Homework 3: Relational Database Design Theory
(100 points)

Due Date: Friday, Feb 3 (5:00 PM)

Submission

All HW assignments should contain both your student ID and your name and must be submitted online, as a SQL script text file (not PDF this time!), via the associated dropbox on EEE. See the table below for the HW submission opportunities. Note that after 5 PM on Sunday no further HW submissions will be accepted. (We will be releasing the solution at that time.) Please strive to get all your work in on time! If possible, try to save the one dropped assignment for the end of the term when you are most likely to want/need it.

<table>
<thead>
<tr>
<th>Date / Time</th>
<th>Grade Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friday, Feb 3 (5:00 PM)</td>
<td>Full credit will be available</td>
</tr>
<tr>
<td>Saturday, Feb 4 (5:00 PM)</td>
<td>20 points will be deducted</td>
</tr>
<tr>
<td>Sunday, Feb 5 (5:00 PM)</td>
<td>40 points will be deducted</td>
</tr>
</tbody>
</table>

Relational Design Theory [100 pts]

While creating the DDL for the TopicalBirds.com relations, you realized that there are a few more features that you really should have implemented. For example, it would be nice to keep information about your users’ mobile devices. However, as a co-founder of the company, you are also partly responsible for helping to build the actual business and to raise more venture capital, so it’s hard for you to spend much more time on grunt work. Thus, you have hired a team of UCI undergraduate students as interns to add additional features to your relational database design. Unfortunately, your startup budget is limited, so you can’t afford CS122a alumni - instead, your interns are hackers who’ve never taken an actual database course. If nothing else, though, they’re quick - they have already completed their design for the new features. They’ve just handed you the script attached to the end of this document to serve as the documentation for their design. The script shows their new tables as well as providing a snippet of example data for each one. Analyze each of the new tables using your newly gained knowledge of relational database design theory. That is, based on combining the initial HW1 information with the schemas for these new tables and
the comments that their script includes, your job here is to carefully review their relational design(s) by figuring out what the resulting functional dependencies and keys are and normalizing their new tables if necessary. (Each new table can be analyzed independently of the others.)

1. BirdDevice [25pts]
(a) [5pts] List all of the functional dependencies involving the attributes of the relation in question. (Just give the basic list, not the closure of that list. 😊)

Answer 1:
(btag, phonenum) -> btag, phonenum, network, serialno, make, model
(btag, make, serialno) -> btag, phonenum, network, serialno, make, model
(make, serialno) -> phonenum, network, serialno, make, model
phonenum -> phonenum, network, serialno, make, model

(In this universe, a device has attributes phonenum, network, serialno, make, and model, and either phonenum or (make, seriano) can be used to identify the device.)

--or--

Answer 2:
(btag, phonenum) -> btag, phonenum, network, serialno, make, model
(btag, make, serialno) -> btag, phonenum, network, serialno, make, model
(make, serialno) -> model
phonenum -> network

(In this alternate universe, one phone number can be associated with multiple devices, meaning that dialing one number will ring through to multiple devices.)

(b) [12pts] Identify which normal form their relation is in by answering the following questions:

b-1> [4pts] Does their relation satisfy 1NF [Yes/No]? Show your reasoning.
Yes, as all attributes are atomic

b-2> [4pts] Does their relation satisfy 2NF [Yes/No]? Show your reasoning.
No, because both universes above have FDs that are partial dependencies (where btag isn’t part of their left-hand side).

b-3> [4pts] Does their relation satisfy 3NF [Yes/No]? Show your reasoning.
No, since it wasn’t even 2NF.
(c) [8pts] Normalize their relation, putting it in either BCNF or into 3NF if BCNF is not possible. Say which normal form your final answer is in, and briefly explain why that’s the case.

**Answer 1’s BCNF normalization:**
DeviceInfo(phonenum, network, serialno, make, model)
OwnerInfo(btag, make, serialno)
--- or ---
DeviceInfo(phonenum, network, serialno, make, model)
OwnerInfo(btag, phonenum)

**Answer 2’s BCNF normalization:**
DeviceInfo(serialno, make, model)
OwnerInfo(btag, phonenum, make, serialno)
NetworkInfo(phonenum, network)
2. AdShown [25pts]
(a) [5pts] List all of the functional dependencies involving the attributes of the relation in question. (Just give the basic list, not the closure of that list.)

(aid, btag, shown_at) -> aid, wtag, wbname, btag, shown_at
aid -> wtag           -- based on info from HW #1 (recall that Watchers own Ads, 1:N)
wtag -> wbname        -- also based on info from HW #1

(b) [12pts] Identify which normal form their relation is in by answering the following questions:

b-1> [4pts] Does their relation satisfy 1NF [Yes/No]? Show your reasoning.
Yes, as all attributes are atomic.

b-2> [4pts] Does their relation satisfy 2NF [Yes/No]? Show your reasoning.
No. It's not in 2NF because aid -> wtag is a partial dependency (aid is not a candidate key).

b-3> [4pts] Does their relation satisfy 3NF [Yes/No]? Show your reasoning.
No, since it wasn’t even in 2NF.

(c) [8pts] Normalize their relation, putting it in either BCNF or into 3NF if BCNF is not possible. Say which normal form your final answer is in, and briefly explain why that's the case.
R1(aid, btag, shown_at)
R2(aid, wtag)
R3(wtag, wbname)
This is BCNF since all FDs’ left-hand sides are candidate keys.
(R2 and R3 could actually be dropped, as their info exists elsewhere, i.e., in HW #2’s solution.)
3. Address [25pts]
(a) [5pts] List all of the functional dependencies involving the attributes of the relation in question. (Just give the basic list, not the closure of that list. 😊)

loc_id -> bldg_number, street_name, city, state, country, mailcode
(street, state) -> city

(b) [12pts] Identify which normal form their relation is in by answering the following questions:
   b-1> [4pts] Does their relation satisfy 1NF [Yes/No]? Show your reasoning.
   Yes, as all attributes are atomic.

   b-2> [4pts] Does their relation satisfy 2NF [Yes/No]? Show your reasoning.
   Yes, as there are no partial dependencies.

   b-3> [4pts] Does their relation satisfy 3NF [Yes/No]? Show your reasoning.
   No, as there is a transitive dependency from loc_id through to city.

(c) [8pts] Normalize their relation, putting it in either BCNF or into 3NF if BCNF is not possible. Say which normal form your final answer is in, and briefly explain why that's the case.
R1(loc_id, bldg_number, street_name, state, country, mailcode)
R2(street, state, city)
This is BCNF, as all FD left-hand sides are now candidate keys.
4. **BirdCall [25pts]**

(a) [5pts] List all of the functional dependencies involving the attributes of the relation in question. (Just give the basic list, not the closure of that list. 😊)

\[(\text{phone1, phone2, start\_time}) \rightarrow \text{phone1, phone2, start\_time, duration}\]

(b) [12pts] Identify which normal form their relation is in by answering the following questions:

- b-1> [4pts] Does their relation satisfy 1NF [Yes/No]? Show your reasoning.
  
  Yes, as all attributes are atomic.

- b-2> [4pts] Does their relation satisfy 2NF [Yes/No]? Show your reasoning.
  
  Yes, as there are no partial dependencies.

- b-3> [4pts] Does their relation satisfy 3NF [Yes/No]? Show your reasoning.
  
  Yes, as there are no transitive dependencies.

(c) [8pts] Normalize their relation, putting it in either BCNF or into 3NF if BCNF is not possible. Say which normal form your final answer is in, and briefly explain why that’s the case.

It’s BCNF already, as the only dependency has the candidate key as its LHS.