Assignment 3 (CMPT 354): SQL Programming and Normalization
Due: Nov 25 (Sat), 11:59pm
Total mark: 94
Weight=24%

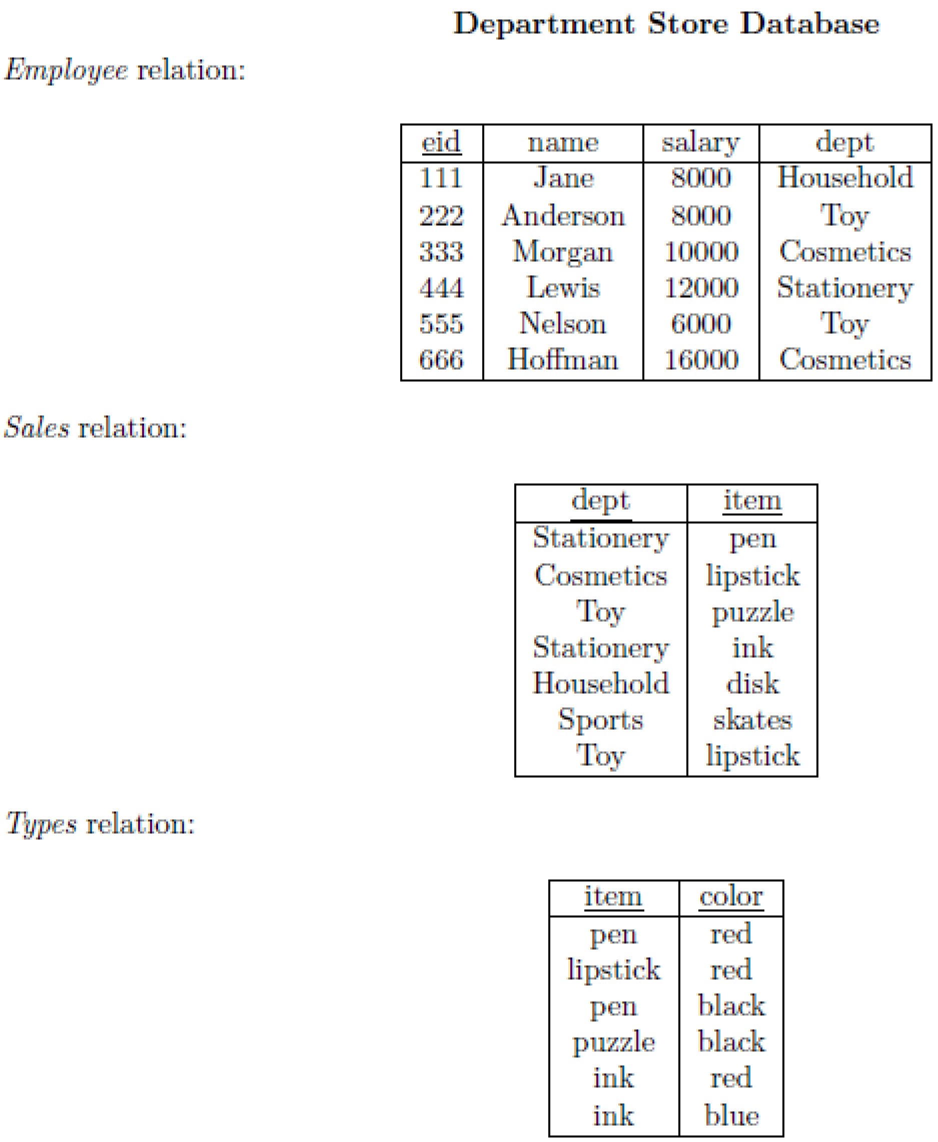
Submission instruction: (1) This assignment must be done by each student independently. (2) Submission is through coursys.sfu.ca in a single pdf file with maximum file size of 10MB. Late submission will not be accepted as answers will be posted immediately after deadline. (3) The student is responsible for submitting the assignment successfully before the deadline, and only the submission that is in the system before the deadline will be accepted.

This assignment has two parts: Part 1 on SQL programming, and Part 2 on functional dependencies and normalization. The detail of Part 2 will be added later. Since Part 1 involves the setup of DMBS servers and running SQL, you should start working on this part as early as possible.

The note to TA: The deadline for finishing marking is Dec 4, 2023

Part 1: SQL Programming (46 marks)

This assignment will use the following "Department-Store Database". The underlined fields are the attributes of primary keys.



Department Store Database

Employee relation:

eid	name	salary	dept
111	Jane	8000	Household
222	Anderson	8000	Toy
333	Morgan	10000	Cosmetics
444	Lewis	12000	Stationery
555	Nelson	6000	Toy
666	Hoffman	16000	Cosmetics

Sales relation:

dept	item
Stationery	pen
Cosmetics	lipstick
Toy	puzzle
Stationery	ink
Household	disk
Sports	skates
Toy	lipstick

Types relation:

item	color
pen	red
lipstick	red
pen	black
puzzle	black
ink	red
ink	blue

DBMS Environments. All SQL statements in Part 1 will be run on a DBMS server. The choices of DBMS servers are SQLite or MYSQL. Alternatively, you can also use the Microsoft SQL Server 2019 on CYPRESS.csil.sfu.ca - the CSIL SQL Server. A SQL database on Microsoft SQL Server has been set up for each of you (but you need to create the tables and enter data into the tables). To use Microsoft SQL Server, you can log into CSIL computers that already have SQL Server Management Studio installed, and access CSIL SQL Server by following the email instruction sent to you at the beginning of this semester from helpdesk@cs.sfu.ca.

There are a lot of online materials on setting up MySQL and SQLite on your computers. Here are a few but you can find more online:

https://www.kdnuggets.com/2022/09/free-sql-database-course.html (Installation on Windows OS Computers, by Matthew Mayo) https://dev.mysql.com/doc/refman/8.0/en/ (MySQL 8.0 Reference Manual)

SQLite: https://www.sqlite.org/index.html, https://www.sqlitetutorial.net/

For each task below, write the SQL statement for finding the answer requested by the task. For submission purpose, you need to specify the DBMS environment used (i.e., MySQL, SQLite, Microsoft SQL Server) and include the screenshot of the SQL statements and the answer returned by running the SQL statements.

Task 1 (5 marks): Create the above database schema using CREATE TABLE statements, including primary key constraints, and the constraint that salary is integer in the range [5000,20000]. You can assume CHAR (20) type for all other attributes.

```
CREATE TABLE Employee (
eid CHAR (20),
name CHAR (20),
salary INTEGER,
dept CHAR (20),
PRIMARY KEY (eid),
CHECK (salary >= 5000 AND salary <= 20000))

CREATE TABLE Sales (
dept CHAR (20),
item CHAR (20),
PRIMARY KEY (dept,item))

CREATE TABLE Types (
item CHAR (20),
color CHAR (20),
PRIMARY KEY (item,color))
```

Task 2 (5 marks): Insert the above records into the tables using INSERT statements.

```
INSERT INTO Employee (eid, name, salary, dept) VALUES (111, Jane, 8000, Household);
INSERT INTO Employee (eid, name, salary, dept) VALUES (222, Anderson, 8000, Toy);
INSERT INTO Employee (eid, name, salary, dept) VALUES (333, Morgan, 10000, Cosmetics);
INSERT INTO Employee (eid, name, salary, dept) VALUES (444, Lewis, 12000, Stationery);
INSERT INTO Employee (eid, name, salary, dept) VALUES (555, Nelson, 6000, Toy);
INSERT INTO Employee (eid, name, salary, dept) VALUES (666, Hoffman, 16000, Cosmetics);

INSERT INTO Sales (dept, item) VALUES (Stationery, pen);
INSERT INTO Sales (dept, item) VALUES (Cosmetics, lipstick);
INSERT INTO Sales (dept, item) VALUES (Toy, puzzle);
INSERT INTO Sales (dept, item) VALUES (Stationery, ink);
INSERT INTO Sales (dept, item) VALUES (Household, disk);
INSERT INTO Sales (dept, item) VALUES (Sports, states);
INSERT INTO Sales (dept, item) VALUES (Toy, lipstick);

INSERT INTO Types (item, color) VALUES (pen, red);
INSERT INTO Types (item, color) VALUES (lipstick, red);
INSERT INTO Types (item, color) VALUES (pen, black);
INSERT INTO Types (item, color) VALUES (puzzle, black);
INSERT INTO Types (item, color) VALUES (ink, red);
INSERT INTO Types (item, color) VALUES (ink, blue);
```

Task 3 (42 marks, 6 marks each): Compute the answers to the following queries using SELECT statements. Your SQL statements should be correct for ALL instances of data, not just for the above instance. For example, to find the departments that have a larger average salary than that of ”Stationery” department, we do not accept the SQL that uses 12000 as the average salary of ”Stationery” department because it only works for the above instance.

1. Compute the maximum salary for each department that sells at least two distinct items.

```
SELECT dept, MAX(salary)
FROM Employee
GROUP BY dept
HAVING dept IN
          (SELECT S.dept
          FROM Sales S
          GROUP BY S.dept
         HAVING COUNT(*) >= 2)
```

Answer: (Stationary, 12000) and (Toy, 8000).

2. Compute the names of the employees who work in a department that sells some item in black color

```
SELECT E.name
FROM Employee E, Sales S, Types T
WHERE E.dept=S.dept AND S.item = T.item AND T.color="black"
```

Answer: Lewis, Anderson, and Nelson

3. For each department that has a larger average salary than that of "Stationery" department, find its average salary.

```
SELECT E.dept, AVG(E.Salary)
FROM Employee E
GROUP BY E.dept
HAVING AvG(E.salary) >
```

```
          (SELECT AVG(E2.salary)
          FROM Employee E2
          WHERE E2.dept="Stationery")
```

Answer: (Cosmetics, 13K)

4. Find the number of the departments that have a smaller average salary than that of "Stationery" department.

```
SELECT COUNT(DISTINCT E.dept)
FROM Employee E
WHERE E.dept IN
           (SELECT E1.dept
           FROM Employee E1
           GROUP BY E1.dept
           HAVING AvG(E1.salary) <
                   (SELECT AVG(E2.salary)
                   ROM Employee E2
                   WHERE E2.dept="Stationery"))
```

Answer: 2 (i.e., Household and Toy).

5. Which department pays every of its employees at least 7000?

```
SELECT dept
FROM Employee
GROUP BY dept
HAVING MIN(salary) >=  7000
```

Answer: Household, Cosmetics, Stationery.

6. Which departments sell all items sold by “Cosmetics” department

```
SELECT S.dept
FROM Sales S
WHERE NOT EXISTS
           (SELECT item
           FROM Sales
          WHERE dept="Cosmetics"
          EXCEPT
          SELECT item
          FROM Sales
          WHERE dept=S.dept)
```

Answer: Toy

Part 2: FD and Normalization (48 marks)

Question 1 (10 marks, 5 marks for correct instances and 5 marks for discussion). Consider a relation R=(S, A, C, D, T), representing that the student (S) has the address (A), takes the course (C) from the teacher (T) who is from the dept (D). Assume the following FDs F hold on R:

C -> T: a course determines its teacher, i.e., each course is taught by only one teacher
S -> A: a student determines its address, i.e., each student has only one address.
T -> D: a teacher determines its department, i.e., each teacher is from only one department.

However, a student can take multiple courses, a teacher can teach multiple courses, and each department can have multiple teachers. Use an instance of R to explain data redundancy, update anomaly, insertion anomaly, and deletion anomaly that may exist for the above schema R and FDs.

Answer:

S	A	C	D	T
Joe	1th Ave	database	Computing Sci.	wangk
Joe	1th Ave	Operating Sys.	Math	Tom
Jane	Third Ave	database	Computing Sci.	wangk
Jane	Third Ave	Statistics	Computing Sci	wangk

1. Data redundancy: (S,A) is repeated for each course taken by the same S because S -> A (red). (D,T) is repeated for each course taught by the same T because T -> D (green). (C,T) is repeated for each student taking the same C because C -> T.
2. Update anomaly: if Joe changes the current address, we have to change all the repeated occurrences of the address for Joe; if wangk changes the current dept, we have to changes all the repeated occurrences of the department for wangk; if the database course changes its teacher, we have to change all the repeated occurrences of the teacher for database.
3. Insertion anomaly: We cannot insert a student and its address if the student has not taken any course, similarly, we cannot record the department of a teacher who has not taught any course.
4. Deletion anomaly: If we delete the only record for Tom (say Tom no longer teaches Math), we lose the Tom's department information.

Question 2 (38 marks) Continue with the R and FDs F in Question 1.

1. (3 marks) Find all keys of R with respect to F.
2. (3 marks) Test if R in BCNF with respect to F, why?

3. (10 marks) Produce a BCNF decomposition through a series of binary decomposition. For each binary decomposition, tell the FD used for the decomposition and show the FDs holding on the decomposed tables.
4. (3 marks) Explain why the decomposed tables produced in 3 is a better representation than the original single table R.
5. (3 marks) Is the final decomposition in 3 dependency-preserving, why
6. (3 marks) Is the original schema R in 3NF with respect to F, why
7. (10 marks0 If the answer to 6 is no, produce a 3NF decomposition that is lossless and dependency-preserving.
8. (3 marks) Is the decomposition produced in 7 in BCNF?

   Answer

1. {C,S} is a key because the closure {C,S}^+ = {C,S,T,A,D} contain all attributes, and C or S alone can determine all attributes. {C,S} is the only key because they cannot be added to the closure by any FD.
2. From 1, all three FDs are violations of BCNF because none of C, S, T is a superkey in R.
3. We can choose any of three FDs to decompose R, and different choices lead to a different decomposition.

Choose C->T to decompose R:

C^+ = {C,T,D}, Decompose R into R1=CTD and R2=(SACDT-CTD) union C = SAC.

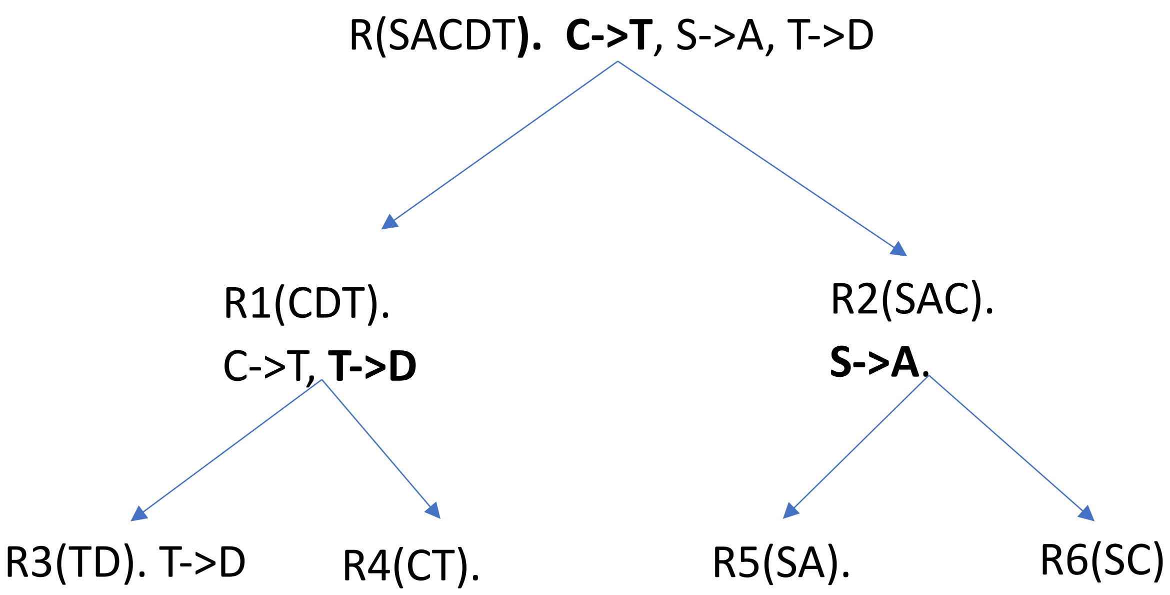
R1 is not in BNCF: {C->T, T->D} holds on R1 and T is not a superkey on R1 (i.e., does not determine C). We decompose R1 into R3=TD and R4=CT.

R3 is in BCNF: T->D holds on R3 and T is a superkey on R3. Similarly, R4 is BCNF.

R2 is not BCNF: {S->A} holds on R2 and S is not a superkey on R2. Decompose R2 into R5=SA and R6=SC.

R5 is BCNF: S->A holds on R5 and S is a superkey on R5. Similarly, R6 is BCNF (because no FD holds on R6, thus, no violation of BCNF).

The above decomposition is summarized in the tree below. Boldface indicates the violation used for decomposition at each step. If there are more than one violation at each step, the choice of each violation will lead to a different tree.



R(SACDT**). C->T**, S->A, T->D

R1(CDT).
C->T, **T->D**

R2(SAC).
S->A.

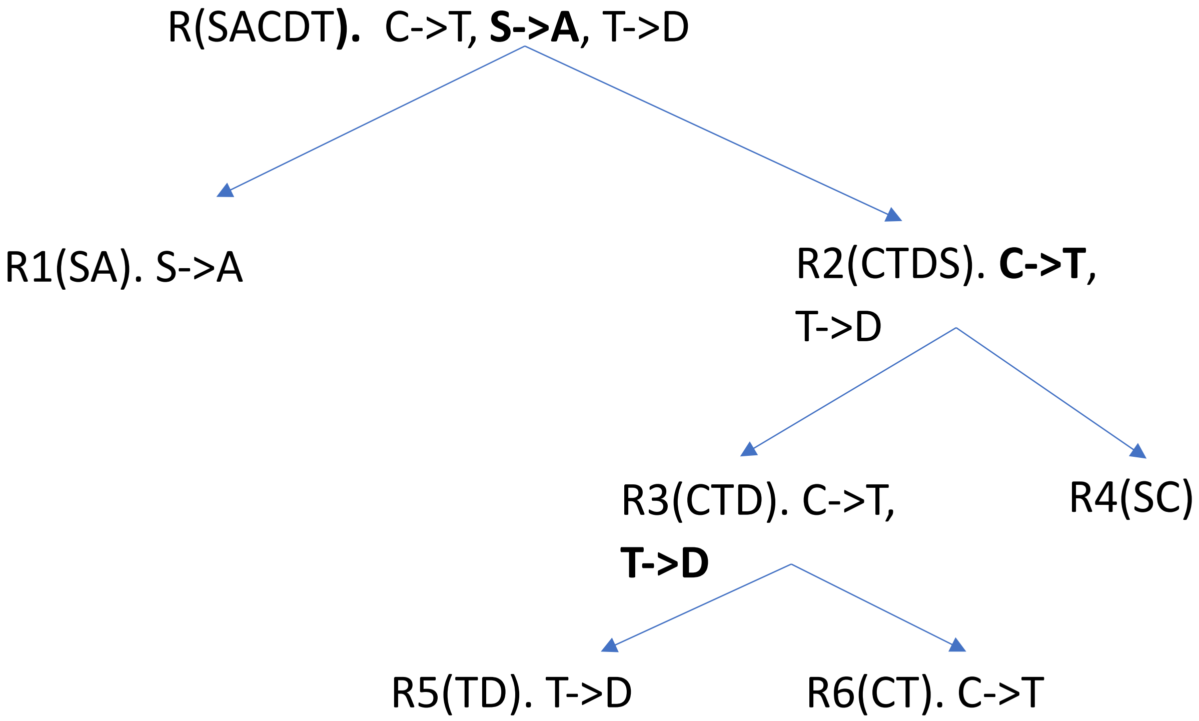
R3(TD). T->D

R4(CT).

R5(SA).

R6(SC)

Choose S->T to decompose R:



R(SACDT**).** C->T, **S->A**, T->D

R1(SA). S->A

R2(CTDS). **C->T**,
T->D

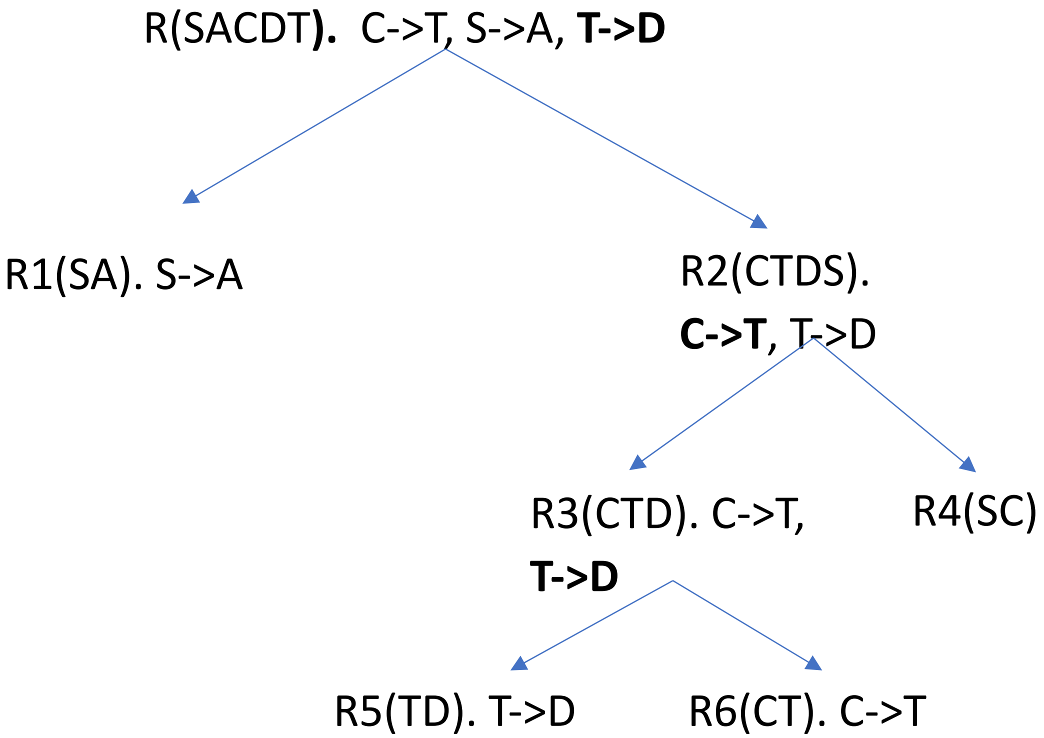
R3(CTD). C->T,
T->D

R4(SC)

R5(TD). T->D

R6(CT). C->T

Choose T->D to decompose R:



4. For the decomposition in the first tree, R3(TD) stores the information about each teacher (i.e., their departments, etc), R4(CT) stores the information about the (possibly multiple) courses taught by teachers, R5(SA) stores the information about students (i.e., their addresses, etc), R6(SC) stores the information about the (possibly multiple) courses taken by students. The redundancy of teachers' and students' information is avoided by storing their information once in the separate tables R3 and R5, independently of the number of courses taught by each teacher or the courses taken by each student. Consequently, the update anomaly, insertion anomaly and deletion anomaly are also avoided.
5. For the decomposition in the first tree, dependency is preserved because each of T->D, C->T, and S->A is contained in R3, R4, R5, respectively.
6. R is not 3NF. From the answer to 1, CS is the *only* key in R, which means that A,D,T are non-prime attributes (i.e., not contained in any key). So the three FDs in F={C->T, S->A, T-> D} are violations of 3NF, i.e., left-hand sides are not superkeys and the right-hand sides are prime attributes.
7. We apply the 3NF synthesis to produce a 3NF decomposition. The first step is to find a minimal cover of F. Since F is already a minimal cover, we can map each FD in F to a table: R1=CT, R2=SA, R3=TD. This step ensures that each of R1,R2,R3 is in 3NF with respect to its own FDs and the decomposition {R1,R2,R3} is dep-preserving the FDs in F. For lossless decomposition, since the

key CS of R is not contained in any of R1,R2, and R3, we need to add the key CS as an additional table R4. The final decomposition is R1=CT, R2=SA, R3=TD, R4=CS, which satisfies all properties: 3NF, dep-preserving, and lossless decomposition.

8. The decomposition produced in 7, i.e., R1=CT, R2=SA, R3=TD, R4=CS, is also in BCNF.