



درس: پایگاه داده  
استاد: دکتر فرضی

دانشگاه صنعتی خواجه نصیرالدین طوسی

تمرین دستورات SQL  
مهلت بارگذاری: ۱۳۹۹/۰۸/۳۰

۱. با در نظر گرفتن جداول کارمند (Employee)، پروژه (Project) و پروژه‌های در حال اجرا (OnGoing) برای یک سازمان دولتی، به سوالات الف تا ج پاسخ دهید:

Employee (Emp #, Mgr #, Ename, Dep\_Name, Age, Income)

Project (Proj #, Pname, City)

OnGoing (Emp #, Proj #, Hours)

الف) برای پرسمان جبر رابطه‌ای زیر، معادل SQL را بنویسید:

$\pi_{Employee.Emp\#} \left( \sigma_{Employee.Dep\_Name = E.Dep\_Name} \left( Employee \times \rho_E \left( \pi_{Dep\_Name} \left( \sigma_{Ename = "Amir"} (Employee) \right) \right) \right) \right)$

پاسخ:

$\pi_{Employee.Emp\#} \left( \sigma_{Employee.Dep\_Name = E.Dep\_Name} \left( Employee \times \rho_E \left( \pi_{Dep\_Name} \left( \sigma_{Ename = "Amir"} (Employee) \right) \right) \right) \right)$

SELECT Emp #

FROM Employee

WHERE Dep\_Name in (SELECT Dep\_Name

FROM Employee

WHERE Ename = "Amir")

ب) خروجی دستور SQL زیر چه خواهد بود؟

CREATE VIEW List AS

SELECT Mgr #, count(\*)

FROM Employee

GROUP BY Mgr #

پاسخ:

ابتدا تمام رکوردهای جدول کارمندان را براساس کد مدیر گروه‌بندی می‌کند. سپس در هر گروه تعداد رکوردهای موجود را شمارش نموده همراه با کد مدیر آن گروه ثبت می‌کند. در نهایت از خروجی این دستور، یک جدول مجازی با نام List تولید می‌کند. به بیان دیگر "یک جدول مجازی (View) با نام List شامل دو ستون تولید می‌کند، ستون اول کد مدیران و ستون دوم تعداد کارمندانی که زیر نظر هر مدیر".

ج) معادل SQL پرسمان‌های زیر را بنویسید:

"نام و درآمد کارمندانی که درآمدهای آنها از متوسط درآمد دپارتمان‌شان کمتر ولی از متوسط درآمد کل کارکنان سازمان بیشتر باشد"

پاسخ:

نام و درآمد کارکنان با درآمد بیشتر از میانگین سازمان به تفکیک در هر دپارتمان با این شرط که درآمدهای آنها از متوسط درآمد در همان دپارتمان نیز کمتر باشد

```
SELECT Ename, Income
FROM (SELECT *
      FROM Employee
      WHERE Income > (SELECT AVG(Income) FROM Employee))
GROUP BY Dep_Name
HAVING Income < AVG (Income)
```

نام و درآمد کارکنان با حقوق بیش از میانگین سازمان به تفکیک در هر دپارتمان

کلیه کارمندانی که درآمدهای آنها از میانگین درآمد کل سازمان بیشتر است

میانگین درآمد همه کارکنان سازمان

۲. درستی یا نادرستی عبارتهای زیر را با ذکر دلیل تعیین نمایید.

الف) بعد از تعریف یک View روی پایگاه داده، اجرای عملیات خواندن سطرها نسبت به بروزرسانی آنها پیچیدگی کمتری دارد. **صحیح**

View یک جدول مجازی است که روی یک یا چند جدول پایه تعریف می‌شود، بنابراین انجام هرگونه تغییر مانند حذف، درج یا بروزرسانی سطرها باید به جدول پایه منتقل شود. این امر در عمل با مشکلات بسیاری مواجه است. اما خواندن از view بدون هیچ مشکلی امکان پذیر است.

ب) دستور Distinct فقط در نحوه نمایش جواب موثر است و در محتوای جواب دخالتی ندارد. **صحیح**

دستور Distinct موارد تکراری را در نمایش جدول حذف می‌کند، اما در متن تولید پاسخ تاثیری ندارد. از مثالهای دیگر می‌توان به دستور order by (موثر در نمایش) و دستور Group by (موثر در محتوا) اشاره کرد.

ج) جدول میانی جدولی است که توسط سیستم ایجاد، مدیریت و حذف می‌شود و کاربرد دخالتی در آن ندارد. **صحیح**

د) برای اعمال قانون اول جامعیت داده‌ای در SQL کافی است در تعریف ویژگی که به عنوان کلید اصلی است، Unique و Not Null اضافه شود. **غلط**، Not null بودن به قانون دوم جامعیت داده‌ای (Entity integrity) اشاره دارد.

۳. جداول زیر را برای ثبت مشخصات تولیدکنندگان (Supplier)، تولیدات (Products)، مشتریان (Customers) و فروش (Sales) در نظر بگیرید. منظور از جدول Sales این است که "تولید کننده‌ی SR# کالای P# را به تعداد Amount برای مشتری C# تولید و در تاریخ Date به او فروخته است". براساس این جداول، معادل SQL پرسمان‌های زیر را بنویسید:

Supplier (SR #, SRname, SRcity)

Product (P#, Pname, Price, Status)

Customer (C #, Cname, Ccity, Cstreet)

Sales (S #, C #, P #, SR #, Date, Amount)

الف) شناسه مشتریانی را معین کنید که تمام محصولات مورد نیازشان را فقط یک تولید کننده، تولید کرده باشد.

پاسخ:

SELECT C#

FROM Sales

GROUP BY C#

HAVING COUNT(Distinct SR#) = 1

چهارم: در گروه‌های منتخب، کد مشتری را پیدا کن

اول: محصولات را براساس اینکه کدام مشتری آن را خریده است گروه‌بندی کن

دوم: در گروه متعلق به هر مشتری، تعداد تهیه‌کنندگان واحد را بشمار  
سوم: فقط گروه‌هایی را انتخاب کن که تعداد تهیه‌کنندگان محصولاتش فقط یکی باشد.

ب) وضعیت تمام محصولاتی که شرکتی با نام "Arj" تولید نموده است را به "Not Available" تغییر دهید.

پاسخ:

UPDATE Product

SET Product.Status = "Not Available"

FROM Product

INNER JOIN Sales

ON Product.P# = Sales.P#

INNER JOIN Supplier

ON Sales.SR# = Supplier.SR#

WHERE Supplier.SRname = "Arj"

لیست کلیه محصولات تولید شده توسط تولیدکنندگان همراه با نام تولید کننده

لیست کلیه محصولات تولید شده توسط شرکت ارج

ج) محصولی با کد P13 نام "Refrigerator" با قیمت 12000000 به جدول محصولات بیافزایید. (توضیح دهید پس از اجرای این دستور، مقدار صفت Status چه خواهد بود؟)

پاسخ:

```
INSERT INTO Product(P#, Pname, Price)
VALUES ('P13', 'Refrigerator', 12000000)
```

در دستور INSERT مربوط به SQL، اگر صفتی در هنگام درج، مقداردهی نشود، به صورت پیش فرض مقدار ناشناخته (NULL) خواهد گرفت. بنابراین صفت Status مقدار NULL خواهد داشت.

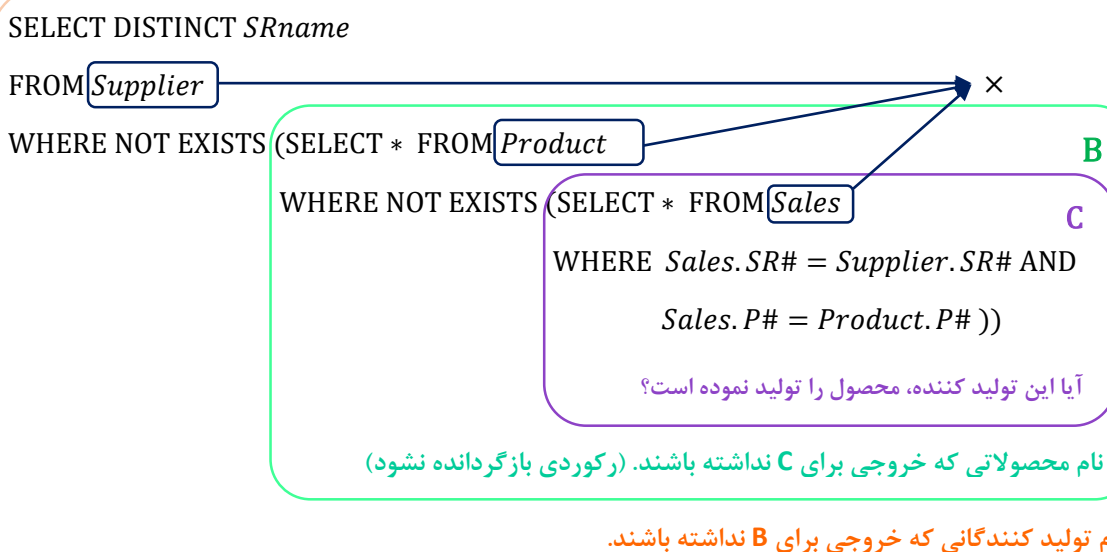
۴. با توجه به جداول ارائه شده در سوال (۳) به سوالات زیر پاسخ دهید:

الف) خروجی دستور SQL زیر چیست؟

```
SELECT DISTINCT SRname
FROM Supplier
WHERE NOT EXISTS (SELECT * FROM Product
                  WHERE NOT EXISTS (SELECT * FROM Sales
                                    WHERE Sales.SR# = Supplier.SR# AND
                                           Sales.P# = Product.P# ))
```

پاسخ:

نام تولید کنندگانی که تمام تولیدات را تولید نموده اند. به بیان دقیقتر، نام تولید کنندگانی که قطعه ای نباشد که تولید نکرده باشد.



ب) دستور بخش (الف) را با استفاده از Count بازنویسی کنید.

پاسخ:

```
SELECT DISTINCT SRname
FROM Supplier
WHERE (SELECT COUNT (Sales.P#)
      FROM Sales
      WHERE Sales.SR# = Supplier.SR#) = (SELECT COUNT Products.P# FROM Product)
```

تعداد کل محصولات تولید شده = تعداد کل محصولاتی که این فروشنده تولید کرده است

۵. با توجه به جداول ارائه شده در سوال (۳) به سوالات زیر پاسخ دهید:

الف) خروجی دستور زیر چند رکورد خواهد داشت؟

```
SELECT Pname, Price
FROM Product
WHERE Price ≥ (SELECT MAX(Price) FROM Product)
```

پاسخ:

دستور اول اسامی تولیداتی را باز می گرداند که قیمتشان بزرگتر یا مساوی بیشینه قیمت کالاها باشد. حالت بیشتر از maximum که هیچگاه اتفاق نمی افتد. اما بسته به رکوردهای اطلاعاتی که در جدول قرار دارد، ممکن است تنها یک محصول با بیشترین قیمت یا بیش از یک محصول با قیمت برابر با بیشینه بازگرداند.

ب) آیا خروجی دستور بخش (الف) با دستور زیر یکسان است؟

```
SELECT Pname, Price
FROM Product
WHERE Price ≥ MAX(Price)
```

پاسخ:

دستور (ب) اصولاً صحیح نیست. زیرا از توابع محاسباتی نظیر AVG، MAX، MIN و ... نباید در قسمت شرط عبارت SQL استفاده نمود. به این علت که، تا هر کجای جدول که پیش می روند بسته به چیدمان رکوردها در جدول، ممکن است نتایج متفاوتی را ارائه دهند. به عنوان مثال، اگر محصولات به ترتیب صعودی قیمتشان در جدول چیده شده باشند، در هر رکورد، آن محصول به عنوان محصول با بیشینه قیمت شناخته می شود. لذا نتیجه دستور SQL شامل کل جدول خواهد بود. اگر رکوردها به ترتیب نزولی قیمتشان چیده شده باشند، تنها اولین کالا یا کالاهای اول جدول که بیشترین قیمت را دارند، بازگردانده خواهند شد.

۶. فرض کنید جداول زیر را در اختیار داریم:

Loan_number	Branch_name	Amount
L-170	Shariati	3000
L-230	Kaj	4000
L-260	Ferdos	1700

loan

Customer_name	Loan_number
Ahmadi	L-170
Gholami	L-230
Mohammadi	L-155

borrower

الف) عبارت SQL ای بنویسید که حاصل آن جدول زیر باشد:

Loan_number	Branch_name	Amount	Customer_name
L-170	Shariati	3000	Ahmadi
L-230	Kaj	4000	Gholami
L-260	Ferdos	1700	Null
L-155	Null	Null	Mohammadi

پاسخ:

ماهیت این رابطه Join است. اما باید جهت الحاق را تعیین نمود.

- با توجه به اینکه در رکورد سوم جدول، مقدار Customer\_name برابر با Null است روشن است که سعی دارد تمام رکوردهای جدول loan حتما در خروجی ظاهر شوند، حتی اگر با هیچ مشتری الحاق نشوند. لذا قطعا LEFT OUTER JOIN است.
- از سوی دیگر رکورد چهارم جدول خروجی شامل Branch\_name و Amount برابر با Null است. یعنی سعی دارد تمام رکوردهای جدول borrower نیز در خروجی ظاهر شود، حتی اگر با هیچ وامی الحاق نشود. بنابراین قطعا RIGHT OUTER JOIN نیز است.

لذا زمانیکه هر دو را دارد، FULL خواهد بود.

```
SELECT *  
FROM loan FULL OUTER JOIN borrower  
USING Loan_number
```

ب-۱) شماره وام افرادی که نام آنها ۶ حرفی است.

پاسخ:

```
SELECT Loan_number  
FROM borrower  
WHERE Custmer_name like '_____' یا WHERE LEN(Custmer_name) = 6
```

ب-۲) نام افرادی که اسامی آنها با M شروع نمی شود.

پاسخ:

```
SELECT Customer_name FROM borrower WHERE Custmer_name not like 'M%' یا  
SUBSTRING (Custmer_name , 1, 1) <> 'M'
```

۷. جداول زیر را در بانک اطلاعاتی در نظر بگیرید:

Sailor (S#, Sname, Rating, Age)

Boat (B#, Bname, Color)

Reserves (S#, B#, Date)

الف) با استفاده از Except، پرسمانی بنویسید که "نام و درجه ملوان‌هایی که همه قایق‌های زرد را رزرو کرده‌اند"، به ما ارائه دهد.

پاسخ:

SELECT S.Sname, S.Rating

FROM Sailor S

WHERE NOT EXISTS

کد تمام قایق‌های زرد  
کد قایق‌های زردی که توسط این ملوان رزرو نشده‌اند  
(SELECT B.B# FROM Boat B WHERE B.Color = 'Yellow')  
EXCEPT  
(SELECT R.B# FROM Reserves R WHERE S.S# = R.S#))  
کد تمام قایق‌های که توسط ملوان با کد S# رزرو شده‌اند

باید نام و درجه ملوان‌هایی انتخاب شود که قایق زرد رزرو شده نداشته باشد. یعنی این جدول برایش موجود نباشد.

ب) بخش الف را بار دیگر با کمک دستور Not Exists بازنویسی کنید.

پاسخ:

SELECT S.Sname, S.Rating

FROM Sailor S

WHERE NOT EXISTS (SELECT B.B#

FROM Boat B

WHERE B.Color = 'Yellow' AND

NOT EXISTS (SELECT R.B#

FROM Reserves R

WHERE S.S# = R.S#

AND R.B# = B.B# '))

آیا قایق B# توسط ملوان S# رزرو شده است

نباید قایقی زردی باشد که ملوان S# آن را رزرو نکرده باشد

۸. در یک سامانه مدیریت کارمندان ، علاوه بر مشخصات هر کارمند، مشخصات یکی از بستگان نزدیک نیز نگهداری می‌شود. با توجه به جداول زیر اجرای کدامیک از دستورات زیر قطعا با Referential Integrity Error مواجه نمی‌شود؟ پاسخ خود را توضیح دهید.

Employee (Fname, Lname, Ssn, Bdate, Address, Gender, Salary, Dno)

Dependent (Essn, Depndent\_Fname, Gender, Bdate, Relationship)

الف) UPDATE Employee

ب) DELETE FROM Dependent

ج) DELETE FROM Employee

د) INSERT INTO Dependent

پاسخ:

از آنجایی که کلید اصلی جدول Employee به عنوان کلید خارجی در جدول Dependent درج شده است، هرگونه حذف یا بروزرسانی جدول Employee ممکن است به نقض جامعیت ارجاع منتهی شود. لذا الف و ج نادرست است. از سوی دیگر، به ازای هریک از بستگان درج شده در جدول Dependent حداقل باید یک کارمند وجود داشته باشد، لذا زمانی که یک وابسته جدید در جدول Dependent درج می‌شود، ممکن است کارمند مربوطه در جدول Employee وجود نداشته باشد. عملیات درج با خطا مواجه شود. اما حذف از جدول Dependent همواره بدون خطا امکان‌پذیر است. زیرا سطرهای این جدول مورد ارجاع جداول دیگر نیست.

۹. پایگاه داده زیر را برای یک دانشگاه در نظر بگیرید:

Course (C#, Cname, Units, Cfield)

TC (T#, C#, Time, Place)

Teacher (T#, Tname, Tfield)

الف) خروجی پرسمان زیر چیست؟

SELECT T#

FROM Teacher

WHERE T# NOT IN (SELECT A.T#

FROM TC A, TC B

WHERE A.T# = B.T# AND A.C# <> B.C#)

پاسخ: کد استادی تمام اساتیدی که حداکثر یک درس تدریس نموده‌اند. SELECT T# FROM Teacher

WHERE T# NOT IN (SELECT A.T#

FROM TC A, TC B

WHERE A.T# = B.T# AND A.C# <> B.C#)



ب) پرسمان بخش الف را با استفاده از Join و Group By باز نویسی کنید.

پاسخ:

```
SELECT T#  
FROM Teacher NATURAL JOIN TC  
GROUP BY T#  
HAVING COUNT(*) < 2
```

ج) کد SQL معادل با پرسمان زیر را بنویسید.

نام مدرسانی که حداقل ۱۰ درس از رشته تحصیلی خود را تدریس کرده‌اند.

پاسخ:

```
SELECT Teacher.Tname  
FROM Teacher, TC, Course  
WHERE Teacher.T# = TC.T# AND Course.C# = TC.C# AND Course.Cfield = Teacher.Tfield  
GROUP BY Teacher.T#  
HAVING COUNT(*) > 10
```

رشته درس با رشته مدرس مشابه باشد

درسه‌ها را براساس کد استادی که آنها را تدریس می‌کند گروه بندی نموده

فقط گروه‌هایی انتخاب می‌شوند که تعدادشان بیش از ۱۰ باشد

۱۰. با توجه به جدول Teacher در سوال ۹، توضیح دهید دستور زیر چه چیزی را بررسی می‌کند؟

```
CREATE ASSERTION check – Teacher CHECK  
(NOT EXISTS ((SELECT * FROM Teacher) EXCEPT (SELECT * FROM Teacher WHERE Tname  
= Tname OR Tfield = Tfield))))
```

پاسخ:

توجه! هیچگاه NULL با هیچ مقداری حتی خود NULL برابر نیست. به همین علت اگر Tname مقدار NULL داشته باشد، هیچگاه Tname = Tname نمی‌شود (یعنی Null = Null نیست)

براین اساس، این دستور چک می‌کند که دو ستون Tname و Tfield همزمان مقدار NULL نداشته باشند.

```
CREATE ASSERTION check – Teacher CHECK  
(NOT EXISTS  
((SELECT * FROM Teacher)  
EXCEPT  
(SELECT * FROM Teacher WHERE Tname = Tname OR Tfield = Tfield))))
```

کلیه سطرهای جدول استاد

سطرهایی از جدول Teacher که ستون Tname یا Tfield مقدار NULL داشته باشند یا هیچکدام NULL نباشد.

وجود نداشته باشد سطری از کل جدول استاد که، که هر دو ستون Tname و Tfield آن NULL باشد

## مروری بر قوانین جامعیت

- **قانون اول جامعیت (جامعیت درون رابطه‌ای یا Intra-Relation Integrity):**  
مطابق این قاعده هر جدول (رابطه) باید به تنهایی درست باشد، یعنی نباید تعریف رابطه در تئوری جبر رابطه‌ای را نقض کند. به عنوان نمونه یک رابطه نباید تاپل تکراری داشته باشد، در یک رابطه نباید تاپل‌هایی با درجه‌های متفاوت وجود داشته باشند، هر رابطه باید حتماً کلید اصلی داشته باشد و مؤلفه‌های متناظر در تاپل‌های یک دامنه باید هم‌دامنه باشند.
- **قانون دوم جامعیت (جامعیت موجودیت یا Entity Integrity):**  
مطابق این قاعده که به **کلید اصلی** مربوط است، هیچ یک از مؤلفه‌های کلید اصلی نباید Null باشند. این قاعده در برخی منابع با کمی بسط **جامعیت دامنه** نیز نامیده می‌شود که مطابق با آن مقادیر همه مؤلفه‌های تاپل‌ها در یک رابطه باید در دامنه تعریف شده برای آن مؤلفه قرار داشته باشند و همچنین کلید اصلی یا بخش‌های مختلف آن نباید مقدار Null داشته باشد.
- **قانون سوم جامعیت (جامعیت ارجاع یا Referential Integrity):**  
این قاعده محدودیت‌هایی را برای **کلید خارجی** مطرح می‌نماید که مطابق با آن باید جامعیت ارجاع هم به صورت ساختاری و هم به صورت محتوایی برقرار باشد.  
در جامعیت ساختاری لازم است اگر ستونی در یک جدول به عنوان کلید خارجی تعریف می‌شود، در همان جدول یا جدول دیگر به عنوان کلید کاندید مطرح باشد.  
در جامعیت ارجاع محتوایی نیز لازم است هرگونه اعمال تغییر روی مقدار کلید خارجی هماهنگ با مقدار کلید کاندید مربوطه در جدول مرجع صورت پذیرد. ضمن آنکه هرگونه تغییر محتوایی کلید کاندید در جدول مرجع هم باید هماهنگ با کلیدهای خارجی مربوطه در جدول ارجاع‌کننده انجام گیرد.

## پایان