CSCI4333 Database Design & Implement

Lecture 12 – Relational Algebra 1

Instructor: Dr. Yifeng Gao

Relational Query Languages

- *Query languages*: Allow manipulation and retrieval of data from a database.
- Relational model supports simple, powerful QLs:
  - Strong formal foundation based on logic.
  - Allows for much optimization.
- Query Languages != programming languages!
  - QLs not expected to be "Turing complete".
  - QLs not intended to be used for complex calculations.
  - QLs support easy, efficient access to large data sets.

Formal Relational Query Languages

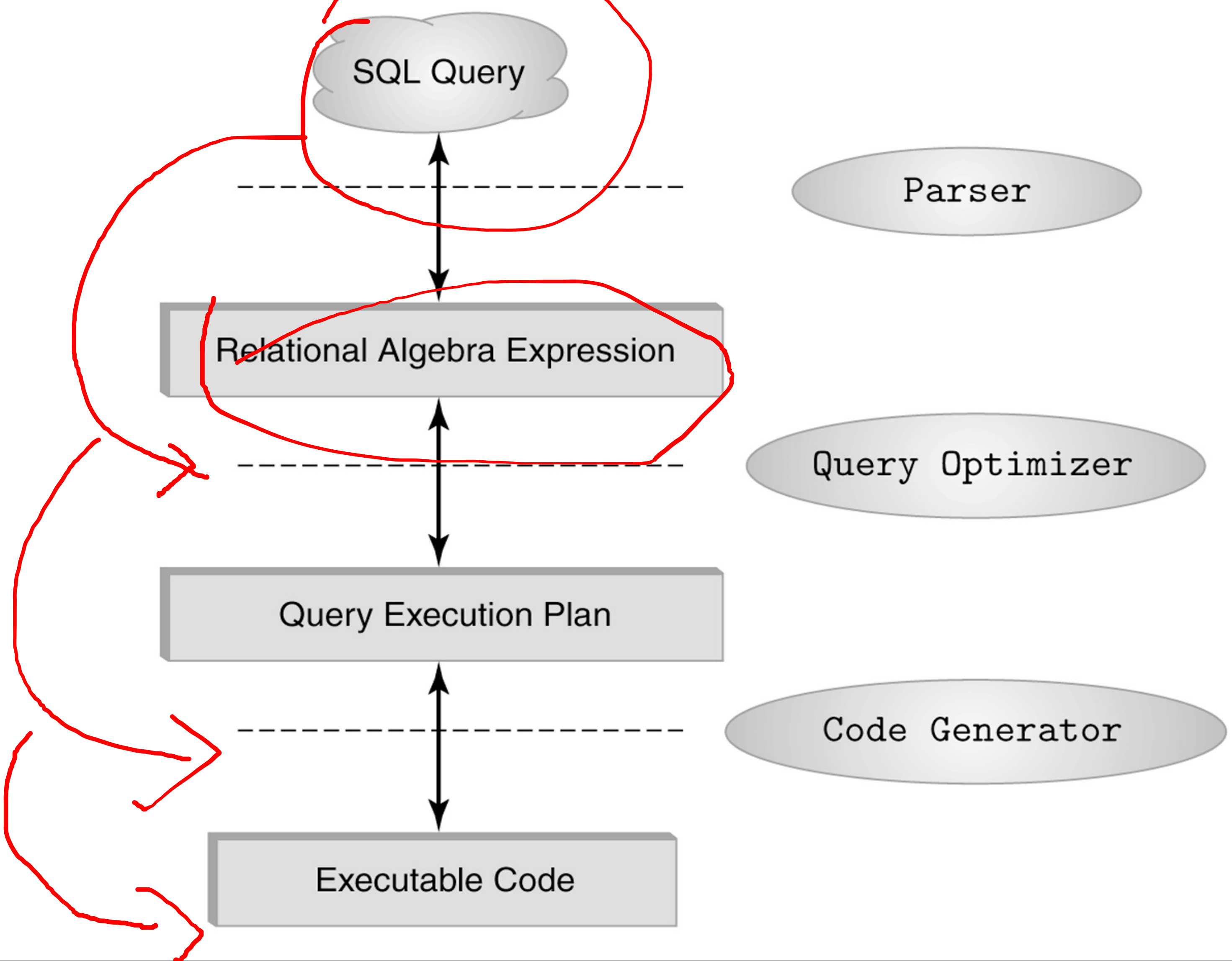
Two mathematical Query Languages form the basis for "real" languages (e.g. SQL), and for implementation:

❶ *Relational Algebra*: More operational, very useful for representing execution plans.

❷ *Relational Calculus*: Lets users describe what they want, rather than how to compute it. (Non-operational, *declarative*.)

☛ *Understanding Algebra is key to understanding SQL, and query processing!*

The Role of Relational Algebra in a DBMS



Algebra Preliminaries

- A query is applied to *relation instances*, and the result of a query is also a relation instance.

Relational Algebra

SQL is closely based on relational algebra.

- Procedural language
- Five basic operators
  - **selection** select
  - **projection** project
  - **union** (why no intersection?)
  - **set difference** difference
  - **cross product** Cartesian product

- The are some other operators which are composed of the above operators. These show up so often that we give them special names.
- The operators take one or two relations as inputs and give a new relation as a result.

Select Operation – Example

- Relation *r*

A	*B*	*C*	*D*
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

<u>Intuition</u>: The **select** operation allows us to retrieve some rows of a relation

Ex:

If I want to retrieve all the rows of the relation *r* where

1. the value in field *A* equals the value in field *B*,
2. the value in field *D* is greater than 5.

- $\sigma_{A=B \wedge D > 5}(r)$

A	*B*	*C*	*D*
α	α	1	7
β	β	23	10

lowercase Greek sigma

Select Operation

- Notation: $\sigma_p(r)$ lowercase Greek sigma σ
- p is called the **selection** predicate
- Defined as:

$$\sigma_p(\boldsymbol{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$$

Where p is a formula consisting of *terms* connected by : $\wedge$ **(and)**, $\vee$ **(or)**, $\neg$ **(not)**
Each *term* is one of:

<attribute> *op* <attribute> or <constant>

where *op* is one of: $=, \neq, >, \geq, <, \leq$

- Example of selection:

$$\sigma_{name='Lee'}(professor)$$

Quick Question

- Notation: $\sigma_p(r)$,
- What is *p*, *term*, *op*, *r* in the following query?
  - $\sigma_{A=B \wedge D > 5}(r)$
  - $\sigma_{name= \text{'Lee'}}(professor)$
  - p = A=B ^ D > 5
  - A=B, op "="
  - D > 5, op ">"

Project Operation – Example II

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- $\pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

<u>Intuition</u>: The project operation removes the columns that are not listed in the notation. Duplicate rows are removed, since relations are sets.

Here there are two rows with $A = \alpha$ and $C = 1$. So one was discarded.

Project Operation

- Notation:

$$\pi_{A1,\ A2,\ \ldots,\ Ak}\ (r)$$

Greek lower-case pi

where A_1, A_2 are attribute names and r is a relation name.

- The result is defined as the relation of k columns obtained by erasing the columns that are not listed
- Duplicate rows removed from result, since relations are sets.

Quick Question

- What is result of: $\pi_{A}(r)$

A	*B*	*C*
α	10	1
α	20	1
β	30	1
β	40	2

A
α
β

Union Operation – Example

Relations *r, s:*

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cup s$:

A	B
α	1
α	2
β	1
β	3

Intuition: The **union** operation concatenates two relations **vertically** and removes duplicate rows (since relations are sets).

Union Operation

- Notation: $r \cup s$
- Defined as:

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

"Union-compatible"

For $r \cup s$ to be valid.

1. *r, s* must have the *same arity* (same number of attributes)
2. The attribute domains must be *compatible* (e.g., 2nd column of *r* deals with the same type of values as does the 2nd column of *s*).

Although the field types must be the same, the names can be different. For example:

professor(*PID* : string, *name* : string)

lecturer(*LID* : string, *first_name* : string)

professor $\cup$ *lecturer* is valid.

Set Difference Operation – Example

Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r – s:

A	B
α	1
β	1

Intuition: The **set difference** operation returns all the rows that are in *r* but not in *s*.

Set Difference Operation

- Notation $r - s$
- Defined as:

$$r - s = \{t \mid t \in r \textbf{ and } \mathrm{t} \notin s\}$$

- Set differences must be taken between *compatible* relations.

"Union-compatible"

  - r and s must have the *same arity*
  - attribute domains of r and s must be compatible
- Note that in general $r - s \neq s - r$

Cross-Product Operation-Example

Relations *r, s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x *s*:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Intuition: The **cross product** operation returns all possible combinations of rows in *r* with rows in *s*.

In other words the result is every possible pairing of the rows of *r* and *s*.

Cross-Product Operation

- Notation $r \times s$
- Defined as:

$$r \times s = \{t\,q \mid t \in r \textbf{ and } q \in s\}$$

- Assume that attributes of r(R) and s(S) are disjoint. (That is, $R \cap S = \emptyset$).
- If attributes names of *r(R)* and *s(S*) are not disjoint, then renaming must be used.

Composition of Operations

- We can build expressions using multiple operations
- Example: $\sigma_{A=C}(r \times s)$

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

$r \times s$:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$\sigma_{A=C}(r \times s)$ ⇨

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

"take the cross product of *r* and *s*, then return only the rows where *A* equals *C*"

Question

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

What is result of: $\sigma_{A=\alpha}(r) \; x \; \pi_{C,D}\left(\sigma_{E=a}(s)\right)$?

Question

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

What is result of: $\boldsymbol{\pi_A(r) \cup \pi_C(s)}$?