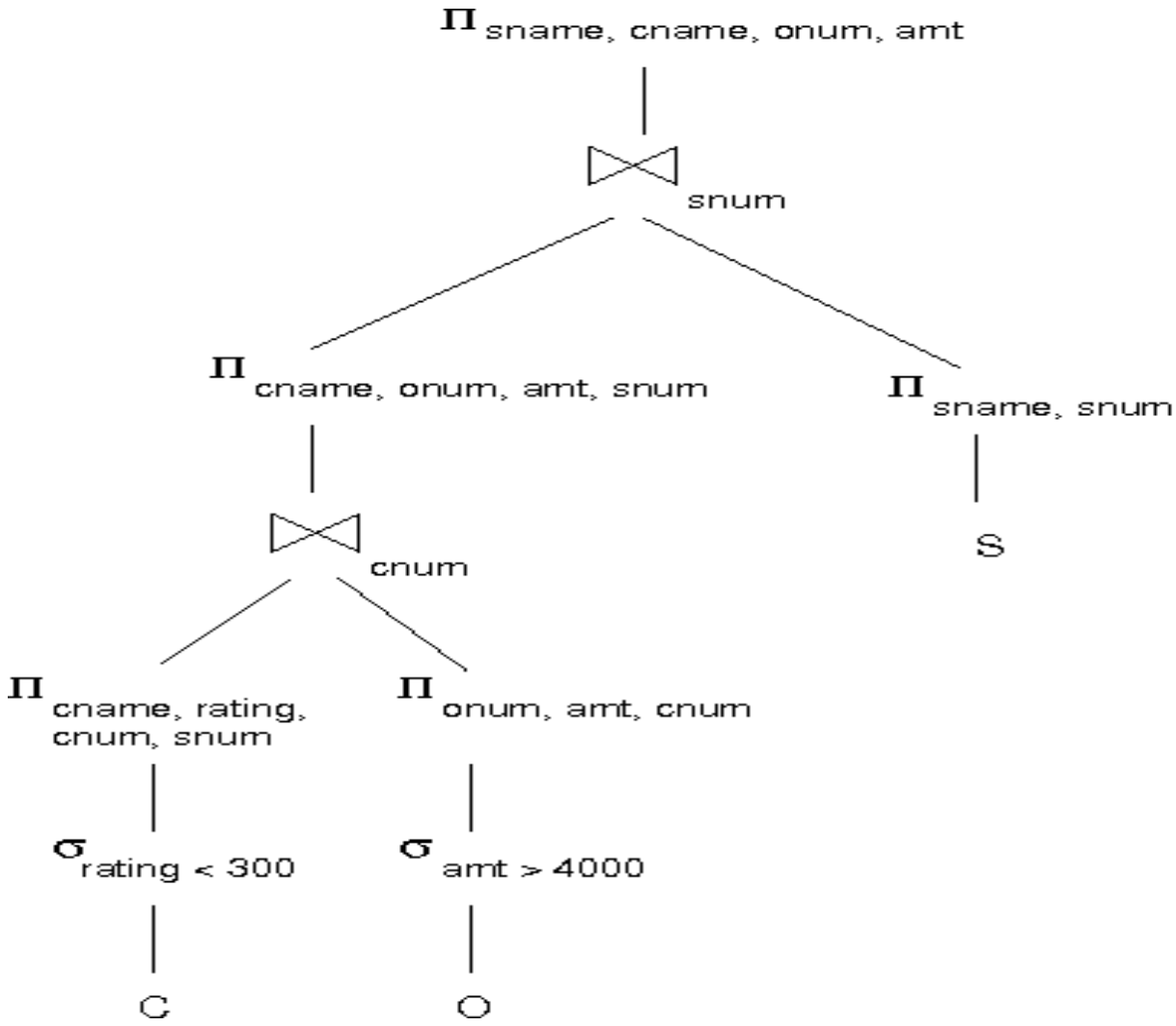


An "Optimised?" Form



An "Optimised?" Form

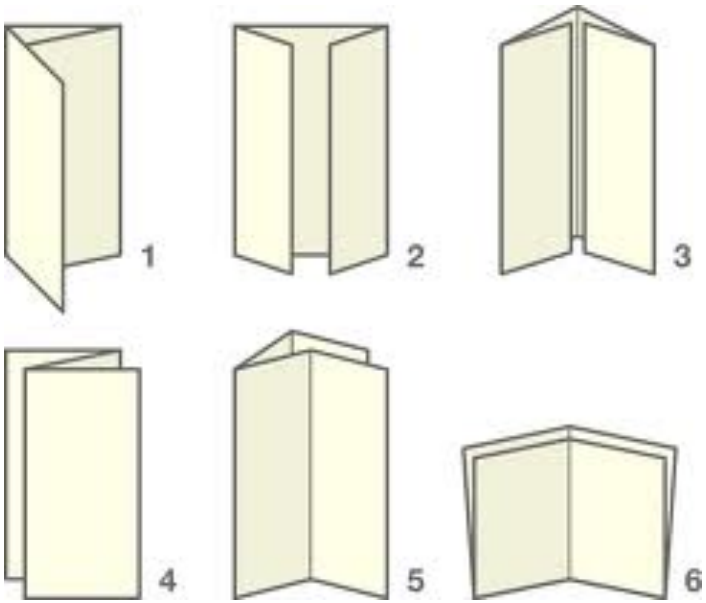




Heuristics in Query Optimisation

- SELECT as early as possible - reduces tuples
- PROJECT as early as possible - reduces attributes
- JOIN later
- Consider sizes of intermediate result files

  some operators (join, times, union) can produce large intermediate tables



Some Transformational Rules

- Cascade of SELECT

  $\sigma_{c1\ AND\ c2\ AND\ c3\ AND....cn}(R) = \sigma_{c1}(\sigma_{c2}(...(\sigma_{cn}(R)...)$
- Commutativity of SELECT

  $\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$
- Cascade of PROJECT

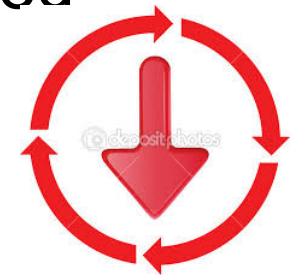
  $\pi_{list1}(\pi_{list2}(\pi_{list3}(R))) = \pi_{list1}(R)$

  and so on
- Commuting SELECT with PROJECT
  - Selection attributes must be in the projection list
- Commutativity of JOIN
- Associativity of JOIN

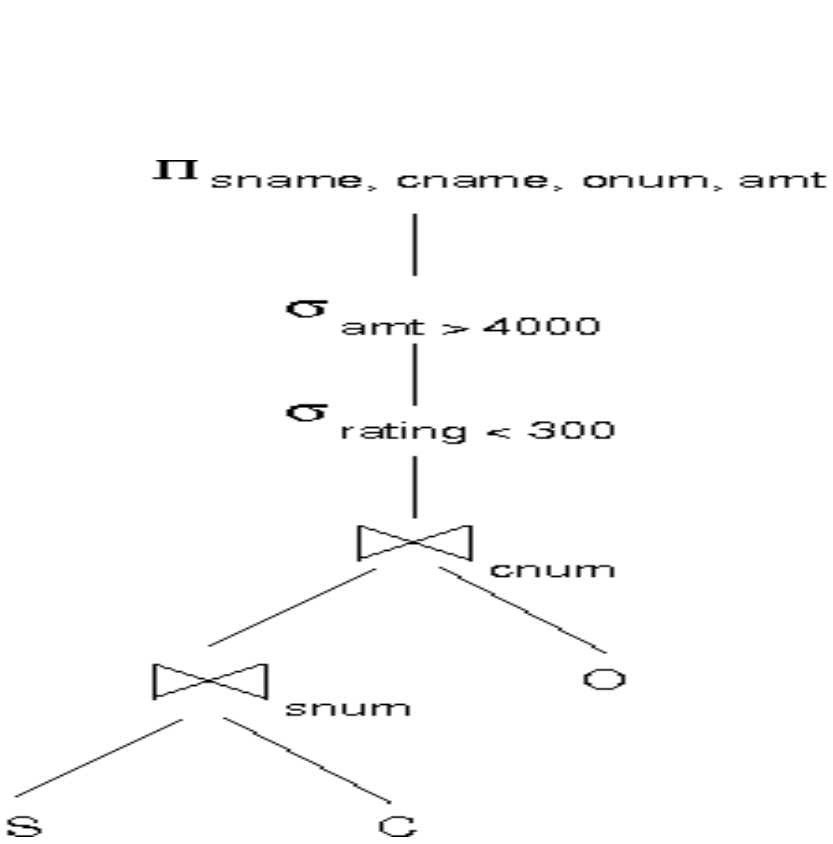
Idea: variations from the mathematical view.

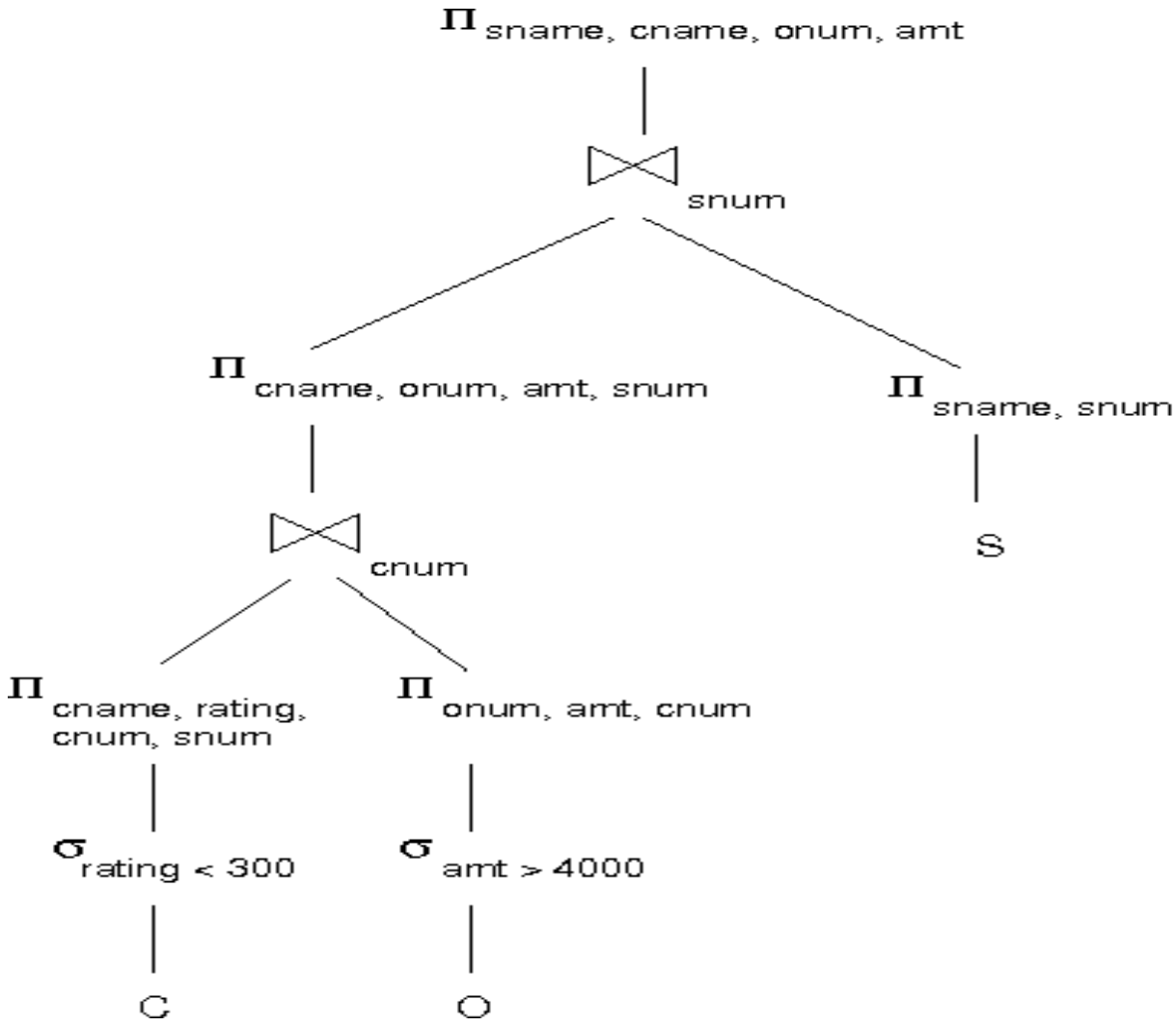
Heuristic Algebraic Optimisation

- Break up any SELECT operations with conjunctive conditions into a cascade of SELECT operations..
  - Allows moving select operations down different branches of the query tree
- Move each SELECT operation as far down the query tree as permitted by the attributes
- Using associativity, rearrange the leaf nodes of the tree so that the leaf node relations with the most restrictive SELECT operations are executed first
- Break down and move lists of projection attributes down the tree as far as possible by creating new PROJECT operations as needed



An "Optimised?" Form





Estimating Cost of Plans

- Access to secondary storage
- Storage costs
- Computation costs
  - in-memory operations such as searching for records, sorting records and performing computations of field values
- Communication costs