

מבחן מסכם במערכות מסדי נתונים

קזינו "lucky strike" הקימה מערכת מידע לניהול המשחקים המתקיימים בקזינו שלה. לצורך שמירת הנתונים נבנו הטבלאות הבאות:

- דילרים (Dealer) – כל דילר מתאפיין במספר עובד ייחודי (מספר סידורי רץ המתחיל מ-1000), שמו המלא, תאריך לידה, תאריך תחילת עבודה ותעודת זהות. רק דילרים בני 24 ומעלה יכולים להתקבל לעבודה בקזינו.
- סוגי משחקים (GameType) – כל סוג משחק מזוהה על ידי שמו (לדוגמא פוקר, בלאק ג'ק וכו') ויש לשמור לו תיאור, כמות משתתפים מינימלית וכמות משתתפים מקסימאלית.
- משחקים (Game) – כל משחק הינו ביצוע של סוג משחק מסוים והוא מתחיל בתאריך ושעה (ייחודיים לסוג משחק הספציפי. באותו תאריך ושעה יכולים להתחיל כמה משחקים, אך הם יהיו בעלי סוג משחק שונה). סוג המשחק, התאריך והשעה מבדילים בין רשומה לרשומה בטבלה זו. בסיום המשחק מזין הדילר שמנהל את המשחק למערכת את שעת סיום המשחק. במקרה ובו סוג המשחק מוצא מהקזינו, יש למחוק את כל המשחקים שהתקיימו לו.
- שחקנים (Player) – השחקנים הינם באי הקזינו. כל אדם המגיע לקזינו צריך לספק את תעודת הזהות שלו, שמו, עיר המגורים שלו ותאריך לידה (רק אנשים בני 18 ומעלה מורשים להיכנס לקזינו). עבור כל שחקן נהיה מעוניינים לשמור גם כן את סוג המשחק האהוב עליו (סוג אחד בלבד). במקרה ובו השחקן לא מעוניין לבחור סוג משחק, יש להשאיר שדה זה ריק.
- שחקנים במשחק (PlayerInAGame) – בכל משחק משחקים כמות משתנה של שחקנים (בהתאם לערכי המינימום והמקסימום לסוג המשחק). בטבלה זו עלינו לשמור עבור כל שחקן באיזה משחק לקח חלק ולדעת את סכום ההימור שלו (סכום שלם בשקלים) וסכום הרווח / הפסד שלו במשחק (סכום ההפסד, אם קיים, הוא כסכום ההימור וסכום הרווח, אם קיים, הוא במקסימום כסכום ההימור של השחקן במשחק).
- קשר בין שחקן לדילר (PlayerDealerRelation) – על מנת לשמור על טוהר המידות, נהיה מעוניינים לדעת האם קיימים דילרים שהם קרובי משפחה של שחקנים בקזינו ולדעת את מידת הקרבה ביניהם (הורה, סב, אח, ילד, דוד או אחיין).

חלק א:

במהלך ההעברה של הטבלאות מהקזינו אליכם חלק מהטבלאות נעלמו. השלם את הטבלאות החסרות על פי הנכתב בסיפור.
טבלת דילר:

```
create table Dealer  
(  
)
```

טבלת סוג משחק:

```
create table GameType  
(  
    name nvarchar(20) primary key,  
    description nvarchar(20) not null,  
    minimumPlayers tinyint check (minimumPlayers > 0) not null,  
    maximumPlayers tinyint check (maximumPlayers > 0) not null,  
    check(minimumPlayers <= maximumPlayers)  
)
```

טבלת משחקים:

```
create table Game  
(  
    gameTypeName nvarchar(20) references GameType(name),  
    gameStartDateTime datetime,  
    gameEndTime time,  
    DealerInGame int not null references Dealer(number),  
    primary key(gameTypeName,gameStartDateTime)  
)
```

טבלת שחקנים:

```
create table Player  
(  
    ID char(9) primary key,  
    firstName nvarchar(20) not null,  
    surName nvarchar(20) not null,
```

```
city nvarchar(20) not null,  
birthDate date check(datediff(year,birthDate,getdate())>=18)  
not null,  
preferred nvarchar(20) references GameType(name)  
)
```

טבלת שחקנים במשחק:

```
create table PlayerInAGame  
(  
    playerID char(9) references Player(ID),  
    gameTypeName nvarchar(20),  
    gameDateTime datetime,  
    betAmount int check(betAmount>0) not null,  
    winLosAmount int,  
    check((winLosAmount = -betAmount) or  
        (winLosAmount>=betAmount)),  
    foreign key(gameTypeName,gameDateTime) references  
        Game(gameTypeName,gameStartDateTime),  
    primary key(playerID,gameTypeName,gameDateTime)  
)
```

טבלת קשר בין שחקן לדילר:

```
create table PlayerDealerRelation  
(  
  
)
```

חלק ב:

כתוב את השאילתות הבאות בשפת SQL:

1. כתוב שאילתה המחזירה משחקים (סוג משחק, תאריך ושעת התחלה) אשר השתתפו בהם יותר מ-10 משתתפים וגם היו יותר מחמישה משתתפים אשר הרוויחו במשחק
2. כתוב שאילתה המחזירה את כל המשחקים (סוג משחק, תאריך ושעת התחלה) בהם סך ההימור של המשתתפים במשחק גדול מ-10,000
3. כתוב שאילתה המחזירה עבור כל שחקן (ת"ז ושם מלא) בכמה סוגי משחקים שונים הוא לקח חלק. את התוצאות יש לסדר על פי כמות המשחקים בסדר יורד
4. כתוב שאילתה המחזירה משחקים (סוג, תאריך ושעת התחלה) בהם כמות השחקנים ששיחקו בפועל במשחק שונה מכמות השחקנים המותרת לסוג משחק זה
5. כתוב שאילתה המחזירה את סוגי המשחקים על פי מידת הפופולאריות שלהם: אם יותר מ-1000 שחקנים שונים שיחקו בו בחודש האחרון יוחזר "פופולארי", אם בין 500 ל-1000 שחקנים שונים שיחקו בו בחודש האחרון יוחזר "פופולארי למחצה" ואם פחות מ-500 שחקנים שונים שיחקו בו בחודש האחרון יוחזר "לא פופולארי"
6. כתוב שאילתה המחזירה מספרי עובדים של דילרים שהמשחק הראשון שהם ניהלו היה המשחק הראשון שהופעל בקזינו
7. כתוב שאילתה המחזירה פרטי שחקנים אשר סכום הכסף הכולל שיצאו איתו מהקזינו בחודש האחרון (רווחים והפסדים ביחד) הינו סכום הכסף הגדול ביותר מבין כל השחקנים
8. כתוב שאילתה המחזירה שחקנים החשודים ברמייה. שחקנים החשודים ברמייה הינם שחקנים אשר קרובי משפחה של דילרים מסויימים והם לא הפסידו באף משחק בו קרוב המשפחה שלהם היה הדילר האחראי על המשחק. על השאילתה להחזיר מספר זיהוי ושם מלא של השחקן בתוספת מספר עובד ושם מלא של הדילר
9. כתוב טריגר הבודק בעת הוספת שחקן למשחק כי כמות השחקנים שמשתתפים במשחק תואמת את הכמות המקסימלית של השחקנים המותרים במשחק. במקרה ובו השחקן גורר חריגה מכמות השחקנים המותרת, יודפס לו כי אינו יכול להשתתף במשחק והשתתפותו תבוטל
10. כתוב stored procedure המקבל מספר זיהוי של שחקן ומחזיר את סוגי המשחקים האהובים עליו מסודרים מהגדול לקטן על פי כמות הפעמים שהשתתף בכל סוג משחק. במקרה ובו לא נשלח מספר זיהוי של שחקן יש להחזיר עבור כל שחקן (מספר הזיהוי שלו) את כמות הפעמים שהשתתף בכל סוג משחק (ממויינים ראשית על פי מספר הזיהוי של השחקן ולאחר מכן על פי כמות המשחקים מהגדול לקטן)

חלק ג:

נתונה השאילתה הבאה:

```
select P.ID, P.firstName + ' ' + P.surName 'fullName',
       count(*) 'numberOfWinningGames',
       count(distinct PIAG.gameTypeName) 'numberOfDistinctGameTypes'
from Player P inner join
     PlayerInAGame PIAG on P.ID = PIAG.playerID
where PIAG.winLosAmount >= 2*PIAG.betAmount
group by P.ID, P.firstName, P.surName
```

הקזינו התחיל לנהל את מסד הנתונים שלו לפני כשנה. מאז, נרשמו 100,000 שחקנים. כמו כן ידוע כי 80% מהשחקנים הם שחקנים שבאים לקזינו רק אם הגיעו לחופשה במלון בו הקזינו נמצא. חופשה במלון הינה חופשה באורך ממוצע של חמישה ימים ובשנה האחרונה לא היה אף אורח שהגיע למלון יותר מפעם אחת. שחקנים אלו משחקים בממוצע בכל יום בחופשה בעשרה משחקים. שאר 20% השחקנים הינם שחקנים קבועים בקזינו, אשר מגיעים בממוצע שלוש פעמים בשבוע לקזינו ומשחקים בממוצע בכל פעם כ-20 משחקים.

הקזינו שלנו הינו קזינו רווחי. ב-50% מהמקרים השחקנים לא מנצחים במשחק. על פי בדיקות סטטיסטיות נמצא כי ב-50% בהם שחקן ניצח במשחק, הוא ניצח בסכום שאותו שם (כלומר לא הפסיד כסף על המשחק). ב-20% מהמקרים בהם שחקן ניצח במשחק, הוא הרוויח 50% נוספים ביחס להימור שלו. ב-10% מהמקרים בהם שחקן ניצח במשחק, הוא הרוויח 100% נוספים ביחס להימור שלו. המקרים מהמקרים בהם שחקן ניצח במשחק, הוא הרוויח מעל ל-100% נוספים ביחס להימור שלו.

1. השלימו את הטבלה הבאה המתארת את גדלי הנתונים בטבלאות המשתתפות בשאילתה.

שימו לב כי גודל כל בלוק בזכרון בדוגמא זו הינו 4,096 וכי ישנם כ-100 בלוקים פנויים

בזכרון לביצוע השאילתה

Table name	N – num of records	Record size	F – num of records in a block	B – num of blocks
Player	100,000	47		
PlayerInAGame		37		

2. חשב את עלות השאילתה ללא שימוש באינדקסים. עבור כל פתרון אפשרי יש לכתוב תכנון של

הפתרון (איך המערכת תריץ את השאילתה). יש לתאר חישוב עלות שאילתה עבור שלוש מתוך האופציות השונות שתארתם בתכנון (ובהכרח להראות שימוש -BNL, INL, סינון קודם לחיבור בין טבלאות)

3. חשב את עלות השאילתה תוך שימוש באינדקס מסוג עץ B+ בעל רמת פיצול של 80% על

התכונה PlayerID בטבלת PlayerInAGame