جبر رابطه ای

جبر:

- نوع داده ها و عملگر های رو ی آنها
- به مجمو عه ای از قوانین و عملگرها که امکان پردازش جداول را فراهم می سازند، جبر رابطه ای می گویند
- نوع داده ها در جبر رابطه ای فقط **رابطه** است،یعنی ورودی و خروجی تمامی عملگر ها رابطه می باشد.

عملگرها در جبر رابطه ای را می توان به چهار دسته تقسیم کرد:

1- عملگر های ساده
- **1**- گزینش(Select ، restrict **یا** σ) *
- **2**-پرتو(project **یا** Π) *

2-عملگر های مجمو عه ای
- **1**- اجتماع($\cup$) *
- **2**-اشتراک($\cap$)
- 3-تفاضل(**−**) *

3- عملگر های پیوند
- **1**-ضرب دکارتی(**X**) *
- **2**-پیوند طبیعی(∞)
- **3**-نیم پیوند($\propto$)
- **4**-پیوند شرطی(X_θ)
- **5**-فرا پیوند

4- عملگر های دیگر
- **1**- نامگذاری(ρ) *
- **2**- شیفت
- **3**- انتساب
- **4**-جایگزینی(**←**)
- **5**- تقسیم(**÷**)

عملگر های اصلی:

عملگر هائی هستند که جهت انجام یک سری از عملیات نیاز به آنها حتمی است(آنهائی که علامت ستاره خورده اند)

عملگر های غیر اصلی عملگرهائی هستند که آنها را به کمک عملگر های اصلی میتوان انجام داد و جهت سهولت کار هستند.

دامنه(Domain) : مجموعه مقادیری است که یک صفت خاصه می تواند اتخاذ کند. به عنوان مثال اگر یک صفت خاصه از نوع **Integer** دو بایتی باشد مقادیری که می تواند اتخاذ نماید 32768 الی 32768- می باشد (دامنه)

رابطه: زیر مجموعه ای است از ضرب دکارتی چند دامنه

مثال: $\{1,2,3\}\times\{4,5\}=\{(1,4),(1,5),(2,4),(2,5),(3,4),(3,5)\}$

R یک رابطه است: $R=\{(1,5),(2,4),(3,4)\}$

مثال: اگر داشته باشیم D2:Integer و D1: String آنگاه هرمجموعه ای که عضو هایش زوج های مرتب (D1, D2) باشند یک رابطه است، بهترین راه نمایش و پیاده سازی رابطه به وسیله جدول است.

D1: String	D2: Integer
علی	10
رضا	20
...	...

تاپل: به هر کدام از سطر های جدول یک تاپل گویند

تاپل(Tuple):

به عضو (3,4) از رابطه R (بالا) یک تاپل گویند. پس تاپل به اعضاء رابطه گفته می شود. به عبارت دیگر تاپل مجموعه ای است از مقادیر صفات خاصه.

کاردینالیتی رابطه: تعداد تاپل های رابطه در یک لحظه از حیات آن، کاردینالیتی رابطه نام دارد و در طول حیات رابطه متغییر است .

تناظر بین مفاهیم رابطه ای و مفاهیم جدول:

رابطه = جدول	درجه = تعداد ستون ها
تاپل = سطر	کاردینالیتی = تعداد سطر ها
صفت = ستون	میدان = مجموعه مقادیر ستون

کلید:

صفت خاصه یا ترکیبی از صفات خاصه که در تمام تاپل های یک مجموعه منحصر به فرد باشد.

ابرکلید(Super Key:s.k)

یعنی هر ترکیبی از صفت ها که خاصیت کلید داشته باشد. این تنها نوع کلید است که الزاما خاصیت حداقلی (Minimality) نیست. یعنی زیر مجمو عه ای از آن هم ممکن است کلید باشد. مثلا «شماره دانشجوئی» و «نام دانشجو-شماره دانشجوئی» هردو ابر کلید هستند.

کمینگی اجزائی(Minimality):

یعنی اگر صفت خاصه ای یا ترکیبی از صفات خاصه کلید باشند، هیچ زیر مجموعه ای از آنها (به جز زیر مجمو عه ای که مساوی خودشان باشد) کلید نباشد. به عبارت دیگر با حذف هر یک از اجزاء A_i ، A_j و... A_k ، یکتائی مقدار از بین می رود.

کلید کاندید(Candidate Key:C.K):

کلیدی است که دارای خاصیت Minimality است. یک رابطه ممکن است چند کلید کاندید داشته باشد

کلید اصلی(Primary Key:P.k):

کلید کاندیدی است که توسط طراح بانک انتخاب و معرفی می شود. هر رابطه ای حتما کلید اصلی دارد، چون هر رابطه ای حداقل یک کلید کاندید دارد.

کلید فرعی یا بدیل(Alternative:A.k):

هر کلید کاندید غیر از کلید اصلی را کلید فرعی می نامند. به عبارتی یکی دیگر از کلید های کاندید است که برای برخی کاربرد ها انتخاب می شود. طراح می تواند در شمای ادراکی هم کلید اصلی را معرفی نماید و هم کلید فرعی را.

به عنوان مثال اگر در بانک اطلاعاتی دانشجویان، شماره دانشجوئی کلید اصلی باشد در صورتی که بخواهیم مشخصات دانشجویان را بر اساس نام نمایش دهیم، صفات نام و شماره شناسنامه می توانند جهت سهولت کار به عنوان کلید فرعی در نظر گرفته شوند.

کلید خارجی(Foreign Key:F.k):

کلید خارجی در رابطه ای مانند R_i ، صفت خاصه یا صفات خاصه ای است که در رابطه دیگر مانند R_j، کلید اصلی یا فرعی باشد

کلید خارجی برای نمایش ارتباطات بین انواع موجودیت ها بکار می رود و تنها کلیدی است که مقدار آن می تواند Null باشد.

S#	P#	J#
S1	P1	J1
S1	P2	J1
S2	P1	J1
S1	P1	J2

در جدول روبرو (S#,P#,J#) کلید کاندید رابطه است. زیرا هیچ یک از صفات خاصه به تنهائی یا دو به دو یکتائی مقدار ندارند.

با آنکه نام دانشجو و شماره دانشجوئی با همدیگر به صورت یکتا تمام دانشجویان را از یکدیگر متمایز می سازند ولی نام دانشجو در این بین زائد است و شماره دانشجوئی برای این منظور کفایت می کند لذا (نام و شماره دانشجوئی) برای مجموعه دانشجویان کلید کاندید نیست. در این حالت به (نام و شماره دانشجوئی) ابر کلید یا Super Key گفته می شود.

مثال های این فصل بر مبنای جدول صفحه بعد ارائه می شوند.

prof:

pname	Office	esp	Degree	Clg#
میر شمسی	4	کامپیوتر	فوق لیسانس	10
ابو طالبی	3	مواد	دکتری	6
قربانی	12	کامپیوتر	دکتری	10
اشرفیزاده	8	شیمی	دکتری	5
هاشمی اصل	10	کامپیوتر	فوق لیسانس	10
جلالی	5	برق	دکتری	7
شیدفر	3	ریاضی	دکتری	1
حسنی	2	ریاضی	دکتری	1
جاهد مطلق	1	کامپیوتر	دکتری	10
ذاکر	4	فیزیک	دکتری	2
مفتون	1	زبان	دکتری	3
صادقیان	3	صنایع	دکتری	4

Stud:

S#	Sname	City	avg	Clg#
71133848	محمدی	تهران	17.24	10
72130502	وکیلی	اصفهان	14.06	10
72203305	علینقیزاده	مشهد	16.42	1
73120504	کمانی	یزد	17.56	4
73166801	احمدی	کرمان	15.44	5
74182532	جوادی	تهران	16.8	5
74209836	حسین زاده	تبریز	12.2	6

Crs:

C#	cname	unit	Clg#
10172	شبیه سازی	3	10
10174	مدار منطقی	3	10
12100	معارف 1	2	12
12564	ریاضی عمومی 1	4	1
51516	شیمی آلی	3	5
71203	کنترل خطی	3	7

Clg:

Clg#	clgname	city	pname
1	ریاضی	تهران	حسنی
10	کامپیوتر	تهران	جاهد مطلق
11	معماری	یزد	نقره کار
2	فیزیک	مشهد	ذاکر
3	زبان	مشهد	مفتون
4	صنایع	تهران	صادقیان
5	شیمی	تهران	اشرفی زاده
6	مواد	تبریز	ابوطالبی
7	برق	تهران	جلالی

sec:

sec#	c#	s#	term	pname	score
1724	10172	71133848	761	هاشمی اصل	14.5
1516	51516	74182532	752	اشرفی زاده	17
1747	10174	71133848	752	میرشمسی	15.75
1747	10174	72130502	752	میرشمسی	12.5
1748	10172	72203305	761	قربانی	16.25

عملگر گزینش:

گزینش سطر هائی از جدول را انتخاب می کند. نام جدول جلو علامت گزینش 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 کلمه یزد را نشان می دهد و شماره دانشکده آنها 4 است.*

دستور:

$$\sigma_{City = \text{"یزد"} \wedge clg\# = 4}(Stud)$$

خروجی:

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

مثال: *با استفاده از عملگر جبر رابطه ای دستوری بنویسید که مشخصات دانشجویان تهرانی را نمایش دهد.*

$$\sigma_{City = \text{"تهران"}}(Stud)$$

S#	Sname	City	avg	Clg#
71133848	محمدی	تهران	17.24	10
74182532	جوادی	تهران	16.8	5

عملگر پرتو:

ستون هائی از جدول را انتخاب می کند و هیچ گونه شرطی اعمال نمی شود. درخروجی پرتو سطر های تکراری حذف می شوند. نام جدول جلو علامت پرتو در پرانتز و ستون های انتخاب شده زیر آن می آید.

مثال: *ستون های شماره دانشجو، نام دانشجو و کد دانشکده دانشجو از جدول دانشجو.*

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

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

مثال: شهرهای مختلف محل تولد دانشجو

دستور: $\Pi_{city}\ (stud)$

خروجی:

City
تهران
اصفهان
مشهد
یزد
کرمان
تبریز

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

مثال: ستون های شماره دانشجوئی، نام، کد دانشکده و میانگین،دانشجویانی که معدل آنها بالای 15 است.

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

دستور معادل: $\sigma_{avg>15}(\Pi_{s\#,sname,c\lg\#,avg}(stud))$

S#	Sname	avg	Clg#
71133848	محمدی	17.24	10
72203305	علینقیزاده	16.42	1
73120504	کمانی	17.56	4
73166801	احمدی	15.44	5
74182532	جوادی	16.8	5

عملگر های مجموعه ای

عملگر های اجتماع ، اشتراک و تفاضل همان معنای خود در تئوری مجموعه ها حفظ کرده اند. ورودی هر کدام دو صفت و خروجی آنها یک رابطه است. روابط ورودی باید همتا(Same arity) باشند،یعنی

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

☜ به عنوان مثال دو رابطه **stud** و **prof** ، **same arity** نیستند چون شرط دوم را ندارند.

مثال: دستوری بنویسید که لیست نام همه افرادی را که در دانشکده ها هستندرا نمایش دهد.

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

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

☜ اگر کاردینالتی R_1 برابر n و کاردینالیتی R_2 برابر m باشد، آنگاه کاردینالیتی $R_1 \cup R_2$:

- حداقل زمانی است که یکی از رابطه ها زیر مجموعه دیگری باشد و برابر $\max(n,m)$ خواهد شد.

- حداکثر زمانی است که تاپل مشترک ندارند و برابر $n+m$ است

مثال: لیست نام اساتیدی که رئیس دانشکده نیستند.

حل: ابتدا نام اساتید را پیدا می کنیم و سپس نام روسای دانشکده ها را از آن تفریق می کنیم

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

خروجی:

pname
میرشمسی
قربانی
هاشمی اصل
شید فر

اگر کاردینالتی R_1 برابر n و کاردینالیتی R_2 برابر m باشد، آنگاه کاردینالیتی $R_1 - R_2$:

- حداقل زمانی است که $R_1 \subseteq R_2$ و برابر صفر است.
- حداکثر زمانی است که تاپل مشترک ندارند و برابر n است

مثال: لیست اسامی اساتید و دانشجویان همنام.

حل: کافی است اشتراک اسامی دانشجویان و اساتید را پیدا کنیم، یعنی دستور : $\Pi_{sname}(stude) \cap \Pi_{pname}(prof)$

اگر کاردینالتی R_1 برابر n و کاردینالیتی R_2 برابر m باشد، آنگاه کاردینالیتی $R_1 \bigcap R_2$:

- حداقل زمانی است تاپل مشترک ندارند و برابر صفر است.
- حداکثر زمانی است که $R_1 \subseteq R_2$ یا $R_2 \subseteq R_1$ و برابر $\min(n,m)$ است.

عملگر های پیوند

الف: ضرب دکارتی

از گرانترین عملگر های بانک رابطه ای است که زمان و فضای زیادی می خواهد و تا حد امکان باید از آن اجتناب کرد.حاصل ضرب دو رابطه، رابطه ای است که تاپل هایش از الصاق هر یک از دو تاپل دو رابطه بدست می آیند. به عبارت دیگر در $R_1 \times R_2$، هر سطر R_1 را پشت سر تمام سطر های R_2 قرار می دهیم.

نکته: اگر جدول A دارای m سطر و n ستون و جدول B دارای p سطر و q ستون باشد. آنگاه $A \times B$ دارای تعداد $m \times p$ سطر و تعداد $n+q$ ستون خواهد داشت.

نکته: ضرب دکارتی در ریاضیات مجموعه ها خاصیت جابه جائی ندارد یعنی($A \times B \neq B \times A$) ولی در جبر رابطه ای چون ترتیب ستون ها مهم نیست، خاصیت جا به جائی دارد، یعنی ($A \times B = B \times A$)

ب: پیوند شرطی(ضرب دکارتی شرطی)($\theta-$ JOIN)

این عملگر، زیر مجموعه ای است از ضرب دکارتی که شرط θ روی سطر های آن اعمال شده باشد. ستون های خروجی معادل ستون های ضرب دکارتی است. در بعضی کتاب ها آن را به صورت X_θ نمایش داده اند که θ شرط مورد نظر می باشد.

مثال. نام و شماره درسی که توسط استاد قربانی ارئه شده است.

حل: نام و شماره درس در جدول "درس" آمده است. این جدول با جدول "استاد" ارتباط ندارد(هیچ کدام کلید خارجی در دیگری ندارد). اما با جدول "گروه درس" ارتباط دارد، پس می توان از پیوند شرطی این دو جدول استفاده کرد به صورت زیر.

$$\Pi_{cname, crs.c\#}(crs\ X_{pname = \text{قربانی} \ \wedge crs.c\# = \sec.c\#}\ \text{sec})$$

خروجی:

C#	cname
10172	شبیه سازی

کاردینالیتی:

- حداقل زمانی است که هیچ شرطی نداریم $\min = 0$
- حداکثر زمانی است که تمام شرط ها باشند.

ج.پیوند طبیعی(natural join):

مشابه پیوند شرطی است با تفاوت های زیر

☜ خودبخود شرط تساوی روی همه ستون های همنام دو جدول اعمال می شود. یعنی فقط سطرهائی را از دو جدول انتخاب می کند، که همه ستون های همنام آن دو جدول مقادیر مساوی داشته باشند. در صورتی که دو جدول ستون همنام نداشته باشند، نتیجه پیوند طبیعی معادل ضرب دکارتی است.

☜ستون های تکراری فقط یکبار در خروجی ظاهر می شوند. از آنجا که فقط مقادیر مساوی آنها انتخاب می شود، نیازی به تکرار ستون یا نقطه گزاری (مثلا sec.c#) نیست.

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

حل. ابتدا شماره دروس استاد قربانی را از جدول "گروه درسی" انتخاب می کنیم، سپس شماره و نام همه درس ها را از جدول درس انتخاب می کنیم، آنگاه آنها را پیوندطبیعی می کنیم به شکل زیر.

$$\Pi_{c\#}\ \sigma_{pname = \text{قربانی}}\ ((\sec) \infty (\Pi_{c\#, cname}(crs))$$

مثال. مشخصات کامل رؤسای دانشکده ها

حل.

دستور

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

قسمتی از خروجی.

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

کاردینالیتی:

- حداقل زمانی است که حداقل یک ستون همنام داشته باشیم ولی مقدار یکسان نداشته باشند(min=0)
- حداکثر زمانی است که هیچ ستون مشترکی نداشته باشیم(max=n*m)

د.عملگر نیم پیوند(Semi join):

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

مثال: خروجی دستور زیر چیست.

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

حل: ظاهرا این دستور مشخصات دروسی را می دهد(بدون مشخصات گروه آنها) که در ترم اول سال 77 (کد 771) در دانشکده 1 ارائه می شود، ولی واقعا چنین نیست. دستور فوق غلط است زیرا ستون ترم مربوط به جدول crs نیست و پس از نیم پیوند حذف می شود پس نمی توان آن را در شرط کنجاند. یکی از پاسخ های صحیح چنین است.

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

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 با کنار رفتن چند ستون، سطر های تکراری پدید می آیند و حذف می شوند.

☜ بر خلاف سایر عملگر های این بخش،ترتیب جداول ورودی در نیم پیوند مهم است($x \propto y \neq y \propto x$) زیرا همیشه ستون های جداول اول را می دهد

کاردینالیتی:

- حداقل زمانی است که ستون های همنام مقدار مشترک ندارند(min=0)
- حداکثر زمانی است که تمام مقادیر ستون های همنام مشترک باشند(max=n)

عملگر های دیگر:

عملگر تغییر نام($\rho_b a$)

نام جدید b را روی جدول a گذاشته می شود. محدوده آن در همان دستوری است که ذکر شده است. در واقع b اشاره گری به a است.

مثال: نام اساتیدی که دفتر کارشان مشترک است.

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

ابتدا ستون های نام استاد و دفتر او از جدول استاد جدا و با نام p نام گذاری می شود. سپس سطر هائی از $prof$ که با p دفتر کارشان یکسان است، ولی نام متفاوتی دارند انتخاب می شوند. باید توجه داشت که همه اساتید با خودشان هم اتاق هستند و اگر شرط $prof.pname \neq p.pname$ پاسخ غلط خواهد بود.

عملگر انتساب(جایگزینی)

با علامت $\leftarrow$ جدول حاصل از دستورات ذخیره می شود تا در ادامه مورد استفاده قرار گیرد. اگر دستوری طولانی باشد می توان با استفاده از جایگزینی، آن را در چند مرحله نوشت. پاسخ پرس وجوها در این موارد از چند دستور تشکیل می شود.

مثال. مشخصات دروسی که توسط استاد قربانی ارائه شده است. (sec)

$$temp \leftarrow \Pi_{c\#} \sigma_{pname = \text{قربانی}}$$

$$temp \propto crs$$

عملگر تقسیم($R_1 \div R_2$)

☜ تمام ستون های R_2 در R_1 می بایست موجود باشند(شمای R_2 می بایست زیر مجموعه شمای R_1 باشد).

☜ ستون های خارج قسمت تقسیم، همان ستون هائی از R_1 است که در R_2 نباشند.

☜ نتیجه همان خارج قسمت، تاپل هائی از R_1 است که به ازای تمام تاپل های R_2 تکرار شده است

مثال. با توجه به R_1 و R_2 خروجی $(R_1 \div R_2)$ چیست.

R_1

S#	cname
10	منطقی
20	منطقی
10	شبیه سازی
30	ذخیره
30	منطقی
10	ذخیره
30	شبیه سازی
20	شبیه سازی
40	شبیه سازی

R_2

cname
منطقی
شبیه سازی
ذخیره

$R_1 \div R_2 \Rightarrow$

S#
10
30

مثال. با توجه به R_1 ، R_2 و R_3 خروجی $(R_1 \div R_2)$ و $(R_1 \div R_3)$ چیست.

S#	P#
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S1	P6
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S4	P5

R_1

P#
P1
P2
P3
P4
P5
P6

R_2

P#
P2
P4

R_3

$R_1 \div R_2 \Rightarrow$

S#
S1

$R_1 \div R_3 \Rightarrow$

S#
S1
S4

مثال. دستور جبر رابطه ای بنویسید که نام و شماره دانشجویانی را برهر که تمام درس های ارئه شده توسط میر شمسی را گرفته اند.

$T_1 \leftarrow \Pi_{\text{sec\#}, c\#}(\sigma_{pname = \text{میرشمسی}} (\text{sec}))$ همه درس های میر شمسی

$T_2 \leftarrow (stud \infty \text{sec})$ تمام دانشجویانی که تمامی درس ها را گرفته اند

$\Pi_{sname, c\#}(T_2 \div T_1)$ تمام دانشجویانی که تمامی درس های میر شمسی را گرفته اند.

☜ تقسیم زمانی استفاده می شود که همه(تمام) حالات یک موضوع را بررسی کنیم.

مثال. نام و شماره دروسی را که توسط همه دانشکده ها ارائه می شود.

$temp \leftarrow \Pi_{c\lg\#}(c\lg)$ تمامی دانشکده ها

$crs \div temp$ جواب

اعمال حذف،اضافه و تغییر رابطه ها

اضافه کردن به کمک عملگر های $\cup$ و $\leftarrow$ انجام می گیرد.

مثال. جهت اضافه کردن (82105200,"رضائی","شیراز",16.5,5) به جدول stud دستورات جبر رابطه ای به صورت زیر است

$stud \leftarrow stud \cup (82105200, "رضائی", "شیراز", 16.5, 5)$

مثال. جهت اضافه کردن درس جدیدی با شماره C101 با نام "بانک اطلاعات نامتمرکز" که چهار واحدی است و توسط دانشکده مهندسی کامپیوتر با کد 10 ارائه می شود، به جدول crs دستورات جبر رابطه ای به صورت زیر است.

$crs \leftarrow crs \cup (C100, "بانک اطلاعات نامتمرکز", 4, 10)$

حذف کردن با دستورات $-$ و $\leftarrow$ قابل انجام است.

مثال. برای حذف دانشجویی با مشخصات 73120504 از جدول stud دستورات جبر رابطه ای به صورت زیر است.

$$stud \leftarrow stud - \sigma_{s\# = 73120504}(stud)$$

مثال. دستور زیر تمام دانشجویانی که معدلی کمتر از 18 را از جدول good_stud حذف میکند.

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

تغییر داده های جدول.

☜ با این عمل نه سطری به جدول افزوده می شود و نه از آن حذف می شود، بلکه داده های سطر های موجود تغییر می کند.

☜ می توان با دستور های گزینش و جایگزینی این عمل را انجام داد.

مثال. افزودن یک واحد به همه دروس $\sigma_{unit \leftarrow unit+1}(crs)$

مثال. تغییر نام باختران به کرمانشاه در جدول دانشجو. $\sigma_{city \leftarrow "کرمانشاه"}(\sigma_{city = "باختران"}(stud))$

توسعه عملگرهای جبر رابطه ای

1- توسعه تصویر(project): به معنای قرار دادن دستورات محاسباتی در عملگر تصویر می باشد.

مثال. حساب بانکی(account)

Customer#	Customer name	blance	With draw	br-name
1	محمودی	$1000	$500	ملی
2	ایمانی	$5000	$3000	سپه
3	رضائی	$4500	$4000	مسکن
4	کریمی	$7000	$3000	ملی

نمایش مقدار باقی مانده حساب مشتری $\Pi_{customer\#, (blance - withdraw)}(ac)$

☜ عمل $blance - withdraw$ برای هر تاپل انجام می گیرد.

☜ برای دادن نام جدید به ستون حاصل از $blance - withdraw$ از دستور as استفاده می کنیم

$$\Pi_{customer\#, (blance - withdraw)\, as\, Rem}(ac)$$

customer	Rem
1	$500
2	$2000
3	$500
4	$3000

☜ در این حالت رابطه ایجاد شده دارای دو ستون با نام های $Rem, customer\#$ خواهد بود که Rem همان باقیمانده حساب مشتریان را نمایش می دهد.

2- توابع تجمعی (aggregate function):

☜ این توابع مجموعه ای از داده ها را گرفته و یک داده را به عنوان خروجی می دهند این توابع عبارتند از

$sum(), avg(), count(), Max(), Min()$

☜ تابع $avg()$ میانگین چند داده را برمیگرداند و تابع $count()$ عمل شمارش را انجام می دهد

☜ شکل کلی استفاده از توابع تجمعی در جبر رابطه ای به صورت زیر است .

$$G_1, G_2, \ldots G_n \zeta^{(R_1)}_{F_1(A_1)\ldots F_n(A_n)}$$

R_1 : رابطه

A_i : صفت خاصه

$F_j(A_i)$: یک تابع تجمعی بر روی صفت خاصه A_i می باشد.

G_i : صفت خاصه ای که تاپل هایش بر اساس آن گروه بندی می شوند.

مثال. دستور $\zeta^{(ac)}_{sum(blance)}$ مجموع موجودیهای تمام مشتریان را از رابطه (ac) می دهد که خروجی دستور یک رابطه می باشد.

مثال. با استفاده از عملگرهای جبر رابطه ای دستوری بنویسید که مجموع موجودیهای افراد را در شعب مختلف نمایش دهد.

جواب. $br-name\,\zeta^{(ac)}_{sum(blance)\,as\,sb}$

خروجی:

Br-name	sb
ملی	$8000
سپه	$5000
مسکن	#4500

اگر تعداد افرادی که در شعب مختلف سپرده گذاری کرده اند را بخواهیم، از تابع تجمعی $count()$; استفاده میکنیم که صفت خاصه ورودی آن $custmer\,name$ خواهد بود.

$$br-name\,\zeta^{ac}_{count(custmer\,name)\,as\,cc}$$

خروجی

Br-name	cc
ملی	2
سپه	1
مسکن	1

☜ در توابع تجمعی اگر بخواهیم مقادیر تکراری یکبار حساب شوند از دستور distinct استفاده مینمائیم، مثلا در صورتی که بخواهیم از رابطه ac نام ها بدون تکرار نمایش داده شوند به این صورت عمل می نمائیم $\zeta^{ac}_{count-distinct(cuustmer\,name)}$

پیوند بیرونی(**extend join**):

انواع پیوند بیرونی عبارتند از :

پیوند بیرونی چپ(⟕)

پیوند بیرونی راست(⟖)

پیوند کامل(⟗)

پیوند بیرونی چپ(⟕):

این پیوند شامل تمام تاپل هائی است که از پیوند طبیعی R_1 و R_2 تشکیل می شوند به اضافه تاپل هائی از R_1 که در پیوند طبیعی R_1 و R_2 ذکر نشده اند، به جای ستون هائی از R_2 که در R_1 وجود ندارند **null** قرار داده می شود.

مثال.

Br#	Br-name	add
1	ملی	آزادی
2	ملی	انقلاب
3	صادرات	آزادی
4	مسکن	انقلاب
5	تجارت	ولیعصر

R_1

Cu#	Cu-name	Br#
100	محمدی	1
101	رضائی	5
205	موسوی	3
300	کریمی	2
400	موسوی	2
401	ایمانی	6

R_2

Br#	Br-name	add	Cu#	Cu-name
1	ملی	آزادی	100	محمدی
2	ملی	انقلاب	300	کریمی
2	ملی	انقلاب	400	موسوی
3	صادرات	آزادی	205	موسوی
5	تجارت	ولیعصر	101	رضائی
4	مسکن	انقلاب	null	null

$R_1 ⟕ R_2$

پیوند بیرونی راست $R_1 ⟖ R_2$

*این پیوند در واقع شامل تاپل هائی است که در پیوند طبیعی R_1 و R_2 تشکیل می شوند بعلاوه تاپل هائی از R_2 که در پیوند طبیعی ذکر نشده اند، در این صورت با ذکر تاپل ها از R_2 به جای تاپل های غیر همنام در R_1 نیز **Null** قرار می گیرد.*

پیوند راست مثال قبلی به صورت زیر خواهد بود.

Br#	Br-name	add	Cu#	Cu-name
1	ملی	آزادی	100	محمدی
2	ملی	انقلاب	300	کریمی
2	ملی	انقلاب	400	موسوی
3	صادرات	آزادی	205	موسوی
5	تجارت	ولیعصر	101	رضائی
6	null	null	401	ایمانی

پیوند بیرونی کامل $R_1 ⟗ R_2$

این پیوند اجتماع دو پیوند بیرونی چپ و راست می باشد، در مثال قبل $R_1 ⟗ R_2$ به صورت زیر خواهد بود.

Br#	Br-name	add	Cu#	Cu-name
1	ملی	آزادی	100	محمدی
2	ملی	انقلاب	300	کریمی
2	ملی	انقلاب	400	موسوی
3	صادرات	آزادی	205	موسوی
5	تجارت	ولیعصر	101	رضائی
4	مسکن	انقلاب	null	null
6	null	null	401	ایمانی

محاسبه کاردینالیتی:

$$cardinality \qquad R_1 ⟕ R_2 \begin{cases} \min = n \\ \max = n*m \end{cases}$$

حالت min *زمانی است که دو رابطه حداقل یک ستون همنام داشته باشند.ولی در ستون های همنام دارای مقادیر مساوی نباشند، در این حالت تنها سطر های R_1 به رابطه اضافه می شودکه همان n است.*

حالت max *زمانی است که دو رابطه هیچ ستون همنامی نداشته باشند که معادل ضرب دکارتی خواهد بود.*

$cardinali\ R_1 \propto R_2$ $\begin{cases}\min = m \\ \max = n*m\end{cases}$ حالت min زمانی رخ می دهد که ستون های همنام دو رابطه مقادیر یکسان نداشته باشند، که در این حالت تنها سطرهای R_2 ذکر خواهند شد، که برابر m می باشد

$cardinali\ R_1 \supset\propto R_2$ $\begin{cases}\min = n+m \\ \max = n*m\end{cases}$ حالت min زمانی رخ می دهد که ستون های همنام دو رابطه مقادیر یکسان نداشته باشند، که در این حالت سطرهای R_1 و R_2 ذکر خواهند شد، که مجموع آنها $n+m$ می باشد

بهینه سازی پرس جوها(Query Optimization)

قواعد بهینه سازی

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

مثال. بهینه $\sigma_{unit=3}(crs \infty \sec) \equiv (\sigma_{unit=3}(crs)) \infty \sec$

در سمت راست(در حالت بهینه) ابتدا عمل گزینش انجام می گیرد، و در نهایت دروس سه واحدی با sec پیوند طبیعی داده می شوند ، در این حالت تعداد تاپل های کمتری در ترکیب $crs, \sec$ وجود خواهد داشت که سبب صرفه جوئی در حافظه خواهد شد.

2. شرط های ترکیبی را تبدیل به شرط های متوالی کنید. این روش باعث بهینه سازی از نظر زمانی خواهد شد.

مثال. $\sigma_{unit=3 \wedge c\lg\#=10}(crs) \equiv \sigma_{unit=3}(\sigma_{c\lg\#=10}(crs))$

3. پرتو را زودتر انجام دهید(ولی دیرتر از گزینش). این کار باعث صرفه جوئی در حافظه می شود.

$$\Pi_{pname,clg\#}(prof \infty clg) \equiv \Pi_{pname,c\lg\#}(prof) \infty \Pi_{pname,c\lg\#}(clg)$$

☜ در پیوند طبیعی باید مراقب باشیم تا در هنگام بهینه سازی، ستون های همنام را از دست ندهیم .

4. استفاده از هم ارزی در جهت بهینه سازی.

$$R_1 \infty R_2 \equiv R_2 \infty R_1 \qquad R_1 \cup R_2 \equiv R_2 \cup R_1$$

$$R_1 \infty (R_2 \infty R_2) \equiv (R_1 \infty R_2) \infty R_3 \qquad R_1 \cap R_2 \equiv R_2 \cap R_1$$

مثال. اگر سایز جدول crs را 100 و سایز جدول sec را 10001 فرض کنیم ثابت کنید $\sec \infty crs$ می تواند صد برابر سریعتر از $crs \infty \sec$ باشد.

حل. این دو دستور با یکدیگر بسیار متفاوتند! زیرا سایز دو جدول بسیار متفاوت است، جدول crs بسیار کوچکتر از جدول sec می باشد، فرض کنید جدول crs در حافظه کش جا بگیرد، در این صورت الگوریتم های دو راه حل بالا را بررسی میکنیم.

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 انتخاب کنیم، به عبارت دیگر تعداد دستیابی به دیسک به اندازه حاصل ضرب سایز دو جدول است.

در الگوریتم 2 هر سطر sec فقط یک بار به حافظه اصلی می آید، زیرا جدول crs به طور کامل در حافظه کش است. تعداد دستیابی به دیسک به اندازه سایز جدول sec است

قواعد جامعیت در رابطه ها:

- جامعیت دامنه ای(Domain Integrity)
- جامعیت درون رابطه ای(inter Relation Integrity)
- جامعیت ارجاعی(Referentiol Integrity)

جامعیت دامنه ی: مقادیری که به صفات یک رابطه داه می شوند، از نوع دامنه آن صفات باشند،به عنوان مثال ، شماره دانشجوئی که به صورت عدد صحیح تعریف شده است، مقدار اعشاری را قبول نکند و همچنین مقادیر کلید ها تهی و تکراری نباشند.

جامعیت درون رابطه ای: به این معناست که هر رابطه به تنهائی درست تعریف شده باشد، به طوری که عضو تکراری نداشته باشد و کلید هایش به درستی معین شده باشند.

جامعیت ارجاع. در این جامعیت باید صفتی که به عنوان کلید خارجی تعریف می شود، در رابطه دیگر کلید اصلی یا فرعی باشد، و مقادیری که به کلید اصلی داده می شوند، در رابطه های دیگر موجود باشند(در رابطه هائی که با آن رابطه در ارتباط هستند)، مثلا می خواهیم درسی با شماره 1000 را به رابطه sec اضافه نمائیم، که این امکان وجود ندارد زیرا در جدول sec (گروه درسی)تنها درس های موجود در crs می توانند قرار بگیرند، و درسی با این شماره در جدول crs وجود ندارد.

- **حساب رابطه ای دامنه ای** (*Domain* Re*lationship Calculas*)

شکل کلی این حساب به این صورت می باشد $\{< c_1, c_2, \ldots, c_n > | \; p(c_1, c_2, \ldots, c_n, c_{n+1}, \ldots)\}$

و بدین معناست که ستون های c_1 تا c_n را بده اگر شرط p برقرار باشد.

☜برای بیان تعلق ستون ها به یک رابطه از عضویت استفاده می کنیم که با $\in$ نشان داده می شود، همچنین در هنگام استفاده از عضویت باید دقت کرد که شرط **same arity** بودن میان ستون های ذکر شده و رابطه داده شده ، رعایت شود

☜. مثلا اگر در بیان شرط داشته باشیم $< c_i \ldots c_j > \in R$ (ستون های c_i تا c_j عضو رابطه R هستند) و در تعریف نام ستون ها یا همان $< c_1 \ldots c_n >$، نام برخی از ستون های ذکر شده در عضویت موجود نباشند، توسط صور وجودی ($\exists$)و یا صور عمومی $(\forall)$ نام آن ستون ها ذکر خواهد شد.

☜ در قسمت شرط ($p()$) با استفاده از سور وجودی ($\exists$) و عمومی ($\forall$) می توان علاوه بر متغیر ها، ثابت هائی را هم ذکر کرد، به شرطی که از نوع دامنه های متناظر با رابطه ذکر شده باشند.

☜ ترکیب شرط ها با $\wedge, \vee, \neg$ و $\Rightarrow$ (معادل "و"، "یا"، "نه"، "نتیجه می دهد") انجام می شود.

☜ جهت ترکیب رابطه ها از ستون های همنام و متغیر های همنام استفاده می گردد.

☜ عدم تعلق با منفی کردن شرط(استفاده از $\in$ و $\neg$) انجام می شود و علامت عدم تعلق($\notin$) تعریف نشده است (زیرا با طبیعت بانک اطلاعاتی همخوانی ندارد)

☜ از هم ارزیهای مجموعه ای می توان در این ترکیب ها استفاده کرد.

$$(R_1 - R_2) \equiv (R_1 \wedge \neg R_2) \quad \leftarrow (R_1 - R_2) \equiv (R_1 \bigcap \overline{R}_2)$$

$$(R_1 \Rightarrow R_2) \equiv (\neg R_1 \vee R_2) \quad \leftarrow (R_1 \Rightarrow R_2) \equiv (\overline{R}_1 \bigcup R_2)$$

مثال. با استفاده از دستورات حساب رابطه ای دامنه ای مشخصات دانشجویانی را که دارای معدل بالاتر از 15 هستند را بدست آورید.

توضیح: در این دستور تنها به رابطه $stud$ نیاز داریم، چون تمام مشخصات دانشجویان با شرایط مورد نظر مد نظر است. در قسمت نام ستونها یا همان نام متغیر ها، نام تمام ستون های رابطه را می آوریم، یعنی خواهیم داشت $< s, sn, c, ave, c\lg >$، در قسمت شرط عنوان می نمائیم این ستون ها از رابطه $stud$ می باشند و شرط هم همان $ave > 15$ است.

$$\{< s, sn, c, ave, c\lg > | < s, sn, c, ave, c\lg > \in stud \wedge ave > 15\}$$

مثال. شماره دانشجوئی دانشجویانی را که معدل زیر 10 دارند را با استفاده از حساب رابطه ای دامنه ای نمایش دهید.

دستور $\{< s > | < s > \in stud \wedge ave < 10\}$ اشتباه می باشد زیرا در رابطه عضویت نام ستون هائی که ذکر می کنیم باید از نظر تعداد و دامنه با ستون های رابطه مورد نظر یکسان باشند یا به عبارتی دارای خاصیت $same\,arity$ باشند در حالیکه $< s >$ تنها یک ستون است و رابطه $stud$ دارای 5 ستون می باشد.در این حال برای رفع مشکل بایستی نام دیگر ستون های رابطه که در عضویت ذکر نشده اند، توسط سور وجودی بیان نمائیم تا خاصیت $same\,arity$ را رعایت کرده باشیم پس داریم $\{< s > | \exists s, c, ave, c\lg(< s, sn, c, ave, c\lg >) \in stud \wedge ave < 10\}$

مثال.نام و تخصص رؤسای دانشکده ها را با استفاده از حساب رابطه ای دامنه ای نمایش دهید.

دستور $\{< p, e > | \exists c, cn, ci(< c, cn, ci, p > \in c\lg) \wedge \exists o, d, cl(< p, o, e, d, cl > \in prof)\}$

این پرس و جو به دو جدول استاد و دانشکده نیاز دارد. متغیر مشترک p دو جدول را به هم پیوند می دهد. متغیر های کمکی هر دو جدول جداگانه تعریف شده اند. به محل باز و بسته شدن پرانتز ها دقت شود. متغیر ها را باید در نزدیکترین محلی که مورد استفاده قرار می گیرند تعریف کرد، و حوزه عملکرد آنها را با پرانتز مشخص نمود. حوزه عملکرد متغیر های خروجی سراسر دستور است.

پایان جلسات چهارم،پنجم،ششم