# horizontal lineDatabase R&D Exercise

Assignment 7

I confirm that this is my own work and that use of material from other sources, including the Internet, has been properly and fully acknowledged and referenced.

|  |  |
| --- | --- |
| Name: | Pang, Jinhao |
| Date: | 2022.11.16 |
| NYU ID: | N19475049 |
| Course Section Number: | csci-ga.2433-001 |



**Total in points** (100 points total): \_\_\_\_\_

**Professor’s Comments:**

|  |
| --- |
|  |

**14.24. Consider the universal relation R = {A, B, C, D, E, F, G, H, I, J} and the set of functional dependencies F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}. What is the key for R? Decompose R into 2NF and then 3NF relations.**

Key for R: A, B

2NF: {{A, B, C},{A, D, E, I, J }, {B, F, G, H}}

3NF: {{D, I, J}, {A, D, E}, {F, G, H}, {B, F}, {A, B, C}}

**14.26.**

**a. Given the previous extension (state), which of the following dependen- cies may hold in the above relation? If the dependency cannot hold, explain why by specifying the tuples that cause the violation.**

**i.A→B, ii.B→C, iii.C→B, iv.B→A, v.C→A**

1. A→B

Cannot hold. Because A cannot uniquely determine B. Tuple 1 and 2 have the same value in A but different values in B

1. B→C

Hold.

1. C→B

Cannot hold. Because C cannot uniquely determine B. Tuple 1 and 3 have the same value in A but different values in B

1. B→A

Cannot hold. Because B cannot uniquely determine A. Tuple 1 and 5 have the same value in A but different values in B

1. C→A

Cannot hold. Because C cannot uniquely determine A. Tuple 1 and 3 have the same value in A but different values in B

**b. Does the above relation have a potential candidate key? If it does, what is it? If it does not, why not?**

Yes. {A, B} or {A, C} can be candidate key.

**14.30. Consider the following relation:**

**CAR\_SALE(Car#, Date\_sold, Salesperson#, Commission%, Discount\_amt)**

**Assume that a car may be sold by multiple salespeople, and hence {Car#, Salesperson#} is the primary key. Additional dependencies are**

**Date\_sold → Discount\_amt and Salesperson# → Commission%**

**Based on the given primary key, is this relation in 1NF, 2NF, or 3NF? Why or why not? How would you successively normalize it completely?**

The domain of attributes includes only atomic values and the value of any attribute is a single value from the domain. Hence, it is 1NF. It’s not 2NF. Because {Car#, Salesperson#} **→** Commission% is not fully functional dependency. It’s not 3NF. Because it has a transitive dependency {Car#, Salesperson#}**→** Date\_sold**→** Discount\_amt.

**2NF:**

CAR\_SALE(Car#, Date\_sold, Salesperson#, Discount\_amt)

Commission(Salesperson#, Commission%)

**3NF:**

CAR\_SALE(Car#, Date\_sold, Salesperson#)

Commission(Salesperson#, Commission%)

Discount(Date\_sold, Discount\_amt)

**14.35.**

**a. Based on a common-sense understanding of the above data, what are the**

**possible candidate keys of this relation?**

{Author, Edition} {Author, Copyright\_Year}

**b. Justify that this relation has the MVD {Book} →{Author} | {Edition, Year}.**

For tuple 1 (t1) and tuple 3 (t2), t1[Book\_Name] = t2[Book\_Name]. For tuple 2 (t3) and tuple 4 (t4), t3[Book\_Name] = t4[Book\_Name] = t1[Book\_Name] = t2[Book\_Name]; t3[{Author} | {Edition, Year}] = t3[{Author} | {Edition, Year}] and t2[{Author} | {Edition, Year}] = t4[{Author} | {Edition, Year}];

As (R- (X U Y)) = {}= Z, Hence t3[Z] = t2[Z] and t1[Z] = t4[Z]. Hence, MVD {Book} ->>{Author} | {Edition, Year}

**c. What would be the decomposition of this relation based on the above MVD? Evaluate each resulting relation for the highest normal form it possesses.**

4NF:

BOOK\_NA(Book\_Name, Author)

|  |  |
| --- | --- |
| Book\_Name | Author |
| DB\_fundamentals | Navathe |
| DB\_fundamentals | Elmasri |

BOOK\_NEC(Book\_Name, Edition, Copright\_Year)

|  |  |  |
| --- | --- | --- |
| Book\_Name | Edition | Copright\_Year |
| DB\_fundamentals | 4 | 2004 |
| DB\_fundamentals | 5 | 2007 |

**15.21. Consider the relation REFRIG(Model#, Year, Price, Manuf\_plant, Color), which is abbreviated as REFRIG(M, Y, P, MP, C), and the following set F of functional dependencies: F = {M → MP, {M, Y} → P, MP → C}**

**a. Evaluate each of the following as a candidate key for REFRIG, giving reasons why it can or cannot be a key: {M}, {M, Y}, {M, C}.**

{M} cannot be a key. Because M cannot uniquely determine P.

{M, Y} can be a key. Because {M, Y} can uniquely determine MP and P, then uniquely determine C.

{M, C} cannot be a key. Because {M, C} cannot uniquely determine P.

**b. Based on the above key determination, state whether the relation REFRIG is in 3NF and in BCNF, and provide proper reasons.**

It is not in 3NF. Because there is a transitive dependency where Z = {MP}, M → Z → C. It is not BCNF as well. Because every relation in BCNF is also in 3NF. Hence, it is not 3NF and BCNF

**c. Consider the decomposition of REFRIG into D = {R1(M, Y, P), R2(M, MP, C)}. Is this decomposition lossless? Show why. (You may consult the test under Property NJB in Section 14.5.1.)**

(R1 ∩ R2) → (R2 – R1)) = M → {MP, C}. As M→MP, MP→C, then M→{MP, C}. Therefore, according to Property NJB, this decomposition is lossless.

**15.31. Consider the following decompositions for the relation schema R of Exercise 14.24. Determine whether each decomposition has (1) the dependency preservation property, and (2) the lossless join property, with respect to F. Also determine which normal form each relation in the decomposition is in.**

**a. D1 ={R1,R2,R3,R4,R5};R1 ={A,B,C},R2 ={A,D,E},R3 ={B,F}, R4 = {F, G, H}, R5 = {D, I, J}**

F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}.

1. Has dependency preservation property.
2. Has the lossless join property

R1: 3NF, R2:3NF, R3:3NF, R4:3NF, R5:3NF

**b. D2 = {R1, R2, R3}; R1 = {A, B, C, D, E}, R2 = {B, F, G, H}, R3 = {D, I, J}**

F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}.

1. Has dependency preservation property.
2. Has the lossless join property

R1: 1NF, R2:2NF, R3:3NF

**c. D3 ={R1,R2,R3,R4,R5};R1 ={A,B,C,D},R2 ={D,E},R3 ={B,F}, R4 = {F, G, H}, R5 = {D, I, J}**

F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}.

1. Not have dependency preservation property.
2. Not have the lossless join property

R1: 1NF, R2: 1NF, R3:3NF, R4:3NF, R5:3NF