

به نام خدا

تمرین دوم پایگاه داده

چمران معینی

۹۹۳۱۰۵۳

## سوال اول

می‌دانیم که عملگر *select* یک عملگر *basic* است. عملگر *intersection* دو عملگر *select* را با یکدیگر ترکیب می‌کند، امام تنها سطرهایی را بازمی‌گرداند که در *select* اول موجود هستند و سطرهایی دقیقاً مشابه آن‌ها، در *select* اول نیز موجود است.

## سوال دوم

کوئری اول را بررسی می‌کنیم. این کوئری، حاصل ضرب کارت‌زین نتیجه‌ی دو کوئری زیر است:

```
SELECT A FROM R WHERE B = 1
```

```
SELECT A FROM S WHERE B = 1
```

یعنی ابتدا از هر دو جدول، سطرهایی را که مقدار ستون *B* مساوی ۱ است، انتخاب می‌کنیم و سپس آن‌ها را با هم ضرب می‌کنیم. اگر در جدول *R* برای مثال، ۲۰ سطر وجود داشته باشد، که از میان این ۲۰ سطر، ۴ سطر باشد که مقدار ستون *B* مساوی ۱ است، و در جدول *S* ۳ سطر باشد که مقدار ستون *B* مساوی ۱ است، این کوئری، ۱۲ سطر را باز می‌گرداند. حال کوئری دوم را بررسی می‌کنیم. در این کوئری، ابتدا ضرب کارت‌زین بین نتیجه‌ی این دو کوئری انجام می‌شود:

```
SELECT A FROM R
```

```
SELECT * FROM S WHERE B = 1
```

یعنی ابتدا از جدول *R* همه‌ی سطرها را انتخاب می‌کنیم و سپس از جدول *S* سطرهایی را که مقدار ستون *B* مساوی ۱ است، انتخاب می‌کنیم. اگر در جدول *R* ۴ سطر باشد و در جدول *S* ۳ سطر باشد، این کوئری، ۱۲ سطر را باز می‌گرداند.

در نهایت، یک عملگر *Project* هم روی نتیجه اعمال شده که ستون‌های *A* و *C* را بازمی‌گرداند. بر اساس فرضی که در بخش قبلی مطرح کردیم، در جدول *R* ۲۰ سطر وجود داشت. همچنین در جدول *S* ۳ سطر وجود داشت که مقدار ستون *B* مساوی ۱ باشد. در نتیجه، از ضرب کارت‌زین این دو، ۶۰ سطر به دست می‌آید. در نهایت، ستون‌های *A* و *C* از این ۶۰ سطر را داریم. در حالی که در کوئری قبلی، تنها ستون *A* از ۱۲ سطر را داشتیم. واضح است که این دو کوئری نتایج متفاوتی را بازمی‌گردانند. حال سومین کوئری را بررسی می‌کنیم.

در این کوئری ابتدا سطرهایی از جدول *S* انتخاب شده‌اند که در آن‌ها مقدار ستون *B* مساوی ۱ است.

سپس بین حاصل و کل جدول *R* عملگر *Natural Join* اعمال می‌شود. این عملگر، ابتدا ستون‌های مشترک را پیدا می‌کند و سپس سطرهایی را از جدول *R* انتخاب می‌کند که مقدار ستون‌های مشترک با سطرهای انتخاب شده از جدول *S* مساوی باشد. در این کوئری، ستون مشترک *B* است، پس مقادیری

بازگردانده می‌شوند که در آن‌ها مقدار  $B$  در هر دو جدول مساوی ۱ باشد. در نهایت هم یک  $Project$  روی نتیجه اعمال می‌شود که ستون‌های  $A$  و  $C$  را باز می‌گرداند.

با فرض‌هایی که در بخش اول در نظر گرفتیم، این کوئری را نیز بررسی می‌کنیم. از اولین کوئری، تعداد ۳ سطر بازمی‌گردد.

هنگامی که روی این ۳ سطر و کل جدول  $R$  عملگر  $NaturalJoin$  اعمال می‌شود، از آن جایی که در سه سطر انتخاب شده، مقدار ستون  $B$  مساوی ۱ است، از آن جایی که در کل جدول  $R$  هم در ۴ ستون این شرط برقرار است، نهایتاً ۱۲ سطر بازمی‌گردد و سپس ستون‌های  $A$  و  $C$  از آن‌ها باز می‌گردد. ظاهراً این کوئری، مشابه کوئری اول عمل می‌کند.

حال برای اطمینان، عملگرهای آنان را بررسی می‌کنیم.

در هر دو کوئری، ابتدا سطرهایی که در آن‌ها مقدار  $B$  مساوی ۱ است، از جدول  $S$  انتخاب شده است. در کوئری اول سپس یک ضرب کارتیزین میان این سطرها و سطرهایی از جدول  $R$  انجام می‌شود که مقدار ستون  $B$  در آن‌ها هم مساوی ۱ باشد.

این دقیقاً همان اتفاقی است که در کوئری سوم رخ می‌دهد هنگامی که عملگر  $NaturalJoin$  اعمال می‌شود.

## سوال سوم

(الف)

در کوئری اول، ابتدا تمام سطرهایی که در آن‌ها مقدار  $salary$  بزرگتر مساوی ۹۰۰۰۰ است انتخاب می‌شوند. سپس مقادیر ستون‌های  $name$  و  $deptname$  از آن‌ها باز می‌گردد. پس نتیجه‌ی نهایی، جدولی به این شکل خواهد بود:

name	deptname
Einstein	Physics
Wu	Finance
Brandt	Comp. Sci.

اما در کوئری دوم، ابتدا ستون‌های  $deptname$  و  $salary$  با عملگر  $Project$  انتخاب می‌شوند. سپس کوئری روی آن‌ها زده می‌شود تا سطرهایی که در آن‌ها مقدار  $salary$  بزرگتر مساوی ۹۰۰۰۰ است انتخاب شوند. با توجه به این که در دیتایی که به این کوئری داده می‌شود، دیگر ستون  $salary$  وجود ندارد، نتیجه‌ی نهایی، جدولی بدون سطر خواهد بود.

(ب)

دیدیم که در کوئری دوم، نتیجه‌ای نداشتیم. پس هنگامی که قرار است عملگر  $Select$  اعمال شود، باید دقت داشت که از چه فیلدهایی استفاده می‌کند، چون اگر از فیلدی استفاده کنیم که در جدول‌هایی

---

که قبلا انتخاب شده‌اند، وجود نداشته باشد، نتیجه‌ی نهایی، جدولی خالی خواهد بود. به طور کلی بهتر است ابتدا عملگر *Select* اعمال شود و سپس عملگر *Project* اعمال شود.

## سوال چهارم

جدول  $T1$  به این شکل است:

A	B
a1	b1
a2	b2
a3	b3

جدول  $T2$  به این شکل است:

B	C
b1	c1
b1	c2
b2	c2
b2	c3
b2	c4

(الف)

حاصل عملگر *Natural Join* بین  $T1$  و  $T2$  از ما خواسته شده است، که به این شکل است:

A	B	C
a1	b1	c1
a1	b1	c2
a2	b2	c2
a2	b2	c3
a2	b2	c4

در دومین کوئری ابتدا عملگر *Project* ستون  $B$  را از جدول‌ها انتخاب می‌کند. حاصل، این دو ستون خواهند بود:

---

B
b1
b2
b3
B
b1
b1
b2
b2
b2

سپس عملگر *setintersection* بین این دو حاصل، اعمال می‌شود و نتیجه‌ی نهایی، به این شکل خواهد بود:

B
b1
b2

(ب)

عملگرهای *NaturalJoin* و *intersection* کاملاً عملگرهای متفاوتی هستند. با عملگر *NaturalJoin* پس از یافتن سطریهایی که مقدارِ سطرهایشان در ستون‌های مشترک، یکی‌ست، ضربِ کارتیزین بین این داده‌ها انجام می‌شود. اما در عملگر *intersection* به نوعی اشتراک گرفته می‌شود و سطریهایی که در هر دو جدول دقیقاً یکی هستند، بازگردانده می‌شوند.

## سوال پنجم

جدول *Student*

---

ID	Name	Nationality
1	Jon Snow	USA
2	Jame Bond	UK
3	Winston Churchill	USA
4	John F. Kennedy	USA
5	Jakie Chan	China
6	Richard White	USA
7	Bruce Lee	USA
8	Hugo Lafayette	France
9	Ben Kenobi	USA
10	Harry Potter	UK
11	Son Goku	Japan
12	Wonder Woman	UK
13	Sun Tzu	China
14	Tony Stark	USA
15	Leia Organa	USA

جدول *Enrollment* به این شکل است:

StudentID	CourseID	Grade	SectionNum	GroupID
1	CS448	A	2	3
4	CS448	A	1	2
5	CS448	B	1	1
6	CS448	A	1	1
9	CS448	B	2	3
10	CS448	A	2	4
11	CS448	C	2	4
12	CS448	A	2	3
13	CS448	A	1	1
2	CS580	A	1	1
3	CS580	A	1	1
4	CS580	B	1	2
6	CS580	A	1	2
8	CS580	A	1	2
10	CS580	A	1	1
12	CS580	B	2	3
15	CS580	A	2	3

جدول *Course* به این شکل است:

CourseID	InstructorID	Name
CS448	7	Introduction to Relational Database Systems
CS390	2	Linear Algebra
CS580	14	Algorithm Design, Analysis, And Implementation

(۱) ابتدا از *Join* جدول *Enrollment* و *Course* دیتاهای دانشجوی شماره‌ی ۵ انتخاب می‌شود، سپس ستون‌های *CourseID* و *Grade* انتخاب می‌شود.

CourseID	Grade
CS448	B

(۲) ابتدا سطرهایی از جدول *Enrollment* که در آن‌ها *StudentID* برابر با ۵ است انتخاب می‌شوند که تنها یک سطر این ویژگی را دارد. سپس با جدول *Course*، *Join* می‌شوند، که طی آن به سطری که در مرحله‌ی قبل انتخاب شده بود، ستون‌هایی اضافه می‌شوند و در نهایت، *CourseID* و *Grade* انتخاب می‌شود.

StudentID	InstructorID	Name
5	7	Introduction to Relational Database Systems

(۳) ابتدا جدول‌های *Enrollment* و *Course* با یکدیگر ضرب کارت‌زین می‌شوند، که هریک از سطرهای جدول *Enrollment* با همه‌ی سطرهای جدول *Course* ضرب کارت‌زین می‌شوند. سپس سطرهایی انتخاب می‌شود که در آن‌ها *StudentID* برابر با ۵ است، که ۳ سطری خواهند بود که حاصل ضرب سطری از *Enrollment* با ۳ سطر *Course* خواهد بود. سپس ستون‌های *CourseID* و *Grade* انتخاب می‌شود.

CourseID	Grade
CS448	B
CS448	B
CS448	B

(۴) با توجه به اصلاحات مطرح شده، از راست به چپ شروع می‌کنیم. ابتدا جدول *Course* را با *Enrollment* بر اساس تنها فیلد مشترکشان یعنی *CourseID*، *Join* می‌کنیم. سپس حاصل را با جدول *Students* بر اساس فیلد *StudentID*، *Join* می‌کنیم. سپس سطرهایی را انتخاب می‌کنیم که در آن‌ها *StudentID* برابر با ۱۰ است. سپس از حاصل، ستون‌های *Students.Name* و *Course.Name* انتخاب می‌شود. خروجی نهایی، نام دانشجویی‌ست که آی‌دی او ۱۰ است، و نام دوره‌هایی که دارد.

Students.Name	Course.Name
Harry Potter	Introduction to Relational Database Systems
Harry Potter	Algorithm Design, Analysis, And Implementation



---

## سوال ششم

(۱)

$$Pets \bowtie_{Pets.OwnerID=Owners.OwnerID} Owners$$

(۲)

$$res : ProcedureHistory \bowtie_{ProcedureHistory.PetID=Pets.PetID} Pets$$

$$\prod_{PetID, Name, Kind, Age, ProcedureType, ProcedureSubCode} (res)$$

(۳)

$$res1 : ProcedureHistory \bowtie_{ProcedureHistory.PetID=Pets.PetID} Pets$$

$$query1 : ProcedureHistory.ProcedureType = ProcedureDetails.ProcedureType$$

$$query2 : ProcedureHistory.ProcedureSubCode = ProcedureDetails.ProcedureSubCode$$

$$res2 : res1 \bowtie_{query1 \& query2} ProcedureDetails$$

$$finalRes : \prod_{Pets.Name, ProcedureDetails.Description} (res2)$$