



تکلیف دوم پایگاه داده یک

فروردین ۱۴۰۳

استاد درس

حمید روایی

سپهر عبادی

۹۹۳۳۲۴۳

سوال ۱:

مطابق با ER-Model طراحی شده در سوال ۷ تکلیف قبل، relation schema مرتبط با آن را بنویسید.

سوال ۲:

با توجه به ساختار پایگاه داده کتابخانه زیر به سوالات با استفاده از جبر رابطه ای پاسخ بدهید.

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)

*CategoryID در جدول Member مشخص کننده موضوع مورد علاقه است.

a. نام کتاب هایی که شخصی با آیدی ۱۲۳ به امانت گرفته و نویسنده کتابها شخصی با آیدی ۴۵۶ باشد و هنوز به کتابخانه برگردانده نشده باشد.

$\pi_{\text{Title}}(\sigma_{\text{MemberID}=123 \wedge \text{AuthorID}=456 \wedge \text{IsReturned}=\text{false}}(\text{Book} \bowtie \text{Borrow}))$

b. نام اعضای را بیابید که به math علاقه دارند و کتابی از موضوع math امانت گرفته اند.

$\pi_{\text{Name}}(\sigma_{\text{CategoryName}=\text{'Math'} \wedge \text{IsReturned}=\text{false}}(\text{Member} \bowtie (\text{Borrow} \bowtie \text{Book} \bowtie \text{Category})))$

c. نام اعضا و لیست کتاب هایی که شخص به آن موضوع علاقه دارد ولی هنوز آن کتاب را به امانت نگرفته است.

$\pi_{\text{Name}, \text{Title}}(\sigma_{\text{CategoryName}=\text{Member.CategoryID} \wedge \text{IsReturned}=\text{false}}(\text{Member} \bowtie (\text{Borrow} \bowtie \text{Book} \bowtie \text{Category})))$

d. لیست نام عضو و نام کتاب هایی را بدست آورید که عضو کتابی از دسته Drama را به امانت برده و هنوز تحویل نداده است و از تاریخ مجاز امانت ان بیش از ده روز گذشته است.

$\pi_{\text{Name}, \text{Title}}(\sigma_{\text{CategoryName}='Drama' \wedge \text{IsReturned}=\text{false} \wedge \text{NumDays}>10}(\text{Member} \bowtie (\text{Borrow} \bowtie \text{Book} \bowtie \text{Category})))$

e. نتیجه عبارت زیر را توضیح دهید.

$\Pi \text{Book.Title}(\sigma_{\text{Category.CategoryName}="Philosophy"}^{\wedge \text{Author.Name} \neq "Plato"}((\text{Book} \bowtie \text{Book.AuthorId}=\text{Author.AuthorIdAuthor}) \bowtie \text{Book.CategoryId}=\text{Category.CategoryIdCategory})) \cap \Pi \text{Book.Title}((\sigma_{\text{IsReturned}=\text{false}(\text{Borrow})} \bowtie \text{Borrow.BookId}=\text{Book.BookIdBook}))$

این عبارت SQL معادل با دو بخش است که یک اشتراک دارند و سپس با هم اشتراک می گذارند:

۱. بخش اول:

$\Pi \text{Book.Title}(\sigma_{\text{Category.CategoryName}="Philosophy"}^{\wedge \text{Author.Name} \neq "Plato"}((\text{Book} \bowtie \text{Book.AuthorId}=\text{Author.AuthorIdAuthor}) \bowtie \text{Book.CategoryId}=\text{Category.CategoryIdCategory}))$

این بخش ابتدا همه ی کتاب ها را انتخاب می کند که دسته بندی آنها "Philosophy" است و نویسنده ی آنها Plato نیست.

برای این کار، ابتدا جدول Book با جدول Author با استفاده از عملیات join ادغام می شود تا نویسنده ی هر کتاب را بدست

آوریم. سپس با جدول Category نیز ادغام می شود تا دسته بندی های هر کتاب را به دست آوریم. در نهایت، با استفاده از انتخاب

(selection)، کتاب‌هایی که دسته‌بندی آن‌ها "Philosophy" است و نویسنده‌ی آن‌ها Plato نیست را انتخاب می‌کنیم و تنها عنوان آن‌ها را با استفاده از عملیات π projection برمی‌گردانیم.

۲. بخش دوم

$\pi \text{Book.Title}((\sigma \text{IsReturned}=\text{false}(\text{Borrow})) \bowtie \text{Borrow.BookId}=\text{Book.BookId} \text{Book})$

این بخش ابتدا همه‌ی کتاب‌هایی را که هنوز تحویل داده نشده‌اند انتخاب می‌کند. برای این کار، با استفاده از انتخاب، از جدول Borrow کتاب‌هایی که تحویل داده نشده‌اند را انتخاب می‌کنیم. سپس با جدول Book با استفاده از عملیات join ادغام می‌شوند تا جزئیات هر کتاب (شامل عنوان) بدست آید. در نهایت، با استفاده از عملیات π projection، تنها عنوان کتاب‌هایی که هنوز تحویل داده نشده‌اند را برمی‌گردانیم. بخش‌های این دو عبارت سپس با استفاده از عملیات اشتراک (intersection) با هم ادغام می‌شوند تا نتیجه‌ی نهایی حاصل شود. این نتیجه شامل عنوان کتاب‌هایی است که هم دسته‌بندی آن‌ها "Philosophy" است و نویسنده‌ی آن‌ها Plato نیستند و هم هنوز توسط کاربران تحویل داده نشده‌اند.

سوال ۳:

با توجه به پایگاه داده دانشگاه به سوالات زیر پاسخ دهید.

a. با توجه به گسترش علوم بین رشته ای برخی از دروس توسط چند دانشکده ارائه میشوند با تغییر جداول موجود (اضافه کردن خصوصیات یا جدول جدید) این امکان را در این پایگاه داده فراهم کنید.

برای فراهم کردن این امکان در پایگاه داده دانشگاه، می‌توانید عملیات زیر را انجام دهید:

۱. اضافه کردن جدول جدید:

ابتدا می‌توانید یک جدول جدید به نام "Interdisciplinary_Courses" یا مشابه آن ایجاد کنید که شامل مواردی مانند "CourseID", "CourseName", و "InterdisciplinaryDepartments" باشد. در این جدول، می‌توانید اطلاعات مربوط به دروسی که توسط چند دانشکده ارائه می‌شوند را نگهداری کنید.

۲. اضافه کردن خصوصیت جدید به جدول موجود:

در صورتی که خصوصیات جدیدی برای دروس موجود مورد نیاز است، می‌توانید این خصوصیت‌ها را به جداول موجود اضافه کنید. به عنوان مثال، می‌توانید یک ستون به نام "Interdisciplinary" به جدول "Courses" اضافه کنید که نشان دهد آیا درس مربوطه بین رشته‌ای است یا خیر.

با انجام یکی از این عملیات، امکان مدیریت و پیگیری دروس بین رشته‌ای در دانشگاه را در پایگاه داده فراهم می‌شود. این امکان به اداره دانشگاه و دپارتمان‌ها کمک می‌کند تا بهترین برنامه درسی را ارائه دهند و از تنوع در ارائه دروس به دانشجویان استفاده کنند.

b. فرض کنید تصمیم گرفته شود که یک درس زین پس در دانشگاه ارائه نشود. آیا حذف رکورد این درس از پایگاه داده روش درستی جهت انجام این کار است؟

حذف رکورد یک درس از پایگاه داده ممکن است در برخی موارد مناسب باشد، اما در بسیاری از موارد به عنوان روش اصلی برای از بین بردن درس‌ها توصیه نمی‌شود. دلایل زیر توضیح می‌دهند چرا حذف رکورد یک درس از پایگاه داده به تنهایی روش مناسبی برای این کار نیست:

۱. از دست رفتن اطلاعات: با حذف رکورد یک درس، اطلاعات مربوط به آن درس از پایگاه داده حذف می‌شود و هیچ اثری از آن درس در آینده قابل بازیابی نیست. این می‌تواند مشکلاتی را ایجاد کند اگر در آینده نیاز به دسترسی به اطلاعات این درس برای مقاصد مختلفی مانند ارزیابی عملکرد دانشجویان یا مطالعات تحقیقاتی پیش رو داشته باشیم.

۲. امکان بازگردانی و بازیابی: حذف رکورد یک درس ممکن است منجر به از دست رفتن اطلاعات مهم و مورد نیاز شود. به جای حذف، بهتر است از روش‌های دیگر مانند غیرفعال‌سازی یا آرشیو کردن استفاده کنید که امکان بازگردانی و بازیابی اطلاعات را حفظ می‌کنند.

۳. امکان وقوع خطا: در صورتی که اطلاعات مربوط به یک درس از پایگاه داده حذف شود، ممکن است خطاهایی رخ دهد که در نتیجه اطلاعات مهم از دست برود یا اطلاعات غیرمورد نیاز حفظ شود. این مسئله می‌تواند پیچیدگی‌های اضافی را به مدیریت داده‌ها اضافه کند.

۴. انطباق با استانداردها و قوانین: در برخی موارد، استانداردها، قوانین و مقررات ممکن است از حذف اطلاعات برخی داده‌ها پیشگیری کنند. به عنوان مثال، در برخی حوزه‌ها ممکن است مقررات حفظ اطلاعات برای مدت زمان معینی را تعیین کنند که در صورت حذف رکوردها، این مقررات را نقض می‌کنند.

با توجه به این موارد، بهتر است که به جای حذف رکورد یک درس، از روش‌های دیگری مانند غیرفعال‌سازی یا آرشیو کردن استفاده کنید تا امکان بازگردانی و بازیابی اطلاعات را حفظ کنید و از از دست رفتن اطلاعات مهم جلوگیری کنید.

c. جدول `time_slot` را در نظر بگیرید با توجه به اینکه یک بازه زمانی خاص ممکن است بیش از یک بار در هفته تکرار شود. توضیح دهید که چرا `start_time, day` بخشی از کلید اصلی هستند اما `end_time` نیست؟

در نظر گرفتن `start_time` و `day` به عنوان بخشی از کلید اصلی در جدول `time_slot` معمولاً به دلایل زیر انجام می‌شود:

۱. تکراری نبودن: با در نظر گرفتن `start_time` و `day` به عنوان بخشی از کلید اصلی، اطمینان حاصل می‌شود که هیچ دو رکوردی در جدول `time_slot` وجود ندارد که همزمان در همان روز و با همان زمان شروع تکرار شوند. این امر از ایجاد داده‌های تکراری جلوگیری می‌کند و اطمینان حاصل می‌شود که هر بازه زمانی فقط یکبار در هر روز و با هر زمان شروع ثبت می‌شود.

۲. محدود کننده بودن: در بسیاری از سناریوها، تعیین یک بازه زمانی با استفاده از شروع آن و روز مربوطه کافی است و نیازی به ذکر زمان پایان نیست. به عنوان مثال، در برنامه‌ریزی کلاس‌ها یک درس، تعیین زمان شروع کلاس و روز آن کافی است و نیازی به ذکر زمان پایان در هر رکورد نیست.

از طرف دیگر، اگر `end_time` را به عنوان بخشی از کلید اصلی در نظر بگیریم، این به معنای این است که هر بازه زمانی باید با زمان خاصی پایان یابد و این اجبار می کند که برای هر بازه زمانی، زمان پایان دقیقاً مشخص شود. این ممکن است منجر به پیچیدگی بیشتر در مدیریت و ثبت داده ها شود، زیرا ممکن است برای بازه های زمانی کوتاه و یا متغیر، زمان پایان دقیقاً قابل تعیین نباشد. بنابراین، با در نظر گرفتن `start_time` و `day` به عنوان بخشی از کلید اصلی در جدول `time_slot`، از مزایای بالا بهره مند می شویم و از ایجاد داده های تکراری جلوگیری می کنیم، در حالی که انعطاف و سهولت در ثبت اطلاعات را حفظ می کنیم.

سوال ۴:

تفاوت بین candidate key و primary key را توضیح دهید. همچنین super key را تعریف کنید.

به ترتیب، تعریف و تفاوت بین این سه مفهوم را بررسی می‌کنیم:

۱. Super Key:

یک Super Key مجموعه‌ای از یک یا چند ستون در یک جدول است که توانایی یکتا شناسایی هر رکورد را دارد. به عبارت دیگر، Super Key حاوی ستون‌هایی است که می‌توانند به تنهایی یا به ترکیب با ستون‌های دیگر، هر رکورد را به طور یکتا مشخص کنند. به عنوان مثال، در یک جدول اطلاعات فردی، می‌توان شماره تماس (Phone Number) به تنهایی یا ترکیبی از شماره تماس و نام (Phone Number, Name) را به عنوان یک Super Key در نظر گرفت.

۲. Candidate Key:

Candidate Key یک Super Key است که برای شناسایی هر رکورد به طور یکتا استفاده می‌شود و دیگر Super Key ها این ویژگی را ندارند. به عبارت دیگر، یک Candidate Key یک زیرمجموعه از Super Key است که هر رکورد را به طور یکتا شناسایی می‌کند و هیچ ستون اضافی ندارد. در بسیاری از موارد، از میان چندین Candidate Key ممکن است یکی انتخاب شود و به عنوان کلید اصلی (Primary Key) استفاده شود.

۳. Primary Key:

Primary Key یک Candidate Key است که به عنوان کلید اصلی برای شناسایی هر رکورد در یک جدول انتخاب می‌شود. یک جدول معمولاً یک Primary Key دارد که می‌تواند توسط سازنده جدول یا مدیر پایگاه داده تعیین شود. Primary Key باید یکتا، غیر تهی و ثابت باشد، به این معنی که برای هیچ رکوردی نمی‌تواند مقدار NULL داشته باشد و باید به طور یکتا هر رکورد را شناسایی کند.

به طور خلاصه، Super Key یک مفهوم عمومی‌تر است که هر مجموعه‌ای از ستون‌ها را که توانایی شناسایی هر رکورد را دارد، مشخص می‌کند. Candidate Key یک زیرمجموعه از Super Key است که به طور یکتا هر رکورد را شناسایی می‌کند و ممکن است چندین Candidate Key در یک جدول وجود داشته باشد. Primary Key یکی از Candidate Key ها است که به عنوان کلید اصلی انتخاب می‌شود و باید به طور یکتا و ثابت باشد.

سوال ۵:

با توجه به دو جدول T۱ و T۲ حاصل جبر رابطه ای را بنویسید.

TABLE T1

P	Q	R
10	a	5
15	b	8
25	a	6

TABLE T2

A	B	C
10	b	6
25	c	3
10	b	5

a. $T1 \bowtie_{T1.P = T2.A} T2$

b. $T1 \bowtie_{T1.Q = T2.B} T2$

c. $T1 \bowtie_{T1.P = T2.A} T2$

d. $T1 \bowtie_{T1.Q = T2.B} T2$

e. $T1 \cup T2$

f. $T1 \bowtie_{(T1.P = T2.A \text{ AND } T1.R = T2.C)} T2$

a.

۱۰	a	۵	۱۰	b	۶
----	---	---	----	---	---

b. Null

c.

۱۰	a	۵	۱۰	b	۶
----	---	---	----	---	---

d. Null

e.

۱۰	a	۵
۱۵	b	۸
۲۵	a	۶
۱۰	b	۶
۲۵	c	۳
۱۰	b	۵

f. Null

سوال ۶:

طرح تامین کننده-قطعات-کاتالوگ را در نظر بگیرید. آنچه را که پرس و جوهای زیر محاسبه میکنند بیان کنید:

***Suppliers* (*sid:integer, sname:string, address:string*)**

***Parts* (*pid:integer, pname:string, color:string*)**

***Catalog* (*sid:integer, pid:integer, cost:real*)**

1. $\pi_{sname}(\pi_{sid}(\sigma_{color="red"}Parts) \bowtie (\sigma_{cost < 100}Catalog) \bowtie Suppliers)$

این پرس وجو اطلاعات لازم برای نمایش اطلاعات تامین کنندگان مورد استفاده در پروژه هایی که در آنها از قطعاتی با رنگ "قرمز" استفاده می شود را بازیابی می کند. به طور دقیق تر، این پرس وجو اطلاعات تامین کنندگان را بازیابی می کند که قطعاتی با رنگ "قرمز" را عرضه می کنند و همچنین این قطعات با تعداد کمتر از ۱۰۰ در گزارش های کاتالوگ آنها آمده است. در واقع، این پرس وجو می تواند به کاربر اطلاعات لازم را برای انتخاب تامین کننده ای که قطعات با رنگ مورد نظر و تعداد مورد نظر را ارائه می دهد، فراهم کند.

2. $(\pi_{sname}((\sigma_{color="red"}Parts) \bowtie (\sigma_{cost < 100}Catalog) \bowtie Suppliers))$

$\cap (\pi_{sname}((\sigma_{color="green"}Parts) \bowtie (\sigma_{cost < 100}Catalog) \bowtie Suppliers))$

این پرس وجو اطلاعات لازم برای نمایش قطعاتی را که با رنگ "قرمز" در کاتالوگ وجود دارند و از تامین کنندگانی تأمین می شوند که تعداد قطعات موجود در کاتالوگ آنها کمتر از ۱۰۰ است را بازیابی می کند. سپس با استفاده از عملگر \cap بازیابی اطلاعات مربوط به قطعاتی که با رنگ "سبز" در کاتالوگ وجود دارند و از تامین کنندگانی تأمین می شوند که تعداد قطعات موجود در کاتالوگ آنها کمتر از ۱۰۰ است نیز انجام می دهد.

در واقع، این پرس و جو به کاربر اطلاعات لازم را برای انتخاب قطعاتی که از تامین کنندگانی با موجودی کمترین تعداد را تأمین می کنند و همچنین از رنگ های مختلفی استفاده می کنند، فراهم می کند.

3. $\pi_{name} ((\pi_{sid, name} ((\sigma_{color="red"} Parts) \bowtie (\sigma_{cost < 100} Catalog) \bowtie Suppliers)) \cap (\pi_{sid, name} ((\sigma_{color="green"} Parts) \bowtie (\sigma_{cost < 100} Catalog) \bowtie Suppliers)))$

این پرس و جو اطلاعات مورد نیاز برای نمایش قطعاتی را که با رنگ "قرمز" در کاتالوگ وجود دارند و از تامین کنندگانی تأمین می شوند که تعداد قطعات موجود در کاتالوگ آنها کمتر از ۱۰۰ است را بازیابی می کند. سپس با استفاده از عملگر \cap ، اطلاعات مربوط به قطعاتی که با رنگ "سبز" در کاتالوگ وجود دارند و از تامین کنندگانی تأمین می شوند که تعداد قطعات موجود در کاتالوگ آنها کمتر از ۱۰۰ است نیز انتخاب می شود.

در واقع، این پرس و جو اطلاعات مورد نیاز برای انتخاب قطعاتی که از تامین کنندگانی با موجودی کمترین تعداد را تأمین می کنند و همچنین از رنگ های مختلفی استفاده می کنند، را فراهم می کند.