

به نام خدا

# پاسخنامه‌ی تمرین دوم درس طراحی پایگاه داده

نیم‌سال اول ۱۴۰۱-۱۴۰۲

## سوال اول

- Intersection:

$$A \cap B = A - (A - B)$$

- Natural Join:

$$A \bowtie B = \Pi_{A1, A2, \dots, An, B1, B2, \dots, Bm, x1, x2, \dots, xk} (\sigma_{A.X1=B.X1, A.X2=B.X2, \dots, A.Xp=B.Xp} (A \times B))$$

- Division:

$$\Pi_{R-S}(r) = \Pi_{R-S}(r) [\Pi_{R-S}(r) * s = \Pi_{R-S,S}(r)]$$

- Outer Join:

$$(R \bowtie S) \cup ((R - \Pi_R(R \bowtie S)) \times \{(\text{null}, \text{null}, \dots, \text{null})\}) \cup \{(\text{null}, \text{null}, \dots, \text{null})\} \times (S - \Pi_S(S \bowtie R)))$$

## سوال دوم

برای جواب به این مسئله این مثال را در نظر داشته باشید:

R

A	B
1	1
2	2

S

B	C
1	2
2	3

پرس و جوی اول

ابتدا از رابطه R ، کلید A سطر هایی را برمی داریم که  $B=1$  دارند ( $R_t$ )، سپس از ستون C سطر هایی را برمی داریم که  $B=1$  دارند ( $S_t$ ). که نتیجه آن ها هر کدام یک جدول با یک سطر و یک ستون است:

$R_t$

A
1

$S_t$

C
2

حال آن دو را باهم با ضرب دکارتی یکی میکنیم که جواب نهایی (F) به صورت زیر است:

F

A	C
1	2

پرس و جوی دوم

ما مقادیر تمام سطر های ستون A از رابطه R را در کل مقادیری که در جدول S دارای  $B=1$  اند، ضرب دکارتی میکنیم. و سپس ستون های A,C را از جدول حاصل جدا میکنیم. نتیجه حاصل از این پرس و جو به شرح زیر است (G):

G

A	C
1	2
2	2

میبینیم که خروجی این دو پرس و جو باهم برابر نیستند.

بخش دوم سوال رابطه ای که میبینیم حاصل از انتخاب ستون های A,C از رابطه ای است که از natural join کردن رابطه R با سطر هایی از رابطه S است که  $B=1$  دارند که خروجی آن دقیقاً با خروجی پرس و جوی مورد اول یکسان است.

## سوال سوم

مورد الف

a)

Name	Dept_name
Einstein	Physics
Wu	Finance
Brandt	Copm. Sci.

b) Error, no result

توجه داشته باشید که جدول خالی به صورت زیر قابل قبول نیست زیرا error به معنای این است که پرسجو اصلا قابل تحلیل کردن نیست و بی معناست اما امکان دارد که پرسجو کاملا بی ایراد باشد اما خروجی آن یک جدول خالی باشد.

name	dept_name

مورد ب

$\pi$	$\sigma$
یک جدول (relation) برمیگرداند با حداکثر n اتریبیوت که تاپلهای آن مقادیر اتریبیوت های ذکر شده در شرط عملگر هستند.	یک جدول (relation) برمیگرداند با n اتریبیوت و تاپلهایی که شرط گفته شده را مهیا سازند

با توجه به توضیحات بالا می دانیم جدول حاصل از عملگر  $\sigma$  میتواند جامع تر از عملگر  $\pi$  باشد ولی عکس این حرف درست نیست.

میتوان گفت که انجام عمل projection روی جدولی که حاصل از عمل selection است، همیشه جواب دارد (البته اگر حاصل selection، تهی نباشد).

اما selection روی جدولی که projection شده است میتواند بی جواب باشد (به علت داشتن شرطی راجع اتریبیوتی که وجود ندارد).  
همچنین اگر، عملیات selection on projection به گونه ای باشد که اتریبیوتی که در select انتخاب می کنیم، در projection نیز وجود داشته باشد، با هیچ خطایی مواجه نخواهیم شد و نتیجه را در قالب جدول نهایی گزارش خواهیم کرد.  
برای مثال مورد آخر می توان به کوئری زیر اشاره کرد:

$\sigma_{name = "Einstien"} (\Pi_{name, dept\_name, salary}(Instructor))$

که نتیجه اش به شکل جدول زیر است:

Name	Dept_name	Salary
Einstien	Physics	۹۵۰۰۰

## سوال چهارم

مورد الف

a)  $T1 \bowtie T2$

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

b)  $\Pi_{T1.B} \cap \Pi_{T2.B}$

B
b1
b1

## مورد ب

تعریف natural join به صورت زیر می باشد:

$$r \bowtie s = \Pi_{R \cup S} (\sigma_{r.A_1=s.A_1 \wedge r.A_2=s.A_2 \wedge \dots \wedge r.A_n=s.A_n} (r \times s))$$

$$\text{where } R \cap S = \{A_1, A_2, \dots, A_n\}$$

به طور عامیانه تر، می توان گفت، "اترپیوت های مشترکی که مقدار یکسانی دارند در کنار مقادیر اترپیوت های غیر مشترک" از لحاظ تعداد ستون ها نیز میتوان گفت بزرگتر مساوی ماکسیمم تعداد ستون ها بین دو جدول، ستون خواهیم داشت.

$$n \geq \max(\text{attr}(T1), \text{attr}(T2))$$

عملیات intersect را نیز می توان به شکل زیر تعریف کرد:

$$r \cap s = r - (r - s)$$

برای گرفتن intersect بین دو جدول، جداول باید تماماً اترپیوت های مشترکی داشته باشند یا اگر تعدادی اترپیوت مشترک اند، با عملیات projection آنها را جدا و جدولی جدید ایجاد کنیم. همچنین از لحاظ تعداد ستون ها نیز میتوان گفت کوچکتر مساوی مینیمم تعداد ستون ها بین دو جدول، ستون خواهیم داشت.

$$n \leq \min(\text{attr}(T1), \text{attr}(T2))$$

\* هرگونه توضیح دیگری که به تعریف عملگر ها و تفاوت میانشان بپردازد، قابل قبول می باشد



## سوال پنجم

### قسمت اول

CourseID	Grade
CS448	B

این پرس و جو پس از پیوند طبیعی بین Courses , Enrollment با فیلد مشترک CourseID , دانشجویان با شماره دانشجویی 5 انتخاب میشوند و از بین فیلد های موجود فیلد های CourseID , Grade نمایش داده میشوند.

در واقع این پرس و جو (اشاره دارد به دانشجویانی که در کورس ها شرکت کرده اند یا) , CourseID , Grade دانشجویانی که ID آن ها 5 است را بر میگرداند.

### قسمت دوم

CourseID	Grade
CS448	B

این پرس و جو ابتدا دانشجویان با  $ID = 5$  را از جدول Enrollment انتخاب کرده و پس از پیوند طبیعی حاصل از این جدول و جدول Courses , از بین فیلد های موجود فیلد های CourseID , Grade نمایش داده میشوند.

در واقع این پرس و جو CourseID , Grade دانشجویانی که ID آن ها 5 است و در کورس نیز شرکت داشته اند را بر میگرداند.

### قسمت سوم

Enrollment.CourseID	Grade
CS448	B

این پرس و جو پس از ضرب کارتزین بین Courses , Enrollment , که تمام فیلد های دو جدول را شامل میشود , دانشجویان با شماره دانشجویی 5 انتخاب میشوند و از بین فیلد های موجود فیلد های CourseID , Grade نمایش داده میشوند.

در واقع این پرس و جو نیز همان CourseID , Grade دانشجویانی که ID آن ها 5 است را بر میگرداند.

#### قسمت چهارم

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

در این پرس و جو ابتدا پیوند طبیعی بین سه جدول Students, Enrollments, Courses با فیلد های مشترک StudentID و CourseID انجام میشود و از بین سطر ها دانشجویان با شماره دانشجویی 10 انتخاب میشود و سپس از بین فیلد ها Students.Name و Courses.Name نمایش داده میشوند.

به طور کلی این پرس و جو نام دانشجو (Students.Name) و کورس هایی که در آن شرکت داشته (Courses.Name) مربوط به دانشجویانی با ID 10 را بر میگرداند.

## سوال ششم

### مورد اول

در خروجی پرس‌جوی داده شده نیاز به اطلاعات Pets در کنار Owners داریم و همچنین در پرس‌جو بیان شده است که تمام مشخصات هر دو جدول را باید در کنار هم داشته باشیم به همین دلیل دیگر هیچ projection یا پرتویی نیاز به انجام نیست.  
پس query به شکل زیر خواهد بود:

$Pets \bowtie Owners$

### مورد دوم

در این مورد خروجی Pet.PetID و Pet.Name و Pet.Kind و Pet.Age را همراه با Procedure\_history.type و Procedure\_history.subcode نیاز داریم.  
پس برای همین نیاز داریم که ابتدا دو رابطه ی pets و procedure\_details را باهم join کنیم و در نهایت آن ستون های مورد نیاز را از جدول نهایی بیرون بکشیم:

$$\Pi_{Subcode, Type, PetID, Name, Kind, Age} (Pets \times_{procedure\_history.PetID = Pets.PetID} Procedure\_history)$$

حال برای ساده تر سازی این پرس‌جو می‌توانیم این گونه استدلال کنیم که چون صرفاً PetID بین هر دو جدول مشترک است و PetID در join شرکت می‌کند پس می‌توان از natural-join نیز استفاده کرد.

$$\Pi_{Subcode, Type, PetID, Name, Kind, Age} (Pets \bowtie Procedure\_history)$$

**توجه داشته باشید که:** در اینجا می‌توان ابتدا از Pets یک projection گرفت و بعد از procedure\_history نیز یک projection گرفت و بعد حاصل را باهم join کرد و باز هم به همین نتیجه رسید.

$$(\Pi_{PetID, Name, Kind, Age} (Pets)) \bowtie (\Pi_{PetID, Subcode, Type} (Pets))$$

### مورد سوم

در خروجی این پرس‌جو نیاز به Pets.PetID و Pets.Name و Procedure\_details.Description داریم.

حال برای اینکه Pets را به Procedure\_details ربط بدهیم اول باید بدانیم که بر روی هر Pet چه عملی انجام شده است و بعد بدانیم که جزییات آن عمل چگونه بوده است. پس به همین دلیل نیاز داریم که ابتدا Pets را با Procedure\_history یک join بزنیم و حاصل آن را با Procedure\_details یک join بزنیم و در نهایت یک projection از جدول حاصل بگیریم.

$$\Pi_{PetID, Name, Description} (Pets \bowtie Procedure\_history \bowtie Procedure\_details)$$