Johns Hopkins Engineering

Principles of Database Systems

Module 8 / Lecture 1

Normalization for Relational Databases II

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Relation Decomposition

- Universal Relation $R = \{ A_1, A_2, \ldots, A_n \}$
- Decomposition of R: $D = \{ R_1, R_2, \ldots, R_n \}$
- No attributes are lost after decomposition. In other words, each attribute in R will appear in at least one relation as *attribute preservation*.
  - $R_1 \cup R_2 \cup R_3 \ldots R_n = R$

Properties of Decompositions

- **Dependence preservation property**
  - Decomposed relations preserve all original dependencies

- **Nonadditive (lossless) join property**
  - After natural joining decomposed relations, no spurious tuples are allowed.
  - Previous 2NF and 3NF examples demonstrate successive nonadditive join decomposition.

Join Loss with Null Values in Foreign Key

(a)

EMPLOYEE

Ename	Ssn	Bdate	Address	Dnum
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX	5
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1
Berger, Anders C.	999775555	1965-04-26	6530 Braes, Bellaire, TX	NULL
Benitez, Carlos M.	888664444	1963-01-09	7654 Beech, Houston, TX	NULL

DEPARTMENT

Dname	Dnum	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

Figure 15.2a Issues with NULL-value joins. Some EMPLOYEE tuples have NULL for the join attribute Dnum.

Join Loss with Null Values in Foreign Key (cont.)

(b)

Ename	Ssn	Bdate	Address	Dnum	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Figure 15.2b Issues with NULL-value joins. Result of applying NATURAL JOIN to the EMPLOYEE and DEPARTMENT relations.

(c)

Ename	Ssn	Bdate	Address	Dnum	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555
Berger, Anders C.	999775555	1965-04-26	6530 Braes, Bellaire, TX	NULL	NULL	NULL
Benitez, Carlos M.	888665555	1963-01-09	7654 Beech, Houston, TX	NULL	NULL	NULL

Figure 15.2c Issues with NULL-value joins. Result of applying LEFT OUTER JOIN to EMPLOYEE and DEPARTMENT.

Join Loss with Null Values in Foreign Key (cont.)

- A similar problem due to the null values in the FK columns is called *dangling* records.

- The problem may occur if the design is decomposed too far.

Join Loss with Null Values in Foreign Key (cont.)

(a) EMPLOYEE_1

Ename	Ssn	Bdate	Address
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX
Berger, Anders C.	999775555	1965-04-26	6530 Braes, Bellaire, TX
Benitez, Carlos M.	888665555	1963-01-09	7654 Beech, Houston, TX

Figure 15.3 The dangling tuple problem. (a) The relation EMPLOYEE_1 (includes all attributes of EMPLOYEE from Figure 15.2(a) except Dnum).

Join Loss with Null Values in Foreign Key (cont.)

(b) EMPLOYEE_2

Ssn	Dnum
123456789	5
333445555	5
999887777	4
987654321	4
666884444	5
453453453	5
987987987	4
888665555	1
999775555	NULL
888664444	NULL

(c) EMPLOYEE_3

Ssn	Dnum
123456789	5
333445555	5
999887777	4
987654321	4
666884444	5
453453453	5
987987987	4
888665555	1

Figure 15.3 The dangling tuple problem. (b) The relation EMPLOYEE_2 (includes Dnum attribute with NULL values). (c) The relation EMPLOYEE_3 (includes Dnum attribute but does not include tuples for which Dnum has NULL values).

Multivalued Dependencies (MVD)

- MVD represents a dependence between attributes (e.g., A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. The set of values for B and C are independent of each other.
- MVDs are the consequence of 1NF, which disallowed an attribute with multiple values.
- Informally, if two independent 1:M relationships are mixed in the same relation, an MVD may arise.

Fourth Normal Form (4NF)

- A relation *R* is in 4NF if and only if, the relation is in Boyce-Codd normal form and contains no *nontrivial multi-valued dependencies*.

Examples:

EMP(ENAME, PNAME, DNAME) →

EMP_PROJECT(ENAME, PNAME) and

EMP_DEPENDENT(ENAME, DNAME)

EMP1(ENAME, DEGREE, LANGUAGE) →

EMP1_DEGREE(ENAME, DEGREE) and

EMP1_LANGUAGE (ENAME, LANGUAGE)

Fourth Normal Form (4NF) (cont.)

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b) EMP_PROJECTS

Ename	Pname
Smith	X
Smith	Y

EMP_DEPENDENTS

Ename	Dname
Smith	John
Smith	Anna

Figure 14.15 Fourth and fifth normal forms. (a) The EMP relation with two MVDs: Ename –>> Pname and Ename –>> Dname. (b) Decomposing the EMP relation into two 4NF relations EMP_PROJECTS and EMP_DEPENDENTS.

Fourth Normal Form (4NF) (cont.)

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John
Brown	W	Jim
Brown	X	Jim
Brown	Y	Jim
Brown	Z	Jim
Brown	W	Joan
Brown	X	Joan
Brown	Y	Joan
Brown	Z	Joan
Brown	W	Bob
Brown	X	Bob
Brown	Y	Bob
Brown	Z	Bob

(b) EMP_PROJECTS

Ename	Pname
Smith	X
Smith	Y
Brown	W
Brown	X
Brown	Y
Brown	Z

EMP_DEPENDENTS

Ename	Dname
Smith	Anna
Smith	John
Brown	Jim
Brown	Joan
Brown	Bob

Figure 15.4 Decomposing a relation state of EMP that is not in 4NF. (a) EMP relation with additional tuples. (b) Two corresponding 4NF relations EMP_PROJECTS and EMP_DEPENDENTS. Decomposing a relation state of EMP that is not in 4NF.

The EMP relation is BCNF. The EMP relation with two MVDs: Ename –>> Pname and Ename –>> Dname; and additional tuples.

Johns Hopkins Engineering

Principles of Database Systems

Module 8 / Lecture 2

Normalization for Relational Databases II

Fifth Normal Form (5NF)

- A relation R is subject to a *join dependency*
  - R can be decomposed to $(R_1, R_2, \ldots, R_n)$ and each has a subset of the attributes of R
  - R can always be recreated by joining the multiple relations $(R_1, R_2, \ldots, R_n)$

- Relation R is in 5NF if and only if, R is in 4NF and the relation has no join dependency.

Fifth Normal Form (5NF) (cont.)

- A cyclical nature of a relation may require further normalization.

  Example with an embedded three M:N relationships:

  SUPPLIER_PART_PROJ(SNAME, PARTNAME, PROJNAME)

  SUPPLIER_PART(SNAME, PARTNAME),
  SUPPLIER_PROJ(SNAME, PROJNAME),
  PART_PROJ(PARTNAME, PROJNAME)

Fifth Normal Form (5NF) (cont.)

(c) SUPPLY

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(d) R_1

Sname	Part_name
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

Sname	Proj_name
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

Part_name	Proj_name
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Figure 14.15 Fourth and fifth normal forms. (c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R_1, R_2, R_3). (d) Decomposing the relation SUPPLY into the 5NF relations R_1, R_2, R_3.

Fifth Normal Form (5NF) (cont.)

S: Supplier, P: Part, J: Project

If S1→P1, S1→J1, P1 → J1 Then S1, P1, J1

SUPPLIER-PART-PROJECT

Supplier#	Part#	Project#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

Legal State with Join Dependency

Decompose

SUPPLIER-PART

Supplier#	Part#
S1	P1
S1	P2
S2	P1

PART-PROJECT

Part#	Project#
P1	J2
P2	J1
P1	J1

PROJECT-SUPPLIER

Project#	Supplier#
J2	S1
J1	S1
J1	S2

SUPPLIER-PART $|X|_{Part\#}$ PART-PROJECT

SUPPLIER-PART-PROJECT

Supplier#	Part#	Project#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S2	P1	J2
S1	P1	J1

Spurious

SUPPLIER-PART-PROJECT $|X|_{Project\#,Supplier\#}$ PROJECT-SUPPLIER

SUPPLIER-PART-PROJECT

Supplier#	Part#	Project#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

Fifth Normal Form (5NF) (cont.)

Example:

- Agents represent companies. Companies represent properties. Agents sell properties.
- Mary sells RE/MAX home property and Century 21 commercial property
- Mary does not sell RE/MAX commercial property nor Century 21 homes

Agent	Company	Property
Mary	RE/MAX	home
Mary	Century 21	commercial property

- If an agent sells a certain property and the agent represents the company, then she sells properties for that company

Agent	Company	Property
Mary	RE/MAX	home
Mary	RE/MAX	commercial property
Mary	Century 21	home
Mary	Century 21	commercial property
Steve	RE/MAX	home

- Repetition of facts for Mary - Sell home twice

Fifth Normal Form (5NF) (cont.)

Example:

- No repetition of facts
- Reconstruct all true facts from 3 relations instead of the single relation.

Agent	Company
Mary	RE/MAX
Mary	Century 21
Steve	RE/MAX

Company	Property
RE/MAX	home
RE/MAX	commercial property
Century 21	home
Century 21	commercial property

Agent	Property
Mary	home
Mary	commercial property
Steve	home

Johns Hopkins Engineering

Principles of Database Systems

Module 8 / Lecture 3

Normalization for Relational Databases II

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Additional Notes on Normalization

- Case tools do not understand functional dependencies (not expert systems). Therefore, they can not fully support all normal forms.
- Normalization is executed in a series of steps. As normalization proceeds, the relations become more restricted to meet the required normal forms' criteria.
- Normalization comprehensively relies on functional dependencies among key attributes and non-key attributes.

Additional Notes on Normalization (cont.)

- Each normalization process may break down a relation into more relations. How much normalization is enough?
  - 3NF is the standard. When a database is *normalized*, it generally implies that the database is in 3NF.
  - In general, a normalized design is in 3NF that may also be BCNF, 4NF, and 5NF without additional decomposition.

Additional Notes on Normalization (cont.)

- **Problems Solved by 1NF:**
  - Resolve embedded multi-valued attributes and repeating groups
  - Resolve embedded one-to-many relationship

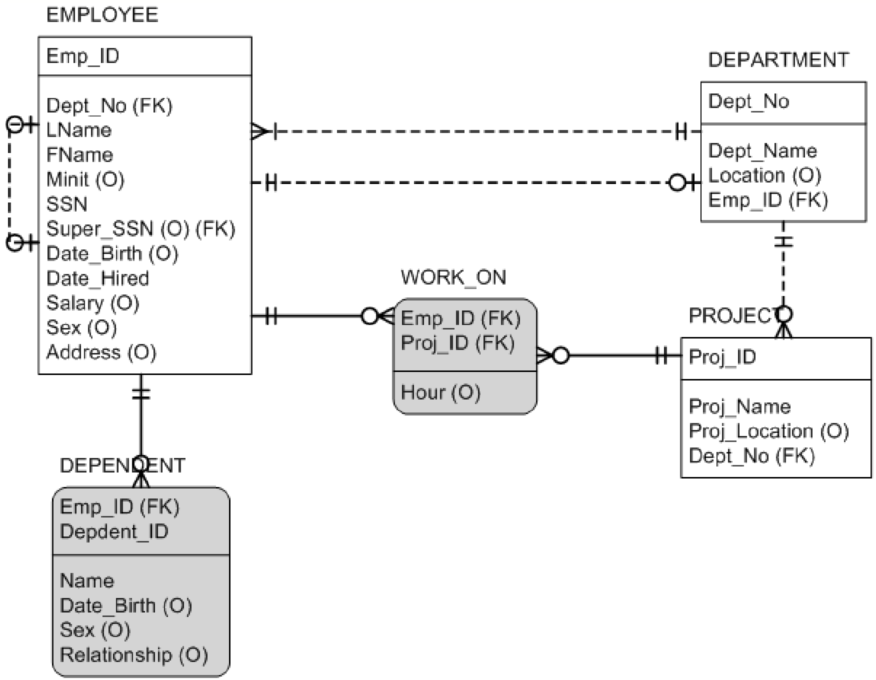
- **Problems Solved by 2NF:**
  - Resolve an attribute that does not depend on the FULL PK, it needs to be taken out to form a new relation
  - Resolve a one-to-many identifying relationship

Additional Notes on Normalization (cont.)

- **Problems Solved by 3NF:**
  - Take the attributes that functionally depend on another non-key attribute out to form a new relation. A new parent table will be created, and the original table is a child table with a non-identifying relationship.
  - Resolve a one-to-many non-identifying relationship since attributes are dependent on another non-key attribute (a PK of a parent table.)

Additional Notes on Normalization (cont.)

- Verify your ERD using normalization as a refinement process



Example: Company ERD
1. Check if all underlying domains contain atomic values (single-valued) only, not a set of values, not repeating groups.
2. Check for every non-key attribute is fully functionally dependent on the full primary key (no partial dependencies).
3. Check for no non-key attribute of R is functionally dependent on another non-key attribute.

Additional Notes on Normalization (cont.)

- A model may be normalized, but it may still not be a correct representation of the business.

- After the normalization process, the database usually consists of more tables. There are conditions that require a database to *denormalize* in favor of performance such as quicker response time, high throughput, and high frequency for a certain set of transactions.

Additional Notes on Normalization (cont.)

- When denormalizing the database design, always start with tables in 3NF.
- A foreign key appearing twice in an entity without rolenames implies a redundant relationship structure in the model.

Performance Issues on Database Design

- It is sometimes necessary to add or change the index structure or create a cluster to improve data access time. Indexes provide quick access to rows of data and avoid full table scan. They are automatically used when referenced in the WHERE clause of a SQL statement.

Performance Issues on Database Design (cont.)

- It is good practice to build indexes for primary keys (unique indexes) and foreign keys (generally non-unique indexes).

- May be helpful to build indexes for alternate keys (unique indexes), and any non-key columns frequently used in WHERE clauses

- Consider all other options prior to denormalization, especially adding or changing the index structure.

Performance Issues on Database Design (cont.)

- Be extremely reluctant to denormalize the default design because it may cause data inconsistency problems
- Can consider denormalizing the design to reduce the number of tables and avoid the join operation to improve system performance in web applications

The Benefits of A Set of Well-designed Tables

- Reduced storage of redundant data, which eliminates the cost of updating duplicates and avoids the risk of inconsistent results based on the duplicates
- Increased ability to effectively enforce integrity constraints
- Increased ability to adapt to the growth and change of the system
- Increased productivity based on the inherent flexibility of well-designed relational systems

Role of Normalization in the Database Development Process

- A refinement process, not as an initial design process
- Intuitively group related attributes to form your entity types and relationships in ERD
- Can be done in an informal manner without the tedious process of recording functional dependencies in practice
- May identify the overlooked M-N relationships

Johns Hopkins Engineering

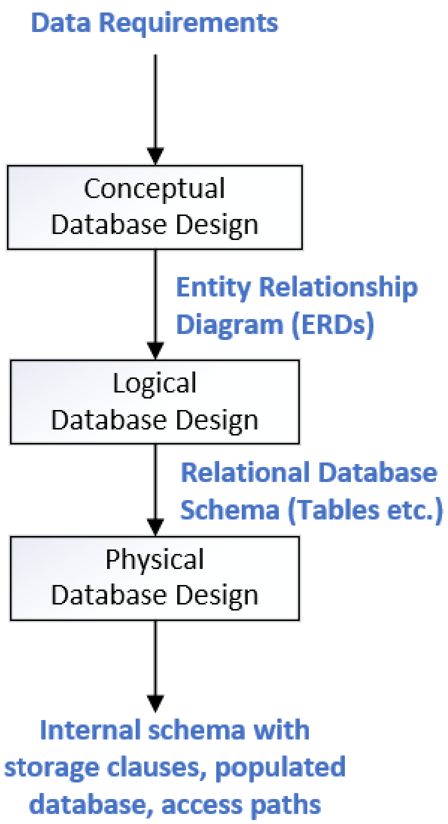
Principles of Database Systems

Module 8 / Lecture 4

ER and EER to Relational Mapping

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Database Design Process



Using a conceptual schema design (Entity Relationship Diagram) to create a relational database schema

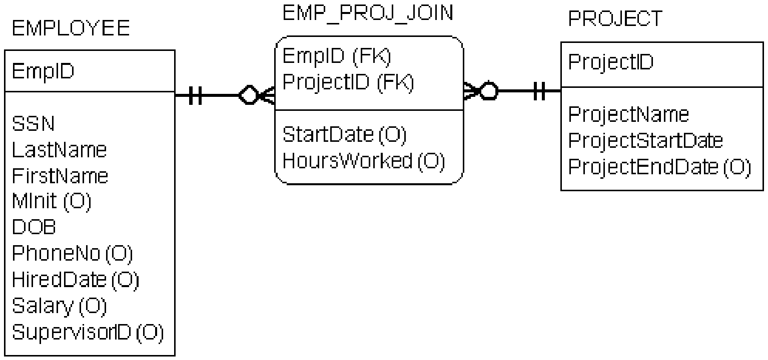
Converting ER Model to Relational Model

- Create relations for the conceptual data model to represent the entity types, relationships, and attributes that have been identified

- Implement the concepts of relational databases, primary keys, foreign keys, and data integrity

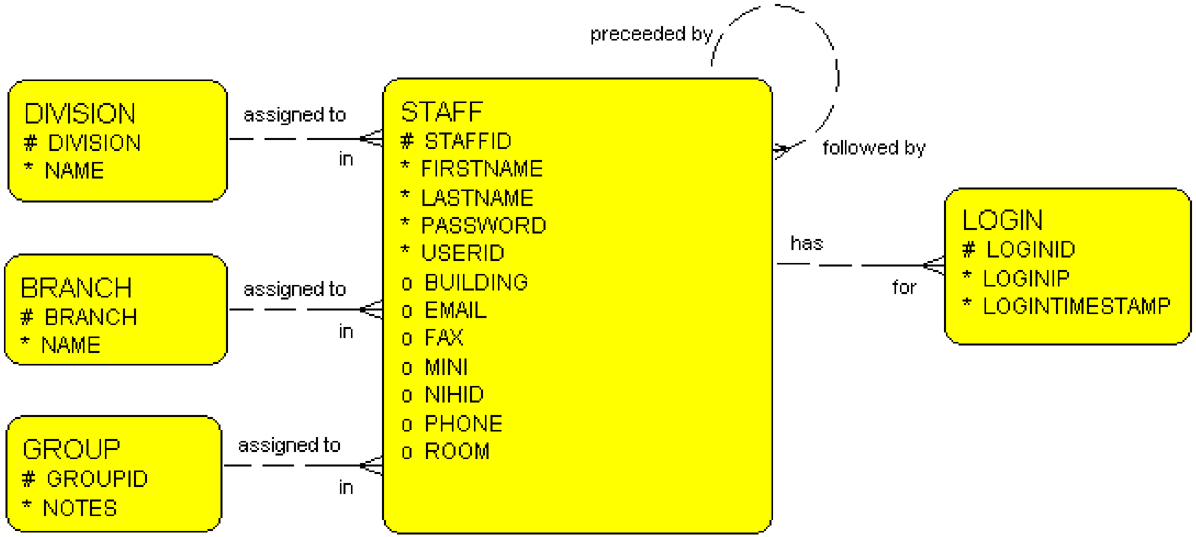
Converting ER Model to Relational Model (Cont.)

- Database design tools may have different graphical representations.
- ERwin and Visio show a foreign key attribute in a child relation:



Converting ER Model to Relational Model (Cont.)

- Oracle designer does not include a foreign key into a child relation. Why?



How to Map Entity Types and Relationships to Relations

ER Model	Mapping to Relation
Strong entity type	Create relation with all simple attributes
Weak entity type	Create relation with all simple attributes, and combine partial key of weak entity and a FK from the parent entity type as the PK
1:1 relationship type with mandatory participation on both sides	Combine entities into one relation or create two relations (see next) (e.g., EMPLOYEE vs. OFFICE or BADGE as 1-1 relationship)

How to Map Entity Types and Relationships to Relations (cont.)

ER Model	Mapping to Relation
1:1 relationship type with mandatory participation on one side	Post PK of entity on optional side to act as FK in relation representing entity on mandatory side (e.g., EMPLOYEE and CAR have 1-1 relationship)
1:M relationship type	Post PK of entity on **one** (parent) side to act as FK in relation representing entity on **many** (child) side

How to Map Entity Types and Relationships to Relations (cont.)

ER Model	Mapping to Relation
M:M relationship type	Create two 1:M relation types and follow above mapping and add additional attributes to the transitional relation (Be aware whether identifying or non-identifying relationships)
Multi-valued attribute	Create a new relation and post a copy of the PK of the parent entity into the new relation to act as a FK (e.g., DEPT_LOCATION)
N-ary relationship type	Create a new relation and post all PKs of the parent entities into the new relation to act as a PK and FKs (e.g., SUP_PRJ_PART_JOIN)

How to Map Entity Types and Relationships to Relations (cont.)

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Mapping COMPANY ER Schema Into A Relational Database Schema

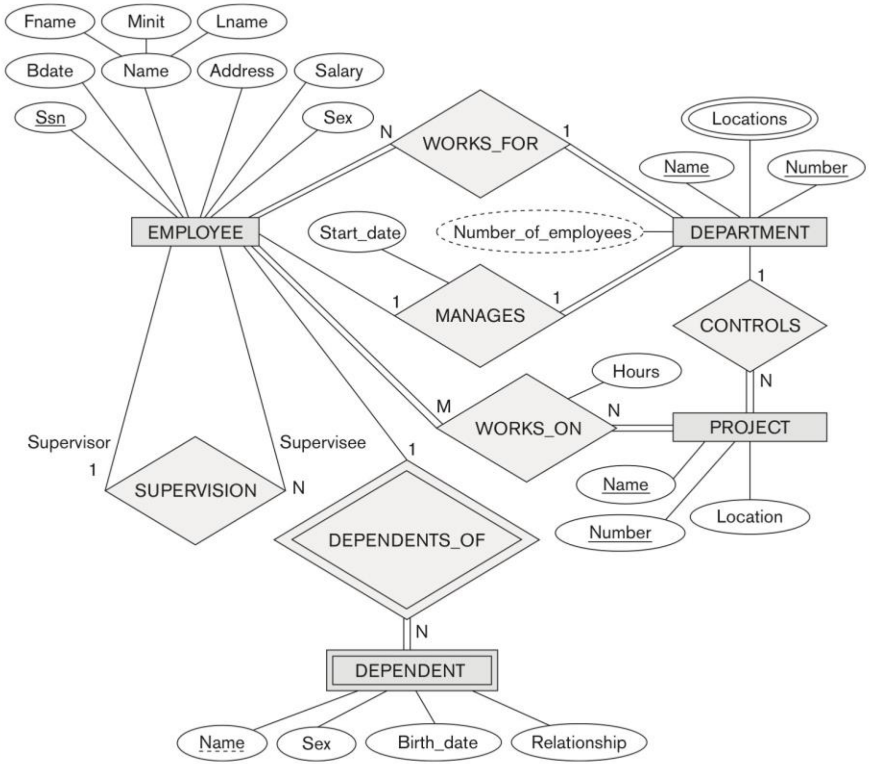


Figure 9.1 The ER conceptual schema diagram for the COMPANY database.

Mapping COMPANY ER Schema Into A Relational Database Schema (cont.)

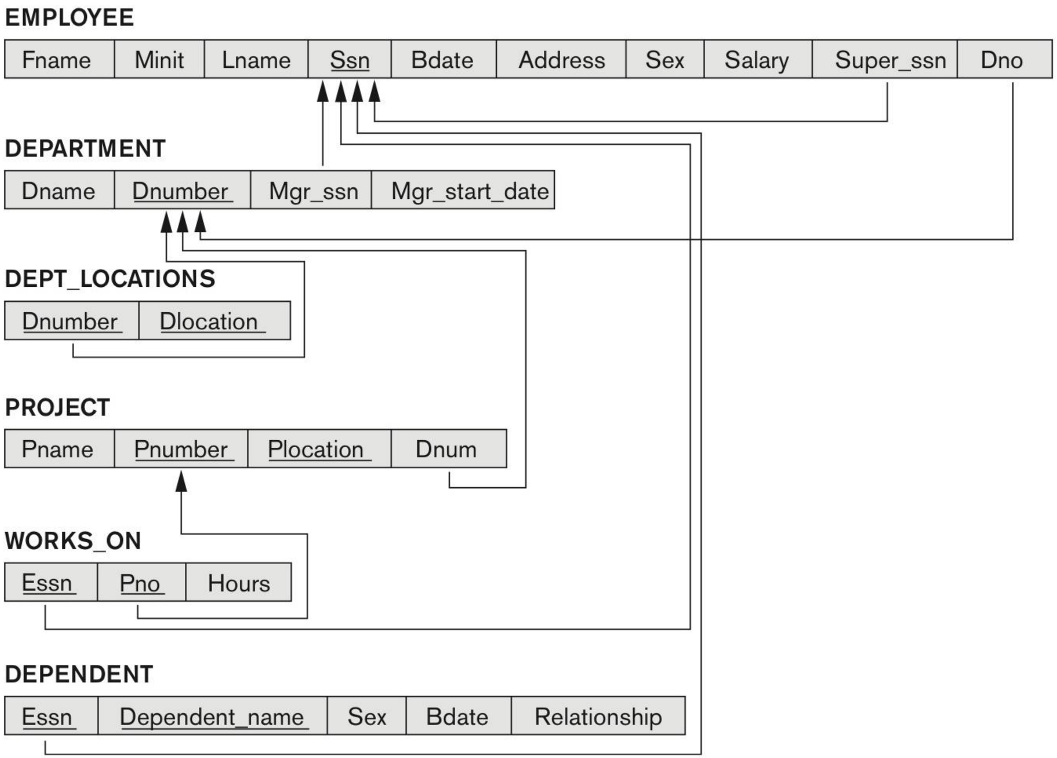
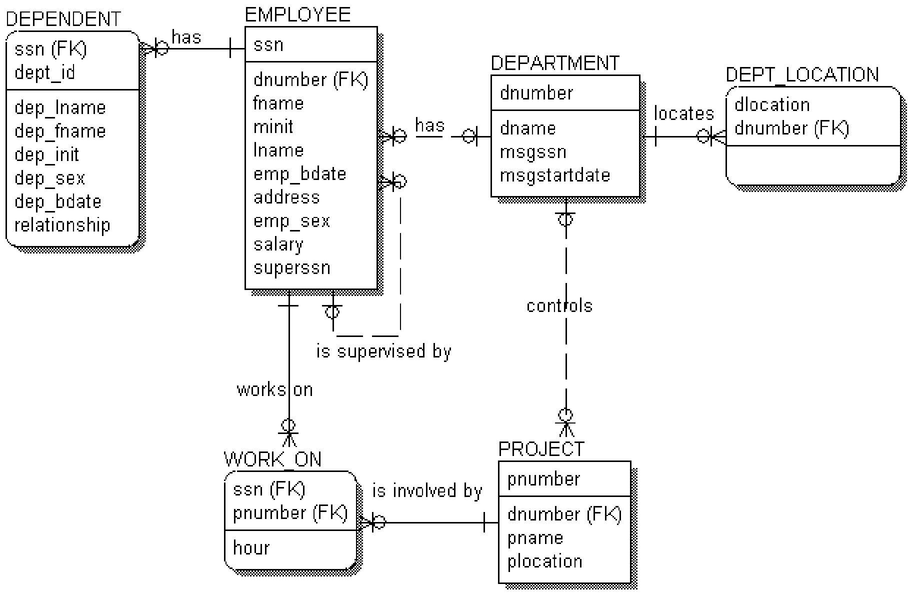


Figure 9.2 Result of mapping the COMPANY ER schema into a relational database schema.

Mapping COMPANY ER Schema Into A Relational Database Schema (cont.)

- IE notation is supported by DB design tools.

ER Diagram for the COMPANY



Mapping A Ternary Relationship Schema

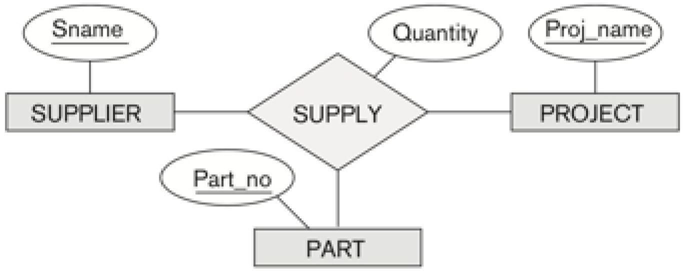


Figure 3.17 Ternary relationship types.
(a) The SUPPLY relationship.

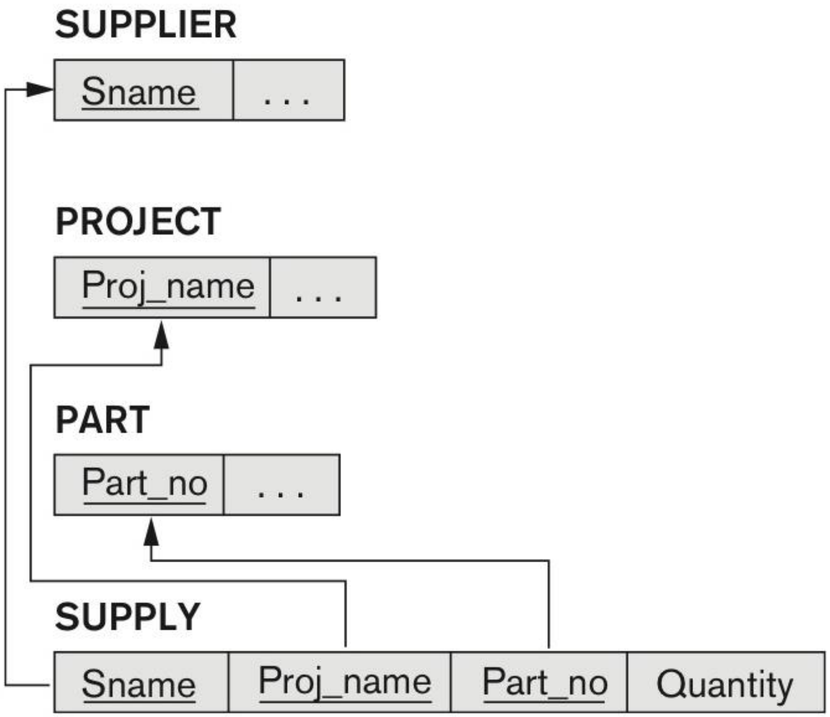


Figure 9.4 Mapping the n-ary relationship type SUPPLY from Figure 3.17(a).

Johns Hopkins Engineering

Principles of Database Systems

Module 8 / Lecture 5

ER and EER to Relational Mapping

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

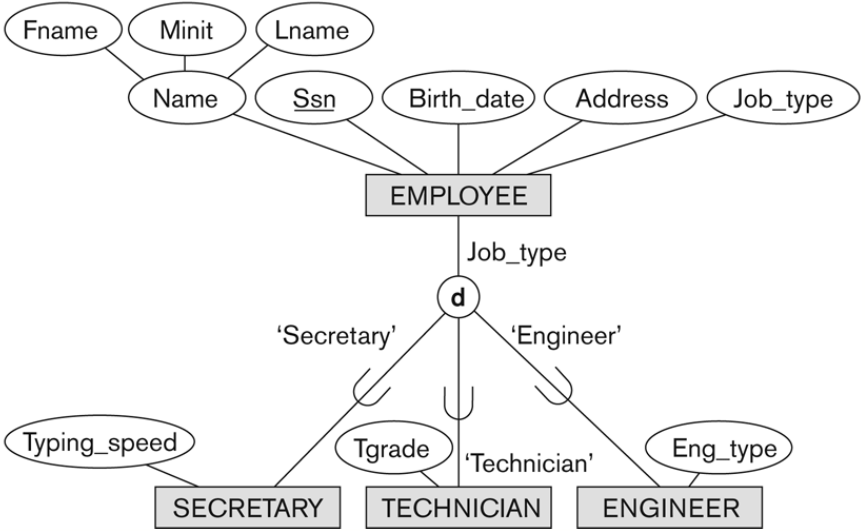
Mapping EER Model Concepts to Relations

- Convert Superclass and Subclass Relationships
  - Option 8A: Create the superclass relation and all subclass relations first, then migrate the PK from the superclass relation into each subclass relation as 1:1 relationships.
  - Multiple-relation option – superclass and subclasses

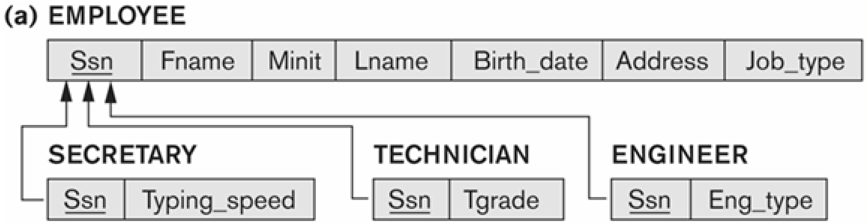
  Example EER schema in Figure 4.4

Mapping EER Model Concepts to Relations (cont.)

Figure 4.4
EER diagram notation for an attribute-defined specialization on Job_type.



Option 8A



Mapping EER Model Concepts to Relations (cont.)

- Convert Superclass and Subclass Relationships
  - Option 8B: Do not create a superclass relation, and create all subclass relations with all attributes from the superclass relation.
  - Multiple-relation option –subclass relations only

  Example:

  Mapping the EER schema in Figure 4.4 (b)

  EMPLOYEE → SECRETARY, TECHNICIAN, ENGINEER

  Mapping the EER schema in Figure 4.3 (b)

Mapping EER Model Concepts to Relations (cont.)

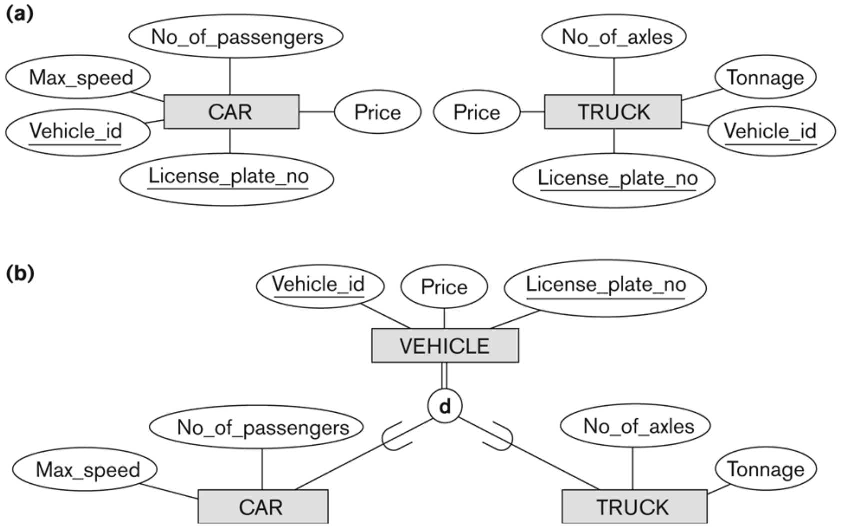


Figure 4.3
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

(b) CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

Option 8B

Mapping EER Model Concepts to Relations (cont.)

- Convert Superclass and Subclass Relationships
  - Option 8C: Create a single relation that combines (union) all attributes from all subclass relations with one type attribute (t). This approach is for a specialization whose subclasses are *disjoint*, and *t* is a type attribute to indicate what the tuple belongs to. Many null values will be created.
  - Single-relation option with one type attribute Example: Figure 4.4

Type flag

(c) EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

Mapping EER Model Concepts to Relations (cont.)

- Convert Superclass and Subclass Relationships
  - Option 8D: Create a single relation that combines (union) all attributes from the superclass and all subclass relations with a set (array) of type Boolean flags to indicate whether the tuple includes/belongs to the types. This approach is for a specialization whose subclasses are *overlapping*.
  - Single-relation option with multiple type attributes Example: Mapping Figure 4.5 using option 8D with Boolean type fields Mflag and Pflag.

Mapping EER Model Concepts to Relations (cont.)

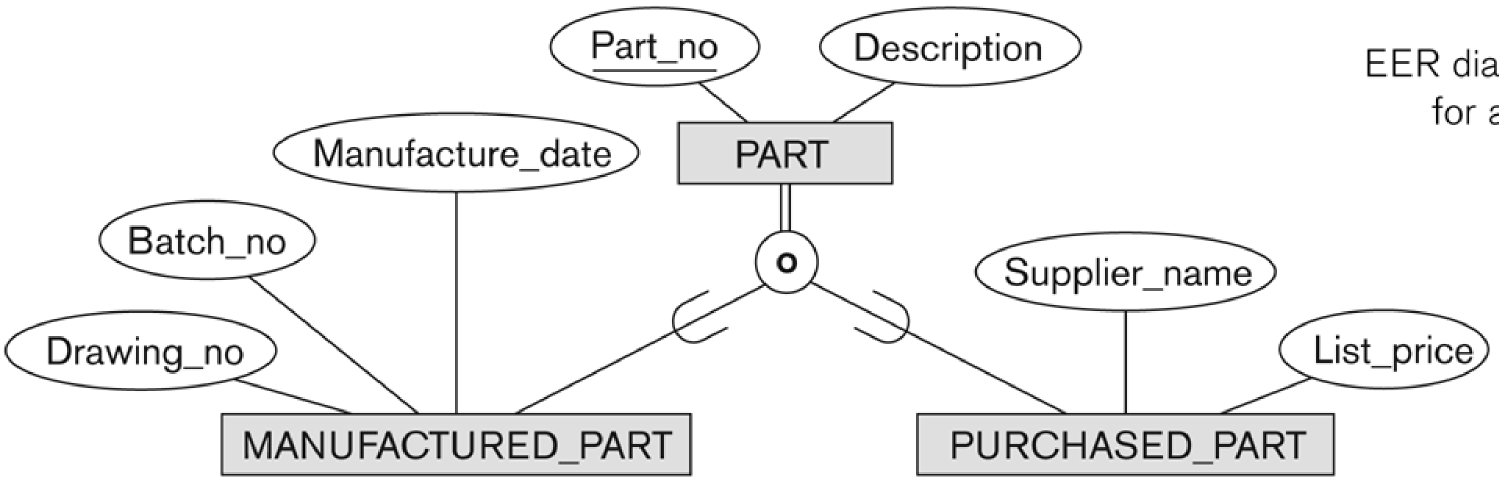


Figure 4.5
EER diagram notation for an overlapping (nondisjoint) specialization.

(d) PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

Option 8D

Boolean flags

Mapping Shared Subclasses

- A shared class is a subclass of several superclasses indicating multiple inheritance (specialization lattice)
- The classes must all have the same key attribute
- Mapping Figure 4.6

Mapping Shared Subclasses (cont.)

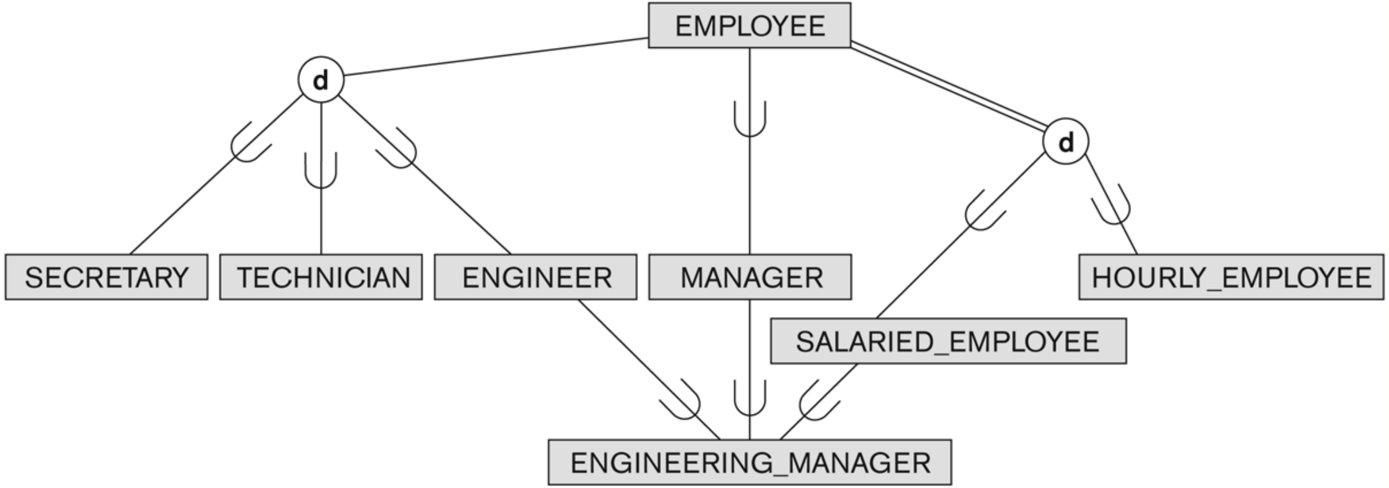


Figure 4.6
A specialization lattice with shared subclass ENGINEERING_MANAGER.

How do you map the EER specialization lattice?

Multiple-relation options (8A or 8B) in slides 15, 17

Single-relation options (8C or 8D) in slides 18, 20

Which one is a better choice and why? Open for Discussions

Mapping of Categories

- Is a subclass of the union of two or more superclasses that can have different entity types

- Can use a surrogate key

  Example: Mapping the EER categories (union types) in Figure 4.8 to relations.

Mapping of Categories (cont.)

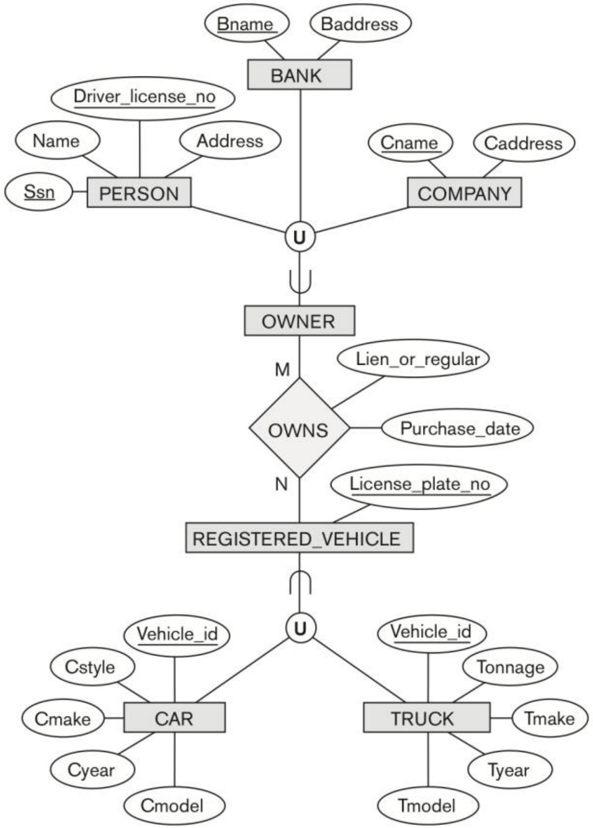


Figure 4.8 Two categories (union types): OWNER and REGISTERED_VEHICLE.

Mapping of Categories (cont.)

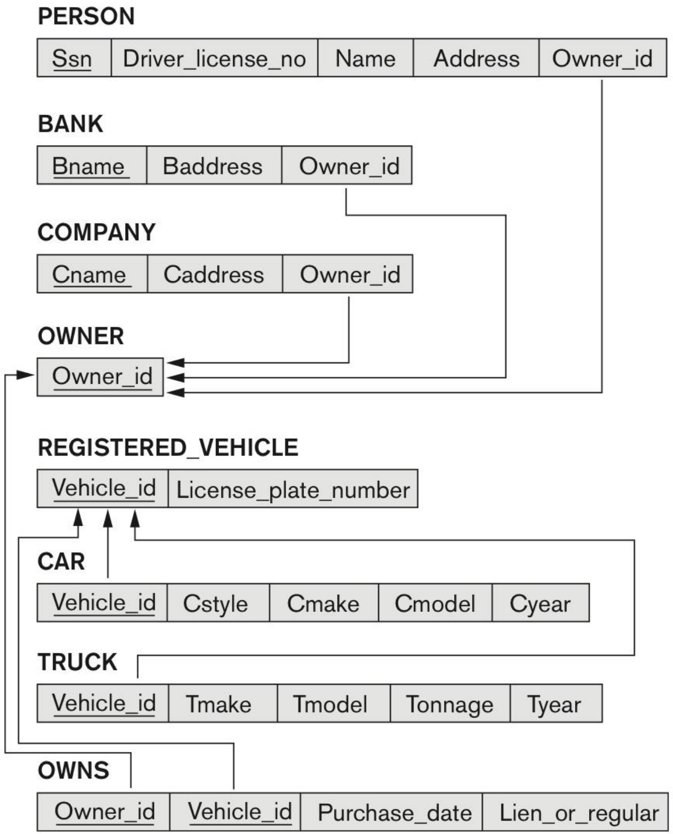


Figure 9.7 Mapping the EER categories (union types) in Figure 4.8 to relations.

Challenges on EER-to-Relation Mapping

- Can have multiple options available for specialization and generalization
  - Create more relations (tables) with multiple-relation options
  - Create fewer relations (tables) with single-relation options

- Consider implementation complications and performance when considering EER-to-Relation mapping

EER-to-Relation Mapping using ERwin

- Use supertype and subtype instead of superclass and subclass
- Create an identifying relationship between a supertype entity and its subtype entities
- Apply a transform to create an identifying relationship between a supertype entity and its subtype entities
  - Create a simple model
  - Improve query performance
  - Simplify application development and maintenance

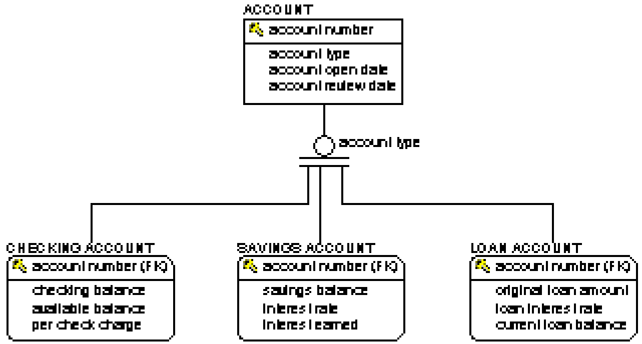
EER-to-Relation Mapping using ERwin (cont.)

Example:

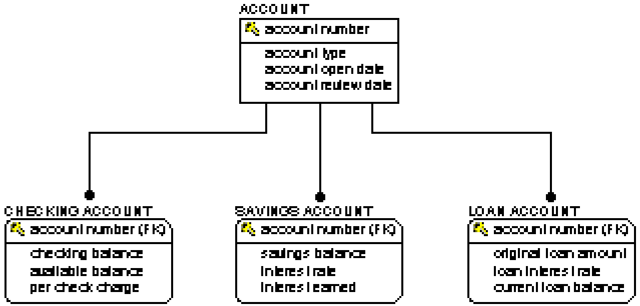
Bank ERD with various account types:
Checking, Saving and Loan

Mapping multiple relations with supertype and subtypes

Comments on this design



Before Applying the Transform



After Applying the Transform

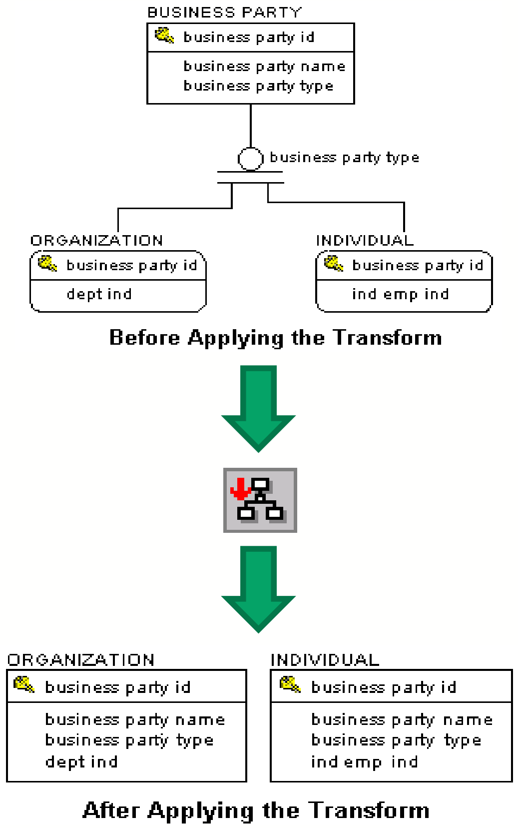
EER-to-Relation Mapping using ERwin (cont.)

Example:

Business Party ERD with Organization and individual types

Mapping to multiple relations with a result of “Rolling Down” two subtype relations

Comments on this design



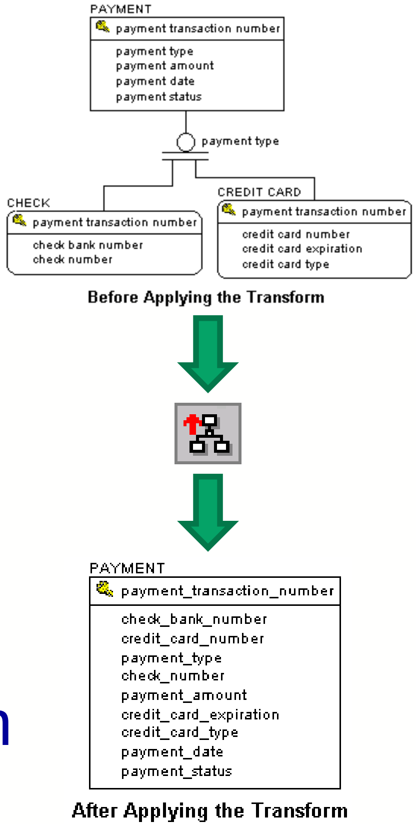
EER-to-Relation Mapping using ERwin (cont.)

Example:

Payment ERD with Check and Credit Card types

Mapping to multiple relations with a result of “Rolling up” a supertype relation

Comments on this design



Relational Mapping Considerations

- Normalized and Compact Design