Physical DB Design

courtesy of Joe Hellerstein for some slides

Jianlin Feng
School of Software
SUN YAT-SEN UNIVERSITY

Physical DB Design

- Query optimizer does what it can to use indices, clustering etc.
- DataBase Administrator (DBA) is expected to set up physical design well.
- Good DBAs understand query optimizers very well.

One Key Decision: Indexes

- Which tables
- Which field(s) should be the search key?
- Multiple indexes?
- Clustering?

Index Selection

- One approach:
  - Consider most important queries in turn.
  - Consider best plan using the current indexes
  - See if better plan is possible with an additional index.
  - If so, create it.
- But consider impact on updates!
  - Indexes can make queries go faster, updates slower.
  - Require disk space, too.

Issues to Consider in Index Selection

- Attributes mentioned in a WHERE clause are candidates for index search keys.
  - Range conditions are sensitive to clustering
  - Exact match conditions don’t require clustering
    - Or do they???? :-)
- Choose indexes that benefit many queries
- NOTE: only one index can be clustered per relation!
  - So choose it wisely!

Example 1

```
SELECT  E.ename, D.mgr
FROM  Emp E, Dept D
WHERE  E.dno=D.dno AND D.dname='Toy'
```

- B+ tree index on *D.dname* supports 'Toy' selection.
  - Given this, index on D.dno is not needed.
- B+ tree on *E.dno* allows us to get matching (inner) Emp tuples for each selected (outer) Dept tuple.
- What if WHERE included: `` ... AND E.age=25'' ?
  - Could retrieve Emp tuples using index on *E.age*, then join with Dept tuples satisfying *dname* selection.
    - Comparable to strategy that used *E.dno* index.
  - So, if *E.age* index is already created, this query provides much less motivation for adding an *E.dno* index.

E. dno = 007 , D. dno 找索引 B+ tree

Example 2

聚簇索引 利于范围查询

```
SELECT  E.ename, D.mgr
FROM  Emp E, Dept D
WHERE  E.sal BETWEEN 10000 AND 20000
  AND E.hobby='Stamps' AND E.dno=D.dno
```

范围查询

- All selections are on Emp so it should be the outer relation in any Index NL join. 嵌套循环，条件选择后为外层
  - Suggests that we build a B+ tree index on *D.dno.*
- What index should we build on Emp?
  - B+ tree on *E.sal* could be used, OR an index on *E.hobby* could be used.
  - Only one of these is needed, and which is better depends upon the selectivity of the conditions.
    - As a rule of thumb, equality selections more selective than range selections. 点查询优于范围查询，选择 E.hobby 建 B+ tree
- *Have to understand optimizers to get this right!*

Examples of Clustering

选择范围越小索引收益越大

- B+ tree index on E.age can be used to get qualifying tuples.
  - ❑ How selective is the condition?
  - ❑ Is the index clustered? ✓

```
SELECT  E.dno
FROM  Emp E
WHERE  E.age>40
```

较边范围查询

- Consider the GROUP BY query.
  - ❑ If many tuples have *E.age* > 10, using *E.age* index and sorting the retrieved tuples may be costly. 选择范围过大
  - ❑ Clustered *E.dno* index may be better!

```
SELECT  E.dno,  COUNT (*)
FROM  Emp E
WHERE  E.age>10
GROUP BY E.dno
```

- Equality queries and duplicates:
  - ❑ Clustering on *E.hobby* helps!

```
SELECT  E.dno
FROM  Emp E
WHERE  E.hobby=Stamps
```