Lab01 week2 ER diagram

1. University

   entities

   Students :

   - Name

   - Student ID **(unique)**

   - SSN **(unique)**

   - current address + phone number

   - permanent address + phone number + (more)

   - date of birth

   - sex

   -class

   - major department

   -minor department (nullable)

   - degree progarm

   Departments:

   -department name **(unique)**

   -department code **(unique)**

   -office number

   -office phone number

   -college

   Courses:

   -course name

   -description

   -course number **(unique)**

-number of semester hours

-level

-offering department

Sections:

-intruction

-semester

-year

-course

-section number **(unique)**

Reports:

- student

-sections

-letter grade

-numeric grade

2. Relationship

Students → departments

Students → courses

Courses → Sections

Students → reports

Reports -> sections

3. Contraint

Students to departments : Many to One

one student can have only one department but one department can have a lot of students

Students to courses : Many to Many

one student can have a lot of courses and one course can have a lot of students

Students to Reports : one to one

one student per report

Reports to sections : one to many

on one reports can have a lot of sections

Course to Section : One to Many

one course can have a lot of sections but one sections is one course only

4. Weak entity type

   Section : Section cant exist without course || section || <=> courses

   Report : Report cant create without student || Report || <=> students

Mail ordering

Entity

- Employees
  - employee ID (unique)
  - name → first name and last name
  - Zip code
- Customers
  - customer ID (unique)
  - name → first name → last name
  - Zip code
- Parts
  - part ID (unique)
  - part name
  - price
  - quantity in stock
- Orders
  - employee ID
  - Customer ID

- order ID (unique)
- quantity
- date of receipt
- expected ship date
- shiped date

Relationship

Employees → orders

orders → parts

Customers → orders

Constraint

Employees → orders : one to many (as one employees can accept a lot of orders)

Orders → parts : many to many (one order can have a lot of parts and one parts can be in a lot of orders)

Customers → orders : one to many (one customer can place a lot of order)