



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

## آزمایشگاه پایگاه داده

دستور کار جلسه سوم  
آشنایی با محیط SQL Server

استاد درس:  
بهناز السادات متولی  
bs.motavali@yahoo.com

مهر ماه ۱۳۹۶

لطفاً نتایج هر بخش را بصورت مجزا با ذکر شماره و عنوان سوال در گزارش کار خود بیاورید.

## تمرین اول

بخش ۱:

۱. دو جدول به نام‌های Stud و Dept مطابق مشخصات زیر ایجاد کنید. (اسم پایگاه داده شما باید شماره گروه تان باشد به طور مثال 01g3Team گروه اول) جدول Stud با جدول Dept باید رابطه یک به چند داشته باشد. این رابطه باید به گونه‌ای باشد که در صورتیکه یک بخش حذف شد تمامی دانشجویان مربوط به آن بخش نیز حذف شوند.

(Tip: Use On Delete Cascade)

Stud

محدودیت	نوع	ستون
Primary Key, Identity	عدد صحیح	Stud_id
	رشته حداکثر ۵۰ کاراکتر	Name
	عدد صحیح	Tuition
Foreign Key	عدد صحیح	DeptNo

Dept

محدودیت	نوع	ستون
Primary Key	عدد صحیح	DeptNo
	رشته ۵۰ کاراکتر	Name

۲. فایل‌های dept.sql و stud.sql را اجرا کنید. (این فایل‌ها حاوی تعدادی دستور Insert هستند)

۳. دستوراتی بنویسید که کارهای زیر را انجام دهند:

- لیستی از نام تمامی دانشجویان به همراه نام بخش آنها.
- لیست دانشجویانی که در بخش‌هایی با نام‌های 'computer' و 'physics' می‌باشند.
- لیست بخش‌هایی که دارای دانشجو نمی‌باشند. (Use: Not Exists)
- لیست دانشجویانی از هر بخش که بیشترین شهریه را در بخش خود پرداخت می‌کنند.

۱. جدول زیر را هم در بانک خود ایجاد کنید.

Prj		
محدودیت	نوع	ستون
Primary Key, Identity	عدد صحیح	Pri_id
	رشته ۵۰ کاراکتر	Name

۲. هر دانشجو می‌تواند در یک یا چند پروژه شرکت داشته باشد. همچنین هر پروژه شامل یک یا چند دانشجو می‌تواند باشد. هر دانشجو در هر پروژه دارای یک نقش می‌باشد. در نتیجه جدول Stud با جدول Prj رابطه چند به چند دارد. این رابطه را برقرار کنید. برای این منظور از جدول میانی به نام PrjMember استفاده کنید. ستون نقش (role از نوع عدد صحیح) باید در کدام جدول قرار گیرد؟ توجه: رابطه باید به گونه‌ای باشد که در صورت حذف یک دانشجو یا پروژه، سطرهای متناظر در جدول PrjMember حذف شوند.

۳. پروژه‌های زیر را وارد کنید:

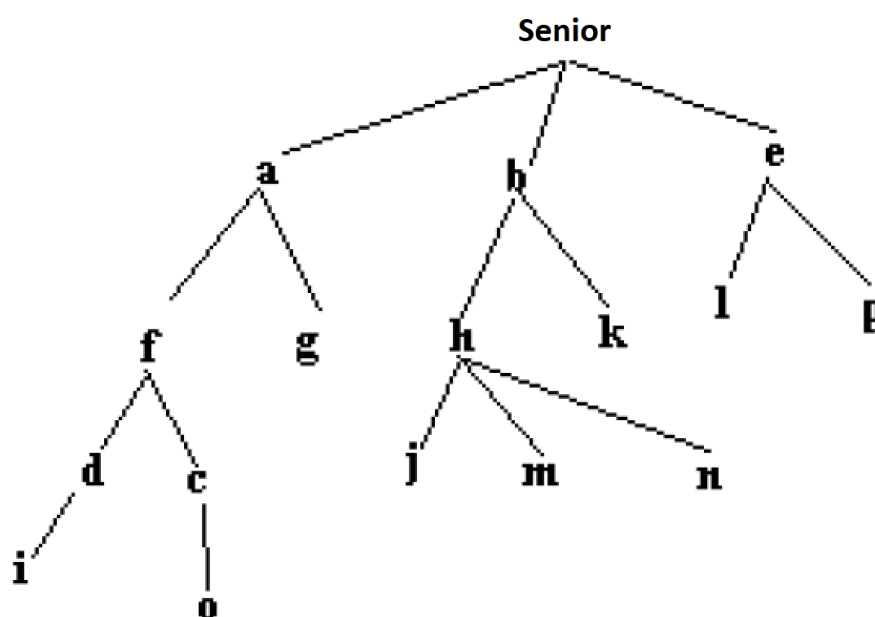
- a. پروژه A: دانشجو master (۱)، دانشجو b (۲)، دانشجو o (۲)
- b. پروژه B: دانشجو c (۳)، دانشجو d (۳)، دانشجو a (۲)، دانشجو e (۲)، دانشجو b (۱)
- c. پروژه C: دانشجو a (۱)، دانشجو f (۲)
- d. پروژه D: دانشجو master (۱)، دانشجو f (۲)، دانشجو h (۳)، دانشجو i (۴)، دانشجو g (۲)، دانشجو k (۳)

۴. دستوراتی بنویسید که کارهای زیر را انجام دهند:

- a. لیست پروژه‌هایی که کمتر از ۴ دانشجو بر روی آنها کار می‌کنند.
- b. لیستی از دانشجویان به همراه نام بخش مربوطه، نام پروژه‌ها و نقش آنها در این پروژه‌ها.
- c. مجموع شهریه افرادی که در پروژه‌ای به نام B مشغولند.
- d. لیستی از پروژه‌ها به همراه متوسط شهریه افرادی که در آنها مشغولند.

۱. هر دانشجو می‌تواند از یک دانشجوی ارشد کمک بگیرد و هر دانشجو ارشد می‌تواند به یک یا چند دانشجو کمک کند. دانشجو ارشد خود نیز یک دانشجو است. در نتیجه جدول دانشجو با خود رابطه یک به چند دارد. با استفاده از این تکنیک می‌توان یک رابطه درختی را در یک جدول گنجاند. زیرا در این صورت خود دانشجو ارشد می‌تواند شامل یک دانشجو دکتری دیگر باشد. در این صورت می‌توان چارت آزمایشگاهی را مدل نمود. برای این منظور به جدول Stud ستونی تحت نام Senior\_id اضافه کنید که کلید خارجی به Stud\_id می‌باشد.

۲. چارت زیر را مدل کنید:



۳. دستوراتی بنویسید که کارهای زیر را انجام دهند:

a. لیست دانشجویانی که با ارشدشان در یک بخش هستند.

b. لیست دانشجویان تحت نظارت فردی به نام 'a'.

## تمرین اختیاری

- جدول Emp\_Table با مشخصات زیر را ایجاد نمایید:

محدودیت	نوع	ستون
Primary Key	عدد صحیح	Emp_id
Not Null	رشته ۵۰ کاراکتر	FName
Not Null	رشته ۵۰ کاراکتر	LName
	رشته با طول متغیر. حداکثر ۲۵۶ کاراکتر.	Address
Default = 0 CHECK Salary >= 0	عدد صحیح	Salary
	رشته ۲۵ کاراکتر	Department

- اطلاعات زیر را در این جدول وارد کنید:

Emp_id	FName	LName	Address	Salary	Department
۱	Ali	Hassani	Tehran	1000	Inventory
2	Maryam	Hosseini		900	Sale
3	Dummy	Foo			

۱. به اول تمامی آدرس‌هایی که Null نیستند Iran اضافه کنید. همچنین اگر اولین حرف کلمه بعدی کوچک است آن را بزرگ کنید. به عنوان مثال اگر آدرس Tehran بود تبدیل به Iran Tehran می‌شود.

۲. لیست افراد را به ترتیب نزولی نام خانوادگی و نام به همراه آدرس بدست آورید. نام و نام خانوادگی باید پشت سر هم و با یک فاصله در یک ستون تحت نام Name بیایند. در صورتیکه آدرس Null بود به جای آن Unknown قرار گیرد.

(use: isnull)

(use: select col as Name from ...)

(use: order by col\_1 desc, col\_2 desc)

۳. لیستی از نام خانوادگی افراد بدست آورید. نام خانوادگی‌های تکراری نباید ظاهر شوند. نام خانوادگی‌ها به ترتیب صعودی طول نام خانوادگی مرتب شده باشند.

(use: distinct and len)

۴. با استفاده از دستور (MySQL: describe or desc) یا (MS-SQL: sp\_help) ساختار جدول را مشاهده نمایید.