به نام آنکه جان را فکرت آموخت

چراغ دل به نور جان برافروخت

پایگاه داده ها (بانک اطلاعاتی)

نوع درس : تئوری

دوره : کارشناسی

رشته : مهندسی کامپیوتر

تعداد واحد : ۳

مدرس : علیرضا عظیمی

1-3/3- گزینش و پرتو

"گزینش" سطرهایی را از جدول انتخاب می کند. نام جدول جلو علامت گـزینش در پرانتز و شرط انتخاب زیر آن می آید. همه ستون های آن جدول در خروجی می آید.

مثال : تمام ستون های جدول دانشجو که شهر آنها کلمـه یـزد را نـشان مـی دهـد و شماره دانشکده آن 4 است.

دستور

$$\sigma_{CITY} = \text{"یزد"} \quad \wedge c\lg\# = 4(stud)$$

خروجی :

S#	Sname	City	Avg	Clg#
73120504	کمانی	یزد	17.56	4

"پرتو" ستون هایی از جداول را انتخاب می کند و هیچگونه شرطی اعمال نمی شود. چون خروجی یک رابطه است و رابطه از تئـوری مجموعـه هـا پیـروی مـی کنـد و در مجموعه عضو تکراری معنی ندارد، بنابراین سطرهای تکراری خروجی پرتو حـذف مـی شوند. نام جدول جلو علامت پرتو درپرانتز و ستون های انتخاب شده زیر آن می آید.

مثال : ستونهای شماره دانشجو، نام دانشجو و کد دانشکده از جدول دانشجو.

دستور :

$$\Pi_{s\#,sname,c\lg\#}(stud)$$

خروجی :

S#	Sname	Clg#
71133848	محمدی	10
72130502	وکیلی	10
72203305	علینقی زاده	1
73120504	کمانی	4
73166801	احمدی	5
74182532	جوادی	5
74209836	حسین زاده	6

مثال : شهر های مختلف محل تولد دانشجویان

دستور

$$\Pi_{city}(stud)$$

خروجی:

City
تهران
اصفهان
مشهد
یزد
کرمان
تبریز

خروجی مثال دوم جدولی است با یک ستون که نام شهرهای محل تولد دانشجویان را بدون تکرار می دهد .

از **ترکیب** گزینش و پرتو می توان اطلاعات بیشتری از جداول به دست آورد.

مثال : ستون های شماره دانشجویی، نام، کـد دانـشکده و میـانگین دانـشجویانی کـه معدل آنها بالای 15 است.

دستور :

$$\prod s_{\#,sname,c\lg\#,avg}(\sigma_{avg>15}(stud))$$

دستور معادل :

$$\sigma_{avg>15}(\prod_{s\#,sname,c\lg\#,avg}(stud))$$

خروجی :

S#	Sname	avg	Clg#
71133848	محمدی	12.24	10
72203305	علینقی زاده	16.42	1
74182532	جوادی	16.8	5
73166801	احمدی	15.44	5
73120504	کمانی	17.56	4

2-3/3- عملگر های مجموعه ای

عملگر های اجتماعی، اشتراک و تفاضل همان معنای خود در تئوری مجموعه را حفظ کرده اند. ورودی هر کدام دو رابطه و خروجی آنها یک رابطه است. روابط ورودی بایـد همتا (***same arity***) باشند یعنی :

- تعداد صفتهای دو رابطه (ستون های دو جدول) مساوی باشد.
- صفتها به ترتیب دارای دامنه های یکسان باشند.

مثال : لیست نام همه افرادی که در دانشکده ها هستند.

حل : این افراد یا استاد هستند یا دانشجو. بنابراین باید نام دانشجویان و نیز نام اسـتاد را جداگانه لیست و سپس با هم اجتماع کرد (اسامی تکراری حذف خواهد شد).

دستور :

$$\Pi_{sname}(stud) \cup \Pi_{pname}(prof)$$

خروجی :

name
احمدی
جوادی
حسین زاده
علینقی زاده
کمانی

ابوطالبی
حسنی
شیدفر
قربانی
میر شمسی
جاهد مطلق

نقره کار
ذاکر
مفتون
هاشمی اصل
صادقیان

محمدی
وکیلی
احمدیان
اشرفی زاده
جلالی

مثال : لیست نام اساتیدی که رئیس دانشکده نیستند.

حل : ابتدا نام اساتید را پیدا می کنیم و سپس نام رؤسای دانشکده ها را از آن تفریـق می کنیم، یعنی

$$\Pi_{pname}(prof) - \Pi_{pname}(c\lg)$$

خروجی

Pname
قربانی
هاشمی اصل
شیدفر
میر شمسی
احمدیان

مثال : لیست اسامی دانشجویان و اساتید همنام .

حل : کافیست اشتراک اسامی دانشجویان و اساتید را پیدا کنیم ، یعنی

$$\prod_{sname}(stud) \cap \prod_{pname}(prof)$$

نکاتی در رابطه با عملگر های مجموعه ای قابل ذکر است:

- این عملگر ها را معمولاً نمی توان با عملگر های دیگر جایگزین کرد مثلاً نمی تـوان آنچه را اجتماع بدست می آورد از طریق دیگری پیدا کرد.
- عملگر های مجموعه ای به تنهایی از توان محاسباتی کافی برخـوردار نیـستند مـثلاً نمی توانیم نام و مدرک تحصیلی اساتیدی که مسئولیت ندارند را لیست کنیم زیرا در جدول دانشکده مدرک تحصیلی ذکر نشده.
- ترتیب جداول ورودی فقط در تفاضل مهم است . به عبارت دیگر

$x \cup y = y \cup x, x \cap y = y \cap x$ ولی $x - y \neq y - x$

3-3/3- عملگر های پیوند

این عملگر ها بسیار قدرتمند و پرکاربرد هستند در استفاده از آنها باید دقت کرد زیـرا بعضاً بسیار گران تمام می شوند (زمان و فضای زیادی می گیرد) و نوعاً می توان آنها را با یکدیگر جایگزین نمود یعنی پرس و جوها را می توان به روش ها ی مختلف پاسخ داد.

الف – ضرب دکارتی

ضرب دکارتی یکی از گرانترین عملگر های بانک رابطه ای است و باید حتی الامکان از آن پرهیز کرد بنابراین در مورادی در پاسخ گویی به بعضی پرس و جوهـا مجبـوریم از

آن استفاده کنیم. ضرب دکارتی دو جدول ، جدولی است که ستون هایش همه ستون های دو جدول و سطرهایش تمام ترکیب های ممکن سطرهای آن دو جدول است اگر دو جدول ستون های همنام داشته باشند در خروجی از نقطه گذاری بـرای تـشخیص آنها استفاده می شود (مثلا clg.clg # , crs .clg #)

مثال : crs × clg

توجه: در اینجا به دلیل بزرگ بودن خروجی از آوردن کل جدول خودداری شده است .

c#	cname	unit	Crs.clg#	Clg.clg#	clgname	city	pname
71203	کنترل خطی	3	7	1	ریاضی	تهران	حسنی
71203	کنترل خطی	3	7	2	فیزیک	مشهد	ذاکر
71203	کنترل خطی	3	7	3	زبان	مشهد	مفتون
71203	کنترل خطی	3	7	4	صنایع	تهران	صادقیان
71203	کنترل خطی	3	7	5	شیمی	تهران	اشرفی زاده
71203	کنترل خطی	3	7	6	مواد	تبریز	ابوطالبی
71203	کنترل خطی	3	7	7	برق	تهران	جلالی
71203	کنترل خطی	3	7	10	کامپیوتر	تهران	جاهد مطلق
71203	کنترل خطی	3	7	11	معماری	یزد	نقره کار
71203	کنترل خطی	3	7	12	معارف	کرج	حبیبی
10172	شبیه سازی	3	10	1	ریاضی	تهران	حسنی
10172	شبیه سازی	3	10	2	فیزیک	مشهد	ذاکر
10172	شبیه سازی	3	10	3	زبان	مشهد	مفتون
10172	شبیه سازی	3	10	4	صنایع	تهران	صادقیان
10172	شبیه سازی	3	10	5	شیمی	تهران	اشرفی زاده
10172	شبیه سازی	3	10	6	مواد	تبریز	ابوطالبی
10172	شبیه سازی	3	10	7	برق	تهران	جلالی
10172	شبیه سازی	3	10	10	کامپیوتر	تهران	جاهد مطلق
10172	شبیه سازی	3	10	11	معماری	یزد	نقره کار
10172	شبیه سازی	3	10	12	معارف	کرج	حبیبی

ب- پیوند شرطی (*teta join*)

این عملگر، زیر مجموعه ای از ضرب دکارتی است که شـرط θ روی سـطر هـای آن اعمال شده باشد. ستون های خروجی معادل ستون های ضرب دکارتی است.

مثال : نام و شماره دروسی که توسط استاد قربانی ارائه شده است.

حل : نام و شماره دروس در جدول درس آمده. این جـدول بـا جـدول اسـتاد ارتبـاط ندارد (هیچکدام کلید خارجی در دیگری ندارد). اما با جدول گـروه درس ارتبـاط دارد. پس می توان از پیوند شرطی این دو جدول استفاده کرد، به صورت زیر :

$$\prod_{cname,crs.c\#}(crs\ X_{pname=\text{"قرباني"} \wedge crs.c\#=\text{sec}.c\#}\ \text{sec})$$

C#	Cname
10172	شبیه سازی

این راه حل بسیار گران است، زیرا جدول بزرگی شامل همه ترکیبهای ممکن سطری و ستونی دو جدول تشکیل می گردد و سپس بخش کوچکی از آن بـه عنـوان خروجـی انتخاب و بقیه دور ریخته می شود. این مشکل را می توان با استفاده از پیوند طبیعی و دستور جایگزینی حل کرد. پیوند طبیعی در زیر می آید و دستور جـایگزینی در جـای خود مورد بررسی قرار خواهد گرفت.

ج – پیوند طبیعی (*natural join*)

پیوند طبیعی از عملگر های اصلی جبر رابطه ای نیست ولی از معروفترین و کارآمدترین آنهاست. این پیوند شرطی است، با تفاوتهای زیر:

1- خودبخود شرط "تساوی" روی همه ستون های همنام دو جدول اعمال می شود یعنی فقط سطرهایی از دو جدول انتخاب می شود که همه ستون های همنام آن دو جدول مقادیر مساوی داشته باشند در صورتی دو جدول ستون همنام نداشته باشد نتیجه پیوند طبیعی معادل ضرب دکارتی است.

2- ستون های تکراری فقط یکبار در خروجی ظاهر می شود از آنجا که فقط مقادیر مساوی آنها انتخاب می شود نیازی به تکرار ستون یانقطه گذاری (مثلاً sec.C#) نیست.

3- نرم افزار هایی شبیه SQL هیچ گاه جدول ترکیبی را تشکیل نمی دهد تا سطر های هم تراز جدا کنند بلکه با استفاده از روش هایی مانند مرتب کردن و شاخص گذاری کار را بسیار ساده می کنند. نتیجه اینکه پیوند طبیعی عملگر گرانی نیست و زمان و فضای آن قابل مقایسه با ضرب دکارتی نمی باشد.

مثال : همان مثال پیوند شرطی را تکرار می کنیم : نام و شماره دروسی که توسط استاد قربانی ارائه شده است.

حل : ابتدا شماره دروس استاد قربانی را از جدول گروه درس انتخاب می کنیم سپس شماره و نام همه درس ها را از جدول درس جدا می کنیم آنگاه آنها را پیوند طبیعی می دهیم به صورت زیر

$$\prod_{c\#\sigma pname = \text{"قربانی"}} (\text{sec})) \infty (\prod_{c\#cname} (crs))$$

مثال : مشخصات کامل رؤسای دانشکده ها

دستور : $(\prod_{pname}(c\lg))\infty prof$

خروجی :

pname	Office	esp	Degree	Clg#
حسنی	2	ریاضی	دکتری	1
جلالی	5	برق	دکتری	7
اشرفی زاده	8	شیمی	دکتری	5

مثال : خروجی دستور $stud\infty c\lg$ چیست ؟

حل : در نگاه اولیه به نظر می رسد که این دستور مشخصات دانشجویان و دانشکده آنها را بدهد زیرا صفت $c\lg\#$ آنها را پیوند طبیعی می دهد. با دقت بیشتر ملاحظه می شود که این برداشت غلط است زیرا صفت مشترک دیگری هم هست و آن صفت city است. یادآوری می شویم که در پیوند طبیعی مساوی بودن همه صفتهای

مشترک بررسی می شود. پس این دستور مشخصات دانشجویان و دانشکده کسانی را می دهد که در همان شهر دانشکده خود به دنیا آمده اند .

مثال : خروجی دستور $prof \infty clg$ چیست ؟

حل : هر یک از این دو جدول یک کلید خارجی در دیگری دارد. بنابراین در پیوند طبیعی آنها شرط تساوی هر دو ستون رعایت می شود. چون ستون pname در جدول clg مربوط به رئیس دانشکده است پس دستور فوق مشخصات کامل دانشکده ها روسای آنها را می دهد.

مثال : نام دانشجویانی که در دروس دانشکده های دیگر نیفتاده اند .

حل : به هیچ وجه نمی توان از دستوری شبیه ($\prod..\sigma..stud \infty crs \infty sec$) استفاده نمود زیرا وجود ستون های همنام clg# در دو جدول stud و crs باعث حذف دروس دانشکده های دیگر می شود . دستور زیر نیز غلط است !(چرا ؟)

$$\prod_{sname} \sigma_{score>10}((stu_{dstud.clg\#} X_{\neq crs.clg\#} crs) \infty sec)$$

د- **نیم پیوند** (***semi – join***)

این عملگر مشابه پیوند طبیعی است با این تفاوت که فقط ستونهای جدول اول را می دهد .

مثال خروجی دستور زیر چیست ؟

$$\sigma_{clg\#=1 \wedge term=771} (crs \infty sec)$$

حل : ظاهراً این دستور مشخصات دروسی را می دهد (بدون مشخصات گروه آنها کـه در ترم اول سال 77 (کد 771) در دانشکده 1 ارائه می شود ولی واقعاً چنین نیست. دستور فوق **غلط** است زیرا ستون term مربوط به جـدول crs نیـست و پـس از نـیم پیوند حذف می شود. پس نمی توان آن را در شرط گنجاند.

یکی از پاسخ های صحیح چنین است:

$$(\sigma_{c\lg\#=1} crs)?(\sigma_{term=771}(\text{sec}))$$

در رابطه با نیم پیوند باید به نکات زیر توجه کرد :

- ممکن است تعداد سطر های خروجی به مراتب کمتر از پیوند طبیعی باشد زیرا بـا کنار رفتن چند ستون ، سطرهای تکراری پدید می آیند و حذف می شوند .
- بر خلاف سایر عملگر های این بخش ، ترتیب جداول ورودی در نـیم پیونـد مهـم است $(x \propto y \neq y?x)$ زیرا همیشه ستون های جدول اول را می دهد .

کاربرد نیم پیوند در بانکهای نامتمرکز است ، اگر جدول x در سایت الـف و جـدول y در سایت ب ذخیره شده باشد و دستور $x \propto y$ را در سایت الـف صـادر کنـیم ، ضـمن اینکه ارزش پیوند طبیعی را دارد از انتقال داده های جدول y از سـایت ب بـه سـایت الف پرهیز می کند .

4-3/3- عملگر های دیگر

الف – **نامگذاری :** با علامت ρ_b^a ، نام b روی جدول a نیزگذاشته مـی شـود . ارزش عملیاتی آن این است که بدون ذخیره سازی مجدد یک جدول می تـوان از آن دو بـار استفاده کرد (در واقع اشاره گر جدیدی تعریف می شـود) . محـدوده عملکـرد آنهـم دستور مربوطه است، یعنی پس از اتمام آن دستور، نام جدید دیگر وجود ندارد. گـاهی اتفاق می افتد که یک پرس و جو دو یا چند بار به یـک جـدول نیـاز دارد . مـثلاً اگـر بخواهیم اساتیدی را که دفتر کارشان مشترک است پیدا کنـیم بایـد دوبـار از جـدول استاد استفاده نمائیم . دستور زیر پاسخ این پرس و جوست :

$$prof_{prof.office=p.office \wedge} X_{prof.pname \neq panam} (\rho_p (\prod_{pname,office} (prof))$$

ابتدا ستونهای نام استاد و دفتر او از جدول استاد جدا و با نام ρ نامگذاری می شود . سپس سطرهائی از prof که با ρ دفتر کار یکسان ولی نام متفاوتی دارند انتخاب می شوند . باید توجه داشت که همه اساتید با خودشان هم اطاق هستند و اگر شرط $prof.pname \neq p.pname$ نباشد ، پاسخ غلط خواهد بود .

ب – **جایگزینی :** با علامت ← ، جدول حاصل از دستورات ذخیـره مـی شـود تـا در ادامه مورد استفاده قرار گیرد. اگـر دسـتوری طـولانی باشـد مـی تـوان بـا اسـتفاده از جایگزینی ، آن را در چند مرحله نوشت . پاسخ پـرس و جوهـا در ایـن مـوارد از چنـد دستور تشکیل می شود .

مثال : مشخصات دروسی که توسط استاد قربانی ارائه شده است .

دستور

$$temp \leftarrow \prod_{c\#} \sigma_{pname=\text{"قرباني"}}(\text{sec})$$
$$temp \infty crs$$

ج – **تقسیم :** با علامت ÷ کاربرد بسیار ارزشمندی دارد ولی متاسفانه زبانهای متداول بانک اطلاعات مستقیماً پیاده سازی شده است و بـه سـختی مـی تـوان آن را معـادل سازی کرد. کاربرد عملگر تقسیم زمانی است که بخواهیم همه حالت های یک اتفاق را بررسی کنیم. مثال های زیر مسئله را روشن تر می کند:

- دانشجویانی که همه درس های استاد تبریزی را گرفته اند
- اساتیدی که همه دفاتر کار کرده اند
- درسهایی که توسط همه دانشکده ها ارائه می شوند

البته جملاتی مانند "همه دانشجویانی که ..." مثال های غلطـی هـستند زیـرا د رآنهـا کلمه "همه" نقشی ندارد ! (طبیعی است که خروجی یک دستور شامل همه پاسخ ها خواهد بود)

پاسخگویی به پرس و جوهای فوق با استفاده از دستور تقسیم بسیار ساده (و بدون آن بسیار مشکل) است ابتدا بخشی را که شامل همه می شود پیدا مـی کنـیم (مقـسوم علیه) سپس بخش دیگر را بر آن تقسیم می کنیم درآن بخش حتماً باید صفت های مندرج در مقسوم علیه وجود داشته باشند و خروجی شامل صفتهای باقیمانده خواهـد بود .

مثال : دنشجویانی که همه درسهای استاد تبریزی را گرفته اند.

حل : ابتدا همه درسهای استاد تبریزی .

$$temp \leftarrow \prod_{c\#} \sigma_{pname=\text{«تبریزی»}}$$

سپس تقسیم.

دستور temp ÷ #C غلط است در stud وجود ندارد .

پاسخ صحیح این است :

$$\prod_{s\#sname,C\#}(stud \infty \sec) \div temp$$

مقسوم دارای ستونهای #C ، و sname و #S است و مقسوم علیه دارای ستون #C

پس خروجی دارای ستونهای sname و #S خواهد بود .

مثال : درسهایی که توسط همه دانشکده ها را ارائه می شوند .

حل : ابتدا همه دانشکده ها :

$$temp \leftarrow \prod_{c\lg\#}$$

$$crs \div temp$$

سؤال : آیا پرس و جوی crs ÷ clg صحیح است؟

جواب : خیر، زیرا همه ستونهای clg در جدول crs وجود ندارد .

مثال : نام دانشجویانی که همه دروس افتاده خود را مجددا گذرانده اند .

حل : آیا پاسخ صحیح است ؟

$$\prod_{sname,s\#,c\#}(stud\infty(\sigma_{score>=10}(\sec))) \div \prod_{S\#,C\#}\sigma_{score<10}(\sec)$$

5-3/3- بهینه سازی پرس و جو

در صفحات گذشته دیدم که یک سوال ممکن است چند پاسخ داشته باشد. آیا همه این پاسخ ها از نظر زمان و فضا معادلند؟ مسلما نه . قبلاً گفته شد که بعضی از عملگرها گران تر از بعضی دیگر هستند و باید از آنها پرهیزکرد ولی مسئله به همین جا ختم نمی شود در جواب یک سوال می توان با جابجا کردن عملگرها به پاسخ متفاوتی رسید آیا همه آنها معدلند ؟

مثال : مشخصات دروس چهار واحدی که در نیسمال اول سال 1377 (کد 771) ارائه شده اند .

حل : واحد درس در جدول درس و نیسمال آن در جدول گروه درس آمده پس به این دو جدول نیاز داریم . فقط با استفاده از σ,∞ , $\prod$ به این سوال پاسخ می دهیم

<u>دستور غلط</u>

$$\sigma_{unit=4\wedge term=771}(crs \propto \sec)$$

این دستور غلط است زیرا پس از عمل نیم پیوند صفت term حذف می شود (چون مربوط به جدول دوم است) و نمی توان روی آن شرط گذاشت .

<u>دستور 1</u>

$$\sigma_{unit=4}(crs \propto (\sigma_{term=771}(\text{sec})))$$

<u>دستور 2</u>

$$\sigma_{unit=4}(crs)) \propto (\sigma_{term=771}(\text{sec}))$$

<u>دستور 3</u>

$$\sigma_{unit=4}(crs) \propto (\sigma_{term=771}(\prod_{C\#.term} (\text{sec})))$$

<u>دستور 4</u>

$$\sigma_{unit=4}(crs) \propto \prod_{C\#} \sigma_{term=771}(\text{sec}))$$

این دستور ها به ترتیب بهتر می شوند. اجازه بدهید دستور 1 را با دستور 4 مقایسه کنیم در دستور 1 جدول بزرگ CRS (که در بخش کوچکی ازآن مربوط به دروس چهار واحدی است) با دروس نیسمال اول 1377 با هم ترکیب می شوند و سپس دروس چهار واحدی جدا می شوند . در دستور 4 ابتدا دو جدول حتی الامکان کوچک می شوند و سپس با هم ترکیب (که بسیار گرانتر است) میگردند . از جدول CRS ابتدا مشخصات کامل دروس چهار واحدی و از جدول sec فقط شماره دروس ترم اول 1377 جدا و سپس عمل ترکیب نیم پیوند انجام می شود .

از طرفی از کاربران بانک اطلاعات توقع نمی رود که دستور بهینه وارد کنند زیرا آنها افرادی عاید هستند که فقط نحوه استفاده از زبانهای پرس و جو را آموزش دیده اند و از کم و کیف مسائل بی خبرند از سوی دیگر غیر فنی خیلی گران تمام می شود و سرعت را پایین می آورد و بعضا ممکن است به دلیل نیاز به فضای زیاد قابل انجام هم

نباشد . راه حل این مشل در دستهای توانمند نرم افزار غول پیکر نظام مدیریت بانـک اطلاعات (***DBMS***) است . خوشبختانه می توان بخش اعظم بهینـه سـازی پـرس و جو را قانونمند کرد و به صورت الگوریتم پیاده سازی نمود . آنچه کاربرد وارد می کنـد ابتدا توسط الگوریتم به دستور بهینه تبدیل می شود و سپس اجرا می گردد.

قواعد بهینه سازی پرسو و جو

قواعد کاملا مشخصی در مدل رابطه ای بهینه سازی پرس و جو وجود دارد که در زیـر خلاصه می شود .

الف ـ گزینش و پرتو

قاعده 1 : گزینش را هر چه ممکن است زودتر انجام دهید .

قاعده 2 : شرطهای ترکیبی را بـه شـرطهای متـوالی تبـدیل کنیـد . مـثلا مـی تـوان $\sigma_{p1\wedge p2}(e)$ را با $\sigma_{p1}(\sigma_{p2}(e))$ جایگزین کرد .

قاعده 3 : پرتو را زود انجام دهید (ولی دیرتر از گزینش)

مثالی که در ابتدای بحث آورده شد همه این موارد را شامل می شود با مثال دیگـری مسئله روشن می گردد.

مثال : دانشکده هایی که در تهران واقعند و رؤسای آنها دکتری ندارند .

حل : مشخصاً به جداول clg و prof نیاز داریم . بـا ∞ و Π,σ پاسـخ مـی گـوییم (نیم پیوند کافی نیست چـون لازم اسـت مـدرک تحـصیلی رئـیس دانـشکده هـم در خروجی بیابید)

دستور :

$$\prod_{\ldots}\sigma_{city=\text{تهران}} \qquad \wedge \deg ree \neq \text{دکتری} \qquad (prof \infty c\lg)$$

این پاسخ بسیار گران و غیر منطقی است زیرا بخـش کـوچکی از جـدول clg و prof کار داریم در حالیکه کل جداول را با هم پیوند طبیعی داده ایم .

$$\prod_{\ldots}\sigma_{city=\text{تهران}} \qquad (c\lg)\infty\prod_{\deg ree,pname}\sigma_{\deg ree\neq\text{دکتری}} \qquad (\text{prof})$$

مثال : اساتیدی که تخصص آنها با تخصص رئیس دانشکده شان متفاوت است .

حل : ابتدا تخصص روسای دانشکده ها را پیدا کنیم سپس با تخصص دیگران مقایـسه می کنیم .

$$temp1 \leftarrow \prod_{pname,esp,c\lg\#} (prof)$$
$$temp2 \leftarrow temp1 \propto c\lg$$
$$temp1_{temp1.c\lg\#=temp2.c\lg\#\wedge temp1.esp\neq temp2.esp}temp2$$

در این مثال نمی توان شرط ترکیبی را شکست زیرا ستون هـای مـورد مقایـسه در دو جدول هستند نه در یک جدول .

ب ـ عملگرهای دیگر

عملگرهای دیگر جبر رابطه ای بسیارند و جایگزینی و جابجا کردن آنها تـاثیر شـگفتی در عملکرد پرس و جو دارد . در اینجا بعـضی از فرمـول هـای معـادل را مـی آوریـم و مسئله را با یک مثال روشن می کنیم .

$$a \infty b = b \infty a$$
$$(a \infty b) \infty c = a \infty (b \infty c)$$

با استفاده از این دو فرمول ، پیوند طبیعی سه جدول را می توان به چندین طریق محاسبه نمود (مثل $(b \infty a) \infty c$ و $c \infty (b \infty a)$ و غیره) عملکرد این پرس و جوها ممکن است با یکدیگر بسیار متفاوت باشد یعنی یکی ده ها برابر گرانتر از دیگری تمام شود. بهینه سازی پرس و جو در چنین مواردی نیاز به دانستن سایز جداول دارد یعنی تعداد سطر و ستون هر کدام. این عمل معمولا از چهار چوب توان کاربران بیرون است زیرا سایز جداول مرتبا عوض می شود. نظام مدیریت بانک اطلاعات (***DBMS***) می تواند آمار جداول را نگهداری و به هنگام کند و از آن در بهینه سازی پرس و جو استفاده نماید. نگهداری این آمارها نیز پرخرج است زیرا جداول مرتباً دستخوش تغییر هستند (بعداً خواهیم دید که با یک دستور می توان سطرهای زیادی را اضافه یا حذف کرد یا تغییر داد) سوال این است که آیا این نوع بهینه سازی مقرون به صرفه است؟ پاسخ تجربی آن این است : قطعا بله دستورها آنقدر با هم متفاوتند که نه تنها نگه داشتن آمار جداول به هنگام کردن آنها می صرفد بلکه عملیات دیگری از قبیل مرتب کردن و شاخص گذاری روی پرنده ها با کلید های متفاوت نیز ضروری است .

اجازه بدهید مسئله را با یک مثال دنبال کنیم .

مثال : مشخصات کامل دروس و رگوه های درسی آنها .

حل : پاسخ مشخص این سوال یک یا دو دستور زیر است :

$crs \infty \sec$ یا $\sec \infty crs$

این دو دستور با یکدیگر بسیار متفاوتند ! زیرا سایز دو جـدول کـاملاً متفـاوت اسـت . جدول crs معمولاً کوچک است و جدول sec بسیار بزرگتر از آن زیرا هم گـروه هـای مختلف دروس را شامل می شود و هم دانشجویان آنها را. فـرض کنیـد جـدولcrs در حافظه ***cache*** (که از نظر عملکرد ، مشابه حافظه جنبی ولی از نظـر سـرعت شـبیه حافظه اصلی است و مقـدار آن همـواره محـدود اسـت) جـا بگیـرد . در ایـن صـورت الگوریتم های دو راه حل بالا را بررسی می کنیم :

الگوریتم 1: $crs \infty \sec$:

برای هر سطر جدول crs {

برای هر سطر جدول sec {

مقایسه کن

انتخاب کن

}

}

الگوریتم 2 : $\sec \infty crs$:

برای هر سطر جدول sec {

برای هر سطر جدول crs {

مقایسه کن

انتخاب کن

}

}

در الگوریتم 1 باید سطر های بسیار زیاد جدول sec را به دفعات وارد حافظه اصلی کنیم و مقایسه و انتخاب را انجام دهیم . به عبارت دیگر تعداد دستیابی به دیسک به اندازه حاصلضرب سایز دو جدول است .

در الگوریتم 2 هر سطر sec فقط یکبار به حافظه اصلی می آید زیرا جدول crs به طور کامل در حافظه cache است و همه مقایسه ها به یکباره انجام می شود . تعداد دستیابی به دیسک به اندازه سایز جدول sec است . ملاحظه می شود که تعداد دستیابی به دیسک در دو جدول بسیار بسیار متفاوت است . مثلا اگر سایز جدول crs را 100 و سایز جدول sec را 1001 فرض کنیم ، الگوریتم اول یک میلیون بار بیشتر به دیسک دستیابی می کند . با عنایت به این نکته که دستیابی به دیسک هزینه ای به مراتب بالاتر از سایر دستور ها دارد (زیرا سرعت رسانه ها به مراتب پایینتر از حافظه اصلی و CPU است) تفاوت دو الگوریتم مشخص می شود . در واقع باید گفت :

میان ماه من تا گردون تفاوت از زمین تا آسمان است .

عملگرهای دیگر وضعیتی مشابه دارند . تعداد دیگری از فرمول های معادل را ذکر می کنیم .

$$a * b = b * a$$
$$(a * b) * c = a * (b * c)$$
$$a \cup b = b \cup a$$
$$(a \cup b) \cup c = a \cup (b \cup c)$$
$$a \cap b = b \cap a$$
$$(a \cap b) \cap c = a \cap (b \cap c)$$
$$\sigma_p(a \cap b) = \sigma_p(a) \cap \sigma_p(b)$$
$$\sigma_p(a - b) = \sigma_p(a) - b = \sigma p(a) - \sigma_p(b)$$

اینگونه فرمول ها بسیارند و تاثیر آنها در زمان و فضای اجرای دستور شگفت آور است . بهینه سایزی پرس و جو جنبه های دیگری هم دارد که از آنها صرف نظر می کنیم . یادآور می شویم که همه نرم افزار های نظام مدیریت بانک اطلاعات (***DBMS***) پرس و جو ها را بهینه نمی کنند و بنابران بهتر است حتی الامکان پرس و جوی بهینه را وارد کنیم.

6-3/3- به روز درآوردن داده ها

کارکردن با جداول در مدل رابطه ای به بازیابی داده ها و پاسخ به پرس و جوها محدود نمی شود بلکه تغییر جداول و داده های آنها را نیز شامل می گردد. تغییر جداول مانند اضافه کردن و حذف کردن یک یا چند ستون و تغییر دامنه ستون ها به جبر رابطه ای مربوط نمی شود زیرا این قبیل تغییرات در واقع تغییر تصویر ادراکی بانک اطلاعات است. جبر رابطه ای با رابطه ها کار دارد نه با تصویر ادراکی آنها .

تغییر داده های جداول در قالب جبررابطه ای می گنجد و سه مورد زیر را شامل می شود:

الف – اضافه کردن داده به جدول

افزودن یک یا چند سطر به جدول در جبر رابطه ای با استفاده ا زعملگر های $\cup, \leftarrow$ قابل انجام است . مثلا اگر بخواهیم درس جدیدی با شماره c101 ، با نام " بانک اطلاعات نامتمرکز " که چهار واحدی است و توسط دانشکده مهندسی کامپیوتر با کد 10 ارائه یم شود را به جدول crs اضافه کنیم می توانیم چنین بنویسیم :

$crs \leftarrow crs \cup \{("c100", "$ **بانک اطلاعات نامتمرکز** $", 4, 10)\}$

همچنین می توان خروجی یک دستور را به جدولی افزود. در این صورت چند سطر با یک دستور وارد جدول می شود. فرض کنید جدولی به نام good_stud با ستون های (معدل کل ، نام دانشجو) وجود دارد و می خواهیم دانشجویان خوب (معدل بالای 18) دانشکده مهندسی کامپیوتر (کد 10) را به آن بیفزاییم دستور زیر این عمل را انجام می دهد :

$$good_stud \leftarrow good_stud \cup \prod_{sname.avg}(\sigma_{avg>18 \wedge c\lg\#=10}(stud))$$

نقش جایگزینی (←) در افزودن داده ها مهم است و در صورت عدم استفاده از آن جدول مورد نظر تغییر نمی کند .

ب – حذف داده از جدول

این عمل نیز در جبر رابطه ای نیاز به عملگر جدیدی ندارد و با دستورات تفاضل و جایگزینی قابل انجام است . با یک دستور ممکن است یک یا چند سطر از جدول حذف شود و یا حذفی صورت صورت نگیرد (بستگی به دستور دارد) . مثلاً دستور زیر تمام دانشجویانی که معدلی کمتر از 18 دارند را از جدول good_stud حذف می کند :

$$good_stud \leftarrow good_stud - \sigma_{avg<18}(good_stud)$$

چند بار استفاده از یک جدول در این دستور مشکلی ایجاد نمی کند زیرا عملیات مختلف آن به ترتیب انجام می شوند .

ج – تغییر داده های جدول

با این عمل نه سطری به جدول افزوده می گردد و نه از آن حذف می شود بلکه داده های سطرهای موجود تغییر می کند . تغییر داده های جدول نیز نیاز بـه دسـتور جدیـدی در جبر رابطه ای ندارد . می توان با دستورهای گزینش و جایگزینی این عمل را انجـام داد . به مثالهای زیر توجه کنید :

مثال : افزودن یک واحد به همه دروس .

دستور : $\sigma_{unit \leftarrow unit+1}(crs)$

مثال : تغییر نام " باختران " به " کرمانشاه " در جدل دانشجو .

دستور : $\sigma_{city \leftarrow \text{"کرمانشاه"}} \quad (\sigma_{city= \text{"باختران"}}(stud))$