Q_1 (a) query 1 and 2 are equivalent.

proof.

since $\alpha \notin S$ and $\beta \in S$, $\{\alpha, \beta\} \subseteq R$, $\sigma(r \times s) = \sigma_{p(\alpha)}(r) \times \sigma_{p(\alpha)}(S)$

since $\alpha \notin S$, then $\sigma_{p(\alpha)}(S) = S$ So $\sigma_{p(\alpha)}(r \times S) = \sigma_{p(\alpha)}(r) \times S$

Query 1 : $\sigma_{p(\alpha)}(r \bowtie_{r.\beta = S.\beta} S) = \sigma_{p(\alpha)}(\sigma_{r.\beta = S.\beta}(r \times S)) = \sigma_{r.\beta = S.\beta}(\sigma_{p(\alpha)}(r \times S))$

Query 2 : $\sigma_{p(\alpha)}(r) \bowtie_{r.\beta = S.\beta} S = \sigma_{r.\beta = S.\beta}(\sigma_{p(\alpha)}(r) \times S)$

$= \sigma_{r.\beta = S.\beta}(\sigma_{p(\alpha)}(r \times S)) = \sigma_{p(\alpha)}(r \bowtie_{r.\beta = S.\beta} S)$

So. query 1 and query 2 are equivalent.

(b) Query 1 : $\sigma_{r.\beta}(r \times s)$ costs mn operations, and return $|\sigma_{r.\beta = s.\beta}(r \times s)|$ tuples.

$\sigma_{p(\alpha)}$(result of $\sigma_{r.\beta = s.\beta}(r \times s)$) costs $|\sigma_{r.\beta = s.\beta}(r \times s)| = f_1 \cdot mn$ operations

where $f_1 := f_{\sigma_{r.\beta = s.\beta}(r \times s)}$

Total operations of $Q_1 = mn + f_1 \cdot mn$

Query 2 : $\sigma_{p(\alpha)}(r)$ cost m operations, return $|\sigma_{p(\alpha)}(r)|$ tuples

$\sigma_{r.\beta = s.\beta}$ (result of $\sigma_{p(\alpha)}(r) \times S$) costs $|\sigma_{p(\alpha)}(r)| \cdot n = f_2 \cdot mn$ operations.

where $f_2 = f_{\sigma_{p(\alpha)}(r)}$

total operations of $Q_2 = m + f_2 mn$

The operation difference of Q_1 and Q_2 = Total operations of Q_1 - Total operations of Q_2

$= m[n(1 + f_1 - f_2) - 1]$

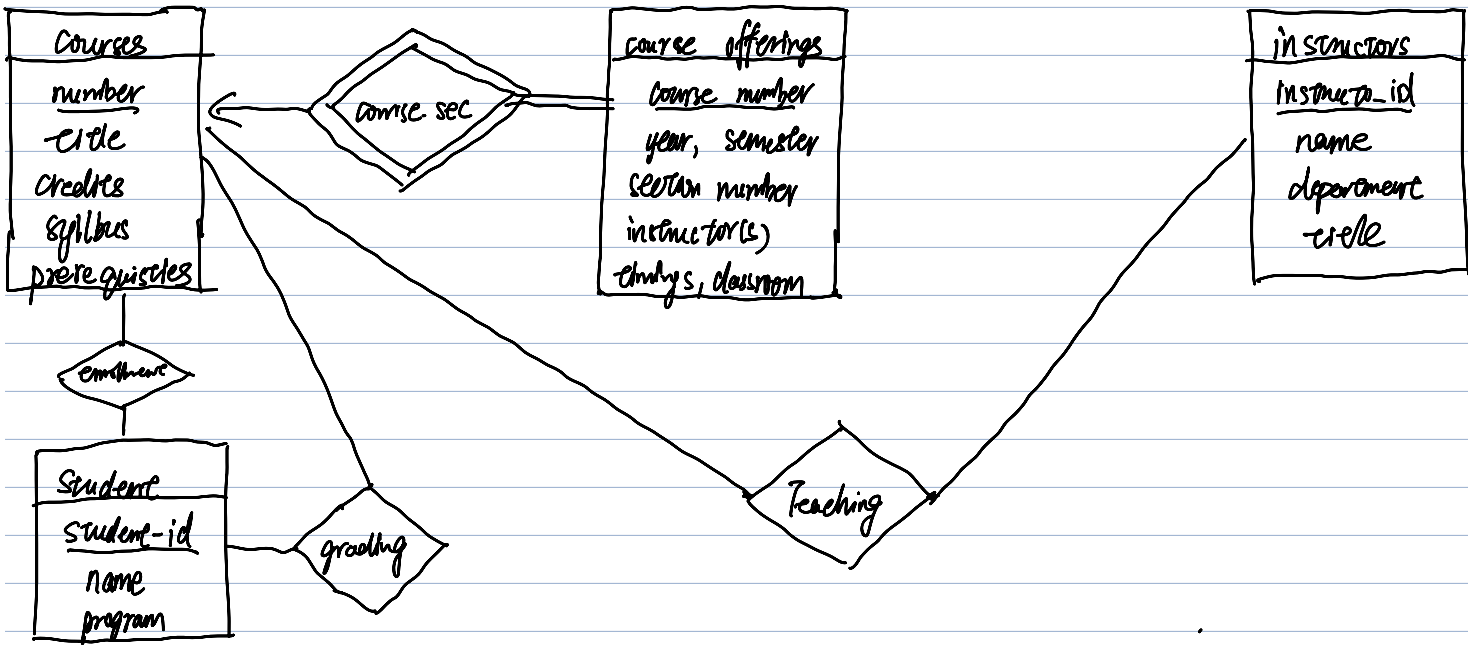
$\because m$ should be greater than 0.

$\therefore$ when $n(1 + f_1 - f_2) - 1 = 0$, Query 1 and Query 2 have the same efficiency.

when $n(1 + f_1 - f_2) - 1 > 0$, Query 2 is of higer efficiency.

when $n(1 + f_1 - f_2) - 1 < 0$, Query 1 is of higer efficiency.

Q2 (a)



Assumption:

"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) Student: Student id (PK), Name, Program
Instructor: Instructor id (PK), Name, Department, Title
Course: Course Number (PK), Title, Credits, Syllabus, Prerequisites
CourseOffering: Course Number (FK), Year, semester, Section Number, Time, Classroom
Enrollment: StudentID (FK), Course Number, Year, Semester, Section Number, Grade.
Number (FK)

Grading: student-id (FK), number (FK)

Course_sec: course_number (FK), section-number (FK), timings (FK), classroom (FK), year, semester, Instructors

Teaching: identification_number (FK), number (FK), title, credits, syllbus, prerequisites.

Q3：

Assumptions:

（Each vehicle type has a unique set of attributes in addition to those inherited from the Vehicle entity.

The sales tax is divided into a general sales tax applicable to all vehicles and an additional tax specific to the commercial or non-commercial category.

The distinction between commercial and non-commercial for vans is determined by their intended use, which is reflected in a boolean attribute and affects tax calculation.

The attributes related to tax are calculated and not stored directly; they are derived from the base price and tax rates.）

“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, Make, Model, Year, BasePrice, GeneralSalesTax (GeneralSalesTax = BasePrice * GeneralTaxRate)

Commercial Vehicle

Inherits: Vehicle attributes

Additional Attributes: LoadCapacity, CommercialTax (CommercialTax = BasePrice * CommercialTaxRate)

Assumption: Commercial vehicles are designed for goods transportation or commercial use.

Non-Commercial Vehicle

Inherits: Vehicle attributes

Additional Attributes: PassengerCapacity, NonCommercialTax (NonCommercialTax = BasePrice * NonCommercialTaxRate)

Assumption: Non-commercial vehicles are designed for personal or family use.

Motorcycle

Inherits: Non-Commercial Vehicle attributes

Additional Attributes: EngineDisplacement, SidecarFlag (boolean indicating the presence of a sidecar)

Assumption: Motorcycles are considered non-commercial vehicles and are not typically used for commercial purposes.

Passenger Car

Inherits: Non-Commercial Vehicle attributes

Additional Attributes: BodyType (e.g., sedan, hatchback, SUV)

Assumption: Passenger cars vary by body type but share common non-commercial attributes.

Van

Inherits: Vehicle attributes (can be either Commercial or Non-Commercial)

Additional Attributes: SeatingConfiguration, CommercialUseFlag (boolean indicating if the van is used for commercial purposes)

Assumption: Vans can be specialized into commercial or non-commercial depending on their use.

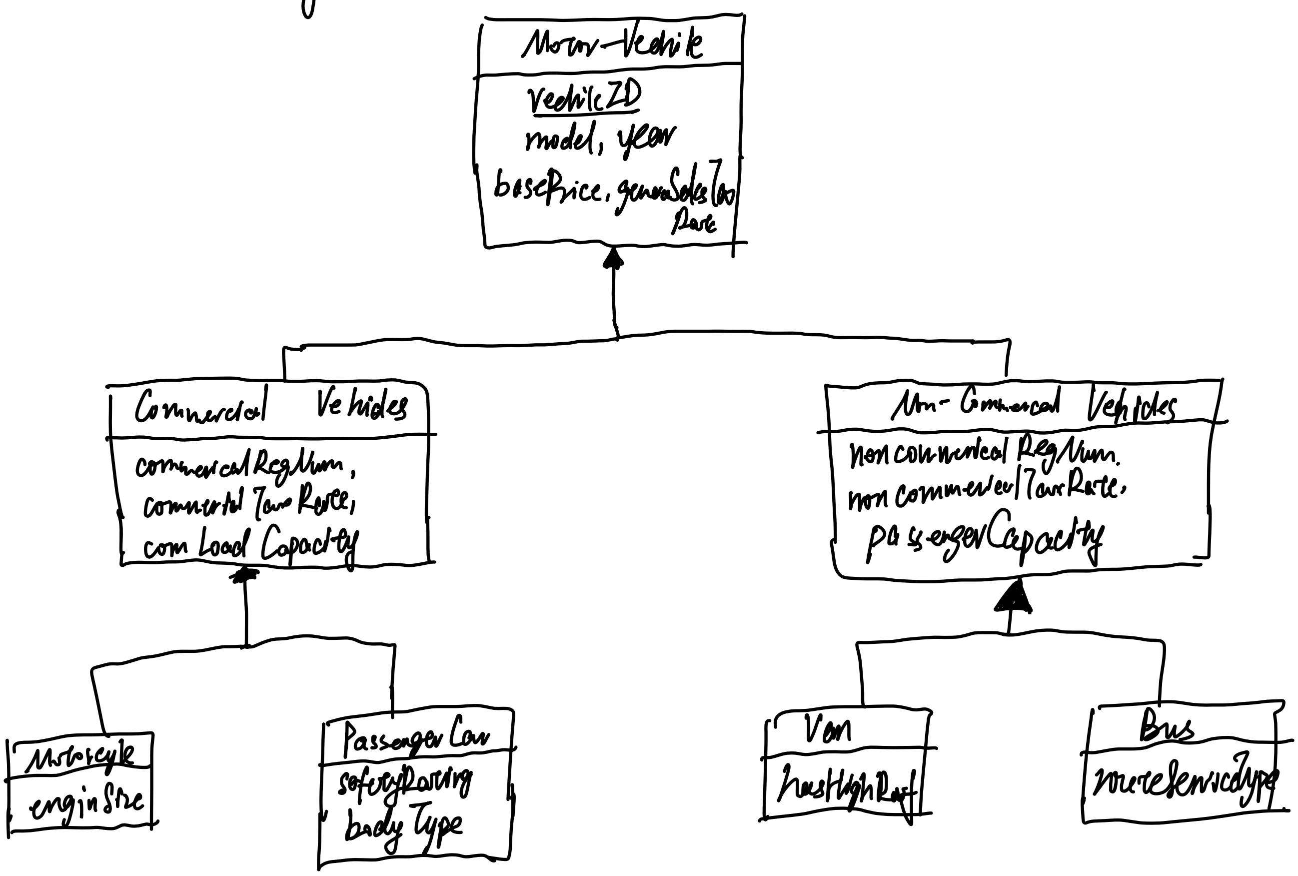
Bus

Inherits: Commercial Vehicle attributes

Additional Attributes: SeatingCapacity, RouteType (e.g., city, intercity)

Assumption: Buses are used for commercial purposes, often for public transportation.

E-R Diagram



Q4

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

Q 5

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

Q6

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

Q7

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

Q8

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

Q9

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

Q10

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

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”

For the second question: It's possible to find all employees using Recursion. First find all employees who are directly supervised by James Borg, then recursively find those employees' subordinates, and so on until there are no more subordinates.