COSC3380: Exam
SQL, relational model, algebra, normalization, ER

Name (Last name, first):

Test score:

Instructions

Give short answers. Write your name on every page. Write an answer in the space provided. The exam will be scanned for grading, write your answers in the back page only if absolutely necessary.

Questions

1. Consider a relation (set of tuples) $T(\underline{K_1, K_2}, A, B, C)$ with composite PK=$\{$K1,K2$\}$ shown below. Does it violate 2NF? If N explain show FDs justifying violation. If Y show queries in relational algebra notation that compute the 2 projections with lossless join decomposition, verifying it. Partial credit: write the queries in SQL.

K1	K2	A	B	C
1	1	1	2	0
1	2	1	1	1
2	1	1	3	1
2	2	1	3	0
3	1	1	1	0
4	1	1	1	0

2. Consider a table T(A,B,C) with simple PK A (one column, valid) and two non-key columns stored on a DBMS with SQL. State if table T is in 1NF, 2NF, 3NF, BCNF: Y/N/maybe

3. Consider an input table in SQL that is very large (millions of rows) a simple PK and many foreign keys (e.g. transaction sales). Which SPJ relational algebra operator can produce duplicate rows during query evaluation in SQL SELECT? Which algorithm or data structure can be used to obtain a set (eliminate duplicates)?

4. List at least 3 reasons why UML notation is preferable to classical ER notation. Explain why classical ER notation had no issues back in the 70s.

5. Remember enforcing set semantics is optional in SQL, which means SQL is not 100% compliant with theory. Consider a plain SQL table (without any specified primary key in the DDL) T(A,B,C). Write a SELECT query that verifies if A can work as a primary key and a SELECT query that can figure out if T is a set or not.

6. Consider two tables T_1 and T_2, where K is a foreign key in T_1, K is a valid primary key in T_2, but the referential integrity constraint is not enforced. Write comparison expressions $(>, <, =)$ between $|T_1 \bowtie T_2|$ with $|T_1|$ and with $|T_2|$ verifying referential intergrity Y/N. If there is not sufficient information to make an assertion you can state "undefined".

7. Consider these tables for a supermarket with sales information:
   customer(custid,name,telephone)
   product(productid,name,price,departmentid)
   department(departmentid,name,aisle)
   salestransaction(tid,custid,transdate)
   salesitem(tid,productid,qty)

   write an SQL query that gives total sales amount, and number of items bought by customers buying from 'Bread' department whose name starts with 'D';

   write an SQL query computing total sales amount by department.

8. Explain reasons why SQL could not initially beat COBOL (and other old languages) in old enterprise databases. State the two main SQL programming mechanisms to read and write tables (i.e. unique programming in SQL compared to other languages like Python, Java or C++). Explain why Python is now competing with SQL for data-oriented tasks.

9. Consider two entities $E_1(K_1, A)$ and $E_2(K_2, B)$ with respective simple PKs K_1, K_2 connected by a binary relationship. Explain how you map an M:N (many to many) relationship to a relational table in a new entity E_3. Can E_3 have attributes? Y/N, why? Explain how tables in SQL are defined based on such mapping (i.e. write the DDL statements).

10. Consider a table $T(\underline{K}, A, B, C)$ with K being its PK, as follows:

```
K A B C
1 x 2 0
2 y 3 1
3 y 1 0
4 x 2 0
5 x 2 1
6 x 4 1
```

    Write a functional dependency (FD) violating 3NF. Write a false functional dependency. Explain why it does not hold. Write a transitive FD.

HARDER: up to 2 questions like these may be included.

11. Explain why 1NF does not make sense in SQL, but it may impact 4NF. Give an example of 4NF on aa table with 4 attributes.

12. Duplicates. Write Y/N. SQL queries can return duplicate rows. SQL queries can produce duplicate rows in intermediate results. SPJ queries in relational algebra can return duplicates. SPJ queries in relational algebra can produce duplicate rows in intermediate results. FK columns can have duplicates.

13. List the 3 levels of database design. Explain which tasks are involved at each level. Is the DB model design method top-down, bottom-up or hybrid? Explain physical aspects in terms of SQL DDLs. Explain at which level is best to apply OOD for app development.

14. Give an airline database example of a 3-ary relationship, write the relationship in plain noun/verb sentences, show its ER diagram in UML notation. Show the ER mapping to physical model (SQL tables).

15. Is it feasible to reach BCNF preserving all FDs? why? Explain why simple PKs make 3NF tables be in BCNF

16. Justify why pointers are bad idea for database design and manipulation. Explain differences between a pointer (reference) and a foreign key. Explain what is a cascading deletion.

17. FDs help detecting redundant patterns in a DB, to maintain it in a correct state. Recall Armstrong axioms. Explain the meaning of complete and sound when applying these axioms. What is F^+?

18. Explain the recursive relationship manages on the Employee table. Show it in an ER diagram in UML and its translation into R.A. or SQL query.

19. Consider these tables for a supermarket with sales information:
    customer(custid,name,telephone)
    product(productid,name,price,departmentid)
    department(departmentid,name,aisle)
    salestransaction(tid,custid)
    salesitem(tid,productid)

    Explain DDL denormalization changes violating 3NF to optimize this query. Why is denormalization needed? How is denormalization decided? Give guidelines.

20. Practice SQL with typical DBs

    For the airline database. Write SQL queries answering these questions:
    passengers that have never flown from city 'X'
    airplanes that have arrived late
    flights that had an increase in sales from one month to the next month
    show top 3 destination cities, which had highest number of arriving passengers in a specific month.

    For the supplier/part database: show suppliers who can supply every part
    parts that have never been ordered
    manufacturers of a specific part, not manufactured by any other supplier
    show the most popular part based on number of parts ordered