Converting ER Diagrams to Tables

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong
An ER diagram is a pictorial representation of the information that can be captured by a database. Such a “picture” serves two purposes:

- It allows database professionals to describe an overall design concisely yet accurately.
- (Most of) it can be easily transformed into the relational schema.

This lecture will talk about the second bullet.
Recall that an ER diagram consists of:

- Entity sets.
- Relationship sets.

We will convert each entity/relationship set to a table, which involves deciding the table’s

- attributes
- and candidate key.
Given an entity set, create a table with the same attributes and candidate key.

Example:

PROF (pid, hkid, dept, rank, salary)
where pid is set as a candidate key.
Let us first consider binary relationship sets. To convert $R$ as shown below

![Relationship Diagram](image)

we create a table whose attributes include:

- the candidate keys of both $A$ and $B$
- the attributes of $R$ (if any).

The candidate key of the table, however, depends on the cardinality constraint of $R$. See the next few slides.
Many-to-many: The candidate key of $R$ includes all the attributes in the candidate keys of $A$ and $B$.

Example:

TAKE (sid, cid, year, grade) where the candidate key is (sid, cid, year).

Convert ER Diagrams to Tables
One-to-many: The candidate key of $R$ is the same as $B$ (i.e., the entity set on the “many” side).

Example:

```
PROF TEACH CLASS
cid
year
. . .
pid
. . .
```

TEACH (pid, cid, year)
where the candidate key is (cid, year).

Think Why?
One-to-one: $R$ has two candidate keys. The first (second) one is the same as $A$ ($B$).

Example:

MARRY (mid, wid, year)
where a candidate key is mid, and another is wid.
Multi-way relationship set $R$: Create a table that includes

- the candidate keys of the participating entity sets
- the attributes of $R$ (if any).

The candidate key of the table includes all the attributes of the candidate keys of the participating entity sets.
PROF (pid, hkid, dept, rank, salary), candidate key pid
CLASS (cid, title, year, dept), candidate key (cid, year)
STU (sid, dept, gpa), candidate key sid
TEACH (pid, cid, year), candidate key (cid, year)
TAKE (cid, year, sid, grade), candidate key (cid, year, sid)
What has not been captured by our conversion?

**Participation constraints!** For example, how to ensure that every class is taken by at least one student?

Participation constraints can be checked whenever necessary with SQL queries. Think: how?

Interestingly, the total participation of CLASS in the relationship set TEACH can be captured with a clever conversion. See the next slide.
PROF (pid, hkid, dept, rank, salary), candidate key pid
CLASS (cid, title, year, dept, pid), candidate key (cid, year)
STU (sid, dept, gpa), candidate key sid
TAKE (cid, year, sid, grade), candidate key (cid, year, sid)

Note that there is no explicit table for TEACH. Think: why doesn’t the trick work for TAKE?