Homework 6: FDs and NFs
(due Nov 8th, 2018, 9:30am, hard-copy in-class please)

Reminders:
  a. Out of 100 points. Contains 5 pages.
  b. Rough time-estimates: ~8-9 hours.
  c. Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
  d. There could be more than one correct answer. We shall accept them all.
  e. Whenever you are making an assumption, please state it clearly.
  f. Important: Unless otherwise mentioned, assume that you need to show your work e.g. if the question asks ‘what are R’s keys?’ we do not just want a list; we want a step-by-step explanation as well.

Q1. FDs Definition [7 points]
(Modified version of Exercise 19.6 in your textbook) Suppose that we have the following three tuples in a legal instance of relation schema S with three attributes ABC:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>X</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>Y</td>
<td>a</td>
</tr>
</tbody>
</table>

Q1.1 (5 points) Which of the following dependencies can you infer does not hold over schema S?

(a) A → B,   (b) BC → A,   (c) B → C,   (d) B → A,   (e) C → AB

Q1.2 (2 points) Can you identify any functional dependencies that may hold over S?

Q2. Inferring FDs [27 points]
Consider the following relations and sets of functional dependencies that hold in those relations. For each sub-part, it is enough for you to list only completely non-trivial FDs with a single attribute on the right hand side.

Note that ‘candidate key’ means just ‘key’ (i.e. both words are interchangeable). A candidate key (or simply key) should imply the entire relation and should be minimal. On the other hand, a ‘superkey’ is any super-set of a candidate key.

Q2.1 R1(A, B, C, D) with FDs A → C, C → D, and D → B.
Q2.1.1 (3 points) What are all the non-trivial FDs that follow from the given FDs?

Q2.1.2 (3 points) What are all the keys (i.e. candidate keys) of R1?

Q2.1.3 (3 points) What are all the superkeys for R1?

Q2.2 R2(A, B, C, D) with FDs AD → C, C → A, and D → B.

Q2.2.1 (3 points) What are all the non-trivial FDs that follow from the given FDs?

Q2.2.2 (3 points) What are all the keys (i.e., candidate keys) of R2?

Q2.2.3 (3 points) What are all the superkeys for R2?

Q2.3 R3(A, B, C, D, E) with FDs AB → D, B → E, C → B, and DE → C.

Q2.3.1 (3 points) What are all the non-trivial FDs that follow from the given FDs?

Q2.3.2 (3 points) What are all the keys (i.e. candidate keys) of R3?

Q2.3.3 (3 points) What are all the superkeys for R3?

Q3. **Projection of FDs [26 points]**

Consider the relation \( R = \{A, B, C, D, E, F\} \) and the following set of FDs on \( R \): \( F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow F\} \)

Consider also the relations: R1 (A, B, C), R2 (A, B, C, E, F), and R3 (A, B, C, D).

Q3.1 (2x3=6 points) Compute the projection of the FD-set \( F \) on each of the three relations R1, R2 and R3. Only write down a minimal cover.

Q3.2 (2x3=6 points) What are all the candidate keys (i.e. the keys) for each of the three relations R1, R2 and R3?

Q3.3 (1x3=3 points) For each of the three relations, indicate if it is (A) BCNF and (B) 3NF. Explain your answer.

Q3.4 (6 points) Is it possible to decompose each R2 and R3 into new schemas such that they are in BCNF and the decomposition is lossless and dependency preserving? Show your work.
Q3.5 (5 points) Decompose R into multiple relations so that they are in 3NF and the decomposition is lossless and dependency preserving. Use the standard 3NF synthesis algorithm. Compare your result with Q3.4.

Q4. Creating dependencies [8 points]
A relation $R(A, B, C)$ satisfies the functional dependencies $A \rightarrow B$ and $A \rightarrow C$.

Q4.1 (4 points) State another functional dependency that $R$ must satisfy so these three functional dependencies (i.e., $A \rightarrow B$, $A \rightarrow C$ and the new functional dependency) allow us to infer that $R$ also $B \rightarrow C$. Your answer cannot contain $B$ or $C$ on the right hand side! Show your work.

For the next two parts of this question, assume that the new functional dependency you stated in Q4.1 also holds in $R$. Again, show your work for the next two parts as well.

Q4.2 (2 points) What are $R$'s keys?

Q4.3 (2x1=2 points) Is $R$ in (A) BCNF and (C) 3NF?

Q5. Lossy decompositions [8 points]
Consider a relation $R(A, B, C, D, E)$. Assume $R$ satisfies the following functional dependencies: $C \rightarrow E$ and $A \rightarrow C$.

Suppose we decompose $R$ into these two relations: $R1(A, B, C)$ and $R2(C, D, E)$. So you can think of $R1 = \Pi_{\{A,B,C\}}(r)$ and $R2 = \Pi_{\{C,D,E\}}(r)$.

Q5.1 (5 points) Show that this decomposition is not a lossless-join by giving an example. That is, give an instance $r$ of $R$ (i.e., give a set of tuples for $R$) such that both dependencies hold on $r$ but $r \neq R1 \bowtie R2$. Give the simplest instance you can find. Also show your work (i.e. compute $R1$ and $R2$ and the join).

Q5.2 (3 points) Can you show that the decomposition is lossy by using the theorem discussed in class? Again, show your work.

Q6. FDs in English language [24 points]
Consider the following table, which stores customer-order information.
CustomerOrderDB (CustomerNo, Name, Balance, CreditLimit, Ship-to-Address, OrderNo, OrderDate, DetailNo, ItemNo, QuantityOrdered, ItemExpireDate, DiscountRate, ItemPrice, ItemTax)

Now consider the following constraints in English:

Statement 1   Every customer has a unique customer number (CustomerNo). Other information for customer like Name, Balance and Credit Limit (CreditLimit) are kept.

Statement 2   Every customer can have several ship-to-addresses.

Statement 3   For every order we have HEADING information: order number (OrderNo), which is unique for every order. Order date (OrderDate) and Ship-to-address of customer.

Statement 4   Customers might request several items in one order. So, the order details that are detail number (DetailNo), item number (ItemNo) and quantity ordered (QuantityOrdered) should be recorded.

Statement 5   Every Item has Item number (ItemNo), Item Expire Date (ItemExpireDate), Discount Rate (DiscountRate), Item Price (ItemPrice) and Item Tax (ItemTax).

Statement 6   Discount Rate (DiscountRate) is computed based on the expiration date of Item (ItemExpireDate); the closer to the date of expiration; the higher discount will be applied.

Statement 7   Item tax (ItemTax) is defined using discount rate (DiscountRate) and item price (ItemPrice). Tax on discounted price is different from the tax on items without discount.

Clearly this is not a good relational design, as it contains redundancies and update, insertion and deletion anomalies. Hence we want to decompose it into a good schema.

Q6.1 (7 points) List the functional dependencies for this relation that you can construct using the English description and also specifically mention the corresponding statement number(s) you used to formulate each FD.

Note: some statement(s) might not result in a FD.

Hint: You should get 6 FDs.
Q6.2 (3 points) Rigorously prove (using Armstrong’s axioms) that you can derive the FD: ItemExpireDate ItemPrice → ItemTax from the set of the FDs you found in Q5.1. Show your steps.

Q6.3 (9 points) Is the above relation in BCNF using the FDs you found in Q6.1? If not, decompose it into a set of BCNF relations.

Q6.4 (4 points) Is your resulting decomposition in Q6.3 dependency-preserving? If it is not dependency-preserving, explain your answer, and then decompose the original relation into a set of 3NF relations instead. Show your steps.