Relational Model 1: Tables and Keys

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong
The relational model is the *de facto* standard implemented in all the major database systems. It defines:

1. the format by which data should be stored;
2. the operations for querying the data.

We will focus on the first aspect in this lecture, leaving the second aspect to the next lecture.

A database conforming to the relational model is called a relational database.
In a relational database, data are stored in **tables**.

### PROF

<table>
<thead>
<tr>
<th>pid</th>
<th>name</th>
<th>dept</th>
<th>rank</th>
<th>sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>Adam</td>
<td>CS</td>
<td>asst</td>
<td>6000</td>
</tr>
<tr>
<td>p2</td>
<td>Bob</td>
<td>EE</td>
<td>asso</td>
<td>8000</td>
</tr>
<tr>
<td>p3</td>
<td>Calvin</td>
<td>CS</td>
<td>full</td>
<td>10000</td>
</tr>
<tr>
<td>p4</td>
<td>Dorothy</td>
<td>EE</td>
<td>asst</td>
<td>5000</td>
</tr>
<tr>
<td>p5</td>
<td>Emily</td>
<td>EE</td>
<td>asso</td>
<td>8500</td>
</tr>
<tr>
<td>p6</td>
<td>Frank</td>
<td>CS</td>
<td>full</td>
<td>9000</td>
</tr>
</tbody>
</table>

- Each row is also called a **tuple**.
- Each column is also called an **attribute**.
- The **relation schema** of a table is the set of its attribute names.
  - E.g., the schema of the above table is \{pid, name, dept, rank, sal\}.
In a table, a candidate key is a minimal set $K$ of attributes such that no two tuples are allowed to be equivalent on all the attributes in $K$.

E.g., in the PROF table of the previous slide, if we set $\{\text{pid}\}$ as a candidate key, then no two tuples can have the same pid.

- A candidate key is designated when the table is created.
- There can be multiple candidate keys.
  - E.g., if you want, you can specify $\{\text{name}\}$ as another candidate key, but do you think it makes sense?
  - How about $\{\text{dept, rank}\}$?
<table>
<thead>
<tr>
<th>cid</th>
<th>title</th>
<th>dept</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>database</td>
<td>CS</td>
<td>2011</td>
</tr>
<tr>
<td>c2</td>
<td>signal processing</td>
<td>EE</td>
<td>2012</td>
</tr>
</tbody>
</table>

How would you set a candidate key?
As a good practice, every table should have at least a candidate key, a convention that will be enforced in the rest of the course. This implies that no two tuples in the table can be entirely equivalent to each other (think: why?).
In a table, if $K$ is a candidate key, any super set of $K$ is called a **super key**.

E.g., in the PROF table (pid, name, dept, rank, sal) in Slide 3, \{pid\} is a candidate key. Hence, all the following are super keys:

- \{pid\}
- \{pid, name\}
- \{pid, dept\}
- \{pid, rank, sal\}
- ...

Relational Model 1: Tables and Keys
Lemma

In a table, no two tuples can be equivalent on all the attributes of a super key.

The proof is easy and left to you.
Foreign Key

Definition
Let $T$ and $T'$ be two tables, and $K$ a candidate key in $T$. If $T'$ also contains $K$, then $K$ is a foreign key of $T'$ referencing $T$.

See the next slide for an example.
Suppose that PROF has a candidate key \{\text{pid}\}, and CLASS has a candidate key \{\text{cid, year}\}. Then:

- \{\text{pid}\} is a foreign key of TEACH referencing PROF.
- \{\text{cid, year}\} is a foreign key of TEACH referencing CLASS.
### How would you designate a candidate key for TEACH?

**TEACH**

<table>
<thead>
<tr>
<th>pid</th>
<th>cid</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td>2011</td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td>2012</td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td>2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>