Week 4 Workshop

Housekeeping information

- SQL Assessment (Assignment 1) will be available on Wattle 23:59 tonight, and the submission via Wattle is due 23:59 Sep 3 (Friday, Week 6)
  - **Individual, no group work!**
  -
  -
  -
  -
- Partial marks may be awarded.

Housekeeping information

- SQL Assessment (Assignment 1) will be available on Wattle 23:59 tonight, and the submission via Wattle is due 23:59 Sep 3 (Friday, Week 6)
  - **Individual, no group work!**
  -
  - Partial marks may be awarded.
- Drop in sessions for Assignment 1 (Week 5 and W
  - Aug 23 (Mon) 2-3 pm (NEW)
  - Aug 24 (Tue) 2-3 pm
  - Aug 25 (Wed) 8-9 pm (NEW)
  - Aug 30 (Mon) 2-3 pm (NEW)
  - Aug 31 (Tue) 2-3 pm
  - Sep 1 (Wed) 8-9 pm (NEW)

Database Design – Four Phases

- Th

2. **Conceptual Design**
   Entity-Relationship Model

3. **Logical Design**
   From Entity-Relationship Model to Rel

4. **Physical Design**

Phase 2: Conceptual Design

- **Conceptual design** is the process of constructing a conceptual data model that is

  - used to communicate the requirements of a nontechnical users.

- A conceptual data model is built using the informati requirements specification.

Note: The conceptual design is based on the **Entity-Relationship Model** in this course.

Model and Modeling

- What is a model?

Model and Modeling

- What is a model?

A model is

-

Model and Modeling

- What is a model?

  A model is

  -
  -
  -

- What does modeling do?

Model and Modeling

- **What is a model?**

  A model is

  -
  -
  -

- **What does modeling do?**

  Modeling

  - creates an understanding and relationsh
  - helps in conceptualising and visualising th
    we may want to build.
  - facilitates specifications of the behaviour of a system
  - gives rise to a template that guides us in constructing a system
  - . . .

Entity-Relationship (ER) Model

- ER diagrams (Peter Chen in 1976):

  - **Key attribute** with *underlined*;

  - **Entity** as *rectangles*;
  - **Relationship** as *diamonds*.

(Exercise 1) Consider the following data requirements for a university student database that is used to keep track of students' transcripts.

- The university keeps track of each student's name, student number, social security number, address, phone, and birthdate. Both social security number and student number have unique values for each student.
- E
de
- E
an
- Each course has a course name, description, cour
semester hours, level, and offering department
is unique for each course.
- Each section of a course has an instructor, semest
number and the section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.
- A grade record refers to each student and a particular section, consisting of a final mark and a letter grade from (F, D, C, B, A).

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

  E

  phone and birthdate. Both social security number

  unique values for each student.

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

  E

  phone and birthdate. Both social security number
  unique values for each student.

Question: What are the entities, relationships an

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

E
phone and birthdate. Both social security number
unique values for each student.

Question: What are the entities, relationships an

- **Entities:** STUDENT
- **Relationships:**
- **Attributes**: name, student number, social security number, address, phone and birthdate for STUDENT

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

E

departments

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

E

departments

Question: What are the entities, relationships an

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

E

departments

Question: What are the entities, relationships an

- **Entities:** STUDENT, DEPARTMENT
- **Relationships:** has_major_with betw

  DEPARTMENT, has_minor_with between STUDENT and DEPARTMENT
- **Attributes**: name for has_major_with , name for has_minor_with

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P**
  an
  ty

ce of

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints on relationsh ... ?

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints or relationsh nt

Cardinality ratios: Every student has at most may offer **many** majors (to different students)

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints or relationsh nt

Cardinality ratios: Every student has at most may offer **many** majors (to different students)

Participation constraints: Every student **must** have one major (**total**) and each department **must** (typically) offer one major (**total**).

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints on relationsh ?

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints or relationsh ? nt

Cardinality ratios: Every student has at most may offer **many** minor (to different students)

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each student has exactly one major, and may hav departments.

Question: What are the constraints or relationsh ? nt

Cardinality ratios: Every student has at most may offer **many** minor (to different students)

Participation constraints: Every student **may or may not** have one minor (**partial**) and each department **must** (typically) offer one minor (**total**).

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the entities, relationships an

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the entities, relationships an

- **Entities:** course

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **R**
- **At**

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the entities, relationships and

- **Entities:** course, department
- **Relationships:** offer (between **department** and **course**)
- **Attributes**: course name, description, course number, number of semester hours and level (of the entity **course**)

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P**
  an
  ty

ce of

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P**
  an
  ty

  ce of

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P**
  an
  ty

  ce of

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the constraints on relationsh

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the constraints on relationsh

Cardinality ratios: Every course is offered by at t and a department may offer **many** courses

Constraints on Relationships

- **Cardinality ratios**: Specifies the *maximum* number of relationships that an entity can participate in.
- **P** ce of an ty

Each course has a course name, description, course number, number of semester hours, level, and offering department.

Question: What are the constraints on relationsh t and

Cardinality ratios: Every course is offered by at a department may offer **many** courses

Participation constraints: Every course **must** be offered by some department (**total**) and each department **may (or may not)** offer any courses (**partial**).

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **Relationships:** Associations between entities.
- **Attributes:** Properties that describe entities and relationships.

E
nu
co
...,

A grade record refers to each student and a particul
a final mark and a letter grade from (F, D, C, B, A).

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **Relationships:** Associations between entities.
- **Attributes:** Properties that describe entities and relationships.

E
nu
co
...,

A grade record refers to each student and a particul
a final mark and a letter grade from (F, D, C, B, A).

Question: What are the entities, relationships an

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **Relationships:** Associations between entities.
- **Attributes:** Properties that describe entities and relationships.

E
nu
co
...,

A grade record refers to each student and a particul
a final mark and a letter grade from (F, D, C, B, A).

Question: What are the entities, relationships an

- **Entities:** section, course, student

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **Relationships:** Associations between entities.
- **Attributes:** Properties that describe entities and relationships.

E
nu
co
...,

A grade record refers to each student and a particul
a final mark and a letter grade from (F, D, C, B, A).

Question: What are the entities, relationships an

- **Entities:** section, course, student
- **Relationships:** section_taught (between **section** and **course**), grade_record (between **student** and **section**)

Entities, Relationships and Attributes

- **Entities:** "Things" in the real world (with independent existence).
- **Relationships:** Associations between entities.
- **Attributes:** Properties that describe entities and relationships.

E
nu
co
...,

A grade record refers to each student and a particul
a final mark and a letter grade from (F, D, C, B, A).

Question: What are the entities, relationships an

- **Entities:** section, course, student
- **Relationships:** section_taught (between **section** and **course**), grade_record (between **student** and **section**)
- **Attributes**: instructor, semester, year, and section number (of the **weak** entity **section**), final mark and letter grade (of the relationship **grade_record**)

(Exercise 1) Consider the following data requirements for a university student database that is used to keep track of students' transcripts.

- The university keeps track of each student's name, student number, social security number, address, phone, and birthdate. Both social security number and student number have unique values for each student.
- E
de
- E
an
- Each course has a course name, description, cour
semester hours, level, and offering department
is unique for each course.
- Each section of a course has an instructor, semest
number and the section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.
- A grade record refers to each student and a particular section, consisting of a final mark and a letter grade from (F, D, C, B, A).

Constructing an ER or EER Model

- Identify the entities (including weak entity types)

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  student, course, department, section (weak entity)

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  student, course, department, section (weak entity)
- Id

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  student, course, department, section (weak entity)

- Id
  - **has_major** (between **student** and **department**)
  - **offer** (between **department** and **c**
  - **section_taught** (between **section**
  - **grade_record** (between **student** a

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  student, course, department, section (weak entity)
- Id
  -
  - **has_major** (between **student** and **department**)
  - **offer** (between **department** and **c**
  - **section_taught** (between **section**
  - **grade_record** (between **student** a
- Identify the attributes of entities and relationships and identify a primary key for each entity type
- Identify cardinality ratios and participation constraints on relationships

Software tool to draw ER diagram

- We require students to use an academic tool, TerraER, to draw the ER di
- T E
- You can download the jar file from the following website: `https://github.com/rterrabh/TerraER ER3. 01/TerraER3.01beta.jar`
- You can double-click that file to execute on Windo that the Java Runtime Environment JRE has been i
- More information on how to use TerraER will be provided next week.

(Exercise 2) A retailer company wants to build a database application for managing information about its sale process. The company sells products in both local shops and webstores on the Internet. Each local shop has a name, contact details (e.g., phone number and email), and a unique location. The database application also needs to store the URL(u
produc
quantit
details such as their name, address and email. Ever
assigned a unique ID. A customer may place an orde
least one product and each order is from either a shop o
Customers have three payment options (i.e., cas
card) but for each order only one payment option can be chosen. A delivery may be requested for each order. After full-payment is received, a delivery would be sent out subject to products' availability. Every delivery has a unique tracking number.

(Exercise 2) A retailer company wants to build a database application for managing information about its sale process. The company sells **products** in both local **shops** and **webstores** on the Internet. Each local **shop** has a name, contact details (e.g., phone number and email), and a unique location. The database application also needs to store the URL(u ery **produ** quantit **customers**' details such as their name, address a **customer** is assigned a unique ID. A **cust** r that consists of at least one **product** and each or a **webstore**. **Customers** have three payme and credit card) but for each **order** only one payment option can be chosen. A **delivery** may be requested for each **order**. After full-payment is received, a **delivery** would be sent out subject to **products**' availability. Every **delivery** has a unique tracking number.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)

Constructing an ER or EER Model

- Identify the entities (including weak entity types)

sh

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co
  - The company sells products in both local on the Internet.
  - Each **order** is associated with either a

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co
  - The company sells products in both local on the Internet.
  - Each **order** is associated with either a
  - subclass **shop**, **webstore**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co

  - The company sells products in both local on the Internet.
  - Each **order** is associated with either a
  - subclass **shop**, **webstore**
  - superclass **store**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co

  - The company sells products in both local on the Internet.
  - Each **order** is associated with either a

    - subclass **shop**, **webstore**
    - superclass **store**
    - disjoint and complete

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  sh
- Id
  co
  - The company sells products in both local on the Internet.
  - Each **order** is associated with either a
    - subclass **shop**, **webstore**
    - superclass **store**
    - disjoint and complete
- Identify the relationships

(Exercise 2) A retailer company wants to build a database application for managing information about its sale process. The company sells **products** in both local **shops** and **webstores** on the Internet. Each local **shop** has a name, contact details (e.g., phone number and email), and a unique location. The database application also needs to store the URL(u ery **produ** quantit **customers**' details such as their name, address a **customer** is assigned a unique ID. A **cust** r that consists of at least one **product** and each or a **webstore**. **Customers** have three payme and credit card) but for each **order** only one payment option can be chosen. A **delivery** may be requested for each **order**. After full-payment is received, a **delivery** would be sent out subject to **products**' availability. Every **delivery** has a unique tracking number.

(Exercise 2) A retailer company wants to build a database application for managing information about its sale process. The company sells **products** in both local **shops** and **webstores** on the Internet. Each local **shop** has a name, contact details (e.g., phone number and email), and a unique location. The database application also needs to store the URL(u ery **produ** quantit **customers**' details such as their name, address and email. Every **customer** is assigned a unique ID. A **cust er** that **consists of** at least one **product** an her a **shop** or a **webstore**. **Customers** have thr paypal, and credit card) but for each **order** only one payment option can be chosen. A **delivery** may be requested **for** each **order**. After full-payment is received, a **delivery** would be sent out subject to **products**' availability. Every **delivery** has a unique tracking number.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  - superclass **store**
- Identify the relationships
  - **customer** **place** **order**
  - **order** **consists of** **product**
  - each order **is from** **store**(superclass) (either subclass **shop** or subclass **webstore**)
  - **delivery** **is for** **order**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  - superclass **store**
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  - superclass **store**
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - Every **product** has a unique productID, a description, an item price, and a quantity in stock.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  - superclass **store**
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - Every **product** has a unique productID, a description, an item price, and a quantity in stock.
  - Attributes for **product**: **productID, description, item price, quantity**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  - superclass **store**
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - Every **product** has a unique productID, a description, an item price, and a quantity in stock.
  - Attributes for **product**: **productID, description, item price, quantity**
  - Primary key for **product**: **productID**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  -
  -
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  -
  -

- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - The database application should also record **customers**' details such as their name, address and email. Every **customer** is assigned a unique ID.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  -
  -
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - The database application should also record **customers**' details such as their name, address and email. Every **customer** is assigned a unique ID.
  - Attributes for **customer**: **name, address, email, CustomerID**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Id
  co
  -
  -
- Identify the relationships
- Identify the attributes of entities and relationships for each entity type
  - The database application should also record **customers**' details such as their name, address and email. Every **customer** is assigned a unique ID.
  - Attributes for **customer**: **name, address, email, CustomerID**
  - Primary key for **customer**: **CustomerID**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Identify subclass/superclass and the corresponding disjointness and co
  -
- Id
- Identify the attributes of entities and relationships for each entity type

Constructing an ER or EER Model

- Identify the entities (including weak entity types) - **shop**, **webstore**, **product**, **customer**, **order**, **delivery**
- Identify subclass/superclass and the corresponding disjointness and co
  -

- Id
- Identify the attributes of entities and relationships for each entity type
  - Each local **shop** has a name, contact de email), and a unique location. The databas store the URL(unique), name and last updated date of each **webstore**.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Identify subclass/superclass and the corresponding disjointness and co
  -
  -

- Id
- Identify the attributes of entities and relationships for each entity type
  - Each local **shop** has a name, contact de email), and a unique location. The databas store the URL(unique), name and last updated date of each **webstore**.
  - Attributes for superclass **store**: **name**, **location/URL**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Identify subclass/superclass and the corresponding disjointness and co
  -
  -

- Id
- Identify the attributes of entities and relationships for each entity type
  - Each local **shop** has a name, contact de email), and a unique location. The databas store the URL(unique), name and last updated date of each **webstore**.
    - Attributes for superclass **store**: **name**, **location/URL**
    - Primary key for superclass **store**: **location/URL**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
  shop, **webstore**, **product**, **customer**, **order**, **delivery**
- Identify subclass/superclass and the corresponding disjointness and co
  -
  -

- Id
- Identify the attributes of entities and relationships for each entity type
  - Each local **shop** has a name, contact de email), and a unique location. The databas store the URL(unique), name and last updated date of each **webstore**.
    - Attributes for superclass **store**: **name**, **location/URL**
    - Primary key for superclass **store**: **location/URL**
    - Attributes for subclass **shop**: **phone number, email**
    - Attributes for subclass **webstore**: **last updated date**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - A **customer** may **place** an **order**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - A **customer** may **place** an **order**
  - Cardinality ratios: A customer may r

    is placed by **one** customer.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - A **customer** may **place** an **order**

  - Cardinality ratios: A customer may r **is placed by** **one** customer.
  - Participation constraints: A **customer** **may or may not** **place** any orders (**Partial**). An **order** **must** **be placed by** one customer (**Total**).

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - A **delivery** may be requested **for**

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

  - A **delivery** may be requested **for**
  - Cardinality ratios: A delivery **is for** at most **one** order and an order **has** at most **one** delivery.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id

- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

  - A **delivery** may be requested **for**
  - Cardinality ratios: A delivery **is for** at most **one** order and an order **has** at most **one** delivery.
  - Participation constraints: A **delivery must be for** an order (**Total**). An **order may or may not have** a delivery (**Partial**).

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - Each order **consists of** at least one

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - Each order **consists of** at least one
  - Cardinality ratios: An order may **co**
    product may **be contained** in **many** orders.

Constructing an ER or EER Model

- Identify the entities (including weak entity types)
- Identify subclass/superclass
- Id
  -
  -
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const
  - Each order **consists of** at least one
  - Cardinality ratios: An order may **co**

    product may **be contained** in **many** orders.
  - Participation constraints: A **order** **must** **contain** some product (**Total**). A **product** **may or may not** **be contained** in an order (**Partial**).

Constructing an ER or EER Model

Constructing an ER or EER Model

- Id
- Id
- Id
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

Constructing an ER or EER Model

Constructing an ER or EER Model

- Id
- Id
- Id
- Identify the attributes of entities and relationships
- Identify cardinality ratios and participation const

- **Not all the constraints can be expressed in the ER model**

(Exercise 2) A retailer company wants to build a database application for managing information about its sale process. The company sells products in both local shops and webstores on the Internet. Each local shop has a name, contact details (e.g., phone number and email), and a unique location. The database application also needs to store the URL(u
produc
quantit
details such as their name, address and email. Ever
assigned a unique ID. A customer may place an orde
least one product and each order is from either a shop o
**Customers have three payment options (i.e., c
credit card) but for each order only one payment option can be chosen.** A delivery may be requested for each order. **After full-payment is received, a delivery would be sent out subject to products' availability**. Every delivery has a tracking number.

Phase 3: Logical Design

- **Logical design** is the process of constructing a logical data model (e.g. relational or object-oriented).
- A c
  be f
  ex
  - **From:** An ER model
  - **To:** Relations with their primary and foreig
    to deal with retrieving, updating and deletion.

Note: The logical design is based on the **relational data model** in this course.

ER-to-Relations Algorithm

- 7-step algorithm to convert the basic ER model into relations, and more steps for the EER model.

- Foreign key approach
- Merged relation appro
- Cross-reference appr

Step 4: Mapping of Binary 1:N Relationshi

Step 5: Mapping of Binary M:N Relationship Types

Step 6: Mapping of Multi-valued Attributes

Step 7: Mapping of N-ary Relationship Types

Step 8: Mapping of Superclass/Subclass

Step 1: Regular Entity types

- For each regular entity type E, **create a relation schema** with the attributes of E (ignore multi-valued attributes until Step 6), where
  - **PK:** the key attributes of E

Step 1: Regular Entity types

- For each regular entity type E, create a relation schema with the attributes of E (ignore multi-valued attributes until Step 6), where
  - **PK:** the key attributes of E

- COURSE(course_num, course_name, description, num_sem_hours, level) with PK: {course_num}
- **Note:** This is not necessarily the final relation schema of COURSE.

Step 2: Weak Entity Types

- For each weak entity type E_w, **create a relation schema** with the attributes of E_w plus the PK of its identifying entity type, where
  - **PK:** the partial key attributes of E_w plus the PK of its identifying entity

Step 2: Weak Entity Types

- For each weak entity type E_w, **create a relation schema** with the attributes of E_w plus the PK of its identifying entity type, where
  - **PK:** the partial key attributes of E_w plus the PK of its identifying entity
  -

- SECTION(section_num, instructor, semester, year, course_num)
  with PK: {section_num, course_number}
  with FK: [course_num]⊆COURSE[course_num]

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where
  -

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where
  -
  -

- DEPARTMENT(Name, Address, Mgr_SSN, _)

  PK: {Name}

  FK: [Mgr_SSN]⊆EMPLOYEE[SSN].

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where
  -
  -

- DEPARTMENT(Name, Address, Mgr_SSN,)

  PK: {Name}

  FK: [Mgr_SSN]⊆EMPLOYEE[SSN].
- How can we model the total participation?

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where
  -

- DEPARTMENT(Name, Address, Mgr_SSN, _ _

  PK: {Name}

  FK: [Mgr_SSN]⊆EMPLOYEE[SSN].
- How can we model the total participation?

  Add `NOT NULL` constraint to Mgr_SSN for total participation.

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where
  -

- DEPARTMENT(Name, Address, Mgr_SSN, _ _

  PK: {Name}

  FK: [Mgr_SSN]⊆EMPLOYEE[SSN].
- Why don't we extend the relation schema of the partial-side entity type?

Step 3: Binary 1:1 Relationship Types - (Foreign key)

- For a 1:1 relationship type R with one total participation, **extend the relation schema of the total-side entity type** by the attributes of R and the PK of the partial-side entity type, where

- DEPARTMENT(Name, Address, Mgr_SSN, ...)

  PK: {Name}

  FK: [Mgr_SSN]⊆EMPLOYEE[SSN].
- Why don't we extend the relation schema of the partial-side entity type?

  This may cause many `NULL` values.

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If participation on both sides is total, we may **merge the relation schemas of both entity types and the attributes of the relat single relation**.

- EMPLOYEE-DEP(SSN, Name, Salary, Start_

  PK: {SSN} or {Dname}

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If participation on both sides is total, we may **merge the relation schemas of both entity types and the attributes of the relat single relation**.

- EMPLOYEE-DEP(SSN, Name, Salary, Start_

  PK: {SSN} or {Dname}
- How can we model the total participations?

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If participation on both sides is total, we may **merge the relation schemas of both entity types and the attributes of the relat single relation**.

- EMPLOYEE-DEP(SSN, Name, Salary, Start_

  PK: {SSN} or {Dname}
- How can we model the total participations?

  Add `NOT NULL` constraint to both SSN and Dname for total participations.

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If participation on both sides is total, we may **merge the relation schemas of both entity types and the attributes of the relat single relation**.

- EMPLOYEE-DEP(SSN, Name, Salary, Start_

  PK: {SSN} or {Dname}
- How can we model the total participations?

  Add `NOT NULL` constraint to both SSN and Dname for total participations.
- **Is merging them always a good solution?**

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If p **hemas of both entity types and the attributes of the relat single relation.**
- However, **merging them is not always a good s**

Step 3: Binary 1:1 Relationship Types - (Merged relation)

- How can we translate the following kind of 1:1 relationship type?

- If p **hemas of both entity types and the attributes of the relat single relation**.

- However, **merging them is not always a good s**
  (1) The two entity types represent different entitie
  (2) The two entity types participate in different relationship types.
  (3) Having separate relation schemas for two entity types often leads to more efficient updates than a single relation schema.
  (4) ...

Step 3: Binary 1:1 Relationship Types - (Cross-reference)

- How can we translate the following kind of 1:1 relationship type?

Step 3: Binary 1:1 Relationship Types - (Cross-reference)

- How can we translate the following kind of 1:1 relationship type?

- If both sides are partial, we may **create a (n** ch cross-references the PKs of the relation schema

Step 3: Binary 1:1 Relationship Types - (Cross-reference)

- How can we translate the following kind of 1:1 relationship type?

- If both sides are partial, we may **create a (n** ch cross-references the PKs of the relation schema

- MANAGES(SSN, Dname, Start_date) with
  PK: {SSN} or {Dname}
  FKs: [SSN]⊆EMPLOYEE[SSN] and [Dname]⊆DEPARTMENT[Name]

Step 3: Binary 1:1 Relationship Types - (Cross-reference)

- How can we translate the following kind of 1:1 relationship type?

- If both sides are partial, we may **create a (n** ch cross-references the PKs of the relation schema

- MANAGES(SSN, Dname, Start_date) with
  PK: {SSN} or {Dname}
  FKs: [SSN]⊆EMPLOYEE[SSN] and [Dname]⊆DEPARTMENT[Name]
- Can we still merge them into a single relation using previous approaches?

Step 3: Binary 1:1 Relationship Types - (Cross-reference)

- How can we translate the following kind of 1:1 relationship type?

- If both sides are partial, we may **create a (n** ch cross-references the PKs of the relation schema
- MANAGES(SSN, Dname, Start_date) with
  PK: {SSN} or {Dname}
  FKs: [SSN]⊆EMPLOYEE[SSN] and [Dname]⊆DEPARTMENT[Name]
- Can we still merge them into a single relation using previous approaches?
  We cannot; otherwise what would be the primary key for the merged relation schema?

Step 4: Binary 1:N Relationship Types

- For each 1:N relationship type R, **extend the relation schema of the N-side entity type** by the attributes of R and the PK of the 1-side entity type, where
  -

Step 4: Binary 1:N Relationship Types

- For each 1:N relationship type R, extend the relation schema of the N-side entity type by the attributes of R and the PK of the 1-side entity type, where
  -

- STUDENT(SSN, Name, Number, DoB, address, phone, major_dept, major_name) with

  PK: {SSN}

  FK: [major_dept]$\subseteq$DEPARTMENT[dept_code]

Step 5: Binary M:N (N:N) Relationship Types

- For each M:N (N:N) relationship type R, **create a relation schema** with the attributes of R plus the PKs of the participating entity types, where
  - **PK:** the combination of the PKs of the participating entity types
  -

- GRADE_RECORD(ssn, section_num, course_num, letter_grade, final_grade)

  PK: {ssn, section_num, course_num}

  FK: [ssn] $\subseteq$ STUDENT[ssn]

  FK: [section_num, course_num] $\subseteq$ SECTION[section_num, course_num].

Step 6: Multi-valued Attributes

- For each multi-valued attribute *A*, **create a relation schema** with an attribute corresponding to *A* plus the PK of the entity/relationship type that has *A* as an attribute, where
  -
  - 4

Step 6: Multi-valued Attributes

- For each multi-valued attribute A, create a relation schema with an attribute corresponding to A plus the PK of the entity/relationship type that has A as an attribute, where
  - https://eduassistpro.github.io/
  - Add WeChat edu_assist_pro

- EMPLOYEE_ADDRESS(SSN, Address) with

  PK: {SSN, Address}

  FK: [SSN]$\subseteq$EMPLOYEE[SSN]

ER-to-Relations Algorithm (Recall)

- The algorithm to first convert the basic ER model into relations, and then convert superclass/subclass from the EER model into relations.

- Foreign key approach
- Merged relation appro
- Cross-reference appr

Step 4: Mapping of Binary 1:N Relationshi

Step 5: Mapping of Binary M:N Relationship Types

Step 6: Mapping of Multi-valued Attributes

Step 7: Mapping of N-ary Relationship Types

Step 8: Mapping of Superclass/Subclass

(Credit Cookie) Graph Model and ER Diagram

(Credit Cookie) Graph Model and ER Diagram

- 1st paper in ACM Transactions on Database Syst
- 1st international conference on very large data ba