Database Systems

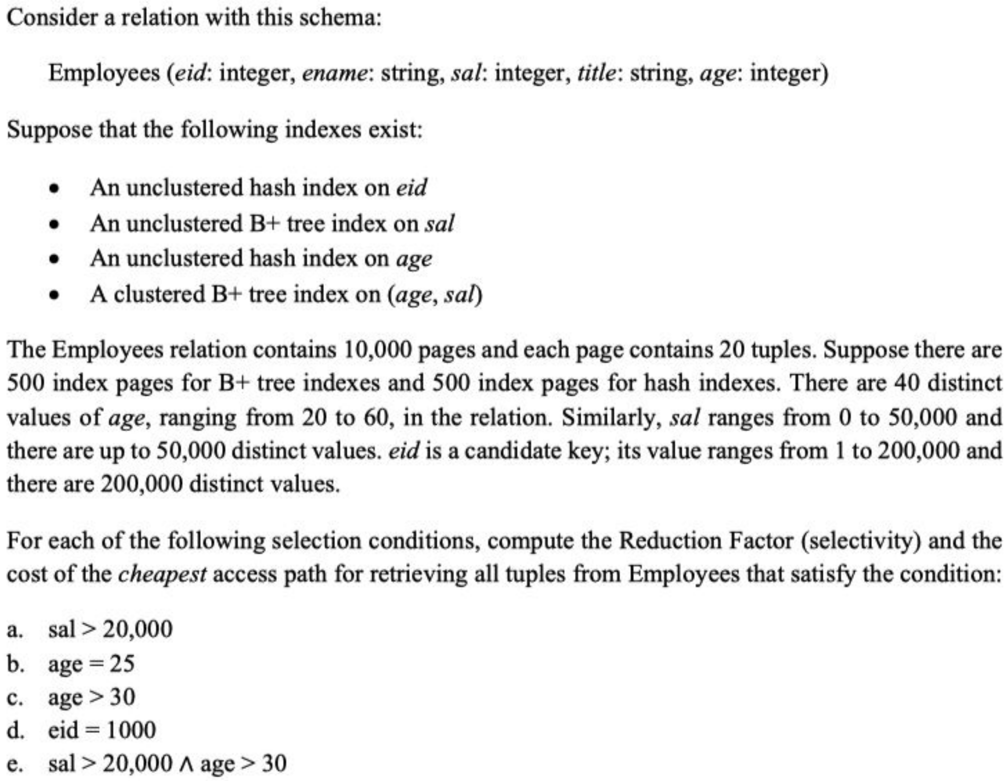
Tutorial Week 8

Objectives

I. Estimate cost of single-relation plans

II. Estimate cost of multi-relation plans

Exercises (Single-Relation Plans)



Consider a relation with this schema:

Employees (*eid*: integer, *ename*: string, *sal*: integer, *title*: string, *age*: integer)

Suppose that the following indexes exist:

- An unclustered hash index on *eid*
- An unclustered B+ tree index on *sal*
- An unclustered hash index on *age*
- A clustered B+ tree index on (*age*, *sal*)

The Employees relation contains 10,000 pages and each page contains 20 tuples. Suppose there are 500 index pages for B+ tree indexes and 500 index pages for hash indexes. There are 40 distinct values of *age*, ranging from 20 to 60, in the relation. Similarly, *sal* ranges from 0 to 50,000 and there are up to 50,000 distinct values. *eid* is a candidate key; its value ranges from 1 to 200,000 and there are 200,000 distinct values.

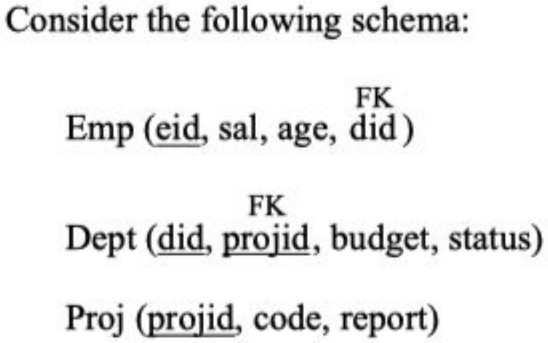
For each of the following selection conditions, compute the Reduction Factor (selectivity) and the cost of the *cheapest* access path for retrieving all tuples from Employees that satisfy the condition:

a. sal > 20,000
b. age = 25
c. age > 30
d. eid = 1000
e. sal > 20,000 ∧ age > 30

Exercises (Single-Relation Plans) (15 mins)

- Step 1: compute reduction factor(s)
- Step 2: consider possible access paths
  - Which index(es) can be used?
  - Also, don’t forget to consider the **cost of a full-table scan!!!**
- Step 3: compute the cost of each possible access path
  - Query processing cost formulae cheat sheet on Canvas!!!

Exercises (Multi-Relation Plans)



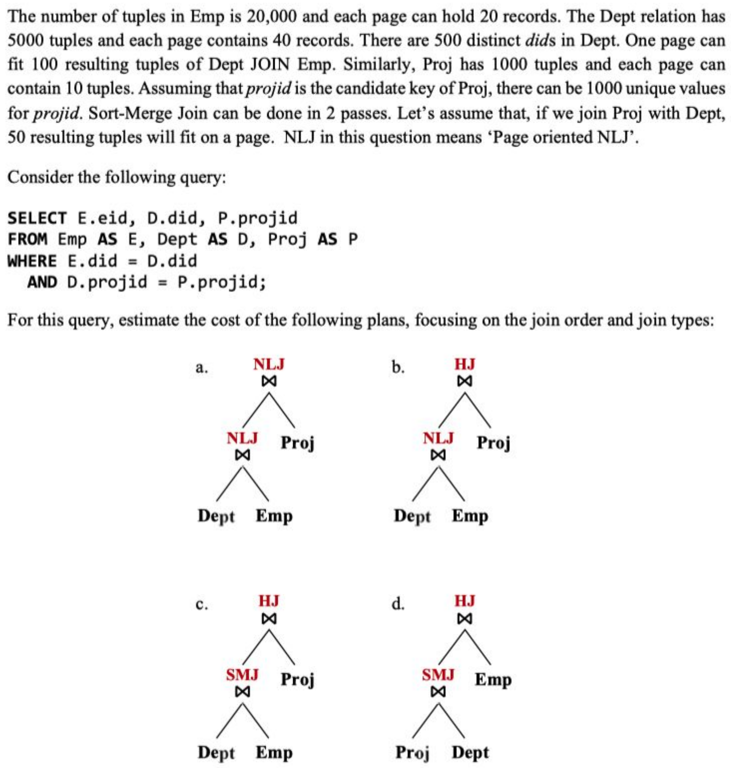
Consider the following schema:

Emp ($\underline{\text{eid}}$, sal, age, $\overset{\text{FK}}{\text{did}}$)

Dept ($\underline{\text{did}}$, $\overset{\text{FK}}{\underline{\text{projid}}}$, budget, status)

Proj ($\underline{\text{projid}}$, code, report)

Exercises (Multi-Relation Plans)



The number of tuples in Emp is 20,000 and each page can hold 20 records. The Dept relation has 5000 tuples and each page contains 40 records. There are 500 distinct *dids* in Dept. One page can fit 100 resulting tuples of Dept JOIN Emp. Similarly, Proj has 1000 tuples and each page can contain 10 tuples. Assuming that *projid* is the candidate key of Proj, there can be 1000 unique values for *projid*. Sort-Merge Join can be done in 2 passes. Let's assume that, if we join Proj with Dept, 50 resulting tuples will fit on a page. NLJ in this question means 'Page oriented NLJ'.

Consider the following query:

```
SELECT E.eid, D.did, P.projid
FROM Emp AS E, Dept AS D, Proj AS P
WHERE E.did = D.did
  AND D.projid = P.projid;
```

For this query, estimate the cost of the following plans, focusing on the join order and join types:

Exercises (Multi-Relation Plans) (20 mins)

- Step 0: find the number of pages of all relations
- Step 1: find the cost of the child join (the join between the first 2 relations)
- Step 2: find the result size of the child join in tuples, and then convert to **number of pages**!!!
- Step 3: find the cost of the parent join
- Step 4: add the cost of the child join and the parent join

Take Home Questions

- More multi-relation plans, but with *different access methods*
- b) and c) are especially tricky
- More questions on Canvas in “Practice for exam (booklet)”

Week 8 Lab

- Canvas → Modules → Week 8 → Lab → L08 Query Optimisation Workbench (PDF)
- Objectives:
    - Use MySQL Execution Plan to understand query execution paths
    - Understand the CREATE INDEX and DROP INDEX syntax
    - Optimize query costs by using indexes to change an execution path
- Breakout rooms, “ask for help” button if you need help or have any questions