Fundamentals of Database Systems

Assignment: 3

Due Date: 23st August, 2017

Instructions

This question paper contains 15 questions in 6 pages.

Q1: Consider the following relation and its functional dependencies, where the primary key is underlined.

r(A,B, C, D, E)
AB → CDE
A → C
B → D

Normalize the above relation to BCNF.

A. r1(A, C); r2(B, D); r3(A,B,C, D, E)

B. r1(A, C); r2(B, D); r3(A,B,C, E)

C. r1(A, C); r2(B, D); r3(A,B,D, E)

D. r1(A, C); r2(B, D); r3(A,B,E)

Explanation: A determines C uniquely and B determines E uniquely, while A and B jointly uniquely identify E. Normalizing to 2NF results in all the relations to be normalized to BCNF.

Q2: Normalize the following relation in 2NF to 3NF.

t(cust_id, name, salesperson, region)
cust_id → name
cust_id → salesperson
cust_id → region
salesperson → region

A. t1(cust_id, name, salesperson); t2(salesperson, region)

B. t1(cust_id, name, salesperson); t2(region)

C. t1(cust_id, name); t2(salesperson, region)

D. The relation is not in 2NF.

Explanation: No non-prime attribute depends on subset of candidate key, hence the relation is in 2NF. salesperson → region creates transitive FD, separately storing the mapping of salesperson to region removes this dependency, hence A is correct.

Q3: State if the following statement is *True* or *False*.

It is possible that a loss-less decomposition does not exhibit dependency preservation.

A. True

B. False

Explanation: A loss-less decomposition may not exhibit dependency preservation.

Q4: Normalize the following relation to BCNF.

CI(clientNo, interviewDate, interviewTime, staffNo, roomNo)
clientNo, interviewDate → interviewTime, staffNo, roomNo
stafftNo, interviewDate, interviewTime → clientNo, roomNo
stafftNo, interviewDate → roomNo

A. I(clientNo, interviewDate, interviewTime, roomNo)
S(stafftNo, interviewDate, roomNo)

B. I(clientNo, interviewDate, interviewTime, staffNo)
S(interviewDate, roomNo)

C. I(clientNo, interviewDate, interviewTime, staffNo)
S(stafftNo, interviewDate, roomNo)

D. I(clientNo, interviewDate, interviewTime, staffNo)
S(stafftNo, roomNo)

Explanation: The last functional dependency violates BCNF rule as determinant is not a super-key. To remove this dependency, decompose the relation and set up a relation for the non-key determinant with attributes functionally dependent on it.

Q5: A relation r(A, B, C) had multivalued dependencies among all its attributes. It was decomposed for achieving a better database as follows:

r1(A, B)
r2(B, C)
r3(C, A)

This is an example of what kind of normalization?

A. BCNF

B. 4NF

C. PJNF

D. DKNF

Explanation: The relation exhibits join dependency, and is normalized to 5NF.

Q6: Consider the following statements:
(I) We can check if a functional dependency holds by examining a single relation instance.
(II) We can check if a functional dependency is violated by examining a single relation instance.

A. Only statement I is true.

B. Only statement II is true.

C. Both I and II are true.

D. Neither I nor II is true.

Explanation: It cannot be checked if an FD is a part of the scheme by examining a single instance because in future there can be tuple(s) violating the FD whereas if in schema we found a tuple which makes the FD wrong, the FD is violated.

Q7: Consider a relation $R = (A, B, C, D, E)$ with multivalued dependencies:
$\{ A \twoheadrightarrow B, B \twoheadrightarrow D \}$
Suppose R contains the tuples $(0, 1, 2, 3, 4)$ and $(0, 5, 6, 7, 8)$. Which of the following tuples *must* also be in R?

A. $(0, 1, 6, 7, 4)$

B. $(0, 5, 6, 3, 8)$

C. $(0, 1, 2, 7, 8)$

D. $(0, 1, 6, 3, 4)$

Explanation: First consider the two given tuples. Try to apply the given MVDs to any pair of tuples that have the same values for the attributes on the left side of the MVD. Each application of an MVD lets you add two tuples to the relation: the tuples formed by swapping the values for the attributes on the right side of the MVD. This can be done using the formal definition of an MVD. Repeat this process until all of the tuples implied by the MVDs are already in the relation.

Q8: Let relation $R(A, B, C, D, E, F, G, H)$ satisfy the following functional dependencies:
$\{ A \rightarrow B, CH \rightarrow A, B \rightarrow E, BD \rightarrow C, EG \rightarrow H, DE \rightarrow F \}$
Which of the following FDs is also guaranteed to be satisfied in R?

A. $BFG \rightarrow AE$

B. $CEG \rightarrow AB$

C. $CGH \rightarrow BF$

D. $ADE \rightarrow CH$

Explanation: To check whether a functional dependency is satisfied by R you need to take its left-hand side, calculate the closure with respect to the original set of the functional dependencies and then check whether the right-hand side is contained within the closure. So in the question only $CEG \rightarrow AB$ is satisfied because closure of CEG = {C, E, G, A, B, H} and AB is in the closure.

Q9: Consider a relation $R = (A, B, C, D)$. For which of the following sets of FDs is R in BCNF?

A. $\{ AC \rightarrow D, D \rightarrow A, D \rightarrow B, D \rightarrow C \}$

B. $\{ ABC \rightarrow D, ACD \rightarrow B, D \rightarrow C, BCD \rightarrow A \}$

C. $\{ AD \rightarrow C, BC \rightarrow A, BD \rightarrow C, CD \rightarrow B \}$

D. $\{ AC \rightarrow B, A \rightarrow D, C \rightarrow A, D \rightarrow B \}$

Explanation: A relation is in BCNF if for every nontrivial FD, the left-side attributes contain a key. To test whether a set of attributes S contains a key, compute the closure of the attributes in S using all of the FDs. If the closure is all attributes of the relation, then the attributes contain a key; otherwise not. You need to go through this process for each of the given FD set to determine if the relation is in BCNF. So, for each option, find the candidate key and if LHS of every FD contains the candidate key then it is in BCNF.

Q10: Consider the following relation instance R:

A	B	C	D	E
a1	b1	c1	d1	e1
a2	b2	c2	d2	e2
a1	b3	c1	d1	e3
a2	b2	c3	d3	e4
a3	b3	c4	d4	e5
a2	b2	c5	d3	e6

Which of the following FDs are satisfied for the above relation instance?
(I) $A \rightarrow B$ (II) $A \rightarrow C$ (III) $C \rightarrow A$ (IV) $AB \rightarrow C$ (V) $BC \rightarrow A$ (VI) $AC \rightarrow B$ (VII) $E \rightarrow ABCD$ (VIII) $D \rightarrow A$

A. II, III and VI only

B. III, IV, V and VII only

C. III, V, VII and VIII only

D. I and V only

Explanation: An FD is a constraint for the tuples in a relation from a database. An FD $X \rightarrow Y$ require that the value of X uniquely determines the value of Y where X and Y are set of attributes. You need check for each of the given FDs to determine if it is satisfied by the relation instance.

Q11: For relation $R = (L, M, N, O, P)$ the following dependencies hold:
$\{ M \rightarrow O, NO \rightarrow P, P \rightarrow L$ and $L \rightarrow MN \}$
R is decomposed into $R1 = (L, M, N, P)$ and $R2 = (M, O)$. The decomposition is:

A. Lossless decomposition and dependency preserving

B. Lossless decomposition and not dependency preserving

C. Lossy decomposition and dependency preserving

D. Lossy decomposition and not dependency preserving

Explanation: To check lossless decomposition, following conditions must hold:
1. Union of Attributes of $R1$ and $R2$ must be equal to attribute of R.
2. Intersection of Attributes of $R1$ and $R2$ must not be NULL.
3. Common attribute must be a key for at least one relation ($R1$ or $R2$)
To check dependency preservation, following condition(s) must hold:
If we decompose a relation R into relations $R1$ and $R2$, all dependencies of R either must be a part of $R1$ or $R2$ or must be derivable from combination of FDs of $R1$ and $R2$.

Q12: Consider a relation $R(A, B, C, D)$. Suppose, the set of functional dependencies F over the relation R is ϕ except the trivial functional dependencies. Assuming atomicity of attributes, which among the following is the highest normal form satisfied by the relation R?

A. 1NF

B. 2NF

C. 3NF

D. BCNF

Explanation: As for a relation to be in some normal form some constraints are imposed on FDs but since there is no non-trivial FD hence the above relation is in highest normal form i.e BCNF.

Q13: Consider the following relation for published books:
BOOK (*book_title*, author, *type_of_book*, price, *author_affiliation*, publisher).
Suppose the following dependencies exist:
{ (*book_title*) → (*publisher*, *type_of_book*); (*type_of_book*) → (*price*); (*author*) → (*author_affiliation*) }.
What is the highest normal form that the above relation satisfies?

A. 1NF

B. 2NF

C. 3NF

D. None of these

Explanation: There exist partial functional dependencies ((*book_title*) → (*publisher*, *type_of_book*) and (*author*) → (*author_affiliation*)) which violates it from being in 2NF, hence it is in 1NF (atomicity of attributes is always assumed until specifically mentioned).

Q14: Consider a relation R with set of functional dependencies F, and another relation S with another set of functional dependencies G. The highest normal forms of one of the relations is BCNF, while for the other, it is 3NF. It is, however, not known which is which. To make a correct identification using a single test, which of the following tests should be used on the relations? (Assume that the closure of F and G are available.)

A. Dependency-preservation

B. 3NF definition

C. Lossless-join

D. BCNF definition

Explanation: This is because the relation whose highest normal form is 3NF will fail the BCNF test, but both will pass the 3NF test.

Q15: Consider a system with the following disk specifications:

- Rotational Speed = $7200rpm$
- Seek Time = $4ms$
- Data stored per track = $153.6KB$
- Size of file A = $200B$
- Number of sectors per track = 300

What is the *average access time* to read file A from disk under sequential and random I/O configuration?

A. Sequential I/O: 8.194*ms*; Random I/O: 8.194*ms*

B. Sequential I/O: 8.176*ms*; Random I/O: 8.194*ms*

C. Sequential I/O: 8.176*ms*; Random I/O: 8.176*ms*

D. Sequential I/O: 8.194*ms*; Random I/O: 8.176*ms*

Explanation: Two important observations: 1. File *A* fits in a single sector therefore random and sequential I/O time will not make difference. 2.The smallest unit that can be read is one sector. Rest of the part is trivial, use the formula mentioned in slide to calculate the average access time.