جبررابطه ای

جبر رابطه ای یک زبان پرس و جو است که عملیات روی پایگاه داده را توسط نمادهایی به صورت فرمولی بیان می کند.

جبر رابطه ای (relational algebra) عملیات روی پایگاه داده را به صورت فرمول بیان می کند. جبر رابطه ای توسط Codd به عنوان مبنای زبان های پرس و جوی پایگاه داده ارائه شد. عملگرهای جبر رابطه ای توسط نمادهائی نمایش داده می شوند .

شش عملگر مبنائی جبر رابطه ای که توسط Codd تعریف شد عبارتند از:

- Selection: σ
- Projection: π
- Cartesian Product: ×
- Set union: $\cup$
- Set difference: –
- Rename: ρ

اکثر عملگرهای دیگر توسط این عملگرها تعریف می شوند. مهمترین آنها عبارتند از :

- Set Intersection: $\cap$
- Division: ÷
- Natural Join: $\bowtie$

هر عملگرجبر رابطه ای روی یک یا دو رابطه به عنوان ورودی عمل کرده و یک رابطه جدید را به عنوان نتیجه تولید می کنند .

Selection

عملگر انتخاب (selection) یک عملگر یکتائی است که سطرهائی از یک رابطه را انتخاب می کند. فرم کلی آن به صورت زیر است :

$\sigma_{\text{شرط}}(R)$

خروجی عملگر selection رابطه ای است شامل سطرهایی از رابطه R که شرط موردنظر در آنها برقرار بوده است .

کاردینالیتی جدول حاصل کمتر یا مساوی جدول اولیه است اما درجه آنها تفاوت نمی کند .

شرط می تواند توسط علائم = ، $\neq$ ، <> ، $<$، $\leq$ ، $\geq$ ، (and) $\wedge$، (or) $\vee$ و (not)~ ساخته شود.

Projection

عملگر پروژه (projection) عملگر یکتائی که ستون هائی از یک رابطه را انتخاب می کند. شکل کلی آن به صورت زیر است :

$\pi a_1,\ldots,a_n(R)$

$a_1,\ldots,a_n$ مجموعه از اسامی صفات خاصه است که از رابطه R انتخاب می شوند. نتیجه عمل پروژه جدولی شامل کلیه تاپل های رابطه R است که محدود به مجموعه صفات مشخص شده است .

اگر در جدول حاصل سطرهایی مشابه هم باشند با هم ترکیب می شوند و سطرهای تکراری حذف می شوند. معمولا یک کلید کاندید را نگه می داریم تا کاردینالیتی حفظ شود. درجه جدول حاصل کمتر یا مساوی جدول اولیه است .

Cartesian Product

ضرب دکارتی (Cartesian Product) عملگری است که روی دو جدول کار می کند و جدول جدیدی را می دهد که که یک رکورد برای هر جفت رکورد ممکن از هر دو جدول دارد. فرم کلی آن به صورت زیر است :

$R \times S$

رکوردهای های رابطه R با کلیه رکوردها رابطه S به این صورت ترکیب می شوند که اولین سطر از رابطه R در کنار اولین سطر رابطه S در جدول حاصل قرار می گیرد و به همین ترتیب تا آخرین سطر S اضافه می شود. همین عمل مجددا برای سطرهای دیگر رابطه R تکرار می شود .

R

ColA	ColB
A	1
B	2
D	3

S

ColA	ColC
A	1
C	2

R × S

ColA	ColB	ColA	ColC
A	1	A	1
A	1	C	2
B	2	A	1
B	2	C	2
D	3	A	1
D	3	C	2

در جدول حاصل احتمال تکرار شدن ستون ها وجود دارد .

درجه جدول حاصل برابر مجموع درجات دو جدول و کاردینالیتی آن برابر حاصل ضرب کاردینالیتی دو رابطه می باشد .

ضرب دکارتی در جبر رابطه ای متفاوت از آنچه درتئوری مجموعه است تعریف می شود .

Set Union

عملگر اجتماع (union) یک عملگر دوتائی است که مشابه عمل اجتماع در تئوری مجموعه ها عمل می کند. فرم کلی آن به صورت زیر است :

$S \cup R$

اجتماع دو رابطه R و S جدولی است شامل کلیه تاپل های رابطه R و رابطه S .

دو رابطه ای که روی آنها عمل اجتماع انجام می شود باید همساز (compatible) باشند، یعنی باید دارای مجموعه صفات خاصه یکسان باشند .

R

ColA	ColB
A	1
B	2
D	3
F	4
E	5

S

ColA	ColB
A	1
C	2
D	3
E	4

R∪S

ColA	ColB
A	1
B	2
D	3
F	4
E	5
C	2
E	4

درجه جدول حاصل تفاوتی نمی کند اما کاردینالیتی آن برابر با مجموع سطرهای هردو جدول منهای سطرهای مشترک است .

Difference

عملگر تفاضل (difference) یک عملگر دو تائی است و مشابه عمل تفاضل در تئوری مجموعه ها است. فرم کلی آن به صورت زیر است :

R - S

تفاضل R و S جدولی است که شامل کلیه تاپل هایی است که در R هست ولی در S نیست. سطر اول رابطه R با کلیه سطرهای رابطه S مقایسه می شود هر کدام که در رابطه S نبود در جدول حاصل قرار می گیرد .

R

ColA	ColB
A	1
B	2
D	3
F	4
E	5

S

ColA	ColB
A	1
C	2
D	3
E	4

R – S

ColA	ColB
B	2
F	4
E	5

S – R

ColA	ColB
C	2
E	4

دو رابطه ای که روی آنها عمل تفاضل انجام می شود باید همساز باشند.

کاردینالیتی جدول حاصل برابر کاردینالیتی رابطه R منهای سطرهای مشابه است. درجه آنها تفاوتی نمی کند .

Rename

عملگر تغییر نام (rename) یک عملگر یکتائی است که برای تغییر نام صفات خاصه یک رابطه یا نام خود رابطه استفاده می شود. تغییر نام به صورت نوشته می شود :

$\rho_{A(B)}$

نتیجه عمل تغییر نام روی رابطه B همان رابطه B است با نام جدید A به بیان دیگر رابطه B را به A تغییر نام می دهد .

Intersection

عملگر اشتراک (intersection) بر اساس عمل اشتراک مجموعه ها می باشد. فرم کلی آن به صورت زیر است :

$S \cap R$

جدول حاصل از اشتراک دو رابطه R و S جدولی است شامل کلیه تاپل هایی که در هر دو جدول وجود دارد .
دو رابطه ای که روی آنها عمل اشتراک انجام می شود باید همساز باشند.

R

ColA	ColB
A	1
B	2
D	3
F	4
E	5

S

ColA	ColB
A	1
C	2
D	3
E	4

$R \cap S$

ColA	ColB
A	1
D	3

Division

عملگر تقسیم (division) روی دو رابطه انجام می شود. فرم کلی آن به صورت زیر است :

$R \div S$

حاصل تقسیم رابطه R بر رابطه S رابطه ای است شامل کلیه تاپل هائی از R برای صفات خاصه مشترک در رابطه S نیز وجود دارد. در جدول حاصل صفات خاصه ای از R اضافه می شود که در S نیست .

R

ColA	ColB
A	α
A	β
A	δ
B	α
B	δ
C	α
C	β

S

ColB
α
β

$R \div S$

ColA
A
C

عمل تقسیم توسط عملگر های مبنائی به صورت زیر شبیه سازی می شود:

$T := \pi_{a1,...,an}(R) \times S$
$U := T - R$
$V := \pi_{a1,...,an}(U)$
$W := \pi_{a1,...,an}(R) - V$

Join

الحاق طبیعی (Natural Join) یک عملگر دوتائی است که به صورت زیر نوشته می شود:

$R \bowtie S$

نتیجه الحاق طبیعی رابطه ای است شامل کلیه ترکیبات تاپل های R و S است که صفات خاصه مشترک آنها برابر است .

الحاق دو رابطه زیر مجموعه ای از ضرب دکارتی است. نتیجه ضرب دکارتی بدون هیچ شرطی است و ممکن است اطلاع جدیدی را ندهد. ولی عملگر الحاق با استفاده از کلید خارجی دو رابطه را با هم ترکیب کرده و با حذف سطرهائی از ضرب دکارتی اطلاع معنی داری را از ترکیب جداول می دهد .

دو رابطه که در الحاق شرکت می کنند باید دارای صفت خاصه مشترکی باشند .

مثال. جدول Loan و Borrower که به ترتیب حاوی مشخصات وام ها و وام گیرنده ها هستند را درنبظر بگیرید .

Loan

Loan_no	Branch_name	amount
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

Borrower

Custumer_name	Loan_no
Jones	L-170
Smith	L-230
Hayes	L-155

Loan ∞ Borrower

Loan_no	Branch_name	amount	Custumer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

مثال. در جدول Borrower کلید خارجی فیلد Loan_no است. الحاق جداول Loan و Borrower اسامی وام گیرنده ها و مقدار وام آنها را می دهد .

وقتی عمل الحاق روی دو رابطه انجام می شود بعضی داده ها از دست می روند. بعضی وقت ها این داده ها اطلاعات مفیدی را دارند. الحاق خارجی (outer join) جداول را به نحوی ترکیب می کند که داده های مورد نظر در جدول نتیجه باقی بمانند .

بسته به اطلاعاتی که حفظ می شود سه نوع الحاق خارجی وجود دارد :

- left outer join
- right outer join
- full outer join

left outer join

نتیجه الحاق چپ مجموعه کلیه تاپل های رابطه R و S است که صفات خاصه مشترک آنها یکسان است بعلاوه تاپل هایی در R که برای صفت خاصه مشترک همنظیری در S ندارد. برای این تاپل ها در صفات خاصه ای که از S اضافه می شوند مقدار nullقرارداده می شود .

الحاق چپ به صورت زیر نوشته می شود:

$R \rtimes S$

مثال. الحاق خارجی چپ دو جدول Loan و Borrower به صورت زیر می شود:

Loan⋈ Borrower

Loan_no	Branch_name	amount	Custumer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	Null

الحاق چپ با استفاده ازالحاق طبیعی و اجتماع بدست می آید:

$R \rtimes S = S \cup (R \bowtie S)$

right outer join

الحاق راست مشابه الحاق چپ است با این تفاوت که کلیه مقادیر رابطه سمت راست عملگر الحاق در نتیجه ظاهر می شود .

$R \ltimes S$

مثال. الحاق خارجی راست دو جدول Loan و Borrower به صورت زیر می شود :

Loan⋉ Borrower

Loan_no	Branch_name	amount	Custumer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	Null	Null	Hayes

الحاق چپ با استفاده ازالحاق طبیعی و اجتماع بدست می آید:

$R \ltimes S = R \cup (R \bowtie S)$

full outer join

الحاق خارجی کامل یا به طور خلاصه الحاق خارجی الحاق خارجی چپ و راست را با هم ترکیب می کند. نتیجه الحاق کامل خارجی مجموعه کلیه ترکیبات تاپل های R و S است که صفات خاصه مشترک آنها برابر است بعلاوه تاپل هائی در S که در R نیستند و تاپلهای R که در S وجود ندارند .

الحاق خارجی دو رابطه R و S به صورت زیر نوشته می شود:

$R \rtimes S$

مثال. الحاق خارجی دو جدول Loan و Borrower به صورت زیر می شود :

Loan⋊ Borrower

Loan_no	Branch_name	amount	Custumer_name
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	Null
L-155	Null	Null	Hayes

الحاق خارجی کامل با اجتماع الحاق چپ و الحاق راست بدست می آید:

$R \ltimes S = (R \bowtie S) \cup (R \propto S)$
$R \ltimes S = R \cup S \cup (R \bowtie S)$

مثال های کاربردی

مثال بانک. رابطه های زیر را درنظر بگیرید:

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
account (account_number, branch_name, balance)
loan (loan_number, branch_name, amount)
depositor (customer_name, account_number)
borrower (customer_name, loan_number)

سوال۱. کلیه شماره وام هائی که مقدارشان از ۱۲۰۰ بیشتر است را پیدا کنید.

$\pi_{\text{loan\_number}}(\sigma_{\text{amount} > 1200}(\text{loan}))$

سوال۲. کلیه مشتریانی که یک وام، یک حساب یا هردو را پیدا کنید.

$\pi_{\text{customer\_name}}(\text{borrower}) \cup \pi_{\text{customer\_name}}(\text{depositor})$

سوال۳. کلیه مشتریانی که یک وام و یک حساب در بانک دارند را پیدا کنید..

$\pi_{\text{customer\_name}}(\text{borrower}) \cap \pi_{\text{customer\_name}}(\text{depositor})$

سوال٤. اسامی کلیه مشتریانی که یک وام در شعبه Perryridge دارند را پیدا کنید ..

$\pi_{\text{customer\_name}}(\sigma_{\text{branch\_name="Perryridge"}}(\text{borrower} \bowtie \text{loan}))$

سوال٥. اسامی کلیه مشتریانی که یا حساب دارند یا وام گرفته اند (ولی نه هردو) را پیدا کنید.

$\pi_{\text{customer\_name}}((\sigma_{\text{account\_number is null or loan\_number is null}}(\text{depositor} \ltimes \text{borrower}))$

سوال٦. اسامی کلیه مشتریانی که یک وام در شعبه Perryridge دارند ولی هیچ حسابی در هیچ شعبه ندارند را پیدا کنید..

$\pi_{\text{customer\_name}}(\sigma_{\text{branch\_name="Perryridge"}}(\text{borrower} \bowtie \text{loan})) - \pi_{\text{customer\_name}}(\text{depositor})$