ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΛ 342 Βάσεις Δεδομένων
Διδάσκων: Γ. Σαμάρας

5η σειρά ασκήσεων: Συναρτησιακές Εξαρτήσεις και Κανονικοποίηση.

Λύσεις

Μέρος Α. <u>Συναρτησιακές Εξαρτήσεις</u>

1. **Αποδείξτε ή διαψεύστε τους ακόλουθους κανόνες. Μπορείτε να χρησιμοποιήστε τους κανόνες του Armstrong ή να δώσετε αντιπαράδειγμα.**
   a. **{W→Y, X→Z} |= {WX→Y }**
   b. **{X→Y} και Z ⊆ Y |= { X→Z }**
   c. **{ X→Y, X→W, WY→Z} |= {X→Z}**
   d. **{XY→Z, Y→W} |= {XW→Z}**
   e. **{X→Z, Y→Z} |= {X→Y}**
   f. **{X→Y, XY→Z} |= {X→Z}**

Answer:
(a) {W ->Y, X ->Z} |= {WX ->Y } Proof:
(1) W ->Y (given)
(2) X ->Z (given)
(3) WX ->YX (using IR2 (augmentation) to augment (1) with X)
(4) YX ->Y (using IR1 (reflexivity), knowing that Y subset-of YX)
(5) WX ->Y (using IR3 (transitivity) on (3) and (4))

(b) {X ->Y} and Z subset-of Y |= { X ->Z } Proof:
(1) X ->Y (given)
(2) Y ->Z (using IR1 (reflexivity), given that Z subset-of Y)
(3) X ->Z (using IR3 (transitivity) on (1) and (2))

(c) { X ->Y, X ->W, WY ->Z} |= {X ->Z} Proof:
(1) X ->Y (given)
(2) X ->W (given)
(3) WY ->Z (given)
(4) X ->XY (using IR2 (augmentation) to augment (1) with X)
(5) XY ->WY (using IR2 (augmentation) to augment (2) with Y)

(6) X ->WY (using IR3 (transitivity) on (4) and (5))
(7) X ->Z (using IR3 (transitivity) on (6) and (3))

(d) {XY ->Z, Y ->W} |= {XW ->Z}
Disproof: X Y Z W
t 1 =x 1 y 1 z 1 w 1
t 2 =x 1 y 2 z 2 w 1
The above two tuples satisfy XY ->Z and Y ->W but do not satisfy XW ->Z

(e) {X ->Z, Y ->Z} |= {X ->Y}
Disproof: X Y Z
t 1 =x 1 y 1 z 1
t 2 =x 1 y 2 z 1
The above two tuples satisfy X ->Z and Y ->Z but do not satisfy X ->Y

(f) {X ->Y, XY ->Z} |= {X ->Z} Proof:
(1) X ->Y (given)
(2) XY ->Z (given)
(3) X ->XY (using IR2 (augmentation) to augment (1) with X)
(4) X ->Z (using IR3 (transitivity) on (3) and (2))

2. **Θεωρείστε τα ακόλουθα σύνολα συναρτησιακών εξαρτήσεων: F={A→C, AC→D, E→AD, E→H} και G={A→CD, E→AH}. Είναι ισοδύναμα;**

Answer:
To show equivalence, we prove that G is covered by F and F is covered by G.
Proof that G is covered by F:
{A} + = {A, C, D} (with respect to F), which covers A ->CD in G
{E} + = {E, A, D, H, C} (with respect to F), which covers E ->AH in G
Proof that F is covered by G:
{A} + = {A, C, D} (with respect to G), which covers A ->C in F
{A, C} + = {A, C, D} (with respect to G), which covers AC ->D in F
{E} + = {E, A, H, C, D} (with respect to G), which covers E ->AD and E ->H in F

3. **Θεωρείστε το σχήμα EMP_DEPT της εικόνας 14.3(a) (από το επισυναπτόμενο φυλλάδιο) και το σύνολο συναρτησιακών εξαρτήσεων G = {SSN→{ENAME, BDATE, ADDRESS, DNUMBER}, DNUMBER→{DNAME, DMGRSSN} }. Βρείτε τις κλειστότητες (closures) των {SSN}+ και {DNUMBER}+ σε σχέση με το G.**

Answer:
{SSN} + ={SSN, ENAME, BDATE, ADDRESS, DNUMBER, DNAME, DMGRSSN}
{DNUMBER} + ={DNUMBER, DNAME, DMGRSSN}

4. Είναι το σύνολο συναρτησιακών εξαρτήσεων G της προηγούμενης άσκησης ελάχιστο (minimal); Αν όχι, δώστε ένα ελάχιστο σύνολο ισοδύναμο με το G (αποδείξτε ότι είναι ισοδύναμο με το G).

Answer:
The set G of functional dependencies in Exercise 4 is not minimal, because it violates rule 1 of minimality (every FD has a single attribute for its right hand side).
The set F is an equivalent minimal set: F= {SSN->{ENAME},SSN->{BDATE}, SSN->{ADDRESS}, SSN ->{DNUMBER} , DNUMBER ->{DNAME}, DNUMBER->{DMGRSSN}}

To show equivalence, we prove that G is covered by F and F is covered by G.
Proof that G is covered by F:
{SSN} + = {SSN, ENAME, BDATE, ADDRESS, DNUMBER, DNAME, DMGRSSN}
(with respect to F), which covers SSN ->{ENAME, BDATE, ADDRESS, DNUMBER} in G
{DNUMBER} + ={DNUMBER, DNAME, DMGRSSN}
(with respect to F), which covers DNUMBER ->{DNAME, DMGRSSN} in G

Proof that F is covered by G:
{SSN} + = {SSN, ENAME, BDATE, ADDRESS, DNUMBER, DNAME, DMGRSSN}
(with respect to G), which covers SSN ->{ENAME}, SSN ->{BDATE},
SSN ->{ADDRESS}, and SSN ->{DNUMBER} in F
{DNUMBER} + ={DNUMBER, DNAME, DMGRSSN}
(with respect to G), which covers DNUMBER->{DNAME} & DNUMBER->{DMGRSSN} in F

5. Ποιες ανωμαλίες ενημέρωσης παρατηρείτε στις σχέσεις EMP_PROJ και EMP_DEPT των εικόνων 14.3 και 14.4 (από το επισυναπτόμενο φυλλάδιο);

Answer:
In EMP_PROJ, the partial dependencies {SSN}->{ENAME} and {PNUMBER}->{PNAME, PLOCATION} can cause anomalies. For example, if a PROJECT temporarily has no EMPLOYEEs working on it, its information (PNAME, PNUMBER, PLOCATION) will not be represented in the database when the last EMPLOYEE working on it is removed (deletion anomaly). A new PROJECT cannot be added unless at least one EMPLOYEE is assigned to work on it (insertion anomaly). Inserting a new tuple relating an existing EMPLOYEE to an existing PROJECT requires checking both partial dependencies; for example, if a different value is entered for PLOCATION than those values in other tuples with the same value for PNUMBER, we get an update anomaly. Similar comments apply to EMPLOYEE information. The reason is that EMP_PROJ represents the relationship between EMPLOYEEs and PROJECTs, and at the same time represents information concerning EMPLOYEE and PROJECT entities.

In EMP_DEPT, the transitive dependency {SSN}->{DNUMBER}->{DNAME, DMGRSSN} can cause anomalies. For example, if a DEPARTMENT temporarily has no EMPLOYEEs working for it, its information (DNAME, DNUMBER, DMGRSSN) will not be represented in the database when the last EMPLOYEE working on it is removed (deletion anomaly). A new DEPARTMENT cannot be added unless at least one EMPLOYEE is assigned to work on it (insertion anomaly). Inserting a new tuple relating a new EMPLOYEE to an existing DEPARTMENT requires checking the transitive dependencies; for example, if a different value is entered for DMGRSSN than those values in other tuples with the same value for DNUMBER, we get an update anomaly. The reason is that EMP_DEPT represents the relationship between EMPLOYEEs and DEPARTMENTs, and at the same time represents information concerning EMPLOYEE and DEPARTMENT entities.

Μέρος Β. Κανονικοποίηση

6. **Σε πια κανονική μορφή είναι το σχήμα LOTS της εικόνας 1.(a) (από το επισυναπτόμενο φυλλάδιο) σε σχέση με τους ορισμούς των κανονικών μορφών που υπολογίζουν μόνο το πρωτεύων κλειδί (primary key); Θα ήταν στην ίδια κανονική μορφή αν χρησιμοποιούσατε τους γενικούς ορισμούς των κανονικών μορφών;**

Answer:
If we only take the primary key into account, the LOTS relation schema in Figure 1(a) will be in 2NF since there are no partial dependencies on the primary key .

However, it is not in 3NF, since there are the following two transitive dependencies on the primary key:
PROPERTY_ID# ->COUNTY_NAME ->TAX_RATE, and PROPERTY_ID# ->AREA ->PRICE.

Now, if we take all keys into account and use the general definition of 2NF and 3NF, the LOTS relation schema will only be in 1NF because there is a partial dependency COUNTY_NAME ->TAX_RATE on the secondary key {COUNTY_NAME, LOT#}, which violates 2NF.

7. **Αποδείξτε ότι κάθε σχέση με μόνο δυο χαρακτηριστικά είναι σε BCNF.**

Answer:
Consider a relation schema R={A, B} with two attributes. The only possible (non-trivial) FDs are {A} ->{B} and {B} ->{A}. There are four possible cases:

(i) No FD holds in R. In this case, the key is {A, B} and the relation satisfies BCNF.
(ii) Only {A} ->{B} holds. In this case, the key is {A} and the relation satisfies BCNF.
(iii) Only {B} ->{A} holds. In this case, the key is {B} and the relation satisfies BCNF.
(iv) Both {A} ->{B} and {B} ->{A} hold. In this case, there are two keys {A} and {B} and the relation satisfies BCNF.

Hence, any relation with two attributes is in BCNF.

8. **Θεωρείστε το ακόλουθο σχήμα R = {A, B, C, D, E, F, G, H, I} και το σύνολο συναρτησιακών εξαρτήσεων F = { {A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J} }. Πιο είναι το κλειδί του R? Αποσυνθέστε το R σε σχέσεις 2NF, και μετά σε σχέσεις 3NF.**

Answer:
A minimal set of attributes whose closure includes all the attributes in R is a key. (One can also apply algorithm 15.4a (see chapter 15 in the textbook)). Since the closure of {A, B}, {A, B}+ = R, one key of R is {A, B} (in this case, it is the only key).

To normalize R intuitively into 2NF then 3NF, we take the following steps (alternatively, we can apply the algorithms discussed in Chapter 15):
First, identify partial dependencies that violate 2NF. These are attributes that are functionally dependent on either parts of the key, {A} or {B}, alone. We can calculate the closures {A}+ and {B}+ to determine partially dependent attributes:
{A}+ = {A, D, E, I, J}. Hence {A} -> {D, E, I, J} ({A} -> {A} is a trivial dependency)
{B}+ = {B, F, G, H}, hence {A} -> {F, G, H} ({B} -> {B} is a trivial dependency)

To normalize into 2NF, we remove the attributes that are functionally dependent on part of the key (A or B) from R and place them in separate relations R1 and R2, along with the part of the key they depend on (A or B), which are copied into each of these relations but also remains in the original relation, which we call R3 below:
R1 = {A, D, E, I, J}, R2 = {B, F, G, H}, R3 = {A, B, C}

The new keys for R1, R2, R3 are underlined. Next, we look for transitive dependencies in R1, R2, R3. The relation R1 has the transitive dependency {A} -> {D} -> {I, J}, so we remove the transitively dependent attributes {I, J} from R1 into a relation R11 and copy the attribute D they are dependent on into R11. The remaining attributes are kept in a relation R12. Hence, R1 is decomposed into R11 and R12 as follows:
R11 = {D, I, J}, R12 = {A, D, E}

The relation R2 is similarly decomposed into R21 and R22 based on the transitive dependency {B} -> {F} -> {G, H}:
R2 = {F, G, H}, R2 = {B, F}

The final set of relations in 3NF are {R11, R12, R21, R22, R3}

9. Επαναλάβετε την άσκηση 8 με το σύνολο συναρτησιακών εξαρτήσεων G = { {A, B}→{C}, {B, D}→{E, F}, {A, D}→{G, H}, {A}→{I}, {H}→{J} }.

Answer:
To help in solving this problem systematically, we can first find the closures of all single attributes to see if any is a key on its own as follows:
{A}+ -> {A, I}, {B}+ -> {B}, {C}+ -> {C}, {D}+ -> {D}, {E}+ -> {E}, {F}+ -> {F},
{G}+ -> {G}, {H}+ -> {H, J}, {I}+ -> {I}, {J}+ -> {J}

Since none of the single attributes is a key, we next calculate the closures of pairs of attributes that are possible keys:
{A, B}+ -> {A, B, C, I}, {B, D}+ -> {B, D, E, F}, {A, D}+ -> {A, D, G, H, I, J}

None of these pairs are keys either since none of the closures includes all attributes. But the union of the three closures includes all the attributes:
{A, B, D}+ -> {A, B, C, D, E, F, G, H, I}

Hence, {A, B, D} is a key. (Note: Algorithm 15.4a (see chapter 15 in the textbook) can be used to determine a key). Based on the above analysis, we decompose as follows, in a similar manner to problem 8, starting with the following relation R:
R = {A, B, D, C, E, F, G, H, I}

The first-level partial dependencies on the key (which violate 2NF) are:
{A, B} -> {C, I}, {B, D} -> {E, F}, {A, D}+ -> {G, H, I, J}

Hence, R is decomposed into R1, R2, R3, R4 (keys are underlined):
R1 = {A, B, C, I}, R2 = {B, D, E, F}, R3 = {A, D, G, H, I, J}, R4 = {A, B, D}

Additional partial dependencies exist in R1 and R3 because {A} -> {I}. Hence, we remove {I} into R5, so the following relations are the result of 2NF decomposition:
R1 = {A, B, C}, R2 = {B, D, E, F}, R3 = {A, D, G, H, J}, R4 = {A, B, D}, R5 = {A, I}

Next, we check for transitive dependencies in each of the relations (which violate 3NF).
Only R3 has a transitive dependency {A, D} -> {H} -> {J}, so it is decomposed into R31 and R32 as follows:
R31 = {H, J}, R32 = {A, D, G, H}
The final set of 3NF relations is {R1, R2, R31, R32, R4, R5}