

به نام خدا
پاسخنامه تکلیف اول درس پایگاه داده ها ۱
ترم مهر ۱۴۰۰

۱. به سوالات زیر پاسخ دهید.

a. چرا برای عملیات پایگاه داده در یک سامانه نرم افزاری باید بیشتر کارهای مربوط به داده ها به کمک DBMS انجام شود؟ همچنین توضیح دهید چرا نمی شود همه ی کارها را با فایل های معمولی و داخل برنامه به جای پایگاه داده انجام داد؟

پایگاه داده ها سه مزیت مهم نسبت به سیستم های ذخیره داده ساده تر مانند فایل دارد.
که عبارتند از کیفیت بهتر در دسترسی، یکپارچگی و امنیت داده ها

۱- دسترسی : به معنای دسترسی کاربران به داده های موجود است.
در پایگاه داده ها حجم زیادی از داده ها در یک فضا ذخیره میشود و چند کاربر همزمان می توانند داده ها را بخوانند و آن ها را تغییر دهند. سیستم های پایگاه داده قابلیت sort شدن و جستجو را دارند و داده ی مورد نظر به راحتی و با سرعت پیدا میشود.
اگر نیازهای ما تغییر کرد، تغییر ساختار پایگاه داده نسبت به فایل آسانتر است.

۲- یکپارچگی: استفاده از سیستم های پایگاه داده به ما این اطمینان را می دهد که داده ی ما درست است یا به عبارتی یکپارچگی دارد.
برای اطمینان از یکپارچگی یک پایگاه داده، هر تغییر یا تراکنش باید از مجموعه ی قوانینی پیروی کنند که ما تحت عنوان ACID، این قوانین را می شناسیم. این ویژگی ها عبارتند از:
اتمیک بودن : به معنای اینکه یک تراکنش یا به صورت کامل اجرا می شود و یا اصلا اجرا نمیشود. از این دو حالت خارج نیست و اگر تراکنش در زمان اجرا بنا به دلایلی مانند همزمانی با یک تراکنش ناسازگار و یا غیرمجاز بودن تراکنش و یا تمام شدن زمان مشخص شده، شکست بخورد باید rollback شود و به حالت قبل از تراکنش بازگردد.

سازگاری داده ها : هر تغییر قبل از انجام باید اعتبارسنجی شود.
ایزوله سازی: پایگاه داده ها اجازه ی تغییرات همزمان را می دهد. هر تغییر باید نسبت به بقیه ایزوله باشد.
پایداری : هر تغییر در پایگاه داده ها ماندگار است .
همچنین پایگاه داده ها، مکانیزم هایی دارند که توزیع و پشتیبان گیری و افزونگی داده ها را به عهده دارند به این ترتیب تا حدی اطمینان حاصل می شود که داده ی ما از بین نمی رود.

۳- امنیت: اگرچه دسترسی به فایل های متنی و صفحات گسترده می تواند ایمن باشد، اما زمانی که یک نفر به فایل دسترسی پیدا کرد، به کل داده های فایل دسترسی پیدا میکند. اما پایگاه داده ها این امکان را دارد که به هر کاربر حق دسترسی های متفاوت به قسمت های مختلف بدهد. همچنین این حقوق می تواند متفاوت

باشد مثلاً حق خواندن به یک کاربر داده شود ولی حق نوشتن به او داده نشود. و یا مثلاً یک کاربر تنها بتواند اسم کارمندان را ببیند و حقوق آن ها از دید او مخفی باشد.

b. DBMS چگونه به درخواست کاربر پاسخ می دهد؟ توضیح با رعایت ترتیب عملیات.

یکی از وظایف مهم DBMS پردازش Query هاست. کامپایلر کوئری را parse یا پیمایش میکند. در این قسمت مفسر DDL، دستورات DDL را تفسیر میکنند که تعریف آن ها در دیکشنری وجود دارد و کامپایلر DML، دستورات DML را به دستورات سطح پایین که برای موتور ارزیابی کوئری قابل فهم است تبدیل میکند و کوئری به یک عبارت جبر رابطه ای تبدیل می شود سپس plan های مختلف اجرای کوئری به دست می آید (با توجه به عواملی مانند اندیس گذاری و یا اندازه ی جدول ها، اجرای قسمت های یک کوئری با ترتیب های متفاوت، در زمان اجرا تفاوت ایجاد میکند). سپس بهینه ساز کوئری با توجه به آماری که از اجراهای پیشین دارد و سباز جدول ها، execution plan ای را که پیش بینی کرده کمترین هزینه را دارد انتخاب میکند. سپس موتور ارزیابی کوئری که دستورات سطح پایین کامپایل شده را اجرا میکند به سراغ داده ها می رود و کوئری را اجرا میکند و یک خروجی به ما می دهد.

c. با یک طرح، دو بخش اصلی DBMS و محیط پایگاه داده ها را نشان دهید و اجزای تشکیل دهنده DBMS را نام ببرید.

شکل صفحه ی ۲۲ کتاب

اجزای تشکیل دهنده DBMS :

مدیر انبار داده:

مدیر احراز هویت و درستی (Authorization and integrity manager)

مدیر تراکنش ها (Transaction manager)

مدیر فایل (File manager)

مدیر بافر (Buffer manager)

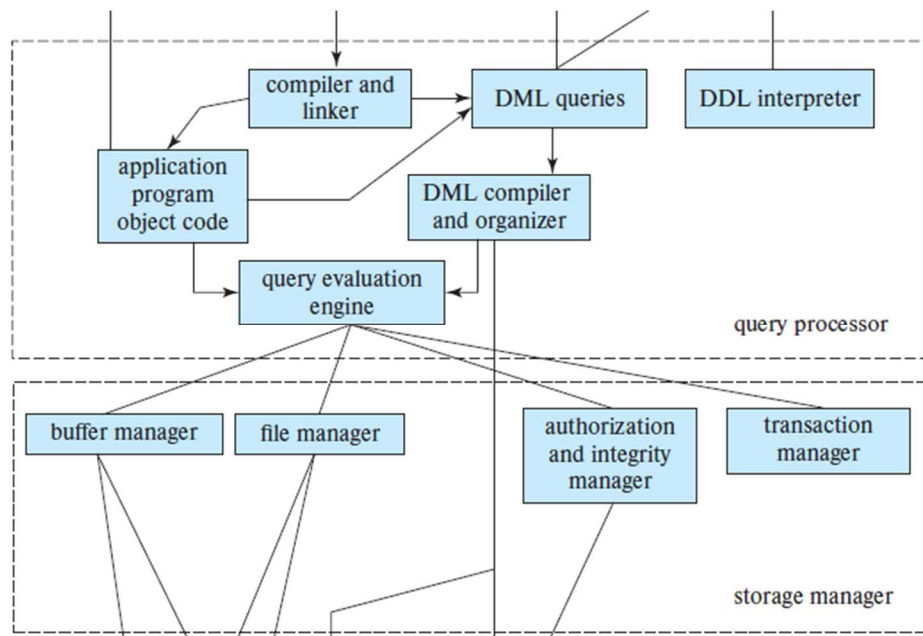
پردازنده پرس و جو (The Query Processor)

DDL interpreter

DML compiler

DML queries

Query evaluation engine



۲. دو طراحی برای یک نمونه سامانه کاربردی مبتنی پایگاه ارائه دهید (به صورت ترسیم ساختار)، یکی دو لایه (Two-Tier) و یکی سه لایه (Three-Tier). در مورد مزایا و معایب آنها در هر یک از موارد سرعت، امنیت، افزونگی، مقیاس پذیری، انعطاف پذیری و یکپارچگی در جدول بنویسید و مقایسه کنید. در صورت نیاز می توانید از جستجوی اینترنتی در این زمینه استفاده کنید.

توضیحات ساختار طراحی دولایه و سه لایه در صفحه ۲۳ کتاب آمده است. طراحی های متفاوتی برای هر دو معماری انجام شده است که با جستجو در اینترنت می توان مدل های متفاوتی از هر دو را دید. مثال معماری دو لایه می تواند یک سیستم مدیریت دسترسی مانند MS-Access باشد و مثال معماری سه لایه میتواند وبسایت های بزرگ باشد.

سرعت: معیار سرعت در معماری دو لایه بهتر است به دلیل اینکه واسطه های ارتباطی و مازول ها کمتر است. در نتیجه پیام هایی که رد و بدل می شود نیز کمتر است.

امنیت: سه لایه بهتر است به دلیل اینکه مدیریت داده ها دقیق تر و شفافیت انجام می شود. همچنین سطح دسترسی ها راحت تر مدیریت می شود و جریان داده قابل کنترل تر است.

افزونگی: در سه لایه کمتر است زیرا متولی مدیریت داده مشخص و تعریف شده است و از چند باره کاری جلوگیری می کند.

مقیاس پذیری: در سه لایه بهتر و شفاف تر است. به دلیل مازولاریتی بالاتر معماری سه لایه و ساده تر بودن مازول ها از نظر قابلیت فهم و درک

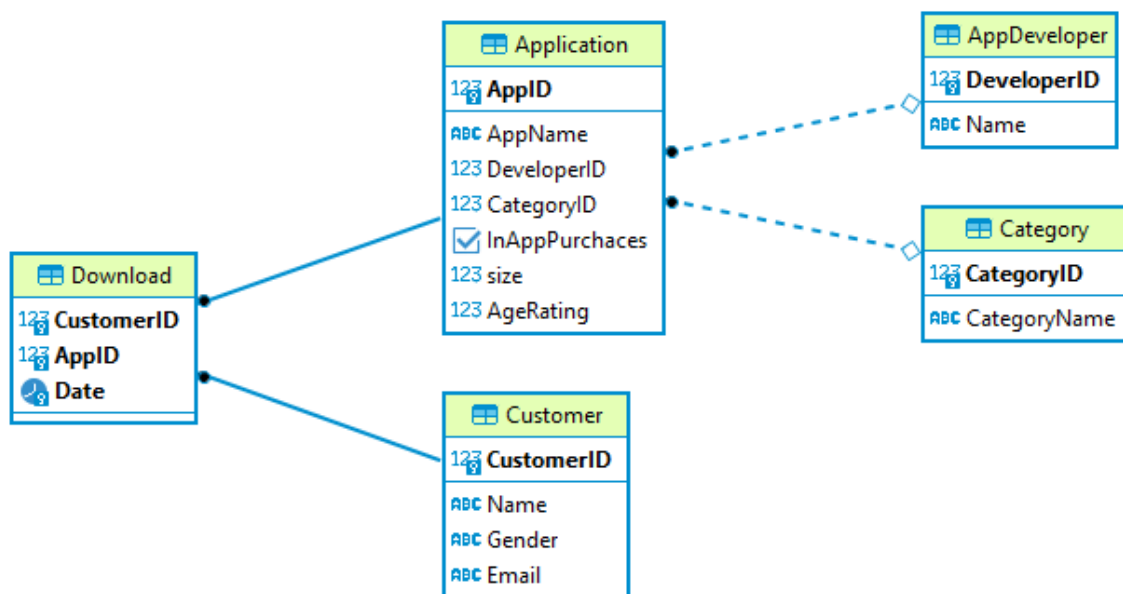
انعطاف پذیری: سه لایه بیشتر است به دلیل مازولاریتی بالاتر و توسعه پذیری بیشتر این معماری و استقلال بیشتر لایه ها و ارتباط مدیریت شده بین لایه ها.

یکپارچگی: امکان ایجاد یکپارچگی در طراحی مبتنی بر معماری سه لایه بسیار بیشتر است. چون لایه ی میانی باعث میشود قبل از به روز رسانی پایگاه داده، داده ی ما اعتبارسنجی شود و باعث ناسازگاری داده ها نشود.

۳. یک فروشگاه آنلاین نرم افزار مانند کافه بازار یا گوگل پلی را در نظر بگیرید. حداقل پنج جدول در نظر بگیرید که اطلاعات این سیستم را ذخیره میکند و به سوالات زیر پاسخ دهید.
a. هر جدول شامل چه اطلاعاتی است و چه خصوصیتی را ذخیره میکند؟

Application(AppID , AppName, DeveloperID, CategoryID, InAppPurchases,Size, AgeRating)
AppDeveloper(DeveloperID, Name)
Customer(CustomerID, Name,Gender, Email)
Download(CostumerID, AppID, Date)
Category(CategoryID , CategoryName)

b. این جداول با هم چه ارتباطی دارند؟ (کلیدهای اصلی و خارجی را مشخص کنید).



در قسمت قبل زیر کلیدهای اصلی با رنگ قرمز مشخص شده اند.

در جدول application:

DeveloperID کلید خارجی است که به کلید اصلی جدول AppDeveloper اشاره میکند.

CategoryID کلید خارجی است که به کلید اصلی جدول Category اشاره می کند.

در جدول download:

CustomerID کلید خارجی است که به کلید اصلی جدول Customer اشاره میکند.

AppID کلید خارجی است که به کلید اصلی جدول Application اشاره میکند.

۴. پایگاه داده نمونه دانشگاه را در نظر بگیرید و به سوالات زیر پاسخ دهید.

a. چه امکانی از پایگاه داده کمک میکند که بتوانیم مشخص کنیم که نمره یک دانشجو هنوز وارد نشده است؟ در رابطه با اهمیت وجود چنین امکانی توضیح دهید.

یکی از مزیت های پایگاه داده این است که به صورت خودکار، برای فیلدی که مقدار ندارد، 'NULL' قرار می دهد. این امکان باعث می شود که ما مجبور نباشیم مقدار هر فیلد را از همان اول مشخص کنیم و Null مقدار نیست که همه ی برنامه نویسان به عنوان پیش فرض برای فیلدی که مقدار ندارد در نظر میگیرند.

b. فرض کنید که استادی وجود دارد که همزمان در دو دانشکده مشغول به کار هست برای ذخیره چنین داده ای چه تغییری باید در پایگاه داده دهیم؟

در طراحی در کنار جدول instructor، جدول Instructor_Department را قرار میدهم و با استفاده از کلید خارجی دو جدول به هم متصل می شوند.

`instructor_Department(instructor_id, Dept_name)`
instructor_id به کلید اصلی جدول instructor اشاره می کند. Dept_name به کلید اصلی جدول دپارتمان اشاره میکند. همچنین فیلد dept_name را از جدول instructor حذف میکنیم.

c. در صورتی که یک دانشجو بتواند حداکثر دو استاد راهنما داشته باشد، پایگاه داده ی ما چه تغییری باید کند؟

راه حل اول:

در پایگاه نمونه دانشگاه میبینیم که در جدول advisor فقط شناسه دانشجو جز کلید اصلی است و طبق خاصیت کلید اصلی، هر دانشجو فقط یکبار میتواند در این جدول ثبت شود چون باید رکورد ها یکتا باشند. پس برای ذخیره چند استاد راهنما کلید جدول advisor باید به مجموعه ی شناسه ی استاد و شناسه ی دانشجو تغییر کند. به این ترتیب هر رکورد با شناسه دانشجو و شناسه استاد یکتا می شود. برای رعایت قید حداکثر، معمولا در لایه ی کاربرد و در کد این قید لحاظ می شود.

راه حل دوم:

جدول advisor باید ستون هایی با عنوان های شناسه های استاد راهنمای اول و دوم و شناسه ی دانشجو داشته باشد. به این ترتیب در طراحی هم قید حداکثر دو دانشجو لحاظ می شود. این راه افزونگی داده دارد و در مباحث بعدی درس با ابزارهایی آشنا می شوید و می توانید به وسیله ی آنها قیود و محدودیت ها را کنترل کنید.

d. با توجه به وجود کلید خارجی از جدول Takes به Student مثالی از عملیات delete و insert در این جداول بنویسید که باعث نقض این کلید خارجی شود.

مثلاً در جدول takes ، ثبت شده است که علی درس پایگاه داده را گرفته است.

(00342, Ali, comp.Sci , 120)

اگر علی از بین دانشجویان حذف شود در حالی که شناسه ی دانشجویی او در جدول takes ثبت شده است باعث می شود به شناسه ای ارجاع دهیم که وجود ندارد و کلید خارجی نقض شود. اگر در هنگام درج یک رکورد در جدول takes ، شناسه ی دانشجویی را درج کنیم که در جدول دانشجوی وجود ندارد، سیستم پایگاه داده خطا می دهد. چون کلید خارجی نقض شده است.

(00530, CS-110, 10, Fall, A)

۵. در پایگاه داده ی زیر شناسه کتاب BookID است. شناسه نویسنده ی کتاب AuthorID است. نام کتاب Title است. موضوع مورد علاقه ی عضو CategoryID است که در جدول Category ثبت شده است. جریمه ی دیرکرد کتاب (مثلاً به تومان) به ازای هر روز Penalty عددی (numeric) است.

تاریخ مجاز برگشت کتاب امانت گرفته شده به میلادی ReturnDate است و به صورت YYYY-MM-DD نشان داده می شود که موقع امانت کتاب ثبت می گردد. همچنین RegistrationDate تاریخ ثبت نام کاربر از نوع date است. برگردانده شدن یا برگردانده نشدن کتاب امانت گرفته شده در فیلد IsReturned از نوع boolean است و تعداد روزهای دیرکرد NumDays است که عددی صحیح بوده و پس از تحویل کتاب ثبت می شود.

Book (BookID, Title, CategoryID, Penalty, AuthorID)

Author (AuthorID, Name, Nationality)

Member (MemberID, Name, CategoryID, IsActive, RegistrationDate)

Borrow (BookID, MemberID, ReturnDate, NumDays, IsReturned)

Category (CategoryID, CategoryName, Description)

a. نام کتاب هایی که عضوی با شناسه ی عضویت ۱۳۵۶ به امانت گرفته ولی هنوز به کتابخانه تحویل نداده است را همراه با تاریخ بازگشت آنها بدست آورید.

$\prod Book.Title, Borrow.ReturnDate$

$(\sigma_{Member.ID=1356 \wedge IsReturned=False} (Borrow)) \bowtie_{Borrow.BookID=Book.BookID} Book$

b. نام اعضای را به دست آورید که در موضوع Physics کتاب امانت گرفته اند و به موضوع Physics نیز علاقه مند هستند.

$\pi_{Member.Name}(((\sigma_{Category.CategoryName="Physics"} Category) \bowtie_{book.CategoryID=Category.CategoryID} Book) \bowtie_{Borrow.BookID=Book.BookID} Borrow) \bowtie_{Member.MemberID=borrow.MemberID} Member) \cap \pi_{Member.Name}(\sigma_{Category.CategoryName="Physics"} Category)$

$\bowtie_{book.CategoryID=Category.CategoryID} Member)$

c. لیست نام عضو و نام کتاب‌هایی را به دست آورید که عضو به موضوع کتاب مورد نظر علاقه‌مند است ولی هنوز آن کتاب را امانت نگرفته است.

$\Pi Member.Name, Book.Title (book \bowtie_{(book.CategoryID=Member.CategoryID)} Member) - \Pi Member.Name, Book.Title ((book \bowtie_{book.CategoryID=Member.CategoryID})Member)$

$\bowtie_{Book.BookID=Borrow.BookID \wedge Member.MemberID=Borrow.MemberID} Borrow$

d. لیست نام عضو و نام کتاب‌هایی را به دست آورید که عضو کتابی از دسته Drama را به امانت برده و هنوز تحویل نداده است و از تاریخ مجاز امانت آن‌ها بیش از ده روز گذشته است. (فرض کنید متغیر Today بیانگر تاریخ امروز است).

$\Pi Member.Name, Book.Title$

$((\sigma_{(Category="Drama")}Category) \bowtie_{(book.CategoryID=Category.CategoryID)} Book)$

$\bowtie_{(Book.BookID=Borrow.BookID)}$

$((\sigma_{(IsReturned=False \wedge ReturnDate < Today-10)} Borrow) \bowtie_{Member.MemberID=Borrow.MemberID} Member)$

e. نتیجه عبارت زیر را توضیح دهید.

$\Pi Member.Name, Book.Title (\sigma_{Borrow.NumDays * Book.Penalty \geq 1} \dots ((Borrow \bowtie_{Borrow.BookID=Book.BookID} Book) \bowtie_{Borrow.MemberID=Member.MemberID} Member))$

نام عضو و کتاب‌هایی را برمیگرداند که عضو بیش از صد هزار تومن برای آن کتاب جریمه شده است.

f. نتیجه عبارت زیر را توضیح دهید.

$\Pi_{Book.Title} (\sigma_{Category.CategoryName="Philosophy" \wedge Author.Name="Plato"} ((Book \bowtie_{Book.AuthorID=Author.AuthorID} Author) \bowtie_{Book.CategoryID=Category.CategoryID} Category)) \cap \Pi_{Book.Title} ((\sigma_{IsReturned=false} (Borrow)) \bowtie_{Borrow.BookID=Book.BookID} Book)$

نام کتاب‌های فلسفی را برمیگرداند که نویسنده‌ی آنها افلاطون نیست و کسی به امانت گرفته است.