Database Development and Design (CPT201)

Tutorial 4: Query Optimisation

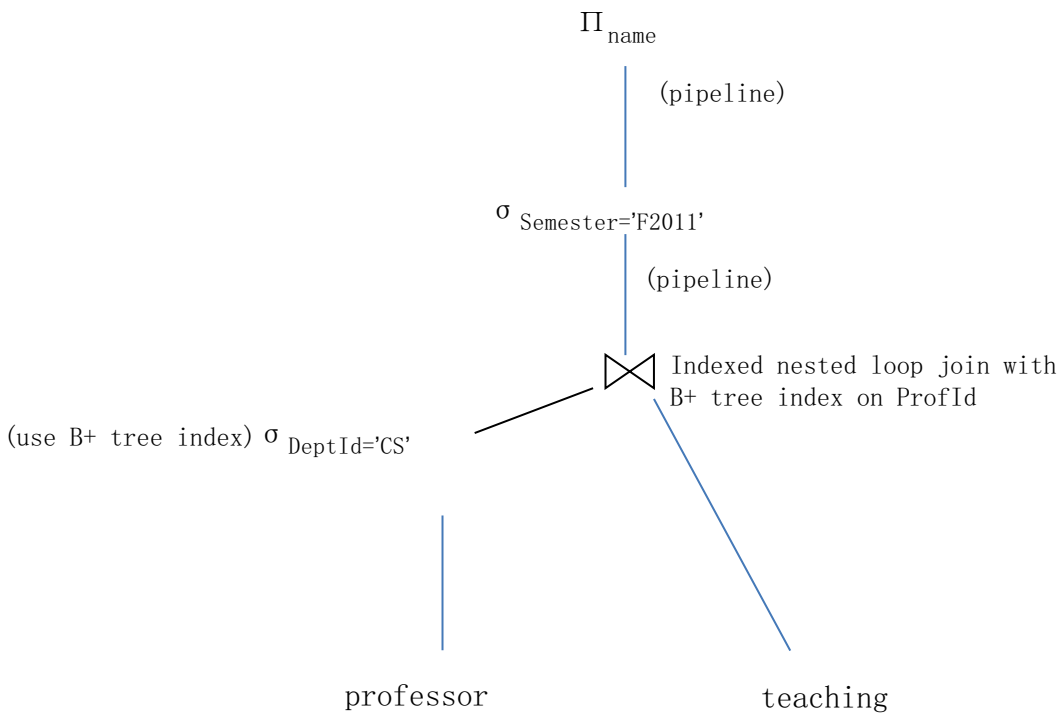
Dr. Wei Wang

Department of Computing

Q1

- Given two relations professor(ProfId, Name, DeptId) and teaching(ProfId, CourseCode, Semester) with the following catalog information:
  - $n_{professor}$ = 1,000
  - $b_{professor}$ =200
  - index: a clustered 2-level B^+ -tree on DeptId for relation professor.
  - V(DeptId, professor) = 50
  - $n_{teaching}$ = 10,000
  - $b_{teaching}$ =1,000
  - index: a B+ tree index on ProfId for relation teaching.
- Assume that the average cost (the number of block transfers) of using the B+ tree index to fetch all matching tuples from relation teaching for one tuple in professor is X. Furthermore, no block transfers are needed for an operation when pipelining is used. Answer the following questions.
  - (a). What is the estimated size of evaluating $\delta_{DeptID='CS'}$ Professor?
  - (b). What is the estimated number of block transfers for evaluating $\delta_{DeptID='CS'}$ using the B+ tree?
  - (c). Based on the answers you have for $\delta_{DeptID='CS'}$ Professor, what is the estimated number of block transfers for evaluating the *join* operation?
  - (d). What is the total number of block transfers for the whole evaluation plan?

Q1 cont'd


Π name
(pipeline)
σ Semester='F2011'
(pipeline)
Indexed nested loop join with
B+ tree index on ProfId
(use B+ tree index) σ DeptId='CS'
professor
teaching

Q1 Solution

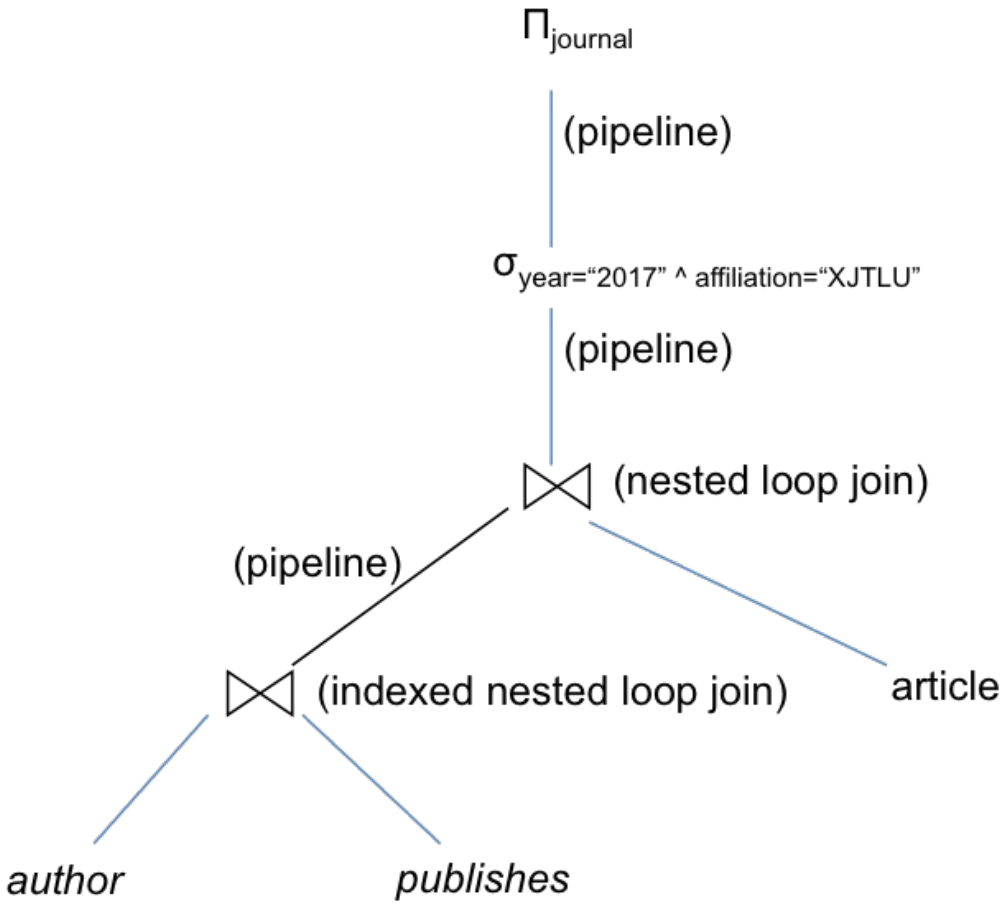
Note: in all questions, if pipelining cannot be used in the evaluation plan then materialisation has to be used.

- (a). 1,000/50=20 tuples (or 200/50=4 blocks)
- (b). 20 records fit into 4 blocks, so the number of block transfer is 2+4=6
- (c). Using indexed nested look join based on the B+ tree index, professor relation (after selection has 20 tuples) is the outer relation. So the number is b_r+n*c = 4+20*X.
- (d). If pipelining is used then we do not need any block transfer, but where pipelining is not used, we have add the cost of storing the intermediate blocks. So the total number of block transfer is 6+4+4+20X.

Q2

- Consider the following three relations, evaluation plan and their catalog information. "*authorId*" and "*articleID*" are the primary keys for relations *author* and *article*, respectively. The two attributes in *publishes* are the foreign keys referencing *author* and *article*, respectively.
  - *author (authorID, name, affiliation, email)*
  - *publishes (articleID, authorID)*
  - *article (articleID, title, journal, year, publisher)*
- number of records in *author*, n_{author} = 5,000;
- number of blocks in *author*, b_{author} = 500;
- number of records in *publishes*, $n_{publishes}$ = 100,000;
- number of blocks in *publishes*, $b_{publishes}$ = 100;
- number of records in *article*, $n_{article}$ = 20,000;
- number of blocks in *article*, $b_{article}$ = 2,000;
- index: a primary B^+-tree index of height 3 on the *authorID* attribute of *publishes* relation;
- number of distinct values for the attribute *affiliation* in the *author* relation, *V(author, affiliation) = 500;*
- number of distinct values for the attribute *year* in the *article* relation, *V(article, year) = 10;*
- Assume that all selections are performed using linear scan.

Q2 cont'd

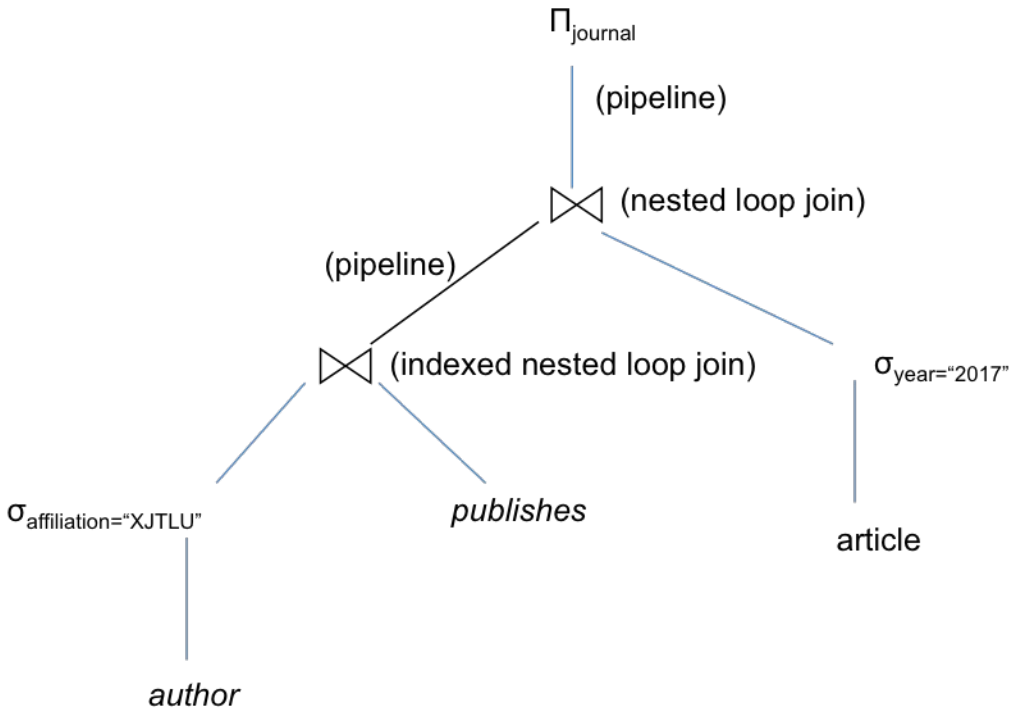

Π journal
(pipeline)
σ year="2017" ^ affiliation="XJTLU"
(pipeline)
(nested loop join)
(pipeline)
(indexed nested loop join)
article
author
publishes

Q2 cont'd

- 1) One of the heuristic rules for query optimisation is to perform selection operations as early as possible. In the above query tree, the selection predicate *affiliation*="XJTLU" can be performed on relation *author* first and *year*="2017" can be performed on relation *article* first. Draw the equivalent evaluation tree based on this heuristic rule and the equivalence rules for algebra expressions.
- 2) What is the estimated size of the operation "$\sigma_{affiliation="XJTLU"}(author) \bowtie publishes$"?
- 3) Assume that linear scan is used to evaluate all the selection operations in the plan. What is the total number of block transfers for the optimised evaluation plan? Note that no intermediate relations need to be saved as the result of using pipelining.

Q2 solution

1)



Q2 solution cont'd

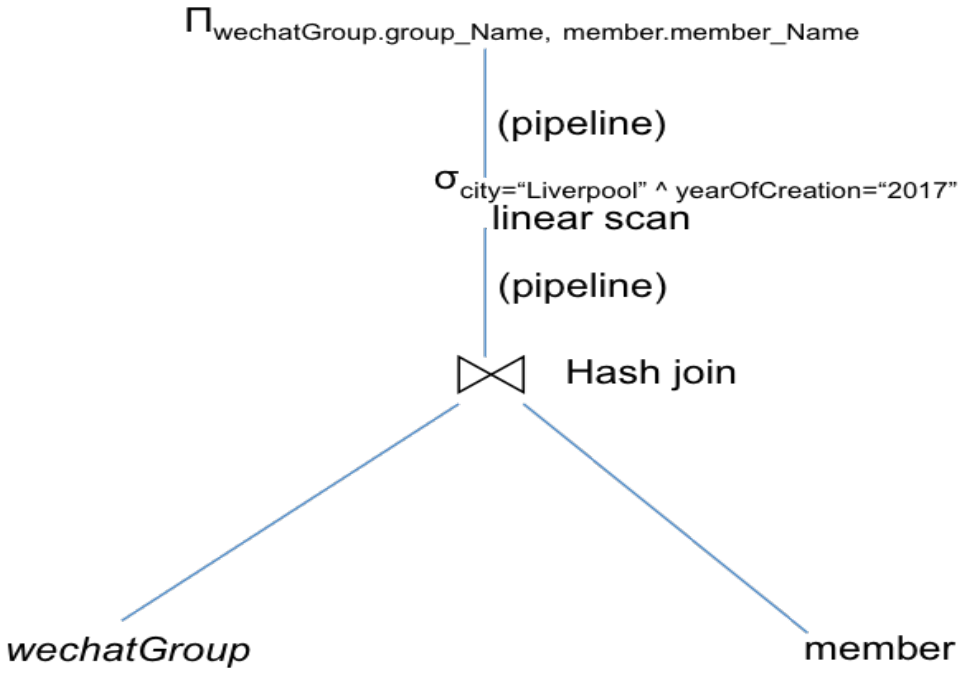
- 2) the above operation can be written as $\sigma_{affiliation="XJTLU"}$(author $\bowtie$ publishes). Since *authorID* in *publishes* is the foreign key referencing *author*, the estimated size is the number of records in relation *publishes*, which is 100,000. Then we perform the selection, *V(author, affiliation) = 500,* the size is 100,000/500=200.
- 3)
  - 1. For $\sigma_{affiliation="XJTLU"}$(author), the number of block transfer is 500;
  - 2. Since *V(author, affiliation)=500,* so the selection size is 1 block (10 records in relation *author*). Because that pipelining cannot be used for indexed nested loop join, so we need to store 1 block to disk;
  - 3. For "author $\bowtie$ publishes", it uses indexed nested loop join and the B+Tree index on *publishes*, the number of block transfer is 1+10(3+1)= 41;
  - 4. For $\sigma_{year="2017"}$(article), the number of block transfer is 2,000;
  - 5. As pipelining is used anywhere else, no block transfer is needed. So the total number of block transfer is 500+1+41+2,000 = 2,542

Q3

- Consider the following two relations and their catalog information. Answer the questions below.
  - *wechatGroup*(*gourp_ID*, *group_Name*, *size*, *yearOfCreation*, *owner*)
  - *member*(*member_ID*, *member_Name*, *city*, *group_ID*)
  - *group_ID* and *member_ID* are the keys for the two relations, respectively.
  - number of tuples in relation *wechatGroup*, n_r = 1,000
  - number of blocks in *wechatGroup*, b_r =200
  - number of distinct values on attribute *yearOfCreation* in *wechatGroup*, V(*yearOfCreation*, *wechatGroup*) = 100
  - number of tuples in relation *member*, n_s = 10,000
  - number of blocks in relation *member*, b_s =1,000
  - number of distinct values on attribute *city* in *member*, V(*city*, *member*) = 100

Q3 cont'd

- A query evaluation plan is shown below.



Q3 cont'd

- (1) What is the relational algebra expression for the given evaluation plan?
- (2) One of the heuristic rules for query optimisation is to perform selection operations as early as possible. Write the equivalent algebra expression for the answer from Question Q3.(1).
- (3) Suppose that all selections are evaluated by using linear scan and pipelining is used for projection. Draw an annotated evaluation tree for the relational algebra expression obtained from Question Q3.(2).
- (4) Assume that for each tuple in $\sigma_{yearOfCreation="2017"}(wechatGroup)$, the average cost of performing the hash join is 3 block transfers (including all the block transfer for relation partition and the build/probe procedure). What is the total number of block transfers for the whole evaluation plan in Question Q3.(3)?

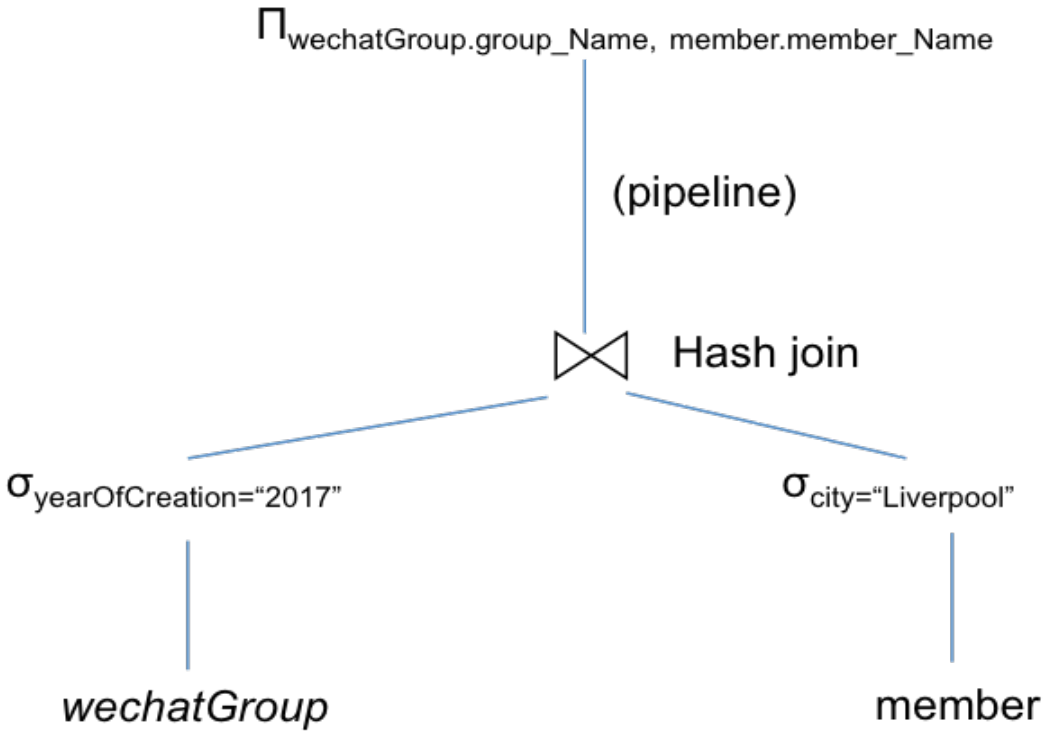
Q3 Solution

- (1) $\Pi_{\text{wechatGroup.group\_Name, member.member\_Name}}(\sigma_{\text{city="Liverpool"} \wedge \text{yearOfCreation="2017"}}(\text{wechatGroup} \bowtie \text{member}))$

- (2) $\Pi_{\text{wechatGroup.group\_Name, member.member\_Name}}((\sigma_{\text{yearOfCreation="2017"}}(\text{wechatGroup})) \bowtie (\sigma_{\text{city="Liverpool"}}(\text{member})))$

Q3 Solution cont'd

- (3)



Q3 Solution cont'd

- (4)
  - Linear scan $\sigma_{yearOfCreation="2017"}(wechatGroup)$ needs 200 block transfers.
  - The size is n_r/ V(A, r), which is 1,000/100 = 10 tuples (or 2 blocks). The result of this selection has to be stored temporarily in 2 blocks, as pipelining cannot be used for hash join.
  - Linear scan $\sigma_{city="Liverpool"}(member)$ needs 1,000 block transfers.
  - The result of this selection has to be stored temporarily in 1,000/100=10 blocks, as pipelining cannot be used for hash join.
  - It is said that for each tuple from $\sigma_{yearOfCreation="2017"}(wechatGroup)$, the average cost of performing the hash join is 3 block transfers (including all the block transfer for relation partition and the build/probe procedure). There are 10 tuples after the selection, so 30 block transfers are needed here.
  - As pipelining is used for projection so there are no block transfers. The number of block transfers for the whole evaluation plan is 200+2+1,000+10+30 = 1242.

Q4

The database of an online movie website has the following three relations.

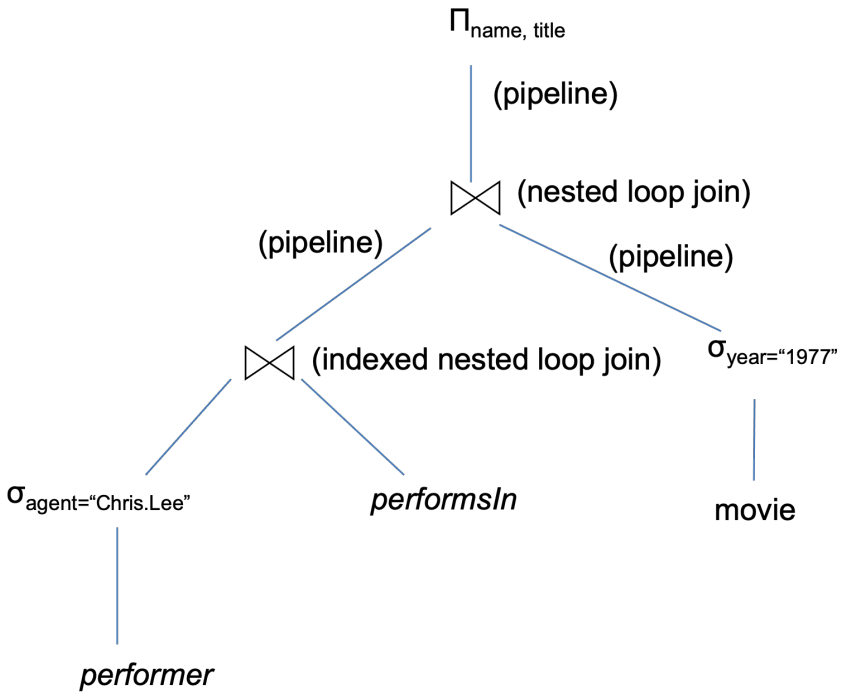
- *performer (pid, name, sex, agent, country, email)*
- *performsIn (pid, mid)*
- *movie (mid, title, year, language, producer)*

The attributes "*pid*" and "*mid*" are the keys for relations *performer* and *movie*, respectively. The two attributes in *performsIn* are the foreign keys referencing *performer* and *movie*, respectively. An agent for a performer is someone who signs a contract with and represents the performer. The catalog information is given below.

- In relation *performer*, number of records $n_{performer}$ = 3,800; number of blocks $b_{performer}$ = 520;
- In relation *performsIn*, number of records $n_{performsIn}$ = 70,050; number of blocks $b_{performsIn}$ = 86;
- In relation *movie*, number of records, n_{movie} = 31,000; number of blocks b_{movie} = 2,700;
- Index: a primary B^+-tree index of height 7 on the *pid* attribute of the *performsIn* relation;
- Number of distinct values for the attribute *agent* in the *performer* relation *V(performer, agent) = 500;*

Q4 cont'd

Assume that linear scan is used to evaluate all the selection operations, and no intermediate relations need to be stored as the result of using pipelining. Only an estimate is needed with the evaluation plan shown in the diagram below. What would be the total number of block transfers? Justify your answer.



Q4 Solutions

- (1) For $\sigma_{agent="Chris.Lee"(performer)}$, the number of block transfer is 520;
- (2) Since V(agent, performer)=500, so the selection size is 2 blocks (3800/500>3800/520; $\lceil 3800/500 \rceil$=8 tuples). Because that pipelining cannot be used for indexed nested loop join, so we need to store 2 blocks to disk;
- (3) For "performer $\bowtie$ performsIn", it uses indexed nested loop join and the B+Tree index on performsIn, the number of block transfer is 2+8(7+1)= 66.
- (4) For σyear="2017"(movie), the number of block transfer is 2,700;
- (5) As pipelining is used elsewhere, no block transfer is needed.
- (6) So the total number of block transfer is 520+2+66+2,700 = 3,288