Relational Algebra Heuristic Optimization

SWEN 304

Trimester 2, 2017

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

slides by: Pavle Morgan & Hui Ma

Summary or Relational Operations

- SELECT $\sigma_c(r(N))$: choose rows
- PROJECT $\pi_{A1,\ldots,Ak}(r(N))$: choose columns
- RENAME $\delta_{A1\rightarrow B1,\ldots,Ak\rightarrow Bk}(r(N))$: rename attributes
- JOIN: combine tables
  - Natural Join ($r(N_1)$ * $r(N_2)$ or
  - Equi-Join $r(N_1) \bowtie_{A1=B1,\ldots,Ak=Bk} r(N_2))$
- CARTESIAN PRODUCT (x): combine tables
- Set operations
  - UNION ($\cup$),
  - INTERSECTION ($\cap$),
  - DIFFERENCE (or MINUS, –)
- Additional Relational Operations
  - OUTER JOINS, AGGREGATE FUNCTIONS, e.g., SUM, COUNT, AVG, MIN, MAX

A Sample Relational Database

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Exercises

- Suppose we are given the university database instance as in slide 9. Write queries in relational algebra for the following queries
  1. Find all students with their ID who got at least one 'A+'
  2. Find students with their ID, FName, who have enrolled in C302
  3. Find students with their IDs who have enrolled in 'C201' but not 'C302'
  4. Find students who have enrolled in both 'M214' and 'C302'
  5. Find students who have neither enrolled in 'M214' nor in 'C302'
  6. Find students who major in 'Math' and got 'A+' in at least one course offered by computer science department

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

1. Find all students with their student ID who have got at least one 'A+'

$$\pi_{StudId}\ (\sigma_{Grade='A+'}\ (Grades))$$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

2. Find students with their ID, FName, who have enrolled in 'C302'

$\pi_{\text{StudId, FName}} (\sigma_{\text{CourId = 'C302'}} (\text{Student} * \text{Grades}))$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

3. Find students with their IDs who have enrolled in 'C201' but not 'C302'

$\pi_{StudId}(\sigma_{CourId = 'C201'}(Grades)) - \pi_{StudId}(\sigma_{CourId = 'C302'}(Grades))$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

4. Find students who have enrolled in both 'M214' and 'C302'

$$\pi_{StudId}(\sigma_{CourId = 'M214'}(Grades)) \cap \pi_{StudId}(\sigma_{CourId = 'C302'}(Grades))$$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

5. Find students who have neither enrolled in 'M214' nor in 'C302'

$$(\pi_{StudId}(Student) - \pi_{StudId}(\sigma_{CourId = 'M214'}(Grades))) - \pi_{StudId}(\sigma_{CourId = 'C302'}(Grades))$$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

6. Find students who major in 'Math' and got 'A+' in at least one course offered by computer science department

$\pi_{StudId}\ (\sigma_{Grade='A+' \wedge Major = 'Math' \wedge Dept = 'Comp'}\ (Course * (Student * Grades)))$

Exercises

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Find students who always take courses from Comp department.

Heuristic Query Optimization

- Process for heuristics optimization
  1. The parser of a high-level query generates an initial internal representation
  2. Apply heuristics rules to optimize the internal representation
  3. A query execution plan is generated to execute groups of operations based on the access paths available on the files involved in the query

- The main heuristic is to apply first the operations that reduce the size of intermediate results
  - E.g., Apply SELECT and PROJECT operations before applying the JOIN or other binary operations

Using Heuristics in Query Optimization (1)

- General Transformation Rules for Relational Algebra Operations:

1. Cascade of σ: A conjunctive selection condition can be broken up into a cascade (sequence) of individual σ operations:
   - $\sigma_{c1 \wedge c2 \wedge \ldots \wedge cn}(r(N)) = \sigma_{c1}(\sigma_{c2}(\ldots(\sigma_{cn}(r(N)))\ldots))$
2. Commutativity of σ: The σ operation is commutative:
   - $\sigma_{c1}(\sigma_{c2}(r(N))) = \sigma_{c2}(\sigma_{c1}(r(N)))$
3. Cascade of π: In a cascade (sequence) of π operations, all but the last one can be ignored:
   - $\pi_{List1}(\pi_{List2}(\ldots(\pi_{Listn}(r(N)))\ldots)) = \pi_{List1}(r(N))$
4. Commuting σ with π: If the selection condition c involves **only** the attributes A1, ..., An in the projection list, the two operations can be commuted:
   - $\pi_{A1, A2, \ldots, An}(\sigma_c(r(N))) = \sigma_c(\pi_{A1, A2, \ldots, An}(r(N)))$

Using Heuristics in Query Optimization (2)

- General Transformation Rules for Relational Algebra Operations (contd.):

5. Commutativity of ⋈ (and x): The ⋈ operation is commutative as is the x operation:
   - $r(N_1) \bowtie_C r(N_2) = r(N_2) \bowtie_C r(N_1)$; $r(N_1) \times r(N_2) = r(N_2) \times r(N_1)$
6. Commuting σ with ⋈ (or x): If all the attributes in the selection condition c involve only the attributes of one of the relations being joined—say, N_1—the two operations can be commuted as follows:
   - $\sigma_c (r(N_1) \bowtie r(N_2)) = \sigma_c (r(N_1)) \bowtie r(N_2)$
- Alternatively, if the selection condition c can be written as (c1 and c2), where condition c1 involves only the attributes of N_1 and condition c2 involves only the attributes of N_2, the operations commute as follows:
   - $\sigma_c(r(N_1) \bowtie r(N_2)) = \sigma_{c1}(r(N_1)) \bowtie \sigma_{c2} (r(N_2))$

Using Heuristics in Query Optimization (3)

- General Transformation Rules for Relational Algebra Operations (contd.):

7. Commuting π with $\bowtie$ (or x): Suppose that the projection list is AL = $\{A_1, ..., A_n, B_1, ..., B_m\}$, where $A_1, ..., A_n$ are attributes of N_1 and $B_1, ..., B_m$ are attributes of N_2. If the join condition c involves only attributes in AL, the two operations can be commuted as follows:
   - $\pi_{AL}(r(N_1) \bowtie_C r(N_2)) = (\pi_{A1, ..., An}(r(N_1))) \bowtie_C (\pi_{B1, ..., Bm}(r(N_2)))$
- If the join condition C contains additional attributes not in AL, these must be added to the projection list, and a final π operation is needed.
   - $\pi_{AL}(r(N_1) \bowtie_C r(N_2)) = \pi_{AL}(\pi_{AL1}(r(N_1)) \bowtie_C \pi_{AL2}(r(N_2)))$,

   where AL_i contains the common attributes in Ni and AL and the common attributes of N1 and N2

Using Heuristics in Query Optimization (4)

- General Transformation Rules for Relational Algebra Operations (contd.):

8. Commutativity of set operations: The set operations ∪ and ∩ are commutative but "–" is not.
9. Associativity of ⋈ , x, ∪, and ∩ : These four operations are individually associative; that is, if θ stands for any one of these four operations (throughout the expression), we have
   - $(r(N_1)\ \theta\ r(N_2))\ \theta\ r(N_3) = r(N_1)\ \theta\ (r(N_2)\ \theta\ r(N_3))$
10. Commuting σ with set operations: The σ operation commutes with ∪ , ∩ , and –. If θ stands for any one of these three operations, we have
    - $\sigma_c\ (r(N_1)\ \theta\ r(N_2)) = (\sigma_c\ (r(N_1)))\ \theta\ (\sigma_c\ (r(N_2)))$

Using Heuristics in Query Optimization (5)

- General Transformation Rules for Relational Algebra Operations (contd.):

11. The π operation commutes with $\cup$.

$$\pi_{AL}(r(N_1) \cup r(N_2)) = (\pi_{AL}(r(N_1))) \cup (\pi_{AL}(r(N_2)))$$

12. Converting a (σ, x) sequence into $\bowtie$: If the condition c of a σ that follows a x Corresponds to a join condition, convert the (σ, x) sequence into a $\bowtie$ as follows:

$$(\sigma_C(r(N_1) \times r(N_2))) = (r(N_1) \bowtie_C r(N_1))$$

Using Heuristics in Query Optimization (6)

- Summary of Heuristics for Algebraic Optimization:
  1. The main heuristic is to apply first the operations that reduce the size of intermediate results.
  2. Perform select operations as early as possible to reduce the number of tuples and perform project operations as early as possible to reduce the number of attributes. (This is done by moving select and project operations as far down the tree as possible, e.g. rule 6, 7, 10, 11)
  3. The select and join operations that are most restrictive should be executed before other similar operations. (This is done by reordering the leaf nodes of the tree among themselves and adjusting the rest of the tree appropriately.)

Heuristic Query Optimization Exercise

- For a given relation schema:

  MOVIE(title, genre, director)

  ACTOR(mtitle, name, year_born)

  MOVIE_AWARD(title, production_year, award_name year_of_award)

- Draw query trees for the following queries

$\pi_{title}\ (\sigma_{genre=\ 'war'}(MOVIE) \bowtie_{title\ =\ mtitle} \sigma_{name=\ 'Tom\ Cruise'}(ACTOR))$

$\pi_{director}\ (MOVIE\ *\ \sigma_{award\_name\ =\ 'Oscar'}(MOVIE\_AWARD))$

- Optimize the query trees using heuristic rules