Lois Delcambre
Neena Maldikar
Suki Kangas

cs.pdx.edu/~lmd/cs386
Relational DBs
1970 Edgar Codd
mathematical defin of a relation
and a query language
relational model
**Schema for the relation**

**Student** (id, name, age, major)

- **Attribute names**: id, name, age, major

One row
One tuple

{ (101, 'David', 21, 'CS')
(102, 'Mary', 20, 'EE')
(103, 'David', 30, 'CS') }

**Domains for the attributes**
- **id**: integer
- **name**: character varying (40)
- **age**: 3 digit integer
- **major**: character * 15
\( 3 \) \( 1 \) \( 0 \) \( 1 \)
\( 1 \) \( 0 \) \( 2 \)
\( 0 \) \( 3 \), \( \ldots \), \( 3 \)

id-domain \( \times \) name-domain \( \times \) age-domain \( \times \) major-domain

current instance of a relation - the Student relation - is a subset of the cross product of the domains.
relational algebra

select age <= 20 from Student

query

note: relations are sets
\( \text{Student}(id, \text{name}, \text{age}, \text{major}) \)

\[
\begin{align*}
\text{project} \quad \Pi_{\text{major}, \text{name}} \quad \text{Student} \\
\text{query} \quad \Phi \quad (\text{`cs', `David'}) \\
\quad \quad (\text{`EE', `Mary'})
\end{align*}
\]
Cross product \( \times \)

\[
\text{Student} \times \text{Student}
\]

\[
(\text{Mid Student}) \times (\text{Mid Student})
\]

<table>
<thead>
<tr>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>103</td>
</tr>
</tbody>
</table>
\[ \wedge \cup \] union intersection set difference

For the query to be well-formed, the two relations must have the same number of columns and corresponding attributes must be defined on the same domain. Union-compatible.
employee(emp_id, name, gender, job)

position(id, title)
7.a. $\text{Employee}$

\[ \pi \text{ employee}
\text{emp.id, name, gender, job} \]

\[ \sigma \text{ employee emp.id=emp.id} \]

\[
\begin{align*}
6. & \quad \sigma \text{ Employee job=7} \\
7. & \quad \pi \text{ employee name, gender, job} \\
8. & \quad (\pi \text{ emp.id Employee}) \times (\pi \text{ emp.id Employee})
\end{align*}
\]
\((\pi_{e1.\text{emp-id}, e1.\text{gender}}(e1)) \times (\pi_{e2.\text{emp-id}, e2.\text{gender}}(e2))\)
SQL - standard language
- queries and many more kinds of statements
- tables: default, bags (they can have duplicates)

dbclass.cs.pdx.edu

a site where you each have a DB
\[
\pi (0) \; (\text{employee } \times \text{ position}) = \text{job = id}
\]

\[
\text{SELECT } \ldots \text{ employee, position}
\]

\[
\text{WHERE job = id}
\]
CREATE TABLE widget
    ( sku integer, primary key,
      name character (15),
      price integer )

INSERT INTO widget

VALUES ( 104, 'small motor', 20 ),
        ( 105, 'med. motor', 30 )

ALTER TABLE widget RENAME TO
ALTER TABLE widget ADD COLUMN weight integer
Query 1:
(—
— )
INTERSECT
(—
— )

Query 2:
(SELECT ...
FROM ...
WHERE ...)
UNION
(SELECT ...
FROM ...
WHERE ...)

Query 3:
(—)
EXCEPT
(—)