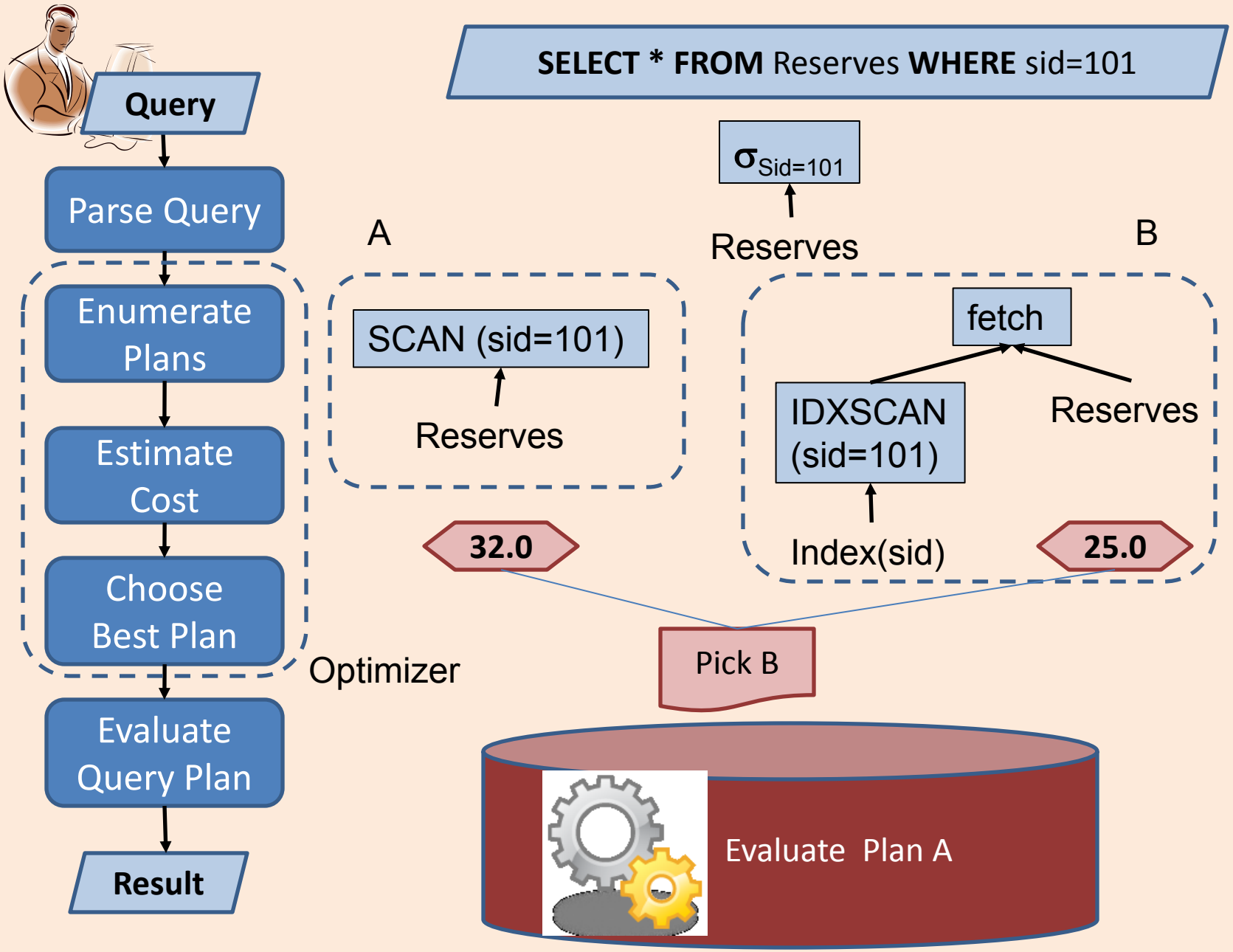
ICS 421 Spring 2010

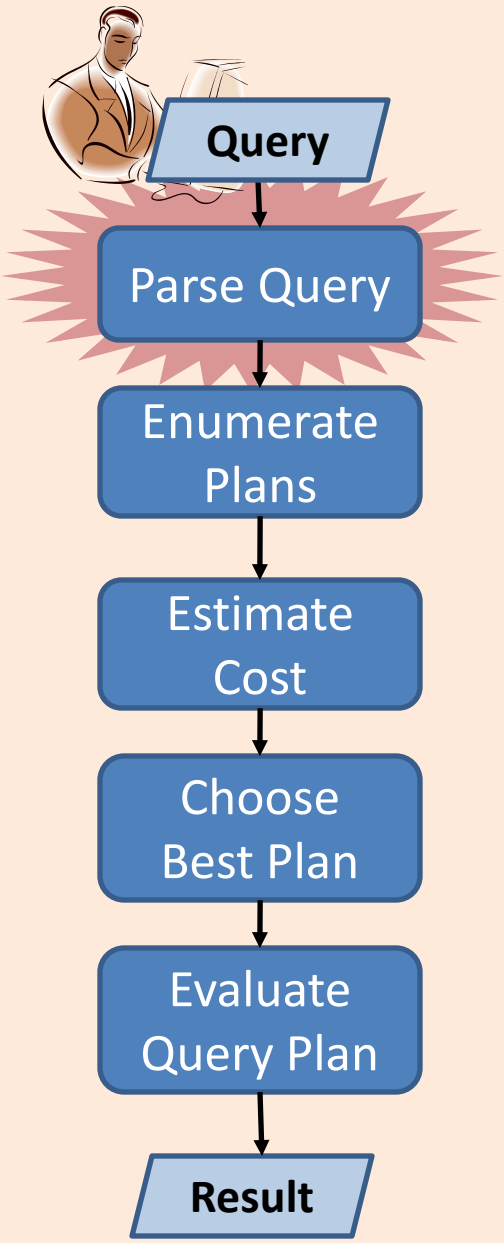
Query Evaluation (i)

Asst. Prof. Lipyeow Lim

Information & Computer Science Department

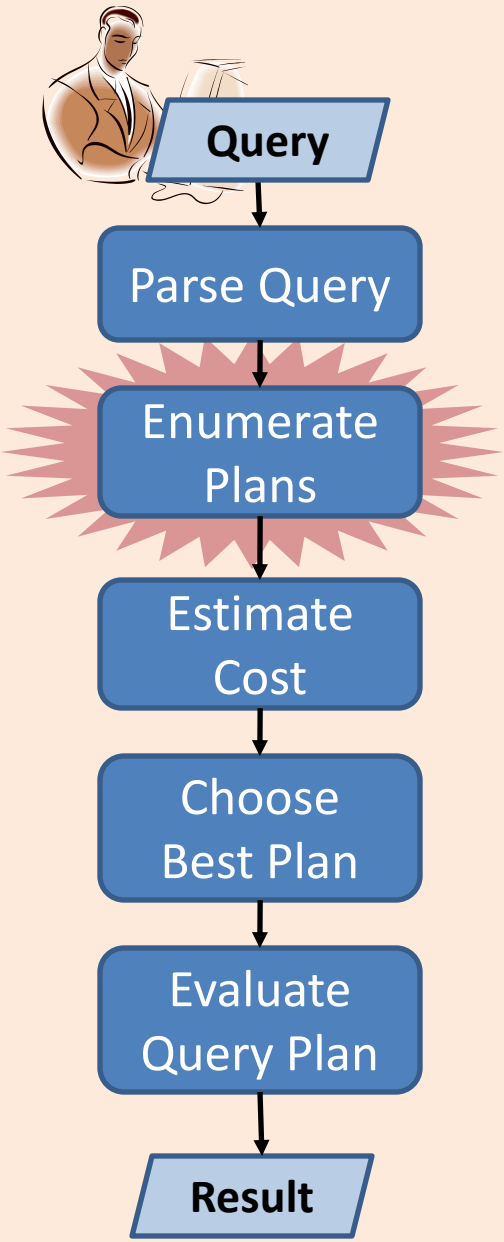
University of Hawaii at Manoa


SELECT * FROM Reserves WHERE sid=101
Query
Parse Query
Enumerate Plans
Estimate Cost
Choose Best Plan
Optimizer
Evaluate Query Plan
Result
σSid=101
Reserves
A
SCAN (sid=101)
Reserves
32.0
B
fetch
IDXSCAN (sid=101)
Reserves
Index(sid)
25.0
Pick B
Evaluate Plan A



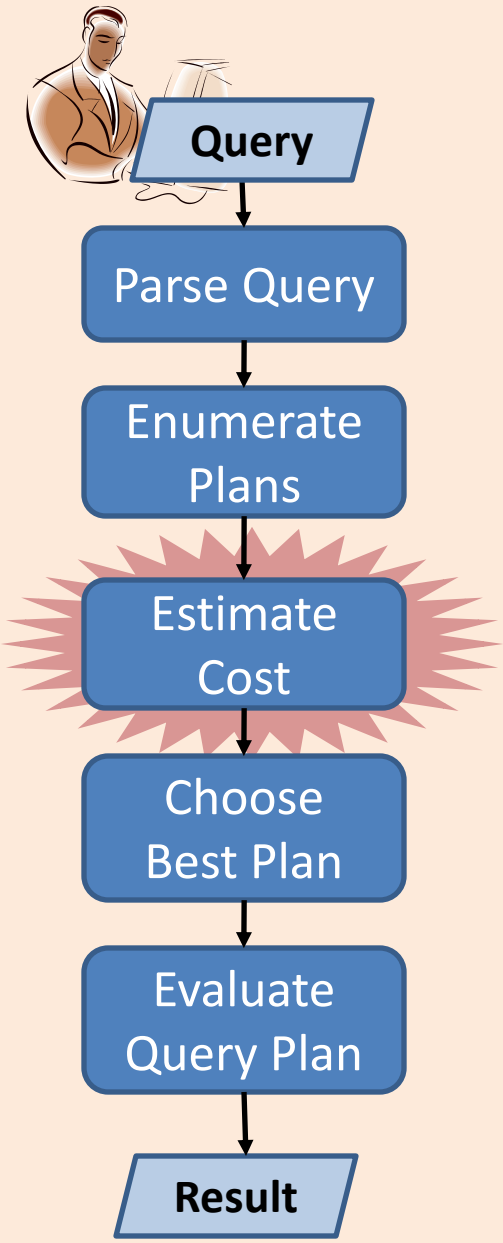
Parse Query

- Input : SQL
  - Eg. SELECT-FROM-WHERE, CREATE TABLE, DROP TABLE statements
- Output: Some data structure to represent the "query"
  - Relational algebra ?
- Also checks syntax, resolves aliases, binds names in SQL to objects in the catalog
- How ?



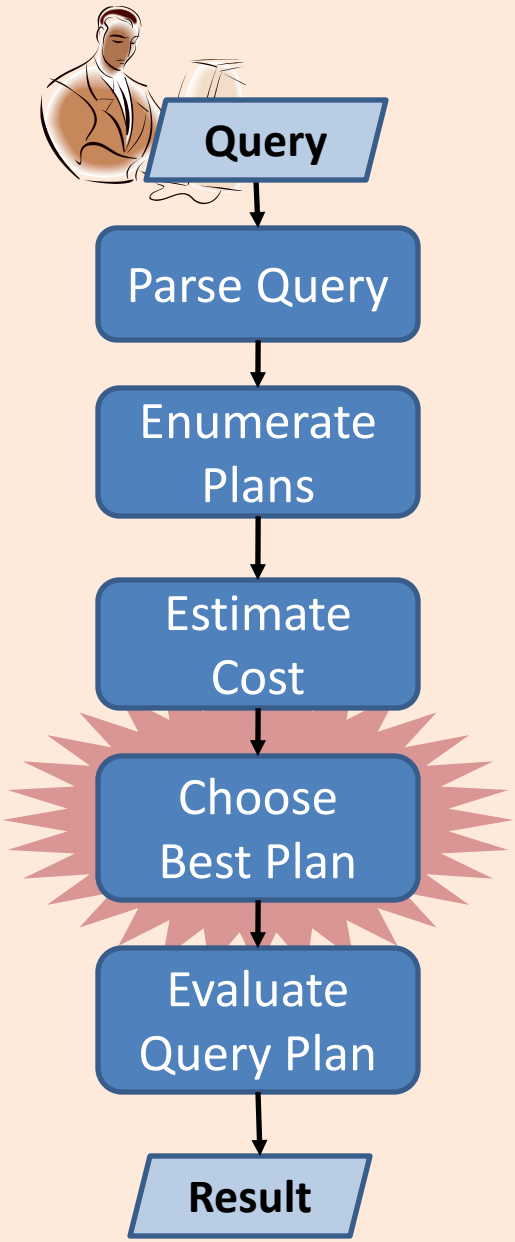
Enumerate Plans

- **Input** : a data structure representing the "query"
- **Output**: a collection of equivalent query evaluation plans
- **Query Execution Plan** (QEP): tree of database operators.
  - high-level: RA operators are used
  - low-level: RA operators with particular implementation algorithm.
- **Plan enumeration**: find **equivalent** plans
  - Different QEPs that return the same results
  - Query rewriting : transformation of one QEP to another equivalent QEP.



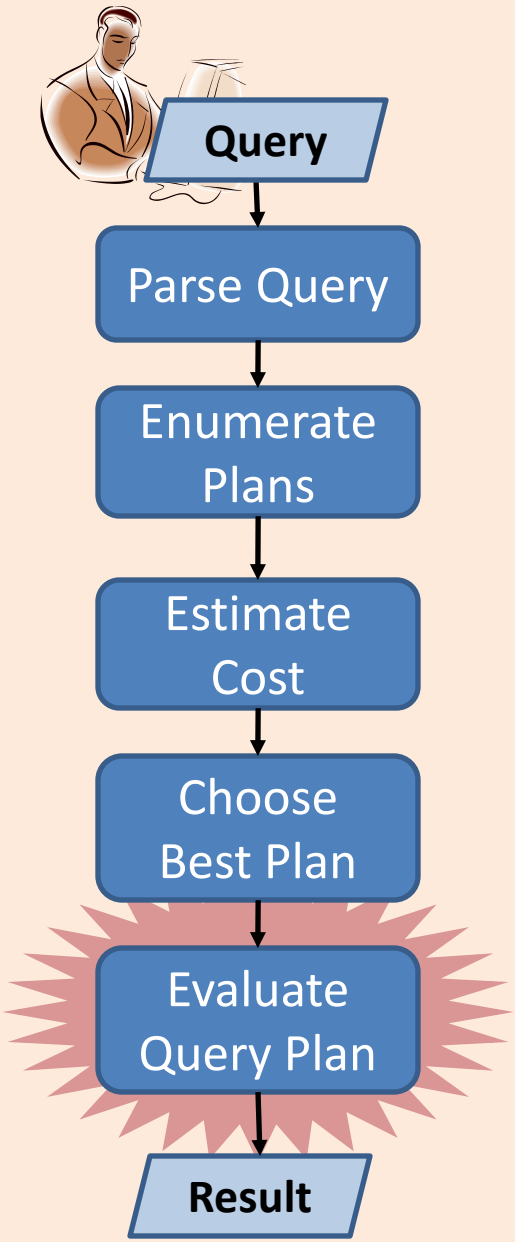
Estimate Cost

- **Input** : a collection of equivalent query evaluation plans
- **Output**: a cost estimate for each QEP in the collection
- **Cost estimation:** a mapping of a QEP to a cost
  - **Cost Model:** a model of what counts in the cost estimate. Eg. Disk accesses, CPU cost …
- Statistics about the data and the hardware are used.



Choose Best Plan

- **Input** : a collection of equivalent query evaluation plans and their cost estimate
- **Output**: best QEP in the collection
- The steps: enumerate plans, estimate cost, choose best plan collectively called the:
- **Query Optimizer:**
  - Explores the space of equivalent plan for a query
  - Chooses the best plan according to a cost model



Evaluate Query Plan

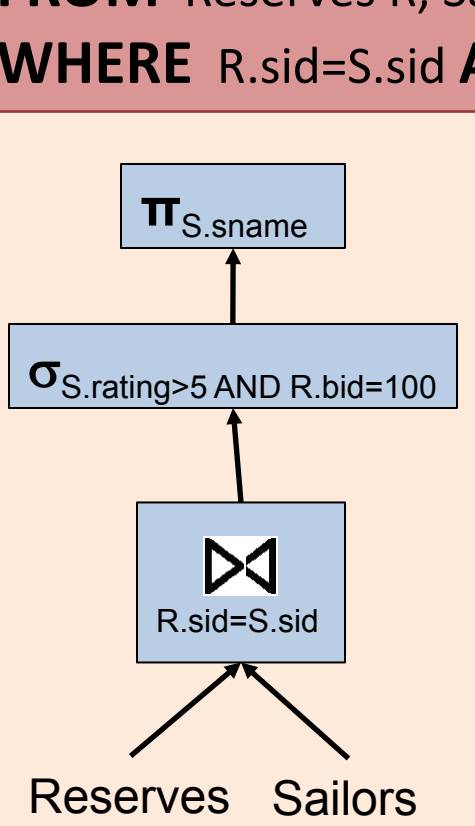
- **Input** : a QEP (hopefully the best)
- **Output**: Query results
- Often includes a "code generation" step to generate a lower level QEP in executable "code".
- **Query evaluation engine** is a "virtual machine" that executes some code representing low level QEP.

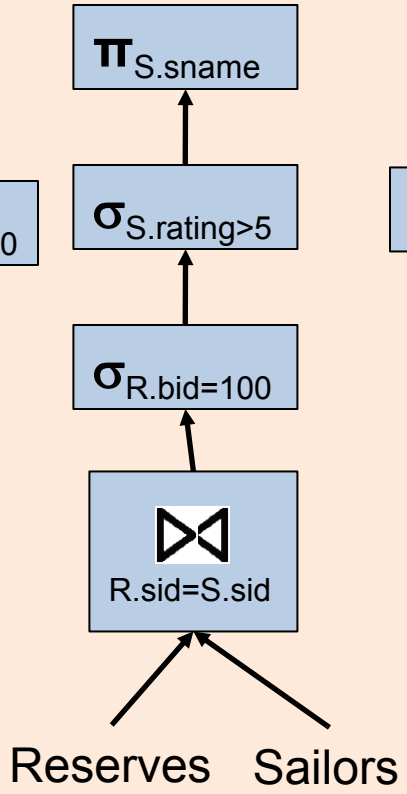
Query Execution Plans (QEPs)

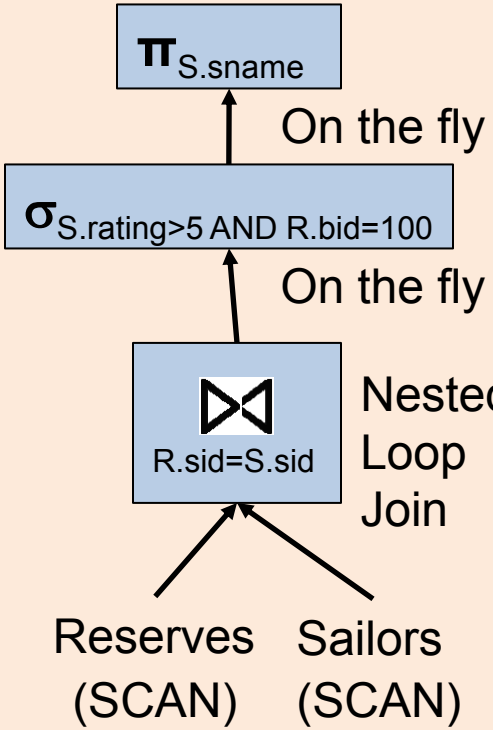
- A **tree** of database operators: each operator is a RA operator with specific implementation
- **Selection σ**: Index Scan or Table Scan
- **Projection π**:
  - Without DISTINCT : Table Scan
  - With DISTINCT : requires sorting or index scan
- **Join ⋈** :
  - Nested loop joins (naïve)
  - Index nested loop joins
  - Sort merge joins
- **Sort :**
  - In-memory sort
  - External sort

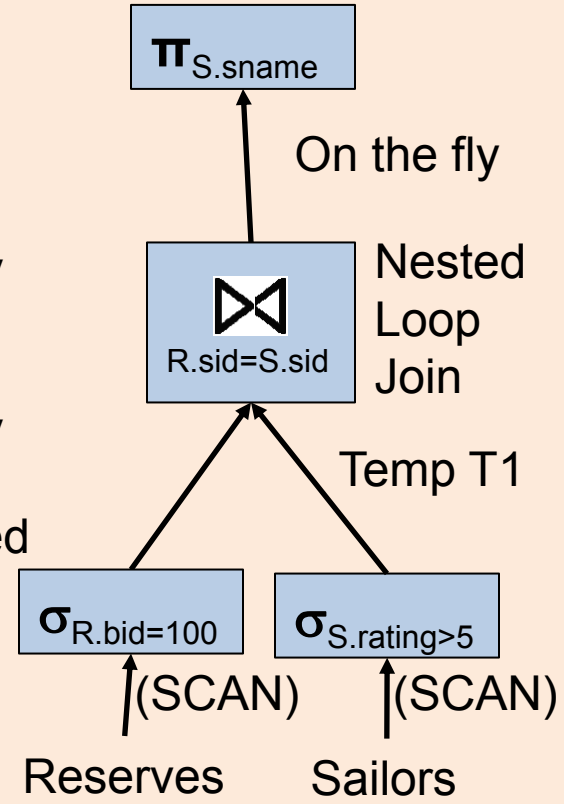
QEP Examples

SELECT S.sname
FROM Reserves R, Sailors S
WHERE R.sid=S.sid **AND** R.bid=100 **AND** S.rating>5



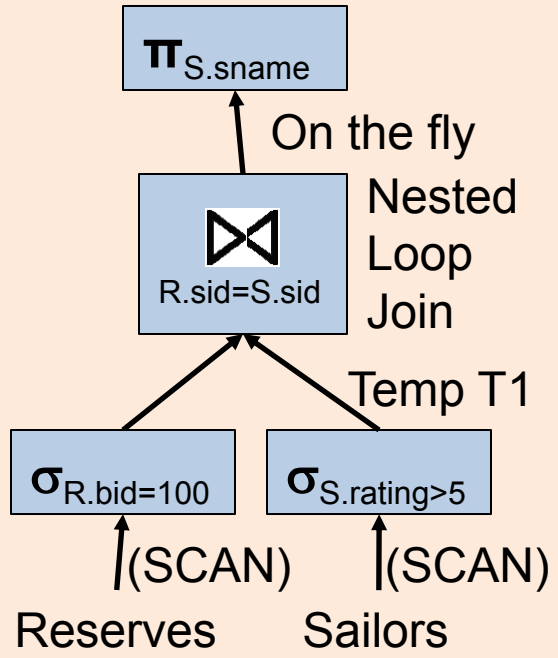


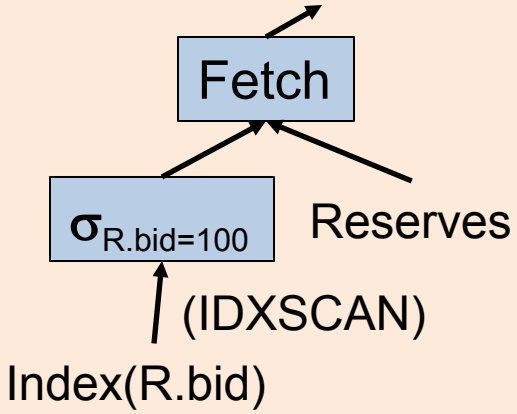




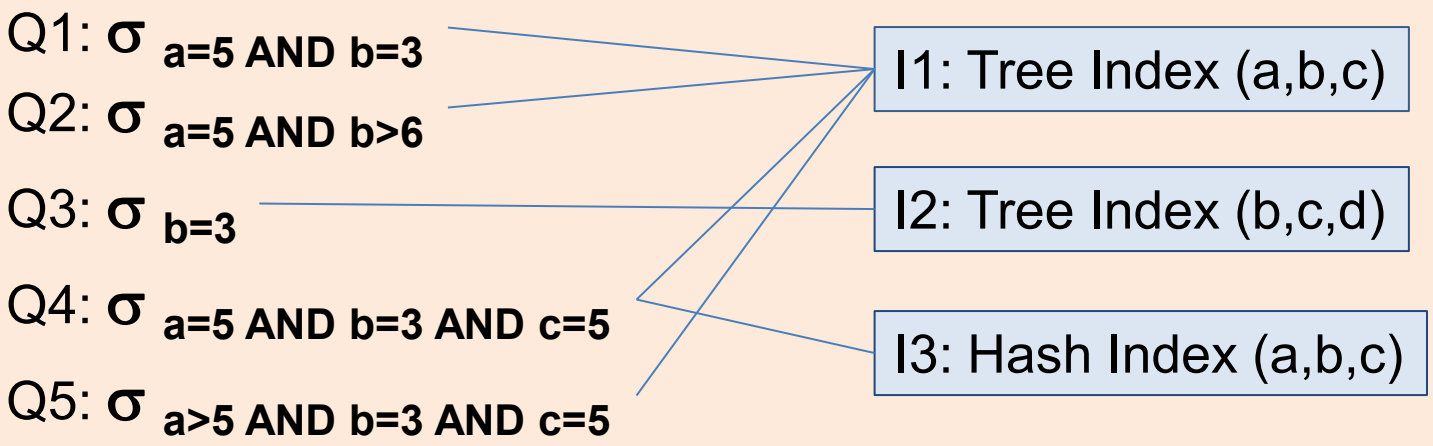
Access Paths

- An **access path** is a method of retrieving tuples. Eg. Given a query with a selection condition:
  - File or table scan
  - Index scan
- **Index matching problem:** given a selection condition, which indexes can be used for the selection, i.e., matches the selection ?
  - Selection condition normalized to conjunctive normal form (CNF), where each term is a *conjunct*
  - Eg. (day<8/9/94 **AND** rname='Paul') **OR** bid=5 **OR** sid=3
  - **CNF**: (day<8/9/94 **OR** bid=5 **OR** sid=3) AND (rname='Paul' **OR** bid=5 **OR** sid=3)





Index Matching



- A **tree index** matches a selection condition if the selection condition is a prefix of the index search key.
- A **hash index** matches a selection condition if the selection condition has a term *attribute=value* for every attribute in the index search key

One Approach to Selections

1. Find the *most selective access path,* retrieve tuples using it
2. Apply remaining terms in selection not matched by the chosen access path

- The **selectivity** of an access path is the size of the result set (in terms of tuples or pages).
  - Sometimes selectivity is also used to mean **reduction factor:** fraction of tuples in a table retrieved by the access path or selection condition.
- Eg. Consider the selection:

  day<8/9/94 **AND** bid=5 **AND** sid=3
  - Tree Index(day)
  - Hash index (bid,sid)