

CS4604 Midterm

October 12, 2006

Please enter the following information:

Name:

9 digit ID:

Please do not write below this line.

Question	Maximum Score	Score
1	5	
2	5	
3	5	
4	10	
5	15	
6	20	
7	40	
Total	100	

1. (5 points) Circle **True** or **False**. A many-many relationship R connects entity sets E and F . In an E/R diagram, we can use a rounded arrow directed towards F to indicate that R has referential integrity from E to F .
2. (5 points) Circle **True** or **False**. R is a relation in 4NF. If the MD $A \twoheadrightarrow BC$ holds in R and is non-trivial, then the FD $A \rightarrow BC$ holds in R .
3. (5 points) Circle *all* the possibilities that correctly complete the sentence: “The closure of a set of attributes is a set of _____.”
 - (a) functional dependencies
 - (b) attributes
 - (c) keys
 - (d) FDs
 - (e) multi-valued dependencies

4. (10 points) Draw an E/R diagram to model project groups in CS4604. Keep in mind that each enrolled student (identified by a PID) can work at most one project. Each project, identified uniquely by its name, can have at most two groups working on it. Be sure to identify all the appropriate multiplicity and referential integrity constraints in the diagram. Indicate key attributes in each entity set. As always, use “Notes:” to identify aspects of the problem that you cannot model.

5. (15 points) A key is *simple* if it consists of only one attribute. Prove that if a relation R is in 3NF and if every key in R is simple, then R is in BCNF. Your proof should be general, e.g., it should not assume that R has a fixed number of, say two or three, attributes.

6. (20 points) Use an E/R diagram to model the following simplified scenario describing house sales. The sale of a house involves many parties. A seller owns the house. A buyer wants to buy the house. They agree on a price for the house. Each party engages a realtor. (Realtors may themselves buy houses.) The buyer finances the purchases of the house using a mortgage from a bank. A particular buyer-seller pair may be involved in more than one transaction. For each transaction, each realtor gets a commission that is a percentage of the purchase price of the house. Indicate appropriate multiplicity constraints. For each relationship, write a brief sentence explaining why you think it should have the indicated multiplicity (you can use space on the next page). Minimise the number of entity sets you use. However, you should not decrease the number of entity sets at the cost of introducing attributes that will not have meaningful values for all entities. For instance, you cannot model this scenario using just one entity set with attributes for every aspect of this scenario. You can use the next page too for your answer.

7. (40 points) Excited by what you are learning in CS4604, you decide to create a database to track the songs your favourite band plays in its live concerts. Since you decide that E/R diagrams are for kids, you decide to create a relation schema directly for your database. After much consideration, you believe that a single schema will serve:

Concerts(City, Venue, Year, Month, Date, Song, Album).

In this relation, **City** (e.g., “Blacksburg”) and **Venue** (e.g., “Cassell Colisseum”) record where the concert took place and **Year**, **Month**, and **Date** keep track of when the concert took place. The idea is that these five attributes uniquely specify a concert. The attribute **Song** records the name of a song performed at a concert. You add the attribute **Album** to record which album the song belongs to. Perfect!

However, after using the database for a few months, you realise that your band (and the real world) have some characteristics that you should model in your database. Convert each of the next four sentences about **Concerts** into a functional or a multi-valued dependency. You can use the first letter of each attribute as an abbreviation for the attribute. Consider each of these four sentences independently. If you cannot write down a functional or a multi-valued dependency, say so, and explain why you cannot, if possible. Do not assume any other constraints, even if they seem reasonable to you.

- (a) (3 points) Each song appears in at most one album. In other words, the band does not repeat the same song in different albums.

- (b) (3 points) A city does not have two venues with the same name. In other words, **City** and **Venue** serve to identify the location of a concert uniquely.

- (c) (3 points) In an effort to please its fans, the band plays at most one song from any album in a given concert.

- (d) (3 points) The manager books the band in any city at most once every year.

Problem continues on the next page.

For the next two parts of this question, assume that all the functional and/or multi-valued dependencies you specified in the previous parts hold in **Concerts**, as do any dependencies that follow from them. However, no other dependencies hold in **Concerts**.

(e) (6 points) What are the keys for **Concerts**?

(f) (10 points) You realise that you must decompose **Concerts** into multiple relations. Here is a candidate decomposition into two relations:

Concerts1(City, Venue, Year, Month, Date)

Concerts2(City, Year, Song, Album)

(i) (6 points) For each relation **Concerts1** and **Concerts2**, state what normal forms it satisfies.

(ii) (4 points) For each of the four FDs from the previous page, specify which relation (**Concerts1**, **Concerts2**, both, or neither) you can use to verify the FD.

7a

7b

7c

7d

Problem continues on the next page.

(g) (4 points) After a few years of using relations **Concerts1** and **Concerts2** and faithfully recording the band's performances in it, you realise that your beloved band plays a mean trick on its audience. In each city, all its concerts (spread over different years, of course) feature exactly the same songs! For example, the songs the band played in Blacksburg in 1996 are identical to the songs the band played in Blacksburg in 2000 and in 2004. Write down the dependencies that model this discovery and the names of the relations (**Concerts1** and/or **Concerts2**) in which these dependencies hold.

(h) (4 points) Decompose the relations affected in the previous part into 4NF.

SCRATCH SPACE

SCRATCH SPACE

SCRATCH SPACE