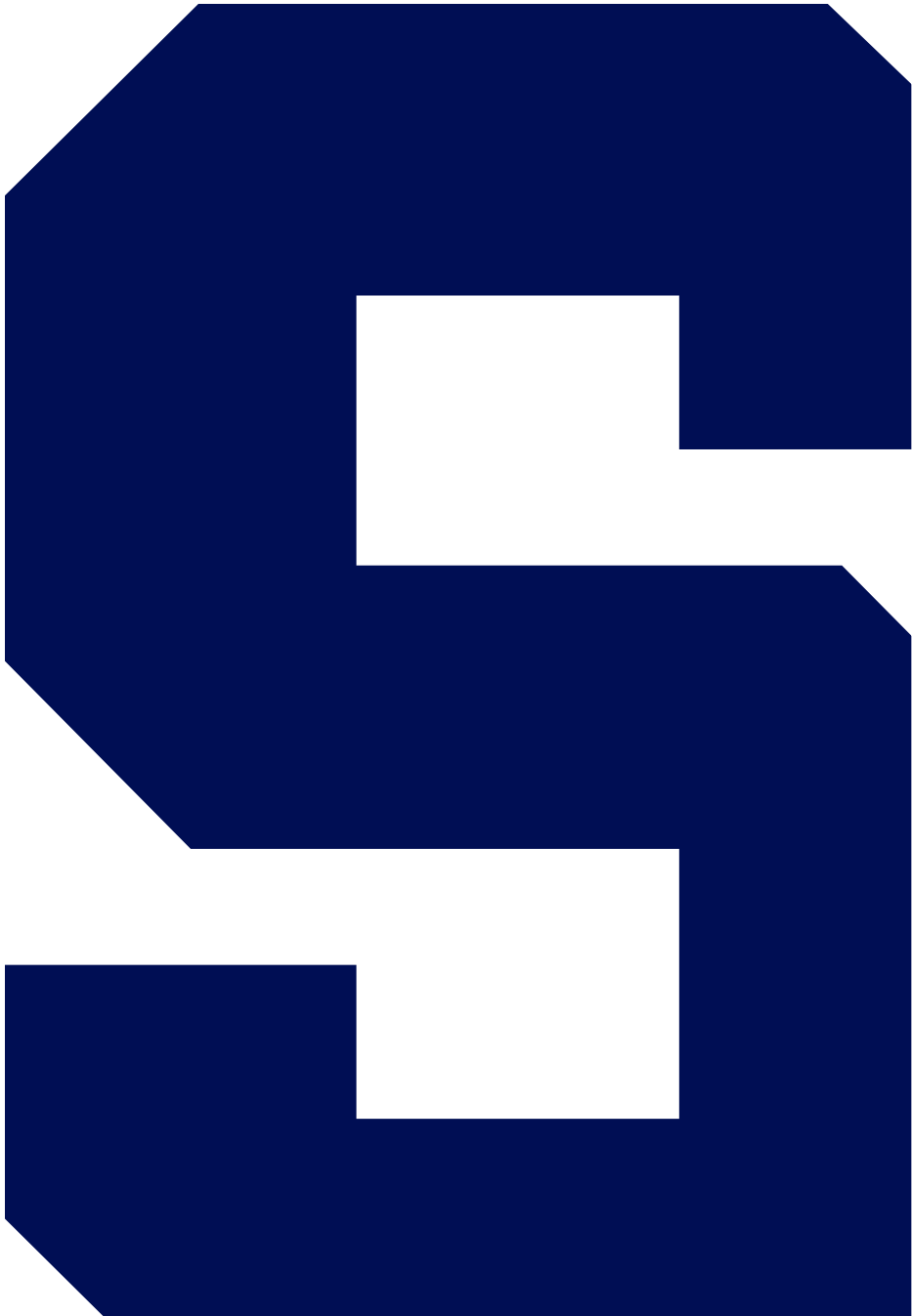
Data Modeling



Agenda

- What are conceptual and logical data modeling?
- How do I design a database?
- Understanding data requirements
- Conceptual modeling
  - Formalizing data requirements using the entity-relationship model
  - Drawing crow’s foot entity-relationship diagrams (ERDs)
- Logical modeling
  - Components of the relational logical model
  - Mapping entities, attributes, and relationships

Conceptual and Logical Modeling

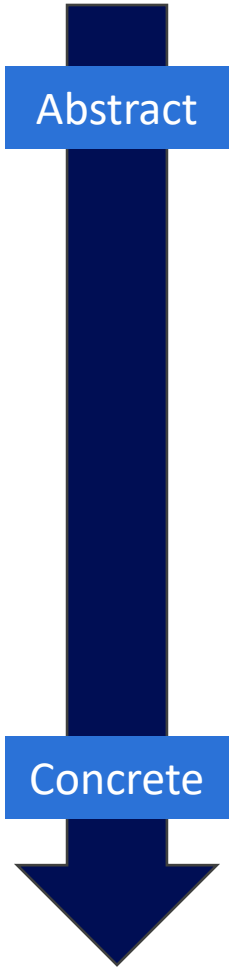
Conceptual modeling

- Identify the data requirements
- These represent the needs of the database
- Entity-relationship modeling is a common approach for representing the conceptual model
- Conceptual model takes no technology into consideration

Logical modeling

- Completed after conceptual modeling
- The design (or blueprint) of a conceptual model
- Specifies how the data requirements will be implemented on a DBMS
- This considers a specific database implementation model, such as relational, graph, or document

Recall: Data Models



Conceptual data model
- An abstract representation of the data requirements
- No implementation of the database itself; no implementation model selected

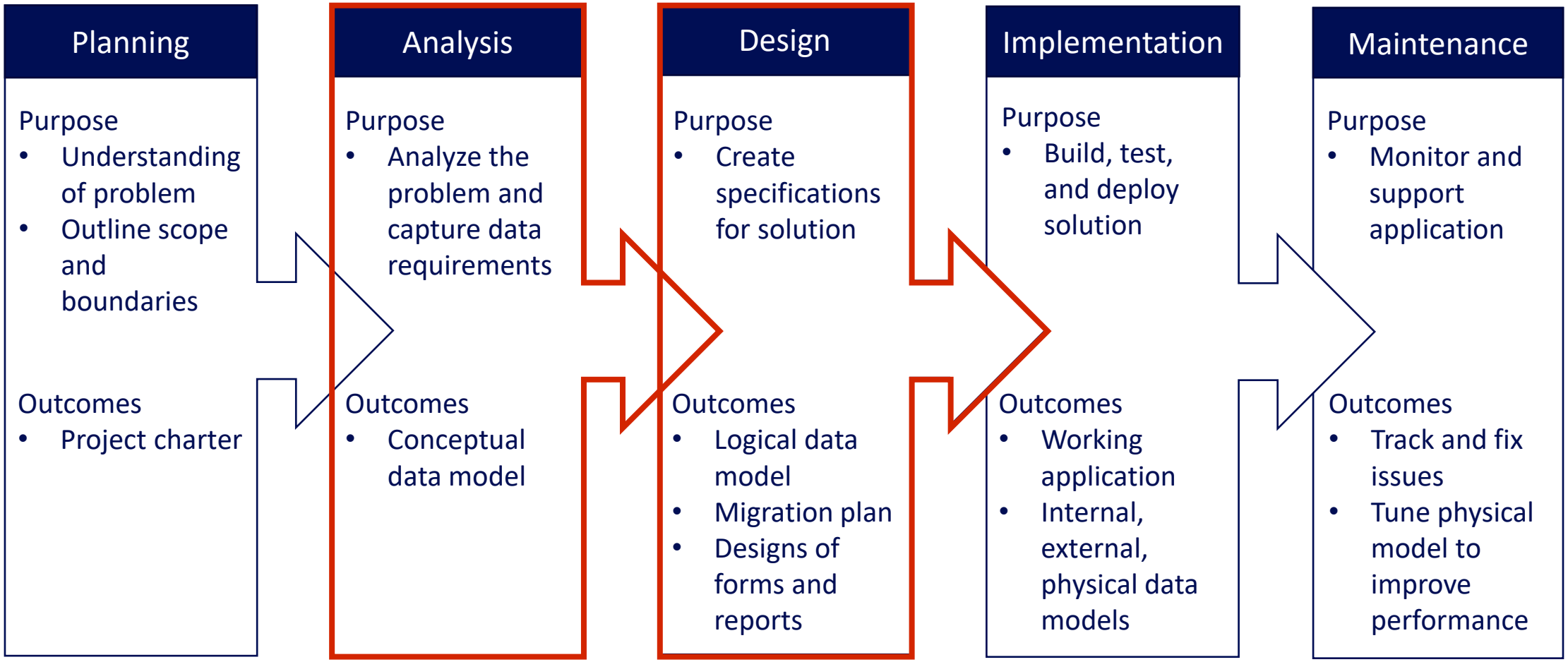
Logical data model
- Mapping of the conceptual model to an implementation model
- No implementation of the database; model selected

Internal data model
- The internalized implementation of the database application
- A DBMS and implementation model selected

External data model
- User's view of the database application
- DBMS and implementation model selected, internal model implemented

Physical data model
- How the internal/external model is stored by the DBMS and operating system
- All aspects of the database are an implementation

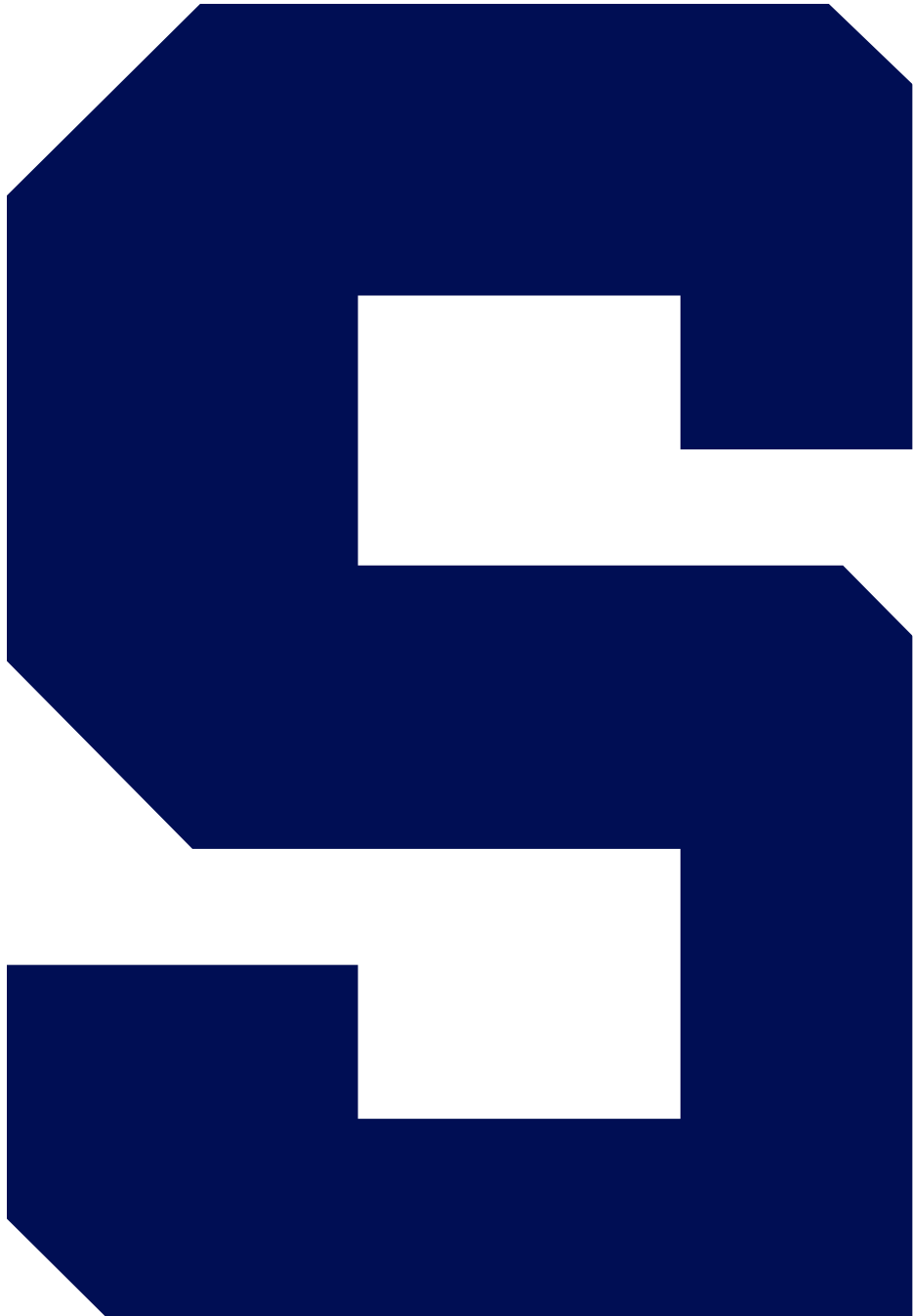
Recall: DBLC


Planning
Purpose
• Understanding of problem
• Outline scope and boundaries
Outcomes
• Project charter
Analysis
Purpose
• Analyze the problem and capture data requirements
Outcomes
• Conceptual data model
Design
Purpose
• Create specifications for solution
Outcomes
• Logical data model
• Migration plan
• Designs of forms and reports
Implementation
Purpose
• Build, test, and deploy solution
Outcomes
• Working application
• Internal, external, physical data models
Maintenance
Purpose
• Monitor and support application
Outcomes
• Track and fix issues
• Tune physical model to improve performance

Data Modeling

The End

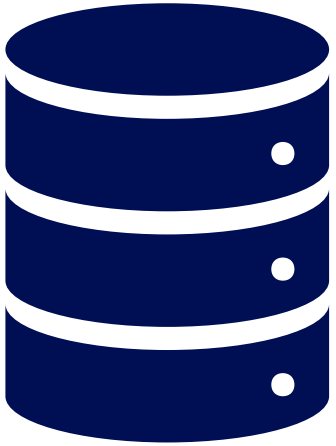
Syracuse University
School of Information Studies



Conceptual Data Modeling

How Do I Design a Database?

First, you must understand the problem, then you must define it.



Conceptual Data Modeling

- The process of collecting then documenting the data requirements of a problem to gain a well-defined understanding of it
- Goals
  - Understand the problem.
  - Define the problem formally.
- No consideration for how it will be implemented
  - Technology, DBMS, or implementation data model are not considered.

What Exactly Is a Requirement?

- A statement that identifies the needs of a system or application
- Two types
  1. Functional requirement: features, implemented in software
  2. Nonfunctional requirement: not a feature, but guidelines, behaviors, and expectations
- Conceptual modeling is concerned with data requirements
  - Functional requirements address what data will be stored specifically

Examples of Requirements

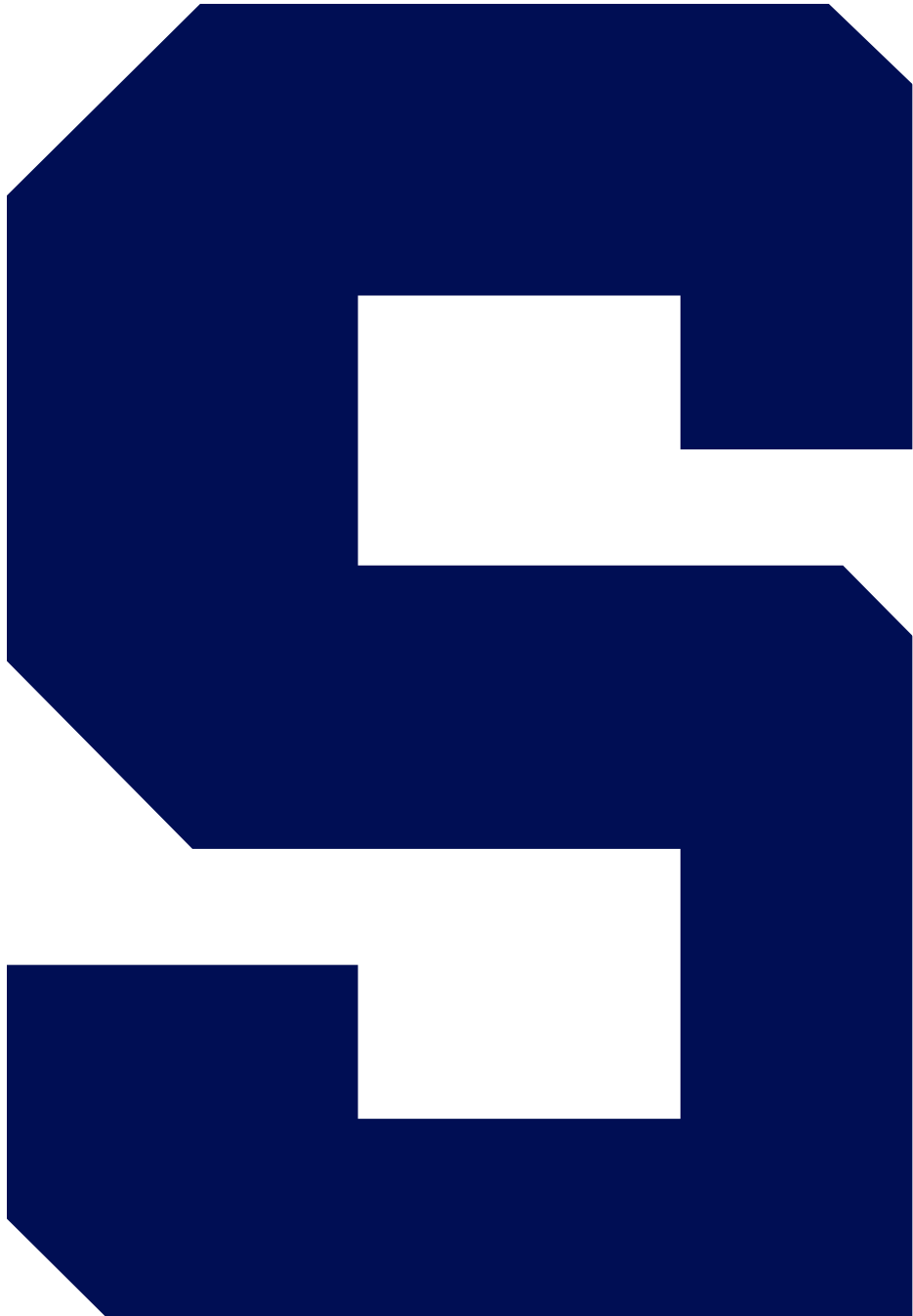
Nonfunctional	Functional	Data
• Customers should be able to access the system 24/7. • The system should scale to support 10,000 concurrent users. • Images are stored in Amazon EC2.	• A customer should be able to search the catalog by product name. • Vendors should be able to build a landing pages for their products. • Vendors can upload images of their products.	• A product should contain name, price, and vendor information. • Vendors can create one or more landing pages, but a landing page is created by just one vendor. • A product can have several images associated with it.

Syracuse University
School of Information Studies

Conceptual Data Modeling

The End

Syracuse University
School of Information Studies

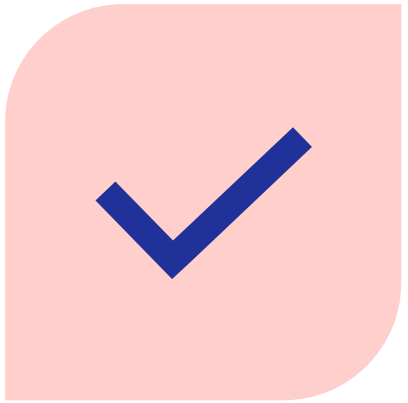


Gathering Data Requirements

Requirements Gathering

Identify
key stakeholders.

Identify
other systems.



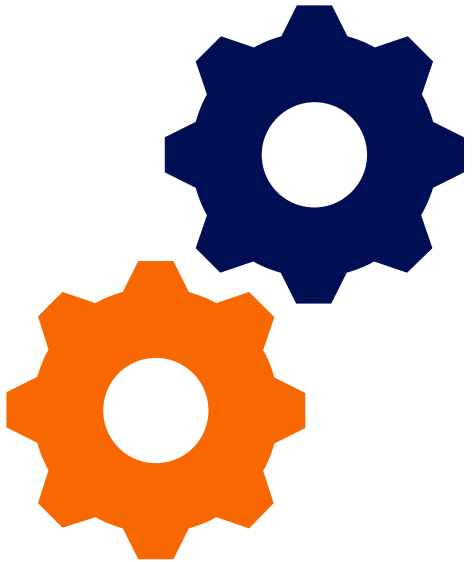
Set the scope
and boundaries.

Key Stakeholders

- Any individual with a vested interest in the outcome of the system
- End users and their managers
- The development team
- Those who will maintain and support it
- External entities

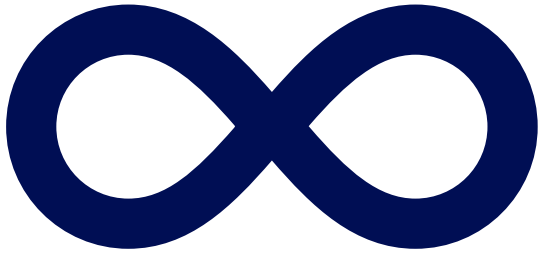
Dependent Systems

- Inflows: providers of data
- Outflows: consumers of Information
- Infrastructure: technology used to run the system

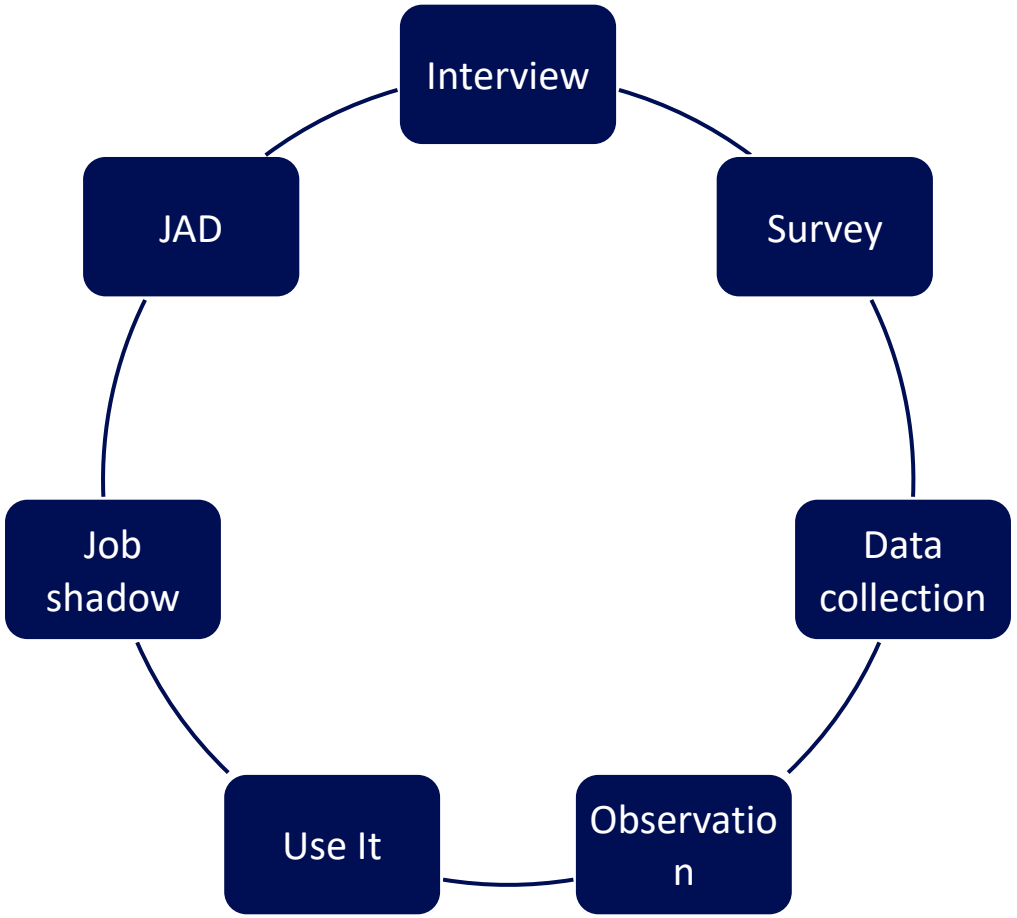


Scope and Boundaries

- What is outside what will be done?
- Prioritizing requirements: not relevant, low importance, or critical
- Resource constraints: people, money, time
- MVP: minimum viable product; better approaches start small and iterate with subsequent versions



Requirements-Gathering Techniques



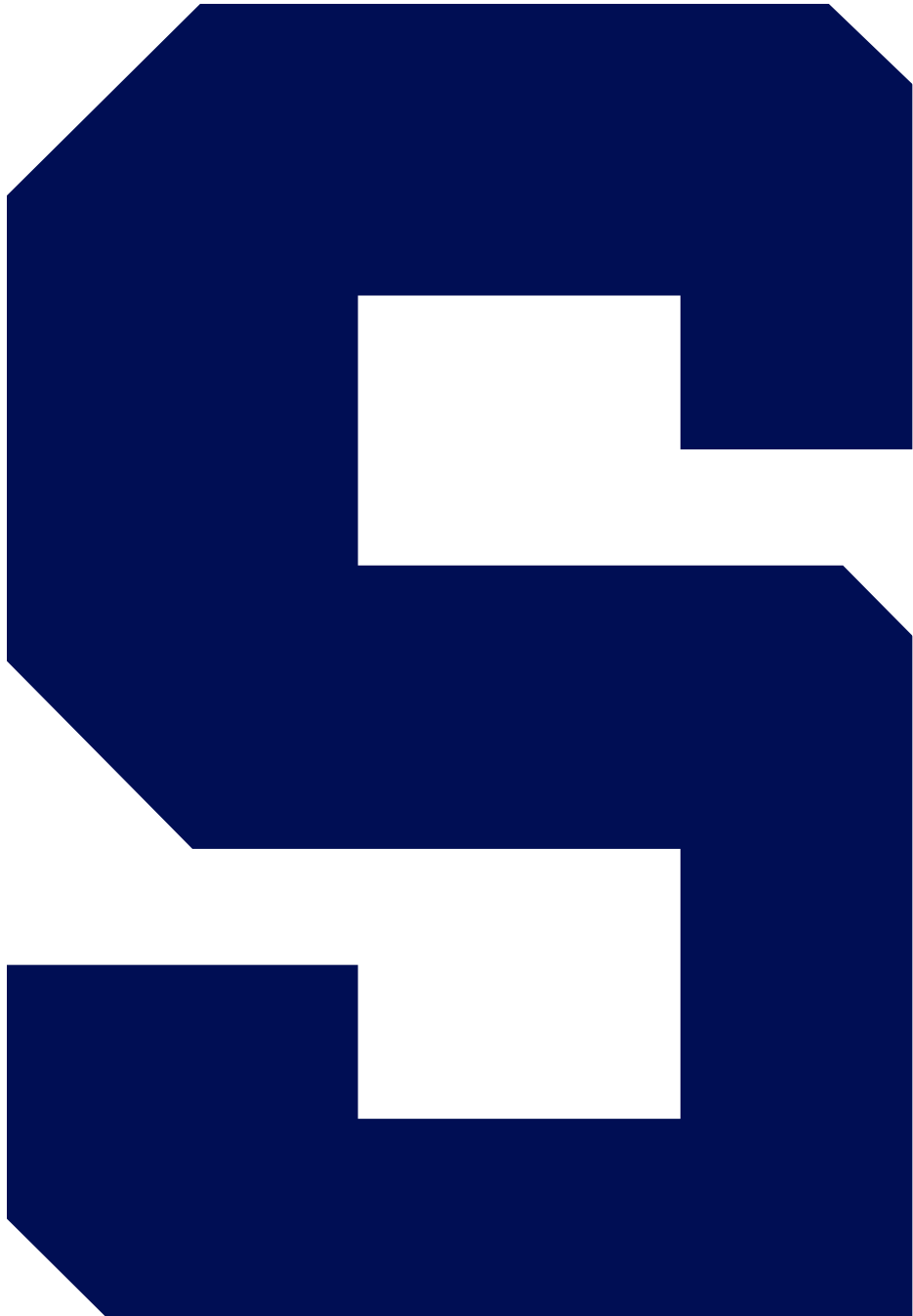
The Importance of Problem Domain

You Cannot Solve a Problem You Do Not Understand

Gathering Data Requirements

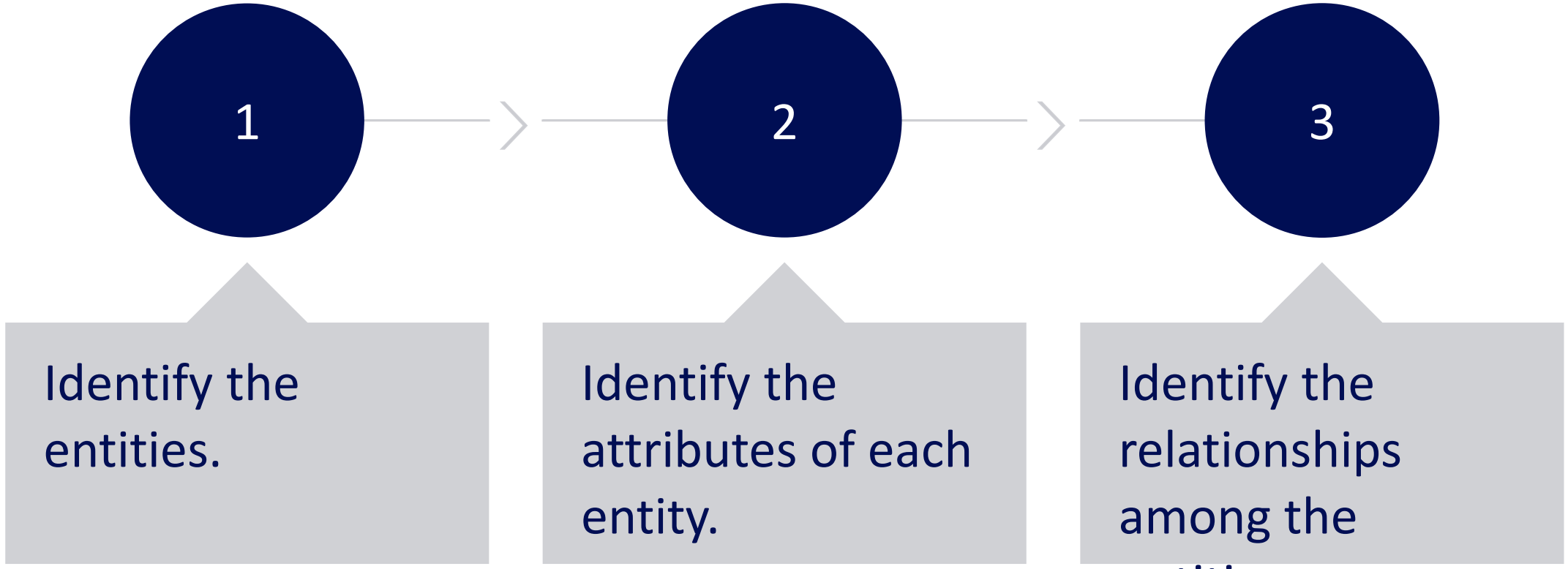
The End

Syracuse University
School of Information Studies



Entity-Relationship Modeling

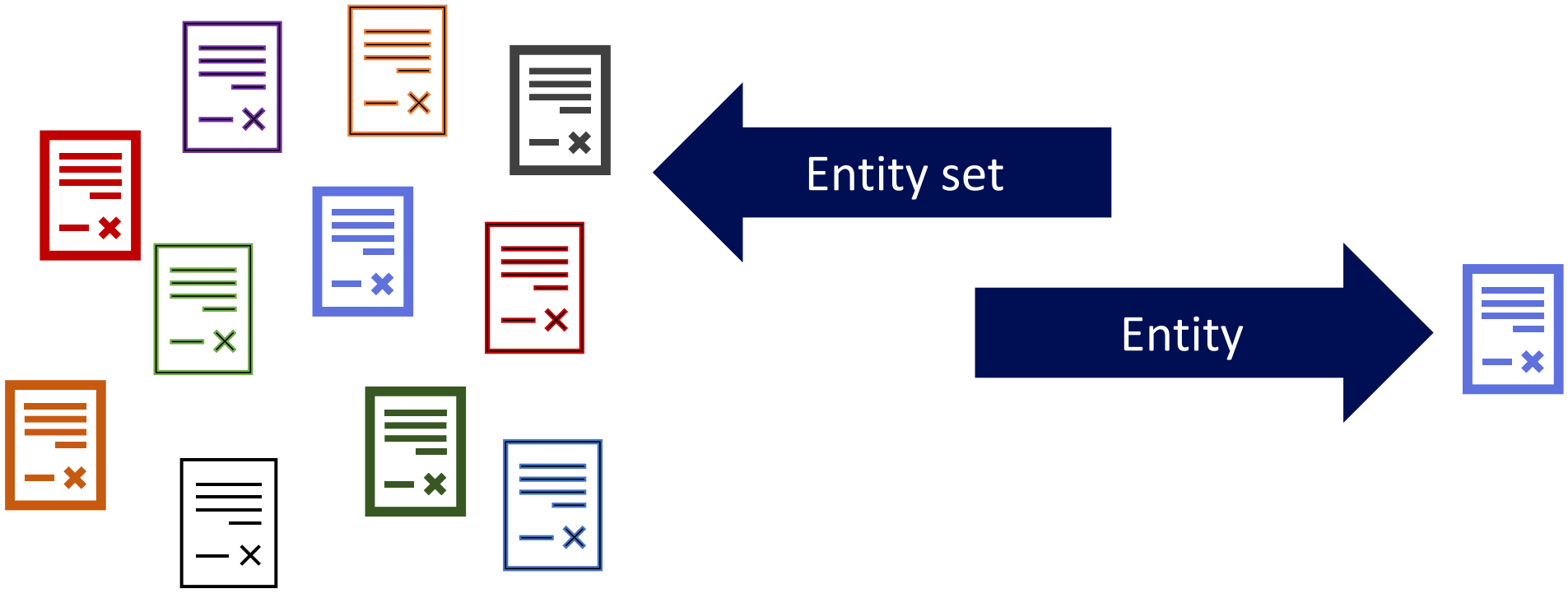
Entity-Relationship (E-R) Modeling



Identify Entities

- Things for which we need to store
- These should be data as objects and have composite attributes
- Examples
  - People
  - Places
  - Things
  - Events
- Entities are individual things, like an invoice or book
- Entity set: collection of entities

Entities and Entity Sets



Examples of Entities

- People
  - Customer
  - Employee
  - Student
- Places
  - Department
  - Airport
  - Business
- Things
  - Computer
  - Vehicle
  - Equipment
- Events
  - Payment
  - Appointment
  - Accident

Identify Attributes of Each Entity

- Entities are composite objects and therefore should have attributes
- Possible attribute properties
  - Unique: natural/business keys
  - Required: must be included for every entity
  - Composite: data as objects, should be resolved to atomic values on implementation
  - Derived: attribute is calculated from other attributes
  - Multi-valued: atomic but has more than one value

Example of Attributes

Customer

- Email [RU]
- Name [RC]
- Address [C]
- Phone [M] (home phone, cell phone)

Payment

- Date of payment [R]
- Amount [R]
- Method [R] (credit card, check, cash)

Identify the Relationships Among the Entities

- The entities that take part in the relationship are known as the participants.
- Relationships are expressed in two directions.
- Cardinality represents the minimum and maximum number of participants.
- The maximum cardinality on each side identifies the relationship classification.
  - 1-1
  - 1-M
  - M-M

Example of Relationships

- 1-1 classification
  - A resident is issued 0 or 1 driver's license
  - A driver's license is issued to 1 and only 1 resident
- 1-M classification
  - A customer submits 0 or more payments
  - A payment submitted by 1 and only 1 customer
- M-M classification
  - A student is a member of 0 or more clubs
  - A club enrolls 0 or more students

When Capturing E-R Data Requirements

- Include contextual descriptions for the entities.
  - Example: A product is an item for sale in our catalog.
- Include a data definition for the attributes.
  - Example: The product UPC [RU] is the product's universal product code. This is globally unique.
- Specify data relationships for both participants and use a verb that describes the business rule.
  - No: A vendor has 0 or more products.
  - Yes: A vendor supplies 0-M products; a product is supplied by 1-1 vendor.

Entity-Relationship Modeling

The End

Syracuse University
School of Information Studies

Example

Netflix Requirements

Example: Data Requirements for Netflix

- Entities
  - Customer: an external-facing user of the service
  - Payment: monies owed for using the service
- Attributes (example for customer)
  - Email (required, unique): must be included when signing up for the service
  - Billing address (required, composite): typical international mailing address format used for billing a customer
- Relationships (1-M customer payment)
  - A customer submits 0 or more payments
  - A payment is submitted by 1 and only 1 customer

Example: Data Requirements for Netflix (cont.)

Entities and Attributes			
Entity	**Attribute**	**Props**	**Descripion**
Customer	Email	RU	Required to use service, bus. Key
	Billing Address	RC	Typical international mailing address
	Name	RC	Full name last and first
	Phone	M	Phone number (home, cell)
Payment	Date	R	Date of Payment
	Amount	R	Amount paid in USD
	Method	R	Payment Method: Cash, Credit, Debit

Relationships					
Relationship	**Entity**	**Rule**	**Min**	**Max**	**Entity**
Customer Payment 1-M	Customer	Submits	0	M	Payment
	Payment	Submitted By	1	1	Customer

Example

The End

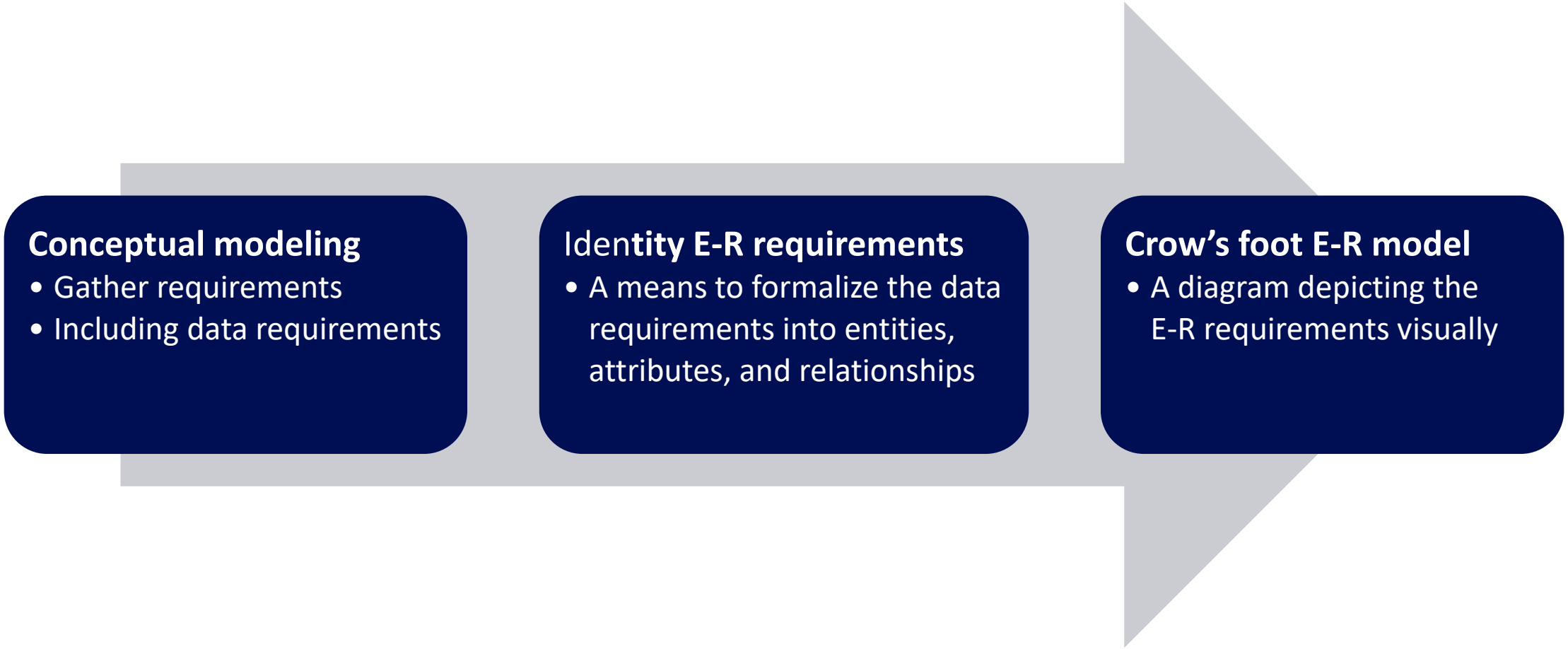
Syracuse University
School of Information Studies

Drawing Crow's Foot E-R Diagrams

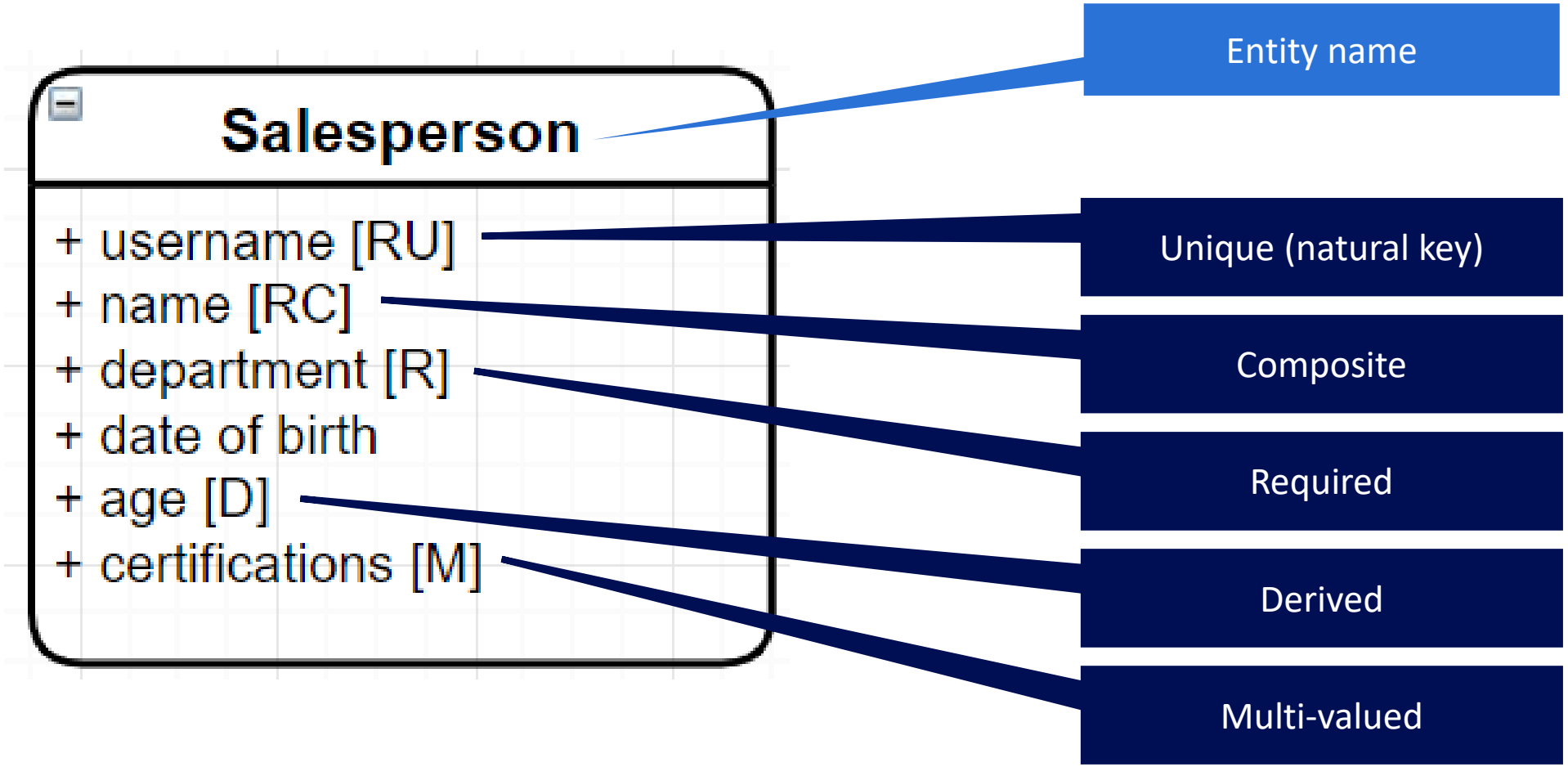
E-R Diagrams

- Draw an entity-relationship diagram from our E-R modeling data requirements.
- The diagram is a documentation and communications tool, representing the conceptual data model.
- A few different notations exist, such as Chen, Bachman, UML, and crow's foot.
- We will use the crow's foot notation, which is popular, easy to learn, and very intuitive for others to read.

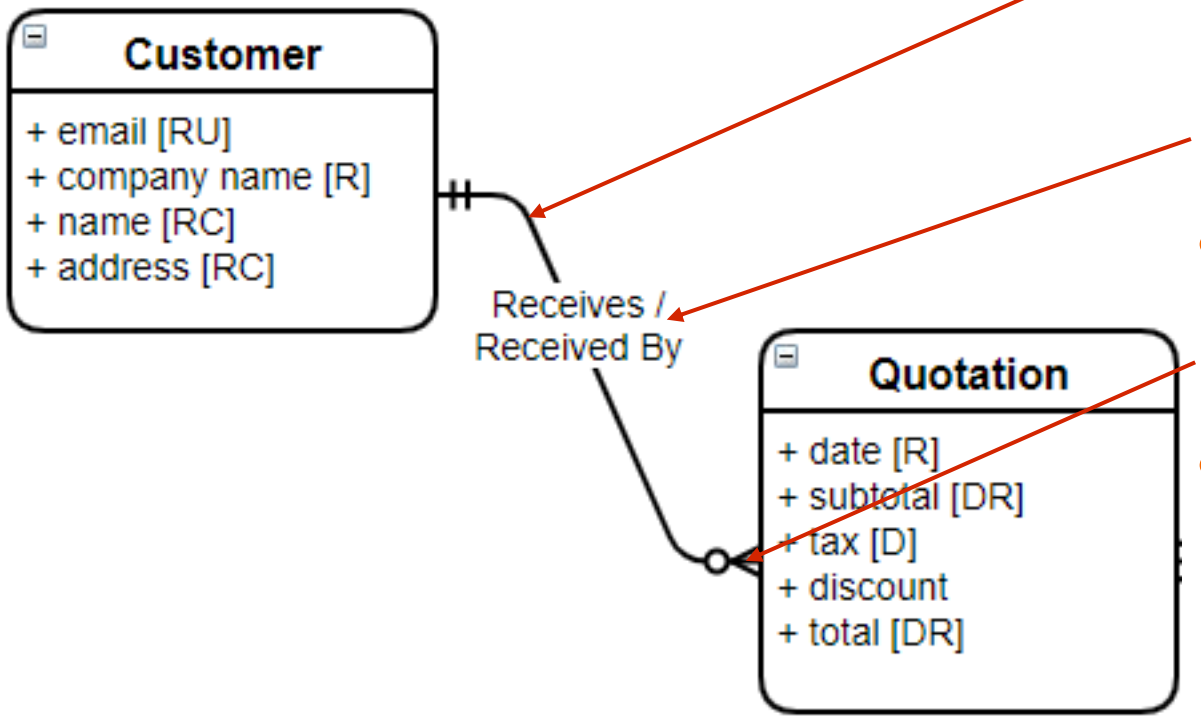
Conceptual Modeling Summarized



Crow's Foot ERD Entity and Attributes



Relationships



- A line connects the participating entity of a relationship.
- The business rule is labeled on the line.
- Cardinality is displayed on each end of the line.

Valid Cardinalities

- Zero or one (minimum = 0, maximum = 1)

- One and only one (minimum = 1, maximum = 1)

- Zero or more (minimum = 0, maximum = many)

- One or more (minimum = 1, maximum = many)

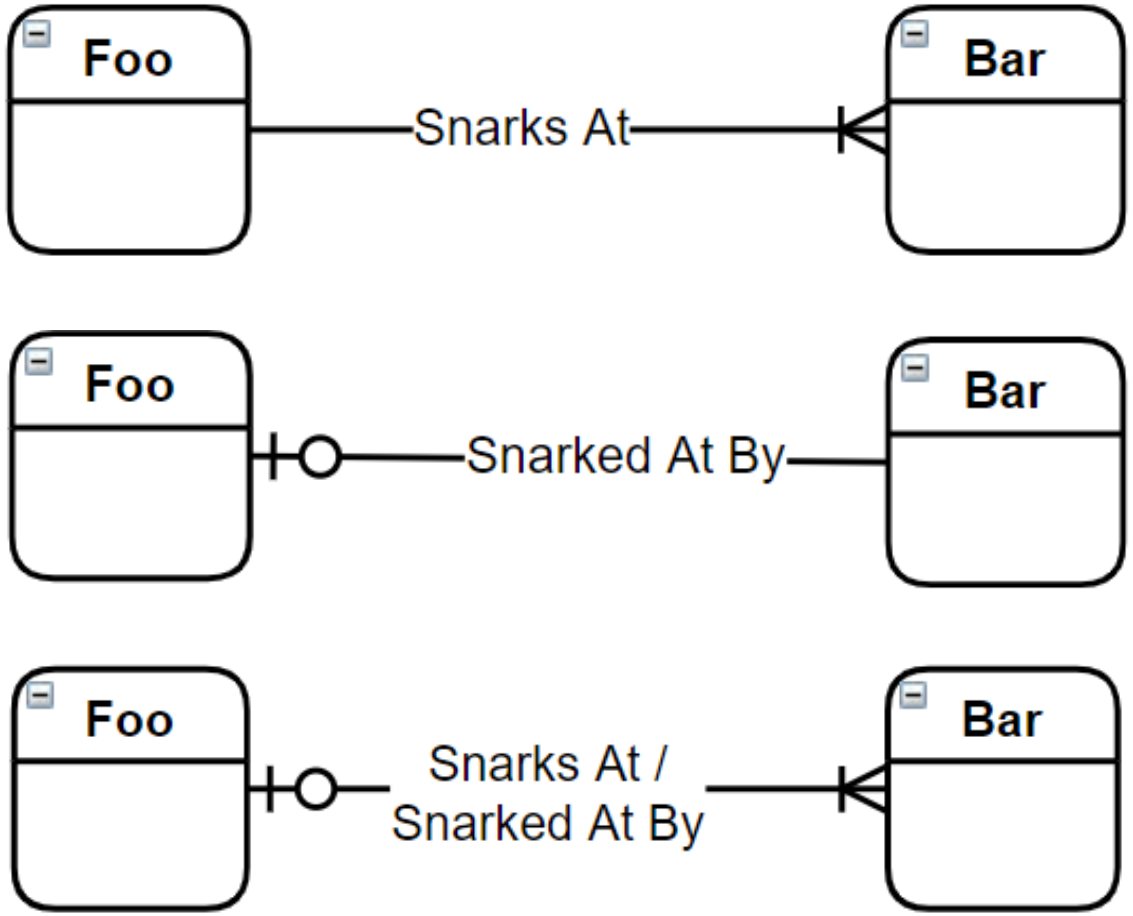
Drawing Relationships From Requirements

"A foo snarks at one or more bar"

+

"A bar snarked at by zero or one foo"

=



Syracuse University
School of Information Studies

Drawing Crow's Foot E-R Diagrams

The End

Syracuse University
School of Information Studies

Demo

Draw a Crow’s Foot ERD

Demo: Draw a Crow's Foot ERD

- We will use Draw.io from Diagrams.net to draw the diagram
- Data requirements

Entities and Attributes

Entity	Attribute	Props	Descripion
Customer	Email	RU	Required to use service, bus. Key
	Billing Address	RC	Typical international mailing address
	Name	RC	Full name last and first
	Phone	M	Phone number (home, cell)
Payment	Date	R	Date of Payment
	Amount	R	Amount paid in USD
	Method	R	Payment Method: Cash, Credit, Debit

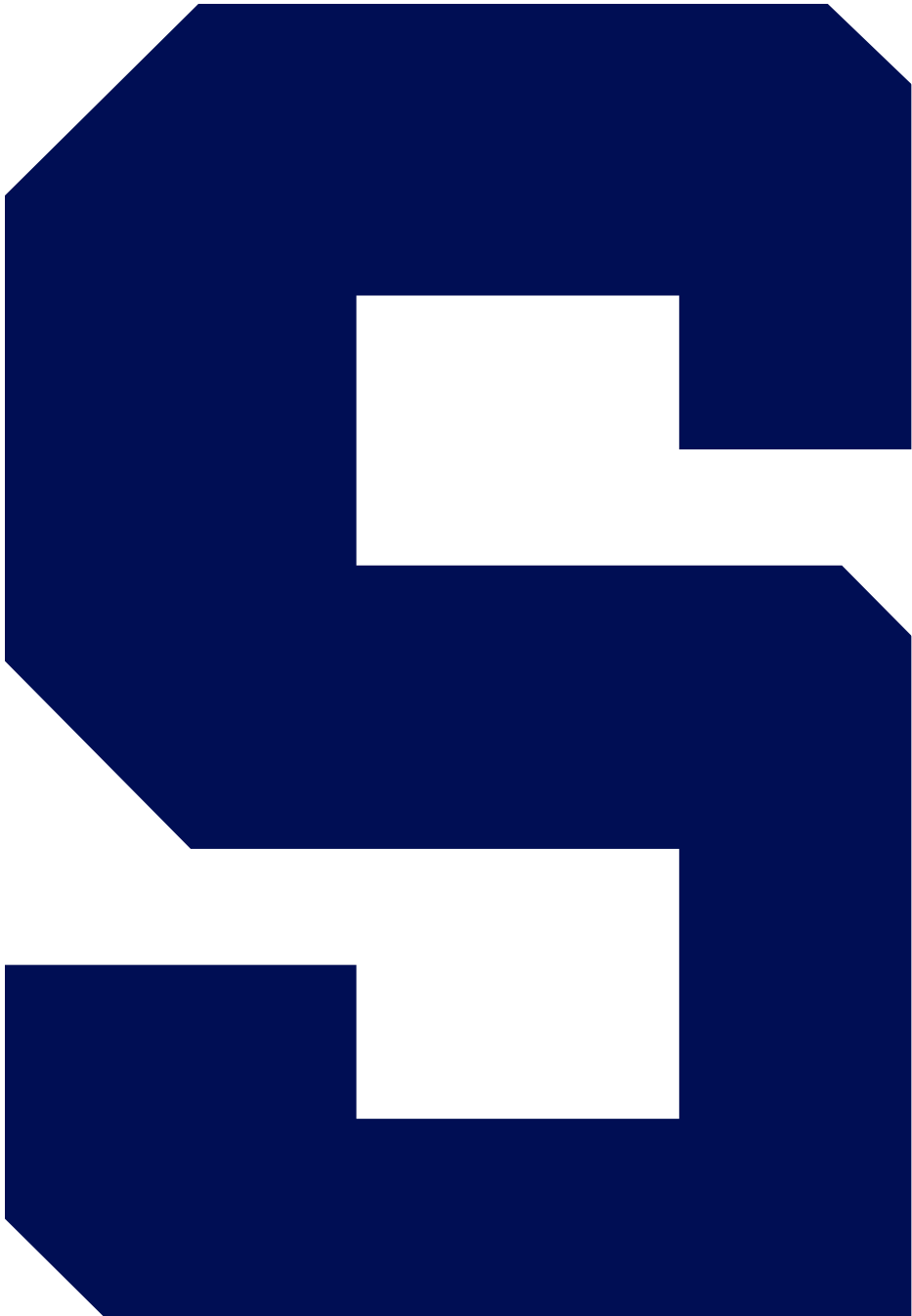
Relationships

Relationship	Entity	Rule	Min	Max	Entity
Customer Payment 1-M	Customer	Submits	0	M	Payment
	Payment	Submitted By	1	1	Customer

Demo: Draw a Crow's Foot ERD

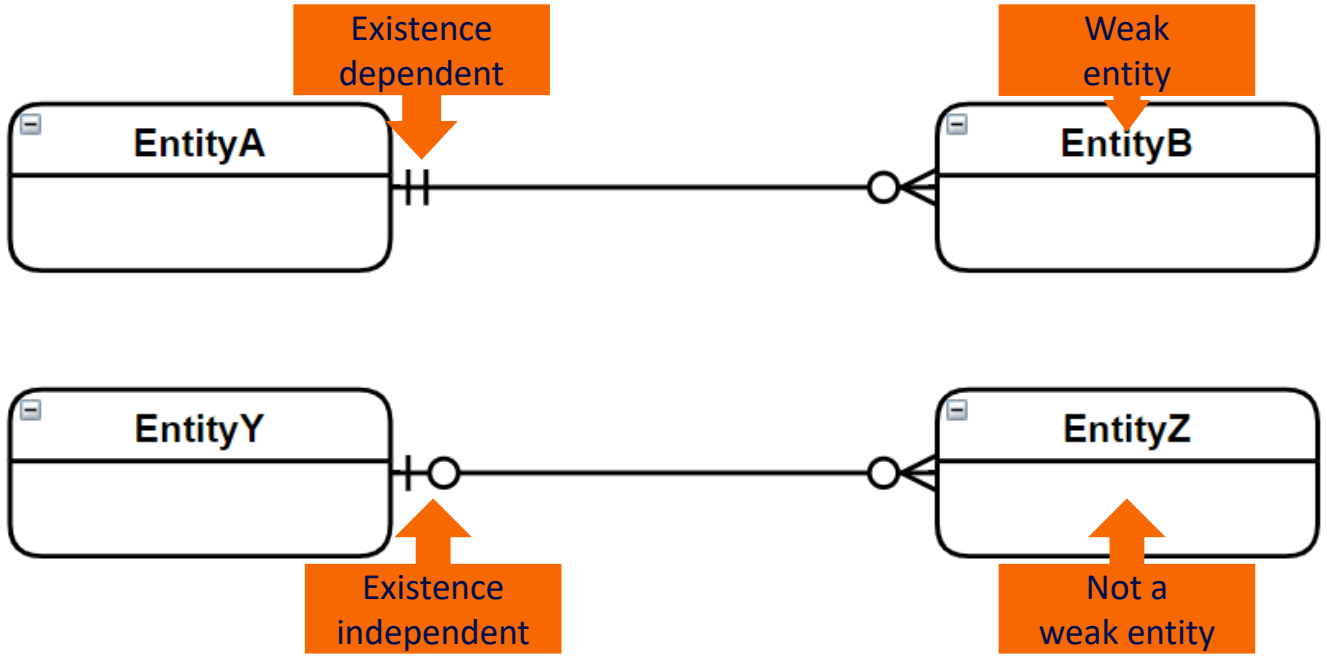
The End

Advanced E-R Modeling

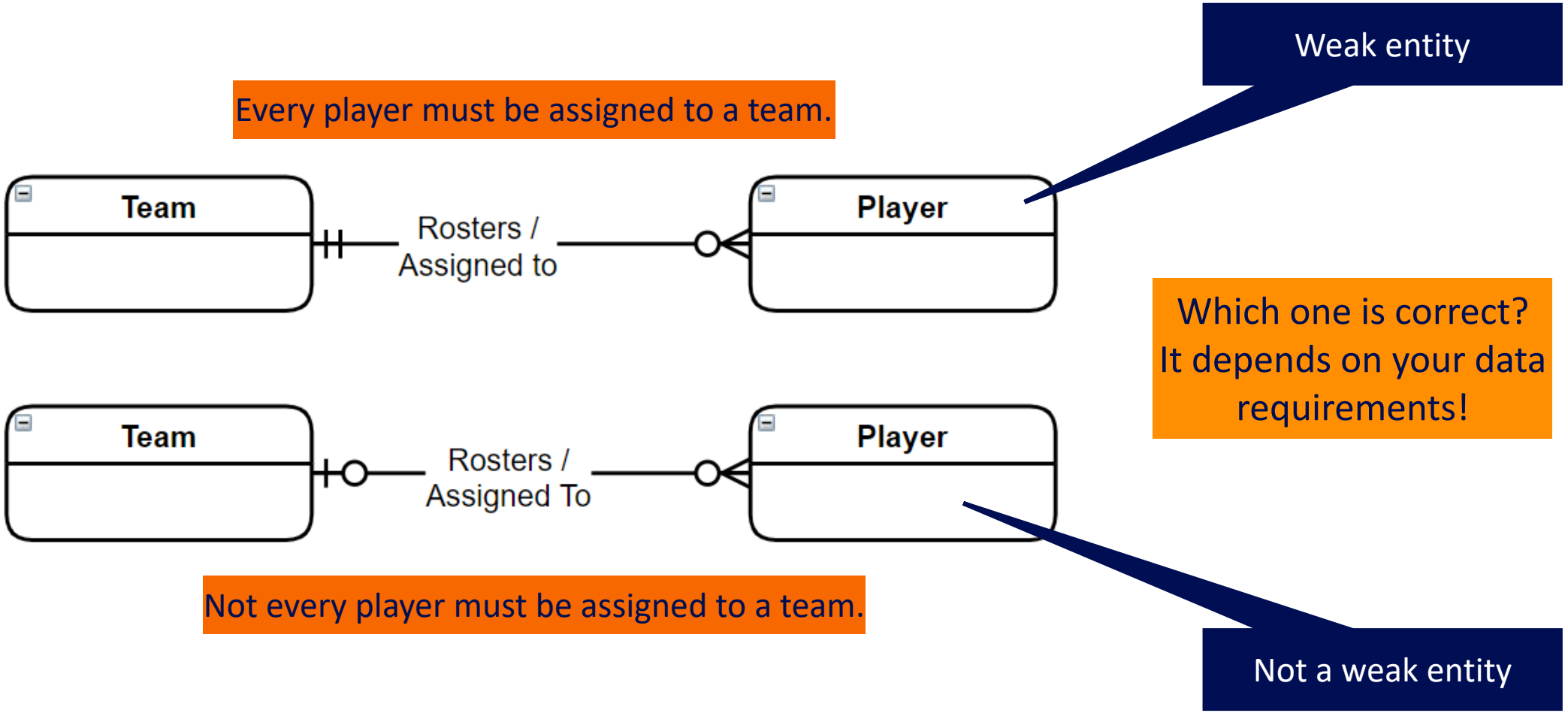


Weak Entities

- For a 1-M classification, a weak entity occurs when the entity on the many side cannot exist without the participating entity on the 1 side.
- In this case, the relationship is said to be existence dependent.

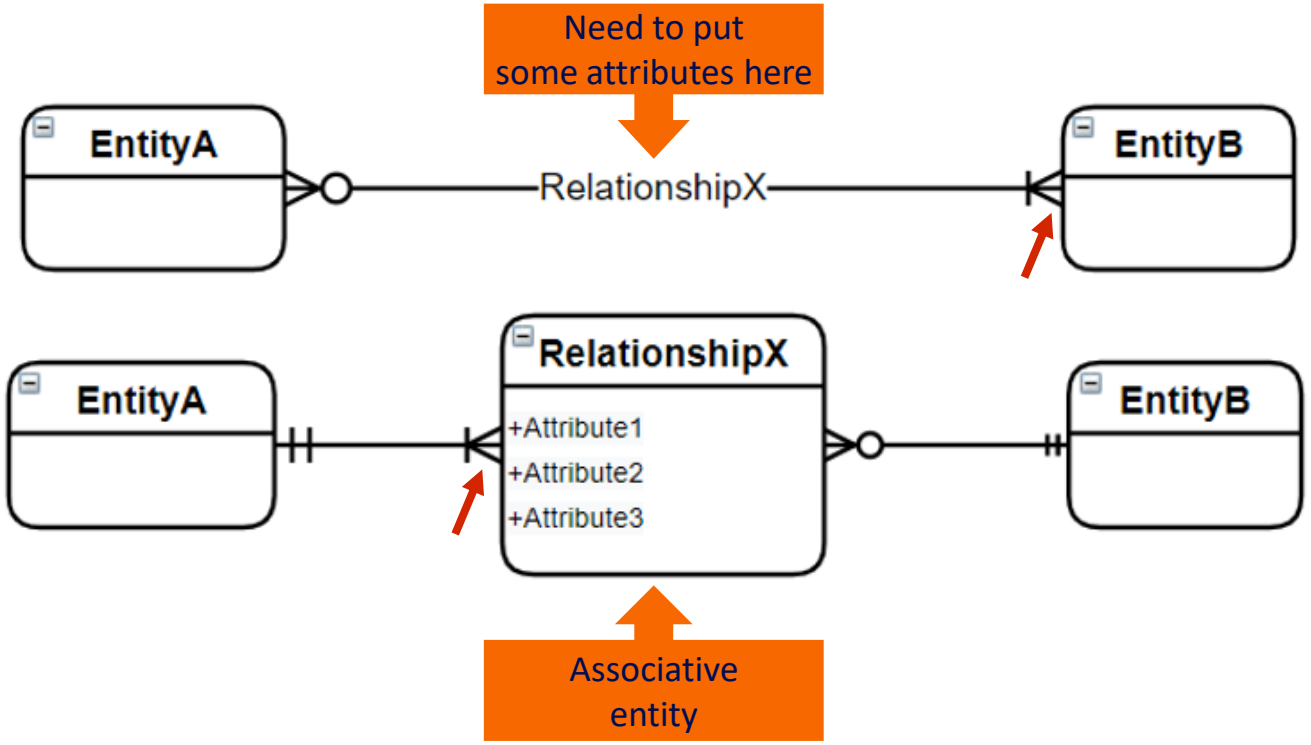


Example: Checking for Weak Entities

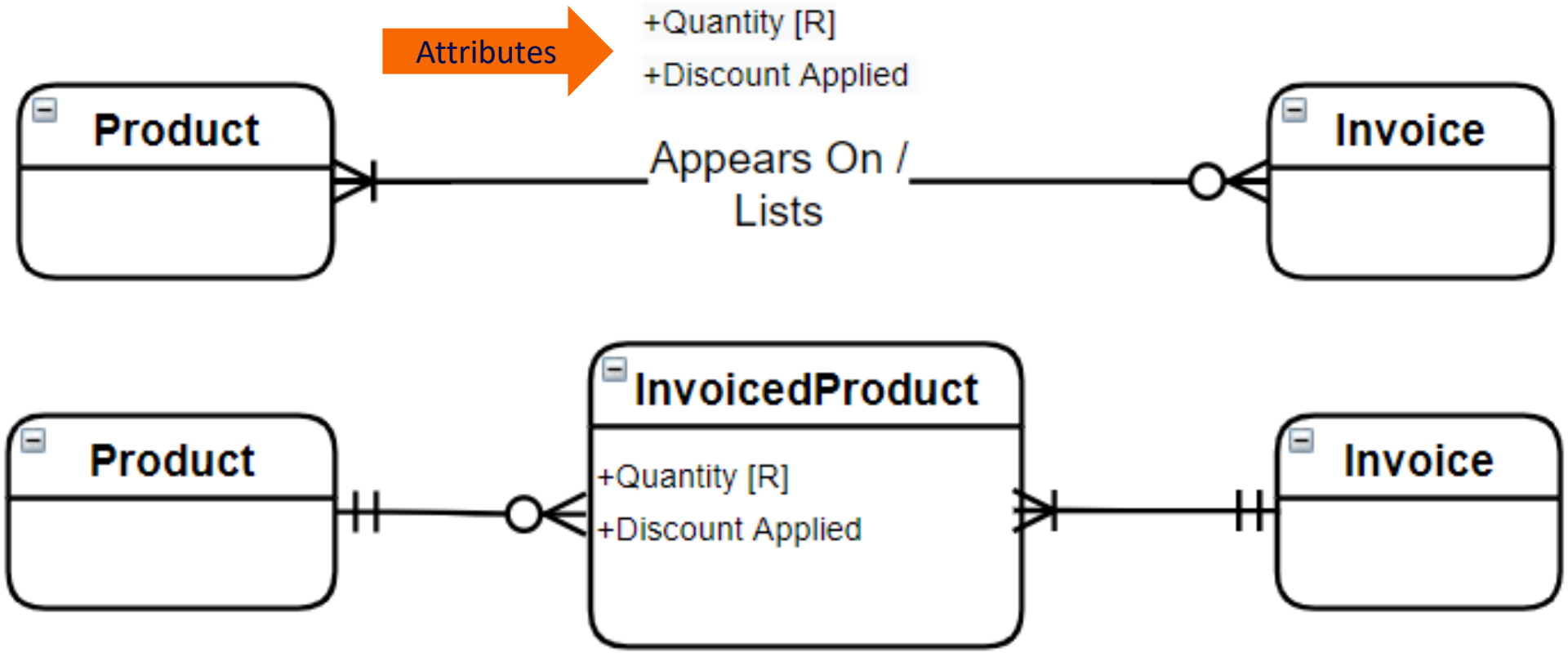


Associative Entities

- An M-N classification that contains attributes must be resolved to its own entity. This is known as an associative entity.
- Cardinality on the associative entity matches that on the original relationship.

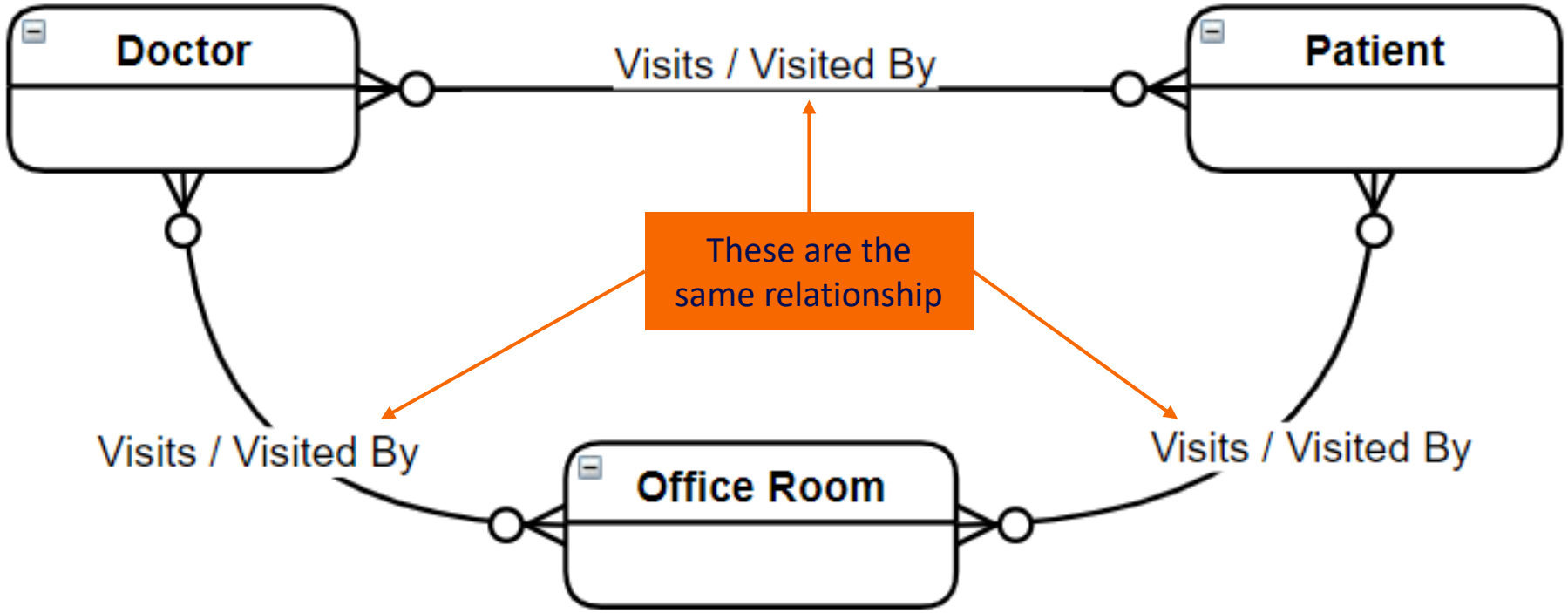


Example: Resolving Associative Entities



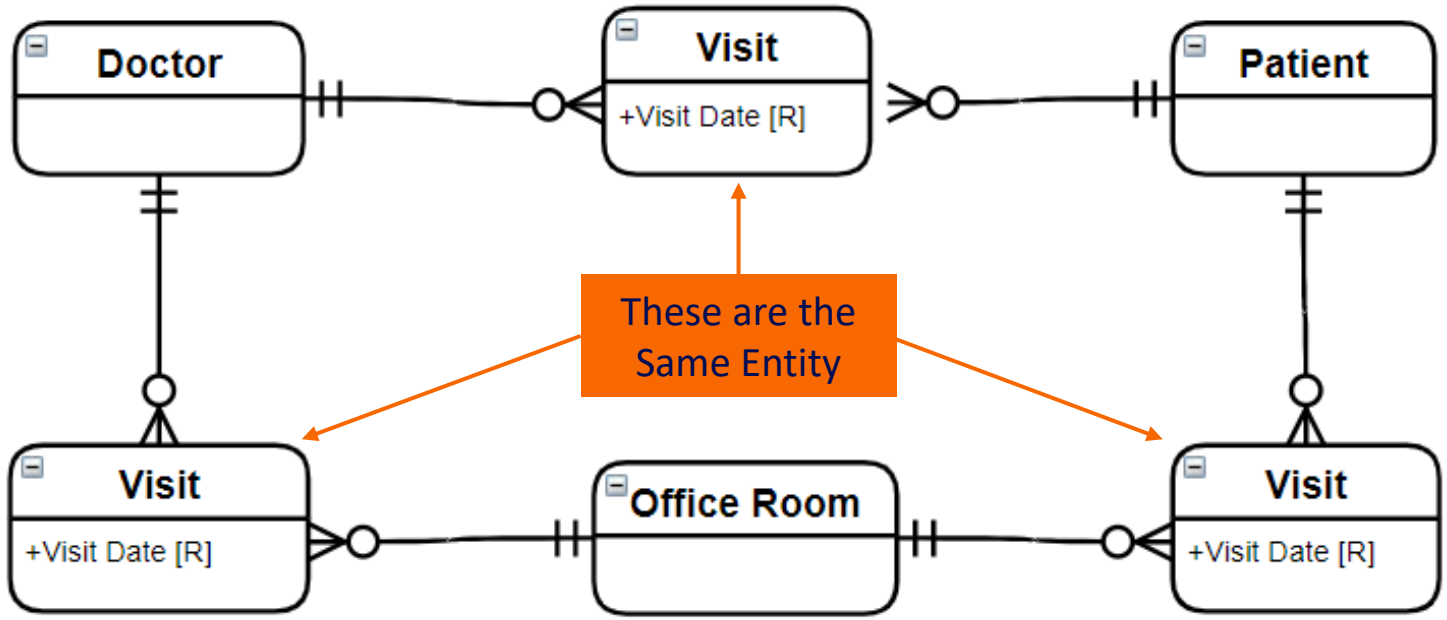
Tertiary/N-ary Relationships

Sometimes the same relationship is re-used among several entities.



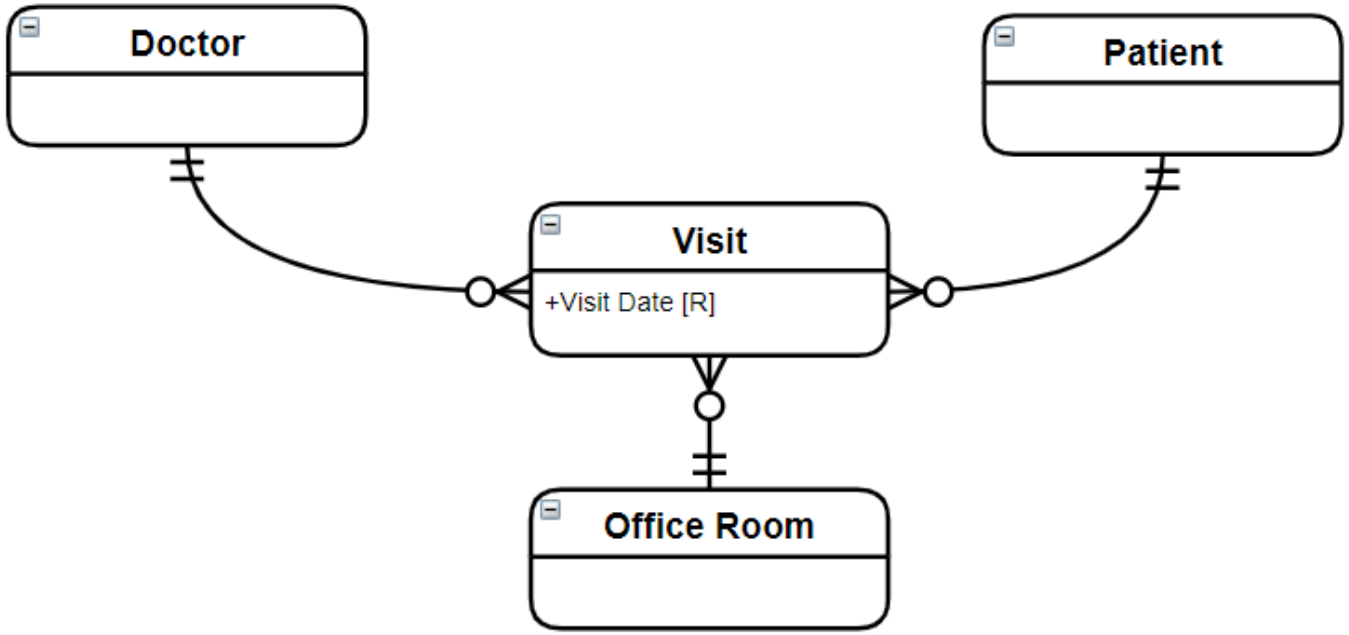
If We Apply the Associative Entity Rule

We end up with the same entity three times if they are associative.



Representing an N-ary Relationship

- This is redundant and can be simplified.
- Treat N-ary relationships as associative entities.

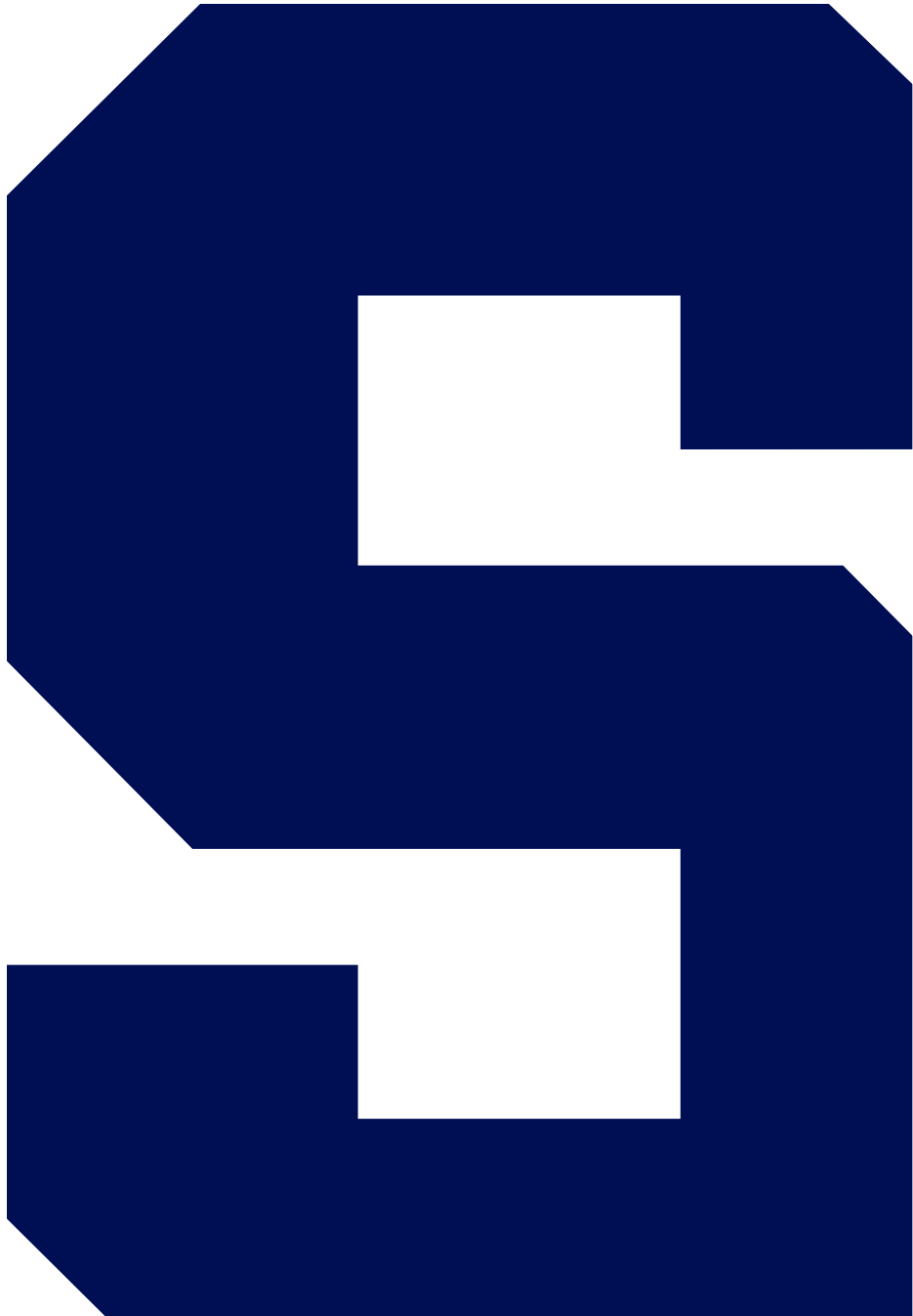


Syracuse University
School of Information Studies

Advanced E-R Modeling

The End

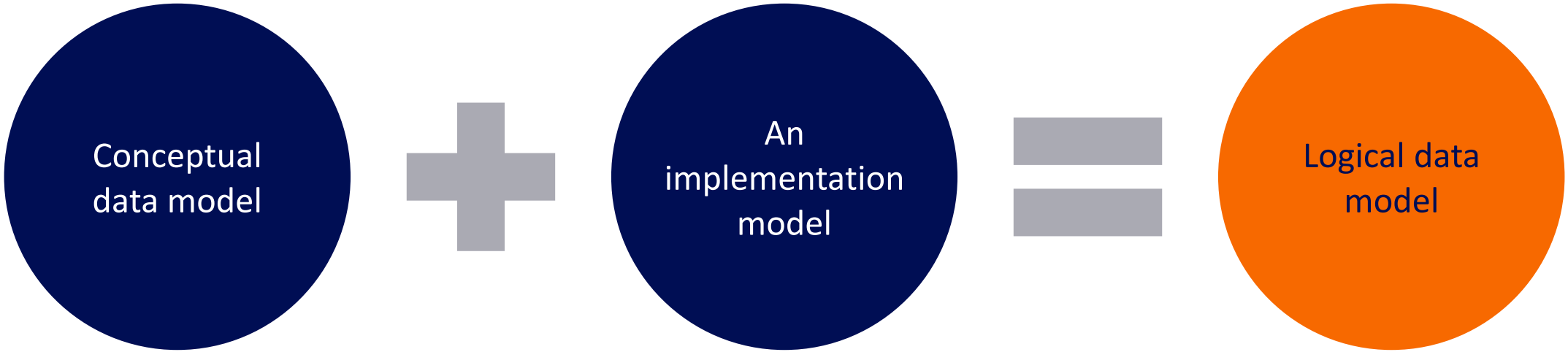
Syracuse University
School of Information Studies



Logical Data Modeling

What Is a Logical Data Model?

- A logical data model is the implementation of a conceptual data model using a particular database implementation model.
- The logical model will depend on an implementation but does not represent an actual implementation.

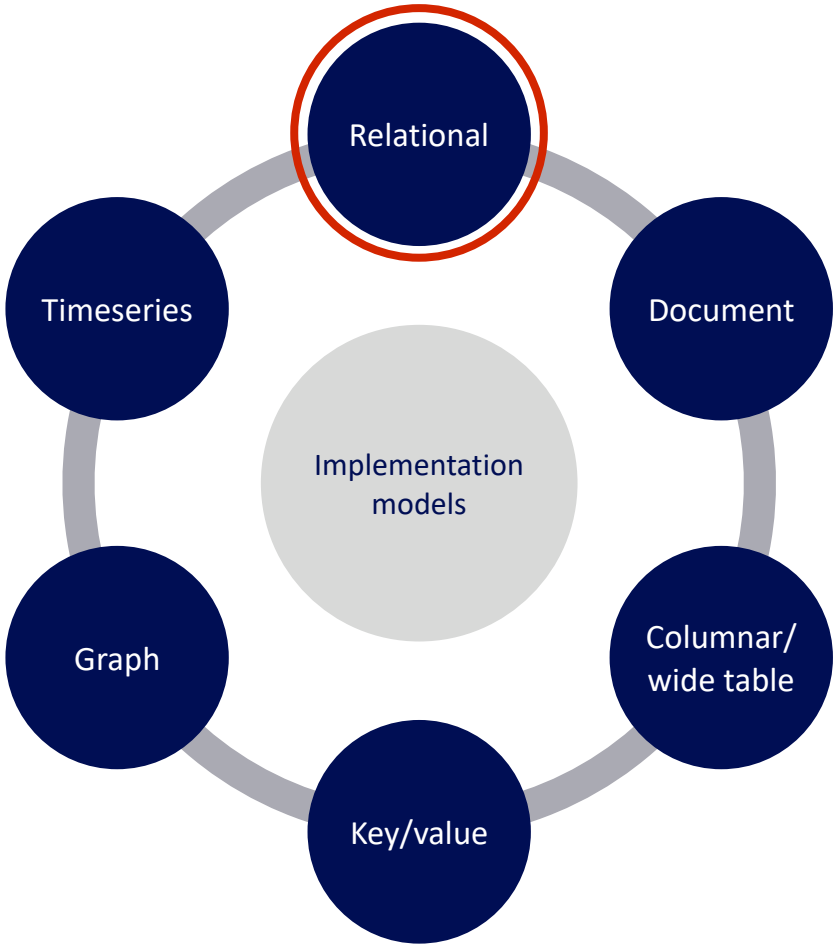


Purpose of the Logical Data Model

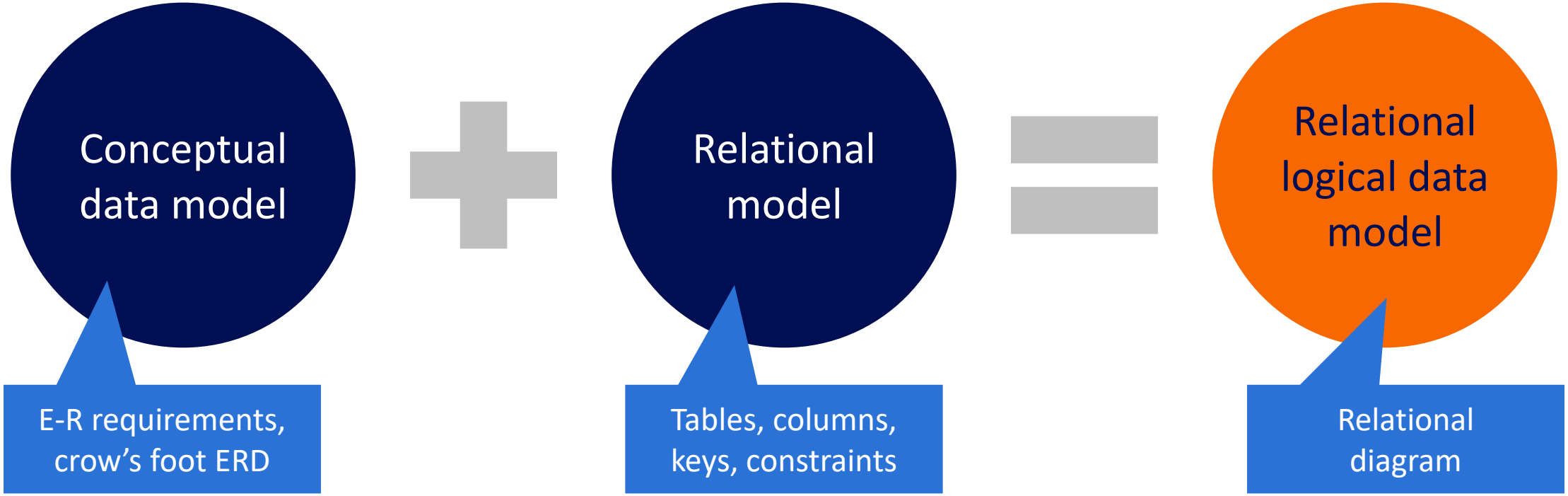
- Technical documentation
- Communication with DBAs and DBDs
- Any skilled individual can read a logical data model and know exactly how to build the database schema from it

Recall: Implementation Models

- A logical model must consider an implementation.
- The approach to logical modeling will depend on the implementation.
  - For example: The way we design a relational logical model is different from a columnar logical model.
- Relational is a good choice, as it is general purpose.



Logical Modeling With Relational

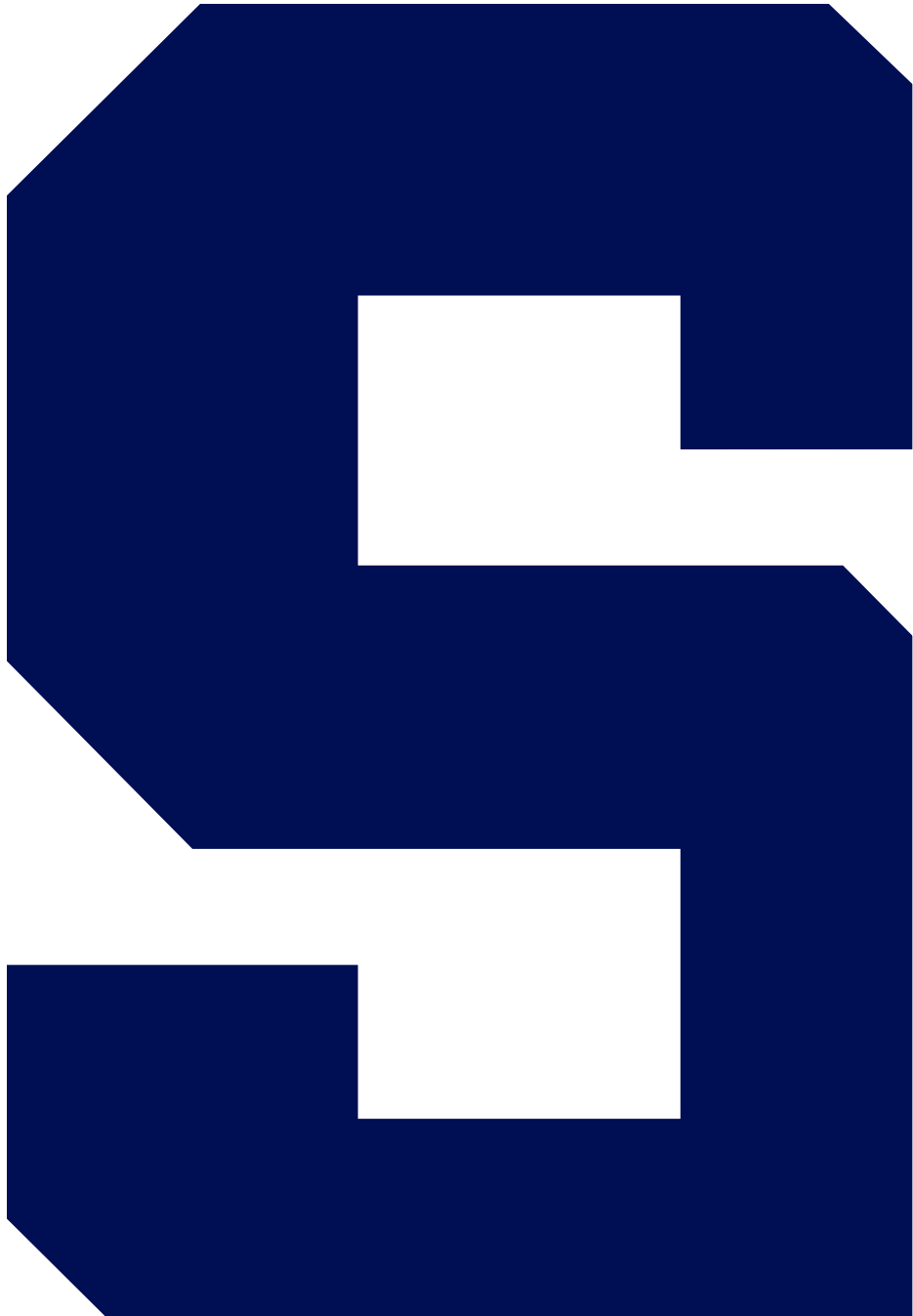


Logical Data Modeling

The End

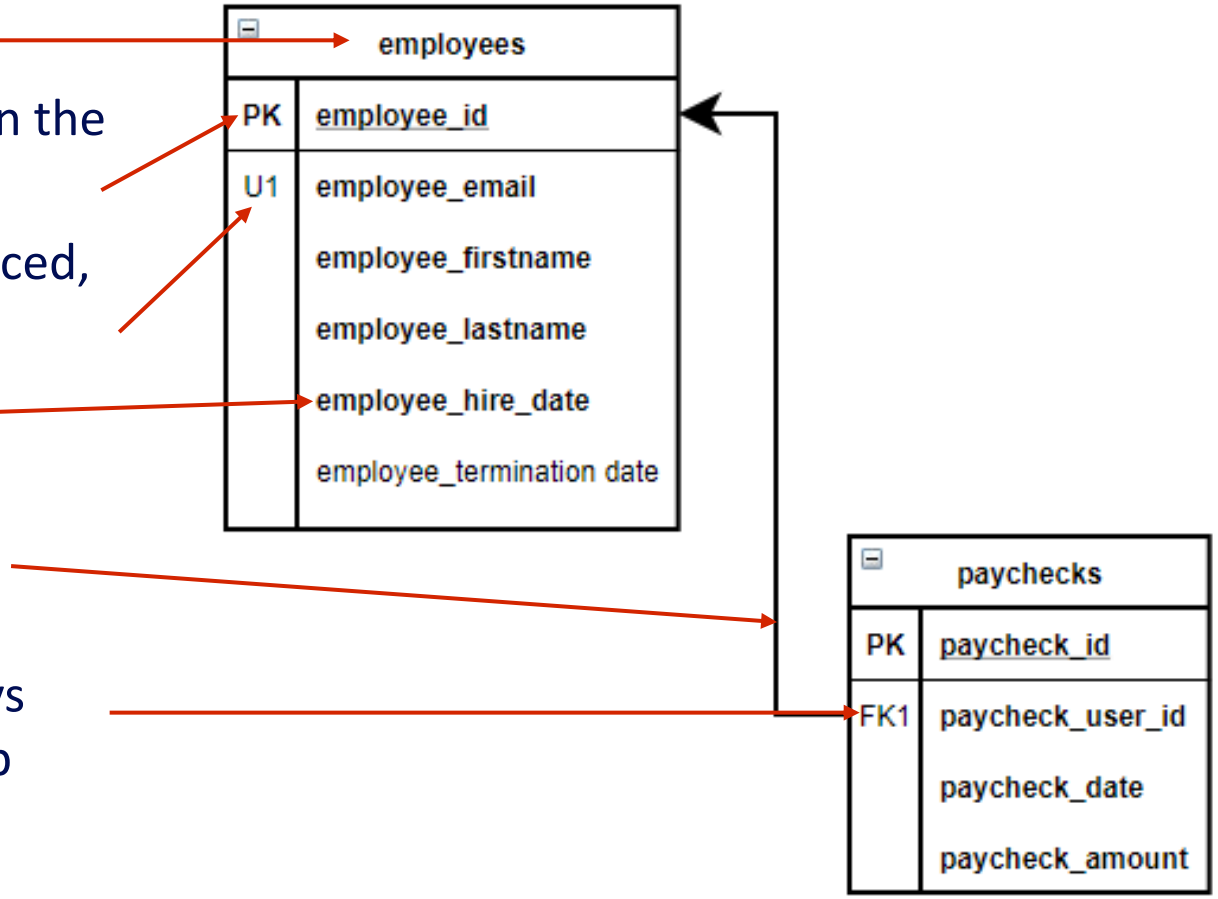
Syracuse University
School of Information Studies

Relational Notation for Logical Models



Relational Notation

- Box is a table; name of table is at the top
- Columns are placed below the table name in the box
- Primary key columns are underlined, boldfaced, and labeled with PK annotation
- Unique constraints are labeled with the U annotation
- Required columns are in boldface
- A PK/FK relationship between tables is established with an arrow (→), which always points to the primary key of the relationship
- Foreign key columns are labeled with FK annotation



Transitioning From Conceptual to Logical

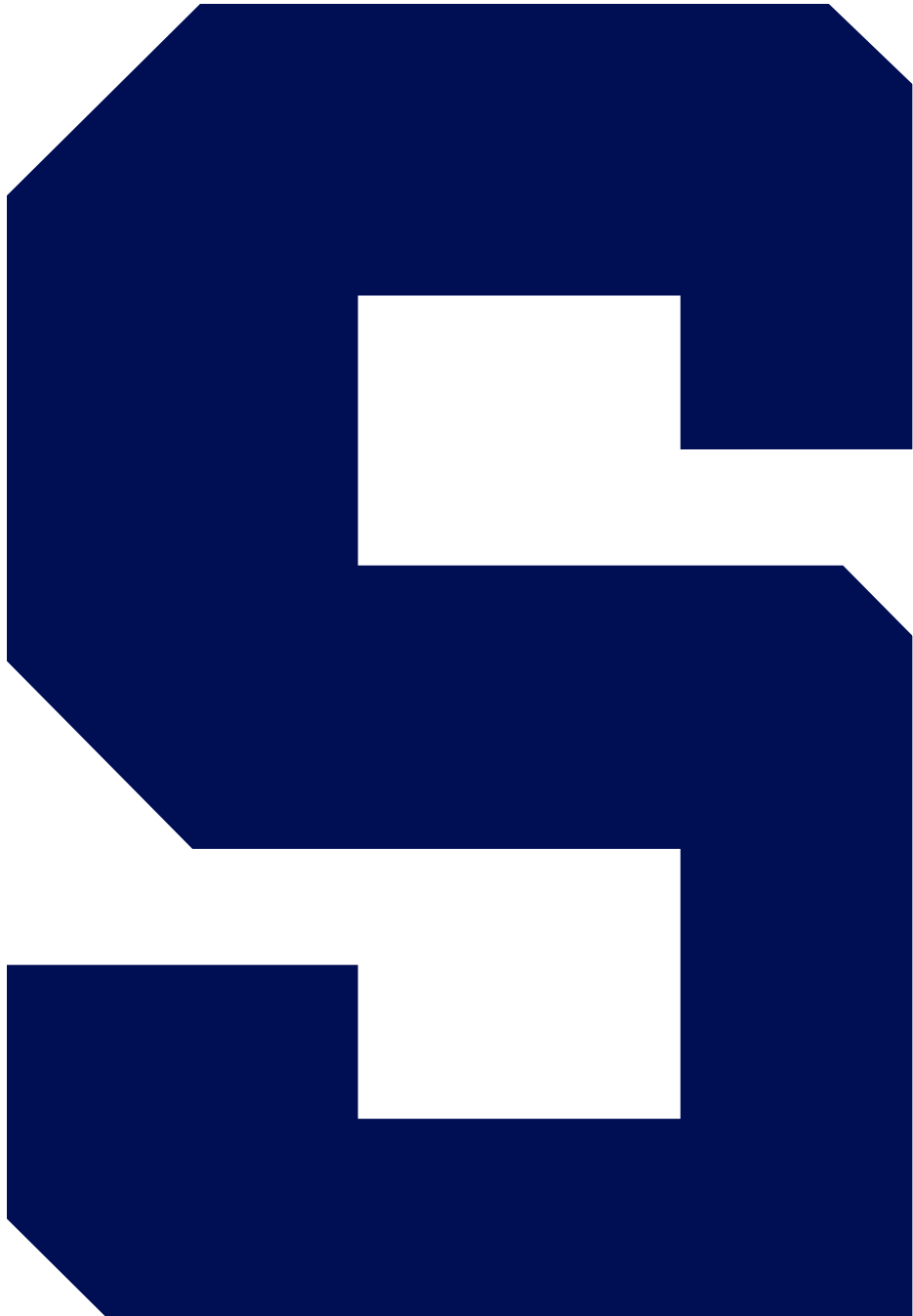
Conceptual	Relational logical
ERD	Relational diagram
Entity set	Relation/table
Entity	Row in table
Attribute	Column in table
Unique property (natural key)	Unique constraint
Required property	Do not allow NULL
Relationship	Foreign key
?	Primary key

Syracuse University
School of Information Studies

Relational Notation for Logical Models

The End

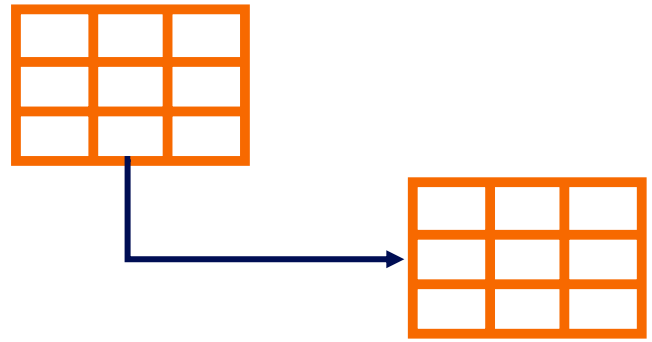
Syracuse University
School of Information Studies



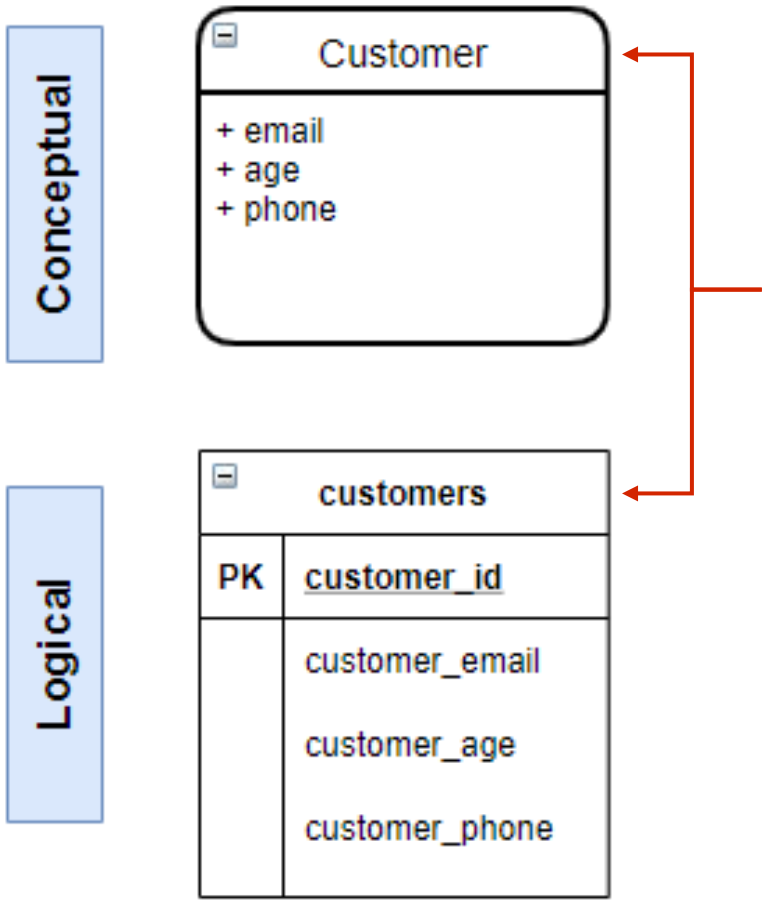
Mapping
Entities and Attributes

Mapping

The process of transforming a conceptual data model into a relational logical data model

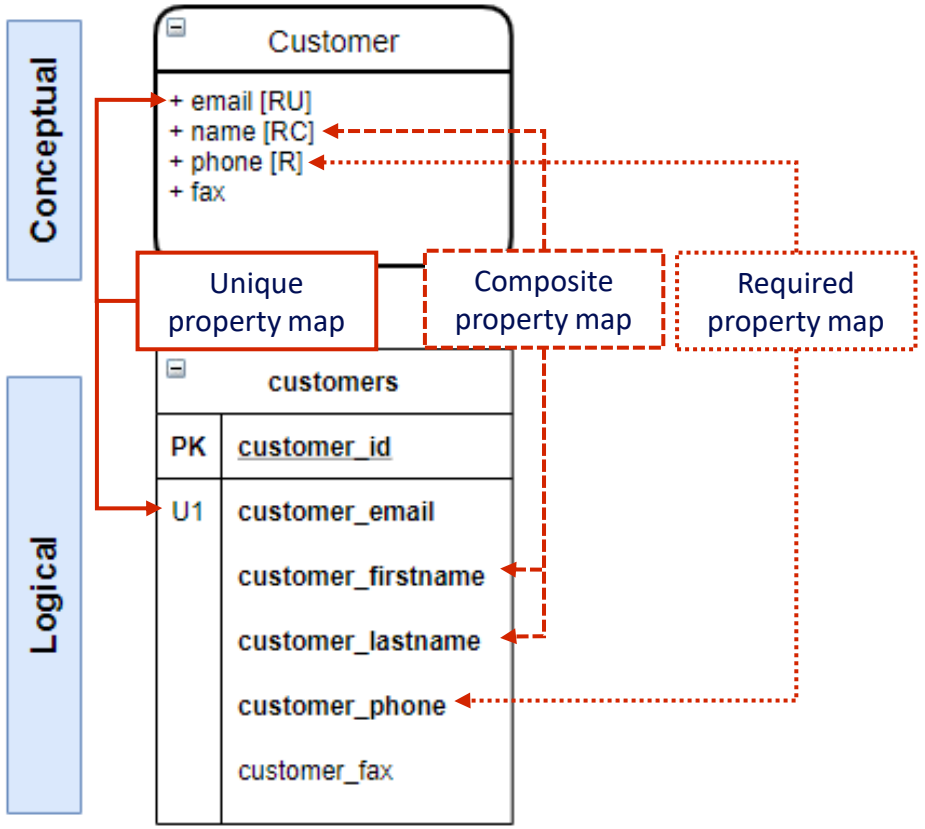


Mapping Entities to Tables



- Entities are singular.
- Tables are plural.
- Tables are like entity sets: They contain more than one entity.
- Logical models are technical documentation, so follow your naming conventions!
- This is exactly how you want the tables created! Foreign key columns are labeled with FK annotation.

Mapping Attributes



- Follow naming conventions.
- Assign primary key (PK) for each table, typically a surrogate key or natural key.
- [U] attribute properties get a unique key constraint.
- [C] attribute properties are broken up into simple attributes.
- [R] attribute properties are not NULL and are represented in boldface.
- Non-PK constraints are annotated and numbered within the table U1, U2, FK1, FK2.

Naming Conventions Are Important!

- Table names: plural, all lowercase; e.g., customers
- Multiple words in any object name: separated with an underscore character; e.g., customer_phone
- Column names: prefixed with the singular of the table name, all lowercase; e.g., customer_last_name
- Surrogate keys: suffixed with _id; e.g., customer_id
- Constraint names: constraint type (pk, fk, u, ck, df) plus table plus column; e.g., pk_customers_customer_id or fk_orders_customer_id

Primary Key Selection

- What does one row in this table mean?
- Always preserve entity integrity!
- When you use a surrogate key, you will need a unique constraint.

student_id	student_name	student_major	student_gpa
1	Bobby	Accounting	3.14
2	Johnny	Finance	4.00
3	Rickey	Informatics	3.51
4	Mike	Music	2.90
5	Mike	Music	2.90

Syracuse University
School of Information Studies

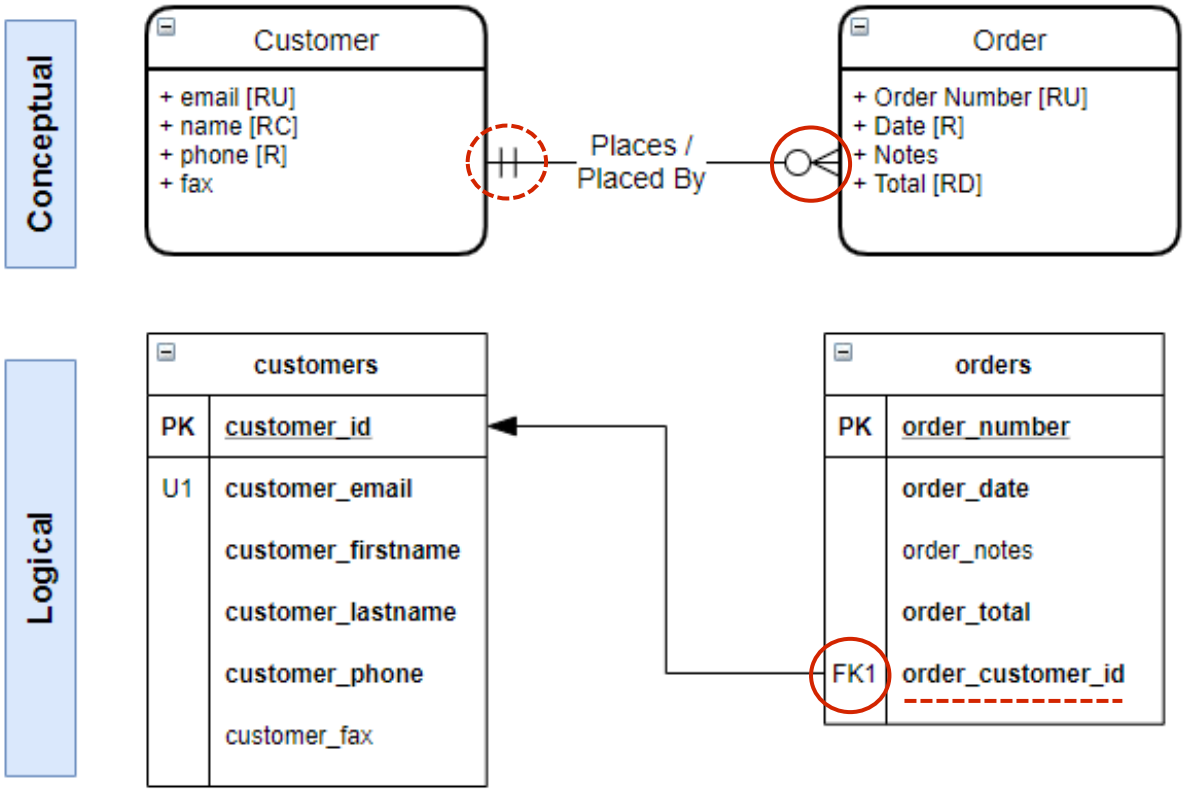
Mapping Entities and Attributes

The End

Syracuse University
School of Information Studies

Mapping 1-M Classifications

Mapping 1-M Classifications



- For relationships with a 1-M classification, the FK is placed on the many side of the relationship
- The one side of the relationship determines if the FK allows NULL:
  FK NULL
  FK NOT NULL

Visualizing Relationships With Rows of Data

A customer places 0 or more orders; an order is placed by 1 and only 1 customer.

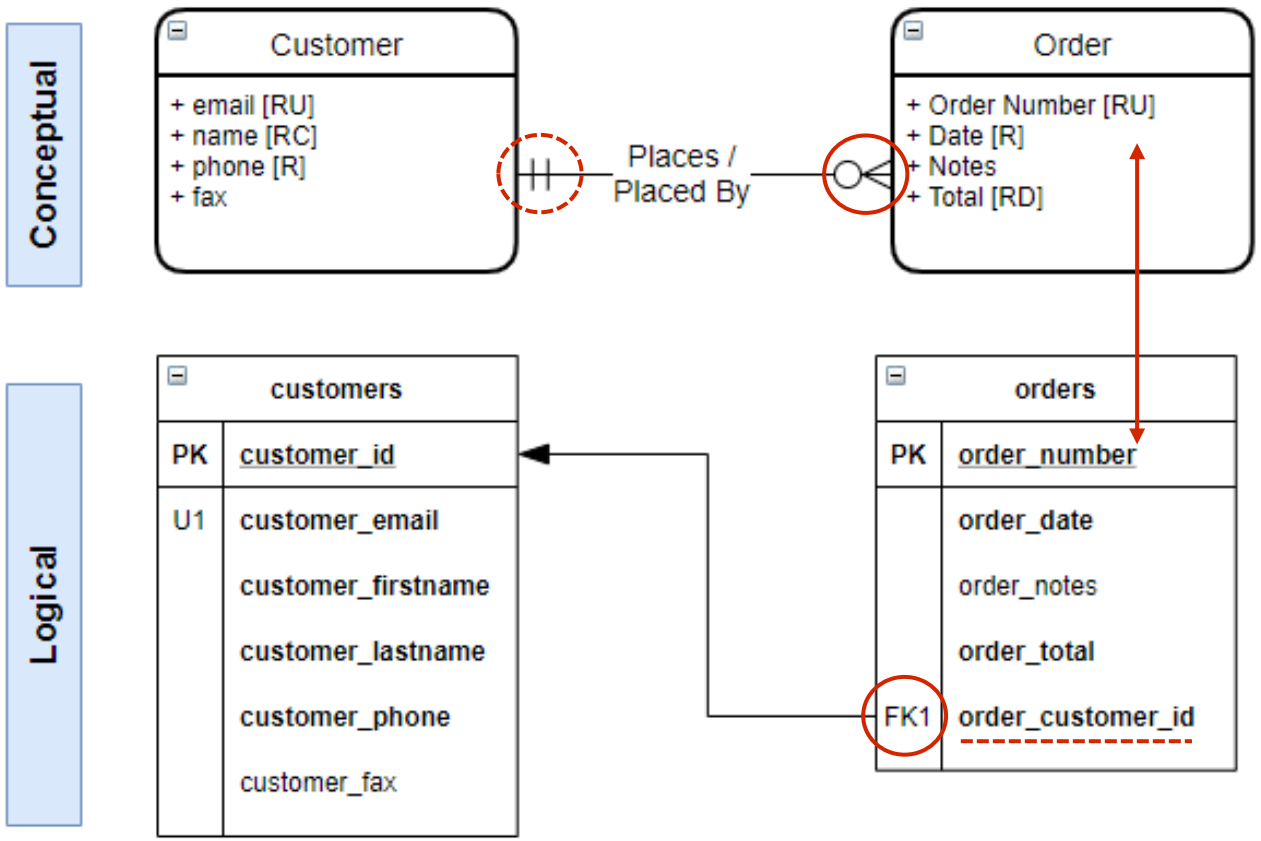
customers

customer_id	customer_email	customer_firs
1	lkarforless@superrito.com	Lisa
2	jking@gustr.com	Joe
3	mmeadows@dayrep.com	Misty

orders

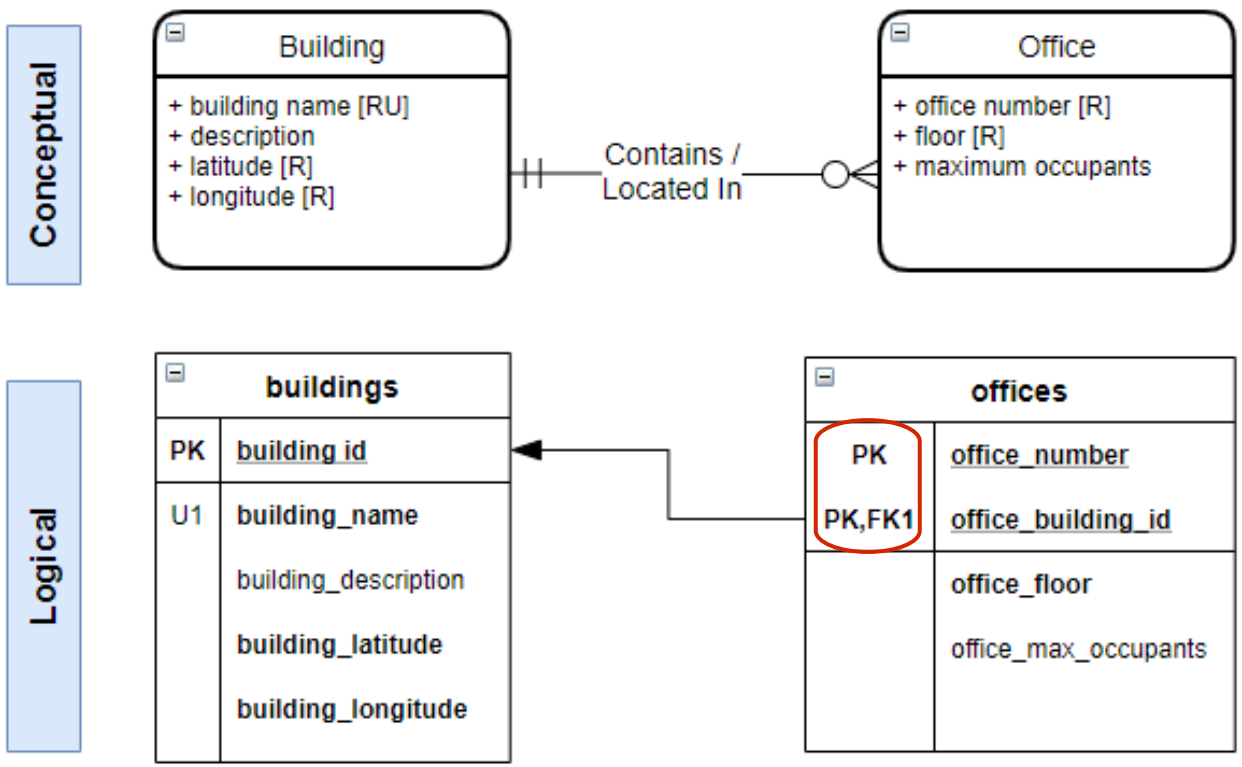
order_number	order_date	order_total	order_customer_id
4	2009-01-02	1.56	1
20	2009-01-08	15.61	2
21	2009-01-09	24.59	3
27	2009-12-10	31.61	3
51	2009-12-20	59.71	3
57	2009-01-22	44.49	2
68	2009-01-26	79.61	3
69	2009-11-27	26.93	1

Mapping 1-M Weak Entity With a Natural Key



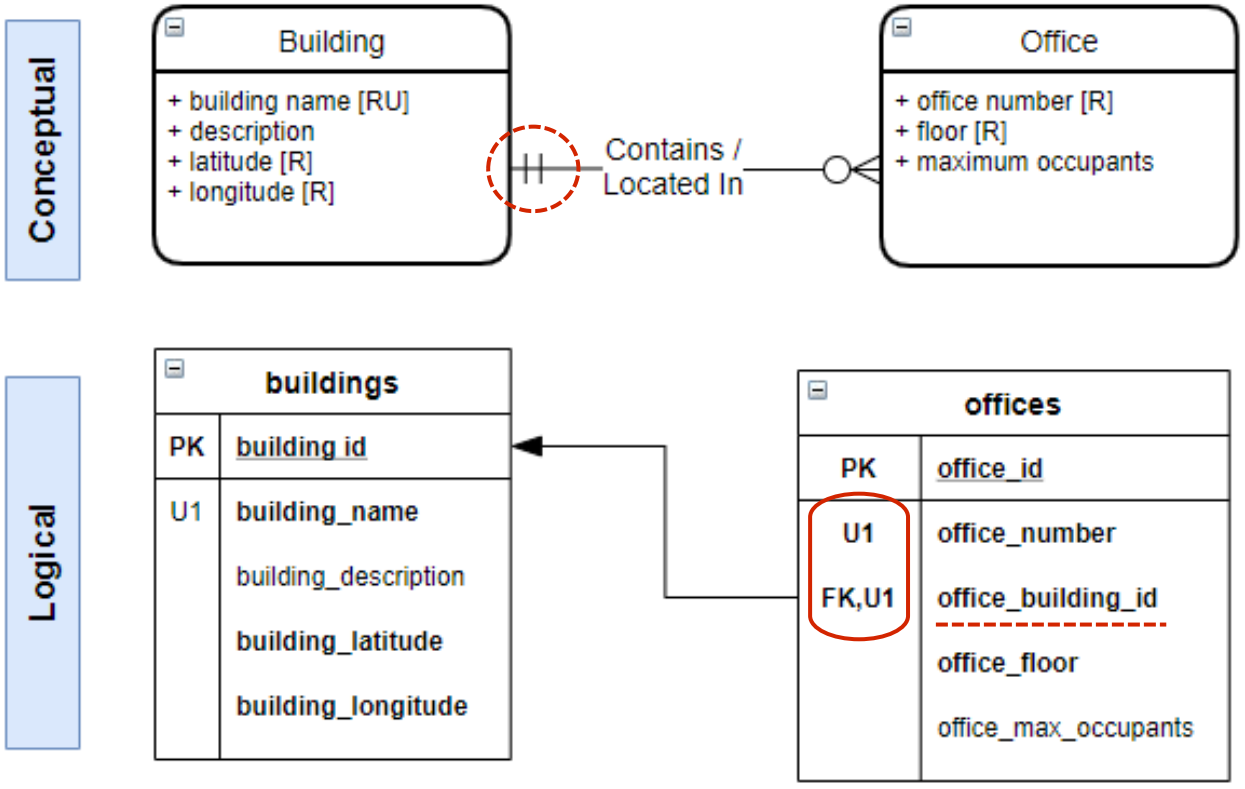
- The natural key is the PK.
- The FK is once again placed on the many side of the relationship.
- A foreign key is required (NOT NULL) to ensure mandatory participation.

Mapping 1-M Weak Entity With No Natural Key



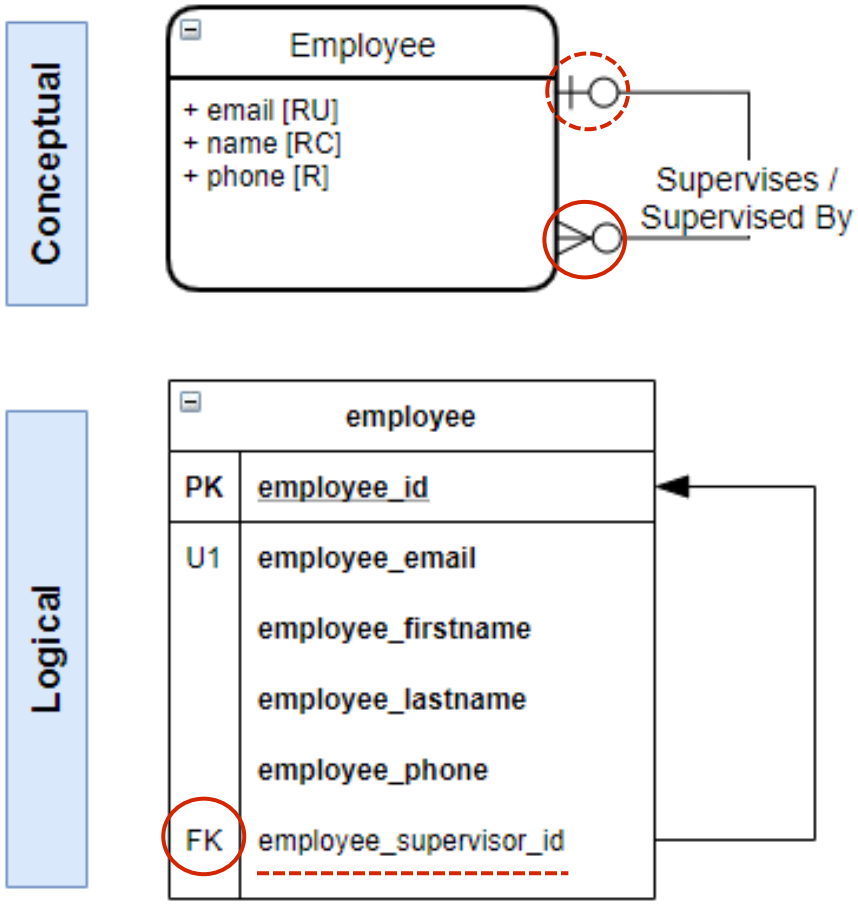
- With no natural key, the weak entity should be mapped as a composite PK (the FK is part of the PK).
- The two values that make up the PK ensure entity integrity.

Mapping 1-M Weak Entity With a Surrogate Key



- If a surrogate PK is chosen for the weak entity, then a unique constraint will be required to ensure entity integrity.
- The unique key should be composite and include the FK.

Mapping 1-M Unary Relationship



- When an entity is related to itself, it is a unary relationship.
- To map, we follow the same rule, placing the FK on the many side, which adds the FK to the same table.
- Notice in this example that NULLS are allowed in the FK.

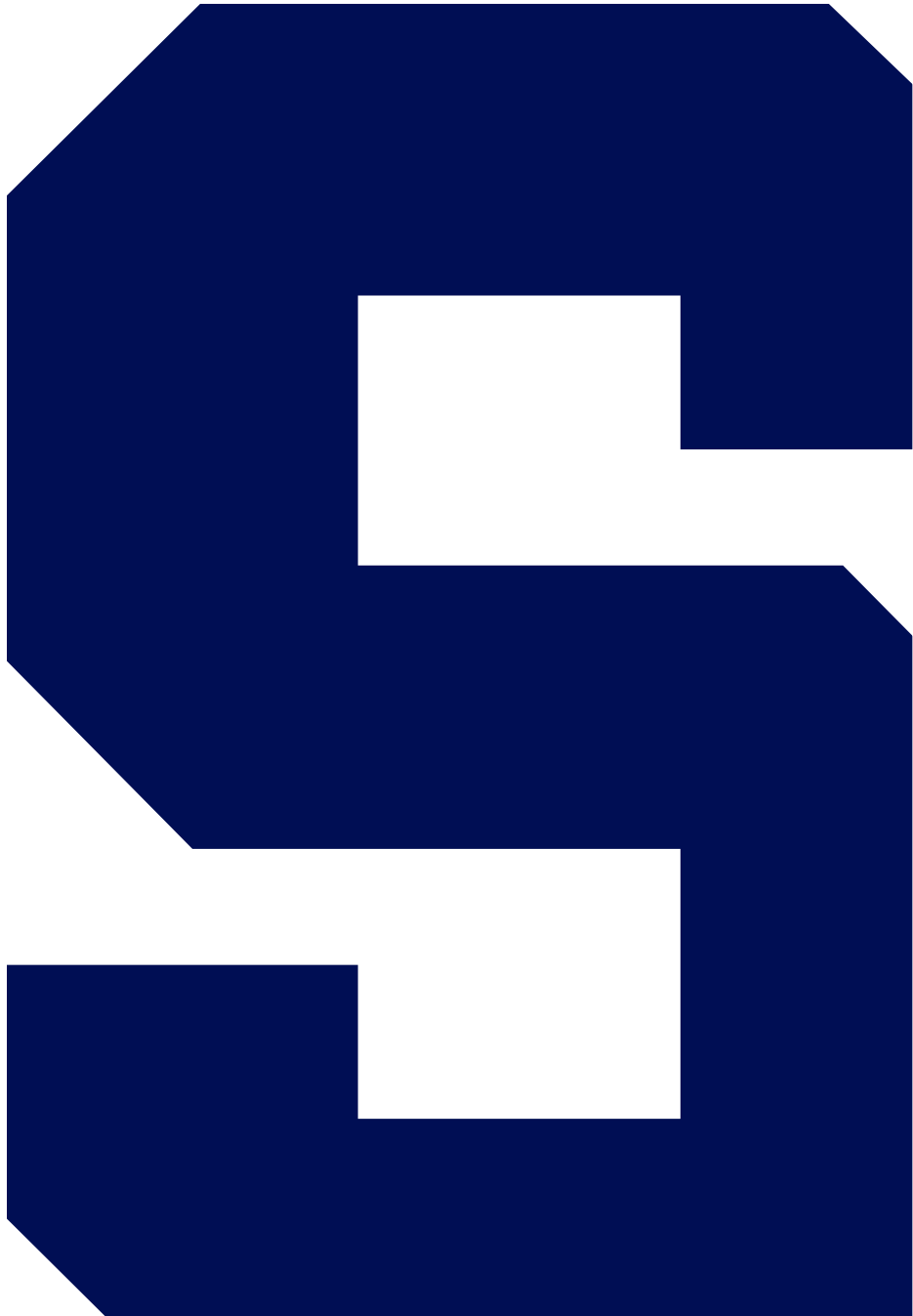
Syracuse University
School of Information Studies

Mapping 1-M Classifications

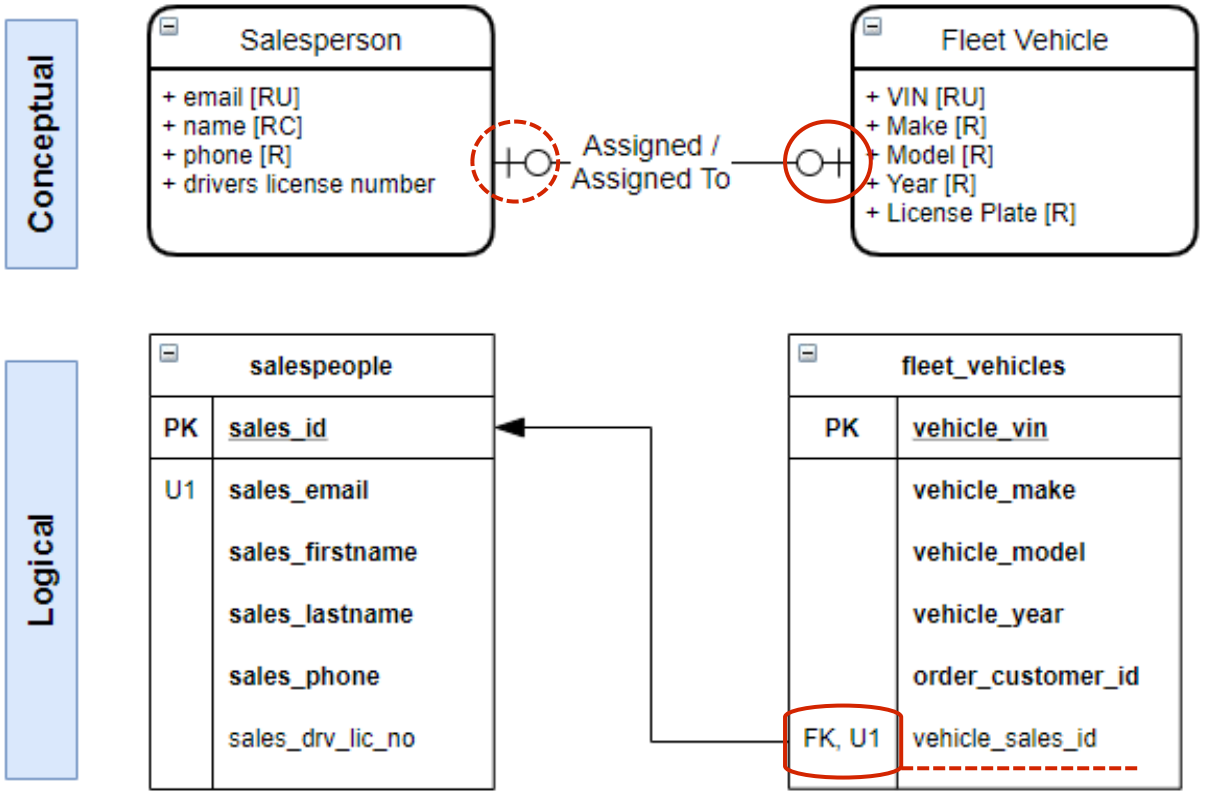
The End

Syracuse University
School of Information Studies

Mapping 1-1 Classifications

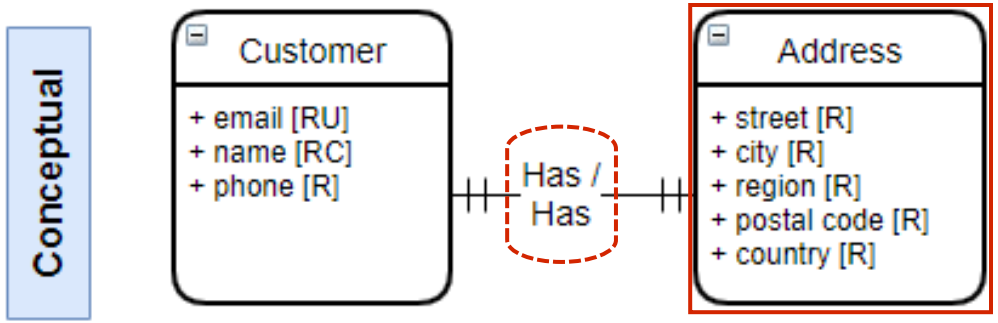


Mapping 1-1 Optional

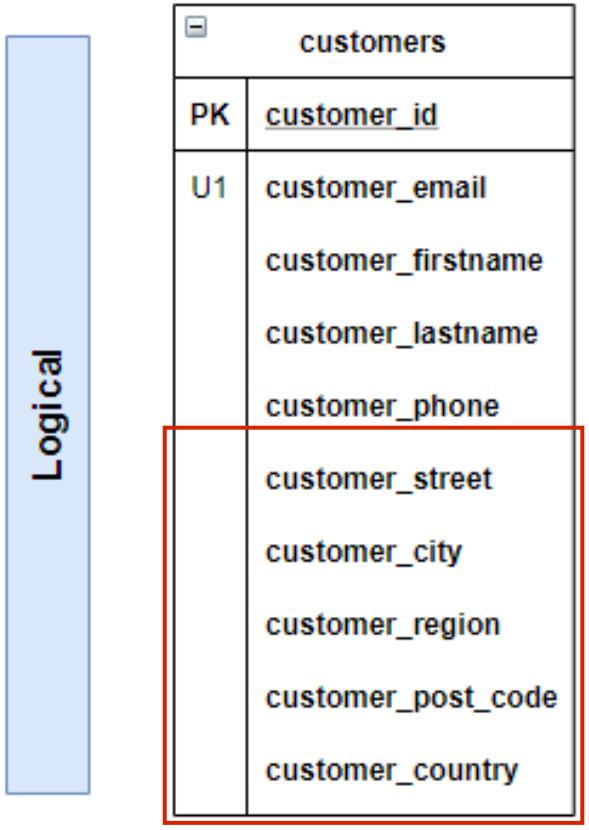


- 1-1 classifications are uncommon.
- FK placement is on the optional side and combined with a unique constraint.
- If both sides are optional, the side with more rows should be the PK side.

Mapping 1-1 Mandatory



- Often, there is no meaningful conceptual business rule.
- Since both sides are mandatory, combine attributes into a single table.



Syracuse University
School of Information Studies

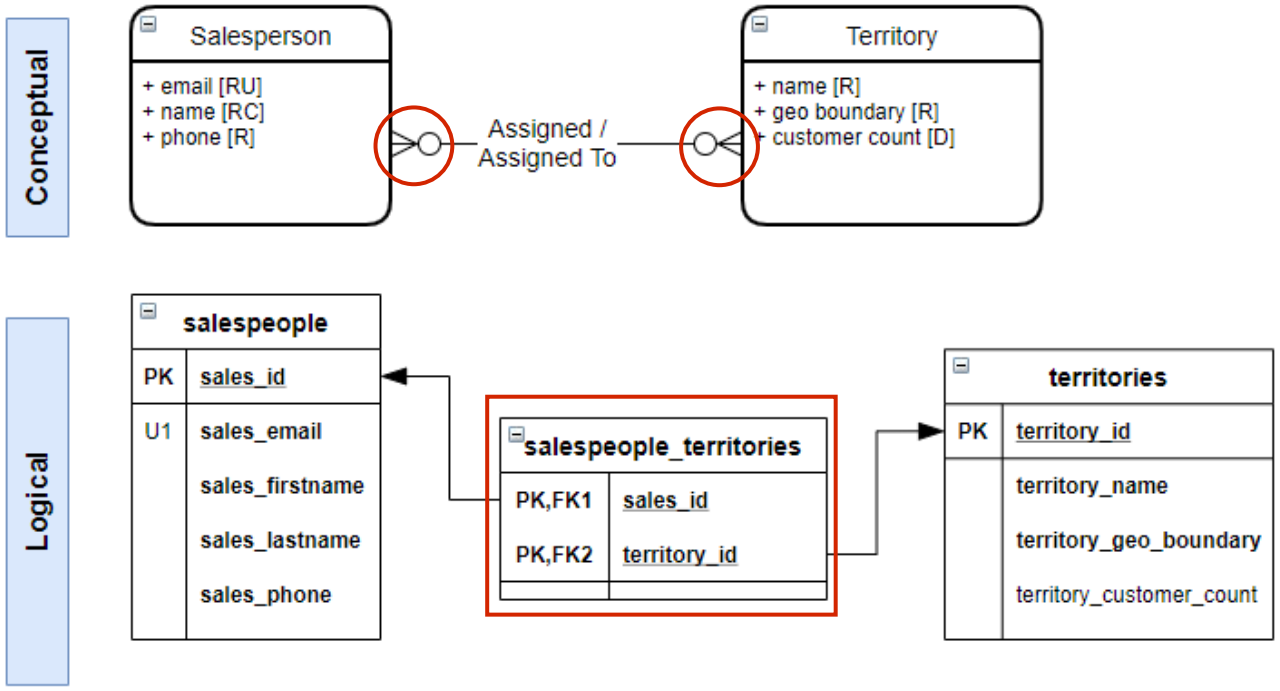
Mapping 1-1 Classifications

The End

Syracuse University
School of Information Studies

Mapping M-M Classifications

Mapping a M-M relationship



- True M-M relationships are mapped with a bridge table, consisting of a composite PK comprised of both FKs.
- The bridge table has no non-key columns.

Visualizing Relationships With Rows of Data

A salesperson is assigned 0 or more territories.

salespeople

sales_id	sales_email	sales_firstna
1	gmoore@gustr.com	Gamble
2	rcarz@gustr.com	Rusty
3	hbarr@gustr.com	Heath
4	jipper@gustr.com	June

salespeople_territories

sales_id	territory_id
1	102
2	101
2	102
2	103
3	102
3	103

territories

territory_id	territory_name
101	New York
102	New Jersey
103	Connecticut

Visualizing Relationships With Rows of Data (cont.)

A territory is assigned 0 or more salespeople.

salespeople

sales_id	sales_email	sales_firstna
1	gmoore@gustr.com	Gamble
2	rcarz@gustr.com	Rusty
3	hbarr@gustr.com	Heath
4	jipper@gustr.com	June

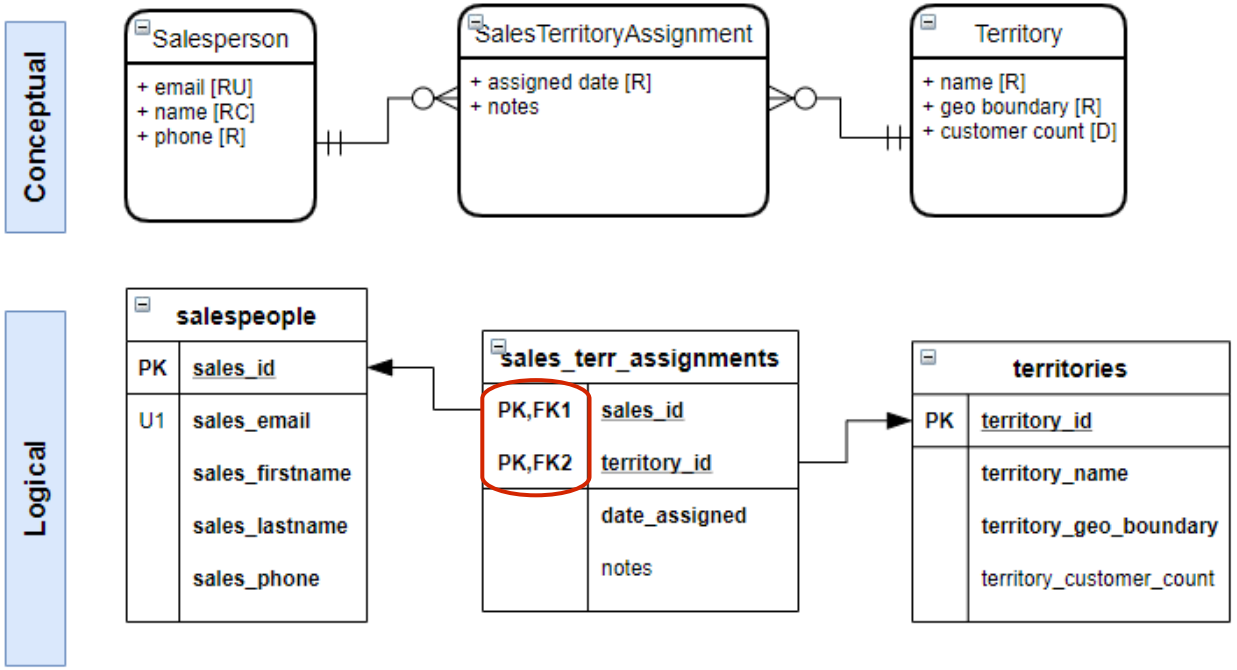
salespeople_territories

sales_id	territory_id
1	102
2	101
2	102
2	103
3	102
3	103

territories

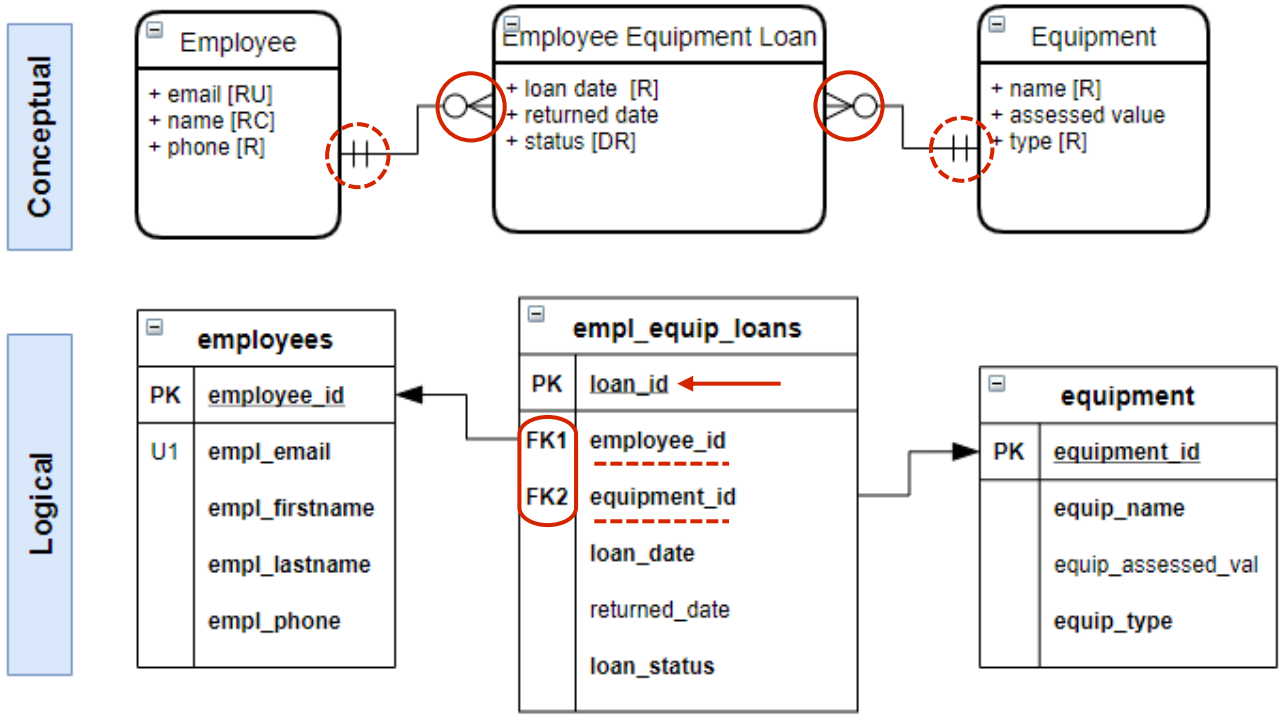
territory_id	territory_name
101	New York
102	New Jersey
103	Connecticut

Mapping Associative Entities: Composite PK



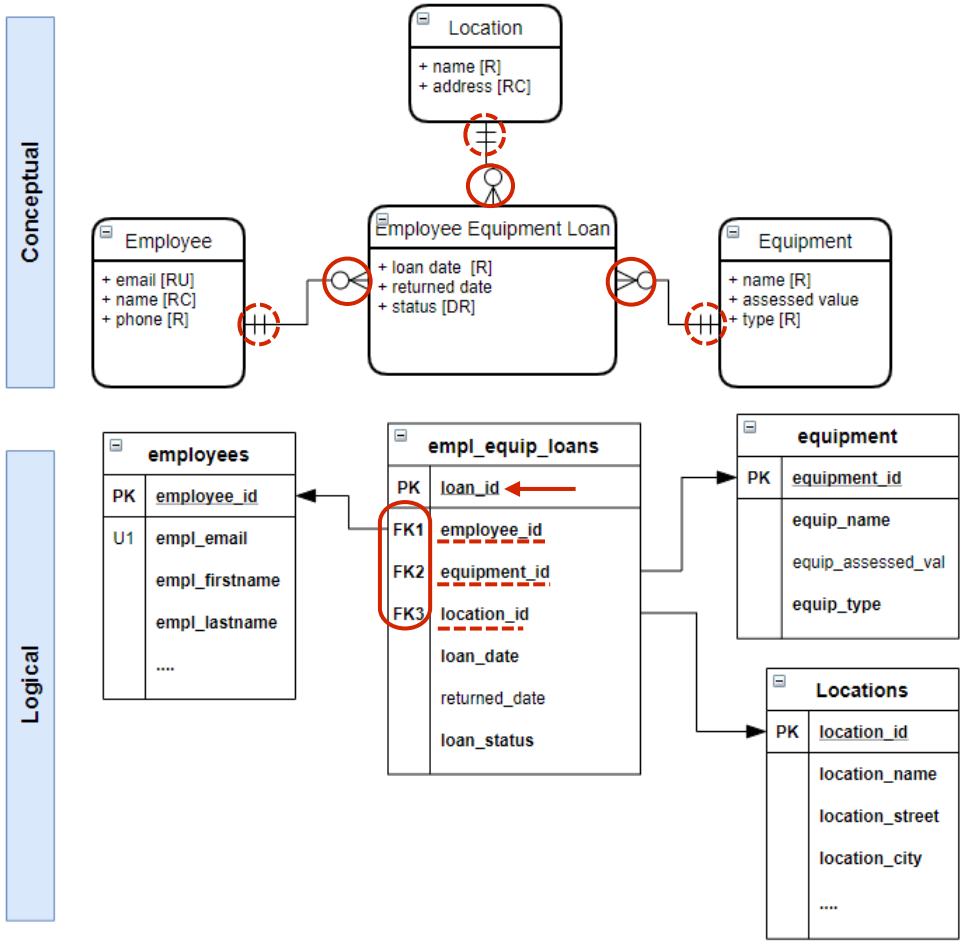
- Associative entities where the two FK values cannot repeat (in order to preserve entity integrity), should be mapped as a bridge table, with a composite PK consisting of both FKs.

Mapping Associative Entities: Surrogate PK



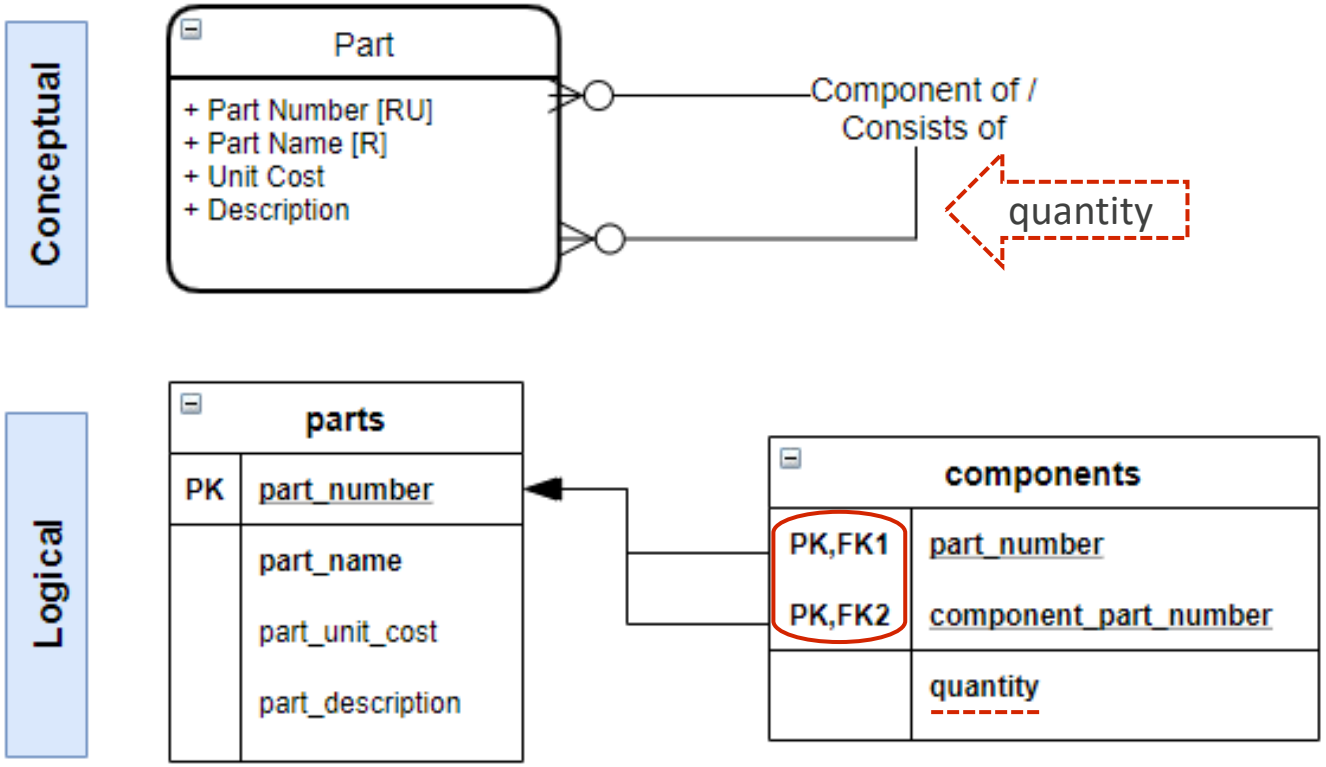
- Associative entities where the two FK values can repeat should be mapped as two regular 1-M relationships.
- The one side is always mandatory.

Mapping Tertiary Relationships



- N-ary relationships are mapped in the same manner as associative entities.
- Once again, the decision to use a composite PK is based on whether the values can repeat.
- In this case, we used a surrogate key.

Mapping a M-M Unary Relationship



- Map this relationship as a bridge table with itself.
- These are most often associative entities and have attributes on the relationship.

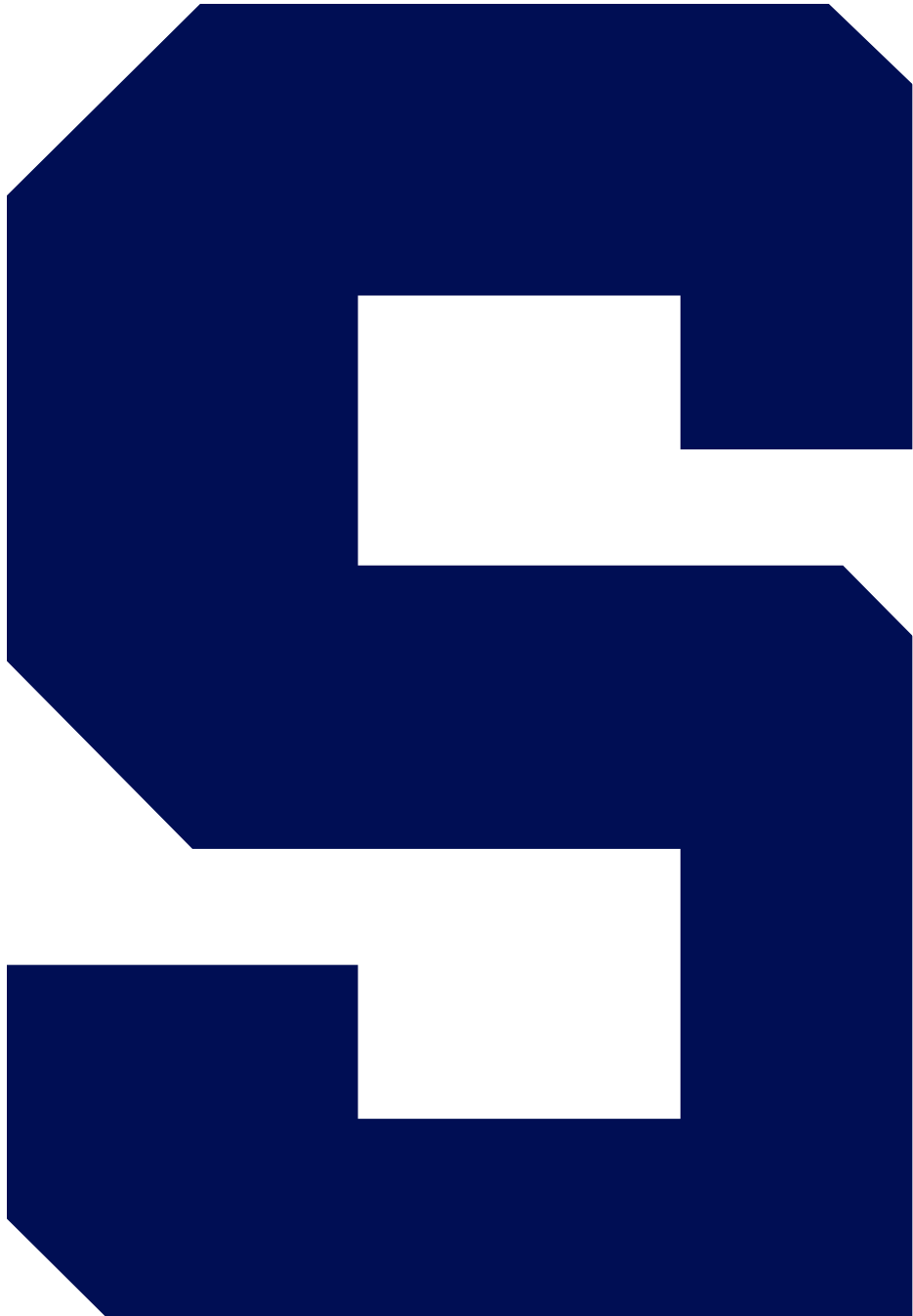
Syracuse University
School of Information Studies

Mapping M-M Classifications

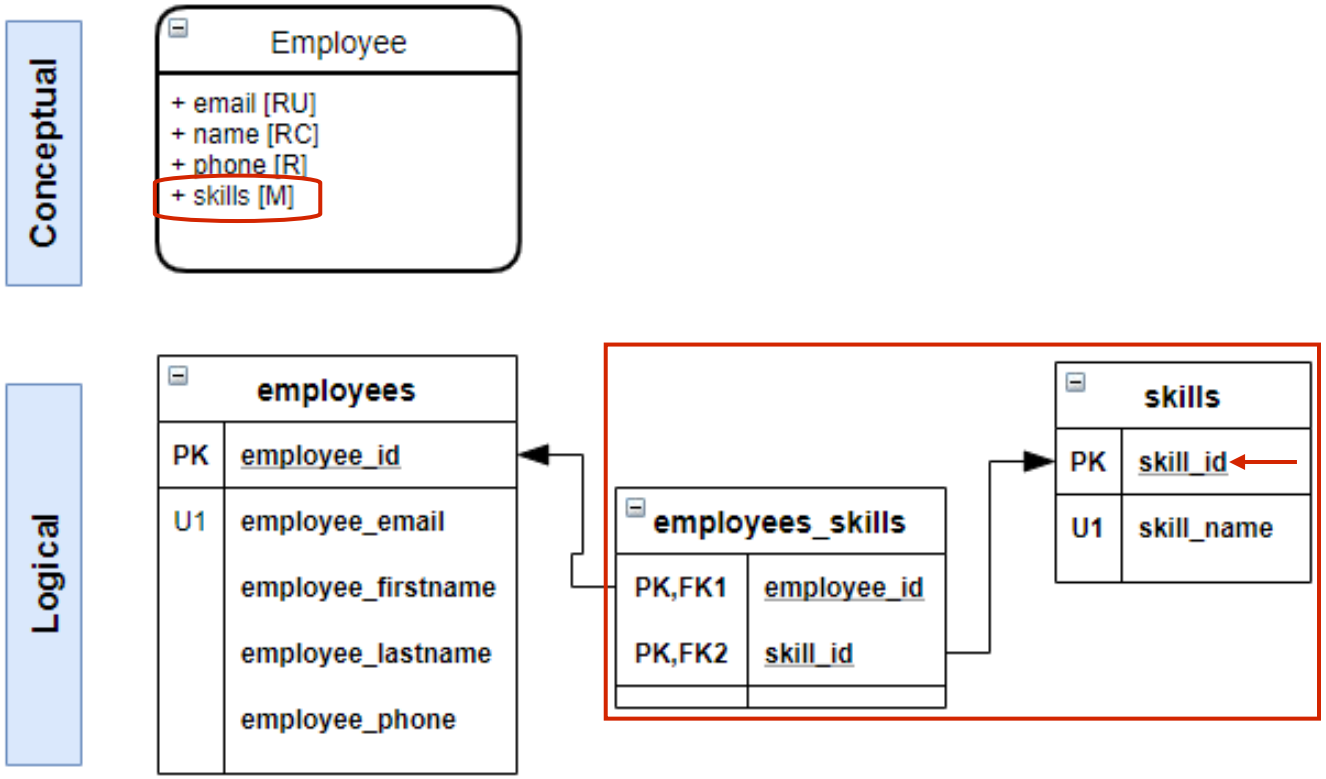
The End

Syracuse University
School of Information Studies

Advanced Attribute Mapping

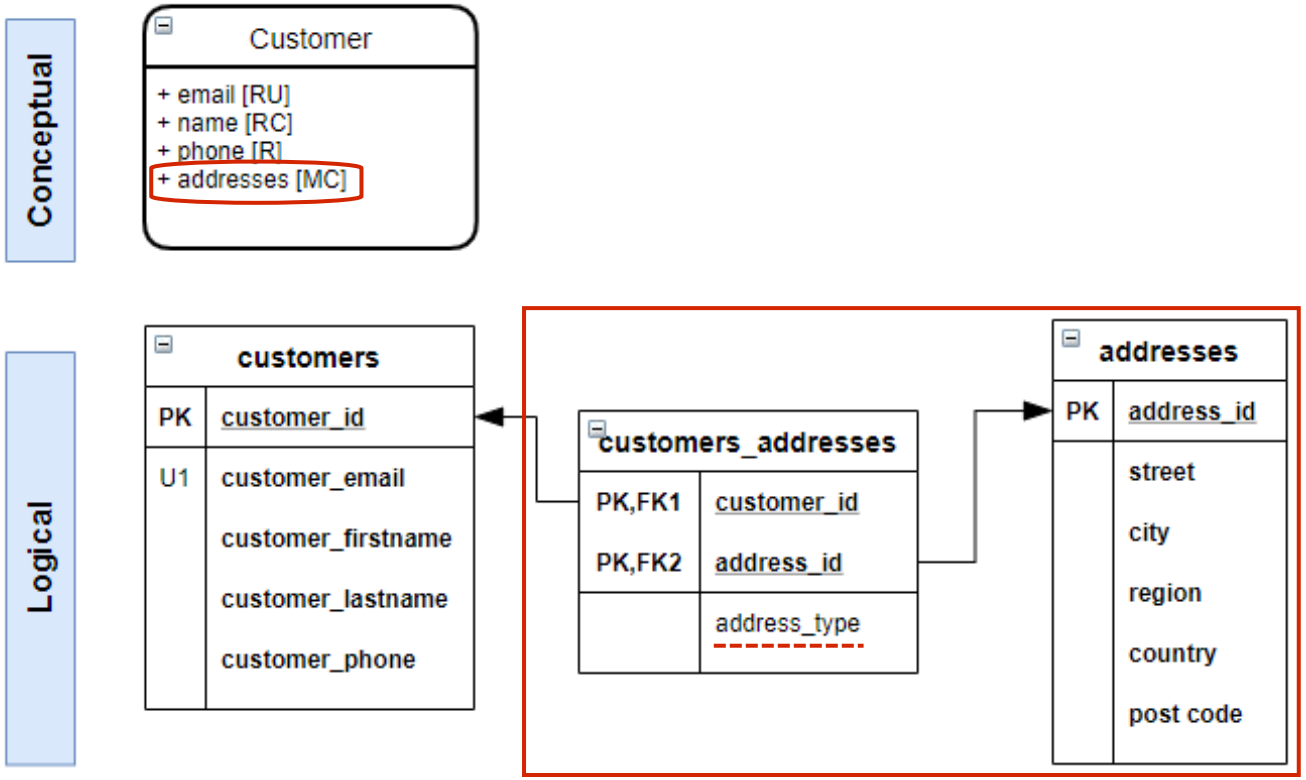


Mapping M Attributes



- Treat the M attribute as a lookup table, then follow the M-M relationship rule.
- The PK can be the attribute itself or a surrogate key, as in this example.

Mapping MC Attributes



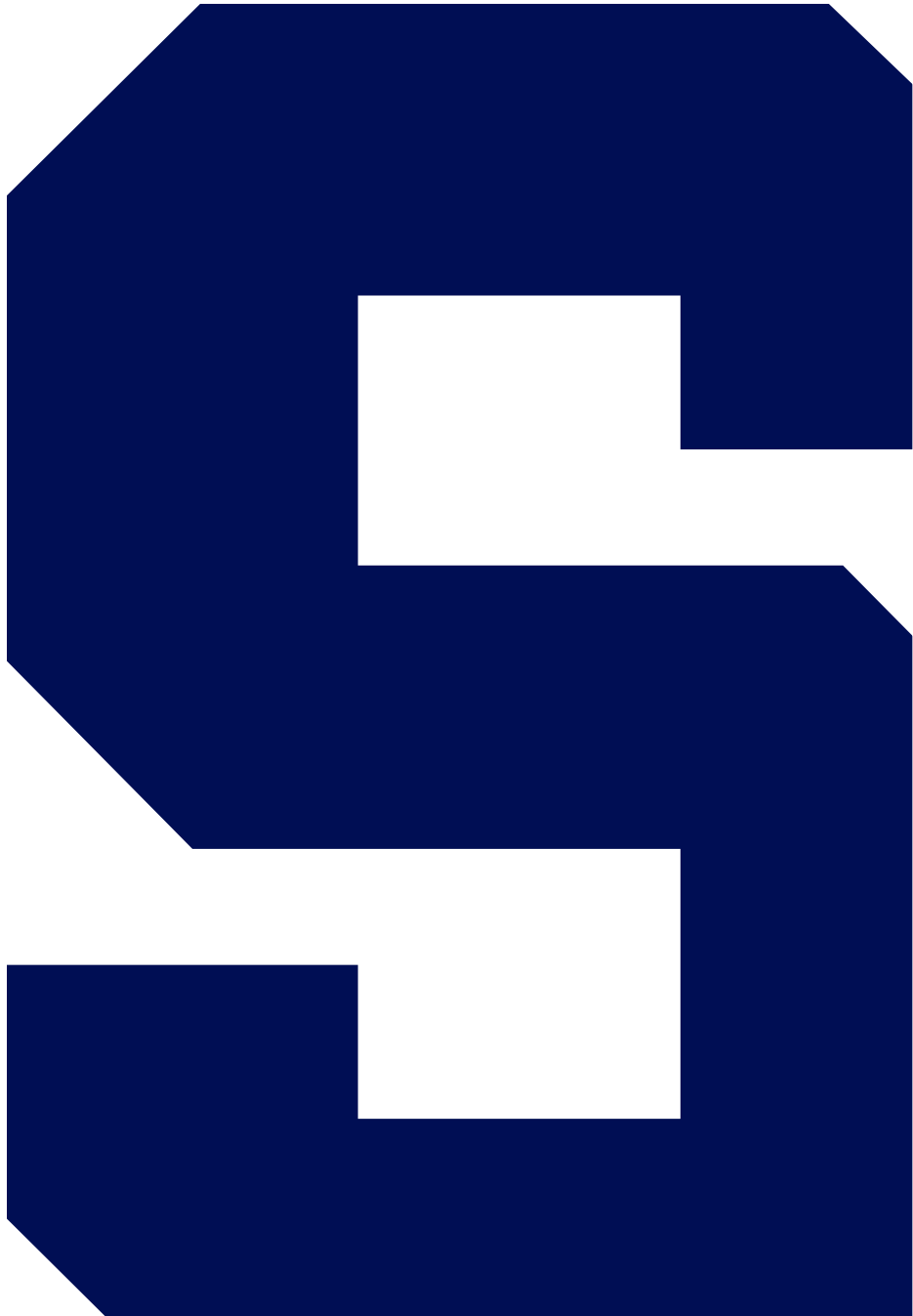
- For MC attribute mapping, treat the attribute as its own table, then join back to the original table in and M-M relationship.
- Include the type discriminator if needed.

Syracuse University
School of Information Studies

Advanced Attribute Mapping

The End

Syracuse University
School of Information Studies

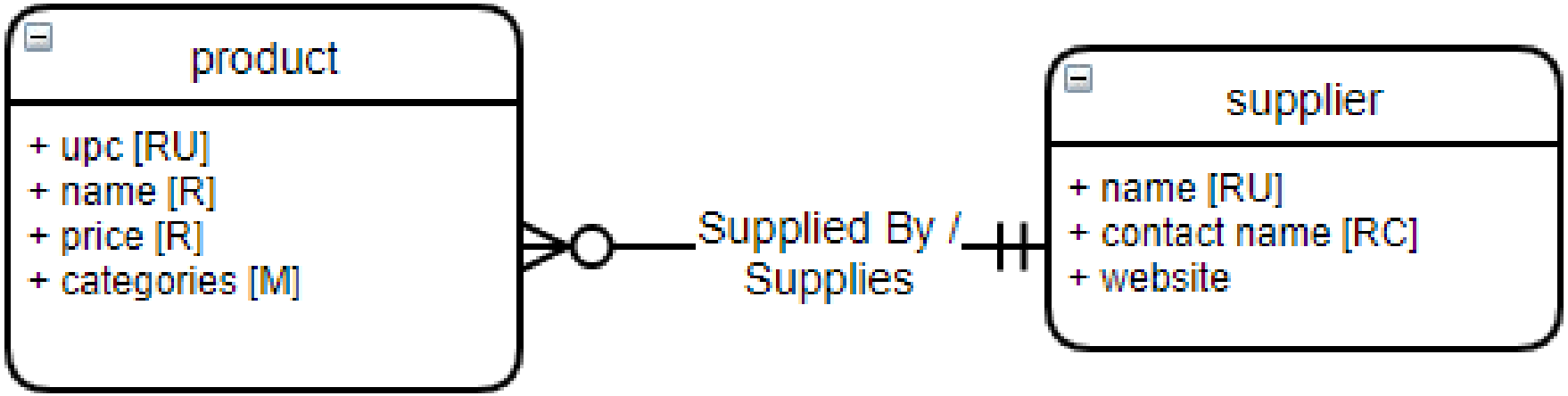


Demo

Drawing a Logical Model

Demo: Draw a Logical Model

- We will use Draw.io from Diagrams.net to draw the diagram
- Conceptual E-R requirements



Syracuse University
School of Information Studies

Demo: Drawing a Logical Model

The End

Syracuse University
School of Information Studies

Summary

Summary

- Data requirements are functional requirements that address the storage of data.
- Data requirements can be formalized using entity-relationship modeling.
- E-R models can be drawn with a crow’s foot diagram.
- Logical data modeling is a technical specification for how a database should be built. It considers an implementation model but is not an actual implementation.
- Entities map to tables; attributes map to table columns.
- When mapping 1-M relationships, the FK goes on the M side, and the 1 side cardinality determines NULL in the FK.
- Mapping M-M relationships requires a bridge table with a composite PK.

Syracuse University
School of Information Studies

Summary

The End