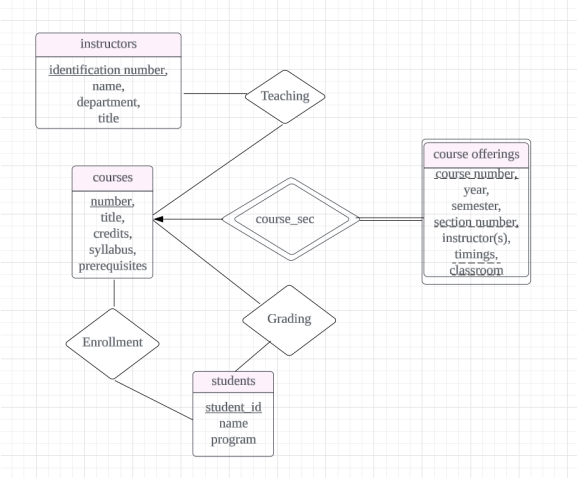


assumptions:

- “student_id” uniquely determines each student in “students”.
- “identification number” uniquely determines each instructor in “instructors”.
- “number” uniquely determines each course in “courses”.
- {“course number”, “section number”, “timings”, “classroom”} uniquely determines each course section in “course offering.
- every student might have multiple instructors, and every instructor might have multiple students.
- each course is taught by one single instructor, with probably multiple sections.
- one instructor can teach many courses
- every student can take multiple courses, but can only enroll one fixed section of each course, so *a student in one specific course can only have one grade, and we could know its enrollment as long as we know the course and the student*
- “course offering” is a weak entity associated with “courses”

(b)

relational schemas:

relationship sets:

- Enrollment(Student_id, number)
- Grading(student_id, number)
- course_sec(course number, section number, timings, classroom, year, semester, instructor(s))
- Teaching (identification_number, number, title, credits, syllabus, prerequisites)

entity sets:

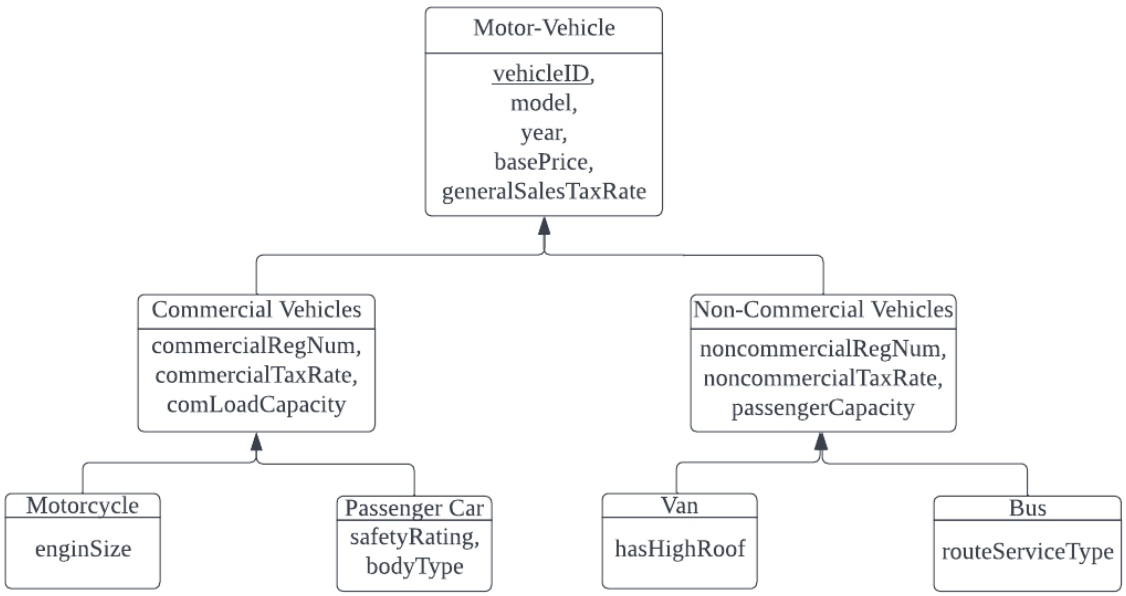
- course offerings(course number, year, semester, section number, instructor(s), timings, classroom)
- courses(number, title, credits, syllabus, prerequisites)
- instructors(identification number, name, department, title)

- students(student_id, name, program)

Question 3

- Assumptions:
  - “Commercial Vehicles” are used for business purposes and may include “van” and “bus”
  - “Non-Commercial Vehicles” are used for personal or private, and may include “motorcycles” and “passenger cars”
  - “General Sales tax” is a fixed percentage for all vehicles
  - “Additional tax” is a fixed percentage for all vehicles in a certain category, and vehicles in different category have different percentage.
  - we separate “attributes” in “entity” as “inherited attributes” and “additional attributes” for clarification, while “attributes” of “entity” is the summation of these two.
  - “Motor-Vehicle” has attributes: vehicleID, model, year, basePrice, generalSalesTaxRate, where vehicleID is a primary key
  - “Commercial Vehicles” and “Non-Commercial Vehicles” are both lower-level entities of “Motor-Vehicle”, and they inherit all attributes in “Motor-Vehicle”.
  - “Van” and “Bus” are both lower-level entities of “Commercial Vehicles”, and “Motorcycles” and “Passenger Cars” are both lower-level entities of “Non-Commercial Vehicles”.
  - “Commercial Vehicles” entity has some additional attributes besides the ones inherited from “Motor-Vehicles”: commercialRegNum, commercialTaxRate, comLoadCapacity
  - “Non-Commercial Vehicles” entity has some additional attributes besides the ones inherited from “Motor-Vehicles”: noncommercialRegNum, noncommercialTaxRate, passengerCapacity
  - “Motorcycle” entity has an additional attribute besides the ones inherited from “Non-Commercial Vehicles”: engineSize
  - “Passenger Car” has additional attributes besides the ones inherited from “Non-Commercial Vehicles”: bodyType, safetyRating
  - “Van” has an additional attribute besides the ones inherited from “Commercial Vehicles”: hasHighRoof
  - “Van” has an additional attribute besides the ones inherited from “Commercial Vehicles”: routeServiceType
- Entity:
  - “Motor-Vehicle” entity:
    - attributes: vehicleID, model, year, basePrice, generalSalesTaxRate
  - “Commercial Vehicles” entity:
    - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate
    - additional attributes: commercialRegNum, commercialTaxRate, comLoadCapacity
  - “Non-Commercial Vehicles” entity:
    - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate

        - additional attributes: noncommercialRegNum, noncommercialTaxRate, passengerCapacity
    - “Motorcycle” entity:
        - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate, noncommercialRegNum, noncommercialTaxRate, passengerCapacity
        - additional attributes: engineSize
    - “Passenger Car” entity:
        - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate, noncommercialRegNum, noncommercialTaxRate, passengerCapacity
        - additional attributes: safetyRating, bodyType
    - “Van” entity:
        - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate, commercialRegNum, commercialTaxRate, comLoadCapacity
        - additional attributes: hasHighRoof
    - “Bus” entity:
        - inherited attributes: vehicleID, model, year, basePrice, generalSalesTaxRate, commercialRegNum, commercialTaxRate, comLoadCapacity
        - additional attributes: routeServiceType
- related ER Diagram:



Question 4

$\prod_{Fname,Minit,Lname,Address}(EMPLOYEE \bowtie_{EMPLOYEE.Dno=DEPARTMENT.Dnumber} (\sigma_{Dname='Research'}(DEPARTMENT))$

Question 5

$\prod_{Pnumber,Lname,Address,Bdate}((\sigma_{Plocation='Stafford'}(PROJECT)) \bowtie_{PROJECT.Dnum=DEPARTMENT.Dnumber} (DEPARTMENT \bowtie_{DEPARTMENT.mgr\_ssn=EMPLOYEE.Ssn} EMPLOYEE))$

Question 6

$\prod_{Pnumber}((\sigma_{Lname='Smith'}(EMPLOYEE) \bowtie WORKS\_ON) \bowtie PROJECT) \cup \prod_{Pnumber}((\sigma_{Lname='Smith'}(EMPLOYEE) \bowtie_{EMPLOYEE.Ssn=DEPARTMENT.Mgr\_ssn} DEPARTMENT) \bowtie_{DEPARTMENT.Dnumber=PROJECT.num} PROJECT)$

Question 7

$\prod_{Fname,Minit,Lname}(EMPLOYEE) - \prod_{Fname,Minit,Lname}(DEPENDENT \bowtie EMPLOYEE)$

Question 8

$\prod_{Fname,Minit,Lname}(EMPLOYEE \bowtie (DEPARTMENT \bowtie_{DEPARTMENT.Mgr\_ssn=DEPENDENT.Essn} DEPENDENT))$

Question 9

$(\rho_{Supervisor}(\sigma_{Fname='James' \wedge Lname='Borg'}(EMPLOYEE))) \bowtie_{Supervisor.Ssn=EMPLOYEE.Super\_ssn} EMPLOYEE$

Question 10

James_Borg ← $\sigma_{Fname='James' \wedge Lname='Borg'}(EMPLOYEE)$

Directed_Subordinates ← $\sigma_{James\_Borg.Ssn=EMPLOYEE.Super\_ssn}(EMPLOYEE)$

Indirected_Subordinates ← $\sigma_{Directed\_Subordinates.Ssn=EMPLOYEE.Super\_ssn}(EMPLOYEE)$

⇒ All Employees directly supervised by those directly supervised by “James Borg” are listed in “Indirected_Subordinates”

Would it be possible to find all employees supervised by a given employee at all levels?

It's possible to find all employees using Recursion.

Given any supervisors, we can find their directed_subordinates, and find their indirected_subordinates, and find their next_level_indirected_subordinates and next level and next level .etc with same method as above.