

## به نام خدا

تکلیف اول دیتابیس

حدیث غفوری 9825413

سوال 1

1.a چرا از DBMS استفاده میکنیم؟

1. Data security.

DBMS به سازمانها اجازه می دهد تا سیاست هایی را اجرا کنند که انطباق و امنیت را فعال می کند. DBMS همچنین وظیفه حفظ عملکرد مطلوب عملیات query را در حین اطمینان از اعتبار ، امنیت و سازگاری داده هایی که در پایگاه داده به روز شده است ، دارد.

2. Data sharing.

همکاری سریع و موثر بین کاربران

3. Data access and auditing.

دسترسی کنترل شده به پایگاه داده و ثبت فعالیت ها هنگام دسترسی باعث افزایش امنیت سازمان ها میشود.

4. Data integration.

به جای استفاده از منابع زیادی از پایگاه داده از یک رابط واحد برای مدیریت پایگاه های داده با روابط منطقی و فیزیکی استفاده میشود.

5. Abstraction and independence.

سازمانها می توانند طرح فیزیکی سیستمهای پایگاه داده را بدون نیاز به تغییر در طرح منطقی حاکم بر روابط پایگاه داده تغییر دهند. در نتیجه ، سازمان ها می توانند فضای ذخیره سازی را ارتقا دهند و زیرساخت ها را بدون تأثیر بر عملیات پایگاه داده ، ارتقا دهند. به طور مشابه ، تغییرات در طرح منطقی را می توان بدون تغییر برنامه ها و سرویس هایی که به پایگاه های داده دسترسی دارند ، اعمال کرد.

معایب ذخیره سازی داده ها در فایل (بدون DBMS)

1. data redundancy and consistency

مثلا اینکه یک اسم چندین بار با فرمت های متفاوتی ذخیره شود.

2. Difficulty in accessing data

اگر یک DBMS نداشته باشیم برای دسترسی به دیتاها باید خودمون پروگرام بسازیم.

3. Data isolation

اگر فایل های مختلف با فرمت های مختلف ذخیره شوند مشکل پردازش پیش میاد.

4. Integrity problems

در سیستم های بزرگ مثل بانک ها همیشه دیتاها را دستی کنترل کرد.

همچنین وجود یک سری قیدها مثل منفی نشدن حساب کاربران و .. هم وجود دارد.

چک کردن منفی نشدن حساب را میتوانیم به کمک DBMS انجام دهیم.

#### 5. Atomicity of updates

بحث عملیات ها و تراکنش ها مثل انتقال پول از یک حساب به حساب دیگر

ما از یک DBMS انتظار داریم که اگر مشکلی مانند قطعی برق و ... برای کاربران پیش آمد دیتاها را به جای اول برگردانند و این کار را با گرفتن snapshot و رصد کردن اتفاقات انجام میدهند.

#### 6. Concurrent access by multiple users

دیتابیس باید توانایی این را داشته باشد که چندین کاربر بتوانند دسترسی موردنیاز همزمان به بخش های مختلف دیتابیس را داشته باشند.

#### 7. Security problems

مسایل امنیت و دسترسی به داده ها

یک DBMS باید کمک کند تا از دیتاهای کاربران محافظت کنیم. همچنین به تعیین دسترسی ها هم کمک میکند.

#### (1.b)

1-کاربر با استفاده از بعضی زیر زبان داده ها مانند (SQL) در خواست يك دسترسی می نماید.

2-DBMS در خواست مورد نظر را دریافت و آن را تحلیل می کند.

3-DBMS بترتیب دید کاربر از داده های ذخیره شده و حفظ امنیت داده در محیط بانک اطلاعاتی را بازرسی و مرور می کند. (کنترل امنیت و جامعیت داده ها )

4-DBMS عملکردهای لازم را برای داده های ذخیره شده اجرا می کند.

#### (1.c)

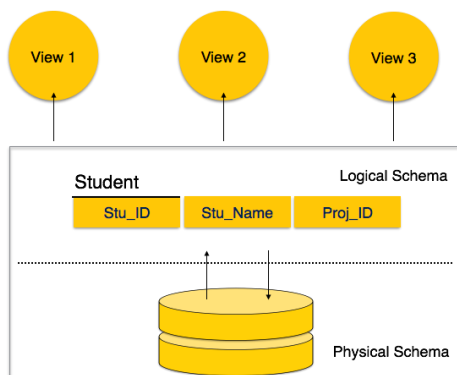
دو بخش اصلی DBMS :

#### Physical Database Schema

این طرح به ذخیره واقعی داده ها و نوع ذخیره سازی آن مانند فایل ها ، فهرست ها و ... مربوط می شود. نحوه ذخیره سازی داده ها در یک حافظه ثانویه را مشخص می کند.

#### Logical Database Schema

این طرح همه محدودیت های منطقی را که باید بر روی داده های ذخیره شده اعمال شود ، تعریف می کند. جداول ، نماها و محدودیت های یکپارچگی را تعریف می کند.

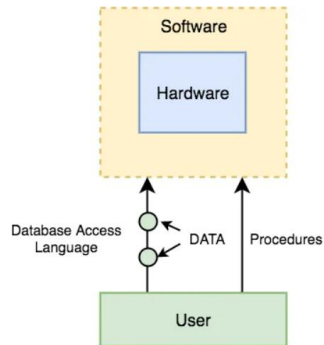


اجزای عملکردی یک سیستم پایگاه داده را می توان به موارد زیر تقسیم کرد:

The storage manager, The query processor component, The transaction management component.

اجزای تشکیل دهنده ی DBMS:

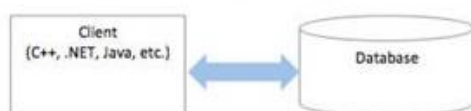
Hardware , Software,Data,Procedures,Database Access Language



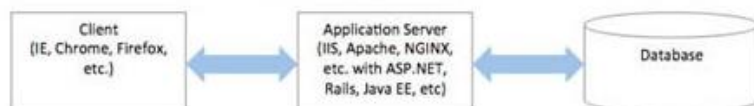
## سوال 2

ویژگی ها	معماری دو لایه	معماری سه لایه
سرعت	سرعت کمتری دارد	سرعت بیشتری دارد.
امنیت	امنیت کمتری دارد زیرا کاربر می تواند مستقیماً با پایگاه داده ارتباط برقرار کند.	امنیت بیشتری دارد. زیرا مشتری مجاز به ارتباط مستقیم با پایگاه داده نیست.
افزونگی	هر زمان که کاربران به سرعت افزایش می یابند ، باعث کاهش عملکرد می شود.	بیشتر است.
مقیاس پذیری	نسبت به معماری سه لایه ضعیف تر است.	بهتر است . به دلیل استقرار توزیع شده سرورهای برنامه در حال حاضر ، نیازی به ارتباطات جداگانه بین سرورس گیرنده و سرور نیست.
انعطاف پذیری	انعطاف پذیری کمتری دارد	انعطاف بیشتری دارد. آنها می توانند داده ها را از منابع مختلف ادغام کنند.
یک پارچگی		از آنجا که یک لایه میانی بین سرورس گیرنده و سرور وجود دارد ، می توان از خرابی اطلاعات جلوگیری یا حذف کرد.

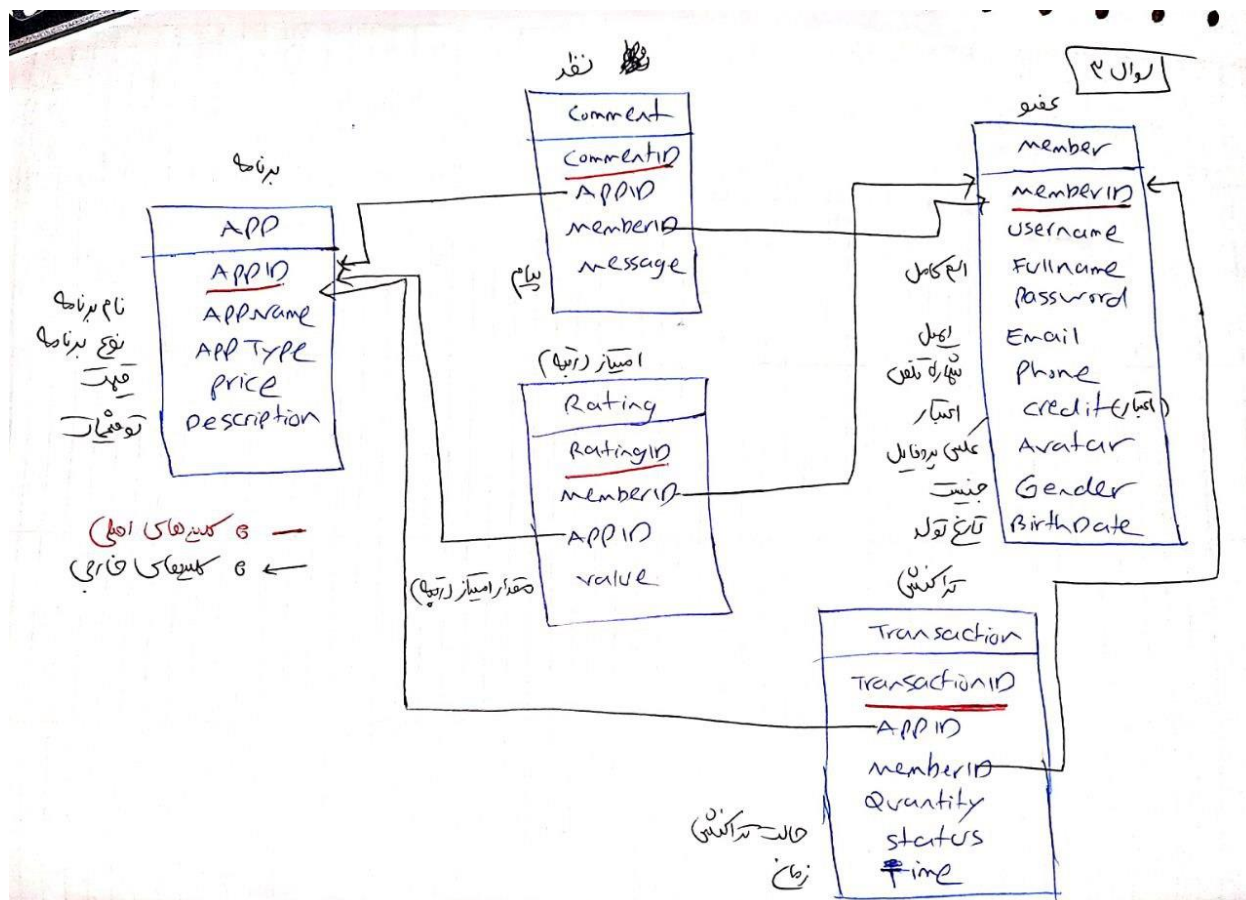
### 2-Tiered Architecture



### 3-Tiered Architecture



### سوال 3



### سوال 4

(4.a)

مقدار NULL یک نشانگر ویژه است که در SQL استفاده می شود تا نشان دهد که مقدار داده در پایگاه داده وجود ندارد. به عبارت دیگر، نشان دادن مقداری که گم شده اند یا ما آنها را نمی شناسیم، فقط یک مکان نگهدارنده است.

هرگز نباید مقادیر NULL را برای صفرها یا رشته های خالی اشتباه بگیرید. این یک تصور غلط دیگر است. وقتی یک مقدار NULL را در یک جدول می بینید، به این معنی است که هنوز مقدار آن را نمی دانید. وقتی یک مقدار صفر می بینید، بدین معنی است که مقدار کل یا عدد صفر صفر است. می دانید که مقدار باید صفر باشد. NULL را "ناشناخته" در نظر بگیرید.

می توانید از مقادیر NULL برای هر نوع داده از جمله اعداد صحیح، اعشار، رشته ها یا Boolean ها استفاده کنید. اگرچه بسیاری از مدیران پایگاه داده از NULL استفاده می کنند، اما معمولاً از NULL ها برای مقادیر عددی استفاده می کنند. دلیل این است که NULL هایی که برای مقادیر عددی استفاده می شوند هنگام ایجاد کد برای محاسبه داده ها گیج کننده می شوند.

پایگاه های داده به شما امکان می دهد NULL را به صورت خودکار در زمانی که هیچ مقداری در دسترس نیست وارد کنید. این بدان معناست که نیازی به تعیین مقدار برای هر نگهدارنده مکان ندارید. شما فقط از NULL استفاده می کنید و هر برنامه نویس پایگاه داده آن را به عنوان مکان یاب پیش فرض تشخیص می دهد.

اگر از مقادیر صفر استفاده می کردید ، برنامه های شما تفاوت بین ورود کاربر به صفر یا پایگاه داده ای که صفر را به عنوان مکان ذخیره ذخیره می کرد ، نمی دانستند. این یکی از دلایلی است که طرفداران ارزش پایگاه داده NULL آن را ترجیح می دهند.

(4.b)

میشود از شیوه ای که جدول های student , takes یا instructor , teaches دارند استفاده کرد یعنی به جای اینکه بخواهیم جزییات را وارد جدول استادها کنیم میتوانیم یک جدول دیگه تعریف کنیم (مثلا works) که در انجا اسم دانشکده هایی که استاد مدنظر (Instructor\_id) در انها تدریس میکند جداگانه آورده شود.

مثل جدول زیر

Instructor_id	Dept_name
123	Physics
123	Computer

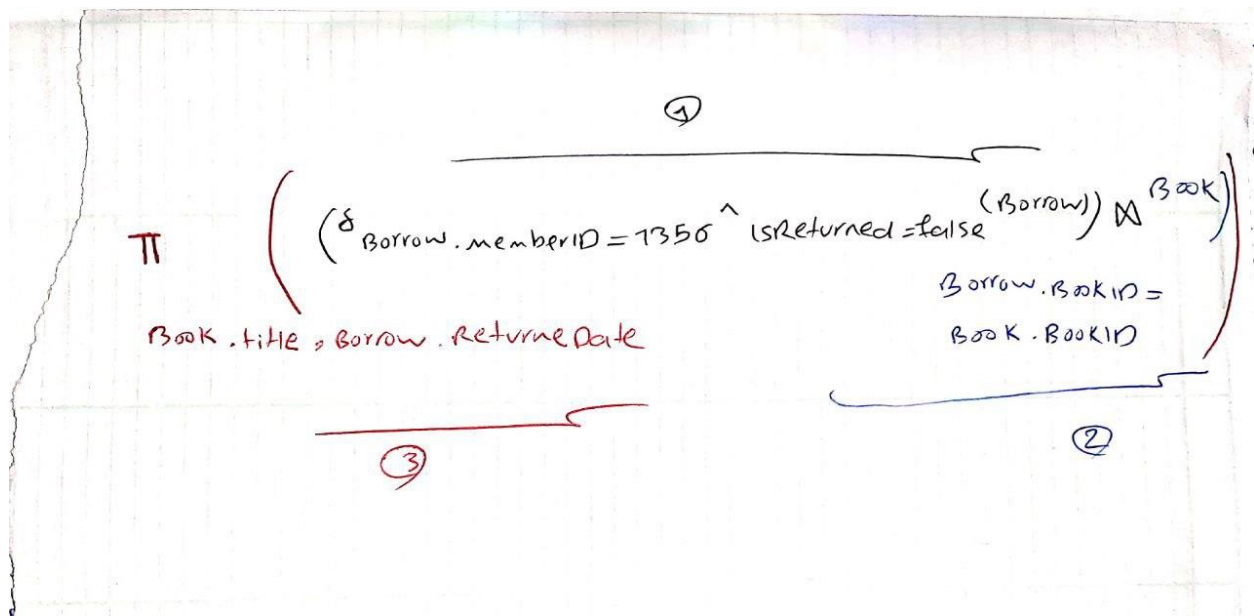
(4.c)

میتوانیم به جای جدول advisor دو ستون جدید برای هر دانشجویی به جدول student اضافه کنیم با نام های advisor1,advisor2 و داخل انها ایدی استاد راهنما را قرار دهیم.

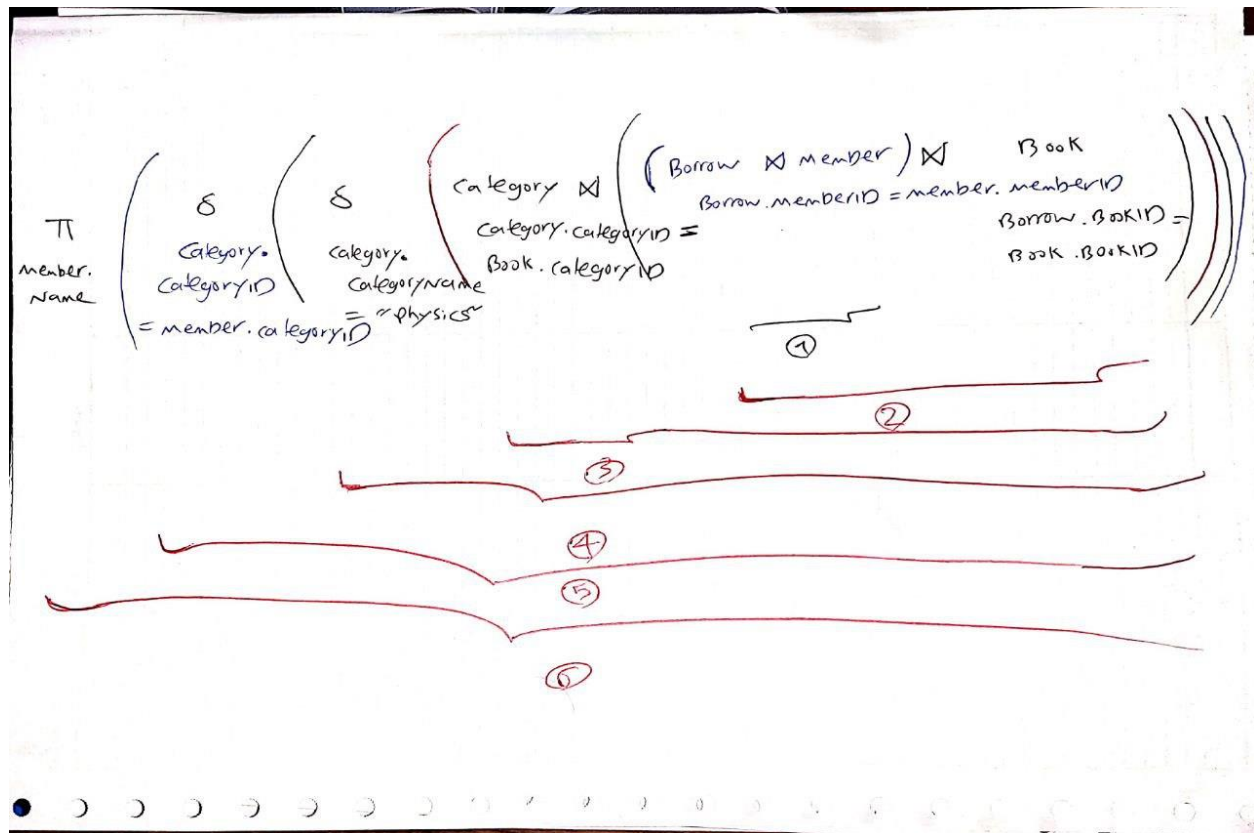
اگر هم دانشجو فقط یک استاد راهنما دارد و ممکن است استاد راهنمای دوم هم در آینده داشته باشد خانه ی advisor2 را null میگذاریم.

(4.d)

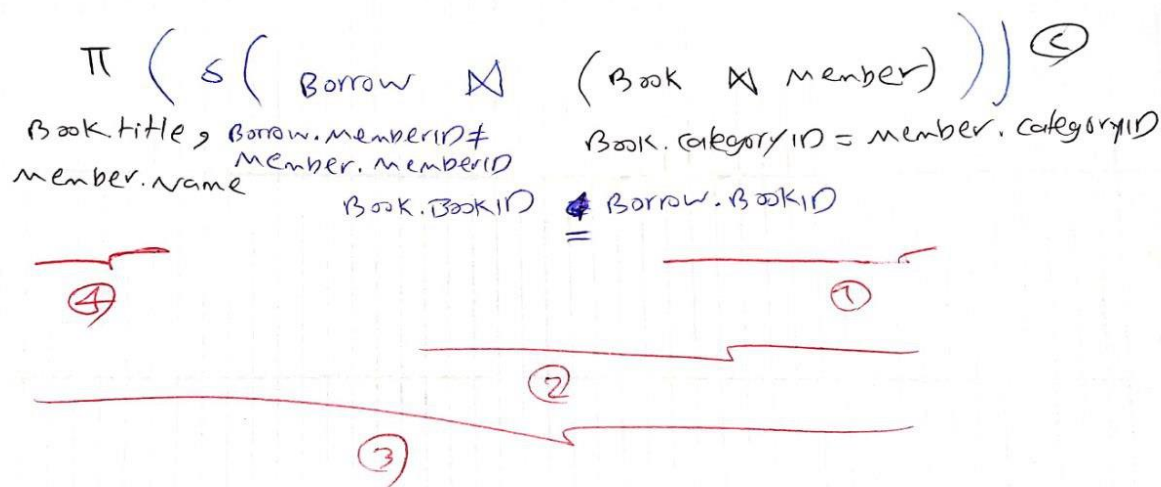
(a)



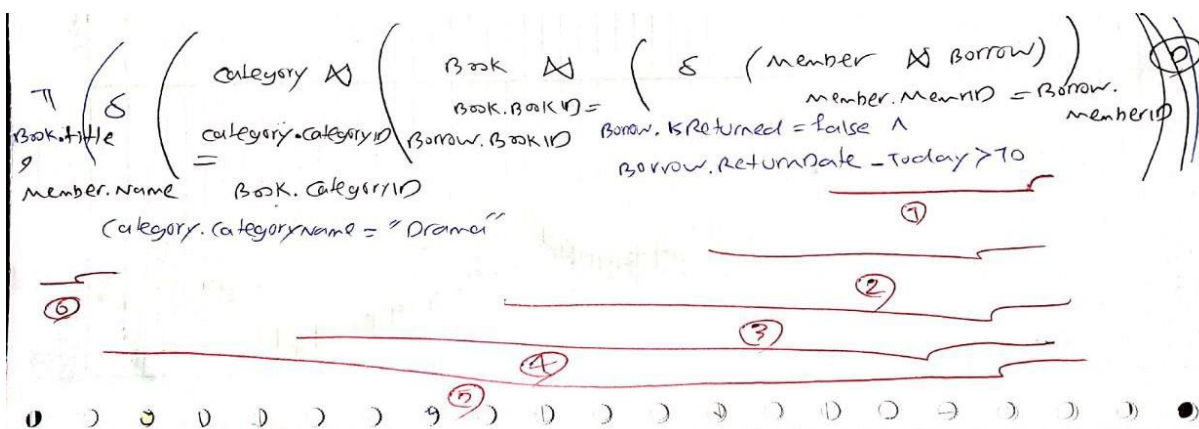
(b)



(c)



(d)



(e)

۱. پیدا کردن مشخصات کتاب هایی که به امانت گرفته شده اند به کمک جدول book و سپس پیدا کردن مشخصات افرادی که ان کتاب ها را به امانت گرفته اند.

۲. سپس مشخص کردن کتاب هایی که به امانت گرفته شده اند و هزینه ی دیرکرد آنها (در حالت کل که از ضرب تعداد روز های دیرکرد و هزینه ی هرروز به دست آمده) از ۱۰۰۰۰۰ بزرگتر یا مساوی شده است

۳. در نهایت که کتاب ها با ویژگی ۲ مشخص شدند، اسم کتاب و اسم شخصی که کتاب را به امانت برده را مشخص کن.



پس در نهایت = پیدا کردن اسم کتاب و اسم شخصی که کتاب هایی را به امانت گرفته که جریمه ی کلی آن بزرگتر مساوی ۱۰۰۰۰۰ شده است.

(f)

۱. پیدا کردن نویسنده ی هر کتاب و سپس پیدا کردن category هر کتاب و سپس انتخاب category با نام philosophy و انتخاب کتاب ها به جز نویسنده plato در نهایت انتخاب اسم کتاب ها

۲. مشخص کردن کتاب هایی که هنوز پس داده نشده اند و سپس پیدا کردن مشخصات تک تک آنها از جدول book و انتخاب ستون title از جواب نهایی => پیدا کردن اسم کتاب هایی که امانت گرفته شدن اما هنوز پس داده نشده اند.

در نهایت اشتراک 1 و 2 را میگیریم.

پس در نهایت = پیدا کردن اسم کتاب هایی که اسم نویسنده آنها plato نباشد و جزو Category با نام philosophy باشد و کتاب به امانت گرفته شده است و هنوز برگردانده نشده است.