DATABASE TECHNOLOGY - 1MB025

Fall 2005

An introductory course on database systems

http://user.it.uu.se/~udbl/dbt-ht2005/
alt. http://www.it.uu.se/edu/course/homepage/dbastekn/ht05/

Kjell Orsborn
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden

Introduction to Relational Algebra

Elmasri/Navathe ch 6
Padron-McCarthy/Risch ch 10??

Kjell Orsborn

Department of Information Technology
Uppsala University, Uppsala, Sweden

Query languages

- Languages where users can express what information to retrieve from the database.
- Categories of query languages:
  - Procedural
  - Non-procedural (declarative)
- Formal ("pure") languages:
  - Relational algebra
  - Relational calculus
    - Tuple-relational calculus
    - Domain-relational calculus
  - Formal languages form underlying basis of query languages that people use.

Relational algebra

- **Relational algebra** is a procedural langaue
- Operations in relational algebra takes two or more relations as arguments and return a new relation.
- Relational algebraic operations:
  - Operations from set theory:
    - Union, Intersection, Difference, Cartesian product
  - Operations specifically introduced for the relational data model:
    - Select, Project, Join
- It have been shown that the *select, project, union, difference,* and *cartesian product* operations form a complete set. That is any other relational algebra operation can be expressed in these.

Operations from set theory

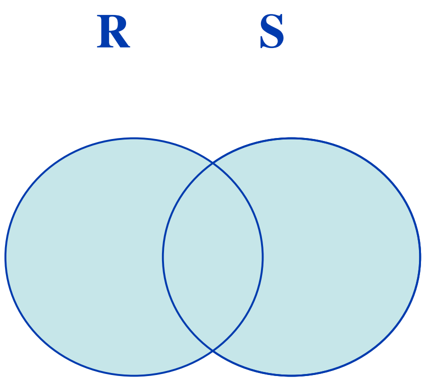
- Relations are required to be **union compatible** to be able to take part in the union, intersection and difference operations.
- Two relations R_1 and R_2 is said to be union-compatible if:

$R_1 \subseteq D_1 \times D_2 \times \ldots \times D_n$ and
$R_2 \subseteq D_1 \times D_2 \times \ldots \times D_n$

i.e. if they have the same degree and the same domains.

Union operation

- The **union** of two union-compatible relations *R* and *S* is the set of all tuples that either occur in *R*, *S*, or in both.
- Notation: $R \cup S$
- Defined as: $R \cup S = \{t \mid t \in R \text{ or } t \in S\}$
- For example:



A	*B*
a	1
a	2
b	1

$\cup$

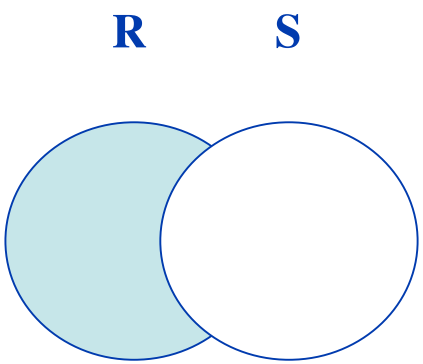
A	*B*
a	2
b	3

$=$

A	*B*
a	1
a	2
b	1
b	3

Difference operation

- The **difference** between two union-compatible sets *R* and *S* is the set of all tuples that occur in *R* but not in *S*.
- Notation: R – S
- Defined as: $R - S = \{t \mid t \in R \text{ and } t \notin S\}$
- For example:



A	*B*
a	1
a	2
b	1

–

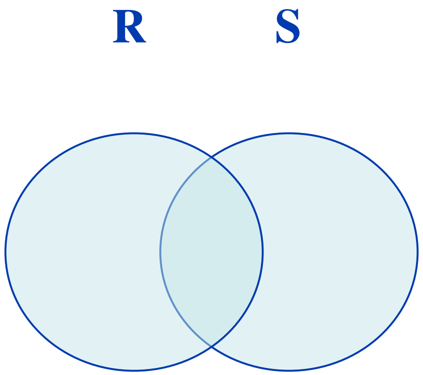
A	*B*
a	2
b	3

=

A	*B*
a	1
b	1

Intersection

- The **intersection** of two union-compatible sets *R* and *S*, is the set of all tuples that occur in both *R* and *S*.
- Notation: $R \cap S$
- Defined as: $R \cap S = \{t \mid t \in R \text{ and } t \in S\}$
- For example:



A	*B*
a	1
a	2
b	1

$\cap$

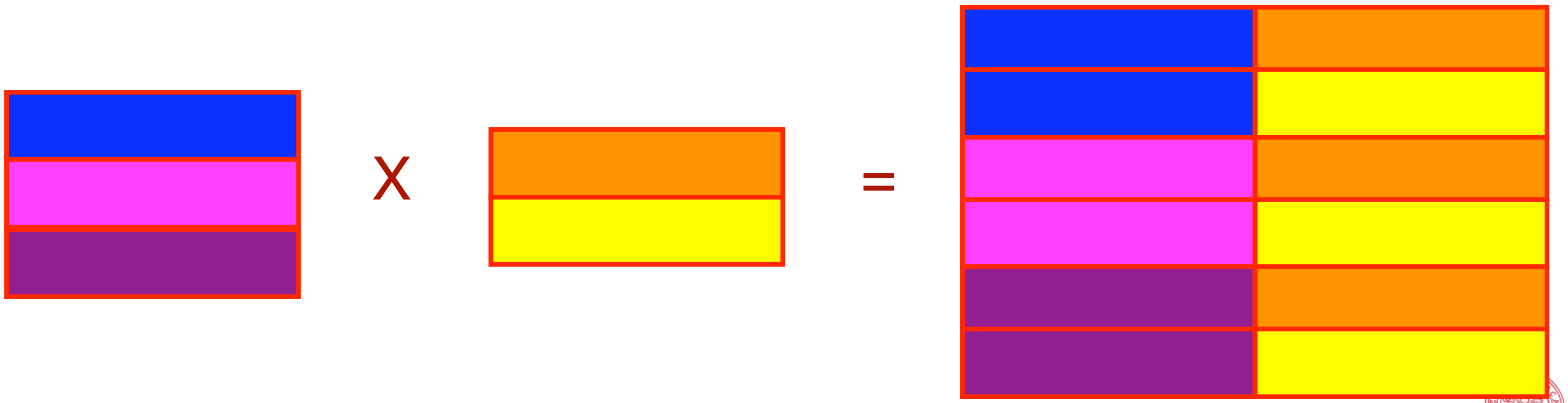
A	*B*
a	2
b	3

$=$

A	*B*
a	2

Cartesian product

- Let R and S be relations with k1 and k2 arities resp. The **cartesian product** of *R* and *S* is the set of all possible k_1+k_2 tuples where the first k_1 components constitute a tuple in *R* and the last k_2 components a tuple in *S*.
- Notation: $R \times S$
- Defined as: $R \times S = \{t\,q \mid t \in R \text{ and } q \in S\}$
- Assume that attributes of r(R) and s(S) are disjoint. (i.e. $R \cap S = \emptyset$). If attributes of r(R) and s(S) are not disjoint, then renaming must be used.



Cartesian product example

A	B
a	1
b	2

×

C	D
a	5
b	5
b	6
c	5

=

A	B	C	D
a	1	a	5
a	1	b	5
a	1	b	6
a	1	c	5
b	2	a	5
b	2	b	5
b	2	b	6
b	2	c	5

Selection operation

- The selection operator, σ, selects a specific set of tuples from a relation according to a selection condition (or selection predicate) *P*.
- Notation: $\sigma_p(R)$
- Defined as: $\sigma_p(R) = \{t \mid t \in R \text{ and } P(t)\}$ (i.e. the set of tuples t in *R* that fulfills the condition *P*)
- Where *P* is a logical expression(*) consisting of terms connected by:
  ∧ (**and**), ∨ (**or**), ¬ (**not**)
  and each term is one of:
  <attribute> *op* <attribute> or <constant>
  where *op* is one of: =, ≠, >, ≥, <, ≤

  Example: $\sigma_{SALARY>30000}(EMPLOYEE)$

(*) a formula in propositional calculus

Selection example

$$R \quad = $$

A	*B*	*C*	*D*
a	a	1	7
a	b	5	7
b	b	2	3
b	b	4	9

$$\sigma_{A=B \wedge D > 5}(R) \quad = $$

A	*B*	*C*	*D*
a	a	1	7
b	b	4	9

Projection operation

- The **projection** operator, Π, picks out (or projects) listed columns from a relation and creates a new relation consisting of these columns.
- Notation: $\Pi_{A_1, A_2, \ldots, A_k}(R)$
  where A_1, A_2 are attribute names and R is a relation name.
- The result is a new relation of k columns.
- Duplicate rows removed from result, since relations are sets.

Example: $\Pi_{LNAME, FNAME, SALARY}(EMPLOYEE)$

Projection example

R =

A	*B*	*C*
a	1	1
a	2	1
b	3	1
b	4	2

$\Pi_{A,C}(\mathrm{R})$ =

A	*C*
a	1
~~a~~	~~1~~
b	1
b	2

=

A	*C*
a	1
b	1
b	2

Join operator $\bowtie$

- The **join** operator, $\otimes$ (almost, correct $\bowtie$), creates a new relation by joining related tuples from two relations.
- Notation: R $\otimes_C$ S
  C is the join condition which has the form $A_r \, \theta \, A_s$, where θ is one of {=, <, >, ≤, ≥, ≠}. Several terms can be connected as $C_1 \wedge C_2 \wedge \ldots C_k$.
- A join operation with this kind of general join condition is called "Theta join".

Example Theta join

R

A	B	C
1	2	3
6	7	8
9	7	8

$\otimes_{A \leq D}$

S

B	C	D
2	3	4
7	3	5
7	8	9

=

$R \otimes_{A \leq D} S$

A	B	C	B	C	D
1	2	3	2	3	4
1	2	3	7	3	5
1	2	3	7	8	9
6	7	8	7	8	9
9	7	8	7	8	9

Equijoin

- The same as join but it is required that attribute A_r and attribute A_s should have the same value.
- Notation: R $\otimes_C$ S
  C is the join condition which has the form $A_r = A_s$. Several terms can be connected as $C_1 \wedge C_2 \wedge \ldots C_k$.

Example Equijoin

R S $R \otimes_{B=C} S$

A	B
a	2
a	4

$\otimes_{B=C}$

C	D	E
2	d	e
4	d	e
9	d	e

=

A	B	C	D	E
a	2	2	d	e
a	4	4	d	e

Natural join

- **Natural join** is equivalent with the application of join to R and S with the equality condition $A_r = A_s$ (i.e. an equijoin) and then removing the redundant column A_s in the result.
- Notation: R $*_{Ar,As}$ S
  A_r, A_s are attribute pairs that should fulfil the join condition which has the form $A_r = A_s$. Several terms can be connected as $C_1 \wedge C_2 \wedge ... C_k$.

Example Natural join

R S $R *_{B=C} S$

A	*B*
a	2
a	4

$\otimes_{B=C}$

C	*D*	*E*
2	d	e
4	d	e
9	d	e

=

A	*B*	*D*	*E*
a	2	d	e
a	4	d	e

UPPSALA UNIVERSITET

Composition of operations

- Expressions can be built by composing multiple operations
- Example: $\sigma_{A=C}(R \times S)$

$R \times S$ =

A	B
a	1
b	2

$\times$

C	D
a	5
b	5
b	6
c	5

=

A	B	C	D
a	1	a	5
a	1	b	5
a	1	b	6
a	1	c	5
b	2	a	5
b	2	b	5
b	2	b	6
b	2	c	5

$\sigma_{A=C}(R \times S)$ =

A	B	C	D
a	1	a	5
b	2	b	5
b	2	b	6

Assignment operation

- The assignment operation ($\leftarrow$) makes it possible to assign the result of an expression to a temporary relation variable.
- Example:

$$temp \leftarrow \sigma_{dno = 5}(EMPLOYEE)$$

$$result \leftarrow \Pi_{fname,lname,salary}(temp)$$

- The result to the right of the $\leftarrow$ is assigned to the relation variable on the left of the $\leftarrow$.
- The variable may use variable in subsequent expressions.

Renaming relations and attribute

- The assignment operation can also be used to rename relations and attributes.
- Example:
  NEWEMP $\leftarrow \sigma_{dno = 5}$(EMPLOYEE)
  R(FIRSTNAME,LASTNAME,SALARY) $\leftarrow$
  $\Pi_{fname,lname,salary}$ (NEWEMP)

UPPSALA UNIVERSITET

Division operation

- Suited to queries that include the phrase "for all".
- Let R and S be relations on schemas R and S respectively, where
  $R = (A_1,\ldots,A_m ,B_1,\ldots,B_n)$
  $S = (B_1,\ldots,B_n)$
- The result of R ÷ S is a relation on schema
  $R - S = (A_1,\ldots,A_m)$
  $R \div S = \{t \mid t \in \Pi_{R\text{-}S}(R)\ \forall u \in S \wedge tu \in R\}$

Example Division operation

R   S $R \div S$

A	B
a	1
a	2
a	3
b	1
c	1
d	1
d	3
d	4
d	6
e	1
e	2

÷

B
1
2

=

A
a
e

Relation algebra as a query language

- Relational schema: *supplies(sname, iname, price)*
- "What is the names of the suppliers that supply cheese?"

$$\pi_{sname}(\sigma_{iname='CHEESE'}(SUPPLIES))$$

- "What is the name and price of the items that cost less than 5 \$ and that are supplied by WALMART"

$$\pi_{iname,price}(\sigma_{sname='WALMART' \wedge price < 5}(SUPPLIES))$$

UPPSALA UNIVERSITET

Additional relational operations

- Outer join and outer union (presented together with SQL)
- Aggregate functions (presented together with SQL)
- Update operations (presented together with SQL)
  - (not part of pure query language)