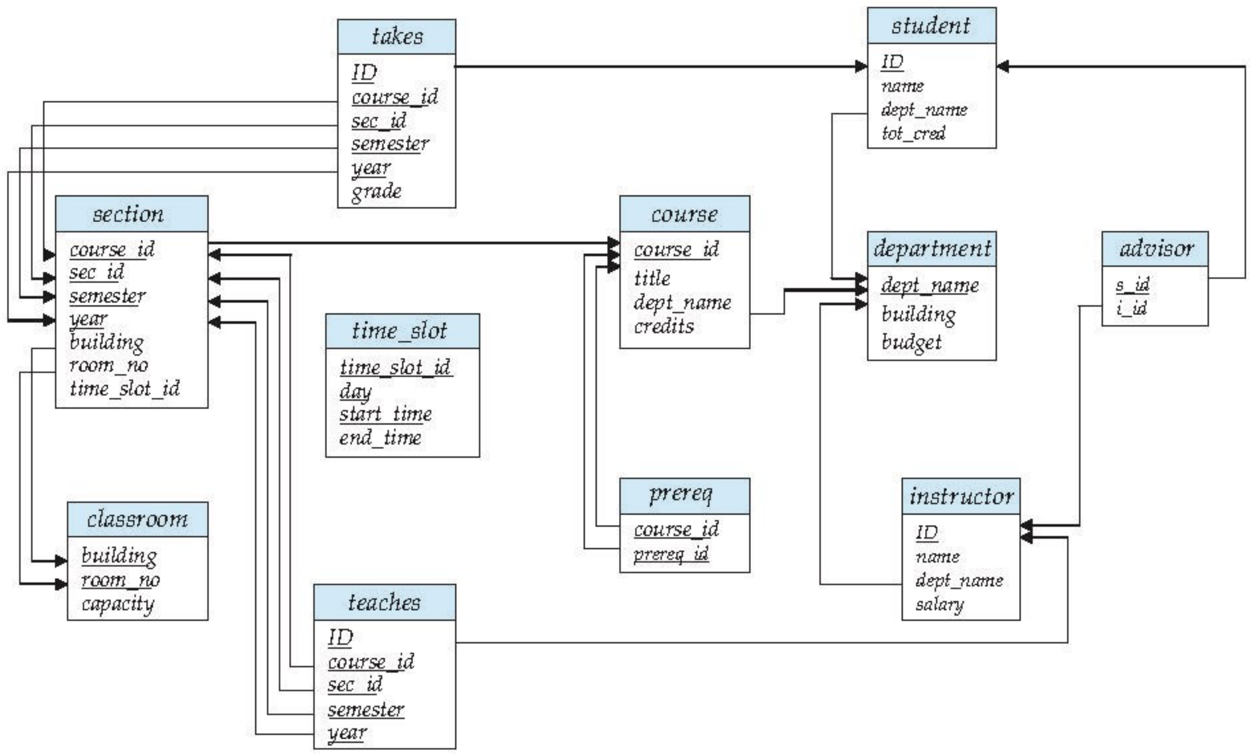


2.2 Consider the foreign key constraint from the dept name attribute of instructor to the departmentrelation. Give examples of inserts and deletes to these relations, which can cause a violation of the foreign key constraint.

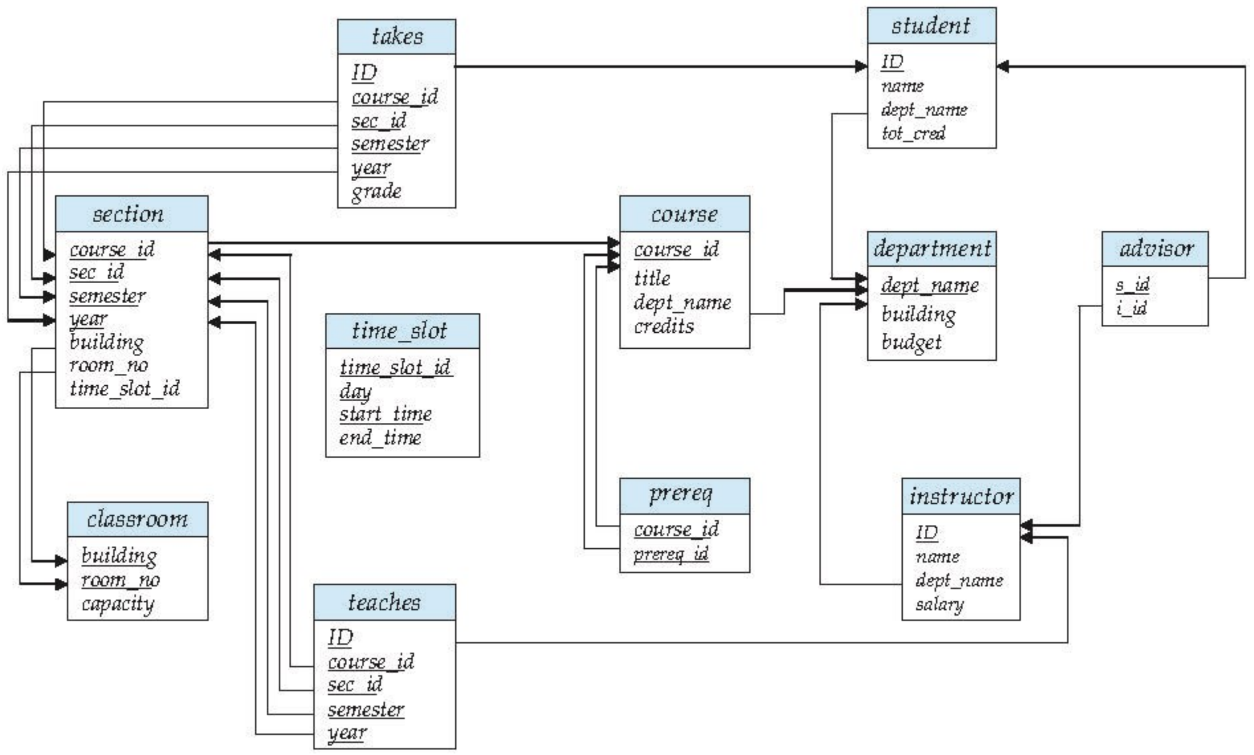
Answer:

• Inserting a tuple: (10111, Ostrom, Economics, 110,000)

into the instructor table, where the department table does not have the department Economics, would violate the foreign key constraint.

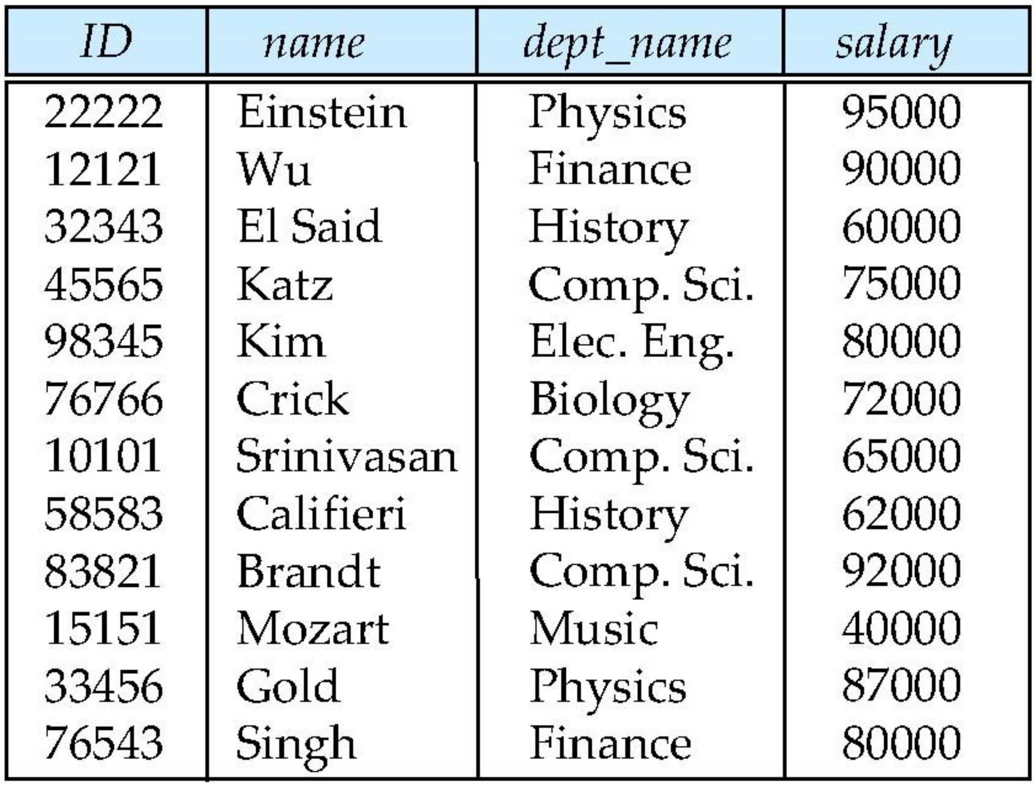
• Deleting the tuple: (Biology, Watson, 90000)

from the department table, where at least one student or instructor tuple has dept name as Biology, would violate the foreign key constraint.



2.3 Consider the time slot relation. Given that a particular time slot can meet more than once in a week, explain why day and start time are part of the primary key of this relation, while end time is not.

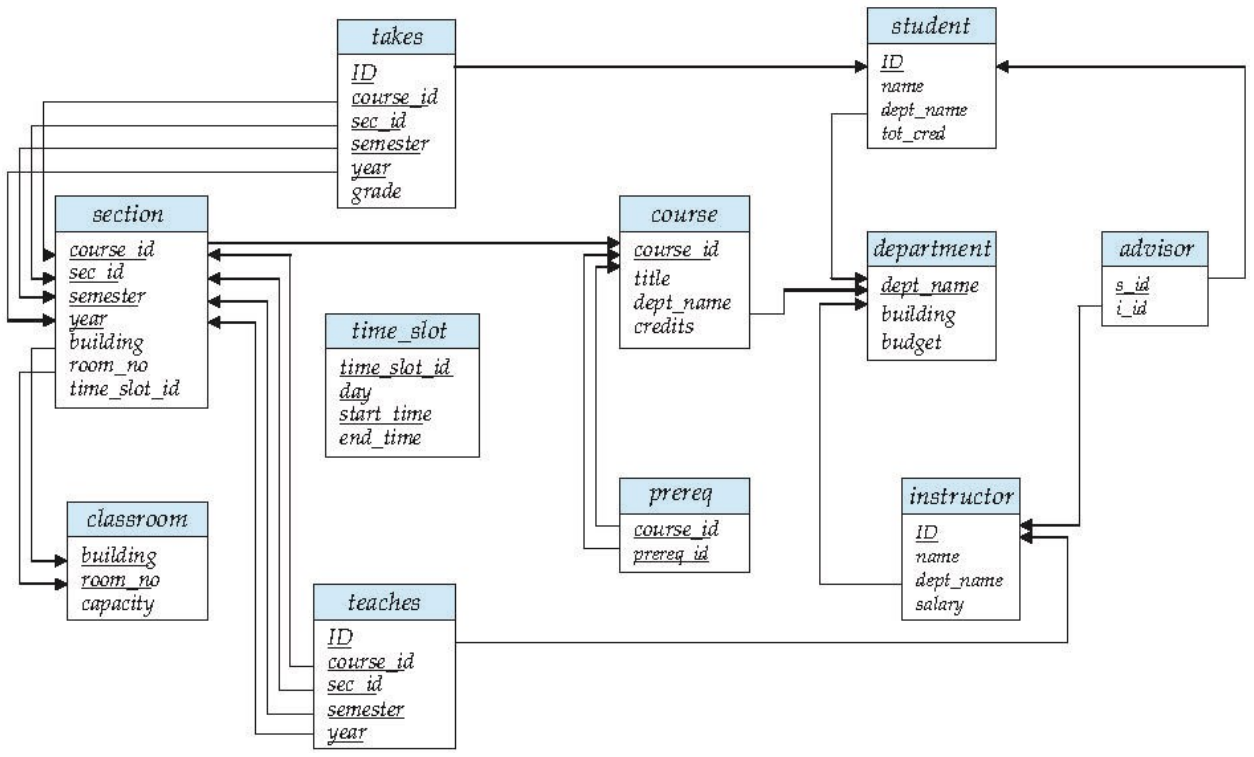
Answer: The attributes day and start time are part of the primary key since a particular class will most likely meet on several different days, and may even meet more than once in a day. However, end time is not part of the primary key since a particular class that starts at a particular time on a particular day cannot end at more than one time.



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

2.4 In the instance of instructor shown in below no two instructors have the same name. From this, can we conclude that name can be used as a superkey (or primary key) of instructor?

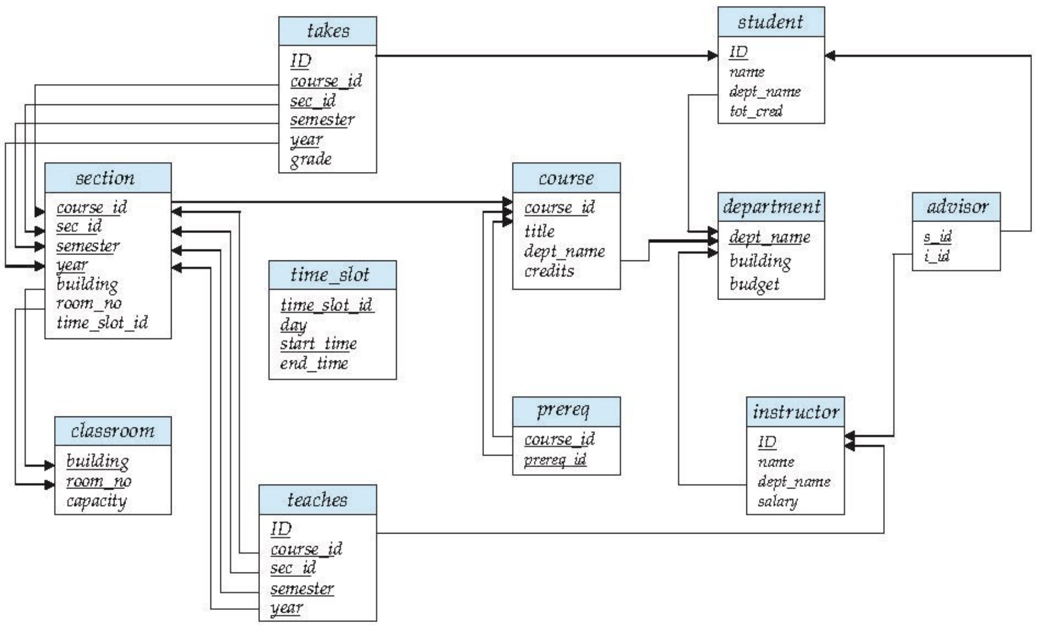
Answer: No. For this possible instance of the instructor table the names are unique, but in general this may not be always the case (unless the university has a rule that two instructors cannot have the same name, which is a rather unlikely scenario).



2.5 What is the result of first performing the cross product of student and advisor, and then performing a selection operation on the result with the predicate s_id = ID? (Using the symbolic notation of relational algebra, this query can be written as $\sigma_{s\_id=student.ID}(student \times advisor)$.)

Answer: The result attributes include all attribute values of student followed by all attributes of advisor. The tuples in the result are as follows. For each student who has an advisor, the result has a row containing that students attributes, followed by an s_id attribute identical to the students ID attribute, followed by the i_id attribute containing the ID of the student's advisor.

Students who do not have an advisor will not appear in the result. A student who has more than one advisor will appear a corresponding number of times in the result.



2.6 Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does.

a. $\sigma_{year \geq 2009}(takes) \bowtie student$

b. $\sigma_{year \geq 2009}(takes \bowtie student)$

c. $\pi_{ID,name,course\ id}(student \bowtie takes)$

Answer:

a. For each student who takes at least one course in 2009, display the students information along with the information about what courses the student took. The attributes in the result are:

ID, name, dept_name, tot_cred, course_id, section_id, semester, year, grade

b. Same as (a); selection can be done before the join operation.

c. Provide a list of consisting of ID, name, course id of all students who took any course in the university.

2.7 Consider the relational database defined below. Give an expression in the relational algebra to express each of the following queries:

employee (person_name, street, city)
works(person_name, company_name, salary)
company (company_name, city)

a. Find the names of all employees who live in city “Miami”.

b. Find the names of all employees whose salary is greater than $100,000.

c. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.

Answer:

a. $\pi_{\text{person\_name}}(\sigma_{\text{city = “Miami”}}(\text{employee}))$

b. $\pi_{\text{person\_name}}(\sigma_{\text{salary > 100000}}(\text{works}))$

c. $\pi_{\text{person\_name}}(\sigma_{\text{city = “Miami”} \wedge \text{salary>100000}}(\text{employee} \bowtie \text{works}))$

2.8 Consider the bank database below. Give an expression in the relational algebra for each of the following queries.

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

a. Find the names of all branches located in “Chicago”.

b. Find the names of all borrowers who have a loan in branch “Downtown”.

Answer:

a. $\pi_{\text{branch\_name}} (\sigma_{\text{branch city = “Chicago”}} (\text{branch}))$

b. $\pi_{\text{customer\_name}} (\pi_{\text{branch name = “Downtown”}} (\text{borrower} \bowtie \text{loan}))$