**CSC 430 – Database Management Systems**
Exam 2

Name: _____

Instructions:

- Put your name in the appropriate place at the top of this page;
- Do not use red ink;
- **Closed books and notes;**
- **No electronic devices are allowed; This includes calculators.**
- You will only receive points for a question if you attempt to answer it;
- For full credit, list all formulas that provide the basis for calculations and show all work;
- If you aren't clear about a question, state your assumptions first followed by your answer;
- When finished with the exam read and sign the pledge at the bottom of this page.

Good luck!

**"On my honor as a Louisiana Tech student, I have neither given nor received unauthorized assistance on this academic work."**

_____

Student signature

**Section A: Multiple-choice questions.** Total: 10 points.

Please, circle a single correct option. Each question is worth 2 points.

1. Select correct statement(s):
    a. Data Definition Language (DDL) includes commands used to define and modify database schema.
    b. Data Manipulation Language (DML) includes commands used to retrieve and manipulate data in a database.
    c. Data Definition Language (DDL) includes commands used to retrieve and manipulate data in a database.
    d. Data Manipulation Language (DML) includes commands used to define and modify database schema.
    e. Only a, b.
    f. Only b, c.
    g. Only c, d.
    h. Only a, c.

2. Select informal relational database design guideline(s):
    a. Making sure semantics of the attributes is clear in the schema.
    b. Reducing the redundant information in tuples.
    c. Reducing the NULL values in tuples.
    d. Disallowing possibility of generating spurious tuples.
    e. All of the above.
    f. Only a, b.
    g. Only b, c.
    h. Only c, d.

3. Select correct statement(s):
    a. Functional dependency is a constraint between two sets of data attributes derived from semantic of these attributes.
    b. Functional dependency can only be inferred from the state of the database.
    c. Functional dependencies are used in the relation normalization process.
    d. All of the above.
    e. Only a, b.
    f. Only b, c.
    g. Only a, c.

4. Select goal(s) of normalization process:
    a. Minimizing redundancy.
    b. Maximizing amount of candidate keys.
    c. Minimizing data manipulation anomalies.
    d. Minimizing the number of tables.
    e. All of the above.
    f. Only a, c.
    g. Only b, c.
    h. Only a, d.

5. Select correct statement(s):
    a. Indexes are auxiliary access structures used to speed up the retrieval of the records.
    b. Clustering index is used when data file is ordered by a non-key attribute.
    c. Primary indexes are used when data file is ordered by primary key attributes.
    d. Secondary index is defined over non-ordering attributes of a record.
    e. All of the above.
    f. None of the above.
    g. Only a, b, c.
    h. Only a, b, d.

**Section B: Open-ended questions.** Total: 25 points.
To get full points provide complete answer, be specific and concise.

6. **(5 pts)** Explain one unique purpose of triggers for a database?

7. **(5 pts)** Explain one purpose of views for a database?

8. **(5 pts)** Clearly cross out any parts of this table that are incorrect.

| Triggers | Views | Stored Procedures | Functions |
|---|---|---|---|
| Invoked when you CALL them | Invoked when you use them in FROM clause | Invoked when you CALL them | Invoked when you use them in FROM clause |
| Can execute any SQL commands | Can execute any SQL commands | Can only SELECT | Can SELECT and calculate |
| Can stop an operation from occurring | | Can take parameters | Can take parameters |
| Can produce one result (as a returned value) | | Can produce many results (as OUT parameter(s)) | Can produce one result (as a returned value) |

9. **(5 pts)** What is the difference between "delete", "drop", and "truncate" in SQL? How exactly are they used and how do they work?

10. **(5 pts)** State the condition(s) for each normal form to be satisfied.

    First normal form:

    Second normal form:

    Third normal form:

**Section C: Practical questions.** Total: 65 points.

To get full points show all work, provide all formulas and calculations.

11. **(8 pts)** Write a query to retrieve first name, last name and address of all employees who work for the 'Research' department. For full points - use aliasing when joining tables and rename attributes of the resulting relation to "First_Name", "Last_Name", and "Employee_Address". Database state is provided for you on a separate page.

12. **(8 pts)** Write a query to retrieve project number, project name, and the number of employees for each project, ordering results by project number. Database state is provided for you on a separate page.

13. **(10 pts)** Write a function to get the count of employees that have a salary greater than the given minSalary parameter. You must give correct characteristics to the function for full credit.

14. **(10 pts)** Write a trigger that would append the words "in_progress" to the pname attribute value of any newly created project.

15. **(5 pts)** Define which of the provided functional dependencies may hold for the following relation. If the dependency cannot hold – justify your answer, by specifying at least one tuple that causes the violation.

| Instructor | Course | Text | Quarter |
|---|---|---|---|
| Smith | Data Structures | Hoffman | Fall |
| Smith | Systems Programming | Martin | Fall |
| Hall | Programming Languages | Bartam | Winter |
| Brown | Databases | Elmasri | Spring |
| Brown | Data Mining | Williams | Spring |
| Hall | Data Structures | Hoffman | Winter |
| Johnson | Databases | Elmasri | Summer |

A. Text -> Course

B. Text -> Instructor

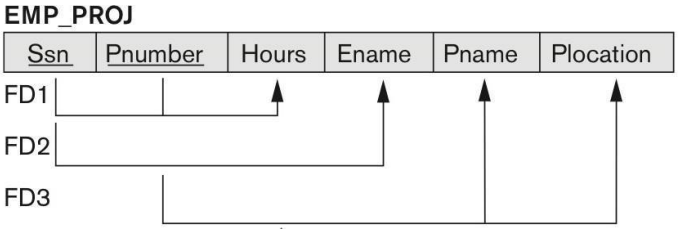
C. Quarter -> Instructor

D. Course -> Quarter

E. Course -> Text

F. Instructor -> {Text, Quarter}

16. **(5 pts)** What is the highest normal form of the following relation? To support your answer, specify which functional dependencies violate which of the normal forms.

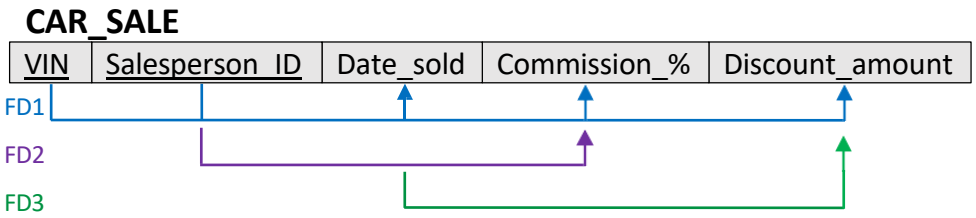



FD1: {SSN, Pnumber} -> Hours

FD2: Ssn -> Ename

FD3: Pnumber -> {Pname, Plocation}

17. **(5 pts)** Define which of the described functional dependencies (FD1, FD2, and FD3) violate second normal form (2NF) and which violate third normal form (3NF). Briefly explain how they violate it.




Primary key of this relation is a combination of {VIN, Salesperson_ID}, assuming that the car can be sold by multiple salespeople.

FD1: {VIN, Salesperson_ID} -> {Date_sold, Commission_%, Discount_amount)

Primary key (VIN, Salesperson_ID) functionally defines every non-prime attribute.

FD2: Salesperson_ID -> Commission_%

The commission percentage is based on the specific salesperson.

FD3: Date_sold -> Discount_amount

The discount amount is determined by the date the car was sold (e.g. seasonal discounts).

18. **(14 pts)** Consider a disk with a block size B = 1024 bytes. Suppose file has r = 32,000 DEPARTMENT records of fixed length. Each record has following attributes: Dname (36 bytes), Dnumber (8 bytes), Mgr_ssn (9 bytes), Mgr_start_date (11 bytes). In addition, suppose the file is sorted on Dnumber primary key attribute. Assuming an unspanned organization, calculate the record size **R**, the size of file ordering key **V**, blocking factor **bfr**, and the number of file blocks **b.**

    Now calculate the number of block accesses required to search for a record with a given primary key value. We are assuming we do NOT have an index.

    Show all formulas (with variable names), all calculations (with values plugged in), and give final numeric answers for each (i.e. don't leave any final answer as a math equation).

**Bonus:** 10 points total. Answer all parts correctly for full bonus points (partial credit is given).

Part A

Given the following code:

```
$query = "SELECT lname, salary FROM employee WHERE lname = '$lname' AND ssn = '$ssn'";
```

where $query is the query we are about to run, $lname is the last name that the user entered into the form and $ssn is the social security number that the user entered into the form. If this query were to run after we give it user data, what would the user need to give it to be able to use SQL injection to get the salary of all employees?

last name: _____

ssn: _____

You should give the text the user should enter, where they should enter it at, and explain why each part of that text is important (i.e. what does it specifically do).

Part B

If this $query were to run as a multiquery (i.e. can contain multiple queries separated by semi colons), is that more dangerous? How can that be exploited further to do dangerous things?

Part C

What is one recommended way we can protect ourselves against these exploits?

**EMPLOYEE**

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|---|---|---|---|---|---|---|---|---|---|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | NULL | 1 |

**DEPARTMENT**

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|---|---|---|---|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

**DEPT_LOCATIONS**

| Dnumber | Dlocation |
|---|---|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| 5 | Houston |

**WORKS_ON**

| Essn | Pno | Hours |
|---|---|---|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

**PROJECT**

| Pname | Pnumber | Plocation | Dnum |
|---|---|---|---|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

**DEPENDENT**

| Essn | Dependent_name | Sex | Bdate | Relationship |
|---|---|---|---|---|
| 333445555 | Alice | F | 1986-04-05 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |