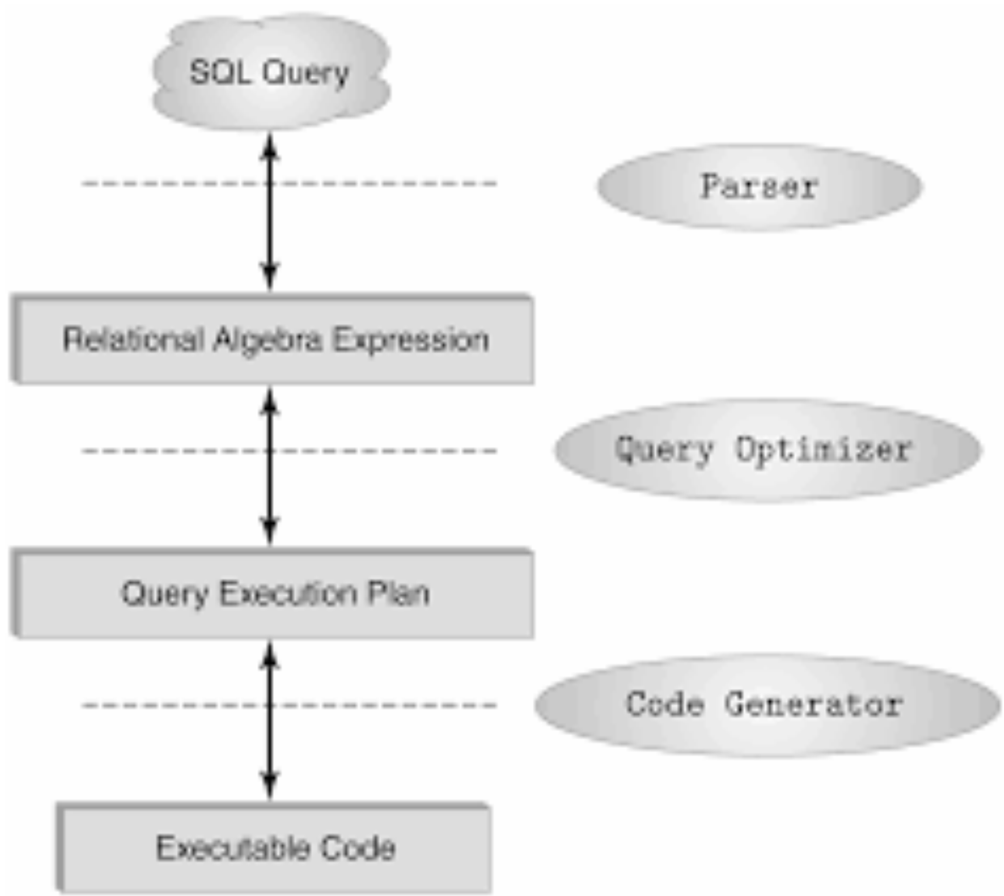
Lecture 6
Relational Algebra

What is Relational Algebra?

- A set of operations each of which takes a relation (or relations) as input and produces a relation as output
- **Procedural:** specify how to construct the result
- Two groups of operations
  - Set operations from mathematical set theory
    - Union
    - Intersection
    - Set Difference
    - Cartesian Product
  - Operations developed specially for relational databases
    - Project
    - Select
    - Join and others

Schematic View of Query Processing



SELECT

- Selects the tuples (rows) from a relation R that satisfy a certain *selection condition* c
- Form of the operation: σ_{C} (R)
- The condition c is a Boolean expression on the attributes of R of two forms:
  - <attribute name> <comparison op> <constant value>
  - <attribute name> <comparison op> <attribute name>
- Resulting relation has the *same attributes* as R and includes each tuple whose attribute values satisfy the condition c

SELECT Example

Routes

RId	RName	Grade	Rating	Height
1	Last Tango	II	12	100
2	Garden Path	I	2	60
3	The Sluice	I	8	60
4	Picnic	III	3	400

$\sigma_{Height=60}\text{Routes}:$

RId	RName	Grade	Rating	Height
2	Garden Path	I	2	60
3	The Sluice	I	8	60

What Can Go in a Selection Condition?

- Conditions are built up from Boolean-valued operations on the field names
  - E.g., Height >=100, RName = ‘Picnic’
- Clauses can be arbitrarily connected by the Boolean operator OR, AND, NOT to form a general selection condition
  - E.g., $\sigma_{(\text{Dno}=4\ \textbf{AND}\ \text{Salary}>25000)\ \textbf{OR}\ (\text{Dno}=5\ \textbf{AND}\ \text{Salary}>30000)}$ (EMPLOYEE)

Characteristics of Selection

- Selection operator is unary
- Apply to a single relation R
- Test each tuple individually
- Degree of the relation resulting from a SELECT operation = the degree of R
- # of tuples in the resulting relation is always less than or equal to the # of tuples in R
- σ is commutative

PROJECT

- Select some columns from the table and discard other columns
- Keeps only certain attributes (columns) from a relation R specified in an *attribute list* L
- Form of operation: $\pi_{L}(R)$

PROJECT Example

Routes:

RId	RName	Grade	Rating	Height
1	Last Tango	II	12	100
2	Garden Path	I	2	60
3	The Sluice	I	8	60
4	Picnic	III	3	400

$\pi_{RId,Height}\text{Routes}:$

RId	Height
1	100
2	60
3	60
4	400

Projection Discussion

- Suppose the result of a projection has a repeated value, how do we treat it?

$$\pi_{Height}\text{Routes}$$

Height
100
60
60
400

Height
100
60
400

- In “pure” relational algebra the answer is always a set (the second answer)
- However SQL and some other languages return, by default, a multiset

Characteristics of Projection

- Projection operator is unary
- Apply to a single relation R
- The degree of the resulting relation is equal to the number of attributes in <attribute list>
- # of tuples in the resulting relation is always less than or equal to # of tuples in R
- π is not commutative

Database Queries

- Queries are formed by building up expressions with the operations of the relational algebra
- For example, select-project expressions are very common:

$$\pi_{Name,Age}(\sigma_{Age \geq 30}\text{Climbers})$$

  - What does this mean in English?
  - Also, could we interchange the order of the σ and π? Can we always do this?

Sequences of Operations

- Write the operations as a single **relational algebra expression** by nesting the operations
- E.g., to retrieve the first name, last name, and salary of all employees who work in department number 5, we can apply a select and a project operation

$$\pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\text{EMPLOYEE}))$$

- OR explicitly show the sequence of operations, giving a name to each intermediate relation:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\text{EMPLOYEE})$$

$$\text{RESULT} \leftarrow \pi_{\text{FNAME, LNAME, SALARY}}(\text{DEP5\_EMPS})$$

Set Operations

- Binary operations from mathematical set theory
  - UNION: $R_1 \cup R_2$
  - INTERSECTION: $R_1 \cap R_2$
  - SET DIFFERENCE: $R_1 - R_2$
  - CARTESIAN PRODUCT: $R_1 \times R_2$

Union Compatibility

- For $\cup$, $\cap$, $-$, the operand relations $R_1(A_1, A_2, ..., A_n)$ and $R_2(B_1, B_2, ..., B_n)$ must be union-compatible if
  - R_1 and R_2 have the same number of attributes, and
  - $dom(A_i)=dom(B_i)$ for $i=1, 2, ..., n$
- The resulting relation for $R_1 \cup R_2$, or $R_1 \cap R_2$, or $R_1 - R_2$ has the same attribute names as the *first* operand relation R_1 (by convention)

Do we need to check union compatibility for Cartesian product?

Yes

No

Do we need to check union compatibility for Cartesian product?

Yes

0%

No

0%

Do we need to check union compatibility for Cartesian product?

Yes

0%

No

0%

UNION

- Two relations are union-compatible
- $R_1 \cup R_2$: includes all tuples that are either in R_1 or in R_2 or in both R_1 and R_2
- Duplicates are eliminated

UNION Example

Climbers:

CId	CName	Skill	Age
123	Edmund	EXP	80
214	Arnold	BEG	25
313	Bridget	EXP	33
212	James	MED	27

Hikers:

CId	CName	Skill	Age
214	Arnold	BEG	25
898	Jane	MED	39

$\text{Climbers} \cup \text{Hikers}:$

CId	CName	Skill	Age
123	Edmund	EXP	80
214	Arnold	BEG	25
313	Bridget	EXP	33
212	James	MED	27
898	Jane	MED	39

INTERSECTION

- Two relations are union-compatible
- $R_1 \cap R_2$: includes all tuples that are in both R_1 and R_2

INTERSECTION Example

Climbers:

CId	CName	Skill	Age
123	Edmund	EXP	80
214	Arnold	BEG	25
313	Bridget	EXP	33
212	James	MED	27

Hikers:

CId	CName	Skill	Age
214	Arnold	BEG	25
898	Jane	MED	39

$\text{Climbers} \cap \text{Hikers}:$

CId	CName	Skill	Age
214	Arnold	BEG	25

SET DIFFERENCE (or MINUS)

- Two relations are union-compatible
- $R_1 - R_2$: includes all tuples that are in R_1 but not in R_2

SET DIFFERENCE Example

Beginners:

CId	CName	Skill	Age
214	Arnold	BEG	25
212	James	MED	27

Climbers:

CId	CName	Skill	Age
123	Edmund	EXP	80
214	Arnold	BEG	25
313	Bridget	EXP	33
212	James	MED	27

Climbers – Beginners:

CId	CName	Skill	Age
123	Edmund	EXP	80
313	Bridget	EXP	33

Characteristics of Set Operations

- Both union and intersection are *commutative operations*

  $\mathbf{R \cup S = S \cup R}$, **and** $\mathbf{R \cap S = S \cap R}$

- Both union and intersection are *associative operations*

  $\mathbf{R \cup (S \cup T) = (R \cup S) \cup T}$, **and**

  $\mathbf{(R \cap S) \cap T = R \cap (S \cap T)}$

- The minus operation is *not commutative*

  $\mathbf{R - S \neq S - R}$

Exercise

- Customer: C_Name, C_Street, C_City
- Depositor: C_Name, Account#
- Loan: B_Name, Loan_#, Amount
- Borrower: C_Name, Loan#

Exercise (Cont.)

- Select tuples of the loan relation where the branch name is “GMU”.
- Find all tuples in which the amount of loan is more than $12000.
- Find those tuples pertaining to loans of more than $12000 made by “GMU” branch.
- List all loan numbers and the amount of loan.
- Find the names of all customers who live in “Fairfax”.
- Find the names of all bank customers who have either an account or a loan or both.
- Find the names of all customers of the bank who have an account but not a loan.
- Find the names of all customers who have both a loan and an account.