CSCI 4320
Spring 2025
Exam 1

Last Name: _____

First Name: _____

When taking this exam you may use a single 3×5 card containing SQL syntax. The card must be written by you. Write your name on this paper and write your answers on the answer sheets provided.

1. (4 pts) What does SQL stand for and why is it important to know?
2. (2 pts each) Expand each acronym:
   (a) DBMS
   (b) DBA
   (c) ER (as in ER diagram)
3. (4 pts) What is the difference between a database and a DBMS?
4. (3 pts) There are three levels of data abstraction relavent to the study of database systems: the view level, the logical level, and the physical level. Explain any one of the three.
5. (3 pts each) Briefly define each of the following terms:
   (a) relation
   (b) foreign key
   (c) UML
6. (6 pts) Name three reasons why a DBMS might be preferred over using programmer-defined data files for storing information.
7. (4 pts) Give an example (and explain it) of a table that is in 1NF but not in BCNF. Be sure to name the field(s) that compose(s) the primary key. Do *not* use the table given problem 8.
8. 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.

CID	ContribName	PID	Politician	Party	Contributions
12489	Sam Smith	23	Joe Biden	Democratic	$50, 6/1/2023; $150, 9/16/2023; $2,000, 1/26/2024
12489	Sam Smith	27	Kamala Harris	Democratic	$1,000, 2/12/2024
11921	Mary Jones	18	Donald Trump	Republican	$3,000, 3/5/2024
13823	Henry James	18	Donald Trump	Republican	$250, 12/12/2023; $500, 2/29/2024
13823	Henry James	27	Kamala Harris	Democratic	$250, 12/12/2023; $500, 2/29/2024

(a) (4 pts) List the functional dependencies that exist in this table.

(b) (4 pts) Name as many reasons as you can think of that this table is not in BCNF.

(c) (2 pts) Based on your answer to question 8b, how would you classify the form of this table (unnormalized or 1NF)?

(d) (8 pts) Convert the table to BCNF. 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). Write the new design using list notation.

9. Consider the following three relations/tables that represent the efforts of a restaurant to keep track of various dinner parties and their orders.

   - dinner_party=(party_id,name,number)
   - menu=(menu_id,description)
   - orders=(menu_id,party_id,quantity)

   Assume that matching field names represent related fields among the tables.

Table: dinner_party

party_id	name	number
1	Johnson	4
2	Linebarger	2
3	Garcia	3
4	O'Toole	8
5	Hill	4
6	Anderson	5

Table: menu

menu_id	description
1	Fried Rice
2	Chicken Pot Pie
3	Beef KaBobs
4	Lamb Chops
5	ToFu

Table: orders

menu_id	party_id	quantity
1	1	1
2	1	1
4	1	2
1	2	1
3	2	1
1	3	2
2	3	2

Table: orders (continued)

menu_id	party_id	quantity
3	4	4
1	4	2
2	4	1
3	6	2
1	6	2
4	6	1

(a) (4 pts) Draw an ER-diagram representing these tables and their relationships. Be sure to specify the cardinality of the relationships based on the structure of the tables and what you know of the problem domain.

(b) (4 pts) Represent the tables graphically again, this time using UML notation.

(c) (3 pts each) Given the structure and contents of the tables above, show the output that would be produced by each of these queries. For queries that produce more than 20 rows I will accept intelligent use of dots.

i. `SELECT name, description FROM dinner_party INNER JOIN orders USING(party_id) INNER JOIN menu USING (menu_id)`

ii. `SELECT name, description FROM dinner_party LEFT OUTER JOIN orders ON (dinner_party.party_id=orders.party_id) NATURAL RIGHT OUTER JOIN menu`

(d) (3 pts each) Construct SQL statements to achieve the following results (do *not* assume that the contents of the tables exactly match the samples given above):

i. List all parties with 4 or more members.

ii. List all parties who have not placed an order.

iii. List all menu items along with the number of times they've been ordered. Sort the results showing the most often ordered items first. If an item has not been ordered then it should not show up in the list.

iv. Repeat the query of problem 9(d)iii except include in the list items that have not been ordered.

v. List all orders (including name of ordering party, quantity ordered, and description of menu item ordered) of all parties having "son" in the name.

vi. Create a view named `all_orders` that lists all the items ordered together with the dinner party that ordered them. The view should display all fields from all three tables. The results should be ordered first by party name and then by the quantity ordered with most frequently ordered listed first.

vii. Use the view you create in problem 9(d)vi to produce a list of parties who have ordered, but have ordered fewer items than members in their dinner party. (NOTE: In the instance above, O'Toole is an example of this).

viii. Remove all parties that do not have any items ordered.

ix. Change all parties that have "Smith" anywhere in the name to 10 people in their party.