Databases

Lecture 6 - Design Guideline

Emily Lucia Antosch

HAW Hamburg

07.04.2025

Contents

1. Introduction

1.1 Where are we right now?

- Last time, we looked at constraints and notations.
- Today, we'll be discussing
  - what guidelines exists that allow you to design databases
  - how you can use a cookbook to always create great data storage in databases
  - what normalization is.

1.1 Where are we right now?

1. Introduction
2. Basics
3. SQL
4. Entity-Relationship-Model
5. Relationships
6. Constraints
7. **Notation & Guideline**
8. More SQL
9. Subqueries & Views
10. Transactions
11. Database Applications
12. Integrity, Trigger & Security

1.2 What is the goal of this chapter?

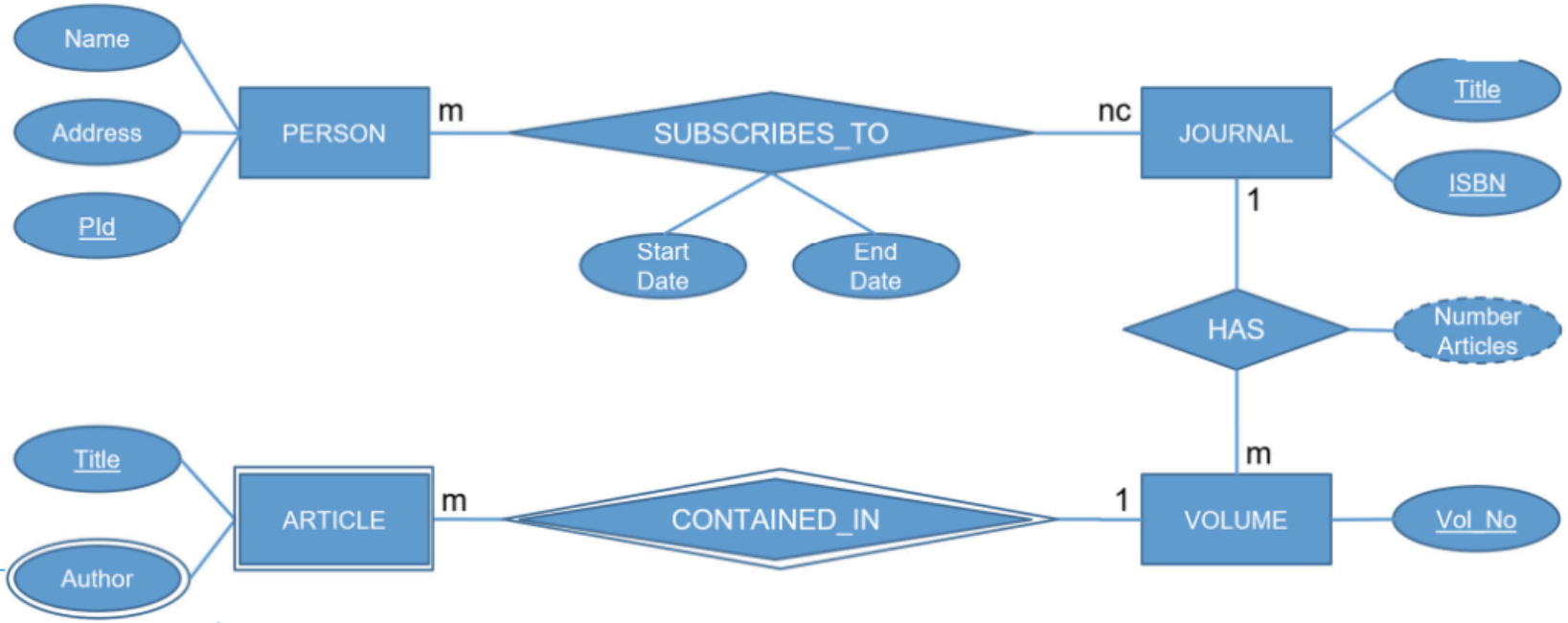
- At the end of this lesson, you will be able to
  - create databases based on guidelines
  - use normalization to design tables and relations.

2. Relationships

2.1 Assignment Journal

Convert the ERD to a RM

2.1 Assignment Journal



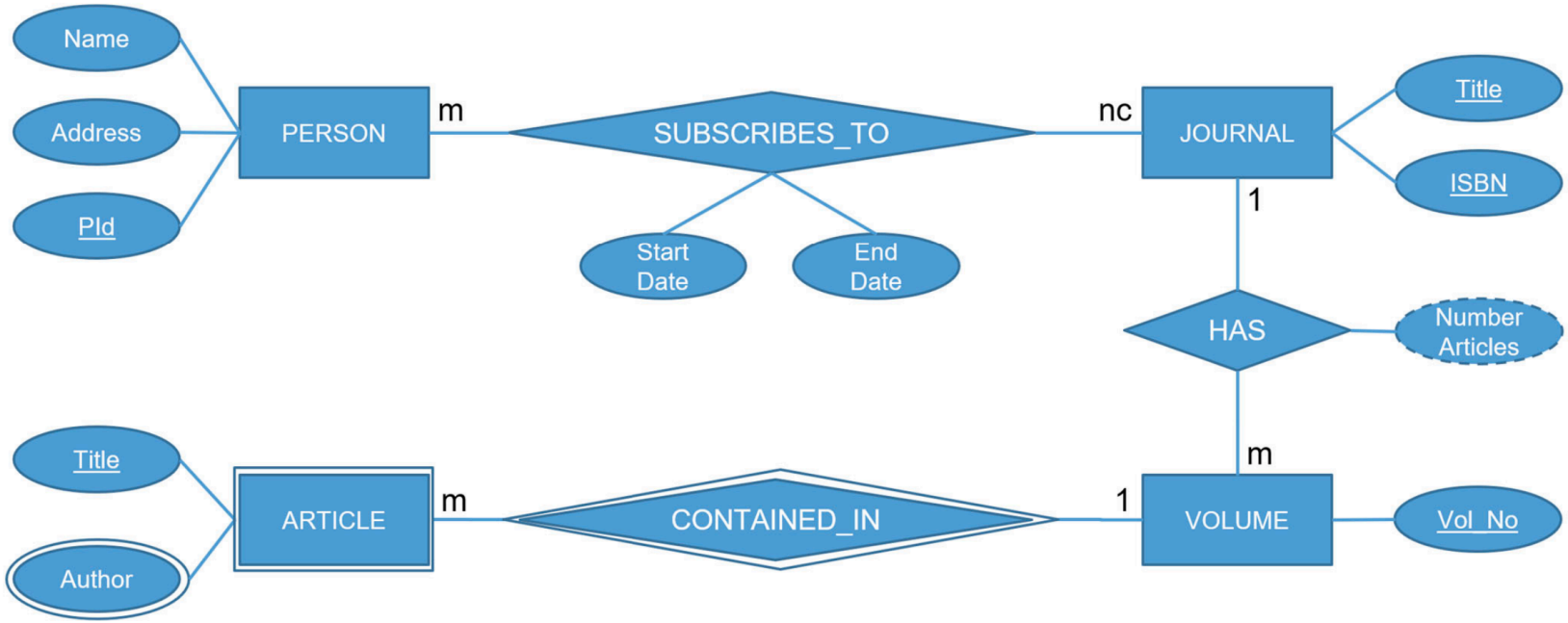
Seven Steps

1. Mapping of regular entity types
2. Mapping of weak entity types
3. Mapping of binary 1:1 relationships
4. Mapping of binary 1:n relationships
5. Mapping of binary m:n relationships
6. Mapping of multivalued attributes
7. Mapping of n-ary relationships

2.1 Assignment Journal

Implement RM with SQL

2.1 Assignment Journal



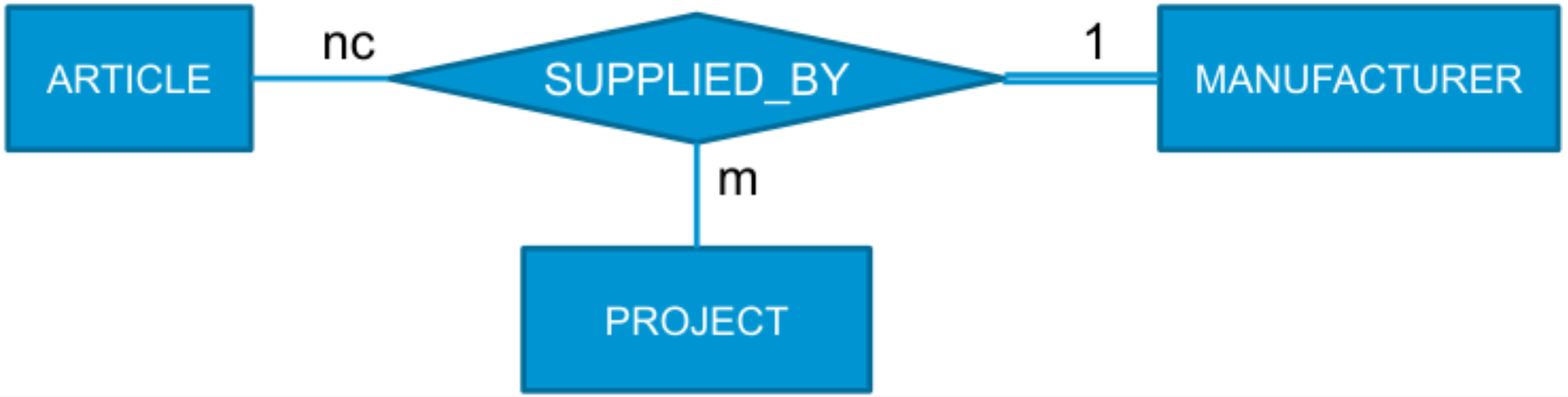

```
[CONSTRAINT [symbol]] FOREIGN KEY
    [index_name] (col_name, ...)
    REFERENCES tbl_name (col_name,...)
    [ON DELETE reference_option]
    [ON UPDATE reference_option]

reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

MC Notation - Tenary Relationship

Example

- Manufacturers supply items for projects.
- A manufacturer must supply at least one item.
- An article from in-house production does not have to be supplied for a project but can be supplied for many projects.
- A project uses at least one item.
- An item is supplied by only one manufacturer for a project.


ARTICLE
nc
SUPPLIED_BY
1
MANUFACTURER
m
PROJECT

MC Notation - Tenary Relationship

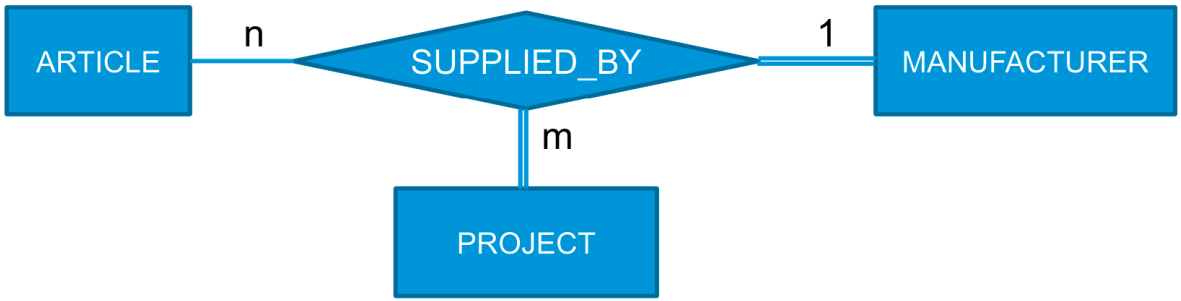
? Question

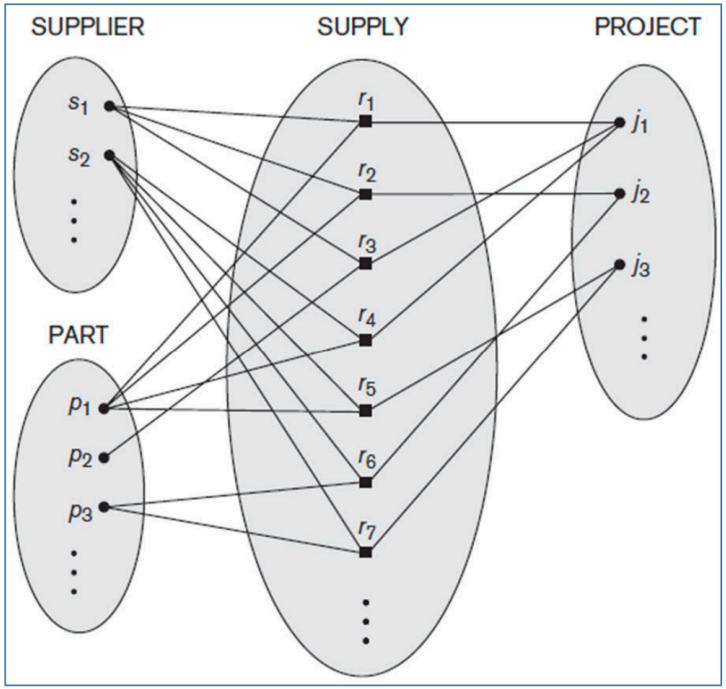
Cardinality: Can an entity of entity type A and an entity of entity type B be related to multiple entities of entity type C?

? Question

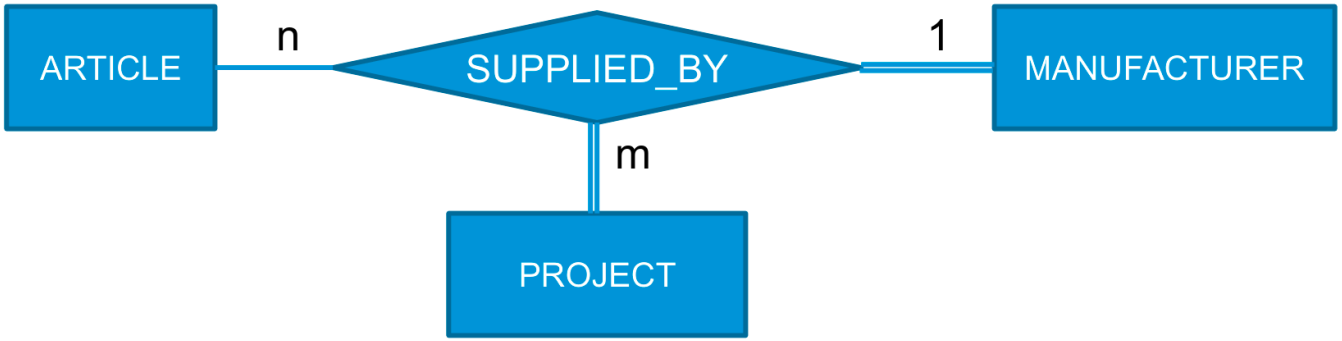
Participation: Must an entity type A be related to at least one entity type B and one entity type C?

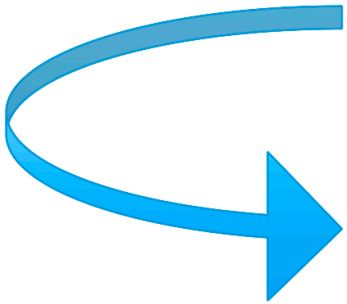
Tenary Relationship Types



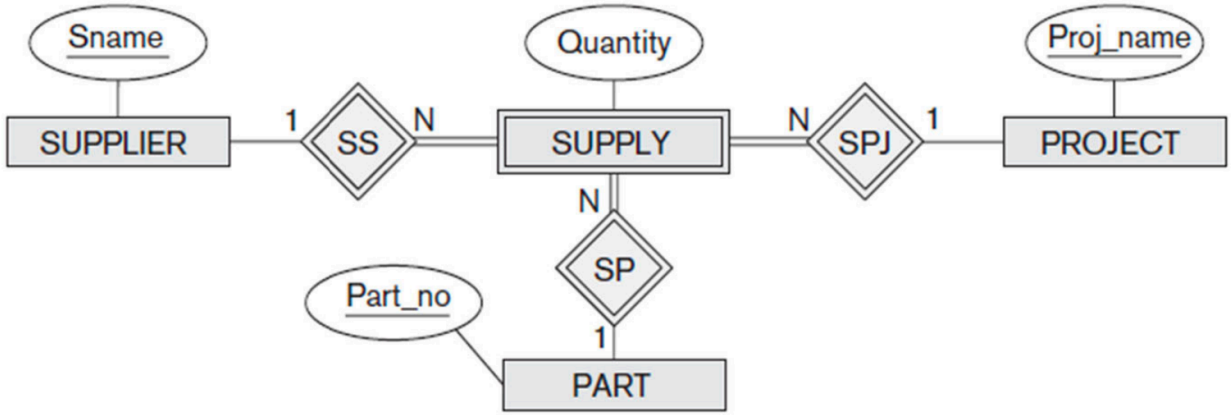


Tenary Relationship Types

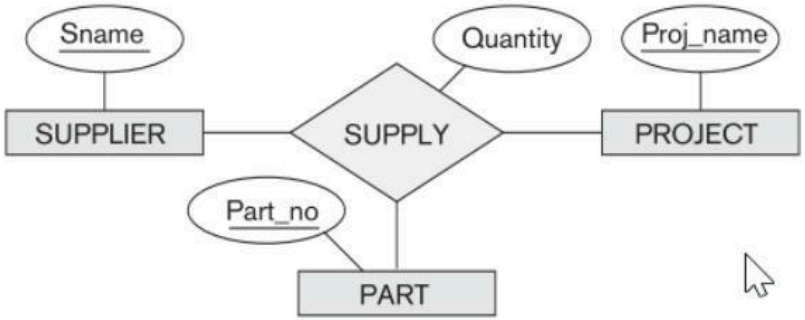


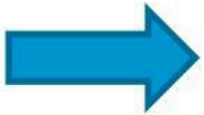


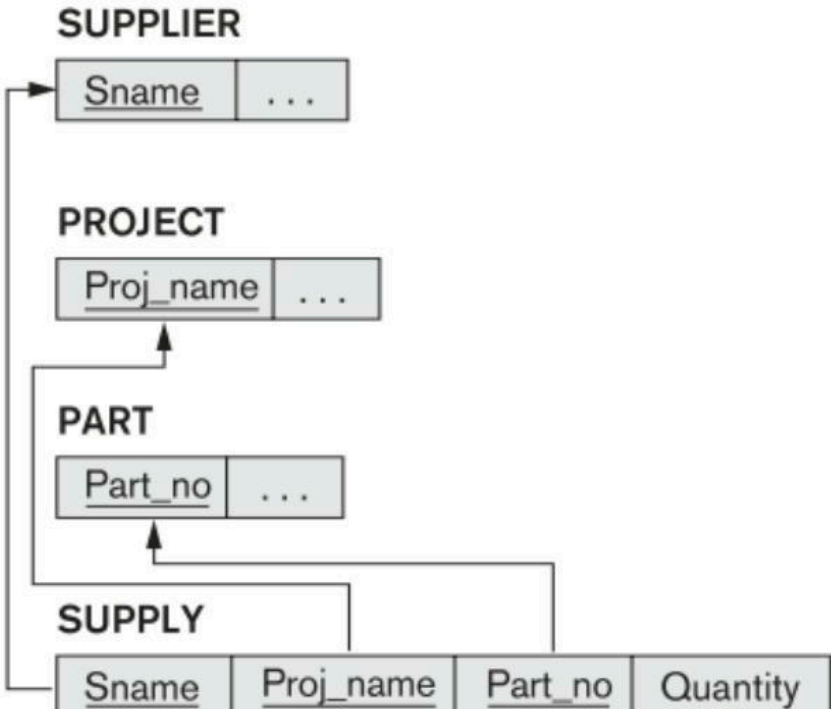
If you do only have binary relationship types



Mapping of N-ary Relationship Types

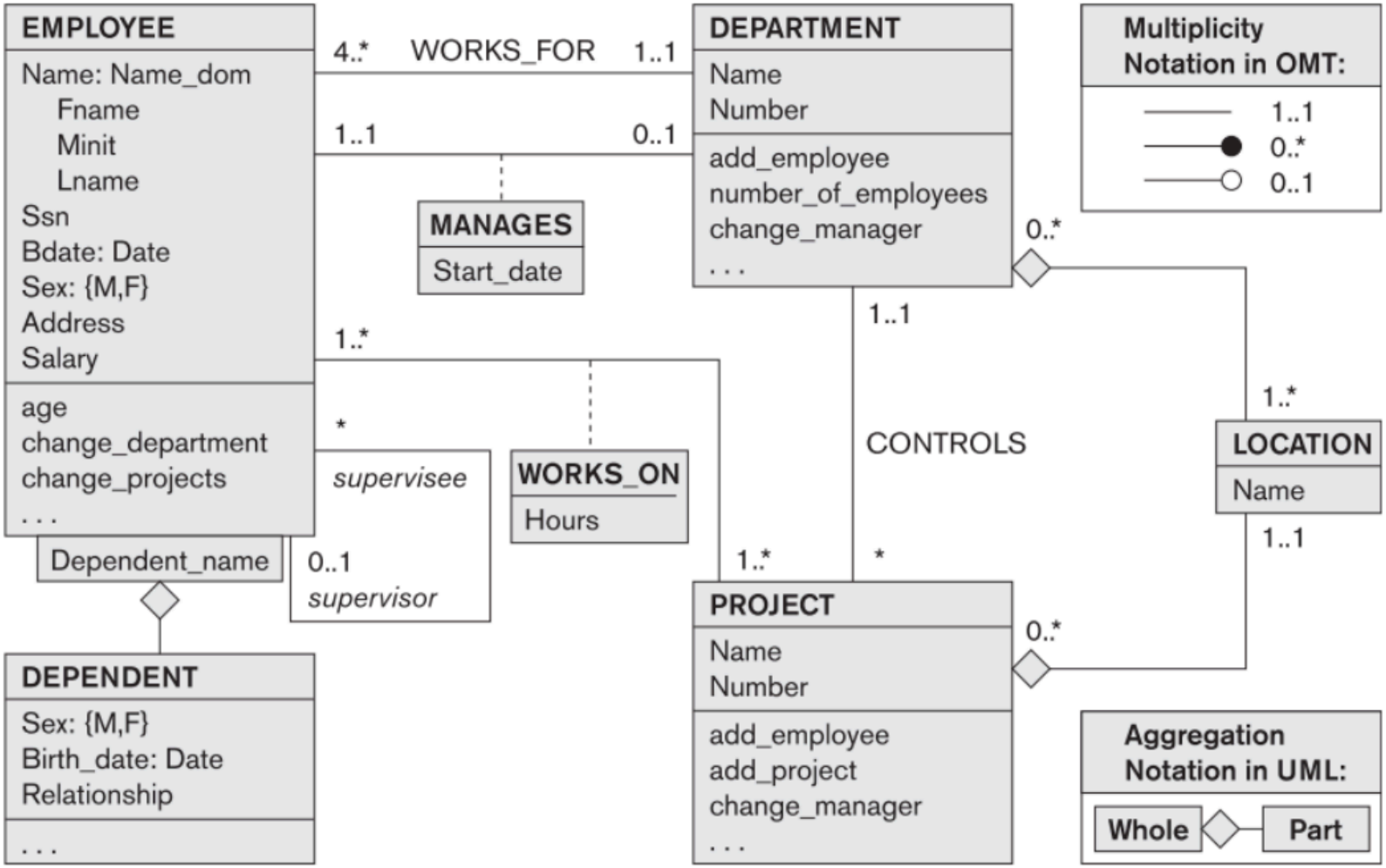






2.2 ERM

UML Notation


EMPLOYEE
Name: Name_dom
Fname
Minit
Lname
Ssn
Bdate: Date
Sex: {M,F}
Address
Salary
age
change_department
change_projects
. . .
Dependent_name
DEPENDENT
Sex: {M,F}
Birth_date: Date
Relationship
. . .
4..*
WORKS_FOR
1..1
1..1
0..1
MANAGES
Start_date
1..*
*
supervisee
0..1
supervisor
WORKS_ON
Hours
DEPARTMENT
Name
Number
add_employee
number_of_employees
change_manager
. . .
0..*
1..1
CONTROLS
1..*
*
PROJECT
Name
Number
add_employee
add_project
change_manager
. . .
1..*
LOCATION
Name
1..1
0..*
Multiplicity
Notation in OMT:
1..1
0..*
0..1
Aggregation
Notation in UML:
Whole
Part

Summary

ER Model	Relational Model
Entity type	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 attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary key

3. Constraints

3.1 Overview

Basics

- Three categories
  1. Constraints that are inherent in the data model **inherent model-based constraints** or **implicit constraints** Example: no duplicate tuples in a relation
  2. Constraints that can be directly expressed in schemas of the data model **schema-based constraints** or **explicit constraints** Example: Domain constraints, key constraints, constraints on NULL, entity integrity constraints and referential integrity constraints

3.1 Overview

3. Constraints that cannot be directly expressed in the schemas of the data model, and hence must be expressed and enforced by the application programs **application-based** or **semantic constraints** or **business rules**

3.1 Overview

Basics

Constraint	Number of affected Relations
Domain constraints	1
Constraints on `NULL`	1
Entity integrity constraints (primary key)	1
Referential integrity constraints	$\geqslant 1$
Semantic integrity constraints	$\geqslant 1$
Functional dependencies constraint	$\geqslant 1$

4. Notation & Guidelines

MC Notation

- Participation constraints
- Relationships can be mandatory or optional
- Types
  - Exactly one element: 1
  - One or no element: c (or 1c)
  - No or many elements: mc (or nc)
  - One or many elements: m (or n)

Info

Also called must-can notation!

MC Notation

- Every person owns no, one, or more bikes Every bike has exactly one owner.

- A person is married to no or exactly one person

4.1 ERM

MC Notation

- Binary 1:N Relationships
  - Chen Notation

  - MC Notation

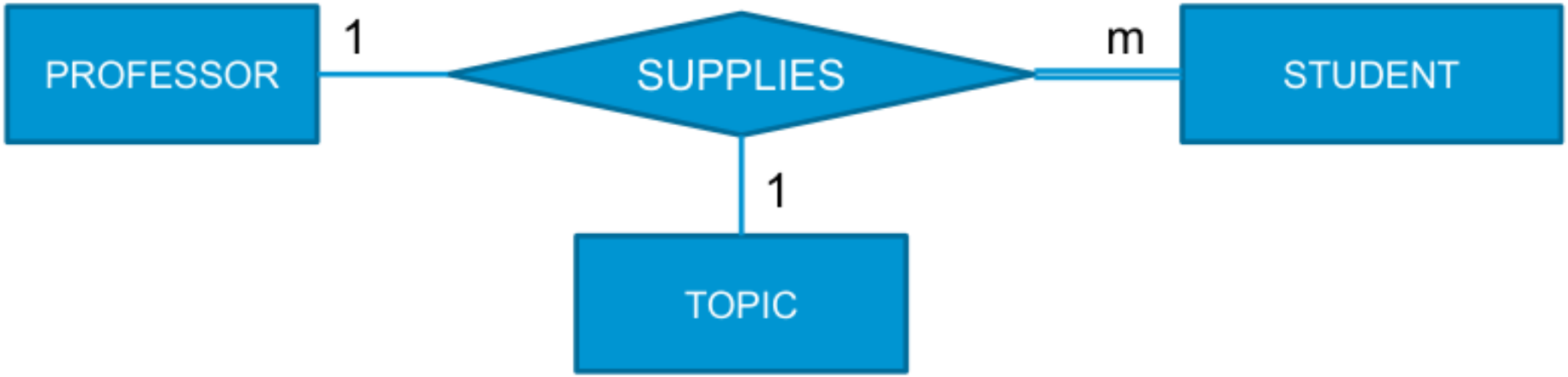
MC Notation

- CEO - Company ?:?
- Country - King ?:?
- University - Student ?:?
- Person - Car ?:?
- Student - Course ?:?
- Customer - Article ?:?
- Customer - Invoice ?:?

MC Notation

- To prevent students from concentrating on one professor, they may only work with one professor on one seminar topic.
- In addition, a student can only work on a seminar topic with one professor.
- However, a professor may assign a seminar topic more than once.
- Students must attend seminars, but seminar topics do not have to be chosen.

4.1 ERM



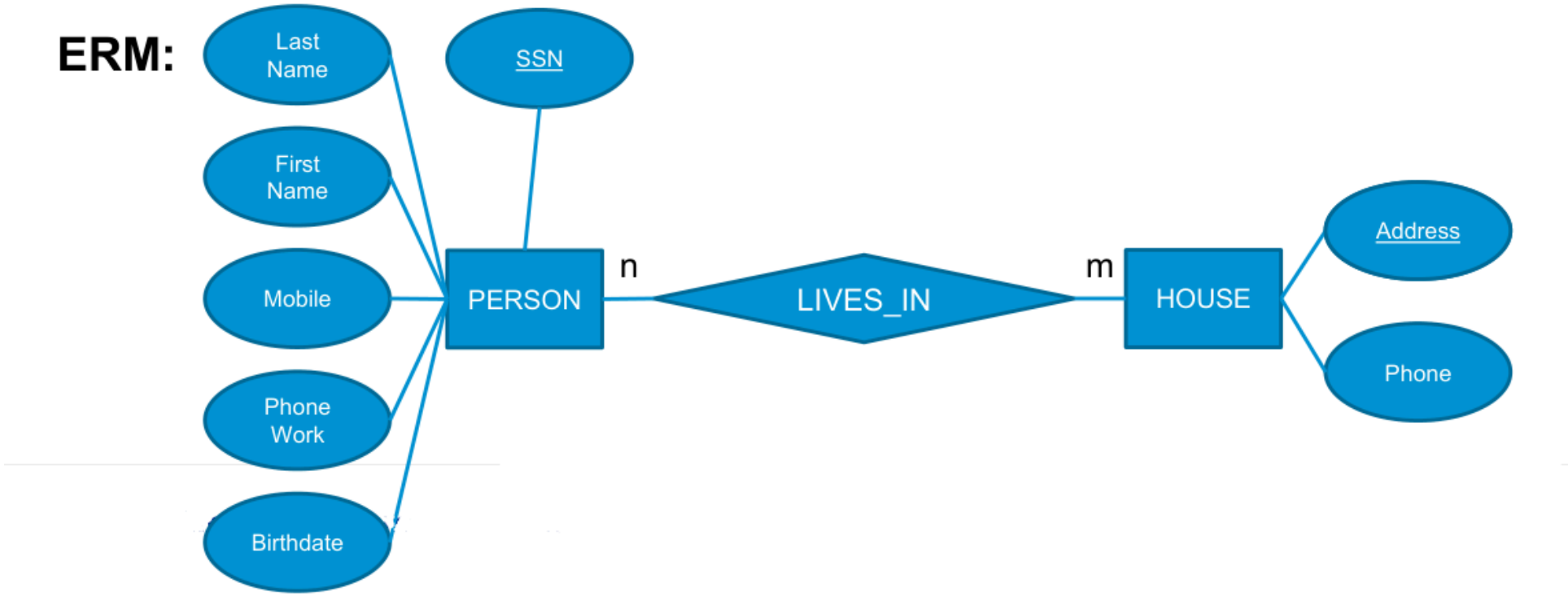
4.1 ERM

MC Notation

Comparison ERM and RM

- ERM:
  - Conceptual Database Design
  - Describes a collection of **entities**, also called as real-world **objects** and **relations** between those entities
  - Basic elements: **entity type**, **relationship type** and **attributes**
  - **Constraints** like **Cardinality**, **Participation ratio** and **Keys**

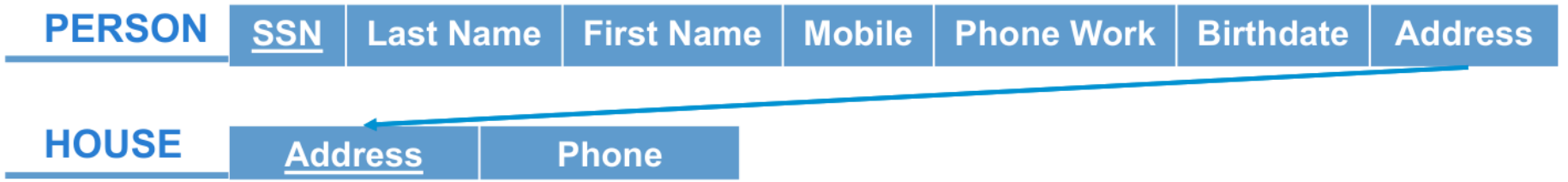
4.1 ERM


ERM:
Last Name
SSN
First Name
Mobile
Phone Work
Birthdate
PERSON
n
LIVES_IN
m
HOUSE
Address
Phone

Comparison ERM and RM

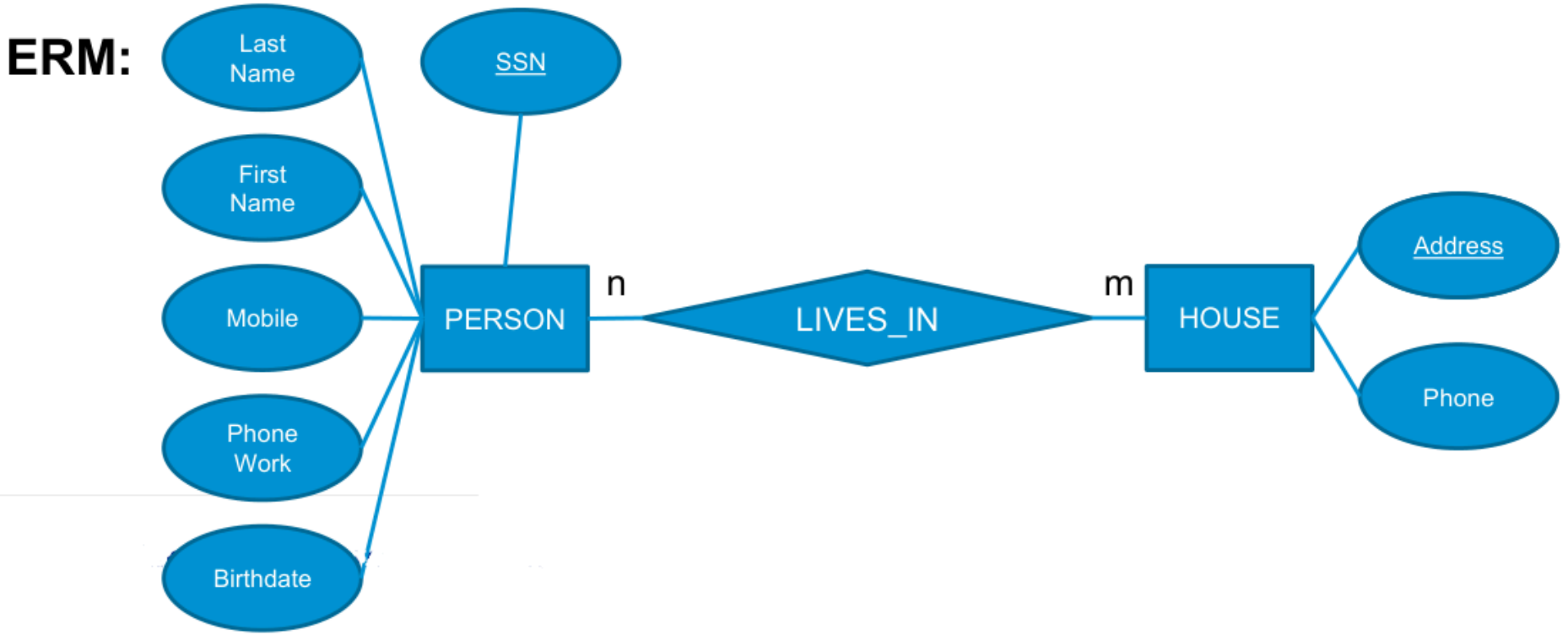
- Relational Model:
  - Logical Database Design
  - Describes data and relation among those data by tables
  - Basic elements: Relations and Attributes
  - Constraints: Domain constraints, key constraints, constraints on `NULL`, entity integrity constraints and referential integrity constraints

Relational Model

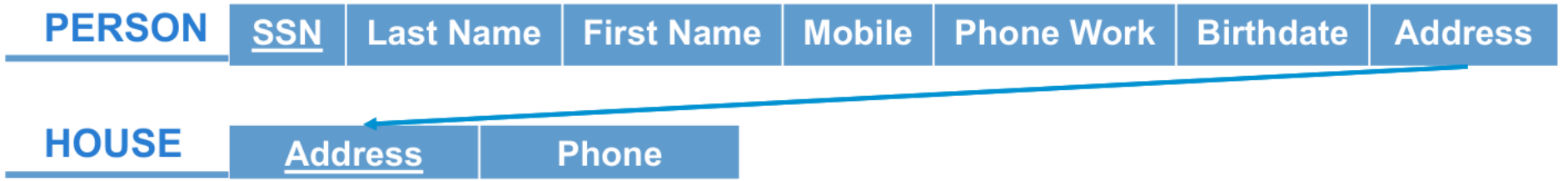


4.1 ERM

Comparison ERM and RM



Relational Model



Comparison ERM and RM

Aspect	ERM	RM
Basic	It represents the collection of objects called entities and relation between those entities	It represents the collection of tables and the relation between those tables
Describe	ERMs describe data as entity set, relationship set and attributes	Relational model describes data in a table as domains, attributes, tuples

Aspect	ERM	RM
Relationship	In an ERM, it is easier to understand the relationships between entities	Comparatively, it is less easy to derive a relation between tables in relational model
Mapping	ERM describes mapping cardinalities	Relational model does not describe mapping cardinalities

Mapping of ERM to RM

Main rules

- Entity types
  - Mapped to relations
  - Relations contain the attributes
  - Composite attributes: set of simple attributes
- Relationship Types
  - Foreign keys or
    - Relations plus Foreign keys

Mapping of ERM to RM

- Seven Steps
  1. Mapping of regular entity types
  2. Mapping of weak entity types
  3. Mapping of binary 1:1 relationships
  4. Mapping of binary 1:n relationships
  5. Mapping of binary m:n relationships
  6. Mapping of multivalued attributes
  7. Mapping of n-ary relationships

Design Guidelines

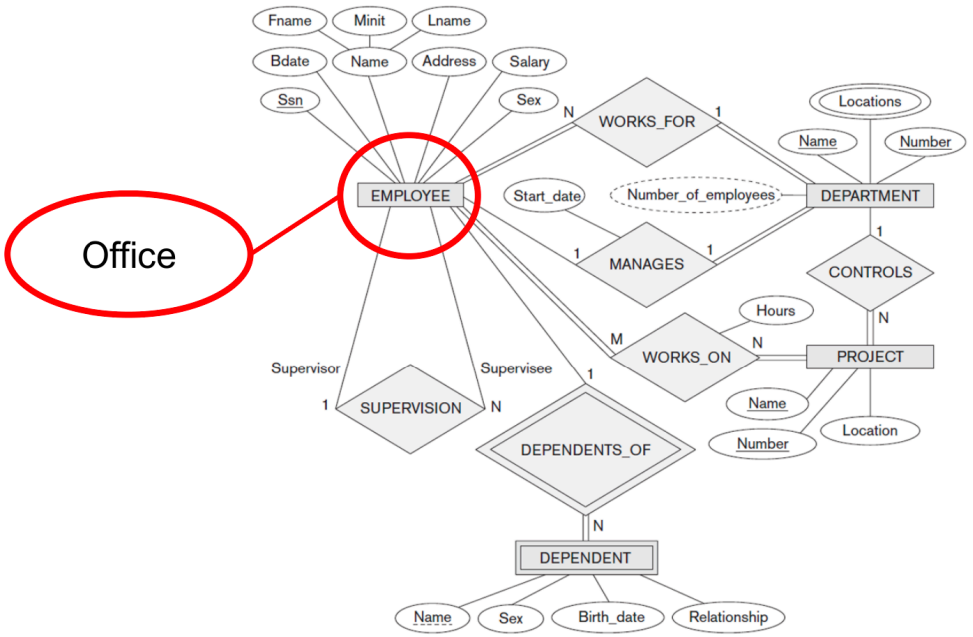
- Guideline 1
  - Clear semantics of relation attributes
  - Meaning of attributes should be easy to explain
  - Do not mix attributes from different entity types or relationship types into one relation

Design Guidelines

- Guideline 2
  - Avoid schemas that cause insertion, deletion, or modification anomalies
  - If anomalies are present, note them clearly!
  - Someone must take care of them (application, triggers)

Design Guideline

- Guideline 3
  - Avoid attributes whose values are frequently NULL
  - NULL values should be the exception
  - Attributes that are NULL frequently could be placed in separate relations (with the primary key): 1:c



Example

Attribute Office if only 15 percent of employees have individual offices

Design Guideline

- Guideline 4
  - Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations
  - Joining on such attributes may produce **spurious tuples**
  - Matching attributes in relations should be (foreign key, primary key) combinations Do not forget the Foreign Keys!

Design Guideline - Spurious Tuples

- Combining relations should produce facts only

Example

- Two relations
  - Employee
  - Project
- When joining these relations, we want only tuples for real existing combinations of employees and projects
- Other combinations: **Spurious tuples**

4.1 ERM

- Spurious tuples are created when two tables are joined on attributes that are neither primary keys nor foreign keys

Design Guidelines

Why don’t we put everything in one big table?

ALL

Professor	Lecture	Student	LiteracyTitle	ISBN	Semester	…
Smith	C++	John	Learn C++	12345	4	…
Smith	C++	Julie	Learn C++	12345	4	…
Collins	DBMS	Julie	Databases	23456	4	…
Collins	DBMS	Anna	Databases	23456	4	…
…	…	…	…	…	…	…

Design Guidelines

- One problem is: Some data is the always the same (E.g., ISBN)
  - This problem occurs in nearly all applications
  - There are almost one or several attributes, which define the values of the remaining attributes
  - This fact is called **functional dependency**
  - Functional dependency is relevant in practical applications

ALL

Professor	Lecture	Student	LiteracyTitle	ISBN	Semester	…
Smith	C++	John	Learn C++	12345	4	…
Smith	C++	Julie	Learn C++	12345	4	…
Collins	DBMS	Julie	Databases	23456	4	…
Collins	DBMS	Anna	Databases	23456	4	…
…	…	…	…	…	…	…

Design Guidelines - Functional Dependencies

- Definition: There is a relational schema called R, and X and Y are any attribute sets of the attributes of R. Then Y is functional dependent on X, noted as $X \longrightarrow Y$, if and only if every value of X in R defines exactly one value of Y in R
- In other words: Whenever two tuples of relation R match in their X values, then the Y values of these tuples match as well

Y X

ALL	Professor	Lecture	Student	LiteracyTitle	ISBN	Semester	…
	Smith	C++	John	Learn C++	12345	4	…
	Smith	C++	Julie	Learn C++	12345	4	…
	Collins	DBMS	Julie	Databases	23456	4	…
	Collins	DBMS	Anna	Databases	23456	4	…
	…	…	…	…	…	…	…

Design Guidelines - Functional Dependencies

- Are used to specify formal measures of the “goodness” of relational designs
- Can help to identify redundancy and suggest refinements
- FD: If for 2 tuples X is the same, then Y must also be the same
- Notation: X $\longrightarrow$ Y Read: “X determines Y”
- Generalization of keys
- A key determines all attributes: ൭◌൩

Design Guidelines - Functional Dependencies

- Actual data can help to identify FDs
- Note: FD is a statement about **all allowable** tuples!
- Based on semantics, NOT instances!
- Full functional dependency:
  - $X \rightarrow Y$ holds only for complete X, not for subset of X
  - You cannot remove an element of X without destroying the FD
  - Opposite: Partial Functional Dependency

Normalization

- Normal Forms provide quality statements on relations
- The process of decomposing unsatisfactory “bad” relations by breaking up their attributes into smaller relations
- There are different Normal Forms: 1NF, 2NF, 3NF, BCNF, …

Normalization

Title	Author	Author Nationality	Format	Price	Subject	Pages	Thickness	Publisher	Publisher Country	Publication Type	Genre ID	Genre Name
Beginning MySQL Database Design and Optimization	Chad Russell	American	Hardcover	49.99	MySQL, Database, Design	520	Thick	Apress	USA	E-book	1	Tutorial

Normalization - 1NF

- The **First Normal Form** (1NF) defines that the values in each column of a table must be atomic
- Solution: Separate the duplicities into multiple columns using repeating groups ‘subject’

Title	Author	Author Nationality	Format	Price	Subject	Pages	Thickness	Publisher	Publisher Country	Publication Type	Genre ID	Genre Name
Beginning MySQL Database Design and Optimization	Chad Russell	American	Hardcover	49.99	MySQL, Database, Design	520	Thick	Apress	USA	E-book	1	Tutorial

Title	Format	Author	Author Nationality	Price	Subject 1	Subject 2	Subject 3	Pages	Thickness	Publisher	Publisher country	Genre ID	Genre Name
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	MySQL	Database	Design	520	Thick	Apress	USA	1	Tutorial

? Question

What are the disadvantages?

Normalization - 1NF

- Alternative Solution:

Title	Author	Author Nationality	Format	Price	Subject	Pages	Thickness	Publisher	Publisher Country	Publication Type	Genre ID	Genre Name
Beginning MySQL Database Design and Optimization	Chad Russell	American	Hardcover	49.99	MySQL, Database, Design	520	Thick	Apress	USA	E-book	1	Tutorial

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*

Subject

Subject ID	Subject name
1	MySQL
2	Database
3	Design

Title - Subject

Title	*Subject ID*
Beginning MySQL Database Design and Optimization	1
Beginning MySQL Database Design and Optimization	2
Beginning MySQL Database Design and Optimization	3

Normalization - 1NF

- Alternative Solution:

Title	Author	Author Nationality	Format	Price	Subject	Pages	Thickness	Publisher	Publisher Country	Publication Type	Genre ID	Genre Name
Beginning MySQL Database Design and Optimization	Chad Russell	American	Hardcover	49.99	MySQL, Database, Design	520	Thick	Apress	USA	E-book	1	Tutorial

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*

Subject

Subject ID	Subject name
1	MySQL
2	Database
3	Design

Title - Subject

Title	*Subject ID*
Beginning MySQL Database Design and Optimization	1
Beginning MySQL Database Design and Optimization	2
Beginning MySQL Database Design and Optimization	3

Normalization - 1NF

- Decomposition to 1NF:
  - Composite Attributes
    - Split into single atomic attributes
  - Multi-valued Attributes
    - Decompose to new tuples
      - Results in redundancies
      - Further decomposition in following steps
- or decompose to new relation with FK

Normalization - 2NF

- A relation is in **Second Normal Form** (2NF), if
  - It is in 1NF and
  - Every attribute in R is fully functional dependent on every key in R (or is part of the key)
- In other words: If there is a key with different attributes (e.g., title and format) and an attribute depends on just a part of this key (e.g., title), then the second Normal Form is violated
- To create the Second Normal Form, you must decompose the relation: Depending attribute with part of key in new relation, delete depending attribute in old relation

Normalization - 2NF

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*
Beginning MySQL Database Design and Optimization	E-book	Chad Russell	American	22.34	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E-book	E.F.Codd	British	13.88	538	Thick	2	Popular science	*2*
The Relational Model for Database Management: Version 2	Paperback	E.F.Codd	British	39.99	538	Thick	2	Popular science	*2*

- Primary key: Title + Format
- Functional dependencies:
  - Title → Author
  - Title → Author Nationality
  - Title, Format → Price
  - Title → Pages

4.1 ERM

- Title ⟶ Thickness
- Title ⟶ GenreID
- Title ⟶ GenreName
- Title ⟶ PublisherID

Normalization - 2NF

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*
Beginning MySQL Database Design and Optimization	E-book	Chad Russell	American	22.34	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E-book	E.F.Codd	British	13.88	538	Thick	2	Popular science	*2*
The Relational Model for Database Management: Version 2	Paperback	E.F.Codd	British	39.99	538	Thick	2	Popular science	*2*

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*

Format - Prices

Title	Format	Price
Beginning MySQL Database Design and Optimization	Hardcover	49.99
Beginning MySQL Database Design and Optimization	E-book	22.34
The Relational Model for Database Management: Version 2	E-book	13.88
The Relational Model for Database Management: Version 2	Paperback	39.99

Normalization - 2NF

- Decomposition to Second Normal Form
  - Has the relation in 1NF following design (keyPart1, keyPart2, noKey1, noKey2) and there is a functional dependency FD: keyPart2 $\longrightarrow$ noKey2
  - Then the decomposition respecting this FD results in following schema: (keyPart1, keyPart2, noKey1) (keyPart2, noKey2)

Normalization - 2NF

- Decomposition to Second Normal Form
  - Has the relation in 1NF following design (keyPart1, keyPart2, noKey1, noKey2) and there is a functional dependency FD: keyPart2 → noKey2

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*
Beginning MySQL Database Design and Optimization	E-book	Chad Russell	American	22.34	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E-book	E.F.Codd	British	13.88	538	Thick	2	Popular science	*2*
The Relational Model for Database Management: Version 2	Paperback	E.F.Codd	British	39.99	538	Thick	2	Popular science	*2*

Normalization - 2NF

- Decomposition to Second Normal Form
  - Then the decomposition respecting this FD results in following schema: (keyPart1, keyPart2, noKey1) (keyPart2, noKey2)

Book

Title	Format	Author	Author Nationality	Price	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Hardcover	Chad Russell	American	49.99	520	Thick	1	Tutorial	*1*
Beginning MySQL Database Design and Optimization	E-book	Chad Russell	American	22.34	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E-book	E.F.Codd	British	13.88	538	Thick	2	Popular science	*2*
The Relational Model for Database Management: Version 2	Paperback	E.F.Codd	British	39.99	538	Thick	2	Popular science	*2*

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*

Format - Prices

Title	Format	Price
Beginning MySQL Database Design and Optimization	Hardcover	49.99
Beginning MySQL Database Design and Optimization	E-book	22.34
The Relational Model for Database Management: Version 2	E-book	13.88
The Relational Model for Database Management: Version 2	Paperback	39.99

Normalization - 3NF

- The **Third Normal Form** (3NF) describes the problem
  - Informal: A relation is in 3NF if every tuple consists of a primary key and a set of other attributes that are independent of each other
  - Formal: A relation is in 3NF if
    - It is in 2NF and
    - Every non-primary-key attribute is directly dependent on the primary key (especially no transitive dependencies)
  - In other words: A table in third normal form (3NF) is a table in 2NF that has no transitive dependencies

Normalization - 3NF

Transitive Dependency

- Z is transitive dependent on X, if
  - $X \longrightarrow Y \longrightarrow Z$
  - But not $Y \longrightarrow X$
- Second condition important: not a transitive dependency if X and Y are both keys!

Normalization - 3NF

Transitive Dependency

- Z is transitive dependent on X, if
  - $X \rightarrow Y \rightarrow Z$
  - But not $Y \rightarrow X$
- Second condition important: not a transitive dependency if X and Y are both keys!

Normalization - 3NF

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*
Learning SQL	Alan Beaulieu	American	338	Slim	1	Tutorial	*3*
SQL Cookbook	Anthony Molinaro	American	636	Thick	1	Tutorial	*3*

- Functional dependencies:
  - Title → Author
  - Title → Author Nationality → Author
  - Title, Format → Price
  - Title → Pages
  - Title → Thickness

4.1 ERM

- Title $\longrightarrow$ GenreID
- Title $\longrightarrow$ GenreName $\longrightarrow$ GenreID
- Title $\longrightarrow$ PublisherID

Normalization - 3NF

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*
Learning SQL	Alan Beaulieu	American	338	Slim	1	Tutorial	*3*
SQL Cookbook	Anthony Molinaro	American	636	Thick	1	Tutorial	*3*

Book

Title	Author	Author Nationality	Pages	Thickness	*Genre ID*	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	*1*	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	*2*	*2*
Learning SQL	Alan Beaulieu	American	338	Slim	*1*	*3*
SQL Cookbook	Anthony Molinaro	American	636	Thick	*1*	*3*

Book Genres

Genre ID	Genre Name
1	Tutorial
2	Popular science

Normalization - 3NF

- Decomposition to Third Normal Form
  - The relation in 2NF has the schema (KeyPart1, NonKey1, NonKey2) and there is a functional dependency FD: NonKey1 $\longrightarrow$ NonKey2
  - Then the decomposition results in a new schema, which is in Third Normal Form (KeyPart1, NonKey1) (NonKey1, NonKey2)

Normalization - 3NF

- Decomposition to Third Normal Form
  - The relation in 2NF has the schema (KeyPart1, NonKey1, NonKey2) and there is a functional dependency FD: NonKey1 ⟶ NonKey2

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*
Learning SQL	Alan Beaulieu	American	338	Slim	1	Tutorial	*3*
SQL Cookbook	Anthony Molinaro	American	636	Thick	1	Tutorial	*3*

Normalization - 3NF

- Decomposition to Third Normal Form
  - Then the decomposition results in a new schema, which is in Third Normal Form (KeyPart1, NonKey1) (NonKey1, NonKey2)

Book

Title	Author	Author Nationality	Pages	Thickness	Genre ID	Genre Name	*Publisher ID*
Beginning MySQL Database Design and Optimization	Chad Russell	American	520	Thick	1	Tutorial	*1*
The Relational Model for Database Management: Version 2	E.F.Codd	British	538	Thick	2	Popular science	*2*
Learning SQL	Alan Beaulieu	American	338	Slim	1	Tutorial	*3*
SQL Cookbook	Anthony Molinaro	American	636	Thick	1	Tutorial	*3*

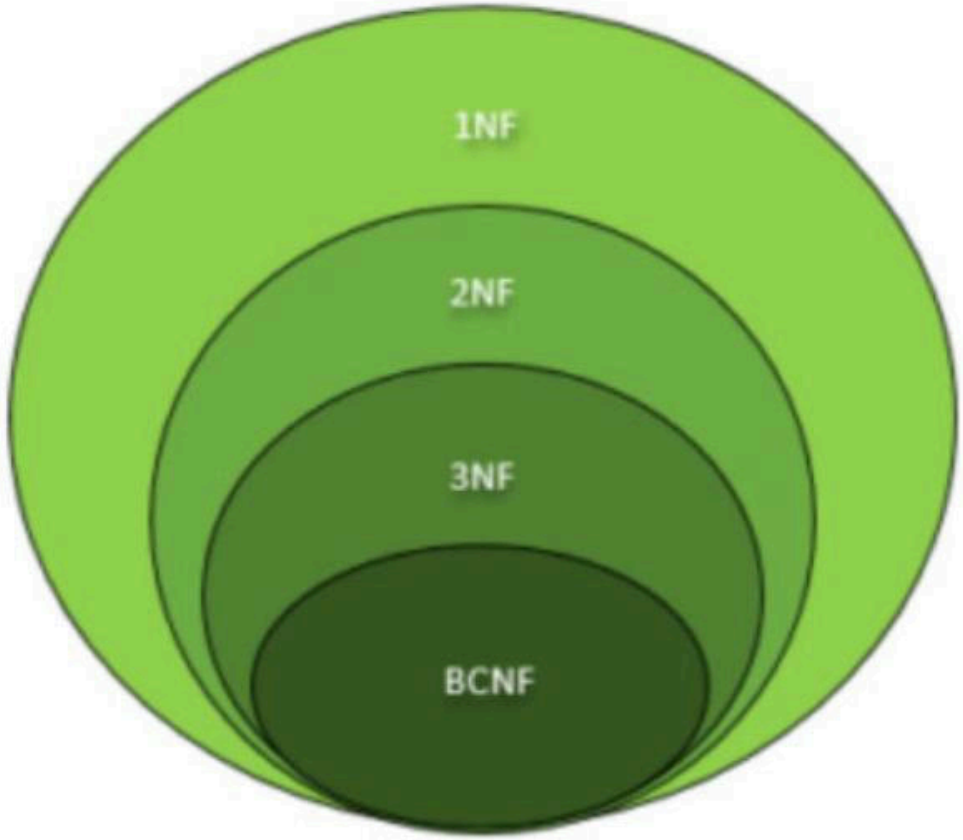
Normalization

! Memorize

For normalization, use functional dependencies. And not the semantics of the attributes!

Normalization - BCNF

- A relational schema R is in Boyce-Codd normal form (BCNF) if, for every one of its dependencies X ⟶ Y, one of the following conditions hold true:
  - X ⟶ Y is a trivial functional dependency (i.e., Y is a subset of X)



  - X is a super key for schema R
- Focusses on FDs within key attributes
- Every relation in BCNF is in 3NF, too … but not the other way round
- Informally: To test whether a relation is in BCNF, identify all the determinants and make sure that they are candidate keys

Normalization - BCNF

	UNF (1970)	1NF (1970)	2NF (1971)	3NF (1971)
Primary key (no duplicate tuples)	✓	✓	✓	✓
No repeating groups	✓	✓	✓	✓
Atomic columns (cells have single value)[8]	✗	✓	✓	✓
Every non-trivial functional dependency either does not begin with a proper subset of a candidate key or ends with a prime attribute (no partial functional dependencies of non-prime attributes on candidate keys)[8]	✗	✗	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with a prime attribute (no transitive functional dependencies of non-prime attributes on candidate keys)[8]	✗	✗	✗	✓
Every non-trivial functional dependency either begins with a superkey or ends with an elementary prime attribute[8]	✗	✗	✗	✗
Every non-trivial functional dependency begins with a superkey[8]	✗	✗	✗	✗
Every non-trivial multivalued dependency begins with a superkey[8]	✗	✗	✗	✗
Every join dependency has a superkey component[9]	✗	✗	✗	✗
Every join dependency has only superkey components[8]	✗	✗	✗	✗
Every constraint is a consequence of domain constraints and key constraints[8]	✗	✗	✗	✗
Every join dependency is trivial[8]	✗	✗	✗	✗

Normalization - 1NF

- **There should be atomic attribute values only!**
- Disallows:
  - composite attributes
  - multivalued attributes
  - nested relations
- PK determines every atomic attribute value
- In SQL-92 it’s not possible to have relations in Non First Normal Form (NFNF)

Normalization - 2NF

- **Relation is in 1NF and every nonkey attribute is full functional dependent on the key!**
  - No nonkey attribute should be functionally dependent on a part of the primary key
  - Applies only to relations where the PK contains multiple attributes

Normalization - 3NF

- **There should be no transitive dependency of a nonkey attribute on the primary key!**
  - No nonkey attribute has a FD on another nonkey attribute
  - Relation needs to be in 1NF and 2NF

Normalization - Rules for Decomposition

- Should be lossless
- Tuples of the original relation can be restored when joining the decomposed relations
- Functional dependencies should be preserved in one of the decomposed relations, so that original FDs can be restored

Assignment: Table into 1NF, 2NF, and 3NF

Student No	Student Name	Major	Course No	Course Name	Instr No	Instr Name	Instr Loc	Grade
123	Smith	DB	1 2 3	DB Math Physics	456 567 678	Jason Meyer Fish	R04 R01 R02	1 1 2
234	Jones	Math	1 2 4	DB Math OP	456 567 789	Jason Meyer Dench	R04 R01 R07	2 1 3

Assignment: Table into 1NF, 2NF, and 3NF

Full_Name	Physical Address	Movies Rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street No 34	Forgetting Sarah Marshal, Daddy‘s Little Girls	Mr.
Robert Phil	5th Avenue No 4	Clash of the Titans	Mr.

5. License Notice

5.1 Attribution

The basis of this course stems from: Professor Dr. Ulrike Herster
The following license applies to all unmodified and modified material used throughout this course.

5.1 Attribution

License

The publication or sharing of slides, image and sound recordings of this course are not permitted.