Roll Number:

CS4.301: Data and Applications (Monsoon 2024)

Quiz 4

Maximum Marks: 20, Time: 35 minutes

- **Keep answers concise. State all assumptions.**
- **All questions are compulsory.**

Question 1

Theoretically, outer join can be computed by executing a combination of relational algebra operators. Show how the following left outer join operation can be achieved through a sequence of relational operations.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

SELECT E.Lname, E.Fname, D.Dname FROM (EMPLOYEE E LEFT OUTER JOIN DEPARTMENT D ON E.Dno = D.Dnumber);

(4 marks)

1 mark for each step in the answer.

If any other sequence of operations is followed, 4 if everything is correct, else 0.

1. Compute the (inner) JOIN of the EMPLOYEE and DEPARTMENT tables.

   $TEMP1 \leftarrow \pi_{Lname,\ Fname,\ Dname}\ (EMPLOYEE \bowtie_{Dno=Dnumber} DEPARTMENT)$

2. Find the EMPLOYEE tuples that do not appear in the (inner) JOIN result.

   $TEMP2 \leftarrow \pi_{Lname,\ Fname}\ (EMPLOYEE) - \pi_{Lname,\ Fname}\ (TEMP1)$

   This minus operation can be achieved by performing an anti-join on Lname, Fname between EMPLOYEE and TEMP1, as we discussed above in Section 18.5.2.

3. Pad each tuple in TEMP2 with a NULL Dname field.

   $TEMP2 \leftarrow TEMP2 \times \mathbf{NULL}$

4. Apply the UNION operation to TEMP1, TEMP2 to produce the LEFT OUTER JOIN result.

   $RESULT \leftarrow TEMP1 \cup TEMP2$

Question 2

Convert the below relation into 1NF and 2NF forms. Describe what problems 1NF and 2NF forms avoid that the original database was susceptible to.

Employee ID	**Phone Number**	**Dept**	**Manager ID**
E01	(9938901015, 9394565700)	Sales	E13
E02	(8096756525)	Development	E17
E03	(7896969420, 8175798520, 9786484547)	Research	E22
E04	(9119110808)	Sales	E13

(3+2 marks)

You are allowed to follow any of the representations of normal forms listed in Elmasri. Depending on which form you follow you will end up with different results:

Case 1:

1NF schema: Table1(Employee_ID, Phone Number, Dept, Manager ID) <- here we create multiple rows to accommodate the multiple values of phone number while making phone number a key attribute.

2NF schema: Table1(Employee_ID, Phone Number) Table2(Employee_ID, Dept, Manager ID) <- since Dept and Manager ID are dependent only on Employee_ID (as is evident from the unnormalized form) they need to be moved to a new table in 2NF.

Case 2:

1NF schema: Table1(Employee_ID, Phone Number 1, Phone Number 2, Phone Number 3, Dept, Manager ID)

2NF schema: no change

Case 3:

1NF schema: Table1(Employee_ID, Phone Number) Table2(Employee_ID, Dept, Manager ID) <- create new table to accommodate multivalued attribute

2NF schema: no change

1.5 marks for 1NF 1.5 marks for 2NF 1+1=2 marks for describing problems with 1NF and 2NF -0.5 if keys are not underlined properly (per table)

Question 3

Suppose a relational schema R(A, B, C, D, E, F, G, H, I) and the set of functional dependencies F: { AB → C, AD → GH, BD → EF, A → I, H → J}.

Is the relation in 3NF? If not decompose it to 3NF.

(4 marks)

Relation is R(A,B,C,D,E,F,G,H,I,J) – J was added during the quiz.

Step 1 – Find Candidate Keys (1 mark)

FDs: AB->C, AD->GH, BD->EF, A->I, H->J

ABD is the candidate key as the closure of ABD includes all attributes in R.

Step 2 – Verify 3NF for each FD (1 mark)

For each X -> Y, is X a superkey?

Are all attributes in Y prime attributes (part of candidate key)?

For all FDs, X is not a superkey, and Y isn't prime attribute. **Therefore, reln is not in 3NF.**

Step 3 – Decomposition (2 marks)

AB->C gives **R1(A,B,C)**

Remaining attributes are (A,B,D,E,F,G,H,I,J)

AD->GH gives **R2(A,D,G,H)**

Remaining attributes are (A,B,D,E,F,I,J)

BD->EF gives **R3(B,D,E,F)**

Remaining attributes are (A,B,D,I.J)

A->I and H->J give **R4(A,I)** and **R5(H,J)**

Remaining attributes are (A,B,D) – Therefore, **R6(A,B,D)**

1 mark for R6, and 1 mark for (R1…R5)

Question 4

Consider the following relations and dependencies. For each part, determine the candidate keys, and decompose the relation into a collection of BCNF relations. Note that the relations must be normalized to 1NF, 2NF, 3NF, and then BCNF. If a normal form is satisfied, mention it.

(a) R1(A,C,B,D,E), A → B, C→ D
(a) R2(A,B,F), AB→ F, B→ F

(7 marks)

Part (a)

Candidate key is {A,C,E}. – 0.5 marks for identifying key

1NF is satisfied. – 0.5 mark (explanation mandatory, else 0)

2NF is not satisfied as B is partially dependent on A and D is partially dependent on C.

Decompose into R1_1(A,B), R1_2(C,D), R1_3(A,C,E)

0.5 marks for identifying 2NF is violated and 1 mark for decomposition

3NF and BCNF are satisfied. – 1 mark (0.5 each with explanation)

Part (b)

Candidate key is {A,B} -- 0.5 marks for identifying key

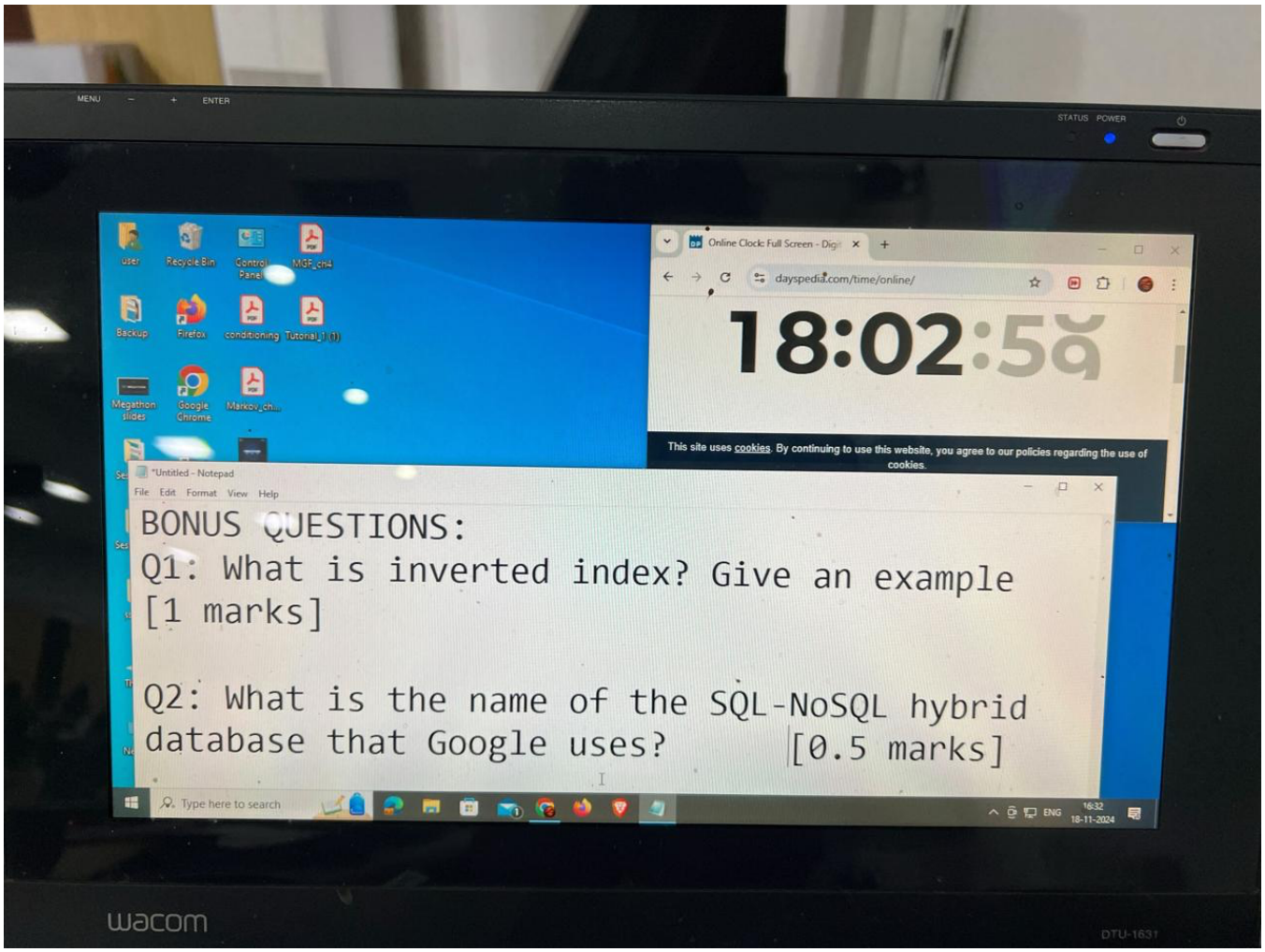
1NF is satisfied. -- 0.5 mark (explanation mandatory, else 0)

B->F is a partial dependency of F on B, therefore 2NF is not satisfied.

Decompose into R2_1(B,F) and R2_2(A,B)

0.5 marks for identifying 2NF is violated and 1 mark for decomposition

3NF and BCNF are satisfied. -- 1 mark (0.5 each with explanation)



1 mark if inverted index is described through example

0.5 mark if F1 database is mentioned

Max marks cannot exceed 20.