CS411: Database Systems **Spring 2018**

Assignment 2

Due Wed, Feb 28 at 11:59pm

General Instructions

- Feel free to talk to other members of the class in doing the homework. You should, however, write down your solutions yourself. *List the names of everyone you worked with at the top of your submission.*
- Keep your solutions brief and clear.
- Please use Piazza if you have questions about the homework but do not post answers. Feel free to use private posts or come to the office hours.

Homework Submission

- We DO NOT accept late homework submissions.
- We will be using Compass for collecting the homework assignments. Please submit your answers via Compass. Hard copies are not accepted.
- Contact the TAs if you are having technical difficulties in submitting the assignment; attempt to submit well in advance of the due date/time.
- The homework must be submitted in **pdf** format. Scanned handwritten and/or hand-drawn pictures in your documents won't be accepted.
- Please do not zip the answer document (PDF) so that the graders can read it directly on Compass. You need to submit one answer document, named as **hw2_netid.pdf**.
- Please see the assignments page for more details. In particular, we will be announcing clarifications, if any, on this page.

1 Short Questions (20 pts)

Provide a short answer (3-4 sentences at most) for each of the following questions. You may use figures if necessary.

1. [2.5] Suppose we have a relation R as given below, representing the exam statistics for course CS411. First project relation R to GPA (i.e., eliminate SID and Name) and then calculate the average GPA, under the set-model and the bag-model respectively. Which model is prefered in this example and why?

SID	Name	GPA
1	James	3
2	Charles	4
3	Doris	4
4	Ada	4

[**A**nswer: With the set-model, we have $\{3, 4\}$, and hence the average is 3.5; while with bag-model, we have $\{3, 4, 4, 4\}$, and hence the average is 3.75. Bag-model is prefered, since it matches the real-world semantics.]

2. [5] Consider a relation $R(A, B, C)$. You may assume there are no null values or duplicates in R. If the result of $\sigma_{Y \neq V}(\rho_{R(X,Y,Z)}R \bowtie \rho_{R(X,V,W)}R)$ is always guaranteed to be empty, then what property of R can you infer? (Hint: think functional dependencies.)
[**A**nswer: $A \rightarrow B$. This is because we renamed the attributes in R and performed a natural join, and consequently pairs of tuples in the renamed relations that agree on X (originally A) are combined. Meanwhile, there are no tuples satisfying the condition $Y \neq V$ (both originally B). Therefore, in the produced relation, with the same X, we must have the same Y and V (B in the original relation), indicating $A \rightarrow B$.]

3. [2.5] Consider any relation R that never contains more than one tuple. Is it true that R must in Boyce-Codd Normal Form (BCNF)? Justify your answer
[**A**nswer: False. Can't tell without seeing the actual schema and FDs, which represent the real-world semantic constraints.]

4. [4] Consider a relation $R(A, B, C, D, E)$ with dependencies $AB \rightarrow CD$, $C \rightarrow AB$ $D \rightarrow AE$, list all minimal keys for R. Also, state whether the relation R is in 3NF **with reasoning**.
[**A**nswer: Minimal keys: 1. AB; 2. C.
No, since in FD $D \rightarrow AE$, D is not a superkey and E is not part of key.]

5. [6] Two sets of functional dependencies (FD's) F and F' are equivalent if all FD's in F' follow from the ones in F, and all FD's in F follow from the ones in F'. Consider the following three sets of functional dependencies:

   - $F1 = A \rightarrow C, B \rightarrow A,$
   - $F2 = B \rightarrow AC$

- $F3 = AB \to C, B \to A$

(a) Are $F1$ and $F2$ equivalent? Justify your answer.

(b) Are $F1$ and $F3$ equivalent? Justify your answer.

(c) Are $F2$ and $F3$ equivalent? Justify your answer.

[**A**nswer:

(a) No. $F1 : A^* = AC, B^* = ABC, C^* = C$; $F2 : A^* = A, B^* = ABC, C^* = C$ Hence, $F1 \Rightarrow F2$, but $F2 \not\Rightarrow F1$ since $A \to C$ does not hold based on $A^* = A$ in $F2$.

(b) No. $F1 : A^* = AC, B^* = ABC, C^* = C$; $F3 : A^* = A, B^* = ABC, C^* = C$. Hence, $F1 \Rightarrow F3$, but $F3 \not\Rightarrow F1$ since $A \to C$ does not hold based on $A^* = A$ in $F3$.

(c) Yes. $F2 : A^* = A, B^* = ABC, C^* = C$; $F3 : A^* = A, B^* = ABC, C^* = C$. Hence, $F2 \Rightarrow F3$, and $F3 \Rightarrow F2$.

Remark: it's also fine if you give a counter example in (a) and (b), and show the relation satisfies one FD set but not the other FD set.]

2 Relational algebra to English (15 pts)

Consider a relation Works (name, company, salary) with no duplicates. Consider the following relational algebra expression, written in linear notation.

$P1(salary) = \pi_{salary}(\sigma_{company=\text{``}IBM\text{''}}(Works))$
$P2(salary) = \pi_{salary}(\rho_{T1(s)}(P1) \bowtie_{s>salary} P1)$
$P3(salary) = P1 - P2$
$Answer(name) = \pi_{name}(Works \bowtie_{salary>s} \rho_{T2(s)}(P3))$

State in English what is computed as the final answer briefly. Long-winded answers will be deducted points. For partial credit, explain what $P1, P2$ and $P3$ contain.
[**A**nswer: Find the names of all people who earn a salary that is higher than the salary earned by any person who works for IBM.
Any answer conveying the same meaning is fine. E.g., "the names of the people whose salaries are higher than the maximum salary among IBM employees".
Specifically: $P1$ depicts all possible salaries for people who work for IBM
$P2$ depicts all the possible salaries except the highest salary for people who work for IBM
$P3$ depicts the highest salary earned by any person who works for IBM
]

3 English to Relational Algebra (20 pts)

Consider the following relational database schema that describes information about students and their courses. A course is uniquely identified by its CODE (e.g., "CS411"), and a student is uniquely identified by his or her SID.
Course(<u>CODE</u>, units, time, room) // all courses
Student(<u>SID</u>, name, level) // all students, level can be "grad" or "undergrad"
Taking(<u>SID</u>, <u>CODE</u>) // current enrollment information

Write a relational algebra expression to list the information (i.e., CODE, units, time, room) of courses that are currently offered but have no graduate students enrolled.
[**A**nswer: $Course - Course \bowtie (\pi_{CODE}(Taking \bowtie (\pi_{SID}(\sigma_{level=\text{``}grad\text{''}}(Student)))))$
or $Course - Course \bowtie (\pi_{CODE}(Taking \bowtie (\sigma_{level=\text{``}grad\text{''}}(Student))))$
Remark: it is also fine if you break the above relational algebra into small pieces, and express in linear notaion.]

4 Data to functional dependency (20 pts)

Consider a relation $R(A, B, C)$, satisfying some functional dependency. Two instances of R are given as below:

A	B	C
2	3	1
2	2	4

A	B	C
2	2	1
3	3	2
4	2	1

Based on R's schema, enumerate all possible completely nontrivial functional dependencies (FDs) with only a single attribute on the right-hand side. Then, based on the instances above, for each FD you listed, label whether it:
H: Definitely holds in R.
NH: Definitely does not hold in R.
CD: Cannot be determined from the information given whether or not it holds in R.
[**A**nswer:

- $A \rightarrow B$ (NH)
- $A \rightarrow C$ (NH)
- $B \rightarrow A$ (NH)
- $B \rightarrow C$ (CD)
- $C \rightarrow A$ (NH)
- $C \rightarrow B$ (CD)
- $AB \rightarrow C$ (CD)
- $AC \rightarrow B$ (CD)
- $BC \rightarrow A$ (NH)

Remarks:

1. FD is a property of real-world. We can not assert a FD hold based on the data only. Thus, we can not label any FDs as H in this question.
2. You should not merge the two instances into one, since it's fine to have different representations, e.g., different scaler, in different instances of the same relation. Thus, as long as each instance does not violate the FD respectively, e.g., $B \rightarrow C$, we should label it as CD instead of NH.

]

5 Normalization (25 Points)

Consider the following relational schema for a chain store:

Sale(clerk, store, city, date, dish, size)
// a clerk sold a dish on a particular day at a given store in a city
Menu(dish, size, price)
// prices and available size for the dish

Make the following assumptions:

- Each clerk works in one store.
- Each store is in one city.
- The price of a dish is different for different sizes. The store has standardized prices: the same sized dish cannot be sold to two persons at two different prices.

1. Specify a set of completely nontrivial functional dependencies for relations Sale and Menu that encodes the assumptions described above and no additional assumptions.
   [**A**nswer: Sale: $clerk \rightarrow store, store \rightarrow city$; Menu: $dish, size \rightarrow price$ and $dish, price \rightarrow size$.
   Remark.With the clarification in bullet 6 in the clarification post, we only have one FD for Menu: $dish, size \rightarrow price$. Both solutions will be considered correct when grading.]

2. Based on your functional dependencies in part (1), specify all minimal keys for relations Sale and Menu.
   [**A**nswer: Sale: (clerk,date,dish,size), Menu: (dish,size) and (dish,price).
   Remark.With the clarification in bullet 6 in the clarification post, we only have one minimal key for Menu: (dish,size). Both solutions will be considered correct when grading.]

3. Are the schemas of Sale and Menu in Boyce-Codd Normal Form (BCNF) according to your answers to (1) and (2)? If not, give a decomposition into BCNF. If yes, justify your answer.
   [**A**nswer: No for Sale, Yes for Menu. BCNF decomposition of Sale: S1(clerk,store), S2(store,city), S3(clerk,date,dish,size). Menu is in BCNF, since the left side of $dish, size \rightarrow price$ is a superkey. (It's fine to add the following statement: the left side of $dish, price \rightarrow size$ is a superkey)]

4. Now add the following assumption:

   - Each city has at most one store and each store has only one clerk.

   Specify additional functional dependencies to take these new assumptions into account.
   [**A**nswer: Sale: $city \rightarrow store, store \rightarrow clerk$;]

5. Based on your functional dependencies for parts (1) and (4) together, specify all minimal keys for relation Sale.
   [**A**nswer: Sale: (clerk,date,dish,size), (store,date,dish,size), (city,date,dish,size).]

6. Are the schemas of Sale and Menu in 3NF according to your answers to (1), (4) and (5)? If not, give a decomposition into 3NF. If yes, justify your answer.
[**A**nswer: Yes. For Sale, we have FDs $clerk \rightarrow store, store \rightarrow city, city \rightarrow store, store \rightarrow clerk$, and the right hand side of each FD is part of the key. For Menu, we have FD $dish, size \rightarrow price$, and its left hand side is a superkey (It's fine to add the following statement: the left side of $dish, price \rightarrow size$ is a superkey).]