Kaleb Himes
kaleb.himes@gmail.com
kaleb@wolfssl.com
CSCI 440, Fall 2014
Sep 26th

Problem 1: "Terms of Endearment" [15 pts]

(a) What is the degree of the Instructors relation?

Degree ↔ # of columns so the degree of instructors is 4.

(b) What is the cardinality of the tuples of Instructors relation?

Cardinality ↔ # of rows so the cardinality of instructors is 25

(c) This is not a flat relational model. Explain why not.

Composite and Multi-valued attributes are not allowed in a flat relational model. To see why this is not a flat relational model one has only to glance at the Degree Column to see this.

(d) Propose a solution to this problem.

1. Split the degree column into multiple columns and name the columns BA, BS, MS, … etc. then constrain those columns so they can only contain 1 of 2 possible values, YES or NO.
2. Or we could just allow for multiple entries of the same Instructor IE:
Enright, Reed, AB, English
Enright, Reed, MA, English

(e) For each of the attributes, give datatype for each (see section 4.1.3). Briefly justify your choice.

| | |
|---|---|
| Lname: | VARCHAR (Not numeric, bit, boolean, date, timestamp etc.) |
| Fname: | VARCHAR (Not numeric, bit, boolean, date, timestamp etc.) |
| Degree: | VARCHAR (Not numeric, bit, boolean, date, timestamp etc.) |
| Department: | VARCHAR (Not numeric, bit, boolean, date, timestamp etc.) |

(f) Considering only the records given to you, determine if each attribute satisfies the UNIQUEness constraint.

| | |
|---|---|
| LName = | NO (two Banks) with different Fname, different degrees and different dept |
| Fname = | NO (two Justin) with different Lname, different degrees and different dept |
| Degree = | NO many repeated identical values that are not equal (IE BS in Ecology != BS in Software Engineering) |
| Department = | YES (even though there's two Forestry, that Forestry is the same dept both times) |

(g) Considering only the records given to you, determine if each attribute satisfies the NOT NULL constraint.

| | |
|---|---|
| Lname = | YES |
| Fname = | NO (pointer has NULL Fname) |
| Degree = | NO (Ricky Bridges has NULL degree) |
| Department = | NO (Justin Case has NULL department) |

(h) Is the NOT NULL constraint an entity integrity constraint or a referential integrity constraint?

It is an entity integrity constraint (3.2.4)

Problem 2: "One Key to Rule Them All"' [12 pts]

(a) List any and all candidate keys.

1. Lname
2. Fname
3. Department

(b) Give a minimal superkey for the Instructors relation. Justify.

A Minimal Superkey I would choose would be {Department, Lname, Fname} .

Where first names and last names can all be repeated (and Degrees but they repeat too often to be used in the minimal super key) therefore we would need all three keys in the minimal superkey. Notice for instance:
Tsofrenic, S 'Kit', Psychology
Tsofrenic, S 'Kit', Neuroscience

(c) Is your minimal superkey single or composite?

It is a composite

(d) Which attribute is the primary key (PK)? How do you know?

LName is the primary key in the table given us and we know this because it is Underlined!

(e) Is the PK single or composite?

It is singular in the table given

(f) Why is this PK a poor choice? Explain. Propose a new PK. Does your solution require changing the schema, the constraints, or both? Explain.

Lname is actually the optimal choice in this situation. According to 3.2.4 on Integrity, Referential Integrity, and Foreign keys: (also reference 1(h) above)

*"The entity integrity constraint states that **no primary key** value can be NULL . This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify some tuples."*

*-Fundamentals of Database Systems 6th Edition Chapter 3 Section 2.4*

I can not propose another PK because every other Column contains at least one null value such that we would be unable to identify some tuples.
This does not require changing the schema, or constraints.

Problem 3: “Relationships are Hard” ’ [30 pts]

(a) Which attribute would most logically be used as a foreign key to another relation R?

I think {Lname, Fname}would be a good foreign key. Our other choices, department, and Degree may not be mentioned in other tables, however to link data to that individual you would want both Fname and Lname, to be referenced in another table. You would need both since neither in and of itself is unique.

(b) Describe, in English, the relation R that uses your aforementioned foreign key. What name would you give to relation R?

The Relation R I have in mind would be something like “Instructors Contact Info” (INSTRUCTORS.CONTACT_INFO and would contain fields such as Cell, Home, email, etc. In order to tie this information to the correct instructor we would need to know their first and last name but we could easily tie this to the correct department by referencing the table provided in the HW.

(c) Give a CREATE TABLE command for your new relation R, including the types and any constraints.

```
CREATE TABLE CONTACT_INFO (
        Lname       VARCHAR(20)
        Fname       VARCHAR(20)
        Cell        VARCHAR(16)     #account for 1-(406)-555-1234
        Home        VARCHAR(16)
        email       VARCHAR(50)     #can be long
        PRIMARY KEY (email),
        FOREIGN KEY (Lname, Fname) REFERENCES INSTRUCTORS(Lname,  Fname)
)
```

NEXT PAGE

(d) Give example tuples for your new relation R, covering at least all the distinct values from the attribute in Instructors you chose as your foreign key.

| Lname | Fname | Cell | Home | email |
|---|---|---|---|---|
| | | | | |
| Enright | Reed | 406-555-1234 | 406-555-9101 | r.enright@example.com |
| Thyme | Justin | 406-555-4321 | 406-555-1019 | j.thyme@example.com |
| Hebbra | Al G. | 406-555-5678 | 406-555-1121 | a.g.hebbra@example.com |
| Dwyer | Barb | 406-555-8765 | 406-555-1211 | b.dwyer@example.com |
| Pointer | NULL | 555-2030 | 555-4040 | point@example.com |
| Banks | River | 555-3020 | 555-4444 | r.banks1@example.com |
| Minor | Bea | 555-2687 | 555-4443 | b.minor@example.com |
| Botes | Roe | 555-7862 | 406-555-4442 | r.botes@example.com |
| Kase | St??r | 406-555-1550 | 406-555-4441 | s.kase@example.com |
| Burns | Willett | 555-0551 | 555-4445 | w.burns@example.com |
| Uyi | Guy | 555-2525 | 406-555-4446 | g.uyi@example.com |
| Claire | Heidi | 208-555-5252 | 555-4447 | h.claire@example.com |
| Paciter | Kay | 406-555-7121 | 555-4448 | k.paciter@example.com |
| Tsofrenic | S `Kit' | 406-555-1271 | 406-555-4449 | s.tsofrenic@example.com |
| Major | Arthur | 406-555-6667 | 555-3331 | a.major@example.com |
| Case | Justin | 555-7666 | 406-555-3332 | j.case@example.com |
| Wyle | Drew | 555-7777 | 555-3334 | d.wyle@example.com |
| Oakey | Carrie | 406-555-7766 | 405-555-3335 | c.oakey@example.com |
| Furr | Douglas | 555-2082 | 555-3336 | d.furr@example.com |
| Power | Max | 406-555-2802 | 555-3337 | m.power@example.com |
| Banks | Robyn | 555-2222 | 406-555-3338 | r.banks2@example.com |
| Vine | Beau | 555-1111 | 555-3339 | v.beau@example.com |
| Domacese | Rand | 555-3333 | 406-555-5559 | r.domacese@example.com |
| Bridges | Ricky T. | 555-9999 | 555-5558 | r.bridges@example.com |

(e) Propose and describe a third relation S of your own choosing. However, you must connect this relation to Instructors and your relation R via foreign keys. What name would you give relation S?

Relation S = INSTRUCTORS.AVAILABILITY

(f) Give a CREATE TABLE command for your new relation S, including the types and any constraints.

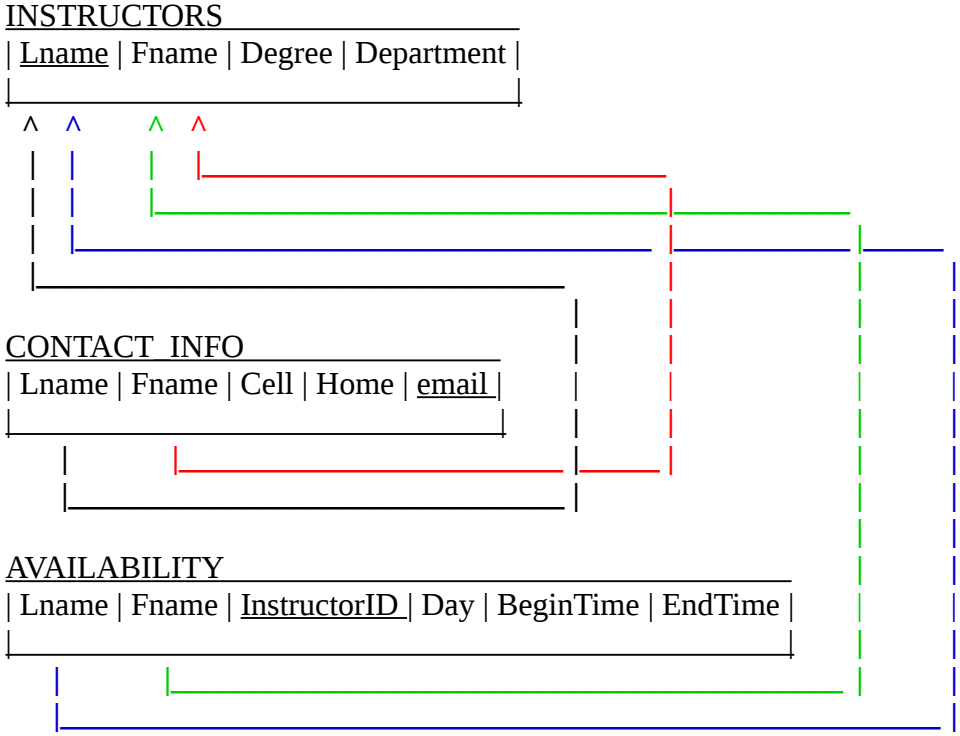
```
CREATE TABLE AVAILABILITY (
        Lname       VARCHAR(20)
        Fname       VARCHAR(20)
        Day         VARCHAR(10)     #Can be Wednesday, or just Wed
        BeginTime   VARCHAR(12)     #08:00:00 AM
        EndTime     VARCHAR(12)

        PRIMARY KEY (InstructorID),
        FOREIGN KEY (Lname, Fname) REFERENCES INSTRUCTORS(Lname,  Fname),
                                   REFERENCES CONTACT_INFO(Lname, Fname)
    )
```

NEXT PAGE

(g) Give (at least) five example tuples for your new relation S.

| Lname | Fname | InstructorID | Day | BeginTime | EndTime |
|---|---|---|---|---|---|
| | | | | | |
| Enright | Reed | 1 | Mon | 08:00:00 AM | 10:00:00 AM |
| Thyme | Justin | 1 | Wed | 12:00:00 PM | 01:30:00 PM |
| Hebbra | Al G. | 1 | Fri | 01:00:00 PM | 02:30:00 PM |
| Dwyer | Barb | 2 | Tues | 08:00:00 AM | 10:00:00 AM |
| Pointer | NULL | 3 | Thur | 09:00:00 AM | 10:00:00 AM |
| Banks | River | 4 | Tues | 01:00:00 PM | 02:00:00 PM |
| Minor | Bea | 4 | Thur | 03:00:00 PM | 04:00:00 PM |
| Botes | Roe | 5 | Mon | 08:00:00 AM | 09:00:00 AM |
| Kase | St??r | 5 | Mon | 11:00:00 AM | 12:00:00 PM |
| Burns | Willett | 5 | Mon | 03:00:00 PM | 04:00:00 PM |

(h) Draw a schema diagram for all three of these relations. Make sure to mark your PK for each relation. Show the relations using arrows. Use Figure 3.7 on p.75 as your guide. You will use these three relations (Instructors, R, and S) in the remaining problems. In your responses, refer to your names for these relations.




NEXT PAGE

2Problem 4: “Pick Me! Pick Me!” ’ [12 pts]

(a) Write a SELECT command that returns the tuples in the Instructors relation of all persons in the Computer Science department.

```
SELECT      *
FROM        INSTRUCTORS
WHERE       Department='Computer Science'
```

(b) Modify your last query to return only the first and last names of the persons in the Computer Science department.

```
SELECT      Fname, Lname
FROM        INSTRUCTORS
WHERE       Department='Computer Science'
```

(c) Modify your last query to sort the results alphabetically by last name.

```
SELECT      Fname, Lname
FROM        INSTRUCTORS
WHERE       Department='Computer Science' ORDER BY Lname;
```

(d) One last improvement. . . Modify your last query to return instructors from ALL departments, and include the Department attribute in your results. Group the results by Department name. Furthermore, within each Department grouping, sort your results alphabetically by last name.

```
SELECT      Fname, Lname, Department
FROM        INSTRUCTORS
ORDER BY    Department,Lname;
```

Problem 5: “The Faithful Departed” [6 pts]

(a) Write a query that lists only the unique department names at this University.

```
SELECT DISTINCT Department
FROM INSTRUCTORS
ORDER BY Department;
```

(b) Write a new SELECT command that returns the first and last name of all Instructors in the Engineering fields (including Computer Science), using only the Instructors relation.

```
SELECT DISTINCT Fname, Lname
FROM Instructors
WHERE Degree LIKE '%Computer Science%' OR Degree LIKE '%Engineering%';
```

Problem 6: “No Tuple For You!” [6 pts]

(a) Instructor Burns has been assigned an extended sabbatical [20 years without parole] . Write a command to remove Mr. Burns from the Instructor relation.

```
DELETE FROM INSTRUCTORS
WHERE Lname='Burns' AND Fname='Willett' AND Department='Forestry';
```

(b) Unfortunately, because of the negative press, the College is forced to close the Forestry Department. Write an SQL command to remove all the Forestry instructors from the Instructors relation. If relevant, also remove any tuples pertaining to Forestry from your relations R and/or S.

```
DELETE FROM Instructors, Contact_info, Availability
USING Instructors INNER JOIN Contact_info, INNER JOIN Availability
WHERE Instructors.Lname='Furr' AND Instructors.Fname='Douglas'
        AND Contact_info.Lname = Instructors.Lname
        AND Contact_info.Fname = Instructors.Fname
        AND Availability.Lname = Instructors.Lname
        AND Availability.Fname = Instructors.Fname;

DELETE FROM Contact_info, Availability
USING Contact_info INNER JOIN Availability
WHERE Contact_info.Lname='Burns' AND Contact_info.Fname='Willett'
        AND Availability.Lname = Contact_info.Lname
        AND Availability.Fname = Contact_info.Fname;
```

3Problem 7: “I Thought You Looked Familiar” [7 pts]

(a) Write a (single) query to return the names of all people that share the same first name with another first name in the DB and all those people that share the same last name with another last name in the DB. Note that they do not need to share both first and last name, merely either or. You should eliminate duplicates.

| | |
|---|---|
| Banks | River |
| Banks | Robyn |
| Tsofrenic | S 'Kit' |
| Case | Justin |
| Thyme | Justin |

(Sample Results)

HINT: You might find aliasing useful here. It is possible to join a table with itself, e.g., FROM Instructors AS I1, Instructors AS I2

```
SELECT  Lname, Fname
FROM Instructors AS I1, Instructors AS I2
WHERE I1.Lname=I2.Lname OR I1.Fname=I2.Fname;
```

Problem 8: “Flying High” [4 pts]

(a) Willie Flye (BS, PhD) has just been hired as the instructor of the newly created department of Aeronautical Engineering. Write a command to add Dr. Flye to Instructors. Add additional tuples to your relations R and S if needed.

```
INSERT INTO Instructors (Lname, Fname, Degree, Department)
VALUES ('Flye','Willie','BS , PhD','Aeronautical Engineering');

INSERT INTO Contact_info (Lname, Fname, Cell, Home, email)
VALUES('Flye','Willie','208-555-8393','406-555-1299','w.flye@example.com');

INSERT INTO Availability(Lname, Fname, InstructorID, Day, BeginTime, EndTime)
VALUES('Flye', 'Willie', '23', 'Mon', '08:00:00 AM', '12:00:00 PM');
```

Problem 9: “Parallel Keys” [4 pts]

(a) Oh Happy Day! Ms. Minor has announced her recent marriage to the Art instructor. She has decided to take his surname. Write an query that changes her last name to his last name.

```
UPDATE Instructors
SET Lname = 'Major'
WHERE Lname ='Minor';

UPDATE Contact_info
SET Lname = 'Major'
WHERE Lname = 'Minor';

UPDATE Availability
SET Lname = 'Major'
WHERE Lname = 'Minor';
```

Problem 10: “The Case in Point” [4 pts]

(a) Following an audit of the database, it was discovered that the department name of the Law instructor, Mr. Case, is not correct. Write a query to fix this mistake.

```
UPDATE Instructors
SET Department = 'Criminal Justice' #(Law)
WHERE Lname = 'Case' AND Fname = 'Justin';
```