

**HW4, Database 1**

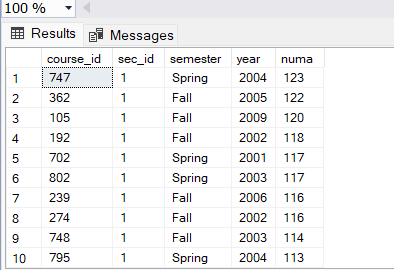
Sepehr Ebadi

9933243

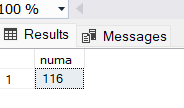
Tir, 1403

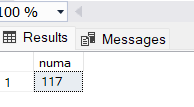
**Q1 :**

**b)**

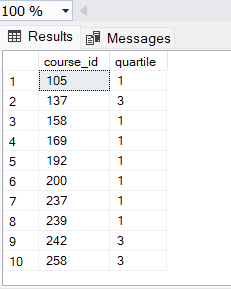


**c)**

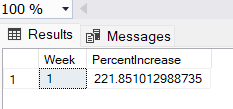




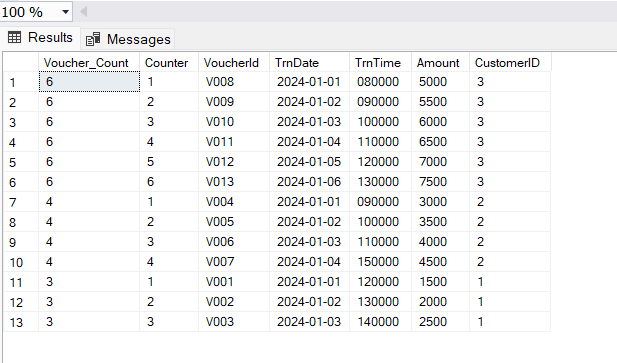
**d)**



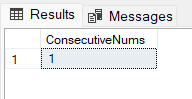
**6)**



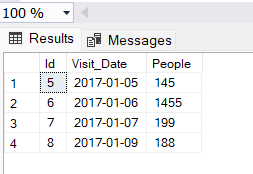
**7)**



**8)**



**9)**



**11)**

الف)

برای یافتن کلیدهای کاندید، ابتدا باید مجموعه‌ی بسته‌ها (closures) را محاسبه کنیم.

**مجموعه‌ی بسته‌ها:**

AB :

AB =

از T -> AB ، T اضافه میشود: ABT = +AB

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

A :

A = +A

از B -> A، B اضافه میشود: AB = +A

از T -> AB ، T اضافه میشود: ABT = +A

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

R :

R = +R

از C->R ، C اضافه میشود: RC = +R

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

NS :

NS = +NS

از BT->NS ، BT اضافه میشود: NSBT = +NS

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

B :

B = +B

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

N :

N = +N

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

S :

S = +S

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

T :

T = +T

دیگر ترکیبی که بتواند چیزی اضافه کند، وجود ندارد.

برای بررسی کلیدهای کاندید، باید بررسی کنیم که آیا ترکیبی از صفات می‌تواند تمامی صفات را پوشش دهد یا خیر. کلید کاندید، کلیدی است که با کوچکترین ترکیب از صفات بتواند تمام صفات رابطه را پوشش دهد.

در اینجا، هیچ کدام از تک‌صفاتی به تنهایی تمامی صفات را پوشش نمی‌دهند. باید ترکیب صفات را بررسی کنیم. از طریق آزمون و خطا مشخص می‌شود که:

ترکیب A,R,N,S میتواند تمامی صفات را پوشش دهد :

ARNSBTC -> ARNSBT -> ARNS = +(A,R,N,S)

بنابراین ترکیب A,R,N,S یک کلید کاندید است.

ب)

برای تجزیه به 3NF، ابتدا باید روابط را تجزیه کنیم:

T->AB :

رابطه جدید : R1(A,B,T)

B -> A :

R2(A,B)

C->R :

R3(R,C)

BT->NS :

R4(N,S,B,T)

به‌این‌ترتیب، روابط به صورت زیر تجزیه می‌شوند:

R1(A,B,T), R2(A,B), R3(R,C), R4(N,S,B,T)

رای بررسی 3NF بودن، باید بررسی کنیم که آیا تمامی صفات غیر کلیدی در هر رابطه به یک کاندید کلید به‌طور کامل تابعی هستند یا خیر.

ج)

برای بررسی lossless بودن:

اگر تقاطع دو رابطه حاوی یک کاندید کلید باشد، تجزیه lossless است.

تقاطع‌های موجود را بررسی می‌کنیم:

R1∩R2=AB

R1∩R3=∅

R1∩R4=∅

R2∩R3=∅

R2∩R4=B

R3∩R4=∅

همان‌طور که مشاهده می‌کنیم، رابطه‌ی AB حاوی یک کاندید کلید است، بنابراین تجزیه lossless است.

برای بررسی dependency preserving بودن:

باید ببینیم که آیا تمامی وابستگی‌ها در تجزیه حفظ می‌شوند یا خیر:

AB→T درR1

A→B درR2

R→C درR3

NS→BT درR4

بنابراین، تمامی وابستگی‌ها حفظ شده‌اند و تجزیه dependency preserving است.

د)

برای تجزیه به BCNF، باید هر رابطه را بررسی کنیم و اگر وابستگی جزئی وجود دارد، تجزیه کنیم:

**رابطه‌ی R1(A,B,T)**:

A→B جزئی است، بنابراین تجزیه می‌کنیم:

R1a(A,B)

R1b(A,T)

**رابطه‌ی R2(A,B)**:

این رابطه قبلاً به صورت BCNF است.

**رابطه‌ی R3(R,C)**:

این رابطه قبلاً به صورت BCNF است.

**رابطه‌ی R4(N,S,B,T)**:

NS→BT کلید کاندید است، بنابراین این رابطه نیز به صورت BCNF است.

در نتیجه:

R1a(A,B),R1b(A,T),R2(A,B),R3(R,C),R4(N,S,B,T)

برای بررسی dependency preserving بودن:

AB→T درR1b

A→B درR1a

R→C درR3

NS→BT درR4

بنابراین، تمامی وابستگی‌ها حفظ شده‌اند و تجزیه dependency preserving است.

**12)**

الف(

اگر یک ویژگی (attribute) به ویژگی دیگری وابسته باشد و هیچ یک از ویژگی‌ها کلید اصلی (Primary Key) نباشند، این وضعیت به عنوان وابستگی انتقالی (Transitive Dependency) شناخته می‌شود. در این حالت، یک ویژگی غیر کلیدی به یک ویژگی غیر کلیدی دیگر وابسته است.

ب(

جدولی که کلید اصلی آن تنها شامل یک ویژگی (attribute) است، اگر در 1NF (اولین فرم نرمال) باشد، خود به خود در 2NF (دومین فرم نرمال) هم خواهد بود. این به این دلیل است که:

در 1NF، تمام ستون‌های جدول تک مقداری (atomic) هستند.

در 2NF، علاوه بر 1NF، تمامی ویژگی‌های غیر کلیدی باید به کل کلید اصلی (که در این حالت تک attribute است) به صورت کامل وابسته باشند. چون کلید اصلی تک attribute است، هیچ وابستگی جزئی (Partial Dependency) وجود ندارد. بنابراین، جدول به طور خودکار در 2NF قرار می‌گیرد.

**13)**

برای پیدا کردن مجموعه کاهش ناپذیر معادل مجموعه F از توابع تبعی F، باید سه مرحله اصلی را طی کنیم: حذف موارد غیر ضروری از سمت راست، حذف موارد غیر ضروری از سمت چپ و حذف وابستگی‌های تابعی اضافی. این مراحل به صورت زیر انجام می‌شود:

مجموعه وابستگی‌های تبعی اولیه F :

F={A→BC,B→C,A→B,AB→C,AC→D}

مرحله 1: حذف موارد غیر ضروری از سمت راست

برای هر وابستگی تابعی X→Y در F:

ررسی کنید که آیا می‌توانید هر عضو از Y را بدون تغییر محتوای مجموعه پوشش داده شده حذف کنید.

**بررسی A→BC :**

آیا A→B می‌تواند به تنهایی A→BC را پوشش دهد؟

بله، زیرا A→B و B→C داریم.

بنابراین A→C را می‌توان از A→BC حذف کرد.

نتیجه: A→B

Top of FormBottom of Form**بررسی B→C :**

نمی‌توان هیچ چیزی را از سمت راست حذف کرد زیرا تنها یک عضو دارد.

نتیجه: B→C

**بررسی A→B :**

نمی‌توان هیچ چیزی را از سمت راست حذف کرد زیرا تنها یک عضو دارد.

نتیجه: A→B

**بررسی AB→C :**

نمی‌توان هیچ چیزی را از سمت راست حذف کرد زیرا تنها یک عضو دارد.

نتیجه : AB→C

**بررسی AC→D :**

نمی‌توان هیچ چیزی را از سمت راست حذف کرد زیرا تنها یک عضو دارد.

نتیجه: AC→D

مرحله 2: حذف موارد غیر ضروری از سمت چپ

برای هر وابستگی تابعی X→Y در F :

بررسی کنید که آیا می‌توانید هر عضو از X را بدون تغییر محتوای مجموعه پوشش داده شده حذف کنید.

**بررسی A→B :**

نمی‌توان هیچ چیزی را از سمت چپ حذف کرد زیرا تنها یک عضو دارد.

نتیجه: A→B

**بررسی B→C :**

نمی‌توان هیچ چیزی را از سمت چپ حذف کرد زیرا تنها یک عضو دارد.

نتیجه: B→C

**بررسی AB→C :**

آیا A به تنهایی AB→C را پوشش می‌دهد؟

نه، زیرا A→B داریم، اما B→C نیاز است.

نتیجه: AB→C

**بررسی AC→D :**

آیا A به تنهایی AC→D را پوشش می‌دهد؟

نه، زیرا A→C داریم، اما AC→D نیاز است.

نتیجه: AC→D

مرحله 3: حذف وابستگی‌های تابعی اضافی

برای هر وابستگی تابعی X→Y در F :

بررسی کنید که آیا می‌توانید آن وابستگی تابعی را حذف کنید بدون اینکه محتوای مجموعه پوشش داده شده تغییر کند.

**بررسی A→B :**

آیا می‌توانیم A→B را حذف کنیم؟

نه، زیرا A→B تنها وابستگی است که B را تعیین می‌کند.

نتیجه: A→B

**بررسی B→C :**

آیا می‌توانیم B→C را حذف کنیم؟

نه، زیرا B→C تنها وابستگی است که C را تعیین می‌کند.

نتیجه: B→C

**بررسی AB→C :**

آیا می‌توانیم AB→C را حذف کنیم؟

بله، زیرا با توجه به A→B و B→C ، می‌توانیم AB→C را استنتاج کنیم.

نتیجه: حذف AB→C

**بررسی AC→D :**

آیا می‌توانیم AC→D را حذف کنیم؟

نه، زیرا AC→D تنها وابستگی است که D را تعیین می‌کند.

نتیجه: AC→D

مجموعه کاهش ناپذیر معادل F :

در نهایت، مجموعه کاهش ناپذیر معادل F به صورت زیر است:

F′={A→B,B→C,AC→D}