We first consider only query languages.

**Relational Algebra:** Queries are expressions over operators and relation names.

**Relational Calculus:** Queries are special formulas of first-order logic with free variables.

**SQL:** Combination from algebra and calculus and additional constructs. Widely used DML for relational databases.

**QBE:** Graphical query language.

**Deductive Databases:** Queries are logical rules.

**Remark:**
- Relational Algebra and (safe) Relational Calculus have the same expressive power. For every expression of the algebra there is an equivalent expression in the calculus, and vice versa.
- A query language is called **relationally complete**, if it is (at least) as expressive as the relational algebra.
- These languages are compromises between efficiency and expressive power; they are not computationally complete (i.e., they cannot simulate a Turing Machine).
- They can be embedded into host languages (e.g. C++ or Java) or extended (PL/SQL), resulting in full computational completeness.
- Deductive Databases (Datalog) are more expressive than relational algebra and calculus.
### 3.1 Relational Algebra: Computations over Relations

**Operations on Tuples – Overview Slide**

Let $\mu \in \text{Tup}(\bar{X})$ where $\bar{X} = \{A_1, \ldots, A_k\}$.

(Formal definition of $\mu$ see Slide 61)

- For $\emptyset \subset \bar{Y} \subseteq \bar{X}$, the expression $\mu[\bar{Y}]$ denotes the **projection** of $\mu$ to $\bar{Y}$.
  
  Result: $\mu[\bar{Y}] \in \text{Tup}(\bar{Y})$ where $\mu[\bar{Y}](A) = \mu(A)$, $A \in \bar{Y}$.

- A **selection condition** $\alpha$ (wrt. $\bar{X}$) is an expression of the form $A \theta B$ or $A \theta c$, or $c \theta A$ where $A, B \in \bar{X}$, $\text{dom}(A) = \text{dom}(B)$, $c \in \text{dom}(A)$, and $\theta$ is a comparison operator on that domain like e.g. $\{=, \neq, \leq, <, \geq, >\}$.

  A tuple $\mu \in \text{Tup}(\bar{X})$ **satisfies** a selection condition $\alpha$, if – according to $\alpha$ – $\mu(A) \theta \mu(B)$ or $\mu(A) \theta c$, or $c \theta \mu(A)$ holds.

  These (atomic) selection conditions can be combined to formulas by using $\land, \lor, \neg$, and $(, )$.

- For $\bar{Y} = \{B_1, \ldots, B_k\}$, the expression $\mu[A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k]$ denotes the **renaming** of $\mu$.
  
  Result: $\mu[\ldots, A_i \rightarrow B_i, \ldots] \in \text{Tup}(\bar{Y})$ where $\mu[\ldots, A_i \rightarrow B_i, \ldots](B_i) = \mu(A_i)$ for $1 \leq i \leq k$.

---

Let $\mu \in \text{Tup}(\bar{X})$ where $\bar{X} = \{A_1, \ldots, A_k\}$.

**Projection (Reduction to a subset of the attributes)**

For $\emptyset \subset \bar{Y} \subseteq \bar{X}$, the expression $\mu[\bar{Y}]$ denotes the **projection** of $\mu$ to $\bar{Y}$.

Result: $\mu[\bar{Y}] \in \text{Tup}(\bar{Y})$ where $\mu[\bar{Y}](A) = \mu(A)$, $A \in \bar{Y}$.

**Example 3.1**

Consider the relation schema $R(\bar{X}) = \text{Continent}(\text{name}, \text{area})$: $\bar{X} = [\text{name}, \text{area}]$

and the tuple $\mu = \boxed{\text{name} \rightarrow \text{“Asia”}, \text{area} \rightarrow 4.50953e+07}$

formally: $\mu(\text{name}) = \text{“Asia”}$, $\mu(\text{area}) = 4.5E7$

**projection attributes:** Let $\bar{Y} = [\text{name}]$

**Result:** $\mu[\text{name}] = \boxed{\text{name} \rightarrow \text{“Asia”}}$
Again, $\mu \in \text{Tup}(\bar{X})$ where $\bar{X} = \{A_1, \ldots, A_k\}$.

Selection (only those tuples that satisfy some condition)

A selection condition $\alpha$ (wrt. $\bar{X}$) is an expression of the form $A \theta B$ or $A \theta c$, or $c \theta A$ where $A, B \in \bar{X}$, $\text{dom}(A) = \text{dom}(B)$, $c \in \text{dom}(A)$, and $\theta$ is a comparison operator on that domain like e.g. $\{=, \neq, \leq, \geq, <, >\}$.

A tuple $\mu \in \text{Tup}(\bar{X})$ satisfies a selection condition $\alpha$, if – according to $\alpha$ – $\mu(A) \theta \mu(B)$ or $\mu(A) \theta c$, or $c \theta \mu(A)$ holds.

Example 3.2
Consider again the relation schema $R(\bar{X}) = \text{continent}(\text{name}, \text{area})$: $\bar{X} = [\text{name}, \text{area}]$.

Selection condition: $\text{area} > 10000000$.

Consider again the tuple $\mu = \text{name} \rightarrow \text{“Asia”}, \text{area} \rightarrow 4.50953e+07$.

formally: $\mu(\text{name}) = \text{“Asia”}$, $\mu(\text{area}) = 4.5E7$

check: $\mu(\text{area}) > 10000000$

Result: yes.

These (atomic) selection conditions can be combined to formulas by using $\land$, $\lor$, $\neg$, and $(, )$.

76

Let $\mu \in \text{Tup}(\bar{X})$ where $\bar{X} = \{A_1, \ldots, A_k\}$.

Renaming (of attributes)

For $\bar{Y} = \{B_1, \ldots, B_k\}$, the expression $\mu[A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k]$ denotes the renaming of $\mu$.

Result: $\mu[A_1 \rightarrow B_1, \ldots, A_i \rightarrow B_i, \ldots] \in \text{Tup}(\bar{Y})$ where $\mu[A_1 \rightarrow B_1, \ldots, A_i \rightarrow B_i, \ldots](B_i) = \mu(A_i)$ for $1 \leq i \leq k$.

Example 3.3
Consider (for a tuple of the table $R(\bar{X}) = \text{encompasses}(\text{country}, \text{continent}, \text{percent})$):

$\bar{X} = [\text{country}, \text{continent}, \text{percent}]$.

Consider the tuple $\mu = \text{country} \rightarrow \text{“R”}, \text{continent} \rightarrow \text{“Asia”}, \text{percent} \rightarrow 80$.

formally: $\mu(\text{country}) = \text{“R”}$, $\mu(\text{continent}) = \text{“Asia”}$, $\mu(\text{percent}) = 80$

Renaming: $\bar{Y} = [\text{code}, \text{name}, \text{percent}]$

Result: a new tuple $\mu[\text{country} \rightarrow \text{code}, \text{continent} \rightarrow \text{name}, \text{percent} \rightarrow \text{percent}] = \text{code} \rightarrow \text{“R”}, \text{name} \rightarrow \text{“Asia”}, \text{percent} \rightarrow 80$ that now fits into the schema new_encompasses(code, name, percent).

The usefulness of renaming will become clear later ...
**Expressions in the Relational Algebra**

What is an algebra?
- An algebra consists of a "domain" (i.e., a set of "things"), and a set of operators.
- Operators map elements of the domain to other elements of the domain.
- Each of the operators has a "semantics", that is, a definition how the result of applying it to some input should look like.
- **Algebra expressions** are built over basic constants and operators (inductive definition).

Relational Algebra
- The "domain" consists of all relations (over arbitrary sets of attributes).
- The operators are then used for combining relations, and for describing computations - e.g., in SQL.
- **Relational algebra expressions** are defined inductively over relations and operators.
- Relational algebra expressions define queries against a relational database.

**Inductive Definition of Expressions**

Atomic Expressions
- For an arbitrary attribute $A$ and a constant $a \in \text{dom}(A)$, the **constant relation** $A \colon \{a\}$ is an algebra expression.
  - Format: $[A]$
  - Result relation: $\{a\}$

- Given a database schema $\mathbf{R} = \{R_1(X_1), \ldots, R_n(X_n)\}$, every relation name $R_i$ is an algebra expression.
  - Format of $R_i$: $X_i$
  - Result relation (wrt. a given database state $S$): the relation $S(R_i)$ that is currently stored in the database.
Structural Induction: Applying an Operator

• takes one or more input relations $in_1, in_2, \ldots$

• produces a result relation $out$:
  – $out$ has a format,
    depends on the formats of the input relations.
  – $out$ is a relation, i.e., it contains some tuples,
    depends on the content of the input relations.

### Base Operators

Let $\bar{X}, \bar{Y}$ formats and $r \in \text{Rel}(\bar{X})$ and $s \in \text{Rel}(\bar{Y})$ relations over $\bar{X}$ and $\bar{Y}$.

**Union**

Assume $r, s \in \text{Rel}(\bar{X})$.

Result format of $r \cup s$: $\bar{X}$

Result relation: $r \cup s = \{\mu \in \text{Tup}(\bar{X}) \mid \mu \in r \text{ or } \mu \in s\}$.

\[
\begin{array}{ccc}
A & B & C \\
r & a & b & c \\
d & a & f \\
c & b & d \\
\end{array}
\quad \begin{array}{ccc}
A & B & C \\
s & b & g & a \\
d & a & f \\
c & b & d \\
\end{array}
\quad \begin{array}{ccc}
A & B & C \\
r \cup s & a & b & c \\
d & a & f \\
c & b & d \\
b & g & a \\
\end{array}
\]
Set Difference

Assume \( r, s \in \text{Rel}(\bar{X}) \).
Result format of \( r \setminus s \): \( \bar{X} \)
Result relation: \( r \setminus s = \{ \mu \in r \mid \mu \notin s \} \).

\[
\begin{array}{ccc}
A & B & C \\
\hline
r = & a & b & c \\
d & a & f \\
c & b & d \\
\end{array}
\]
\[
\begin{array}{ccc}
A & B & C \\
\hline
s = & b & g & a \\
d & a & f \\
c & b & d \\
\end{array}
\]
\[
\begin{array}{ccc}
A & B & C \\
\hline
r \setminus s = & a & b & c \\
d & a & f \\
c & b & d \\
\end{array}
\]

Projection (Reduction to a subset of the attributes)

Assume \( r \in \text{Rel}(\bar{X}) \) and \( \bar{Y} \subseteq \bar{X} \).
Result format of \( \pi[\bar{Y}](r) \): \( \bar{Y} \)
Result relation: \( \pi[\bar{Y}](r) = \{ \mu[\bar{Y}] \mid \mu \in r \} \).

Example 3.4

<table>
<thead>
<tr>
<th>Continent</th>
<th>name</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>name</td>
<td>9562489.6</td>
</tr>
<tr>
<td>Africa</td>
<td>name</td>
<td>3.02547e+07</td>
</tr>
<tr>
<td>Asia</td>
<td>name</td>
<td>4.50953e+07</td>
</tr>
<tr>
<td>America</td>
<td>name</td>
<td>3.9872e+07</td>
</tr>
<tr>
<td>Australia</td>
<td>name</td>
<td>8503474.56</td>
</tr>
</tbody>
</table>

Let \( \bar{Y} = \{ \text{name} \} \)

\[
\begin{align*}
\mu_1[\text{name}] &= \text{name} \rightarrow \text{“Europe”} \\
\mu_2[\text{name}] &= \text{name} \rightarrow \text{“Africa”} \\
\mu_3[\text{name}] &= \text{name} \rightarrow \text{“Asia”} \\
\mu_4[\text{name}] &= \text{name} \rightarrow \text{“America”} \\
\mu_5[\text{name}] &= \text{name} \rightarrow \text{“Australia”}
\end{align*}
\]

\[
\begin{array}{ccc}
\pi[\text{name}](\text{Continents}) \\
\hline
\text{name} & \text{Europe} & \text{Africa} & \text{Asia} & \text{America} & \text{Australia} \\
\end{array}
\]

Selection (Reduction of number of tuples by a condition)

Assume \( r \in \text{Rel}(\bar{X}) \) and a selection condition \( \alpha \) over \( \bar{X} \).

Result format of \( \sigma[\alpha](r) \): \( \bar{X} \)
Result relation: \( \sigma[\alpha](r) = \{ \mu \in r \mid \mu \text{ satisfies } \alpha \} \).

Example 3.5

<table>
<thead>
<tr>
<th>Continent</th>
<th>name</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>9562489.6</td>
<td></td>
</tr>
<tr>
<td>Africa</td>
<td>3.02547e+07</td>
<td></td>
</tr>
<tr>
<td>Asia</td>
<td>4.50953e+07</td>
<td></td>
</tr>
<tr>
<td>America</td>
<td>3.9872e+07</td>
<td></td>
</tr>
<tr>
<td>Australia</td>
<td>8503474.56</td>
<td></td>
</tr>
</tbody>
</table>

Let \( \alpha = \text{“area} > 10000000\)”

| \( \sigma[\text{area} > 10E6](\text{Continent}) \) |
|---------------|----------------|
| name          | area           |
| Africa        | 3.02547e+07    |
| Asia          | 4.50953e+07    |
| America       | 3.9872e+07     |

Renaming (of attributes)

Assume \( r \in \text{Rel}(\bar{X}) \) with \( \bar{X} = [A_1, \ldots, A_k] \) and a renaming \( [A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k] \).

Result format of \( \rho[A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k](r) \): \( [B_1, \ldots, B_k] \)
Result relation: \( \rho[A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k](r) = \{ \mu[A_1 \rightarrow B_1, \ldots, A_k \rightarrow B_k] \mid \mu \in r \} \).

Example 3.6

Consider the renaming of the table \text{encompasses}(\text{country}, \text{continent}, \text{percent}) :

\( \bar{X} = [\text{country}, \text{continent}, \text{percent}] \)

Renaming: \( \bar{Y} = [\text{code}, \text{name}, \text{percent}] \)

| \( \rho[\text{country} \rightarrow \text{code}, \text{continent} \rightarrow \text{name}, \text{percent} \rightarrow \text{percent}](\text{encompasses}) \) |
|-----------------|-------|--------|
| code            | name  | percent |
| R               | Europe| 20     |
| R               | Asia  | 80     |
| D               | Europe| 100    |
| ...             | ...   | ...    |
(Natural) Join (Combining two relations via common attributes)

Assume $r \in \operatorname{Rel}(\bar{X})$ and $s \in \operatorname{Rel}(\bar{Y})$ for arbitrary $\bar{X}, \bar{Y}$.

Convention: For $\bar{X} \cup \bar{Y}$, as a shorthand, write $\bar{X} \bar{Y}$.

for two tuples $\mu_1 = \begin{pmatrix} v_1, \ldots, v_n \end{pmatrix}$ and $\mu_2 = \begin{pmatrix} w_1, \ldots, w_m \end{pmatrix}$, $\mu_1 \mu_2 := \begin{pmatrix} v_1, \ldots, v_n, w_1, \ldots, w_m \end{pmatrix}$

Result format of $r \bowtie s$: $\bar{X}\bar{Y}$.
Result relation: $r \bowtie s = \{ \mu \in \operatorname{Tup}(\bar{X}\bar{Y}) \mid \mu[\bar{X}] \in r$ and $\mu[\bar{Y}] \in s \}$.

Motivation

Simplest Case: $\bar{X} \cap \bar{Y} = \emptyset \Rightarrow \text{Cartesian Product } r \bowtie s = r \times s$
$r \times s = \{ \mu_1 \mu_2 \in \operatorname{Tup}(\bar{X}\bar{Y}) \mid \mu_1 \in r$ and $\mu_2 \in s \}$.

<table>
<thead>
<tr>
<th>$r$ \hspace{1cm}</th>
<th>$s$ \hspace{1cm}</th>
<th>$r \bowtie s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$B$</td>
<td>$C$</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>c</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>e</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>c</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>e</td>
</tr>
</tbody>
</table>

Example 3.7 (Cartesian Product of Continent and Encompasses)

The cartesian product combines everything with everything, not only “meaningful” combinations:

<table>
<thead>
<tr>
<th>Continent \times encompasses</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Africa</td>
</tr>
<tr>
<td>Africa</td>
</tr>
<tr>
<td>Africa</td>
</tr>
<tr>
<td>Africa</td>
</tr>
<tr>
<td>Asia</td>
</tr>
<tr>
<td>Asia</td>
</tr>
<tr>
<td>Asia</td>
</tr>
<tr>
<td>Asia</td>
</tr>
<tr>
<td>:</td>
</tr>
</tbody>
</table>
Back to the Natural Join

General case $\bar{X} \cap \bar{Y} \neq \emptyset$: shared attribute names constrain the result relation.

Again the definition: $r \bowtie s = \{ \mu \in \text{Tup}(XY) | \mu[\bar{X}] \in r \text{ and } \mu[\bar{Y}] \in s \}$. 
(Note: this implies that the tuples $\mu_1 := \mu[\bar{X}] \in r$ and $\mu_2 := \mu[\bar{Y}] \in s$ coincide in the shared attributes $\bar{X} \cap \bar{Y}$)

Example 3.8
Consider $\text{encompasses}(\text{country, continent, percent})$ and $\text{isMember}(\text{organization, country, type})$:

<table>
<thead>
<tr>
<th>encompasses</th>
<th>isMember</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>continent</td>
</tr>
<tr>
<td>$R$</td>
<td>Europe</td>
</tr>
<tr>
<td>$R$</td>
<td>Asia</td>
</tr>
<tr>
<td>$D$</td>
<td>Europe</td>
</tr>
</tbody>
</table>

$\text{encompasses} \bowtie \text{isMember} = \{ \mu \in \text{Tup}(\text{country, continent, percent, org, type}) | \mu[\text{country, continent, percent}] \in \text{encompasses} \text{ and } \mu[\text{org, country, type}] \in \text{isMember} \}$

Example 3.8 (Continued)

$\text{encompasses} \bowtie \text{isMember} = \{ \mu \in \text{Tup}(\text{country, continent, percent, org, type}) | \mu[\text{country, continent, percent}] \in \text{encompasses} \text{ and } \mu[\text{org, country, type}] \in \text{isMember} \}$

start with $(R, \text{Europe}, 20) \in \text{encompasses}$.
check which tuples in $\text{isMember}$ match:

$(UN, R, \text{member}) \in \text{isMember}$ matches:
result: $(R, \text{Europe}, 20, UN, \text{member})$ belongs to the result.
(some more matches ...)

continue with $(R, \text{Asia}, 80) \in \text{encompasses}$.
$(UN, R, \text{member}) \in \text{isMember}$ matches:
result: $(R, \text{Asia}, 80, UN, \text{member})$ belongs to the result.
(some more matches ...)

continue with $(D, \text{Europe}, 100) \in \text{encompasses}$.
$(EU, D, \text{member}) \in \text{isMember}$ matches:
result: $(D, \text{Europe}, 100, EU, \text{member})$ belongs to the result.
$(UN, D, \text{member}) \in \text{isMember}$ matches:
result: $(D, \text{Europe}, 100, UN, \text{member})$ belongs to the result.
(some more matches ...)
Example 3.8 (Continued)
Result:

<table>
<thead>
<tr>
<th>encompasses ⊳ isMember</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

Example 3.9 (and Exercise)
Consider the expression

\[ \text{Continent } \bowtie \rho \{ \text{country } \rightarrow \text{code}, \text{continent } \rightarrow \text{name}, \text{percent } \rightarrow \text{percent}\} (\text{encompasses}) \]

Functionalities of the Join
- Combining relations
- Selective functionality: only matching tuples survive
  (consider joining cities and organizations on headquarters)

**Derived Operators**

**Intersection**
Assume \( r, s \in \text{Rel}(\bar{X}) \).
Then, \( r \cap s = \{ \mu \in \text{Tup}(\bar{X}) \mid \mu \in r \text{ and } \mu \in s \} \).

**Theorem 3.1**
Intersection can be expressed by difference: \( r \cap s = r \setminus (r \setminus s) \).
Combination of Cartesian Product and Selection:
Assume $r \in \text{Rel}(\bar{X})$, and $s \in \text{Rel}(\bar{Y})$, such that $\bar{X} \cap \bar{Y} = \emptyset$, and $A \theta B$ a selection condition.

$$r \bowtie_{A \theta B} s = \{ \mu \in \text{Tup}(\bar{XY}) \mid \mu[\bar{X}] \in r, \mu[\bar{Y}] \in s \text{ and } \mu \text{ satisfies } A \theta B \} = \sigma[A \theta B](r \times s).$$

Equi-Join

$\theta$-join that uses the “=”-predicate.

Example 3.10 (and Exercise)
Consider again Example 3.7:

$\text{Continent} \bowtie \text{encompasses} = \text{Continent} \times \text{encompasses} \text{ contained tuples that did not really make sense.}$

$\text{Continent} \bowtie_{\text{continent} = \text{name}} \text{encompasses} \text{ would be more useful.}$

Furthermore, consider

$\pi[\text{continent}, \text{area}, \text{code}, \text{percent}](\text{Continent} \bowtie_{\text{continent} = \text{name}} \text{encompasses})$:

- removes the - now redundant - “name” column,
- is equivalent to the natural join $(\rho[\text{name} \rightarrow \text{continent}](\text{continent})) \bowtie \text{encompasses}. \square$

Semi-Join

- recall: joins combine, but are also selective
- semi-join acts like a selection on a relation $r$:
  - selection condition not given as a boolean formula on the attributes of $r$, but by “looking into” another relation (a subquery)

Assume $r \in \text{Rel}(\bar{X})$ and $s \in \text{Rel}(\bar{Y})$ such that $\bar{X} \cap \bar{Y} \neq \emptyset$.

Result format of $r \bowtie s$: $\bar{X}$

Result relation: $r \bowtie s = \pi[\bar{X}](r \bowtie s)$

The semi-join $r \bowtie s$ does not return the join, but checks which tuples of $r$ “survive” the join with $s$ (i.e., “which find a counterpart in $s$ wrt. the shared attributes”):

- Used with subqueries: (main query) $\bowtie$ (subquery)
- $r \bowtie s \subseteq r$
- Used for optimizing the evaluation of joins (often in combination with indexes).
Semi-Join: Example

Give the names of all countries where a city with at least 1000000 inhabitants is located:

\[
\pi \text{name} \\
\Join \text{Country.code=City.country} \\
\sigma \text{[population}>1000000] \\
\text{City}
\]

- Have a short look “inside” the subquery, but don’t actually use it:
- look only if there is a big city in this country.
- “if the country code is in the set of country codes ...”:

\[
\pi \text{name} \\
\Join \text{Country.code=City.country} \\
\text{Country} \quad \pi \text{[country]} \quad \text{and put an index on the result set} \\
\sigma \text{[population}>1000000] \\
\text{City}
\]

Outer Join

- The join is the operator for combining relations

Example 3.11

- Persons work in divisions of a company, tools are assigned to the divisions:

<table>
<thead>
<tr>
<th>Works</th>
<th>Tools</th>
<th>Works ⊲ Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Division</td>
<td>Division</td>
</tr>
<tr>
<td>John</td>
<td>Production</td>
<td>Production</td>
</tr>
<tr>
<td>Bill</td>
<td>Production</td>
<td>Research</td>
</tr>
<tr>
<td>John</td>
<td>Research</td>
<td>Research</td>
</tr>
<tr>
<td>Mary</td>
<td>Research</td>
<td>Admin.</td>
</tr>
<tr>
<td>Sue</td>
<td>Sales</td>
<td></td>
</tr>
</tbody>
</table>

- join contains no tuple that describes Sue,
- join contains no tuple that describes the administration or sales division,
- join contains no tuple that shows that there is a typewriter.
Outer Join
Assume \( r \in \text{Rel}(\overline{X}) \) and \( s \in \text{Rel}(\overline{Y}) \).

Result format of \( r \bowtie \Delta s \): \( \overline{XY} \)
The outer join extends the “inner” join with all tuples that have no counterpart in the other relation (filled with null values):

**Example 3.12 (Outer Join)**
Consider again Example 3.11

<table>
<thead>
<tr>
<th>Works ( \bowtie ) Tools</th>
<th>Works ( \bowtie ) Tools</th>
<th>Works ( \bowtie ) Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Division</td>
<td>Tool</td>
</tr>
<tr>
<td>John Production hammer</td>
<td>John Production hammer</td>
<td>John Research pen</td>
</tr>
<tr>
<td>Bill Production hammer</td>
<td>John Research computer</td>
<td>Mary Research pen</td>
</tr>
<tr>
<td>John Research computer</td>
<td>Mary Research computer</td>
<td>Sue Sales NULL</td>
</tr>
<tr>
<td>NULL Admin typewriter</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Formally, the result relation \( r \bowtie \Delta s \) is defined as follows:

\[
J = r \bowtie s \quad \text{— take the (“inner”) join as base}
\]

\[
r_0 = r \setminus \pi[\overline{X}](J) = r \setminus (r \bowtie s) \quad \text{— } r\text{-tuples that “are missing”}
\]

\[
s_0 = s \setminus \pi[\overline{Y}](J) = s \setminus (r \bowtie s) \quad \text{— } s\text{-tuples that “are missing”}
\]

\[
\overline{Y}_0 = \overline{Y} \setminus \overline{X}, \overline{X}_0 = \overline{X} \setminus \overline{Y}
\]

Let \( \mu_s \in \text{Tup}(\overline{Y}_0), \mu_r \in \text{Tup}(\overline{X}_0) \) such that \( \mu_s, \mu_r \) consist only of null values

\[
r \bowtie \Delta s = J \cup (r_0 \times \{\mu_s\}) \cup (s_0 \times \{\mu_r\}).
\]

**Example 3.12 (Continued)**
For the above example,

\[
J = \text{Works} \bowtie \text{Tools}
\]

\[
r_0 = [\text{“Sue”}, \text{“Sales”}], \ s_0 = [\text{“Admin”}, \text{“Typewriter”}]
\]

\[
\overline{Y}_0 = \text{Tool}, \overline{X}_0 = \text{Person}
\]

\[
\mu_s = \begin{bmatrix} \text{Tool} \\ \text{null} \end{bmatrix}, \ \mu_r = \begin{bmatrix} \text{Person} \\ \text{null} \end{bmatrix}
\]

\[
r_0 \times \{\mu_s\} = \begin{bmatrix} \text{Person} & \text{Division} & \text{Tool} \\ \text{Sue} & \text{Sales} & \text{null} \end{bmatrix}, \ s_0 \times \{\mu_r\} = \begin{bmatrix} \text{Person} & \text{Division} & \text{Tool} \\ \text{null} & \text{Admin} & \text{Typewriter} \end{bmatrix}
\]
Left and Right Outer Join

Analogously to the (full) outer join:

\[ r \bowtie s = J \cup (r_0 \times \mu_s) \]

\[ r \bowtie s = J \cup (s_0 \times \mu_r) \]

Generalized Natural Join

Assume \( r_i \subseteq \text{Tup}(\bar{X}_i) \).

Result format: \( \bigcup_{i=1}^n \bar{X}_i \)

Result relation: \( \bowtie_{i=1}^n r_i = \{ \mu \in \text{Tup}(\bigcup_{i=1}^n \bar{X}_i) \mid \mu[\bar{X}_i] \in r_i \} \)

Exercise 3.1

Prove that the Generalized Natural Join is well-defined, i.e., that the order how to join the \( r_i \) does not matter.

Proceed as follows:

• Show that the natural join is commutative,

• Show that the natural join is associative,

• ... then complete the proof.

Relational Division

Assume \( r \in \text{Rel}(\bar{X}) \) and \( s \in \text{Rel}(\bar{Y}) \) such that \( \bar{Y} \subseteq \bar{X} \).

Result format of \( r \div s \): \( \bar{Z} = \bar{X} \setminus \bar{Y} \).

The result relation \( r \div s \) is specified as “all \( \bar{Z} \)-values that occur in \( \pi[\bar{Z}](r) \), with the additional condition that they occur in \( r \) together with each of the \( \bar{Y} \) values that occur in \( s \)”.

Formally,

\[ r \div s = \{ \mu \in \text{Tup}(\bar{Z}) \mid \{ \mu \} \times s \subseteq r \} = \pi[\bar{Z}](r) \setminus \pi[\bar{Z}](\pi[\bar{Z}](r) \times s) \setminus r \).

\[ \text{this implies that } \mu \in \pi[\bar{Z}](r) \]

• Simple observation: \( \pi[\bar{Z}](r) \supseteq r \div s \).

This constrains the set of possible results.

• Often, \( \bar{Z} \) and \( \bar{Y} \) correspond to the keys of relations that represent the instances of entity types.
Example 3.13 (Relational Division)

Compute those organizations that have at least one member on each continent:

First step: which organizations have (some) member on which continents:

\[
\pi \text{[organization,continent]}
\]

\[
\text{ismember} \triangleright \text{encompasses}
\]

\[
\begin{align*}
\text{SELECT DISTINCT } & \text{i.organization, e.continent} \\
\text{FROM ismember i, encompasses e} \\
\text{WHERE i.country= e.country} \\
\text{ORDER by 1}
\end{align*}
\]

<table>
<thead>
<tr>
<th>orgOnCont</th>
</tr>
</thead>
<tbody>
<tr>
<td>organization</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>NATO</td>
</tr>
<tr>
<td>NATO</td>
</tr>
<tr>
<td>NATO</td>
</tr>
</tbody>
</table>

Example 3.13 (Cont’d)

\[
\frac{\pi \text{[name]}}{\text{continent}}
\]

\[
\rho \text{[name→continent]}
\]

\[
\begin{align*}
\rho & \text{[name→continent]} \\
\pi & \text{[name]} \\
\text{continent}
\end{align*}
\]

<table>
<thead>
<tr>
<th>orgOnCont</th>
</tr>
</thead>
<tbody>
<tr>
<td>organization</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>UN</td>
</tr>
<tr>
<td>NATO</td>
</tr>
<tr>
<td>NATO</td>
</tr>
<tr>
<td>NATO</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\rho & \text{[name→continent]} \\
\pi & \text{[name]}(\text{continent})
\end{align*}
\]

<table>
<thead>
<tr>
<th>continent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
</tr>
<tr>
<td>Europe</td>
</tr>
<tr>
<td>Australia</td>
</tr>
<tr>
<td>America</td>
</tr>
<tr>
<td>Africa</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\tilde{X} & = \text{[organization, continent]} \\
\tilde{Y} & = \text{[continent]}
\end{align*}
\]

Thus, \( \tilde{Z} = \text{[organization]} \).

- **UN**: occurs with each continent in orgOnCont
  \( \Rightarrow \) belongs to the result.

- **NATO**: does not occur with each continent in orgOnCont
  \( \Rightarrow \) does not belong to the result.
Example 3.13 (Cont'd)
Consider again the formal algebraic characterization of Division:

\[
    r \div s = \{ \mu \in \text{Tup}(\bar{Z}) \mid \{\mu\} \times s \subseteq r \} = \pi[\bar{Z}](r) \setminus \pi[\bar{Z}](\pi[\bar{Z}](r) \times s \setminus r).
\]

1. \( r = \text{orgOnCont}, s = \pi[\text{name}](\text{continent}), Z = \text{Country}. \)

2. \((\pi[\bar{Z}](r) \times s)\) contains all tuples of organizations with each of the continents, e.g.,
   \((\text{NATO}, \text{Europe}), (\text{NATO}, \text{Asia}), (\text{NATO}, \text{America}), (\text{NATO}, \text{Africa}), (\text{NATO}, \text{Australia})\).

3. \(((\pi[\bar{Z}](r) \times s) \setminus r)\) contains all such tuples which are not “valid”, e.g., \((\text{NATO}, \text{Africa})\).

4. projecting this to the organizations yields all those organizations where a non-valid tuple
   has been generated in (2), i.e., that have no member on some continent (e.g., NATO).

5. \(\pi[\bar{Z}](r)\) is the list of all organizations ...

6. ... subtracting those computed in (4) yields those that have a member on each continent. □

Expressions

• inductively defined: combining expressions by operators

Example 3.14
The names of all cities where (i) headquarters of an organization are located, and (ii) that are
 capitals of a member country of this organization.
As a tree:

\[
\begin{align*}
\pi[\text{City}] & \cap \\
\pi[\text{abbrev}, \text{city}, \text{prov}, \text{country}] & \quad \rho[\text{capital} \rightarrow \text{city}] \\
\text{Organization} & \quad \rho[\text{organization} \rightarrow \text{abbrev}] \\
\pi[\text{abbrev}, \text{capital}, \text{prov}, \text{country}] & \quad \rho[\text{code} \rightarrow \text{country}] \\
\end{align*}
\]

isMember  \\
Country

Note that there are many equivalent expressions.
Expressions in the Relational Algebra as Queries

Let $R = \{ R_1, \ldots, R_k \}$ a set of relation schemata of the form $R_i(\bar{X}_i)$. As already described, an database state to $R$ is a structure $S$ that maps every relation name $R_i$ in $R$ to a relation $S(R_i) \subseteq \text{Tup}(\bar{X}_i)$

Every algebra expression $Q$ defines a query against the state $S$ of the database:

• For given $R$, $Q$ is assigned a format $\Sigma_Q$ (the format of the answer).

• For every database state $S$, $S(Q) \subseteq \text{Tup}(\Sigma_Q)$ is a relation over $\Sigma_Q$, called the answer set for $Q$ wrt. $S$.

• $S(Q)$ can be computed according to the inductive definition, starting with the innermost (atomic) subexpressions.

• Thus, the relational algebra has a functional semantics.

Summary: Inductive Definition of Expressions

Atomic Expressions

• For an arbitrary attribute $A$ and a constant $a \in \text{dom}(A)$, the constant relation $A : \{a\}$ is an algebra expression.

  $\Sigma_{A:\{a\}} = [A]$ and $S(A : \{a\}) = A : \{a\}$

• Every relation name $R$ is an algebra expression.

  $\Sigma_R = \bar{X}$ and $S(R) = S(R)$. 
Compound Expressions
Assume algebra expressions $Q_1, Q_2$ that define $\Sigma_{Q_1}, \Sigma_{Q_2}, S(Q_1),$ and $S(Q_2)$.

Compound algebraic expressions are now formed by the following rules (corresponding to the algebra operators):

**Union**
If $\Sigma_{Q_1} = \Sigma_{Q_2}$, then $Q = (Q_1 \cup Q_2)$ is the union of $Q_1$ and $Q_2$.
$\Sigma_Q = \Sigma_{Q_1}$ and $S(Q) = S(Q_1) \cup S(Q_2)$.

**Difference**
If $\Sigma_{Q_1} = \Sigma_{Q_2}$, then $Q = (Q_1 \setminus Q_2)$ is the difference of $Q_1$ and $Q_2$.
$\Sigma_Q = \Sigma_{Q_1}$ and $S(Q) = S(Q_1) \setminus S(Q_2)$.

**Projection**
For $\emptyset \neq \bar{Y} \subseteq \Sigma_{Q_1}$, $Q = \pi[\bar{Y}](Q_1)$ is the projection of $Q_1$ to the attributes in $\bar{Y}$.
$\Sigma_Q = \bar{Y}$ and $S(Q) = \pi[\bar{Y}](S(Q_1))$.

Selection
For a selection condition $\alpha$ over $\Sigma_{Q_1}$, $Q = \sigma[\alpha]Q_1$ is the selection from $Q_1$ wrt. $\alpha$.
$\Sigma_Q = \Sigma_{Q_1}$ and $S(Q) = \sigma[\alpha](S(Q_1))$.

Natural Join
$Q = (Q_1 \bowtie Q_2)$ is the (natural) join of $Q_1$ and $Q_2$.
$\Sigma_Q = \Sigma_{Q_1} \cup \Sigma_{Q_2}$ and $S(Q) = S(Q_1) \bowtie S(Q_2)$.

Renaming
For $\Sigma_{Q_1} = \{A_1, \ldots, A_k\}$ and $\{B_1, \ldots, B_k\}$ a set of attributes, $\rho[A_1 \to B_1, \ldots, A_k \to B_k]Q_1$ is the renaming of $Q_1$.
$\Sigma_Q = \{B_1, \ldots, B_k\}$ and $S(Q) = \{\mu[A_1 \to B_1, \ldots, A_k \to B_k] \mid \mu \in S(Q_1)\}$.
Example 3.15
Professor(PNr, Name, Office), Course(CNr, Credits, CName) teach(PNr, CNr), examine(PNr, CNr)

• For each professor (name) determine the courses he gives (CName).
  \[ \pi \text{ [Name, CName]} ((Professor \Join teach) \Join Course) \]

• For each professor (name) determine the courses (CName) that he teaches, but that he does not examine.
  \[ \pi \text{ [Name, CName]}((\pi \text{ [Name, CNr]}(\text{Professor} \Join teach)) \setminus (\pi \text{ [Name, CNr]}(\text{Professor} \Join examine))) \Join Course) \]

Simpler expression:
\[ \pi \text{ [Name, CName]} ((\text{Professor} \Join (\text{teach} \setminus \text{examine})) \Join Course) \]

Equivalence of Expressions

Algebra expressions \( Q, Q' \) are called equivalent, \( Q \equiv Q' \), if and only if for all structures \( S \), \( S(Q) = S(Q') \).

Equivalence of expressions is the basis for algebraic optimization.

Let \( \text{attr}(\alpha) \) the set of attributes that occur in a selection condition \( \alpha \), and \( Q, Q_1, Q_2, \ldots \) expressions with formats \( X, X_1, \ldots \).

Projections

- \( Z, \bar{Y} \subseteq \bar{X} \Rightarrow \pi[Z](\pi[\bar{Y}](Q)) \equiv \pi[Z \cap \bar{Y}](Q) \).
- \( \bar{Z} \subseteq \bar{Y} \subseteq \bar{X} \Rightarrow \pi[\bar{Z}](\pi[\bar{Y}](Q)) \equiv \pi[\bar{Z}](Q) \).

Selections

- \( \sigma[\alpha_1](\sigma[\alpha_2](Q)) \equiv \sigma[\alpha_2](\sigma[\alpha_1](Q)) \equiv \sigma[\alpha_1 \land \alpha_2](Q) \).
- \( \text{attr}(\alpha) \subseteq \bar{Y} \subseteq \bar{X} \Rightarrow \pi[\bar{Y}](\sigma[\alpha](Q)) \equiv \sigma[\alpha](\pi[\bar{Y}](Q)) \).

Joins

- \( Q_1 \Join Q_2 \equiv Q_2 \Join Q_1 \).
- \( (Q_1 \Join Q_2) \Join Q_3 \equiv Q_1 \Join (Q_2 \Join Q_3) \).
Joins and other Operations

- \( \text{attr}(\alpha) \subseteq \bar{X}_1 \cap \bar{X}_2 \Rightarrow \sigma[\alpha](Q_1 \bowtie Q_2) \equiv \sigma[\alpha](Q_1) \bowtie \sigma[\alpha](Q_2). \)

- \( \text{attr}(\alpha) \subseteq \bar{X}_1, \text{attr}(\alpha) \cap \bar{X}_2 = \emptyset \Rightarrow \sigma[\alpha](Q_1 \bowtie Q_2) \equiv \sigma[\alpha](Q_1) \bowtie Q_2. \)

- Assume \( \bar{V} \subseteq \overline{X_1X_2} \) and let \( \bar{W} = \bar{X}_1 \cap \overline{VX_2}, \bar{U} = \bar{X}_2 \cap \overline{VX_1}. \) Then, \( \pi[\bar{V}](Q_1 \bowtie Q_2) \equiv \pi[\bar{W}](\pi[\bar{W}](Q_1) \bowtie \pi[\bar{U}](Q_2)); \)

- \( \bar{X}_2 = \bar{X}_3 \Rightarrow Q_1 \bowtie (Q_2 \text{ op } Q_3) \equiv (Q_1 \bowtie Q_2) \text{ op } (Q_1 \bowtie Q_3) \text{ where op } \in \{\cup, \}\).

Exercise 3.2

Prove some of the equalities (use the definitions given on the “Base Operators” slide).

---

Transitive Closure

The transitive closure of a binary relation \( R \), denoted by \( R^* \) is defined as follows:

\[
\begin{align*}
R^1 &= R \\
R^{n+1} &= \{(a, b) | \text{ there is an } s \text{ s.t. } (a, x) \in R^n \text{ and } (x, b) \in R\} \\
R^* &= \bigcup_{1 \leq n} R^n
\end{align*}
\]

Examples:

- \( \text{child}(x,y) \): \( \text{child}^* = \text{descendant} \)
- flight connections
- \( \text{flows_into} \) of rivers in MONDIAL

Theorem 3.2

There is no expression of the relational algebra that computes the transitive closure of arbitrary binary relations \( r \).
Time to play. Perhaps postpone examples after comparison with SQL (next subsections)

Aspects

• join as “extending” operation (cartesian product – “all pairs of X and Y such that ...”)
• equijoin as “restricting” operation
• natural join/equijoin in many cases along key/foreign key relationships
• relational division (in case of queries of the style “return all X that are in a given relation with all Y such that ...”)