CS44800 Fall 2021 Midterm 1 solutions, October 14, 2021
Prof. Chris Clifton

Turn Off Your Cell Phone. Use of any electronic device during the test is prohibited. As previously noted, you are allowed notes: Up to two sheets of 8.5x11 or A4 paper, single-sided (or one sheet double-sided).

Time will be tight. If you spend more than the recommended time on any question, **go on to the next one**. If you can't answer it in the recommended time, you are either giving too much detail or the question is material you don't know well. You can skip one or two parts and still demonstrate what I believe to be an A-level understanding of the material.

Note: It is okay to abbreviate in your answers, as long as the abbreviations are unambiguous and reasonably obvious.

In all cases, it is important that you give some idea of how you derived the answer, not simply give an answer. Setting up the derivation correctly, even if you don't carry out the calculations to get the final answer, is good for nearly full credit.

The last page of the exam contains an example database that you will use for several of the questions. You may tear it off for easy reference.

Solutions given in boldface. Additional explanation in plain text. *Scoring rubric in italics.*

1 Query Languages (6 minutes, 9 points)

Given following schema:

- **MovieStar**(name: string, address: string, gender: char, birthdate: date)
- **StarsIn**(movieTitle: string, movieYear: date, starName: string)

Express the following queries (given as SQL queries, Relational Algebra, or an English Description) in the other forms (e.g., for Part A, give an English-language description and an SQL query.)

A. $\Pi_{name}\sigma_{(gender='f' \wedge birthdate=1990)}(MovieStar)$

SELECT name FROM MovieStar WHERE gender='f' AND birthdate=1990

Give the names of female movie stars born in 1990.

B.
```
SELECT movieTitle
FROM StarsIn , MovieStar
WHERE StarsIn.starName = MovieStar.name AND birthdate = 1960;
```

Find the movies starring movie stars born in 1960. Note that this will list the movie once for each star born in 1960 in the movie.

$\Pi_{movieTitle}(StarsIn \bowtie_{starName=name} (\sigma_{birthdate=1960}MovieStar))$
Note that this requires we interpret this as multiset semantics, otherwise it would be equivalent to a "select distinct" ...

C. Name the movie stars who starred in movies made from from 1970 to 1980.

$\Pi_{name}((\sigma_{movieYear \geq 1970 \wedge movieYear \leq 1980}StarsIn) \bowtie_{starName=name} MovieStar)$
Note: The join isn't the only way to do this, see below.

SELECT starName FROM StarsIn WHERE movieYear between 1970 and 1980

Scoring: 1 for showing some understanding of query (in either translation), 1 each for reasonably close answer

2 Functional Dependencies (5 minutes, 4 points)

Given the schema $R = ABCDEFG$ with functional dependencies:
$A \rightarrow CDE$
$BC \rightarrow EF$

A. Derive one additional non-trivial functional dependency that holds given the two you are given. Show your derivation.

$AB \rightarrow EF$: $A \rightarrow C$ (first FD given), $AB \rightarrow BC$ (augmentation), $AB \rightarrow EF$ (transitivity and the second FD given)

Scoring: 1 for correct, one for derivation. One point for correct derivation of a trivial FD.

B. Can you derive any non-trivial functional dependencies involving G? Explain your answer.

No. Since no FDs involve G, the only rule that could be used would be to create trivial FDs involving G, e.g., ones where G is on both sides, or G is on the left hand side of a rule that already exists.

Scoring: 1 for no, one for explanation.

3 Normalization (4 minutes, 4 points)

A. Give a lossless-join decomposition of R in Boyce-Codd Normal Form for the functional dependencies in Question 2.

R1 = ACDE , R2 = ABFG . This is lossless because A is a superkey for $ACDE$, and appears in $ABFG$. It is in BCNF because for any FD that applies to $R1$ or $R2$, the left side is a superkey. Since no relation contains BC, nothing from the second FD applies to $R1$ or $R2$, so id doesn't violate BCNF.

Scoring: 2 for correct, 1 if on the right track but incorrect

B. Give a lossless-join decomposition of R in Third Normal Form for the functional dependencies in Question 2.

Technically, the $R1$ and $R2$ from the previous part are in BCNF, so they are also in 3NF. But a better decomposition would be dependency preserving. The 3NF composition algorithm gives R1=ACDE, R2=BCEF, R3=ABG as a dependency-preserving, lossless join, 3NF decomposition. Note that this is also BCNF...

Scoring: 2 for correct, 1 if on the right track but incorrect

4 Lossless Join Decomposition (4 minutes, 2 points)

Is the following decomposition of the relation R using the functional dependencies from Question 2 ($R = ABCDEFG$, $A \rightarrow CDE$, $BC \rightarrow EF$):

$R1 = ACDE$
$R2 = BCEFG$

A. A lossless join decomposition?

No. Any superkey for $R1$ includes A, which does not appear in $R2$. Any superkey for $R2$ must include G, which does not appear in $R1$.

Scoring: 1 for good explanation

B. A dependency preserving decomposition?

Yes. We can check if both dependencies are preserved by looking at a single relation. Better still, we could make A a key for $R1$, and BCG a key for $R2$, and the dependencies would be enforced by ensuring that those are keys. Unfortunately, since it isn't a lossless join, it isn't good anyway.

Scoring: 1 for showing understanding of dependency preserving

In both cases, explain/prove your answer.

5 Query Execution (8 minutes, 8 points)

For the following queries, give the output of running the query on the tables on page 7.

A.
```
SELECT CRN, Department, Room
FROM Courses
WHERE Enrollment > 100;
```

26857, CS, FRNY G140

B. $\Pi_{Room,Capacity}(\sigma_{Building=Krannert}Rooms)$

KRAN G016, 120

C.
```
SELECT Courses.CRN, Courses.Department, Courses.Number
FROM Courses, Rooms
WHERE Courses.Enrollment > Rooms.Capacity
        AND Courses.Room = Rooms.Room;
```

17648, STAT, 30100

D. $\Pi_{CAREER,Name}(\sigma_{depts>1}(\gamma_{Name,count(*)depts}Instructors))$

This is actually an illegal query - the projection over CAREER is not possible, since the output of the group by does not include CAREER. But the single tuple
neville, Jennifer Neville
was also accepted, for those who didn't look closely at the "fine print".

Scoring: 2 correct, 1 showing some understanding

6 Keys (8 minutes, 5 points)

For the dataset on page 7:

A. Give a list of all candidate keys for the **Courses** relation that hold on the given data.

CRN is a key, but no other single value is unique. Room and Time together form a key, as do Number and Room or Number and Time. Room and Credits form a key, as do Enrollment and Number or Enrollment and Time. We can see that Department doesn't distinguish anything except with something that is already a key. Anything else forming a key has a subset as a key, so is not a *candidate* key.

Scoring: 1 for only candidate keys, 1 for more than just obvious, 1 for reasonably complete

B. Can CAREER+Name+Department be a candidate key or primary key of the **Instructors** relation (page 7)? Please explain why or why not.

No. While it is a superkey, since
$CAREERDepartment \rightarrow CAREERNameDepartmentCRN$,
there is a subset of that key that is also a superkey, so it is not a *candidate* key.

Scoring: 1 for No, 1 for some understanding of superkey vs. candidate key, 1 for explaining in terms of FDs

7 Data Definition (3 minutes, 4 points)

Give the SQL create table statement appropriate to create the table **Instructors** for the relation on page 7. Include appropriate data types, keys, etc.

CREATE TABLE Instructors (
CAREER char(8) not null,
Name varchar(40),
Department varchar(4) not null,
Office varchar(10),
CRN number,
primary key (CAREER, Department)
foreign key CRN references Courses.CRN
)

Scoring: One for all attributes, one for key definitions, one for reasonable data types, one for something else (not null, unique, foreign key)

8 Functional Dependencies

8.1 Closure (8 minutes, 6 points)

A. Give the closure of CAREER ($CAREER^+$) **based on what holds in the data given.**

This depends on how you understand dealing with nulls. If you assume that null means it could be anything, and thus you can't make any claim about functional dependencies involving columns with null values, then $CAREER^+$ → **CAREER Name Office**. If you ignored the nulls, then $CAREER^+$ → **CAREER Name Office CRN Courses.Department Number Courses.Room Time credits Enrollment Rooms.Room Building Address Capacity**. As an obvious check, if you had CAREER → CRN, then $CAREER^+$ would certainly have to include everything in your answer to CRN^+, from transitivity.

Scoring: 3 for just CAREER Name Office, or 1 for getting a reasonable set within Instructors, 1 for going beyond a single relation, 1 for reasonably close

B. Give the closure of CRN (CRN^+) based on what holds in the data given.

CRN^+ → **Courses.Department Number Courses.Room Time credits Enrollment Rooms.Room Building Address Capacity** Depending on how you thought about nulls, Instructors.Department could be included as well.

Scoring: 1 for complete on courses relation, 1 for going beyond a single relation, 1 for reasonably close

8.2 Additional FDs (4 minutes, 4 points)

A. List one functional dependency that holds on the data given, that includes Courses.Time on the left hand side of the dependency.

Time Room → CRN

Scoring: 1 for holds, 1 for includes Courses.Time and something else or Time → enrollment

B. List one functional dependency that holds on the data given, that includes attributes from more than one of the relations.

Office → Department Number: If I know the Instructor's office, I know the room and time of the course. This assumes every instructor has a private office, and teaches only one course.

Scoring: 1 for correct FD, 1 for involves multiple relations

8.3 Appropriate Functional Dependencies (3 minutes, 2 points)

List one additional functional dependency (not listed in 8, 1 or 2 that you believe should hold in this schema, and briefly explain why.

Room → Capacity: If I know the room, I know how many people it can hold.

Scoring: 1 for anything that holds on the data, or if it doesn't hold, your explanation suggests you realize it doesn't hold on the given data but should, 1 for a reasonable explanation.

9 Multivalued Dependencies (2 minutes, 2 points)

List up to two multivalued dependencies that hold on the data, or briefly explain why there are none.

The simple answer is to list any two functional depndencies, since an FD is also an MVD. But CAREER →→ Department, and it's complement CAREER →→ Name Office CRN (and everything in the closure of CRN) is more interesting.

Scoring: One point for an answer showing some understanding of multivalued dependencies. One for catching something that isn't FDs or knowing that all FDs are also MVDs.

10 Normal Forms (8 minutes, 8 points)

Given only the dependencies in your answer to Question 8, answer the following. Explain your answers.

A. Is the relation **Rooms** in Boyce-Codd Normal Form?

This depends on what you list, answer could be yes or no. **Not counting my answer to 8.3, the answer would be yes - there is no functional dependency where the left side is contained in the relation Rooms, so nothing violates BCNF. But since I said Room → capacity, that is violated, and the answer is no. If I'd listed Room → building address capacity, then the answer would be yes. But if I also had building → address, then the answer would be no.**

Scoring: 1 for correct given answer to 8, 1 for why

B. Is the relation **Instructors** in Boyce-Codd Normal Form?

Since CAREER → Name Office, CAREER would need to be a key for Instructors. Since it isn't, Instructors isn't in BCNF. Note that it wasn't enough to just say "CAREER isn't a key for instructors", you need to also note that there is an FD being violated. Simply the fact that a particular attribute isn't a key, or that there is no key at all, doesn't mean it violates BCNF.

Scoring: 1 for no, 1 for why in either BCNF question

C. Is the relation **Rooms** in Third Normal Form?

If the answer to A is Yes, then it is also in 3NF. If no (as in my case), then it is more difficult. Since Room is not a superkey (I can't derive the FD Room → Building Address Capacity), then Capacity would need to be part of a candidate key. It isn't (based on the FDs, even though it is in the data), so Room → Capacity causes Rooms to violate 3NF.

Scoring: 1 for yes/correct, 1 for why in either 3NF question

D. Given your answer to Question 9, are any of the relations in Fourth Normal Form? Briefly explain.

If there are no MVDs other than FDs, then anything in BCNF is in 4NF, so Courses is in BCNF and 4NF. The MVD I list (CAREER *rightarrow* **Department) causes Instructors to violate 4NF, even if it was in BCNF.**

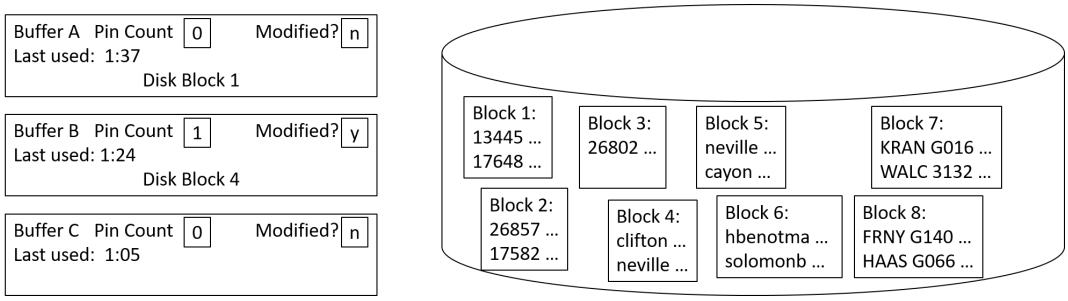
Scoring: 1 for being consistent with the answer to MVD or BCNF, 1 for explanation corresponding to the answer.

11 Buffer Management (10 minutes, 8 points)

Assume that the database on Page 7 is stored on disk with **Courses** in disk blocks 1, 2 and 3, **Instructors** in blocks 4, 5 and 6, and **Rooms** in blocks 7 and 8 (see figure). Assume at most two records can be stored in one disk block. The initial state of the buffer is as shown: There are three buffers, Buffer A has disk block 1, Buffer B has disk block 4 (which is modified and pinned by another query), and Buffer C is unused.

Explain the sequence of reads and writes needed for each of the following queries, and the state of the buffers at the end of the query. Assume no indexes (you need to scan through the entire file.) For each query, assume the starting state as shown. Assume Least Recently Used (LRU) replacement.

A.
```
SELECT Room, Capacity
FROM Rooms
WHERE Building='Haas';
```



A full scan of Rooms goes through Blocks 7 and 8. We can load Block 7 into Buffer C, since it isn't used (and was the least recently used). We then load Block 8 into Buffer A. Since Buffer B is pinned, we can't replace it, even though it would be the least recently used. This is a total of two reads, 0 writes (although Buffer B will eventually need to be written to Block 4.)

Scoring: 1 for leaving pinned, 1 for reading all, 1 for no writes, 1 for leaving correct

B.
```
UPDATE Instructors
SET CRN = 26857
WHERE CRN IS NULL;
```

(start with the same diagram as above)

Instructors is blocks 4, 5, and 6. Block 4 is already in memory, so we can scan it and update the neville record to set CRN to 26857 (setting the modified bit, although it is already set.) We can now unpin (which reduces PIN count to 1), and read block 5 into buffer C (the least recently used.) We again update the neville record in Block 5 (Buffer C) and set the modified bit. We can then unpin. Buffer A then becomes least recently used, so we read Block 6 into Buffer A, discover no changes, and unpin it. This is a total of two reads and 0 writes, although buffers B and C will eventually need to be written to blocks 4 and 5.

Scoring: 1 for all reads, 1 for write if needed or setting modified bit, 1 for not rereading block 4, 1 for not making any other errors

Example tables for Questions 5-10

Consider a database with tables **Courses**, **Instructors** and **Rooms**.

You may separate this page for reference. You don't need to turn this page in unless you've written something on it you want us to look at when grading. If you do turn it in, make sure your name/CAREER account is on it.

Courses

CRN	*Department*	*Number*	*Room*	*Time*	credits	*Enrollment*
13445	CS	44800	KRAN G016	TR 13:30	3	80
17648	STAT	30100	WALC 3132	MW 11:30	3	44
26857	CS	34800	FRNY G140	MW 18:00	3	195
17582	STAT	30100	HAAS G066	MW 14:30	4	42
26802	STAT	34800	HAAS G066	TR 13:30	3	42

Instructors

CAREER	*Name*	*Department*	*Office*	*CRN*
clifton	Chris Clifton	CS	HAAS 222	13445
neville	Jennifer Neville	CS	LWSN 2142D	null
neville	Jennifer Neville	STAT	LWSN 2142D	null
cayon	Laura Cayon	STAT	MATH 510	17648
hbenotma	Hisham Benotman	CS	HAAS 146	26857
solomonb	Bruce Solomon	STAT	HAAS 118	17648

Rooms

Room	*Building*	*Address*	Capacity
KRAN G016	Krannert	403 W State St	120
WALC 3132	Wilmeth	340 Centennial Mall	40
FRNY G140	Forney	480 Stadium Mall	234
HAAS G066	Haas	250 N University St	66