Translation to Relational Schema

Agenda this Week

- Translation of the Entity Relationship Diagram into a Relational Schema

- Understanding the Stable Translation method

- Understanding the concept of Foreign Keys

- Understanding the concept of Referential Integrity

- Perform Translation Exercises

An ER Model is intended as a description of real-world entities. It is constructed in such a way as to allow easy translation to the relational model. The ER diagram represents the conceptual level of database design. A relational schema is at the logical level of database design.

When translating your diagram into a schema, an entity type turns into a table. The name of the entity should become the name of the table. Adhere to the naming restrictions of the DBMS that you are using, avoid reserved words. Many DBMS applications do not allow spaces in names and are not case sensitive. Follow a consistent set of conventions when naming tables.

Each attribute turns into a column in the table. The primary key of the entity is the primary key of the table. It can be composite if required. The primary key can never be null. There is no such thing as a multi-valued attribute in a relational database. If you have a multi-valued attribute, take the attribute and turn it into a new entity of its own. Then make a 1:M

relationship between the new entity and the existing one. Then convert as normal. This will be discussed further when we cover the topic of normalization in the course.

- ■ Considerations for Translating

Normally one should minimize the number of tables is the database to reduce query-processing time. However, we are focusing on the stable translation method which actually increases the number of tables in the database. It does not increase the performance of the system; however it creates a framework that is more easily maintained. System development is always a balancing act. The benefit of the stable translation is that it:

- ■ Does not allow null values if possible
- ■ Provide a semantically clear design
- ■ Provide a design that accommodates potential changes of the schema

The reality of database design is that In general, decreasing the number of tables means increased efficiency of query processing. However, But is also means more null values are allowed in columns during data insertion. Additionally, there less semantic clarity and less flexibility in the model of the structure of the physical database when less tables are use. On the other hand, more tables provide more understanding of the structure. The down side is that query performance is affected by have more tables.

There are three primary Translation Techniques used:

- ■ Stable Translation
- ■ Mapped Translation
- ■ Mapped with Total/Partial Translation

The focus of the course will be using the stable translation method

- General Rules for translation – Regardless of technique

    - A weak (or identifying) relationship is always combined with the weak entity

    - Each M:N relationship becomes a separate table

   - Each ternary relationship becomes a separate table

Specific Rules for Stable Translation

- Every entity becomes a table

    - All the attributes of the entity become the attributes of the table

- Every relationship becomes a table

    - Add RID to the relationship table

    - Add any non-key attribute of the relationship to the relationship table

    - A weak relationship is always combined with the weak entity

- Stable translation is called "stable" because a change in the cardinality or participation constraints does not change the table structure

- Advantage

    - Provides stable schema against constraint changes

- Disadvantage

    - Generates a larger number of tables than other methods

    - More time to process query statements

Mapped Translation

- ■ Advantage
    - ■ Minimize the number of tables
- ■ Disadvantages
    - ■ The schema can be changed if cardinality changes
    - ■ Foreign keys may have null values
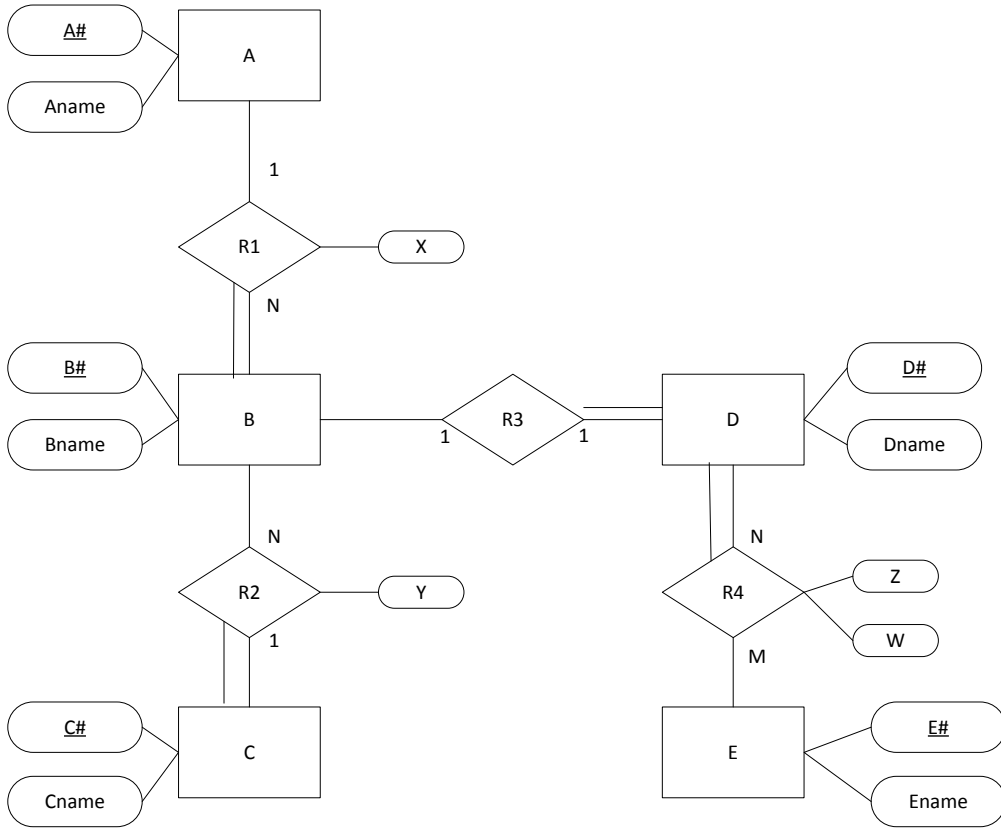
Comparison Stable vs. Mapped
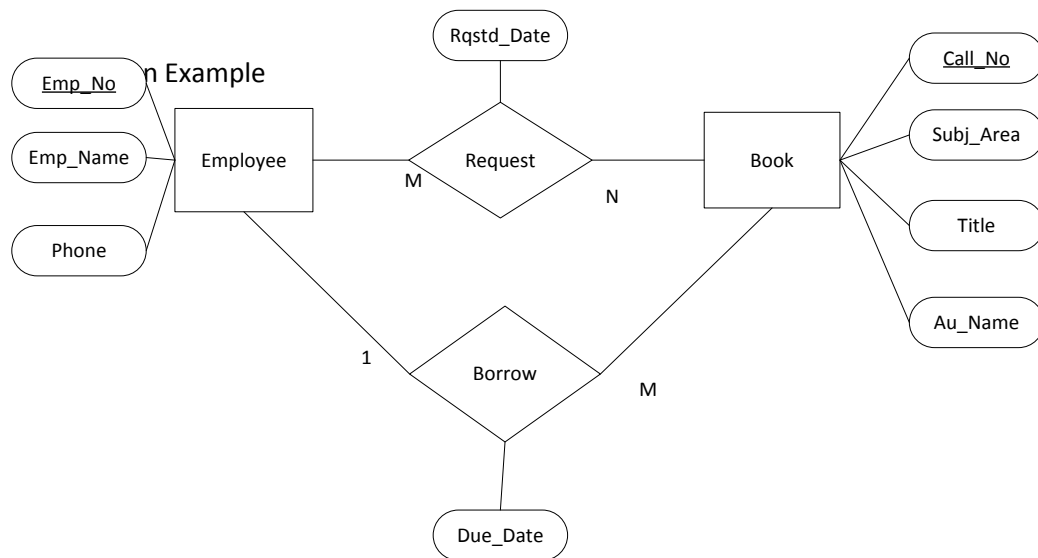
n Example

STABLE TRANSLATION METHOD

Employee(<u>Emp_No</u>, Emp_Name, Phone)

Request(<u>Emp_No</u>, <u>Call_No</u>, Rqstd_Date)

Borrow(<u>Call_No</u>, Emp_No, Due_Date)

Book(<u>Call_No</u>, Subj_Area, Title, Au_Name)

MAPPED TRANSLATION METHOD

<u>Employee(Emp_No, Emp_Name, Phone)</u>

<u>Request(Emp_No, Call_No, Rqstd_Date)</u>

<u>Book(Call_No, Subj_Area, Title, Au_Name, Emp_No, Due_Date)</u>

**Example of the
STABLE TRANSLATION METHOD**

A(<u>A#</u>, Aname)
B(<u>B#</u>, Bname)
C(<u>C#</u>, Cname)
D(<u>D#</u>, Dname)
E(<u>E#</u>, Ename)
R1(<u>B#</u>, A#, X)