Fall, 2007 CIS 550

Database and Information Systems

Solutions to Homework 3

Due on October 22, 2007

Problem 1 [15 points]: Consider a relation R with four attributes $ABCD$. You are given the following dependencies: $A \to B$, $BC \to D$, $C \to AB$.

1. List all keys for R. (other than superkeys)
   C.
2. Is R in 3NF? Why?
   No, because in $A \to B$, A is not a superkey, and B is not part of a key.
3. Is R in BCNF? Why?
   No, because it is not 3NF.

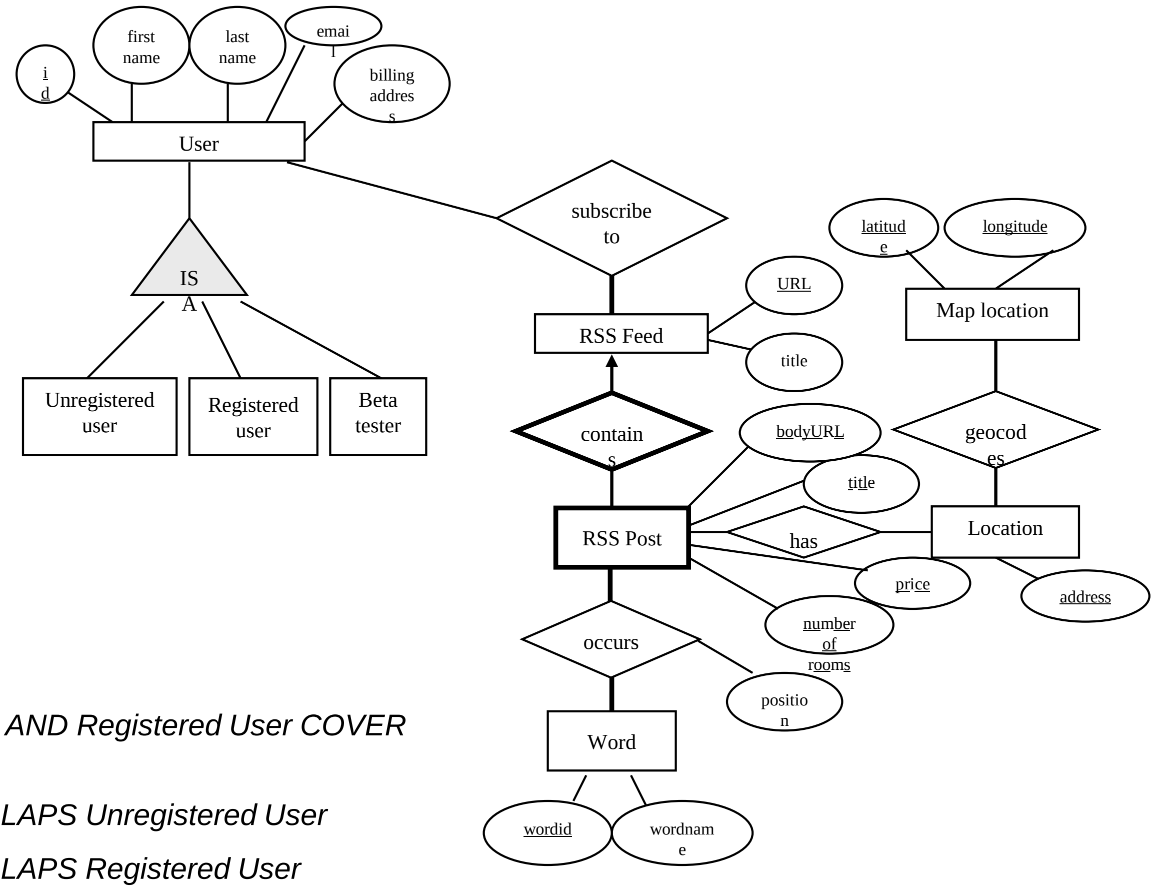
Problem 2 [30 points]: The task is to design an apartment search mash-up system. This system will integrate maps with apartment rental listings. A user will create an account, subscribe to the apartment RSS feeds, search apartments based on keywords, and view the apartments on the maps.

- Each user will have a unique numeric ID. Additionally, the first and last names, email address, and billing address will need to be stored.
- Each user is either an unregistered user or a registered user.
- Original users of the system (some still unregistered users and others registered) are further classified as beta testers.
- Each user may subscribe to one or more apartment RSS feeds. One RSS feed may be shared among multiple users.
- RSS feeds have URLs and titles.
- RSS feeds have multiple RSS posts.
- Each RSS post has a title, location, price, the number of rooms and a URL.
- Each RSS post consists of a set of word occurrences and their positions. An inverted index on words is needed to support keyword search.

- The map can translate a location to several possible geocode locations(namely, latitude and longitude).

Draw an ER diagram for the apartment search mash-up system. The ER diagram should include various attributes, keys, participation constraints, overlap and covering constraints.

Here is one of many possible ER diagrams:



Unregistered User AND Registered User COVER User

Beta Tester OVERLAPS Unregistered User

Beta Tester OVERLAPS Registered User

Problem 3 [25 points]: Consider a relation R with six attributes $ABCDYZ$ and the FD set $F = \{AB \rightarrow Y, AC \rightarrow D, Y \rightarrow C, ZB \rightarrow D, BD \rightarrow Z\}$. Let F^+ denote the closure set of F.

1. For each of the following attribute sets, do the following: (i) write down a minimal cover of the subset of F^+ that holds over the set; (ii) name the strongest normal form that is not violated by the relation containing these attributes; (iii) decompose it into a collection of BCNF relations if it is not already in BCNF.

   (a) $ABDYZ$
   (i) $\{AB \rightarrow Y, AB \rightarrow D, AB \rightarrow Z, ZB \rightarrow D, BD \rightarrow Z\}$ *(ii) 2NF*
   (iii) $R1(ABY), R2(ABD), R3(ABZ), R4(BDZ)$

   (b) $ABCD$
   (i) $\{AB \rightarrow C, AC \rightarrow D\}$ *(ii) 2NF (iii)* $R1(ABC), R2(ACD)$

2. For each of the following decompositions of $R = ABCDYZ$, with the same set of functional dependencies F, say whether the decomposition is (i) dependency preserving, and (ii) lossless join.

   (a) $\{ABYD, ABCZ\}$
   Lossless join, because the key is AB, so $AB \rightarrow ABYD, AB \rightarrow ABCZ$.
   Not dependency preserving, because $Y \rightarrow C$ *is not preserved.*

   (b) $\{ACD, ABYZ, ABDZ\}$
   Not lossless join, because neither $A \rightarrow ACD$, *nor* $A \rightarrow ABYZ$ *can be inferred in* F^+.
   Not dependency preserving, because $Y \rightarrow C$ *is not preserved.*

Problem 4 [20 points]: Suppose you are given a relation $R(A, B, C, D, E)$. For each of the following (complete) sets of FDs, (i) identify the candidate key(s) for R, and (ii) state whether or not the proposed decomposition of R into smaller relations is a "good" decomposition and briefly explain why or why not.

1. $A \rightarrow B$, $B \rightarrow CE$, $C \rightarrow D$. Decompose into AB, BCE, and CD.
   (i) A (ii) good, dependency preserving and lossless join

2. $C \rightarrow A$, $B \rightarrow D$. Decompose into ACE and BD.
   (i) BC (ii) not good, dependency preserving only, but not lossless join

Problem 5 [15 points]: Why do commercial DBMSs support keys and foreign keys, but not general FDs?

Because it is expensive to validate general FDs. A normal key (or foreign key) constraint can be checked directly using an index, whereas an arbitrary FD cannot. In general, if we convert a schema to BCNF or 3NF, we will have most or perhaps even all of the FDs encoded as keys – meaning that support for additional FDs is not particularly important.