Database Design

- A Database Design Methodology
- Mapping of E-R Schemas into Relations

**A DB Design Methodology**

- Identify & describe data and user views
- Conceptual modeling
- View integration
- Operation description
- Map to specific DB schema
- Refine the DB schema
- Refine the operation desc.
- Tuning up performance
- User interfaces
- Application programs

**Regular Entity Types**

- Map each regular entity type $E$ to a relation $R$
- Include in $R$
  - All simple attributes of $E$
  - The simple component (leaf) attributes of all composite attributes
- Use the PK of $E$ as the PK of $R$

Instructor (Emp#, FName, Minit, LName)
Multivalued Attributes

- Create a new relation $M$ for each multivalued attribute of $E$
- Include in $M$ the multiple attribute
- Include the PK of $R$ as the FK in $M$
- The PK of $M$ consists of all attributes of $M$

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp#, FName, MInit, LName</td>
<td>LName, EMP#</td>
</tr>
</tbody>
</table>

Weak Entity Types

- Create a relation $R_w$ for each weak entity type $E_w$
- Include in $R_w$
  - All attributes of $E_w$
  - All attributes of the identifying relationship
- Include the PK of the owner entity type as the FK in $R_w$
- Combine the FK and the partial key of $E_w$ if any as the PK of $R_w$

<table>
<thead>
<tr>
<th>Student</th>
<th>Car</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN, Name, Age, ...</td>
<td>PNum, Color, P_Lot, Date, SSN</td>
</tr>
</tbody>
</table>

Relationships

CASE I: Connectivity - 1:1 Relationship

- Create a relation $R$ for each relationship, if relationship has attributes
- Include the PKs of participating entity types as PK in $R$
- Create a relation $R_n$ for each Entity
- The FK may be placed in either Entity of the relationship, choose the one that makes the most sense

<table>
<thead>
<tr>
<th>Car</th>
<th>Registration</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoID, Make, Model, Yr, Color</td>
<td>TagNo, InsID, ExpDate, AutoID</td>
</tr>
</tbody>
</table>
CASE II: Connectivity - 1:M Relationship

- In a one-to-many relationship
  - Choose the many side relation (one of relations), say R1.
  - Include the PK of the one (other) side relation as the FK in R1.
  - If relationship contains attributes, include in R1.

Child (SSN, Name, DOB, MSSN)
Mother (SSN, Name, DOB, Address)

CASE III: Connectivity - M:N Relationship

- In a many-to-many relationship
  - Create Relations for each Entity include PKs
  - Create a Relationship Relation, R
  - The PK of R consists of the PKs of each Entity relation
  - Any attributes of the Relationship Relation remain with it

PILOT (PID, PName, Address, DOB, YrsWCo)
FLIES (PID, A/C#, Hrs_Flown)
AIRCRAFT (A/C#, SeatCapacity)

N-ary Relationships

- Same as for M:N relationships
**IS-A Relationships**

- Individual relation approach:
  - Map each entity type to a relation
  - For each subtype relation $R_{sub}$:
    - Include the PK of its supertype relation as the FK in $R_{sub}$
    - Combine the FK and the PK of $R_{sub}$ if any as the new PK of $R_{sub}$

  ![Diagram](image)

  - Person
  - Instructor
  - Student
  - Person (SSN, Name, Age, Address, ...)  
  - Instructor (SSN, Salary, Office, ...)  
  - Student (SSN, Year, GPA, ...)

**Other Issues**

- Names conflicts can be resolved by renaming the names
- An unary relationship can be mapped as a binary relationship
- Application programs must enforce connectivity and participation constraints

![Diagram](image)