EER schema to Relational database schema

SWEN304/SWEN439

Trimester 1, 2021

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Transformation EER $\rightarrow$ Relational (1)

- we describe the transformation of an EER schema into a relational database schema
- start with entity types (object types of order 0) and then work one's way up gradually
- Transformation of **entity types**:
  - an entity type $E = (attr(E), key(E))$ leads to a relation schema E'
  - the attributes of E are preserved
  - the domain assignment for the attributes of E is preserved, too
  - the key of E becomes the primary key of the relation schema E'
- **Example**: $\textsc{Department} = (\{\text{No,Budget}\}, \{\text{No}\})$ is transformed to $\textsc{Department} = \{\text{No,Budget}\}$ with primary key $\{\text{No}\}$

Transformation EER $\rightarrow$ RDM (2)

- If attribute names of object types are not unique globally, i.e. some attributes in different object types share the same names, rename attribute names.
- for a relationship type $R = (comp(R), attr(R), id(R))$ and each component $C \in comp(R)$ choose pairwise disjoint sets of new attribute names not occurring in $attr(R)$:

$$k\_attr(C) = \{C.A \text{ where } A \text{ is a key attribute of } C'\}$$

Rename the key attributes of the components of R by attaching the component name to each of its key attributes.

where C' is originating from a prior transformation of component C

- **Example:** rename attribute names for entity types:
  - $\textsc{Departmant} = (\{\text{No, Budget}\}, \{No\})$, rename its key attribute $k\_attr(\textsc{Department})$ to {Department_No}
- If all attribute names of object types are unique globally, can omit the step of renaming attribute names

Transformation EER $\rightarrow$ RDM (3)

- Transformation of **relationship types**:
  - a relationship type $R = (comp(R), attr(R), id(R))$ leads to relation schema R' with

$$attr(R') = \bigcup_{C \in comp(R)} k\_attr(C) \cup attr(R)$$

    Attribute of R' is the union of key attributes of its components and its own attributes
  - the domain assignment of the attributes is preserved
    The domains of attributes in relation schema R' are the same as defined in relationship type R
  - R' has the primary key

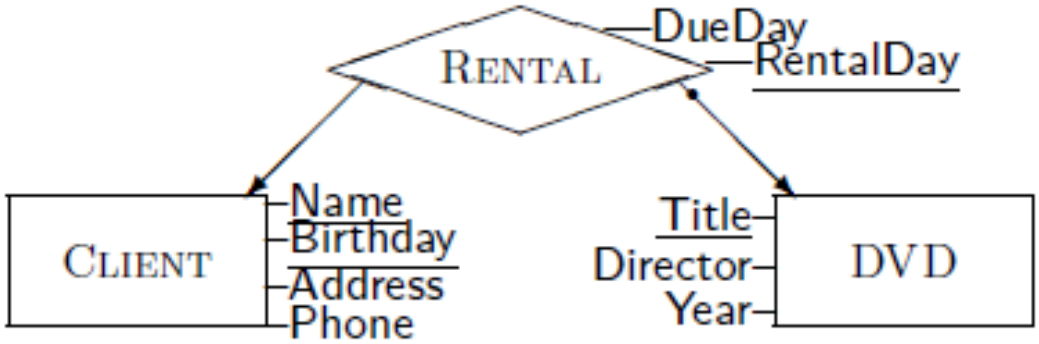
$$\bigcup_{C \in key(R) \cap comp(R)} k\_attr(C) \cup (key(R) \cap attr(R))$$

    The primary key of relation R' is composed of the key attributes of its key components and its own key attributes.
  - each component $C \in comp(R)$ defines a foreign key

$$[C.A_1, \ldots, C.A_n] \subseteq C'[A_1, \ldots, A_n]$$

    on R' for $key(C) = \{A_1, \ldots, A_n\}$

Example - Transforming Relationship Types



All attributes are unique globally, so no need to rename attribute names

- relationship type
  $\textsc{Rental} = (\{\textsc{Client}, \mathrm{DVD}\}, \{\text{DueDay,RentalDay}\}, \{\mathrm{DVD}, \text{RentalDay}\})$
  becomes:
  - $\textsc{Rental} = \{\text{Name, Birthday, Title, DueDay, RentalDay}\}$ with
  - primary key: {Title,RentalDay}
  - foreign keys: [Name, Birthday] $\subseteq$ $\textsc{Client}$[Name, Birthday]
    [Title] $\subseteq$ DVD[Title]

How to handle Clusters

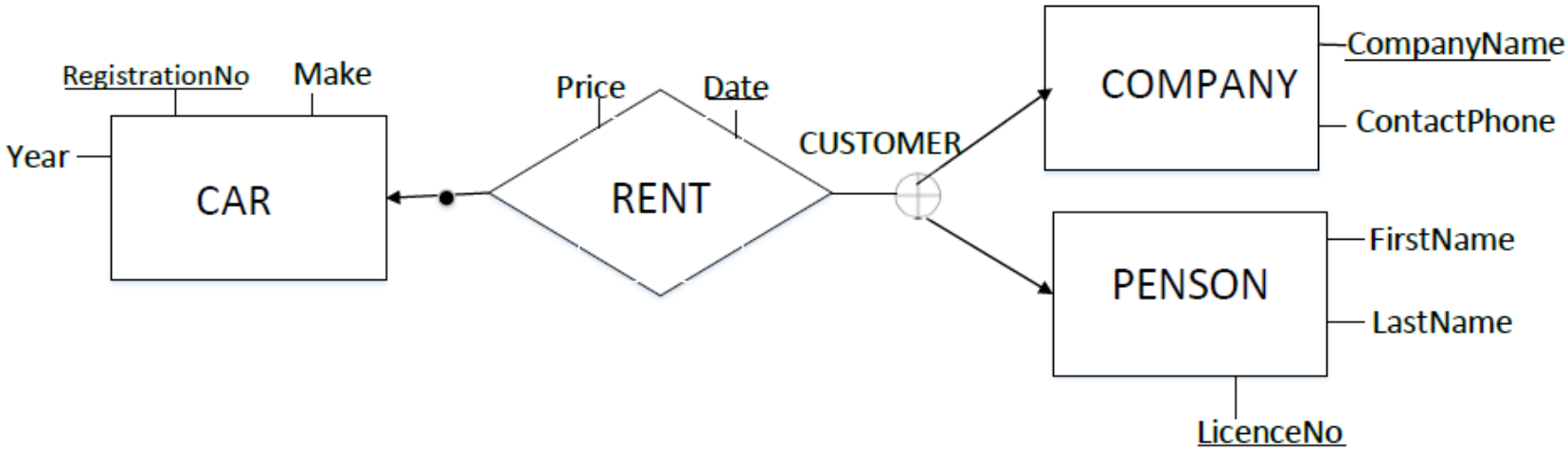
- cluster types used in conceptual design to model alternatives
- the relational data model does not provide similar concept
- transform EER schema with clusters into equivalent EER schema without clusters
- only necessary as pre-processing before actual transformation to the relational data model
- in general: clusters provide database designers a convenient way to model objects in the target of database
- do not recommend to avoid clusters as size of EER schema increases dramatically and becomes harder to comprehend

Transformation EER $\rightarrow$ RDM (3)

- replacing clusters in EER schema is straightforward:
  - cluster types in EER schema $\mathcal{S}$ that are not component of any relationship type can be removed from $\mathcal{S}$
  - consider relationship type R with cluster component $C = C_1 \oplus \cdots \oplus C_n$
  - replace R by n new relationship types $R_1, \ldots, R_n$:
  - for $i = 1, \ldots, n$: R_i obtained from R by replacing every occurrence of C by C_i
  - if R_i still contain clusters, then repeat process of replacing these clusters by its components
  - the final EER schema is cluster-free and previous transformation to RDM can be applied

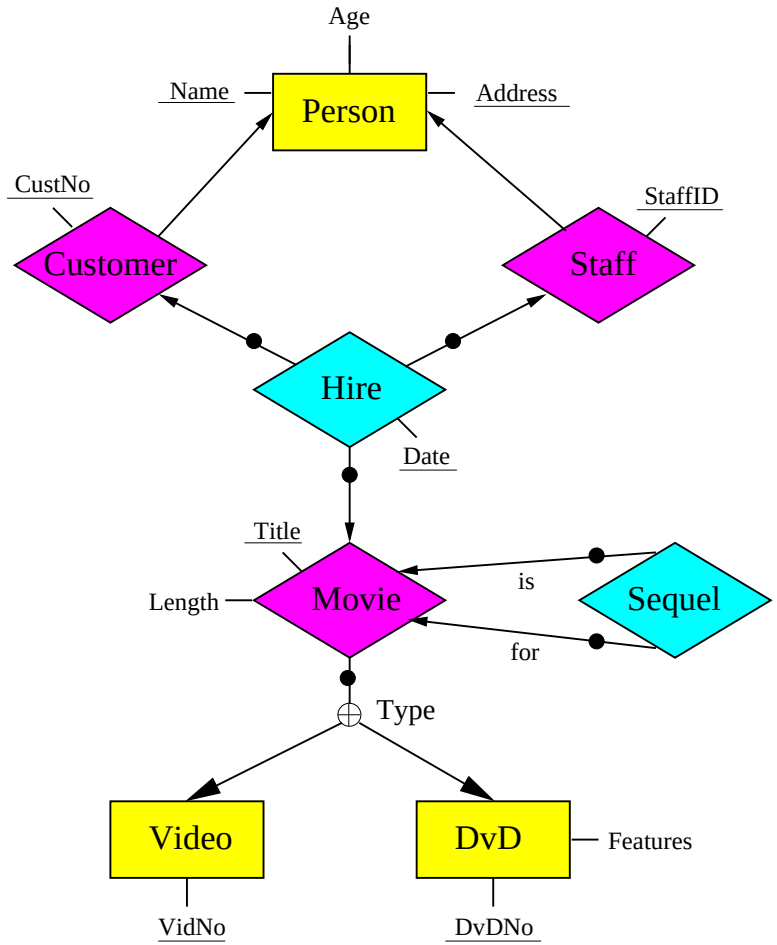
Example - Transforming Clusters

- consider the following EER diagram:



- Rent={Car, Customer},{Date, Price},{Car, Date}) with
  - cluster Customer = Company $\oplus$ Person
- replace Customer by its 2 two components, i.e. object types Company and Person we obtain:
  - Company_Rent:
    ({Car, Company},{Date,Price},{Car, Date})
  - Person_Rent:
    ({Car, Person},{Date,Price},{Car, Date})

EER Schema for MovieDB (1)



- Describe the Entity- and Relationship Types and Clusters in the EER diagram, grouping the object types according to their orders
- Transform the corresponding HERM schema into a Relational Database schema. Make all key constraints explicit.