CS 327E Lecture 9

Shirley Cohen

October 24, 2016

Homework for Today

- Chapters 5 and 10 from the Learning SQL book
- Exercises at the end of assigned chapters

Quiz Question 1

```
mysql> select * from Employee;
+---------+-----------+---------+
| fname   | lname     | dept_id |
+---------+-----------+---------+
| Michael | Smith     |       3 |
| Susan   | Barker    |       3 |
| Robert  | Tyler     |       3 |
| Susan   | Hawthorne |       1 |
| John    | Gooding   |       2 |
+---------+-----------+---------+
```

```
mysql> select * from Department;
+---------+----------------+
| dept_id | name           |
+---------+----------------+
|       1 | Operations     |
|       2 | Loans          |
|       3 | Administration |
+---------+----------------+
```

How many **rows** does the following query return?

```
SELECT e.fname, e.lname, e.dept_id, d.name
FROM Employee e JOIN Department d using (dept_id)
```

A. 0 B. 3 C. 5 D. 15

Quiz Question 2

```
mysql> select * from Employee;
+---------+-----------+---------+
| fname   | lname     | dept_id |
+---------+-----------+---------+
| Michael | Smith     |       3 |
| Susan   | Barker    |       3 |
| Robert  | Tyler     |       3 |
| Susan   | Hawthorne |       1 |
| John    | Gooding   |       2 |
+---------+-----------+---------+
```

```
mysql> select * from Department;
+---------+----------------+
| dept_id | name           |
+---------+----------------+
|       1 | Operations     |
|       2 | Loans          |
|       3 | Administration |
+---------+----------------+
```

How many **columns** does the following query return?

```
SELECT e.*, d.*
FROM Employee e JOIN Department d using (dept_id)
```

A. 0　　B. 3　　C. 4　　D. 5

Quiz Question 3

When can the `using` subclause be used in a join between two tables?

A. Only when the join column has the same name in both tables
B. Only when doing a Cartesian product (or cross join)
C. Only when doing an inner join
D. Only when doing an outer join
E. None of the above

Quiz Question 4

Which one is equivalent to a `JOIN` between two tables?

A. A left outer join
B. A right outer join
C. A full outer join
D. A cross join
E. An inner join

Quiz Question 5

```
mysql> select * from Customer;
+---------+-------------+
| cust_id | fed_id      |
+---------+-------------+
|       1 | 111-11-1111 |
|       2 | 222-22-2222 |
|       3 | 333-33-3333 |
|       4 | 444-44-4444 |
|       5 | 555-55-5555 |
+---------+-------------+
```

```
mysql> select * from Account;
+---------+------------+------------+
| cust_id | account_id | product_cd |
+---------+------------+------------+
|       1 |          1 | CHK        |
|       1 |          2 | SAV        |
|       2 |          3 | CD         |
|       3 |          7 | CHK        |
+---------+------------+------------+
```

How many **rows** does the following query return?

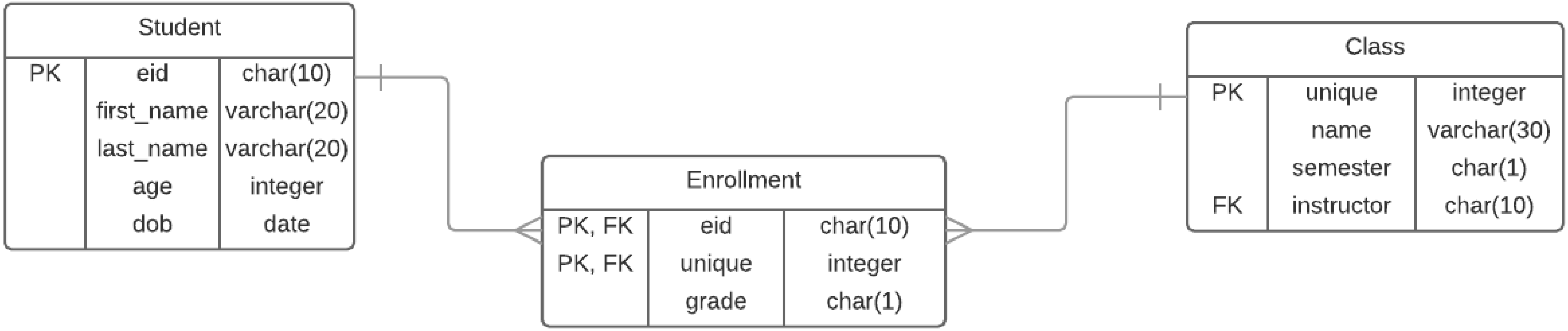
```
SELECT *
FROM Customer c LEFT OUTER JOIN Account a
ON c.cust_id = a.cust_id;
```

A. 3　　B. 4　　C. 5　　D. 6

RDBMS = Joins

Suppose we want to know who was enrolled in CS 327E last Spring. How do we answer this query from this schema?

Logical ERD - UT Class Enrollment



Student		
PK	eid	char(10)
	first_name	varchar(20)
	last_name	varchar(20)
	age	integer
	dob	date

Enrollment		
PK, FK	eid	char(10)
PK, FK	unique	integer
	grade	char(1)

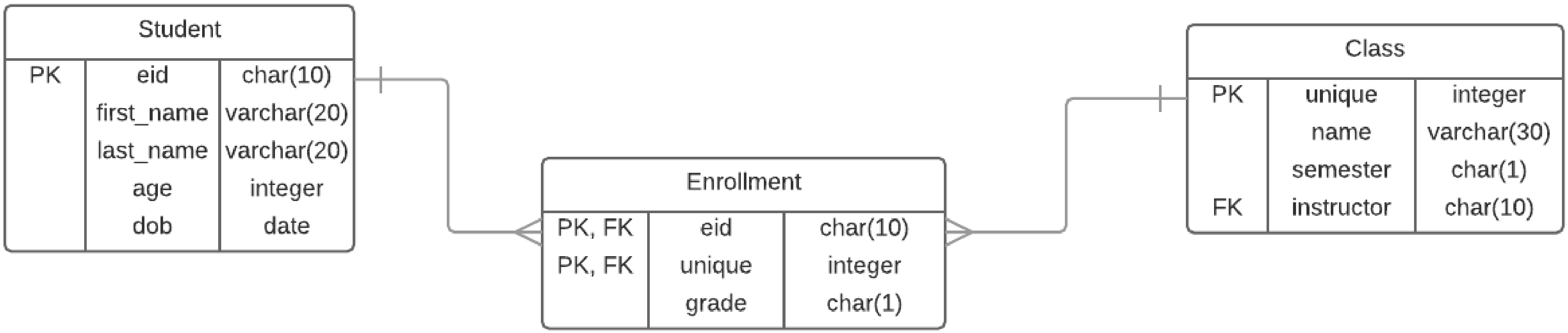
Class		
PK	unique	integer
	name	varchar(30)
	semester	char(1)
FK	instructor	char(10)

Assume:

- 'CS 327E' in `Class.name`
- 'Spring 2016' in `Class.semester`

RDBMS = Joins

Suppose we want to know who was enrolled in CS 327E last Spring. How do we answer this query?



Student		
PK	eid	char(10)
	first_name	varchar(20)
	last_name	varchar(20)
	age	integer
	dob	date

Enrollment		
PK, FK	eid	char(10)
PK, FK	unique	integer
	grade	char(1)

Class		
PK	unique	integer
	name	varchar(30)
	semester	char(1)
FK	instructor	char(10)

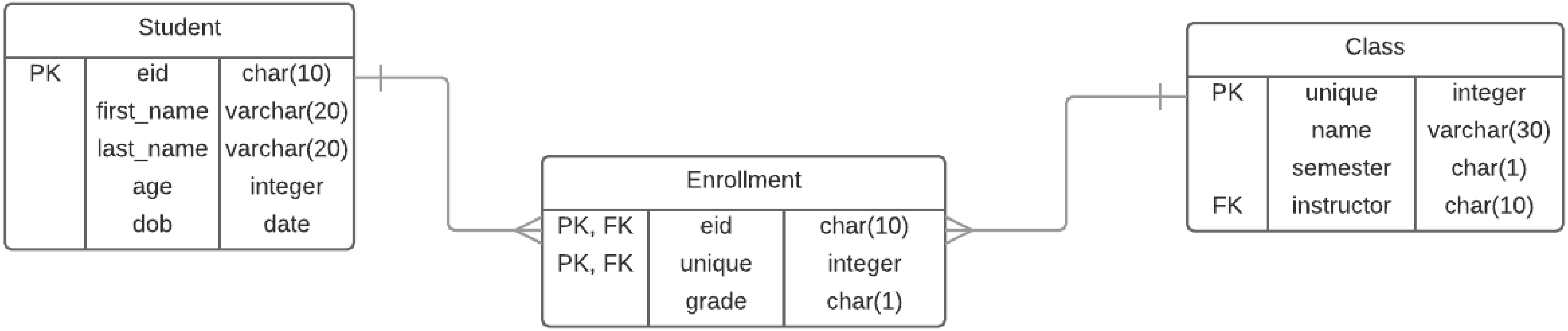
Assume:

- 'CS 327E' in `Class.name`
- 'Spring 2016' in `Class.semester`

```
select *
from Student s join Enrollment e
on s.eid = e.eid join Class c
on e.unique = c.unique
where c.name = 'CS 327E'
and c.semester = 'Spring 2016'
order by s.eid
```

Concept Question 1

Suppose we replace the inner join on Student and Enrollment in the query below with a left-outer join. What additional results would we get back if any?



Student

PK	eid	char(10)
	first_name	varchar(20)
	last_name	varchar(20)
	age	integer
	dob	date

Enrollment

PK, FK	eid	char(10)
PK, FK	unique	integer
	grade	char(1)

Class

PK	unique	integer
	name	varchar(30)
	semester	char(1)
FK	instructor	char(10)

A. Enrollments without students
B. Other class enrollments (e.g. CS 303E, etc.)
C. Students who are not enrolled in CS 327E
D. Students who have taken the class more than once
E. No extra records, nothing would change

```
select *
from Student s join Enrollment e
on s.eid = e.eid join Class c
on e.unique = c.unique
where c.name = 'CS 327E'
and c.semester = 'Spring 2016'
order by s.eid
```

Concept Question 2

Suppose we work for an employment agency and we want to find all job candidates who are skilled in both 'MySQL' and 'Python'. How can we do that based on the table definition below?

```
CREATE TABLE CandidateSkills
(
 candidate_id INTEGER NOT NULL,
 skill_code CHAR(15) NOT NULL,
 PRIMARY KEY (candidate_id, skill_code)
);

INSERT INTO CandidateSkills VALUES(1, 'MySQL')
INSERT INTO CandidateSkills VALUES(1, 'Python')
INSERT INTO CandidateSkills VALUES(2, 'Python')
INSERT INTO CandidateSkills VALUES(3, 'MySQL')
INSERT INTO CandidateSkills VALUES(3, 'Oracle')
```

A. Use this query:

```
SELECT candidate_id
FROM CandidateSkills
WHERE skill_code = 'MySQL'
OR skill_code = 'Python'
```

B. First, select all candidates with MySQL and second, select all candidates with Python. Take the intersection of the two result sets inside the application

C. Use this query:

```
SELECT c1.candidate_id
FROM CandidateSkills as c1,
CandidateSkills as c2
WHERE c1.candidate_id = c2.candidate_id
AND c1.skill_code = 'MySQL'
AND c2.skill_code = 'Python'
```

D. Use this query:

```
SELECT candidate_id
FROM CandidateSkills
WHERE skill_code IN ('MySQL',
'Python')
```

Concept Question 3

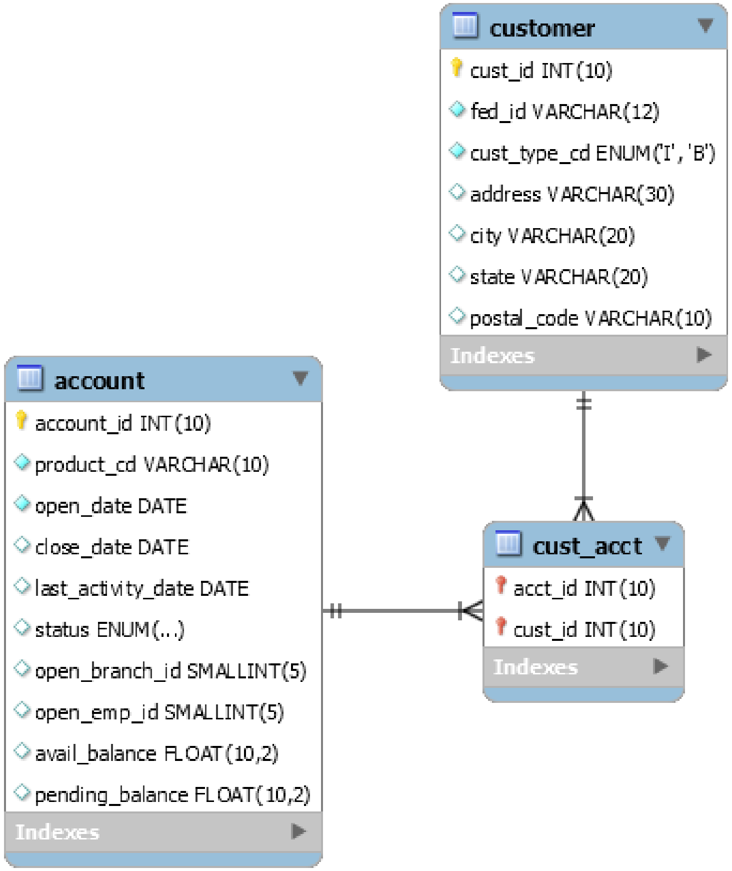
Suppose we change the relationship between `Customer` and `Account` in the Bank database from 1:n to m:n. Unfortunately, we lose some data integrity with this change. How? Which of these queries will find all “orphan” accounts in the database (i.e. accounts without owners)?

```
A.  select a.account_id, ca.acct_id
    from Account a join Cust_Acct ca
    on a.account_id = ca.acct_id
    where ca.acct_id is not null

B.  select a.account_id, ca.acct_id
    from Account a join Cust_Acct ca
    on a.account_id = ca.acct_id
    where ca.acct_id is null

C.  select a.account_id, ca.acct_id
    from Account a left outer join
    Cust_Acct ca
    on a.account_id = ca.acct_id
    where ca.acct_id is null

D.  select a.account_id, ca.acct_id
    from Cust_Acct ca right outer
    join Customer c
    on ca.customer_id = c.cust_id
    where ca.acct_id is null
```



Homework for Next Time

- Read chapter 8 from the book
- Exercises at the end of assigned chapter