

Write your name on this paper and write your answers on the answer sheets provided. You may refer to any SQL document that you have created.

1. (4 pts each) Give an example (and explain it) of a table that meets the specified requirements. Be sure to name the field(s) that compose(s) the primary key. Do *not* use the table given problem 3.
  - (a) 1NF but not 2NF
  - (b) 2NF but not BCNF
  
2. (3 pts each) Briefly define/explain the following terms:
  - (a) JDBC
  - (b) functional dependency
  - (c) foreign key
  
3. Suppose we want to track contributions made by individuals to various political candidates. The field **CID** is the database id number for a give contributor while **PID** is the database id number for a given political candidate. **Party** represents the political affiliation of the candidate. The **Contributions** field contains a list of contributions to that candidate along with the date of contributions.

<u>CID</u>	ContribName	<u>PID</u>	Politician	Party	Contributions
12489	Sam Smith	23	Hillary Clinton	Democratic	\$50, 6/1/2006; \$150, 9/16/2006; \$2,000, 1/26/2007
12489	Sam Smith	27	Barack Obama	Democratic	\$1,000, 2/12/2007
11921	Mary Jones	18	Chris Christie	Republican	\$3,000, 3/5/2007
13823	Henry James	18	Chris Christie	Republican	\$250, 12/12/2006; \$500, 2/29/2007
13823	Henry James	27	Barack Obama	Democratic	\$250, 12/12/2006; \$500, 2/29/2007

- (a) (4 pts) List the functional dependencies that exist in this table.
- (b) (6 pts) Name as many reasons as you can think of that this table is not in 4NF.
- (c) (2 pts) Based on your answer to question 3b, how would you classify the form of this table (unnormalized, 1NF, 2NF, or BCNF)?
- (d) (8 pts) Convert the table to 4NF. For each resulting table identify the primary key field(s). Show the table structure only (i.e., do not list the data in the table for your answer).
- (e) (4 pts) Represent the revised design as an ER diagram. Be sure to specify the cardinality of the relationships.

4. Suppose a database exists to store information about books and authors and that it has been designed as follows:

- author=(author\_id, firstname, lastname)
- publisher=(pub\_id, name, city)
- book=(isbn, author\_id, title, year, pub\_id)

(a) Show the SQL statements necessary to create these tables so that the following integrity rules will be enforced.

- (2 pts) the author's last name is required, but the first name is optional; publisher name and city are required; title is required but year is not
- (3 pts) every book must have a valid author specified; if an author is removed from the list then any books associated with them should be removed as well
- (2 pts) include an SQL statement that will create an entry in the **publisher** table with an id number of 1 and a name of "Unknown" and a city of "n/a"
- (2 pts) if a book's publisher is not known, the publisher id number should default to the value 1
- (4 pts) if a publisher is deleted from the list, any books so assigned should revert to "Unknown"; if a publisher's id number changes then then entries in the **book** table should be changed to match automatically
- (2 pts) require that each publisher's name be unique
- (4 pts) when a new book is being entered make sure that the year is not in the future. HINT: The expression `date_part('year', current_date)` will return the current year (in four digit format). If the year is from the future then deny entry.
- (3 pts) specify a primary key for each table

(b) (4 pts) Write a series of SQL statements that will insert a new book in the database. Make these assumptions:

- The book is entitled "Super Fun" and is authored by "Mary Jones".
- "Mary Jones" does not yet exist in the **author** table.
- The **author** table type for the **author\_id** field is **SERIAL**.
- There are no other authors in the database who have the name "Mary Jones".
- The book is published by an existing publisher whose id is 13.

**EXAM CONTINUES ON PAGE 3 . . .**

5. Suppose the following tables represent statistics for baseball players for various years.

player			stats			
<u>id</u>	lastname	firstname	<u>id</u>	year	rbi	battingavg
19	Smith	Fred	19	2012	57	0.312
19	Smith	Fred	19	2013	60	0.320
20	Jones	Sam	19	2014	47	0.297
21	Gonzalez	Eric	20	2012	null	null
22	Yakamoto	Tom	20	2013	20	0.250
23	Ruth	Babe	20	2014	22	0.270
24	Matthis	Alan	21	2012	null	null
			21	2013	null	null

- (a) (2 pts each) What would be displayed by each of the following queries:
- SELECT SUM(rbi) FROM stats WHERE id=20
  - SELECT COUNT(rbi) FROM stats WHERE id=20
  - SELECT AVG(rbi) FROM stats WHERE id=20
- (b) (3 pts each) Construct a query to accomplish each task below. Do not assume that the data listed in the tables above is complete.
- For each year display the number of null values that exist for batting average.
  - For each player give the average of their available batting averages (whether or not they appear in the **stats** table).
  - Delete all rows in the **stats** table having a null **rbi** value.
6. (16 pts) Write a complete Java method that accepts as a parameter a **Connection** object that has already been initialized and has established an active connection with the database shown in this problem. Your method should prompt the user to enter **rbi** data for all entries in the **stats** table that have a value of **NULL** for that statistic. The amounts entered by the user should be saved in the database. Your solution should work on any data set (not just the exact example shown above). When prompting the user be sure to tell them the player's name for which you are collecting information. Any exceptions should be caught and be accompanied by a programmer-provided error message. Assume that the tables in question are in a PostgreSQL schema named **baseballdb**.