TODAY

• Relational Algebra
  – Group-By Aggregate

• Relational Calculus

• Examples
GROUP BY AGGREGATE

- is part of the so-called extended RA
- helps us to compute counts, sums, min, max, ...

Examples
- What is the average age of the customers?
- How many people bought an iPad?
GROUP BY AGGREGATE

Notation: $\gamma_{X,Agg}(Y)(R)$

- **group by** the attributes in $X$
- **aggregate** the attribute in $Y$
  - SUM, COUNT, AVG (average), MIN, MAX

- Output schema: $X +$ an extra (numerical) attribute
### Example

**Person**

<table>
<thead>
<tr>
<th>SSN</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>934729837</td>
<td>Paris</td>
<td>24</td>
</tr>
<tr>
<td>123123645</td>
<td>John</td>
<td>30</td>
</tr>
<tr>
<td>384475687</td>
<td>Arun</td>
<td>21</td>
</tr>
</tbody>
</table>

$\gamma_{AVG(age)}(Person)$

<table>
<thead>
<tr>
<th>AVG(age)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>
## Example

### Person

<table>
<thead>
<tr>
<th>SSN</th>
<th>name</th>
<th>age</th>
<th>phoneNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>934729837</td>
<td>Paris</td>
<td>24</td>
<td>608-374-8422</td>
</tr>
<tr>
<td>934729837</td>
<td>Paris</td>
<td>24</td>
<td>603-534-8399</td>
</tr>
<tr>
<td>123123645</td>
<td>John</td>
<td>30</td>
<td>608-321-1163</td>
</tr>
<tr>
<td>384475687</td>
<td>Arun</td>
<td>21</td>
<td>206-473-8221</td>
</tr>
</tbody>
</table>

\[ \gamma_{SSN, \text{COUNT}(\text{phoneNumber})}(\text{Person}) \]

<table>
<thead>
<tr>
<th>SSN</th>
<th>\text{COUNT}(\text{phoneNumber})</th>
</tr>
</thead>
<tbody>
<tr>
<td>934729837</td>
<td>2</td>
</tr>
<tr>
<td>123123645</td>
<td>1</td>
</tr>
<tr>
<td>384475687</td>
<td>1</td>
</tr>
</tbody>
</table>
RELATIONAL CALCULUS
RELATIONAL QUERY LANGUAGES

• **Declarative**: Tuple Relational Calculus (TRC)  
  Domain Relational Calculus (DRC)  
  – describe what a user wants, rather than how to compute it

• **Procedural**: Relational Algebra (RA)  
  – operational, useful for representing execution plans
**Relational Calculus**

- declarative query language
- simple subset of First-Order Logic
  - TRC: variables range over *tuples*
  - DRC: variables range over *domain elements*

**Example**

- RA: \( \sigma_{\text{age}>24}(\text{Person}) \)
- TRC: \( \{ t \mid t \in \text{Person} \land t.\text{age} > 24 \} \)
**Tuple Relational Calculus**

A query in TRC has the form: \( \{ t \mid \phi(t) \} \)

- The query returns all tuples \( t \) for which the formula \( \phi(t) \) evaluates to **true**
- A formula is **recursively** defined, starting with simple *atomic formulas*, and building more complex formulas using the logical operators
• an atomic formula can be:
  – $t \in R$
  – $s.A \ op \ t.B$
  – $s.A \ op \ constant$
• a formula is recursively defined from formulas $p$, $q$:
  – $\neg p$, $p \land q$, $p \lor q$, $p \implies q$
  – $\exists t\ (p(t))$
  – $\forall t\ (p(t))$
TRC Example

Sailors (sid, sname, rating, age)
Reserves (sid, bid, day)
Boats (bid, bname, color)

• $\sigma_{\text{rating} > 1 \land \text{age} > 30}(\text{Sailors})$
• $\{ t \mid t \in \text{Sailors} \land t.\text{rating} > 1 \land t.\text{age} > 30 \}$

atomic formulas
**Free and Bound Variables**

Let the TRC query \{t \mid \phi(t)\}

- The use of quantifiers \(\forall t, \exists t\) in a formula binds the variable \(t\)
  - a variable that is not bound is free
- The variable \(t\) that appears to the left of | must be the only free variable in the formula \(\phi(t)\)
TRC Example

Sailors (sid, sname, rating, age)
Reserves (sid, bid, day)
Boats (bid, bname, color)

• $\pi_{sname}(\sigma_{rating>1}(\text{Sailors}))$
• $\{t \mid \exists s \in \text{Sailors} \ (s\.rating > 1 \land t\.sname = s\.sname)\}$

Convention: the attributes of the free variable $t$ are exactly the ones mentioned in the formula!
Codd’s Theorem

• **Codd’s Theorem**: Every RA query can be expressed as a *safe* query in TRC/DRC; the converse is also true

• A query language is *relationally complete* if it can express every query that is expressible in RA/RC
**More Examples**

- **Sailors** \((\text{sid, sname, rating, age})\)
- **Reserves** \((\text{sid, bid, day})\)
- **Boats** \((\text{bid, bname, color})\)

**Q1**: What are the names of the sailors who have reserved boat #100?

\[
\{ t \mid \exists s \in \text{Sailors}, \exists r \in \text{Reserves} (s.\text{sid} = r.\text{sid} \land r.\text{bid} = 100 \land t.\text{sname} = s.\text{sname}) \}
\]
More Examples

Sailors (sid, sname, rating, age)
Reserves (sid, bid, day)
Boats (bid, bname, color)

Q2: Find the names of the sailors who have reserved at least two different boats

Q3: Find the names of the sailors who have reserved all ‘470’ boats
MORE EXAMPLES

Product (pid, name, price, category, maker-cid)
Purchase (buyer-ssn, seller-ssn, store, pid)
Company (cid, name, country)
Person (ssn, name, phone, city)

Q4: Find the phone numbers of people who bought iPads from Fred (the salesman)

Q5: Find the names of people who bought products from the USA
More Examples

Product (pid, name, price, category, maker-cid)
Purchase (buyer-ssn, seller-ssn, store, pid)
Company (cid, name, country)
Person (ssn, name, phone, city)

Q6: Find the names of people who bought products from the USA, but not from Greece

Q7: Find the names of people who bought products from the USA, and live in Madison
**More Examples (Aggregate)**

- **Product** (pid, name, price, category, maker-cid)
- **Purchase** (buyer-ssn, seller-ssn, store, pid)
- **Company** (cid, name, country)
- **Person** (ssn, name, phone, city)

**Q8:** Find the total value of products sold by Fred (the salesman)

**Q9:** Find the average price of computers made in Greece
More Examples: Beers

Likes (drinker, beer)
Frequents (drinker, bar)
Serves (bar, beer)

• Find the bars that serve all beers that Fred likes
More Examples: Beers

Likes (drinker, beer)
Frequents (drinker, bar)
Serves (bar, beer)

- **Average drinker**: frequents some bar that serves some beer they like
- **Prudent drinker**: frequents only bars that serve some beer they like
- **Paranoid drinker**: frequents only bars that serve only beers they like