

2^ο Σετ Ασκήσεων

Βάσεων Δεδομένων

Αλέξανδρος Αγαπητός Χριστοδούλου 4839

Βασίλειος Ιωάννης Μπουζαμπαλίδης 4744

Κωνσταντίνος Παπαδόπουλος 4761

Άσκηση 1

A)

$\pi_{playerName}(\sigma_{pts \geq 10 \wedge reb \geq 10 \wedge ast \geq 10}(gamesDetails) \bowtie players(gamesDetails_{playerId} = players_{playerId}))$

B) $pts1 = pts$

$playerName1 = playerName$

$player1 = \pi_{playerName1, pts1}(players \bowtie gamesDetails)$

$player2 = \pi_{playerName2, pts2}(players \bowtie gamesDetails)$

$mostPoints = \pi_{playerName1}(\pi_{playerId}(players) -$
 $- (\pi_{playerId2}(\sigma_{pts2 < pts1}(player1 \bowtie player2 \bowtie gamesDetails))))$

Γ) $\pi_{playerName}((\pi_{playerId}(players) -$

$\pi_{playerId}(\sigma_{pts < 2}(gamesDetails))) \bowtie players(gameDetails_{playerId} =$
 $players_{playerId}))$

Δ) $\pi_{nickname}(teams) \bowtie (\pi_{visitorTeamId}(\sigma_{homeTeamWins=0}(games)) \cap$
 $(\pi_{visitorTeamId}(\sigma_{homeTeamWins=1}(games))(visitorTeamId = visitorTeamId)))$

Ε) $homeTeams = gameDetails \bowtie games(gameDetails.gameId =$
 $games.gameId \wedge gameDetails.teamId = games.homeTeamId)$

$visitorTeams$

$= gameDetails \bowtie games(gameDetails.gameId$
 $= games.gameId \wedge gameDetails.teamId = games.visitorTeamId)$

$$\begin{aligned}
\text{Teammates} &= \pi_{\text{gameDetails.playerId1}, \text{gameDetails.playerId2}} (\text{homeTeams} \cup \text{visitorTeams}) \\
\text{oppositeTeams} &= \text{gameDetails} \bowtie \text{games}(\text{gameDetails.gameId} \\
&= \text{games.gameId} \wedge \text{gameDetails.teamId} \neq \text{games.homeTeamId}) \\
\text{Opponents} &= \pi_{\text{gameDetails.playerId}, \text{gameDetails.playerId}} (\text{oppositeTeams}) \\
\text{Result} &= \text{Teammates} \cap \text{Opponents}
\end{aligned}$$

Άσκηση 2

A) Δημιουργούμε τους πίνακες εκτελώντας τα παρακάτω script στο MySQLWorkbench

```

4 USE BasketballDB ;
5 DROP TABLE IF EXISTS 'GAMES' ;
6
7 CREATE TABLE 'GAMES' (
8     game_id INTEGER NOT NULL,
9     home_team_id INTEGER,
10    visitor_team_id INTEGER,
11    season INTEGER ,
12    home_team_wins BOOLEAN,
13
14    PRIMARY KEY(game_id),
15    FOREIGN KEY (home_team_id) REFERENCES teams(team_id) ON DELETE CASCADE ON UPDATE CASCADE,
16    FOREIGN KEY (visitor_team_id) REFERENCES teams(team_id) ON DELETE CASCADE ON UPDATE CASCADE,
17    CHECK (home_team_id <> visitor_team_id),
18    CHECK (season > 1900)
19 );
20
21
22 DROP TABLE IF EXISTS 'Games_Details';
23
24 CREATE TABLE 'GAMES_DETAILS' (
25     game_id INTEGER,
26     team_id INTEGER,
27     player_id INTEGER,
28     min VARCHAR(255),
29     pts INTEGER DEFAULT NULL,
30     reb INTEGER DEFAULT NULL,
31     ast INTEGER DEFAULT NULL,
32     stl INTEGER DEFAULT NULL,
33     blk INTEGER DEFAULT NULL,
34
35     PRIMARY KEY(game_id, player_id, team_id),
36     FOREIGN KEY(player_id) REFERENCES 'PLAYERS'(player_id) ON DELETE CASCADE ON UPDATE CASCADE,
37     FOREIGN KEY(game_id) REFERENCES 'GAMES'(game_id) ON DELETE CASCADE ON UPDATE CASCADE,
38     FOREIGN KEY(team_id) REFERENCES 'TEAMS'(team_id) ON DELETE CASCADE ON UPDATE CASCADE
39 );

```

```

5 DROP TABLE IF EXISTS `PLAYERS` ;
6
7 CREATE TABLE `PLAYERS` (
8     player_id INTEGER NOT NULL,
9     player_name VARCHAR(255),
10    team_id INTEGER,
11    season VARCHAR(255),
12
13
14    PRIMARY KEY(player_id,team_id),
15    FOREIGN KEY(team_id) REFERENCES `TEAMS`(team_id) ON DELETE CASCADE ON UPDATE CASCADE,
16    CHECK (season > 1900)
17 );

```

```

1 CREATE DATABASE IF NOT EXISTS `basketballdb`;
2
3 USE `basketballdb`;
4
5 DROP TABLE IF EXISTS `TEAMS`;
6
7 CREATE TABLE `TEAMS` (
8     team_id INTEGER NOT NULL,
9     nickname VARCHAR(255),
10    year_founded INTEGER,
11    city VARCHAR(255),
12
13    PRIMARY KEY(team_id),
14    CHECK (year_founded > 1800)
15 );

```

Παρατηρήσαμε ότι χρειάζεται να μπει ένα check στον πίνακα games ώστε να εξασφαλίσουμε ότι το home_team_id δεν θα είναι ίδιο με το visitor_team_id. Όμως η MySQL δεν δέχεται αυτό τον έλεγχο. Η εναλλακτική που βρήκαμε ψάχνοντας το documentation της MySQL ήταν η χρήση triggers, όμως θεωρήσαμε ότι εφόσον δεν καλύπτονται στο μάθημα δεν μπορούμε να τα χρησιμοποιήσουμε. Συνεπώς κάνουμε την σύμβαση πως εφόσον τα δεδομένα φορτώνονται από έτοιμα αρχεία και δεν αλλάζουν δυναμικά από τους χρήστες, είναι σωστά και δεν περιέχουν πλειάδες με ίδια τα home_team_id, visitor_team_id. Όπως επίσης δεν θα μπορεί να γίνει κάποια προσθήκη τέτοιας πλειάδας από κάποιον χρήστη. Έτσι αφαιρέσαμε το συγκεκριμένο check.

Στην συνέχεια χρησιμοποιούμε την εντολή :

```

LOAD DATA INFILE "C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\teams.csv"
INTO TABLE teams
FIELDS TERMINATED BY ','
LINES TERMINATED BY "\n"
IGNORE 1 ROWS

```

για να φορτώσουμε τα δεδομένα κάθε CSV αρχείου στον αντίστοιχο πίνακα.

Όμως για να μπορέσει να δουλέψει η εντολή χρειάζεται να μεταφέρουμε τα αρχεία στο «C:\ProgramData\MySQL\MySQL Server 8.0\Uploads» λόγω κάποιων security settings του MySQLWorkbench.

Επίσης παρατηρήσαμε πως στο αρχείο «games_detail.csv» υπήρχαν πολλά player_id τα οποία δεν υπήρχαν στο αρχείο «players.csv», συνεπώς παρατηρήθηκε παραβίαση ξένου κλειδιού.

Για αυτό το λόγο υλοποιήσαμε το παρακάτω script σε python το οποίο αφαιρεί από το dataset όσα player_id δεν υπάρχουν στο αρχείο «players.csv».

```
csvFixer.py > ...
1 import pandas as pd
2
3 Codeium: Refactor | Explain | Generate Docstring | X | CodiumAI: Options | Test this function
4 def filter_rows(file1, file2, file3):
5     df1 = pd.read_csv(file1)
6     df2 = pd.read_csv(file2)
7
8
9     df2_filtered = df2[df2['PLAYER_ID'].isin(df1['PLAYER_ID'])]
10    df2_filtered.dropna(subset=['MIN', 'REB', 'AST', 'STL', 'BLK', 'PTS'], inplace=True)
11
12
13    df2_filtered.to_csv(file3, index=False)
14
15 if __name__ == "__main__":
16     file1_path = "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/players.csv"
17     file2_path = "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/games_detail.csv"
18     file3_path = "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/result.csv"
19
20     filter_rows(file1_path, file2_path, file3_path)
21
```

Επίσης το συγκεκριμένο script διαγράφει τις πλειάδες που έχουν κάποια κενή τιμή στα στατιστικά των παικτών.

Έπειτα φορτώνουμε με την ίδια εντολή LOAD DATA INFILE το τροποποιημένο αρχείο «result.csv» στον πίνακα games_details.

B)

Σε κάθε query, στο τέλος του screenshot φαίνεται και το αποτέλεσμα.

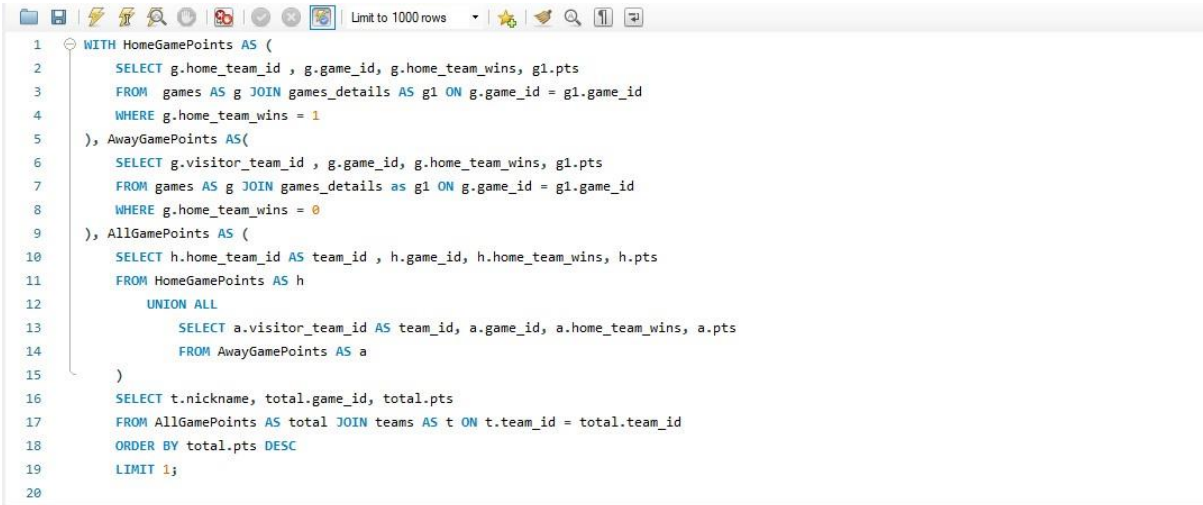
```
SQL File 3*
1 WITH TripleDoubles AS (
2     SELECT
3         p1.player_id, p2.player_name, COUNT(*) AS AllTripleDoubles
4     FROM
5         games_details AS p1 JOIN players AS p2 ON p1.player_id = p2.player_id
6     WHERE
7         (p1.pts >= 10 AND p1.reb >= 10 AND p1.ast >= 10)
8     GROUP BY p1.player_id, p2.player_name
9 )
10 SELECT
11     player_id,
12     player_name,
13     AllTripleDoubles
14 FROM
15     TripleDoubles
16 ORDER BY
17     AllTripleDoubles DESC
18 LIMIT 1;
```

player_id	player_name	AllTripleDoubles
201566	Russell Westbrook	2299


i)

Στο συγκεκριμένο query αρχικά δημιουργούμε τον ενδιάμεσο πίνακα TripleDoubles στον οποίο αποθηκεύουμε τα player_id, player_name και τον αριθμό των triple double (pts, reb, ast) όλων των παιχτών που έχουν πετύχει Triple Double.

Στην συνέχεια επιλέγουμε αυτά τα 3 πεδία από τον πίνακα και εφαρμόζουμε ORDER BY triple doubles DESC LIMIT 1 ώστε να πάρουμε τον παίκτη με τον μεγαλύτερο αριθμό Triple Doubles.



```
1 WITH HomeGamePoints AS (  
2     SELECT g.home_team_id , g.game_id, g.home_team_wins, g1.pts  
3     FROM games AS g JOIN games_details AS g1 ON g.game_id = g1.game_id  
4     WHERE g.home_team_wins = 1  
5 ), AwayGamePoints AS(  
6     SELECT g.visitor_team_id , g.game_id, g.home_team_wins, g1.pts  
7     FROM games AS g JOIN games_details AS g1 ON g.game_id = g1.game_id  
8     WHERE g.home_team_wins = 0  
9 ), AllGamePoints AS (  
10    SELECT h.home_team_id AS team_id , h.game_id, h.home_team_wins, h.pts  
11    FROM HomeGamePoints AS h  
12    UNION ALL  
13    SELECT a.visitor_team_id AS team_id, a.game_id, a.home_team_wins, a.pts  
14    FROM AwayGamePoints AS a  
15 )  
16    SELECT t.nickname, total.game_id, total.pts  
17    FROM AllGamePoints AS total JOIN teams AS t ON t.team_id = total.team_id  
18    ORDER BY total.pts DESC  
19    LIMIT 1;  
20
```



nickname	game_id	pts
Lakers	20500591	81

ii)

Σε αυτό το query, δημιουργούμε 2 ενδιάμεσους πίνακες για όλους τους πόντους που σκόραρε κάθε ομάδα σε παιχνίδια που ήταν γηπεδούχος(HomeGamePoints) και φιλοξενούμενη(AwayGamePoints). Επίσης δημιουργούμε έναν ενδιάμεσο πίνακα AllGamesPoints στον οποίο κάνουμε UNION ALL τους άλλους 2. Τέλος επιλέγουμε το nickname των παιχτών μαζί με το game_id και τους πόντους ώστε να εμφανίζονται ως 3αδες και επιλέγουμε GROUP BY pts DESC LIMIT 1 ώστε να βρούμε την ομάδα με τους περισσότερους.

```

1 • SELECT
2     t.team_id,
3     t.nickname,
4     COUNT(g.game_id) AS totalGames,
5     SUM(g.home_team_wins) AS wins,
6     COUNT(g.game_id) - SUM(g.home_team_wins) AS losses,
7     SUM(g.home_team_wins) / (COUNT(g.game_id) - SUM(g.home_team_wins)) AS winLossRatio
8 FROM
9     teams AS t
10 JOIN
11     games AS g ON t.team_id = g.home_team_id
12 GROUP BY
13     t.team_id
14 HAVING
15     COUNT(g.game_id) > 0;

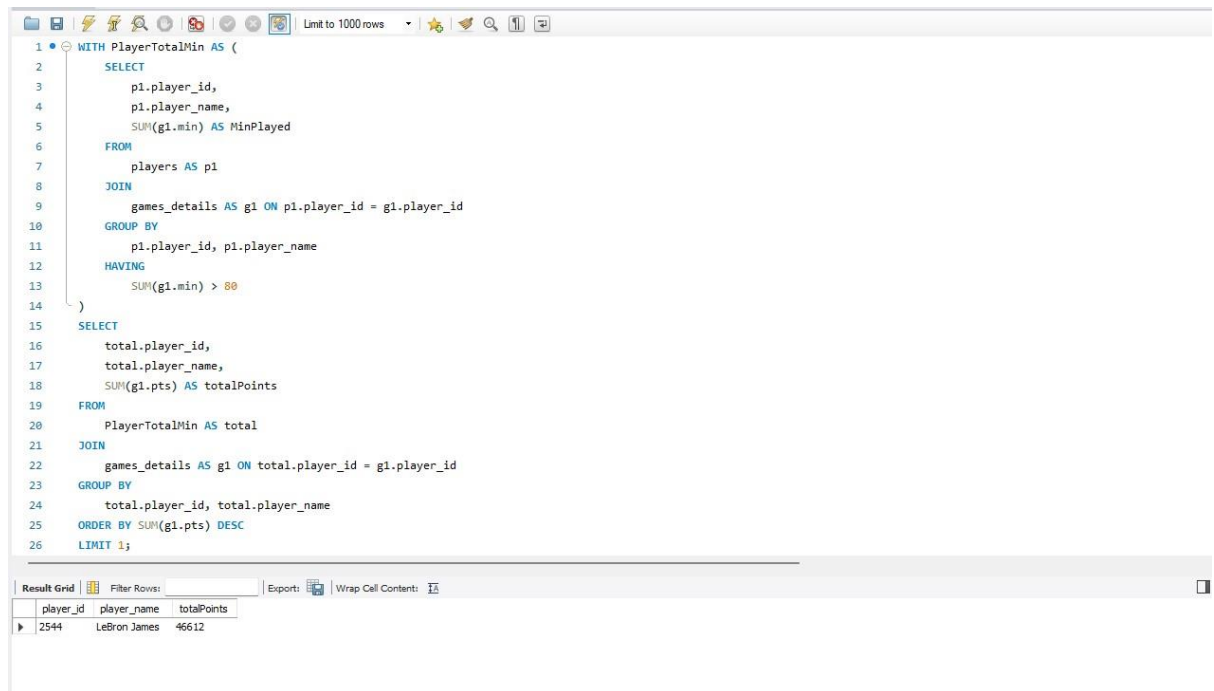
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: [↗](#)

	team_id	nickname	totalGames	wins	losses	winLossRatio
▶	1610612737	Hawks	883	504	379	1.3298
	1610612738	Celtics	949	603	346	1.7428
	1610612739	Cavaliers	920	554	366	1.5137
	1610612740	Pelicans	850	464	386	1.2021
	1610612741	Bulls	897	517	380	1.3605
	1610612742	Mavericks	907	594	313	1.8978
	1610612743	Nuggets	871	588	283	2.0777
	1610612744	Warriors	907	606	301	2.0133
	1610612745	Rockets	899	570	329	1.7325
	1610612746	Clippers	903	538	365	1.4740
	1610612747	Lakers	969	573	396	1.4470
	1610612748	Heat	958	632	326	1.9387
	1610612749	Bucks	882	509	373	1.3646
	1610612750	Timberw...	846	399	447	0.8926
	1610612751	Nets	876	447	429	1.0420
	1610612752	Knicks	840	396	444	0.8919
	1610612753	Magic	876	457	419	1.0907
	1610612754	Pacers	889	550	339	1.6224
	1610612755	76ers	876	473	403	1.1737
	1610612756	Suns	889	513	376	1.3644
	1610612757	Trail Blaz...	871	524	347	1.5101
	1610612758	Kings	838	415	423	0.9811
	1610612759	Spurs	941	684	257	2.6615
	1610612760	Thunder	892	536	356	1.5056
	1610612761	Raptors	891	529	362	1.4613
	1610612762	Jazz	872	573	299	1.9164
	1610612763	Grizzlies	881	527	354	1.4887
	1610612764	Wizards	851	451	400	1.1275
	1610612765	Pistons	902	511	391	1.3069
	1610612766	Hornets	796	391	405	0.9654

iii)

Σε αυτό το query ορίζουμε το πλήθος των παιχνιδιών ως totalGames, το άθροισμα των νικών ως wins και αφαιρώντας από το σύνολο των παιχνιδιών τις νίκες παίρνουμε τις ήττες που ορίζουμε ως losses. Τέλος εφαρμόζουμε την διαίρεση για να επιστραφεί ο λόγος νίκες/ήττες στα εντός έδρας παιχνίδια ως winLossRatio.



```
1 WITH PlayerTotalMin AS (  
2     SELECT  
3         p1.player_id,  
4         p1.player_name,  
5         SUM(g1.min) AS MinPlayed  
6     FROM  
7         players AS p1  
8     JOIN  
9         games_details AS g1 ON p1.player_id = g1.player_id  
10    GROUP BY  
11        p1.player_id, p1.player_name  
12    HAVING  
13        SUM(g1.min) > 80  
14 )  
15 SELECT  
16     total.player_id,  
17     total.player_name,  
18     SUM(g1.pts) AS totalPoints  
19 FROM  
20     PlayerTotalMin AS total  
21 JOIN  
22     games_details AS g1 ON total.player_id = g1.player_id  
23 GROUP BY  
24     total.player_id, total.player_name  
25 ORDER BY SUM(g1.pts) DESC  
26 LIMIT 1;
```

player_id	player_name	totalPoints
2544	LeBron James	46612

iv)

Σε αυτό το query αρχικά δημιουργούμε τον ενδιάμεσο πίνακα PlayerTotalMins στον οποίο αποθηκεύουμε τους παίκτες για τους οποίους το άθροισμα των λεπτών συμμετοχής είναι μεγαλύτερο του 80. Έπειτα επιλέγουμε το player_id, το player_name και το άθροισμα των λεπτών και χρησιμοποιώντας την εντολή ORDER BY SUM(g1.pts) DESC LIMIT 1 επιστρέφεται ο παίκτης που έχει αγωνιστεί σε όλα τα εντός έδρας παιχνίδια πάνω από 80 λεπτά και έχει σκοράρει τους περισσότερους πόντους.

Γ)

Υλοποιήσαμε το ερώτημα σε Python.

Αρχικά χρειάζεται να κάνουμε install το «mysql-connector»

Και στην συνέχεια συνδεόμαστε με την βάση «basketballdb» που έχουμε φτιάξει μέσα στο MySQLWorkbench και εκτελούμε τα 2 query.

Για το πρώτο query χρειάζεται να πάρουμε το nickname της ομάδας από τον χρήστη οπότε δημιουργούμε την μεταβλητή userQuery. Στην συνέχεια επιλέγουμε τα game_id και home_team_wins κάνοντας JOIN τους πίνακες Games με

1 : Τον πίνακα Teams ως home για τους εντός έδρας αγώνες

2: Τον πίνακα Teams ως visitor για τους εκτός έδρας αγώνες

Με την συνθήκη το team_id που υπάρχει ως ξένο κλειδί στον πίνακα Games να ταυτίζεται με το κλειδί team_id του κάθε πίνακα Teams.

userQueries.py > [x] cnx

```
1  import mysql.connector
2
3  cnx = mysql.connector.connect(
4      host="127.0.0.1",
5      user="root",
6      password="1234",
7      database="basketballdb"
8  )
9
10 userQuery = input("Enter team nickname: ")
11
12 cursor = cnx.cursor()
13
14
15
16 query = """
17     SELECT
18         g.game_id,
19         g.home_team_wins
20     FROM
21         Games AS g
22     JOIN
23         Teams AS home ON g.home_team_id = home.team_id
24     JOIN
25         Teams AS visitor ON g.visitor_team_id = visitor.team_id
26     WHERE
27         home.nickname = %s OR visitor.nickname = %s
28 """
29 cursor.execute(query, (userQuery, userQuery))
30
31 results = cursor.fetchall()
32
33
34 for result in results:
35     print(f"Game ID: {result[0]}, Home Team Wins: {result[1]}")
36
37
```


Για το δεύτερο query επιλέγουμε τα player_name και team_id κάνοντας JOIN τον πίνακα players με τον πίνακα teams με συνθήκη τα team_id να είναι ίδια. Με αυτό τον τρόπο παίρνουμε όλους τους παίκτες που έχει κάθε ομάδα και τους αποθηκεύουμε στην λίστα sec_results. Στην συνέχεια χρειάζεται να κάνουμε κάποιες τροποποιήσεις στην λίστα ώστε να εκτυπώνονται τα αποτελέσματα με τον τρόπο που ζητείται στην εκφώνηση.

```
37
38 sec_query = """
39     WITH
40         TeamPlayers AS (
41             SELECT
42                 p.player_name,
43                 p.team_id
44             FROM
45                 players AS p
46             JOIN
47                 teams ON p.team_id = teams.team_id
48         )
49     SELECT
50         tp.team_id,
51         tp.player_name
52     FROM
53         TeamPlayers AS tp
54 """
55
56 cursor.execute(sec_query)
57 sec_results = cursor.fetchall()
58 team_players = {}
59
60 for team_id, player_name in sec_results:
61     if team_id not in team_players:
62         team_players[team_id] = set()
63     team_players[team_id].add(player_name)
64
65 for team_id, players in team_players.items():
66     players_str = ', '.join(players)
67     print("\n" + f"TEAM ID: {team_id}, PLAYERS: {players_str} " + "\n")
68
69 cursor.close()
70 cnx.close()
```

Η εκτέλεση του προγράμματος μαζί με τα αποτελέσματα φαίνεται παρακάτω (δεν συμπεριλάβαμε όλα τα αποτελέσματα καθώς θα έκανε τα screenshot μη αναγνώσιμα)

```
PS C:\Users\kostas\Documents\Sxoli\5oEtos\9th_Semester\Databases> & 'C:\Users\kostas\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\kostas\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '53066' '-.' 'C:\Users\kostas\Documents\Sxoli\5oEtos\9th_Semester\Databases\userQueries.py'

Enter team nickname: Celtics

Game ID: 41100301, Home Team Wins: 1
Game ID: 41100302, Home Team Wins: 1
Game ID: 41100305, Home Team Wins: 0
Game ID: 41100307, Home Team Wins: 1
Game ID: 41900303, Home Team Wins: 0
Game ID: 41900304, Home Team Wins: 1
Game ID: 41900306, Home Team Wins: 1
Game ID: 42100301, Home Team Wins: 1
Game ID: 42100302, Home Team Wins: 0
Game ID: 42100305, Home Team Wins: 0
Game ID: 42100307, Home Team Wins: 0
Game ID: 20300229, Home Team Wins: 1
Game ID: 20300536, Home Team Wins: 1
Game ID: 20400711, Home Team Wins: 1
Game ID: 20401164, Home Team Wins: 0
Game ID: 20500643, Home Team Wins: 1
Game ID: 20600192, Home Team Wins: 0
Game ID: 20601113, Home Team Wins: 1
Game ID: 20700980, Home Team Wins: 0
Game ID: 20701157, Home Team Wins: 0
Game ID: 20800139, Home Team Wins: 0
Game ID: 20800988, Home Team Wins: 1
Game ID: 20900950, Home Team Wins: 1
Game ID: 20901191, Home Team Wins: 0
Game ID: 21000939, Home Team Wins: 0
Game ID: 21100694, Home Team Wins: 0
Game ID: 21200086, Home Team Wins: 0
Game ID: 21200238, Home Team Wins: 1
Game ID: 21300247, Home Team Wins: 1
Game ID: 21300767, Home Team Wins: 0
Game ID: 21400763, Home Team Wins: 1
Game ID: 21401221, Home Team Wins: 0
Game ID: 21500110, Home Team Wins: 0
Game ID: 21500786, Home Team Wins: 1
Game ID: 21600706, Home Team Wins: 0
Game ID: 21700067, Home Team Wins: 0
Game ID: 21701162, Home Team Wins: 1
Game ID: 21800070, Home Team Wins: 1
Game ID: 21900615, Home Team Wins: 1
Game ID: 21901236, Home Team Wins: 1
Game ID: 22000669, Home Team Wins: 1
Game ID: 22000685, Home Team Wins: 0
Game ID: 22100489, Home Team Wins: 1
Game ID: 22101198, Home Team Wins: 1
Game ID: 41700113, Home Team Wins: 1

TEAM ID: 1610612737, PLAYERS: Mike Dunleavy, Magnum Rolle, Isaac Humphries, Kyle Korver, Andrew White III, Jannero Pargo, Adreian Payne, Lamar Patterson, Thomas Robinson, Damien Wilkins, Ricardo Marsh, Pape Sy, Juan Dixon, Marco Belinelli, Matt Costello, Joe Smith, Zaza Pachulia, James Johnson, Isma'il Muhammad, Jerry Stackhouse, Mario West, Royal Ivey, Jared Cunningham, Damion James, Will Bynum, Ray Spalding, Dwight Howard, Jaylen Adams, Cam Reddish, Willie Green, Josh Powell, Ivan Johnson, Quinn Cook, Othello Hunter, Daniel Hamilton, Vince Carter, Anthony Tolliver, Malcolm Delaney, Ryan Kelly, James Anderson, Dahntay Jones, Adonis Thomas, David Lighty, Keith Benson, Donald Sloan, Elton Brand, Maurice Evans, Cartier Martin, Mike Mills, Omeri Spellman, Carlhell Johnson, Evan Turner, Charles Brown Jr., Tyler Dorsey, DeMarre Carroll, Randolph Morris, RJ Hunter, Shelvin Mack, Jeremy Lin, DeAndre Hunter, Garrett Siler, Joe Johnson, Ersan Iltyasova, John Jenkins, Taurean Prince, Jeremy Evans, Vladimir Radmanovic, Etan Thomas, James Munnally, Tiago Splitter, Kirk Hinrich, Josh Magette, Jabari Parker, Jeff Teague, Jason Collins, Devin Harris, Justin Holiday, DeAndre' Bembry, Thabo Sefolosha, Dexter Pittman, Pero Antic, BJ Johnson, Josh Smith, Gary Neal, Dennis Schroder, Tracy McGrady, Richard Delk, Jordan Crawford, Chandler Parsons, Allen Crabbe, Trae Young, Kevin Huerter, Tyrone Wallace, Jordan Mathews, Paul Millsap, Isaiah Taylor, Tyler Zeller, Brad Wanamaker, Marvin Williams, Mike Muscala, Damion Lee, Mike Bibby, Anthony Morrow, Jarell Eddie, Deyonta Davis, Kris Humphries, Jeremy Tyler, Kent Bazemore, Terran Petteway, Armoni Brooks, Johan Petro, Austin Daye, Earl Barron, Dewayne Dedmon, Jordan Sibert, Tyler Cavanaugh, Frank Robinson, Eric Dawson, Tim Hardaway Jr., Justin Anderson, Aaron Miles, DeQuan Jones, Evan Brock, Courtney Sims, John Collins, Edgar Sosa, Nicolas Brussino, Jamal Crawford, C.J. Anderson, Al Horford, Gustavo Ayon, Jose Calderon, Lou Williams, Luke Babbitt, Alex Poythress, Edy Tavares, Damian Jones, Antonius Cleveland, DeShawn Stevenson, Cole Aldrich, Tahjere McCall, Brandon Goodwin, Jaylen Morris, Miles Plumlee, Erick Dampier, Bruno Fernando, Arsalan Kazemi, Marcus Derrickson, Jarrett Jack, Alex Len, Hilton Armstrong, Mike Scott

TEAM ID: 1610612738, PLAYERS: DJ White, Kemba Walker, Andrew White III, Phil Pressey, Paul Pierce, Menad Krstic, Lester Hudson, Jonathan Holmes, Tacko Fall, Carlos Arroyo, Kaiser Gates, Devin Williams, Cody Clarke, Daniel Theis, Shaquille O'Neal, Jarvis Varnado, Robert Williams III, Marcus Smart, Dwight Powell, PJ Dozier, Leandro Barbosa, Guerschon Yabusele, Jordan Mickey, Courtney Lee, Mario West, Ray Allen, E'Twaun Moore, Will Bynum, Greg Stiemsma, Tim Frazier, Shelden Williams, Terry Rozier, Marquis Daniels, Jabari Bird, Rajon Rondo, Nick King, Luke Harangody, Von Wafer, MarShon Brooks, Rasheed Wallace, Marcus Thornton, Vander Blue, Kelly Olynyk, L.J. Peak, Christian Watford, Tayshaun Prince, Sean Williams, Michael Sweetney, Justin Bibbs, Brandon Wright, Max Strus, Tony Allen, Keith Bogans, Evan Turner, Walt Lemon Jr., Ben Bentil, Dionte Christmas, RJ Hunter, Abdel Nader, Grant Williams, Jamar Smith, Amir Johnson, Romeo Langford, Perry Jones III, Chris Johnson, Jayson Tatum, Gerald Green, Corey Walden, Xavier Silas, Stephane Lasme, Jason Terry, Oliver Lafayette, Troy Murphy, Joel Anthony, Marcus Landry, John Holland, Semih Erden, Chris Wilcox, David Lee, Damen Bell-Holter, Jason Collins, Javonte Green, Nate Robinson, Micah Downs, Eddie House, Jonathan Gibson, Greg Monroe, Isaiah Thomas, Kendrick Perkins, Semi Ojeleye, Jonas Jerebko, Jordan Crawford, Yante Maten, Gilbert Brown, James Young, Marcus Georges-Hunt, Kamron Taylor, Jae Crowder, Jeff Roberson, Andre Dawkins, Demetrius Jackson, Tremont Waters, Delonte West, Avery Bradley, Tyler Zeller, Brad Wanamaker, Kevin Garnett, Damion Lee, Jarell Eddie, Mickael Pietrus, Carsen Edwards, Darko Milicic, Kris Humphries, Ryan Hollins, Erik Murphy, Kris Joseph, DeShawn Sims, Jared Sullinger, Jamaal Sampson, Shane Larkin, Rob Kurz, Brandon Bass, Jaylen Brown, Vincent Poirier, Fab Melo, Henry Walker, Aron Baynes, Levi Randolph, Brian Scalabrine, Marcus Morris Sr., Michael Finley, Glen Davis, Terrence Williams, Malcolm Miller, Chris Babb, Sasha Pavlovic, Gordon Hayward, J.R. Giddens, Keyon Dooling, Jameer Nelson, Al Horford, Jerryd Bayless, Shavlik Randolph, Jalen Jones, Enes Kanter, JaJuan Johnson, Gigi Datome, Jermaine O'Neal, Kyrie Irving, Rodney McGruder, Gerald Wallace, Jeff Green, Daniel Dixon, Kadeem Allen, Vitor Faverani
```

Άσκηση 3

A)

Διαδικασία εύρεσης ελάχιστου καλύμματος.

$R(A, B, C, D, E, H)$

$F = \{A \rightarrow BC, AD \rightarrow BE, C \rightarrow A, A \rightarrow H, D \rightarrow E\}$

1) $\{A \rightarrow B, A \rightarrow C, AD \rightarrow B, AD \rightarrow E, C \rightarrow H, A \rightarrow H, D \rightarrow E\}$

Παρατηρούμε ότι το B το οποίο είναι δεξί μέλος της εξάρτησης $AD \rightarrow B$, μπορεί να παραχθεί και από την συναρτησιακή εξάρτηση $A \rightarrow B$. Αρα το D στην εξάρτηση $AD \rightarrow B$ είναι περιττό συνεπώς αφαιρείται.

Επιπλέον, παρομοίως αφαιρείται το A από την συναρτησιακή εξάρτηση $AD \rightarrow E$ λόγω παρουσίας της εξάρτησης $D \rightarrow E$.

2) $\{A \rightarrow B, A \rightarrow C, A \rightarrow B, D \rightarrow E, C \rightarrow H, A \rightarrow H, D \rightarrow E\}$

Αφαιρούμε τις διπλότυπες εξαρτήσεις $A \rightarrow B, D \rightarrow E$

3) $\{A \rightarrow B, A \rightarrow C, D \rightarrow E, C \rightarrow H, A \rightarrow H\}$

Παρατηρούμε ότι το H που είναι δεξί μέλος της συναρτησιακής εξάρτησης $A \rightarrow H$, μπορεί να παραχθεί από την ακολουθία εξαρτήσεων $A \rightarrow C, C \rightarrow H$. Αρα ολόκληρη η εξάρτηση $A \rightarrow H$ είναι περιττή και αφαιρείται.

4) $\{A \rightarrow B, A \rightarrow C, D \rightarrow E, C \rightarrow H\}$

Συνδυάζονται οι πρώτες δυο συναρτησιακές εξαρτήσεις $A \rightarrow B, A \rightarrow C$ σε $A \rightarrow BC$ ώστε να μειωθεί το πλήθος τους για το ελάχιστο κάλυμμα.

5) $\{A \rightarrow BC, D \rightarrow E, C \rightarrow H\}$

Καταλήγουμε στο ελάχιστο κάλυμμα της F.

B)

Για να ανακατασκευάσουμε τη δοθείσα σχέση $R = (A, B, C, D, E, H)$ σε BCNF σχήματα, διατηρώντας τις λειτουργικές εξαρτήσεις, πρέπει να ακολουθήσουμε τον αλγόριθμο ανακατασκευής BCNF.

$$F = \{AB \rightarrow C, BC \rightarrow A, BD \rightarrow E, A \rightarrow D, AC \rightarrow B, E \rightarrow H\}$$

Διασπάμε την F στις

$$F1 = (A, B, C, D)$$

$$F2 = (D, E, H)$$

Πρέπει να ελέγξουμε αν η ένωση των $F1, F2$ δίνει την αρχική F .

Αρχικά υπολογίζουμε τις κλειστότητες για την $F1$

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

$$B^+ = \{B\}$$

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

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

$$(AB)^+ = \{A, B, C\} = AB \rightarrow C$$

$$(BC)^+ = \{A, B, C\} = BC \rightarrow A$$

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

$$(AC)^+ = \{B, C, A\} = AC \rightarrow B$$

$$\text{Τελικά } F1 = \{A, B, C\} = \{AB \rightarrow C, BC \rightarrow A, AC \rightarrow B, A \rightarrow D\}$$

Αντίστοιχα για την $F2$

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

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

$$\text{Τελικά } F2 = \{E, H\} = \{E \rightarrow H\}$$

$$F1 \cup F2 = \{AB \rightarrow C, BC \rightarrow A, AC \rightarrow B, E \rightarrow H, A \rightarrow D\}$$

Παρατηρούμε ότι οι συναρτησιακές εξαρτήσεις της ένωσης των $F1, F2$ είναι μέλη της F άρα η διάσπαση είναι σωστή.

Η πρώτη διάσπαση, $F1$, είναι σε BCNF, καθώς όλα τα χαρακτηριστικά καθορίζονται από το πρωτεύον κλειδί. Το πρωτεύον κλειδί του $F1$ είναι το A και το A καθορίζει το B, C και το D . Επομένως, όλα τα χαρακτηριστικά στο $F1$ καθορίζονται από το πρωτεύον κλειδί.

Κανένα non-prime attribute δεν καθορίζεται από άλλο non-prime attribute.

Δεν υπάρχουν non-prime ιδιότητες στο $F1$.

Η δεύτερη διάσπαση, $F2$, είναι επίσης σε BCNF, καθώς όλα τα χαρακτηριστικά καθορίζονται από το πρωτεύον κλειδί: Το πρωτεύον κλειδί του $F2$ είναι το E και το E καθορίζει το H . Επομένως, όλα τα χαρακτηριστικά στο $F2$ καθορίζονται από το πρωτεύον κλειδί.

Κανένα non-prime attribute δεν καθορίζεται από άλλο non-prime attribute.

Δεν υπάρχουν non-prime ιδιότητες στο $F2$.

Η διάσπαση διατηρεί τις εξαρτήσεις καθώς :

Ας υποθέσουμε ότι $AB \rightarrow C$ ισχύει για το F . Παρατηρούμε ότι υπάρχει στο $F1$ άρα και στην ένωση $F1 \cup F2$, και επειδή οι εξαρτήσεις τους είναι μέλη της F , θα υπάρχει και στην F .

Ας υποθέσουμε ότι $BC \rightarrow A$ ισχύει για το F . Παρατηρούμε ότι υπάρχει στο $F1$ άρα και στην ένωση $F1 \cup F2$, και επειδή οι εξαρτήσεις τους είναι μέλη της F , θα υπάρχει και στην F .

Ας υποθέσουμε ότι $BD \rightarrow E$ ισχύει για το F . Στη συνέχεια, με τη εξάρτηση $AC \rightarrow B$ στο $F1$, ξέρουμε ότι $AC \rightarrow B$ ισχύει στο F . Επίσης με την εξάρτηση $A \rightarrow D$ στο $F1$, ξέρουμε ότι $A \rightarrow D$ ισχύει και στο F . Επομένως, εφόσον τα B, D παράγονται, τότε και το $BD \rightarrow E$ ισχύει και για το F .

Ας υποθέσουμε ότι $A \rightarrow D$ ισχύει για το F . Παρατηρούμε ότι υπάρχει στο $F1$ άρα και στην ένωση $F1 \cup F2$, και επειδή οι εξαρτήσεις τους είναι μέλη της F , θα υπάρχει και στην F .

Ας υποθέσουμε ότι $AC \rightarrow B$ ισχύει για το F . Παρατηρούμε ότι υπάρχει στο $F1$ άρα και στην ένωση $F1 \cup F2$, και επειδή οι εξαρτήσεις τους είναι μέλη της F , θα υπάρχει και στην F .

Ας υποθέσουμε ότι $E \rightarrow H$ ισχύει για το F . Παρατηρούμε ότι υπάρχει στο $F2$ άρα και στην ένωση $F1 \cup F2$, και επειδή οι εξαρτήσεις τους είναι μέλη της F , θα υπάρχει και στην F .

Αρα καταλήγουμε ότι η διάσπαση διατηρεί όλες τις εξαρτήσεις της F .