

Fundamentals Concepts of Data Bases

اصول طراحی پایگاه داده ها

جلسه ۴ : مدل رابطه ای (Relational Model)

مدرس : اسماعیل نورانی

<http://www.nurani.ir/> - Info@nurani.ir

<http://www.nurani.ir/> - Info@nurani.ir

نمونه ای از یک رابطه

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
<i>Jones</i>	Main	Harrison
<i>Smith</i>	North	Rye
<i>Curry</i>	North	Rye
<i>Lindsay</i>	Park	Pittsfield

attributes
(or columns)

tuples
(or rows)

customer

<http://www.nurani.ir/> - Info@nurani.ir

انواع صفت کلید

► اگر قرار دهیم $K \subseteq R$

- یک ابر کلید است (Super Key) از رابطه R است اگر مقادیر صفات داخل k قابلیت منحصر بفرد بودن در بین نمونه های رابطه R را داشته باشد.
- مثال : $\{customer_name, customer_street\}$ و $\{customer_name\}$ هر دو ابر کلید رابطه مشتری محسوب میشوند ، البته با این فرض که هیچ دو مشتری اسم یکسان نداشته باشند .
- K کلید کاندید (candidate key) رابطه R است اگر مینیمال باشد یعنی هیچ زیر مجموعه ای از آن خاصیت منحصر بفرد بودن تاپلها را حفظ نکند (اگر یکی از صفات داخل آن حذف شود ، خاصیت منحصر بفرد بودن از بین برود)
- کلید اصلی (Primary Key) یکی از کلیدهای کاندید که بعنوان کلید رابطه انتخاب میشود

<http://www.nurani.ir/> - Info@nurani.ir

کلید در مدل رابطه ای :

- با اصطلاح عام کلید کاندید (Candidate Key) مطرح است .
تعریف : هر زیر مجموعه از مجموعه عنوان رابطه به صورت $A_1 A_2 \dots A_k$ که دارای دوخاصیت زیر باشد کلید کاندید است :
- ۱ - یکتایی مقدار داشته باشد در طول حیات رابطه (Uniqueness) یعنی در هیچ دو Tuple ای از رابطه این ترکیب مقدار یکسان نداشته باشد .
 - ۲ - کاهش ناپذیری Irreducibility یا Minimality (ایجاز یا کهنگی) داشته باشد . یعنی اگر هر یک از اجزای تشکیل دهنده این ترکیب را ازش بگیریم خاصیت یکتایی مقدار آن از بین برود .

نام رابطه	کلید کاندید
S	S#
P	p#
SP	(S#,P#)

<http://www.nurani.ir/> - Info@nurani.ir

نکته : اگر خاصیت دوم برقرار نباشد ، یعنی زیر مجموعه ای از مجموعه Heading وجود داشته باشد که یکتایی مقدار داشته باشد ولی کاهش ناپذیری نداشته باشد ، به آن Supper Key می گویند (یعنی درون خود حد اقل یک کلید کاندید دارد) اما Supper Key ، Codd را به عنوان یک مفهوم اساسی در مدل رابطه ای مطرح نمی کند .

آیا وجود حد اقل یک کلید کاندید در رابطه محرز است ؟

بله . هر رابطه کلید کاندید دارد . زیرا در بدترین حالت خود مجموعه عنوان می تواند کلید کاندید باشد ، زیرا مجموعه Body ، Tuple تکراری نمی تواند داشته باشد

(S#,P#, DATE)

SPd (S# , P# , Qty , Date)

مثال :

پس اضافه کردن ستون نمی تواند
همینطوری انجام شود . یک ستون
اضافه گردید و کلید کاندید کاملا تغییر
کرد .

S1 p1 100 d1

S2 p1 100 d2

.....

به رابطه ای که مجموعه عنوانش کلید کاندیدش باشد رابطه تمام کلید All-Key می گویند .
کلید Identifier تشخیص Tuple ها است و هرچه کلید طولانی تر باشد برای یافتن Tuple باید مسیر پیچیده تری را رفت

<http://www.nurani.ir/> - Info@nurani.ir

:Primary Key

کلید کاندیدی است که طراح انتخاب می کند بنابر ملاحظات محیطی (یعنی یکی از کلید های کاندید)
مثلا : رابطه S دارای کلیدهای کاندید S# و SNAME است ، اما طراح ممکن است S# را به عنوان کلید اصلی در نظر بگیرد .

آن کلید کاندیدی که توسط آن Access بیشتری به Data انجام می شود به عنوان کلید اصلی مطرح می شود که بستگی به نظر طراح دارد .

کلید اصلی : شناسه تاپل است در رابطه ، امکان آدرس دهی به تاپل در رابطه است در محیط انتزاعی (الزاما صرف معرفی کلید اصلی ایجاب نمی کند حتما سیستم روی کلید اصلی Index بزند بلکه باید درخواست کنید *کلید اصلی حتما در تعریف رابطه باید قید شود . یعنی جزئی از شمای اصلی پایگاه است .

<http://www.nurani.ir/> - Info@nurani.ir

(کلید خارجی):

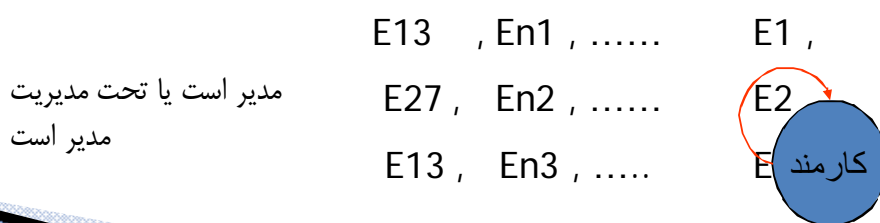
تعریف: A_i در R_2 کلید خارجی است اگر در رابطه ای مثلا R_1 کلید اصلی باشد.

مثال: صفت خاصه $S\#$ در SP کلید خارجی است زیرا $S\#$ در S کلید اصلی است.
مثال: $P\#$ در SP کلید خارجی است زیرا $P\#$ در P کلید اصلی است.

نکات: در تعریف کلید خارجی R_1 و R_2 لزوما مجزا نیستند

مثال:

کارمند EMP (EMP# , EMPNAME , , EMPMGR#)



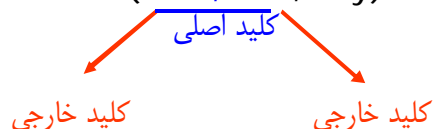
<http://www.nurani.ir/> - Info@nurani.ir

$EMP\#$ کلید اصلی است. $EMGR\#$ در همین رابطه کلید خارجی است.

نکته: یک رابطه $1:N$ یکسویه توسط کلید خارجی پیاده سازی می شود.

نکته: لزومی ندارد کلید خارجی یک رابطه جزئی از کلید اصلی رابطه باشد. هرچند در SP چنین است.

$SP (S\# , P\# , Qty)$



در مدلینگ ارتباط $1:N$ (یک به n) دوسویه از کلید خارجی استفاده می شود و رابطه ای برای بیان ارتباط آن دو موجودیت طراحی می شود که کلید اصلی اش تشکیل شده از کلیدهای اصلی دو رابطه است.

<http://www.nurani.ir/> - Info@nurani.ir

نکته : فایده کلید خارجی چیست ؟

امکانی است برای نشان دادن ارتباط بین موجودیتها . مثل اینکه در رابطه SP از طریق کلید خارجی S# و P# ارتباط دو موجودیت S و P را نشان می دهیم . و نیز امکان ارجاع به موجودیتهاست .
به عنوان مثال وجود S1 در رابطه SP ارجاعی است به نمونه موجودیت S1 در رابطه S .
آیا تنها امکان نشان دادن ارتباط بین موجودیتها کلید خارجی است ؟
خیر . در حالت کلی وجود هر صفت خاصه مشترک در مجموعه عنوان دو رابطه نشاندهنده نوعی ارتباط بین دو نوع موجودیتی است که آن دو رابطه نمایشگر آنها هستند .

مثال :

S(S# , SNAME , STATUS , CITY)

P(P# , , CITY)

CITY در دو رابطه نوعی ارتباط را نشان می دهد مثلا "در یک شهر بودن" "در یک شهر نبودن" و CITY در هیچیک از دو رابطه کلید خارجی نیست . زیرا هیچ جا کلید اصلی نیست .
برای نمایش ارتباط بین دو موجودیت : دو راه داریم :

۱ - استفاده از مفهوم کلید خارجی

۲ - استفاده از یک صفت خاصه مشترک

کلید خارجی بیانگر یک ارتباط محوری و اساسی می باشد اما صفت خاصه مشترک چنین نیست .

مثال :

S(S# , SNAME , . . . , CITY)

P(P# , , CITY)

J(J# , JNAME , CITY)

SPJ(S# , P# , J# , . . . , CITY)

CITY بیانگر نوعی ارتباط است ولی J# و P# و S# بیانگر یک ارتباط محوری و اساسی می باشد

<http://www.nurani.ir/> - Info@nurani.ir

INTEGRITY RULES

قواعد جامعیت

مجموعه قواعدی که به کمک آنها سیستم **صحت** و **دقت** داده های ذخیره شده در پایگاه و نیز ارتباطات بین انواع موجودیتها را کنترل می کند . (Tuple هم موجودیتها را نشان می دهد و هم ارتباطات بین آنها را (دید جدولی))

قواعد جامعیت

۱- کاربری (خاص) : تعریف شده توسط کاربر (منظور شخص DBA) است . کاربران

خارجی باید درخواست خود را به DBA بدهند. (UDR=User Defined Rules)

۲- عام (متا قاعده Meta-Rule) : مستقل از Data های خاص مطرحند ،

Independent از Data های محیط خاص

قواعد جامعیت کاربری (خاص)

میدانی : قاعده ای است ناظر به مقادیر یک میدان

ستونی : علاوه بر قواعد میدانی که بر روی مجموعه مقادیر است ، قواعد جامعیت ستونی اخص از محدودیت میدانی یعنی برای صفات خاصه بخصوصی در رابطه بخصوصی محدودیتی قایل بشویم

<http://www.nurani.ir/> - Info@nurani.ir

مثال میدانی : شماره تهیه کنندگان حتما به این صورت است :

$$R2 = 1111 < dddd < 8888$$

$$R3 = \text{City In } \{ C1, C2, C3, C4, C5 \}$$

ستونی : علاوه بر قواعد میدانی که بر روی مجموعه مقادیر میدان ناظر است ، قواعد جامعیت ستونی اخص از محدودیت میدانی است ، یعنی برای صفات خاصه بخصوصی در رابطه بخصوصی محدودیتی قایل بشویم .

مثال :

$$R4 = 0 < QTY < 100$$

$$R5 = \text{QTY برای تهیه کنندگان ساکن C1 و C2 که قطعات P1 و P2 را تهیه می کنند نمی تواند بیشتر از 50 باشد}$$

(*) یک DBMS واقعی باید امکان بدهد تا DBA انواع قواعد جامعیت را بیان و ایجاد کند و سیستم باید این قواعد را پیاده و عمل کند .

(*) شمای ادراکی فقط از تعریف رابطه تشکیل نمی شود بلکه قواعد جامعیت مربوط به آن رابطه جزئی از شمای ادراکی است .

(*) بخش عمده Overhead عملیات DBMS مربوط به این قواعد جامعیت است .

<http://www.nurani.ir/> - Info@nurani.ir

قواعد جامعیت عام } C1 : قاعده جامعیت موجودیتی Entity Integrity Rule ناظر به کلید اصلی است .
C2 : قاعده جامعیت ارجاعی Refrential Integrity Rule ناظر به کلید خارجی است

نکته : کلید خارجی هم مثل کلید اصلی باید در تعریف رابطه آورده شود

قاعده C1 : می گوید : هیچ جز تشکیل دهنده کلید اصلی نمی تواند Null باشد (هیچ مقدار Null Value داشته باشد)

Null Value : ۱- مقدار ناشناخته : Unknown : مثلا STATUS تهیه کننده ای مشخص نباشد .

۲- مقدار غیر قابل اعمال : Inapplicable : مثلا شماره شناسنامه همسر

برای شخص مجرد غیر قابل اعمال است

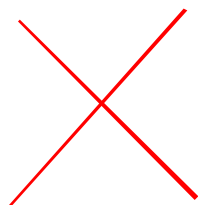
<http://www.nurani.ir/> - Info@nurani.ir

Null Value با صفر یا Blank فرق می کند زیرا آنها مقدارند . هر چند سیستم باید امکانی برای بازنمایی Null-Value داشته باشد. (در کتابها Null-Value را با ؟ نمایش می دهند)

* باید به عنوان یک مقدار ، یک مجموعه عملیاتی مشخص ناظر بر آن وجود داشته باشد (محاسباتی منطقی مقایسه ای)

Null (< = >) Null

به عنوان مثال در مقایسه دو null value چه می توان گفت ؟



<http://www.nurani.ir/> - Info@nurani.ir

نظر Date : مفهوم Null مخرب مدل رابطه است . بلکه مفهوم دیگری به نام

Missing Information باید مطرح شود که ساده تر است .

پس Null ، Overhead و گرفتاریها ی خود را دارد.

دلیل قاعده C1 : زیرا کلید اصلی شناسه Tuple است . شناسه یک نمونه مشخص از یک نمونه موجودیت به معنای عام و متمایز از هر نمونه دیگر ، و کلید اصلی عامل تمیز یک نمونه با نمونه دیگر است و عامل تمیز خود نمی تواند ناشناخته باشد

باید در تعریف رابطه ، Null-Value پذیری یا ناپذیری صفت خاصه را بیان کرد (DBMS باید این امکان را بدهد)

Create Table s

(S# char(6) ,

Sname char(20) NOT NULL ,

....

City Char (12)) PRIMARY KEY S# ;

یعنی در طول حیات این رابطه Sname نمی تواند Null باشد و Sname باید مقدار مشخصی داشته باشد

<http://www.nurani.ir/> - Info@nurani.ir

قواعد جامعیت :

قواعد خاص : تعریف شده توسط کاربر
 قواعد عام :
 C1 : قاعده جامعیت موجودیتی
 C2 : قاعده جامعیت ارجاعی : ناظر به کلید خارجی

c2: اگر A_i در R_2 کلید خارجی باشد (یعنی A_i در رابطه R_1 کلید اصلی است)

$R_2.A_i$:
 ۱- می تواند null باشد اگر جزء تشکیل دهنده کلید اصلی در R_2 نباشد
 در غیر اینصورت (یعنی در صورتیکه null نباشد)
 ۲- باید حتما مقداری داشته باشد که در R_1 وجود دارد.

R1 S(S#..... , City)_

S1
S2
S3

در S وجود ندارد . (Unmatchable مقدار)

R2 Sp S# P# QTY

S1 P1
S1 P1
S1 P2
S2 P1
S3 P2
S3 P1
S4 P1

مثال :

<http://www.nurani.ir/> - Info@nurani.ir

* R_2 به R_1 ارجاع دارد یا رجوع می کند.

R2 → R1

Referenced رابطه مرجع

نمودار ارجاع یا گراف ارجاع (از طریق مقدار کلید

خارجی)

Referencing رابطه رجوع کننده

مثال : $P \leftarrow Sp \rightarrow S$

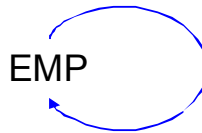
مثال دیگر: EMP(EMP#, EMP name,, EMP MGR#.....)

کلید اصلی

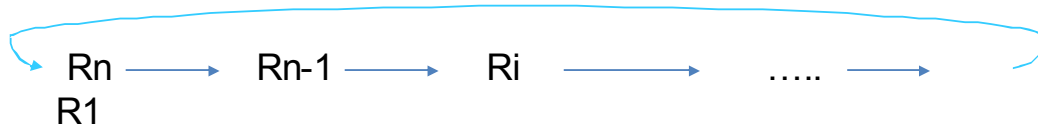
کلید خارجی

این رابطه بر طبق بحث قبلی چون R_1 و R_2 برابرند. بنابراین آنرا Self Referencing گوئیم

<http://www.nurani.ir/> - Info@nurani.ir



شکل کلی یک مسیر ارجاع : $R_n \rightarrow R_{n-1} \rightarrow \dots \rightarrow R_2 \rightarrow R_1$



حلقه ارجاع با n رابطه

مثال دیگر:

$EMP(E\#, Ename, \dots, DEPT\# \dots)$

↓
کلید خارجی

$DEPT(DEPT\#, DEPT\ PHONE, \dots, DEPTMGR\#, \dots)$

↓
شماره کارمند کلید خارجی

<http://www.nurani.ir/> - Info@nurani.ir

DEPTMGR#



سؤال: دو قاعده $c1$ و $c2$ چگونه باید به DBMS داده شود و چگونه رعایت شوند؟

نکته: قواعد $C1$ و $C2$ باید به نحوی به DBMS داده شوند.

نکته: باید توسط DBMS رعایت شوند

قاعده $C1$ از طریق معرفی کلید اصلی در تعریف رابطه و نیز استفاده از گزینه NOT Null

CREATE RELATION S

(S# NOT NULL,
... بدین طریق C1 تعریف و رعایت می گردد

.....

.....

City CHAR (12))

PRIMARY KEY S# ;

برای معرفی قاعده C2: در تعریف رابطه کلید خارجی (در صورت وجود) باید مشخص شود.

<http://www.nurani.ir/> - Info@nurani.ir

CREATE TABLE SP

(S# NOT NULL ,

P# NOT NULL,

QTY Integer)

PRIMARY KEY (S# , p#)

FOREIGN KEY S# references S

DELETE CASCADE

FOREIGN KEY P# references P

DELETE CASCADE;

<http://www.nurani.ir/> - Info@nurani.ir

ایجاد جدول sp در Oracle :

- ▶ CREATE TABLE SP
- ▶ (snum character(3) ,
- ▶ pnum character(3) ,
- ▶ qty integer ,
- ▶ CONSTRAINT pk_snumpnum PRIMARY KEY
- ▶ (snum,pnum),
- ▶ CONSTRAINT fk_snum
- ▶ FOREIGN KEY (snum)
- ▶ REFERENCES s(snum)
- ▶ ON DELETE CASCADE ,
- ▶ CONSTRAINT fk_pnum
- ▶ FOREIGN KEY (pnum)
- ▶ REFERENCES p(pnum)
- ▶ ON DELETE CASCADE)

<http://www.nurani.ir/> - Info@nurani.ir

چگونه قاعده C2 را رعایت کنیم :

سه مکانیسم برای رعایت C2 متصور است :

نکته : قاعده C2 باید در جمیع عملیات روی پایگاه رعایت شود .

مثال : فرض می کنیم در رابطه S تهیه کننده 'S7' وجود ندارد .

INSERT INTO SP
VALUES (S7,P1,60) ;
قاعده C2 می گوید این عمل باید Reject شود

درج یک تاپل

مثال : فرض می کنیم در رابطه S موجودیت S3 را داریم :

DELETE FROM S
WHERE S#='S3' ;

آیا باید اجرا شود ؟ اگر بله با تاپلهای Referencing چه باید کرد ؟؟ و اگر نه ...

<http://www.nurani.ir/> - Info@nurani.ir

سه مکانیسم برای رعایت قاعده C2 :

در عمل Delete (به عنوان مثال) :

۱- روش Cascade : انتشار عملیات به رابطه Referencing وقتی که عمل در رابطه Referenced

انجام شود .مثال : با حذف تاپلهای S3 از S ، تمام تاپلهای رجوع کننده به S3 در رابطه های دیگر نیز باید حذف گردد.

1- DELETE FROM S WHERE S#='S3'

2- DELETE FROM SP WHERE S#='S3'

این روش نوعی Propagating Update است . این روش در بعضی از سیستمهای موجود وجود دارد.

۲- روش Restricted : انجام عمل بطور مشروط

در این روش تاپل مرجع (Referenced) حذف نمی شود و بعبارتی دیگر تا زمانیکه تاپل مرجع تاپل رجوع کننده ای دارد ، عمل حذف انجام نمی شود .

۳- روش هیچمقدارگذاری: (Nullifying)

در این روش با حذف تاپل مرجع مقدار کلید خارجی در تاپل رجوع کننده Null می شود و زمانی امکانپذیر است که کلید خارجی جزء کلید اصلی نباشد .

<http://www.nurani.ir/> - Info@nurani.ir

متداولترین روش : روش اول است .

نکته : در سیستمهای پیش رابطه ای قواعد جابجایی نظیر C1 و C2 وجود دارد و بویژه قاعده C2 در DS ذاتی این سیستمها وجود دارد . یعنی ایده روش Cascade در سیستمهای پیش رابطه ای (مثلا در سلسله مراتبی با حذف نمونه پدر ، نمونه فرزند نیز حذف می شود یا مثلا شبکه ای با حذف نمونه مالک و...) چیز جدیدی نیست و نیاز به تصریح ندارد

ولی در سیستمهای رابطه ای باید تصریح شود زیرا رابطه ها حسب ظاهر منفک از یکدیگرند و ارتباط بین رابطه ها علیرغم مجزا بودن دو رابطه به یکی از روشهای زیر انجام می گیرد

از طریق مکانیسم ارجاع

یا

از طریق صفت خاصه مشترک که در Heading دو رابطه در سیستم رابطه ای باید
تصریح شود

و عمل Cascade در سیستم رابطه ای باید تصریح شود.

نظر Date : بهتر است DBMS قبل از انجام عمل Warning دهد و بعد از کاربر سوال کند

<http://www.nurani.ir/> - Info@nurani.ir

زبان پرس و جو

▶ زبانی که از طریق آن کاربر اطلاعات مورد درخواست خود را از پایگاه داده درخواست میکند .

▶ انواع زبانها

- رویه ای (Procedural)
- نارویه ای (Non Procedural - Declarative)

▶ زبانهای استاندارد پرس و جو

- جبر رابطه ای
- حساب رابطه ای

<http://www.nurani.ir/> - Info@nurani.ir

جبر رابطه ای (Relational Algebra)

شش عملگر اصلی

- select: σ (انتخاب)
- project: Π (تصویر)
- union: \cup (اجتماع)
- set difference: $-$ (تفاضل)
- Cartesian product: \times (ضرب کارتیزین)
- rename: ρ (تغییر نام)

▶ هر کدام از عملگرهای فوق یک یا چند رابطه را بعنوان ورودی گرفته و رابطه را بعنوان خروجی تولید میکنند .

<http://www.nurani.ir/> - Info@nurani.ir

Select

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

<http://www.nurani.ir/> - Info@nurani.ir

Select

- ▶ $\sigma_p(r)$

p شرطی است که برای استخراج سطرهای خاصی اعمال میشود .

\wedge (and), \vee (or), \neg (not)

که در مابین صفات می توانند ظاهر میشود .

<attribute> op <attribute> or <constant>

$Op : =, \neq, >, \geq, <, \leq$

- ▶ Example of selection:

account where $branch_name = "Perryridge"$

- ▶ select $branch_name = "Perryridge"(account)$

- ▶ $\sigma_{branch_name = "Perryridge"}(account)$

<http://www.nurani.ir/> - Info@nurani.ir

Project

► Relation r :

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

<http://www.nurani.ir/> - Info@nurani.ir

پرتو (Project)

► Notation: $\Pi_{A_1, A_2, \dots, A_k}(r)$

که در آن A_1, A_2, \dots صفات و r نام یک رابطه است .
حاصل شامل ستونهای مورد نظر از رابطه که در آن سرهای تکراری حذف شده
اند

$\Pi_{\text{account_number}, \text{balance}}(\text{account})$

<http://www.nurani.ir/> - Info@nurani.ir

اجتماع (Union)

► Relations r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

■ $r \cup s$:

A	B
α	1
α	2
β	1
β	3

<http://www.nurani.ir/> - Info@nurani.ir

Union

► Notation: $r \cup s$

► نحوه تعریف :

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

► تعداد ستونهای هر دو رابطه باید یکسان باشد

► ستونهای متناظر باید از جنس دامنهد سازگار باهم باشند (و نه لزوما هم معنی)

► مثال : کلیه افرادی که یا وام گرفته اند و یا حساب بانکی دارند

$$\Pi_{customer_name}(depositor) \cup \Pi_{customer_name}(borrower)$$

<http://www.nurani.ir/> - Info@nurani.ir

عملگر تفاضل

► Relations r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

■ $r - s$:

A	B
α	1
β	1

<http://www.nurani.ir/> - Info@nurani.ir

عملگر تفاضل

► Notation $r - s$

► تعریف :

$$r - s = \{t \mid t \in r \text{ and } t \notin s\}$$

شرایط موجود در **UNION** در مورد تفاضل نیز صدق میکند

<http://www.nurani.ir/> - Info@nurani.ir

ضرب کارتزین

■ Relations r, s :

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

■ $r \times s$:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

<http://www.nurani.ir/> - Info@nurani.ir

ضرب کارتزین

► Notation $r \times s$

► تعریف :

$$r \times s = \{tq \mid t \in r \text{ and } q \in s\}$$

فرض بر این است که صفات رابطه ها هیچ اشتراکی با هم ندارند. $R \cap S = \emptyset$

► که در غیر اینصورت بایستی از تغییر نام برای صفات تکراری استفاده کرد .

► اگر رابطه r ، m ستون و رابطه s ، n ستون داشته باشد r^*s ، $m+n$ ستون خواهد داشت

► اگر رابطه r ، a سطر و رابطه s ، b سطر داشته باشد r^*s ، $a*b$ سطر خواهد داشت

<http://www.nurani.ir/> - Info@nurani.ir

عملگرهای ترکیبی

▶ Example: $\sigma_{A=C}(r \times s)$

▶ $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

▶ $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

<http://www.nurani.ir/> - Info@nurani.ir

مثال بانکداری

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)

Example Queries

▶ تمامی وام‌هایی که بیش از \$۱۲۰۰ می باشند

$$\sigma_{amount > 1200} (loan)$$

■ شماره وام‌هایی که مقدار آنها بیش از \$1200 است

$$\Pi_{loan_number}(\sigma_{amount > 1200} (loan))$$

■ یافتن اسامی مشتریانی که یا وام یا حساب و یا هر دو را صاحب هستند

$$\Pi_{customer_name} (borrower) \cup \Pi_{customer_name} (depositor)$$

<http://www.nurani.ir/> - Info@nurani.ir

لیست عملگرها

▶ اگر E1, E2 عبارتی از جبر رابطه ای باشند

- $E_1 \cup E_2$
- $E_1 - E_2$
- $E_1 \times E_2$
- $\sigma_p(E_1)$,
- $\Pi_s(E_1)$,
- $\rho_x(E_1)$,

<http://www.nurani.ir/> - Info@nurani.ir

عملگرهای اضافی

Intersection اشتراک ▶

Natural join پیوند طبیعی ▶

Division تقسیم ▶

Assignment انتساب ▶

<http://www.nurani.ir/> - Info@nurani.ir

اشتراک

▶ Notation: $r \cap s$

▶ تعریف :

▶ $r \cap s = \{ t \mid t \in r \text{ and } t \in s \}$

▶ فرض :

◦ r, s تعداد ستونهای یکسانی داشته و ستونهای متناظر سازگارند

◦ توجه $r \cap s = r - (r - s)$

<http://www.nurani.ir/> - Info@nurani.ir

► Relation r, s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

► $r \cap s$

A	B
α	2

<http://www.nurani.ir/> - Info@nurani.ir

پیوند شرطی (teta join)

- به صورت X_θ نمایش داده می شود.
- زیر مجموعه ای از ضرب دکارتی است که شرط θ روی سطرهاى آن اعمال شده باشد.
- ستونهاى خارجى معادل ستونهاى ضرب دکارتى می باشد.
- این راه حل بسیار گران است
- زیرا جدول بساير بزرگى شامل همه ترکیبهاى ممکن سطرى و ستونى دو جدول تشکیل شده و فقط بخش کوچكى از آن به عنوان خروجى برگردانده می شود و بقیه دور ریخته می شود.

<http://www.nurani.ir/> - Info@nurani.ir

پیوند شرطی

Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

A	B	C	D	Bs	Ds	E
α	1	α	a	1	a	α
α	1	α	a	3	a	β
α	1	α	a	1	a	γ
α	1	α	a	2	b	δ
α	1	α	a	3	b	ϵ
β	2	γ	a	1	a	α
β	2	γ	a	3	a	β
β	2	γ	a	1	a	γ
β	2	γ	a	2	b	δ
β	2	γ	a	3	b	ϵ
.
.
.

■ $r \bowtie_{r.B=s.B} s$

A	B	C	D	Bs	Ds	E
α	1	α	a	1	a	α
α	1	α	a	1	a	γ
β	2	γ	a	2	b	δ
α	1	γ	a	1	a	α
α	1	γ	a	1	a	γ
δ	2	β	b	2	b	δ

$r * s$

<http://www.nurani.ir/> - Info@nurani.ir

پیوند طبیعی

■ Notation: $r \bowtie s$

▶ حاصل عبارت زیر مجموعه ای از ضرب کارتزین است و فقط سطرهاي را شامل میشوند که در آن مقدار موجود در ستون هم نام مساوی باشد که در اینصورت ستوان هم نام با مقدار یکسان فقط یکبار در خروجی ظاهر میشود.

▶ Example:

$R = (A, B, C, D)$

$S = (E, B, D)$

◦ Result = (A, B, C, D, E)

◦ $r \bowtie s$:

$$\Pi_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B=s.B \wedge r.D=s.D} (r \times s))$$

<http://www.nurani.ir/> - Info@nurani.ir

پیوند طبیعی

► Relations r, s :

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

■ $r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

A	B	C	D	Bs	Ds	E
α	1	α	a	1	a	α
α	1	α	a	3	a	β
α	1	α	a	1	a	γ
α	1	α	a	2	b	δ
α	1	α	a	3	b	ϵ
β	2	γ	a	1	a	α
β	2	γ	a	3	a	β
β	2	γ	a	1	a	γ
β	2	γ	a	2	b	δ
β	2	γ	a	3	b	ϵ
.
.
.

$r * s$

<http://www.nurani.ir/> - Info@nurani.ir

نیم پیوند (Semi-join)

■ Notation: $r \ltimes s$

► شبیه پیوند طبیعی است با این تفاوت که فقط ستونهای جدول اول را می دهد.

► ممکن است تعداد سطرهای حاصل به مراتب کمتر از پیوند طبیعی باشد
 ◦ چراکه با کنار رفتن ستونهای جدول دوم سطرهای تکراری حاصل شده و حذف خواهد شد.

► این عملگر جابجائی پذیر نیست $s \ltimes r \neq r \ltimes s$

► کاربرد : در پایگاه داده های توزیع شده

<http://www.nurani.ir/> - Info@nurani.ir

نیم پیوند (Semi-join)

► Relations r, s :

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

$r \bowtie s$

A	B	C	D
α	1	α	a
α	1	γ	a
δ	2	β	b

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

<http://www.nurani.ir/> - Info@nurani.ir

نامگذاری (Rename)

Notation: $\rho_b a$

► نام b روی رابطه a نیز گذاشته می شود.

عملگر تقسیم

► Notation: $r \div s$

► برای مواردی مناسب است که در پرس و جو لفظ "همه" استفاده شده است .

- $R = (A_1, \dots, A_m, B_1, \dots, B_n)$

- $S = (B_1, \dots, B_n)$

که در آن ستونهای ظاهر شده در خروجی برابر $R - S = (A_1, \dots, A_m)$

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in s (tu \in r) \}$$

عملگر تقسیم

■ Relations r, s :

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

r

B
1
2

s

■ $r \div s$:

A
α
β

■ Relations r, s :

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	E
a	1
b	1

s

■ $r \div s$:

A	B	C
α	a	γ
γ	a	γ

<http://www.nurani.ir/> - Info@nurani.ir

پرس و جوهای از مثال بانک

■ یافتن اسامی ، شماره وام و مقدار وام مشتریان

borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

$\Pi_{customer_name, loan_number, amount} (borrower \bowtie loan)$

customer_name	loan_number	amount
Adams	L-16	1300
Curry	L-93	500
Hayes	L-15	1500
Jackson	L-14	1500
Jones	L-17	1000
Smith	L-23	2000
Smith	L-11	900
Williams	L-17	1000

<http://www.nurani.ir/> Info@nurani.ir

Bank Example Queries

▶ اسامی مشتریانی که در هر دو شعبه Uptown و Downtown حساب دارند .

$$\Pi_{customer_name}(\sigma_{branch_name = \text{"Downtown"}}(depositor \bowtie account)) \cap \Pi_{customer_name}(\sigma_{branch_name = \text{"Uptown"}}(depositor \bowtie account))$$

Account			Depositor	
account_number	branch_name	balance	customer_name	account_number
A-101	Downtown	500	Hayes	A-102
A-102	Perryridge	400	Johnson	A-101
A-201	Brighton	900	Johnson	A-201
A-215	Mianus	700	Jones	A-217
A-217	Brighton	750	Lindsay	A-222
A-222	Redwood	700	Smith	A-215
A-305	Round Hill	350	Turner	A-305

<http://www.nurani.ir/> - Info@nurani.ir

Bank Example Queries

▶ اسامی مشتریانی که در همه شعب شهر Brooklyn حساب بانکی دارند .

$$\Pi_{customer_name, branch_name}(depositor \bowtie account) \div \Pi_{branch_name}(\sigma_{branch_city = \text{"Brooklyn"}}(branch))$$

Account			branch		
account_number	branch_name	balance	branch_name	branch_city	assets
A-101	Downtown	500	Brighton	Brooklyn	7100000
A-102	Perryridge	400	Downtown	Brooklyn	9000000
A-201	Brighton	900	Mianus	Horseneck	400000
A-215	Mianus	700	North Town	Rye	3700000
A-217	Brighton	750	Perryridge	Horseneck	1700000
A-222	Redwood	700	Pownal	Bennington	300000
A-305	Round Hill	350	Redwood	Palo Alto	2100000
			Round Hill	Horseneck	8000000

<http://www.nurani.ir/> - Info@nurani.ir

Outer Join

- ▶ عملگر توسعه یافته ای از پیوند طبیعی که از گم شدن اطلاعات جلوگیری میکند .
- ▶ در عین حال که عمل پیوند را انجام میدهد . سطرهایی که هیچ معادلی در طرف دیگر برای آنها پیدا نمیشود را نیز در خروجی ظاهر میکند به نحویکه در زیر ستون طرف مقابلش Null قرار میدهد

<http://www.nurani.ir/> - Info@nurani.ir

Outer Join – Example

▶ *loan*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

■ *borrower*

<i>customer_name</i>	<i>loan_number</i>
Jones	L-170
Smith	L-230
Hayes	L-155

▶ Join

loan ⋈ *borrower*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>	<i>customer_name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

<http://www.nurani.ir/> - Info@nurani.ir

Left Outer Join

▶ *loan*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

■ *borrower*

<i>customer_name</i>	<i>loan_number</i>
Jones	L-170
Smith	L-230
Hayes	L-155

■ Left Outer Join

loan $\sqcup\bowtie$ *borrower*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>	<i>customer_name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>

<http://www.nurani.ir/> - Info@nurani.ir

Right Outer Join

▶ *loan*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

■ *borrower*

<i>customer_name</i>	<i>loan_number</i>
Jones	L-170
Smith	L-230
Hayes	L-155

■ Right Outer Join

loan $\bowtie\sqcup$ *borrower*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>	<i>customer_name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	<i>null</i>	<i>null</i>	Hayes

<http://www.nurani.ir/> - Info@nurani.ir

Full Outer Join – Example

▶ *loan*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

■ *borrower*

<i>customer_name</i>	<i>loan_number</i>
Jones	L-170
Smith	L-230
Hayes	L-155

■ Full Outer Join

loan ⋈ *borrower*

<i>loan_number</i>	<i>branch_name</i>	<i>amount</i>	<i>customer_name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>
L-155	<i>null</i>	<i>null</i>	Hayes

<http://www.nurani.ir/> - Info@nurani.ir

نامگذاری (Rename)

Notation: $\rho_{b a}$

▶ مثال : اسامی مشتریان همشهری

▶ *customer* (*customer_name*, *customer_street*, *customer_city*)

▶ Customer $\bowtie_{\text{customer.name} \neq \text{p.name} \wedge \text{customer.city} = \text{p.city}}$ ($\rho_p (\Pi_{\text{name,city}}(\text{Customer}))$)

<http://www.nurani.ir/> - Info@nurani.ir

مقادیر Null

► Null مشخصه یک مقدار نا معلوم و یا غیر موجود است .

► هر گونه محاسبات ریاضی که در آن Null مشارکت کند نتیجه کلی Null را در بر خواهد داشت

- OR: (*unknown or true*) = *true*,
(*unknown or false*) = *unknown*
(*unknown or unknown*) = *unknown*
- AND: (*true and unknown*) = *unknown*,
(*false and unknown*) = *false*,
(*unknown and unknown*) = *unknown*
- NOT: (**not** *unknown*) = *unknown*

<http://www.nurani.ir/> - Info@nurani.ir

تغییر در پایگاه داده

► از طریق عملگرهای زیر

- Deletion
- Insertion
- Updating

<http://www.nurani.ir/> - Info@nurani.ir

Deletion

- ▶ درخواست حذف از پایگاه داده شبیه یک پرس و جو می باشد با این تفادت که به جای نمایش خروجی به کاربر ، سطرهای انتخاب شده از پایگاه داده حذف میشود .
- ▶ همه سطرها امکان دارد حذف شوند و لی مقادیر صفات خاصی را نمی توان حذف کرد .
$$r \leftarrow r - E$$
- ▶ که در آن r یک رابطه و E یک پرس و جو می باشد .

<http://www.nurani.ir/> - Info@nurani.ir

Deletion Examples

- ▶ حذف تمامی حسابهای موجود در شعبه "Perryridge"
$$account \leftarrow account - \sigma \text{ branch_name} = \text{"Perryridge"}(account)$$
- حذف همه وامهایی که مقدارشان زیر ۵۰ می باشد
$$loan \leftarrow loan - \sigma \text{ amount} \geq 0 \text{ and } \text{amount} \leq 50(loan)$$

<http://www.nurani.ir/> - Info@nurani.ir

Insertion

▶ درج یک سطر در یک رابطه

▶
$$r \leftarrow r \cup E$$

$account \leftarrow account \cup \{("A-973", "Perryridge", 1200)\}$

$depositor \leftarrow depositor \cup \{("Smith", "A-973")\}$

<http://www.nurani.ir/> - Info@nurani.ir

Updating

▶ مکانیسمی برای تغییر در بخشی از صفات تاپلها

▶
$$r \leftarrow \Pi_{F_1, F_2, \dots, F_i}(r)$$

$account \leftarrow \Pi_{account_number, branch_name, balance * 1.05}(account)$

<http://www.nurani.ir/> - Info@nurani.ir

عملگر گروه بندی (Summarize)

- ▶ SUMMARIZE R By (Attributes) ADD Aggregate-Expression AS Attribute

Sname	Course name	grade
Ali	DataBase	10.5
Ali	C++	17.5
Amir	Database	12
Amir	C++	20

- ▶ SUMMARIZE STUDENT By (Sname) ADD AVG(grade) AS AVERAGE

Sname	AVERAGE
Ali	14
Amir	16

<http://www.nurani.ir/> - Info@nurani.ir

بهینه سازی پرس و جو

- ▶ ۱- گزینش را هر چه ممکن است زودتر انجام دهید
- ▶ ۲- شرطهای ترکیبی را با شرطهای متوالی جایگزین کنید
- ▶ $\sigma_{p1 \wedge p2}(e) \rightarrow \sigma_{p1}(\sigma_{p2}(e))$
- ▶ ۳- پرتو را بعد از گزینش و زودتر از دیگر عملگرها انجام دهید
- ▶ مثال : دانشکده هایی که در تهران واقعند و روسای آنها دکتری ندارند.
- ▶ $\sigma_{city="تهران"} \wedge degree \neq "دکتری" (prof \bowtie clg) \rightarrow$
- ▶ $\sigma_{city="تهران"}(clg) \bowtie \Pi_{degree, pname} (\sigma_{degree \neq "دکتری"}(prof))$

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای

▶ حساب رابطه ای تاپلی

- کاد دو سال پس از چاپ مقاله اصلی خود و ارائه جبر رابطه ای در مقاله دیگری در سال ۱۹۷۲ تئوری خود را به صورت دیگری نیز ارائه داد که حساب رابطه ای تاپلی نامیده شد .

▶ حساب رابطه ای دامنه ای

- پنج سال پس از آن دانشمند دیگری روش سومی به نام حساب رابطه ای دامنه ای ارائه داد که روش قبلی را ساده تر میکند .

▶ تفاوت : در سطح متغیرهایی است که به کار برده می شود.

- سطح فیلدها (دامنه ها) یا در سطح تاپل

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای تاپلی

▶ متغیر تاپلی

- متغیری که تنها مقادیر مجازش تاپلهای رابطه هستند

▶ سور وجودی \exists

▶ سور همگانی \forall

▶ شکل کلی یک پرس و جو در حساب رابطه ای

- $\{t | P(t)\}$

- یعنی مجموعه تاپلهایی که شرط p درباره آنها صادق باشد .

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای تاپلی

► وامهایی که بیش از ۱۰۰۰ دلار می باشند .

$$\{t \mid t \in loan \wedge t[amount] > 1000\}$$

<http://www.nurani.ir/> - Info@nurani.ir

DRC Formulas

$$<, >, =, \leq, \geq, \neq$$

$$\neg p, p \wedge q, p \vee q$$

$$\exists X (p(X))$$

$$\forall X (p(X))$$

حساب رابطه ای دامنه ای

▶ چون متغیرهای آن از جنس دامنه های یک رابطه هستند به آن حساب رابطه ای دامنه ای گفته میشود .

$$\{ \langle x1, x2, ..., xn \rangle | P(x1, x2, ..., xn) \}$$

▶ وامهایی که بیش از ۱۰۰۰ دلار می باشند

$$\{ \langle l, b, a \rangle | \langle l, b, a \rangle \in loan \wedge a > 1000 \}$$

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای دامنه ای

$$\{ \langle l, b, a \rangle | \langle l, b, a \rangle \in loan \wedge a > 1000 \}$$

▶ $\langle l, b, a \rangle \in loan$ باعث می شود تا اطمینان حاصل شود متغیرهای دامنه ای l, b, a در تناظر با تاپلهای جدول $loan$ می باشند.

▶ $\langle l, b, a \rangle$ بایستی تنها متغیر آزاد باشد

◦ متغیری آزاد می باشد که توسط هیچ یک از سور وجودی و سور همگانی بکاربرده نشده است

▶ شماره وامهایی که مبلغ آنها بیش از ۱۰۰۰ دلار است .

$$\{ \langle l \rangle | \exists b, a (\langle l, b, a \rangle \in loan \wedge a > 1000) \}$$

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای دامنه ای

▶ نام و تخصص روسای دانشکده ها

◦ برای حل این مسئله به دو جدول استاد و دانشکده نیاز داریم

- Prof (pname,office,esp,degree,clg#)
- Clg(clg#,clgname,city,pname)

▶ $\{ \langle p,e \rangle \mid \exists c, cn, ci (\langle c, cn, ci, p \rangle \in clg \wedge \exists o, d (\langle p, o, e, d, c \rangle \in prof)) \}$

▶ برای پیوند دو جدول از متغیرهای همنام p,c استفاده شده است، که برای اعمال این پیوند بایستی پرانتزگذاری صحیح انجام شود.

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای دامنه ای

▶ دانشکده های که همه اساتید آنها مدرک دکتری دارند

◦ برای حل این مسئله به دو جدول استاد و دانشکده نیاز داریم

- Prof (pname,office,esp,degree,clg#)
- Clg(clg#,clgname,city,pname)

▶ $\{ \langle cl, cn, ct, p \rangle \mid \langle cl, cn, ct, p \rangle \in clg \wedge \forall p2, o, e, d (\langle p2, o, e, d, cl \rangle \in prof \wedge d = \text{"دکتری"}) \}$

▶ برای پیوند دو جدول از متغیر همنام cl استفاده شده است

▶ توجه شود که در این مثال هدف ایجاد پیوند از طریق pname نبوده و به همین منظور اسامی متغیرها متفاوت انتخاب شده است

◦ در صورتیکه نام یکسان برای pname انتخاب شود، جواب پرس و جو فقط به ازای روسای دانشکده ها خواهد بود.

<http://www.nurani.ir/> - Info@nurani.ir

حساب رابطه ای دامنه ای

$$\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in \text{Borrower} \wedge \exists b (\langle l, b, a \rangle \in \text{loan} \wedge b = \text{"mirdamad"})) \}$$

loan (loan_number, branch_name, amount)

borrower (customer_name, loan_number)

<http://www.nurani.ir/> - Info@nurani.ir

End of Chapter

<http://www.nurani.ir/> - Info@nurani.ir