



دانشگاه صنعتی اصفهان  
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف دوم درس پایگاه داده‌ها ۱

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: پاییز ۱۴۰۰

مدرس: دکتر ناصر قدیری مدرس

دستیاران آموزشی: عارف آسمند - بهاره حاجی هاشمی - پردیس مرادیکی

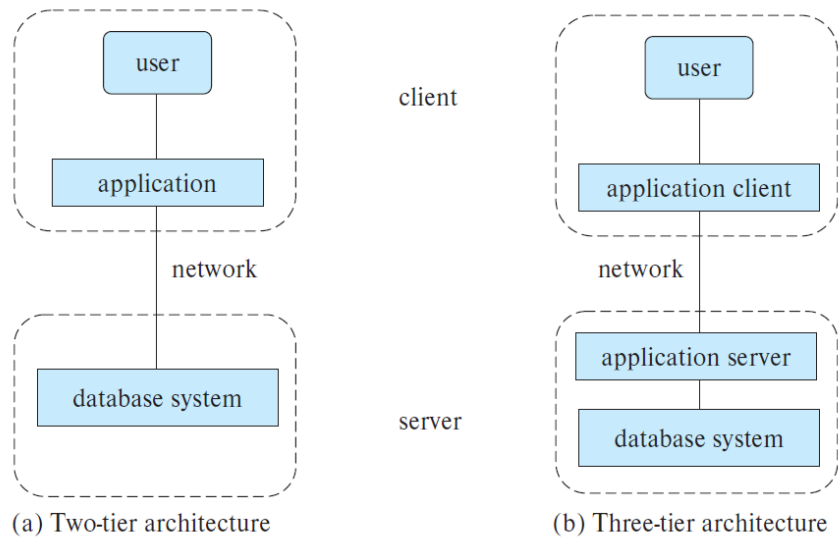
- سیدمهدی موسوی

۱

نوع متغیر (. . . char, varchar, int)	داده
char	سه حرف مخفف شده ماه‌های میلادی
numeric	قیمت دلاری محصولات که همگی دو رقم اعشار دارند
varchar	نام و نام خانوادگی کاربر
char	کد ملی کاربر
real, double precision	ذخیره قیمت لحظه‌ای ارزهای دیجیتال دلار
int	تعداد بازدید یک ویدئو

جدول ۱: جدول شماره ۱

۲



شکل ۱

۳

۱.۳

WHERE 1 <= (SELECT count(name) FROM student)

معماری سه‌لایه	معماری دولایه	ها
بیشتر (سریع‌تر)	کمتر (کندتر)	ب
بیشتر (کلاینت مجاز به تعامل مستقیم با پایگاه داده نمی‌باشد)	کمتر (کلاینت می‌تواند مستقیماً با پایگاه داده تعامل داشته باشد)	ب
بیشتر (چون تعداد لایه‌ها بیشتر است در نتیجه indirectionها بیشتر است)	کمتر	ی
بیشتر (به دلیل مازولاری بالا امکان maintainance آن بیشتر است)	کمتر	یری
بیشتر (امکان افزودن feature در آن ساده‌تر است)	کمتر	یری
بیشتر (به دلیل لایه‌ای بودن یک پارچگی بیشتر حفظ می‌شود)	کمتر	گی

جدول ۲: جدول شماره ۱

## ۲.۳

attributeهای مشخص شده در قسمت قبل با کاراکتر # و @ به ترتیب بیانگر کلید اصلی بودن و کلید خارجی بودن در آن جدول اند.

## ۴

### ۱.۴

مقدار خاص null که در هر domain قرار دارد (به جز مواردی که خود ما محدود کرده باشیم) به این معنی است که ما اطلاعی راجع به این فیلد نداریم. توجه شود که null با صفر متفاوت است. برای مثال اگر حقوق یک کارمند صفر باشد به این معنی است که جمع دریافتی‌ها و کسری‌های او صفر می‌شود، در حالی که اگر حقوق او null باشد به این معنی است که اطلاعی از حقوق او در پایگاه داده موجود نیست. همچنین در پایگاه داده‌ی دانشگاه، null بودن نمره یک دانشجو به معنی موجود نبودن (وارد نشده بودن) آن در پایگاه داده‌ی دانشگاه است.

## ۲.۴

می‌توان یک relation schema به نام employment با attributeهای i\_id که شناسه استاد، و dept\_name که نام دانشکده است، به شکل زیر تعریف کرد:

employment
i_id
dept_name

به این شکل می‌توانیم dept\_name را از instructor حذف کنیم و رابطه‌ی اشتغال را بین دانشکده و استاد برقرار کنیم (شبیه کاری که برای استاد راهنما انجام شد). از این طریق در پایگاه داده یک استاد می‌تواند در بیش از یک دانشکده مشغول به کار باشد.

## ۳.۴

یک راه می‌تواند اضافه کردن یک attribute تحت عنوان second\_advisor به advisor باشد. از آنجایی که می‌دانیم هر دانشجو حداکثر می‌تواند دو استاد راهنما داشته باشد، در صورتی که دانشجو یک استاد راهنما داشت، تنها از first\_advisor استفاده کنیم و فیلد second\_advisor برابر null قرار دهیم.

۴.۴

اگر یک رکورد از student حذف شود، آنگاه رکورد مربوط به رکورد حذف شده در takes نامعتبر خواهد بود. همچنین در صورتی که یک رکورد با ID ناموجود وارد شود دچار نقض می‌شود.

۵

۱.۵

$$\Pi_{Title, ReturnDate}(Borrow \bowtie_{MemberID = 1356 \wedge IsReturned = false} Book)$$

۲.۵

$$\Pi_{Name}(Member \bowtie_{Member.CategoryID = "Physics"} (Book \bowtie_{Book.BookID = Borrow.BookID \wedge CategoryID = "Physics"} Borrow))$$

۳.۵

$$\Pi_{Name, Title}(Member \bowtie_{Member.CategoryID = Book.CategoryID} Book) - \Pi_{Name, Title}(Borrow \bowtie_{Borrow.MemberID = Member.MemberID \wedge Borrow.BookID = Book.BookID} (Member \bowtie_{Member.CategoryID = Book.CategoryID} Book))$$

۴.۵

$$\Pi_{Name, Title}(Member \bowtie_{Member.MemberID = Borrow.MemberID} (Borrow \bowtie_{CategoryID = "Drama" \wedge IsReturned = false \wedge Today - ReturnDate > 10 Days} Book))$$

۵.۵

ابتدا عبارت را به شکل زیر تفکیک می‌کنیم:

$$D \leftarrow Borrow \bowtie_{Borrow.BookID = Book.BookID} Book$$

$$C \leftarrow D \bowtie_{Borrow.MemberID = Member.MemberID} Member$$

$$B \leftarrow \sigma_{Borrow.NumDays \times Book.Penalty \geq 100000}(C)$$

$$A \leftarrow \Pi_{Member.Name, Book.Title}(B)$$

D لیست اطلاعات کتاب‌های امانت گرفته‌شده را همراه با اطلاعات امانت آن‌ها برمی‌گرداند.

C لیست اطلاعات اعضا و اطلاعات کتاب‌های امانت گرفته‌شده آن‌ها را همراه با اطلاعات امانت آن‌ها برمی‌گرداند.

B سطرهایی از C را برمی‌گرداند که جریمه دیرکرد آن‌ها بزرگتر یا مساوی ۱۰۰۰۰۰ تومان باشد.

A که همان عبارت نهایی است ستون‌های نام عضو و نام کتاب را از B برمی‌گرداند.

پس نتیجه عبارت، نام عضو و نام کتاب‌هایی که امانت گرفته‌اند و جریمه دیرکرد آن‌ها بزرگتر یا مساوی ۱۰۰۰۰۰ تومان است می‌باشد.

## ۶.۵

ابتدا عبارت را به شکل زیر تفکیک می‌کنیم:

$$F \leftarrow Book \bowtie_{Book.AuthorID = Author.AuthorID} Author$$

$$E \leftarrow F \bowtie_{Book.CategoryID = Category.CategoryID} Category$$

$$D \leftarrow \sigma_{Category.CategoryName = "Philosophy" \wedge Author.Name \neq "Plato"}(E)$$

$$C \leftarrow (\sigma_{IsReturned = false}(Borrow)) \bowtie_{Borrow.BookID = Book.BookID} Book$$

$$B \leftarrow \Pi_{Book.Title}(C)$$

$$A \leftarrow \Pi_{Book.Title}(D)$$

$$A - B$$

F اطلاعات کتاب‌ها را به همراه اطلاعات نویسنده‌ی آن‌ها برمی‌گرداند.

E اطلاعات F را به تفکیک اطلاعات موضوع آن‌ها برمی‌گرداند.

D سطرهایی از E با موضوع Philosophy که نویسنده آن‌ها Plato نیست را برمی‌گرداند.

C اطلاعات کتاب‌های امانت داده شده اما بازگردانده نشده اند را برمی‌گرداند.

B ستون نام کتاب از C را برمی‌گرداند.

A ستون نام کتاب از D را برمی‌گرداند.

A - B که همان عبارت نهایی است تفاضل B از A را برمی‌گرداند.

پس نتیجه عبارت، نام کتاب‌هایی با موضوع Philosophy که نویسنده آن‌ها Plato نیست و امانت داده شده اما بازگردانده نشده اند را برمی‌گرداند.

## منابع

- [1] <https://www.geeksforgeeks.org/difference-between-two-tier-and-three-tier-database-architecture/>
- [2] <https://medium.com/@gacheruevans0/2-tier-vs-3-tier-architecture-26db56fe7e9c>
- [3] <https://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/>
- [4] <https://www.ibm.com/nl-en/cloud/learn/three-tier-architecture>
- [5] <https://nitrosphere.com/uncategorized/2-tier-vs-3-tier-application-architecture-could-the-winner-be-2-tier-2/>