Your Name:

(1) On the university schema, explain in natural language what the following queries do.

- select * from courses where course_id not in (select course_id from prereq)

  courses that do not have any prerequesites

- select i.id, i.name from instructor join teaches on (id) where year = 2024
  - *Also explain what might be an issue with the query and how to fix it*

    instructors who taught in 2024 – repeats instructors – should add a distinct

- select * from section s where building not in
  (select building from department d, course c where d.name = c.name and c.course_id = s.course_id)

  sections that are taught in a different building that the dept that offers the course

- select d.dept_name from department d, course c
  where d.dept_name = c.dept_name
  group by d.dept_name
  having budget < 10000 * sum(credits)

  dept that have less budget than 10000 times the total credit hours that they teach.

- select * from instructor i, advisor a, student s, teaches t1, takes t2
  where i.id = t1.id and i.id = a.i_id and a.s_id = s.id and s.id = t2.id and
  (t1.course_id, t1.sec_id, t1.semester, t1.year) = (t2.course_id, t2.sec_id, t2.semester, t2.year);

  instructor-advisee pairs where the instructor has taught the advisee

- select distinct t1.id, t1.course_id
  from takes t1, takes t2
  where t1.id = t2.id and t1.course_id = t2.course_id and
  (t1.sec_id, t1.semester, t1.year) != (t2.sec_id, t2.semester, t2.year)

  students who have taken the same course multiple times

(2) On University schema, consider the following 3 relations formed by joining existing relations. What should be primary key of these relations? Briefly explain.

Q1(ID, dept name, course id, sec id, semester, year) =
select teaches.ID, department.dept name, course id, sec id, semester, year)
from teaches, instructor, department
where teaches.id = instructor.id and department.dept name = instructor.dept name;

The primary key is the same as that of the teaches table – all this query does is adds the department name to it.

Q2(course id, title, dept name, credits, prereq id) =
select course.course id, title, dept name, credits, prereq id
from course join prereq on (course.course id = prereq.course id);

course_id, prereq_id is the primary key for this table.

Q3(dept name, building, year, num) =
select d.dept name, d.building, t.year, count(*) as num
from takes t, course c, department d
where c.dept name = d.dept name and t.course id = c.course id group by d.dept name, d.building, t.year;

The combination of dept_name and year form the primary key.

(3) Consider the following relational schema that stores information about the current vehicles and license information (i.e., no historical information). Assume each driver can only hold one license. Multiple drivers may be insured for driving a specific vehicle. Further, plate_number is not unique outside a state (i.e., the same plate number may be associated with different vehicles in different states).

- person (person_id, name, address)
- driver_license (person_id, state, license_number, issue_date, expiry_date)
- vehicle (plate_number, state, model, year)
- insured_driver (plate_number, state, person_id)

What should be the primary key for the driver_license table given the information?

What should be the primary key for the insured_driver relation?

Compute the result of the relational algebra expressions on R(A, B, C), and S(C, D).

- $\pi_A(\sigma_{C=3}(R)$

  {alpha, gamma}

- $\pi A(\sigma_{R.C=S.C}(R \times S))$

  {alpha, beta, gamma}

- $\rho_{X(D)}(\pi B(\sigma_{C=2}R)) \cap \pi_D(\sigma_{C=2}(S))$

  {a, b}

- $\pi_C(R) - \pi_{R.C}(\sigma_{R1.C>R.C}(R \times (\rho_{R1}(R))))$

This query finds the maximum C in R ➔ {3}

A	B	C
α	a	1
β	b	2
α	c	1
α	a	3
γ	c	3
γ	a	2
β	b	2
α	c	2
β	c	2

C	D
1	a
2	b
3	c
2	a