

2^ο Σύνολο Ασκήσεων Ενδεικτικές Λύσεις

games(GAME_ID, HOME_TEAM_ID, VISITOR_TEAM_ID, SEASON, HOME_TEAM_WINS)

GAME_ID: Το μοναδικό αναγνωριστικό του παιχνιδιού

HOME_TEAM_ID: Το αναγνωριστικό της γηπεδούχου ομάδας

VISITOR_TEAM_ID: Το αναγνωριστικό της φιλοξενούμενης ομάδας

SEASON: Η περίοδος που παίχτηκε το παιχνίδι

HOME_TEAM_WINS: Μια δυαδική μεταβλητή για το αν κέρδισε ή όχι η γηπεδούχος ομάδα (1 αν κέρδισε, 0 αν έχασε)

games-detail(GAME_ID, TEAM_ID, PLAYER_ID, MIN, PTS, REB, AST, STL, BLK)

GAME_ID: Το μοναδικό αναγνωριστικό του παιχνιδιού

TEAM_ID: Το μοναδικό αναγνωριστικό της ομάδας

PLAYER_ID: Το μοναδικό αναγνωριστικό του παίκτη

MIN: Ο αριθμός των λεπτών που έπαιξε ο παίκτης

PTS: Ο αριθμός των πόντων που πέτυχε ο παίκτης

REB: Ο αριθμός των rebounds του παίκτη

AST: Ο αριθμός των assists του παίκτη

STL: Ο αριθμός των steals του παίκτη

BLK: Τ Ο αριθμός των blocks του παίκτη

players(PLAYER_ID, PLAYER_NAME, TEAM_ID, SEASON)

PLAYER_ID: Το μοναδικό αναγνωριστικό του παίκτη

PLAYER_NAME: Το όνομα του παίκτη

TEAM_ID: Το αναγνωριστικό της ομάδας που ανήκει ο παίκτης

SEASON: Η περίοδος στην οποία αναφέρεται η εγγραφή

teams(TEAM_ID, NICKNAME, YEAR_FOUNDED, CITY)

TEAM_ID: Το μοναδικό αναγνωριστικό της ομάδας

NICKNAME: Το nickname (παρατσούκλι) της ομάδας (π.χ., "Celtics" για τους Boston Celtics)

YEARFOUNDED: Η χρονιά που ιδρύθηκε η ομάδα

CITY: Η έδρα της ομάδας

Άσκηση 1: Σχεσιακή Άλγεβρα

Για την μετανομοσία χρησιμοποιήστε το σύμβολο \leftarrow

Υπάρχει εναλλακτικός συμβολισμός το ρ

(α) Τα ονόματα των παικτών που έχουν πετύχει triple double, δηλαδή, 10 ή περισσότερα points, assists και rebounds σε έναν αγώνα.

\sqcap PLAYER-NAME $\left(\sqcap$ PLAYER-ID $\left(\sigma$ AST \geq 10 AND PTS \geq 10 AND REB \geq 10 (games-detail) $\right)^*$
↑ players) ↑ φυσική συνένωση

αυτή η πρόβληση δώ είναι απαραίτητη

(β) Το όνομα του παίκτη που έχει πετύχει τους περισσότερους πόντους.

μετονομασία

$g_1 (pid_1, pts_1) \leftarrow \pi_{PLAYER-ID, PTS} (games-detail)$

$g_2 (pid_2, pts_2) \leftarrow \pi_{PLAYER-ID, PTS} (games-detail)$

$Q \leftarrow \left(\pi_{pid_1, pts_1} (g_1 \bowtie_{pts_1 < pts_2} g_2) \right)$

Τα κια σεντ που δει είναι στο Q
 είναι σεντ (pidL, ptsL) για τα οποία
 το ptsL είναι μεγαλύτερο ή ίσο
 από όλα τα pts που υπάρχουν

$\pi_{PLAYER-NAME} \left(\pi_{pidL} (g_1 - Q) \bowtie_{pidL = PLAYER-ID} (players) \right)$

για να πάρουμε το όνομα

~~εξω~~ εξω είναι σήμανση ή αφαίρεση

ή α είναι πρωτο σεντ

αυτή η προβολή δει είναι απαραίτητη

(β) Το όνομα του παίκτη που έχει πετύχει τους περισσότερους πόντους.

Εναλλακτικά θα μπορούσαν να βρουν πρώτα το μεγαλύτερο PTS και μετά τον παίκτη (ή τους παίκτες) που το έχουν πετύχει

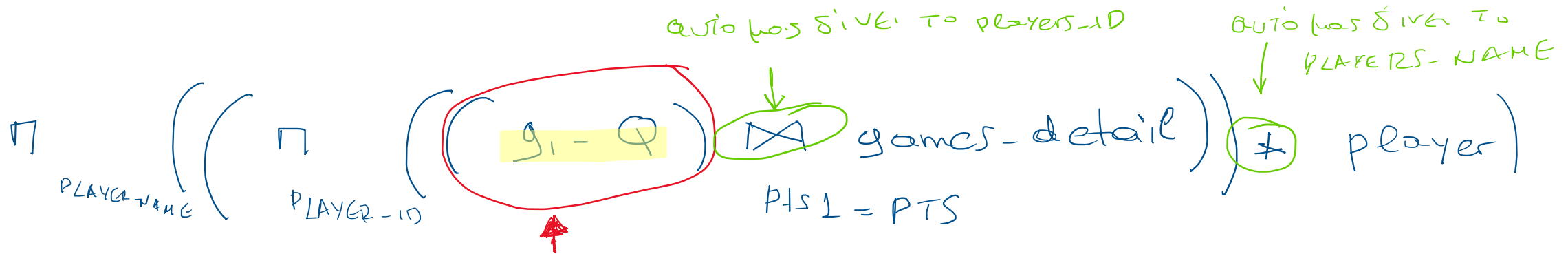
$$g_1(PTS_1) \leftarrow \pi_{PTS}(\text{games_detail})$$

$$g_2(PTS_2) \leftarrow \pi_{PTS}(\text{games_detail})$$

Μετασυσία

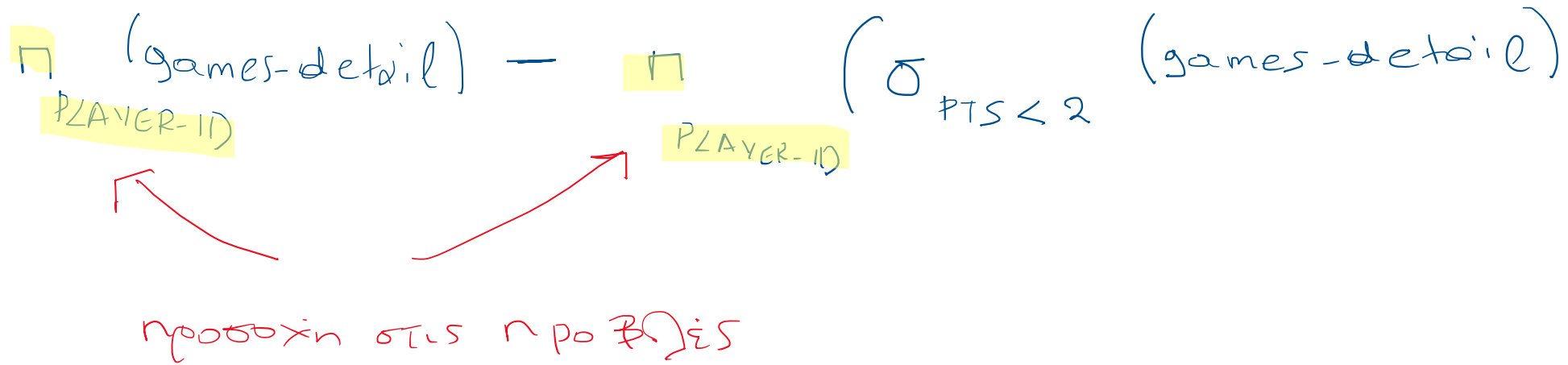
Όπως πριν, αλλά τώρα το ίδιο PTS_2 που δεν εμφανίζεται στο Q είναι το PTS_1 που είναι μεγαλύτερο ή ίσο από όλα τα PTS

$$Q \leftarrow \left(\pi_{PTS_1} \left(g_1 \bowtie_{PTS_1 < PTS_2} g_2 \right) \right)$$

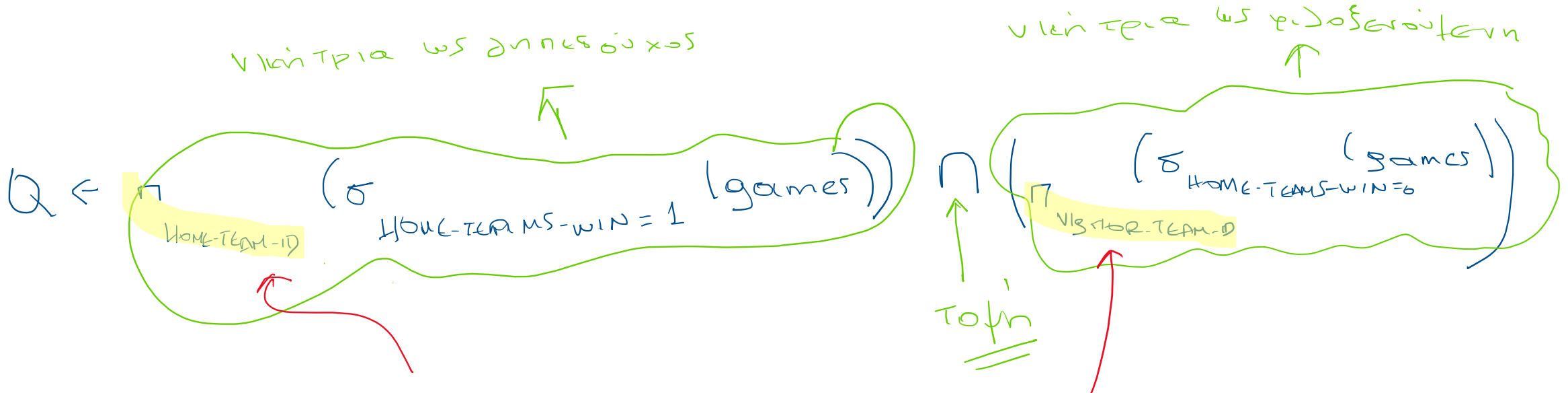


αυτό είναι το μεγαλύτερο PTS
 Ναι, προσοχή στις ηρωβωδές, εδώ είναι ταυτόσημοι τύποι (όχι Java)

(γ) Τους παίκτες που έχουν πετύχει τουλάχιστον δύο πόντους σε καθέναν από τους αγώνες που συμμετείχαν.



(δ) Τα nickname των ομάδων που έχουν κερδίσει τουλάχιστον μια φορά ως γηπεδούχοι και τουλάχιστον μια φορά ως φιλοξενούμενες.



πίνα: προσοχή στις τομές, να δίνου πριν την τομή

$$\pi_{\text{NICKNAME}} \left(Q * \text{teams} \right)$$

(ε) Ζευγάρια παικτών που έχουν παίξει τουλάχιστον σε ένα αγώνα σε αντίπαλες ομάδες και τουλάχιστον σε ένα αγώνα στην ίδια ομάδα.

Θα βρούμε ^① ζεύγη παικτών που έχουν παίξει στην ίδια ομάδα

^② ζεύγη παικτών που έχουν παίξει σε αντίπαλες

και ^③ θα πάρουμε την τομή του ^① και ^②

Ευθεία σταθερά και μια κλίση δύστη - αλλιώς θα χάσω την πιο αργή

(ε) Ζευγάρια παικτών που έχουν παίξει τουλάχιστον σε ένα αγώνα σε αντίπαλες ομάδες και τουλάχιστον σε ένα αγώνα στην ίδια ομάδα.

$$gd_1(g_1, t_1, p_1) \leftarrow \pi_{\text{GAME-ID, TEAM-ID, PLAYER-ID}} (\text{games-detail})$$

$$gd_2(g_2, t_2, p_2) \leftarrow \pi_{\text{GAME-ID, TEAM-ID, PLAYER-ID}} (\text{games-detail})$$

ιδίαι
 κέρδη
 διαδοχικών αγώνων
 ↓

$$Q_1 \leftarrow \pi_{p_1, p_2} (gd_1 \bowtie_{g_1=g_2 \text{ AND } t_1=t_2} gd_2 \text{ AND } p_1 \neq p_2)$$

προαιρετική, δε
 ↓
 κερδίζεται

πρόβλεψη στην τριτοτητα

$$Q_2 \leftarrow \pi_{p_1, p_2} (gd_1 \bowtie_{g_1=g_2 \text{ AND } t_1 \neq t_2} gd_2)$$

$$Q_1 \cap Q_2$$

Άσκηση 2: SQL

Στα επόμενα χρησιμοποιούμε WITH για να ορίσουμε CTE – προσωρινούς πίνακες

Εναλλακτικά οι πίνακες θα μπορούσαν να οριστούν στο FROM με χρήση AS

WITH TM(A, B) AS SFW

FROM (SFW) AS TM

(i) Το όνομα του παίκτη που έχει πετύχει τα περισσότερα triple double.

```
WITH BESTP(PLAYER_ID, TripleDoublesCount) AS  
(SELECT PLAYER_ID, COUNT(*)  
FROM games_detail  
WHERE PTS >= 10 AND REB >= 10 AND AST >= 10  
GROUP BY PLAYER_ID)
```

→ οι values (PLAYER_ID, TripleDoublesCount)
για κάθε παίκτη, #triple double

```
SELECT PLAYER_NAME  
FROM BESTP NATURAL JOIN players  
WHERE TripleDoublesCount >= ALL (SELECT TripleDoublesCount FROM BESTP);
```

← για να πάρουμε το όνομα του παίκτη

το μεγαλύτερο count

```
ή (JOIN players ON BESTP.Pid = players.PLAYER_ID)
```

```
ή TripleDoublesCount IN (SELECT MAX(TripleDoubleCount) FROM BESTP);
```

↑ in =

Γιατί όχι

ORDER BY και LIMIT 1

Πρόβλημα με τις ισοπαλίες στην κορυφή: Τι συμβαίνει αν δύο ή περισσότεροι παίκτες έχουν τον ίδιο αριθμό από triple doubles?

(ii) Την ομάδα που πέτυχε τους περισσότερους πόντους σε έναν αγώνα, σε ποιόν αγώνα και πόσους πόντους, δηλαδή να επιστρέφει τριάδες (όνομα-ομάδας, αναγνωριστικό-αγώνα, αριθμός-πόντων)

```
WITH BESTT(gid, tid, points) AS  
(SELECT GAME_ID, TEAM_ID, SUM(PTS) as points  
  FROM games_detail  
  GROUP BY GAME_ID, TEAM_ID)  
SELECT teams.NICKNAME, BESTT.gid, BESTT.points  
FROM BESTT JOIN teams ON BESTT.tid = t.team_id  
WHERE BESTT.points >= ALL (SELECT points FROM BESTT);
```

← group με δύο γνωρίσματα

↑ όπως πριν

(iii) Για κάθε ομάδα, το λόγο #νίκες/#ήττες στα εντός έδρας παιχνίδια που έχει δώσει.

```
(WITH TL(tid, countL) AS  
  SELECT HOME_TEAM_ID, COUNT(*)  
  FROM games  
  WHERE HOME-TEAM_WINS = 0  
  GROUP BY HOME_TEAM_ID,
```

#ήττες

```
TW(tid, countW) AS  
  SELECT HOME_TEAM_ID, COUNT(*)  
  FROM games  
  WHERE HOME-TEAM_WINS = 1  
  GROUP BY HOME_TEAM_ID)
```

#νίκες

```
SELECT tid, countW/countL  
FROM TL NATURAL JOIN TW;
```

(iv) Ανάμεσα στους παίκτες που έχουν παίξει συνολικά σε όλους τους αγώνες περισσότερο από 80 λεπτά, τον παίκτη που έχει πετύχει τους περισσότερους πόντους.

```
WITH P(pid, pts) AS (SELECT PLAYER_ID, SUM(PTS)
FROM games_detail
GROUP BY PLAYER_ID
HAVING SUM(TIME_TO_SEC(MIN)) > 4800)
SELECT PLAYER_NAME
FROM players JOIN P ON p.PLAYER_ID = P.pid
WHERE P.pts >= ALL (SELECT pts FROM P);
```

η συνθήκη εφαρμόζεται σε κάθε group

[προαιρετική +1 μονάδα] Στόχος της άσκησης είναι να χρησιμοποιήσετε SQL από Java ή Python.

Για να συνδεθείτε (connect) με τη βάση δεδομένων, χρησιμοποιείτε τον κατάλληλο driver.

```
import pymysql
conn = pymysql.connect(host='your_host', user='your_username', passwd='your_password',
db='your_dbname')
```

(i) Ο χρήστης θα δίνει ως είσοδο το όνομα μια ομάδας και θα τυπώνονται στην οθόνη ζεύγη με τα id των αγώνων που έπαιξε αυτή η ομάδα είτε ως γηπεδούχος είτε ως φιλοξενούμενη και 0 ή 1 αν κέρδισε ή έχασε.

```

def get_team_games(team_name):
    with conn.cursor() as cursor:

        query = """
        (select game_id, home_team_wins as won
        from games as g join teams as t on g.HOME_TEAM_ID = t.TEAM_ID
        where nickname = %s)
        union
        (select game_id, home_team_wins as won
        from games as g join teams as t on g.VISITOR_TEAM_ID = t.TEAM_ID
        where nickname = %s);

        """

        cursor.execute(query, team_name)
        for game_id, won in cursor:
            print(f"Game ID: {game_id}, Won: {won}")

team_name = input("Enter the name of the team: ")
get_team_games(team_name)

```

Άσκηση 3: Λογικός Σχεδιασμός

(α) Έστω το σχεσιακό σχήμα $R(A, B, C, D, E, H)$ στο οποίο ισχύει το σύνολο συναρτησιακών εξαρτήσεων $F = \{A \rightarrow BC, AD \rightarrow BE, C \rightarrow H, A \rightarrow H, D \rightarrow E\}$. Υπολογίστε το ελάχιστο κάλυμμα (minimum cover) της F .

Βήμα 1

$A \rightarrow B$
 $A \rightarrow C$
 $AD \rightarrow B$
 $AD \rightarrow E$
 $C \rightarrow H$
 $A \rightarrow H$
 $D \rightarrow E$

Βήμα 2 εστάθηκε περτιτε συνεισφατε στο α.κ.

* A περτιο στο $AD \rightarrow B$

$$(D)^+ = \{D, E\} \text{ οχι}$$

D περτιο στο $AD \rightarrow B$

$$A^+ = \{A, B, C, H\} \text{ ιρατο D περτιο}$$

$A \rightarrow B$ αντι για $AD \rightarrow B$

* A περτιο στο $AD \rightarrow E$

$$(D)^+ = \{D, E\} \text{ ιρα περτιο}$$

$D \rightarrow E$ αντι για $AD \rightarrow E$

(α) Έστω το σχεσιακό σχήμα $R(A, B, C, D, E, H)$ στο οποίο ισχύει το σύνολο συναρτησιακών εξαρτήσεων $F = \{A \rightarrow BC, AD \rightarrow BE, C \rightarrow H, A \rightarrow H, D \rightarrow E\}$. Υπολογίστε το ελάχιστο κάλυμμα (minimum cover) της F .

~~$A \rightarrow B$~~
 $A \rightarrow C$
 $A \rightarrow B$ ~~$AD \rightarrow B$~~
 $D \rightarrow E$ ~~$AD \rightarrow E$~~
 $C \rightarrow H$
 $A \rightarrow H$
 ~~$D \rightarrow E$~~

$A \rightarrow C$
 $A \rightarrow B$
 $D \rightarrow E$
 $C \rightarrow H$
 ~~$A \rightarrow H$~~

Βήμα 3: οφείτουν Σ.Ε.

$A \rightarrow C$?

$A^+ = \{A, B\} \neq C$

$A \rightarrow B$?

$A^+ = \{A, C, H\} \neq B$

$D \rightarrow E$?

$D^+ = \{D\} \neq E$

$C \rightarrow H$?

$C^+ = \{H\} \neq C$

$A \rightarrow H$?

$A^+ = \{C, B, H\} \neq H$

$f_{min} =$

$\{A \rightarrow C,$

$A \rightarrow B$

$D \rightarrow E,$

$C \rightarrow H\}$

(β) Θεωρείστε το σχεσιακό σχήμα (πίνακας) $R = (A, B, C, D, E, H)$ και το σύνολο συναρτησιακών εξαρτήσεων $F = \{A \rightarrow C, B \rightarrow A, BD \rightarrow E, A \rightarrow D, AC \rightarrow B, E \rightarrow H\}$. Δώστε μια διάσπαση της R σε δύο ή περισσότερα BCNF σχεσιακά σχήματα και εξηγήστε αν διατηρεί ή όχι τις εξαρτήσεις.

Υπολογισμός υπονήγιων κλειδιών για να βρούμε αν υπάρχουν Σ.Ε. που παραβιάζουν τον BCNF ορισμό.

$$A^+ = \{D\}$$

$$E^+ = \{E, H\}$$

$$\rightarrow AB^+ = \{A, B, D, E, H, C\}$$

$$\rightarrow AC^+ = \{A, C, B, D, E, H\}$$

$$BD^+ = \{B, D, E, H\}$$

$$\rightarrow BC^+ = \{B, C, A, D, E, H\}$$

(β) Θεωρείστε το σχεσιακό σχήμα (πίνακας) $R = (A, B, C, D, E, H)$ και το σύνολο συναρτησιακών εξαρτήσεων $F = \{A \rightarrow C, BC \rightarrow A, BD \rightarrow E, A \rightarrow D, AC \rightarrow B, E \rightarrow H\}$. Δώστε μια διάσπαση της R σε δύο ή περισσότερα BCNF σχεσιακά σχήματα και εξηγήστε αν διατηρεί ή όχι τις εξαρτήσεις.

$A \rightarrow D$
 $BD \rightarrow E$
 $E \rightarrow H$

προαβλέπων το BCNF

$R \rightarrow (B, D, E)$ BCNF ✓
 $\hookrightarrow A, B, C, D, H$
 χρειάζεται διάσπαση??

$AB \rightarrow C$ ✓
 $BC \rightarrow A$ ✓
 $A \rightarrow D$
 $AC \rightarrow B$ ✓

(A, D) BCNF ✓
 $\hookrightarrow A, B, C, H$

- (A, B, C) ✓
- (A, B, H) ✓

 $AB \rightarrow C$
 $BC \rightarrow A$

(β) Θεωρείστε το σχεσιακό σχήμα (πίνακας) $R = (A, B, C, D, E, H)$ και το σύνολο συναρτησιακών εξαρτήσεων $F = \{AB \rightarrow C, BC \rightarrow A, BD \rightarrow E, A \rightarrow D, AC \rightarrow B, E \rightarrow H\}$. Δώστε μια διάσπαση της R σε δύο ή περισσότερα BCNF σχεσιακά σχήματα και εξηγήστε αν διατηρεί ή όχι τις εξαρτήσεις.

B, D, E
 A, B, C, D, H
 \searrow
 B, D

η ζωή
 κλειδί σε τω...
 α) ...
 losses join

δε διατηρεί τις εξαρτήσεις
 rx $E \Rightarrow H$

