الجبر العلائقي Relational Algebra.

1. Relational Algebra.
2. Role of Relational Algebra in DBMS.
3. Relational Algebra Operations
4. الاختيارSelect
5. الإسقاطProject
6. الاجتماعunion
7. الفرقDifference
8. الجداء الديكارتيCartesian Produce
9. إعادة التسميةRename

المحاضرة السابعة د كندة أبوقاسم

مقدمة : يعتبر النموذج العلائقي أول نموذج للمعطيات لتطبيقات معالجة المعطيات الواسعة الانتشار , وبسبب وجود نظرية رياضية متكاملة تدعم النموذج العلائقي فإن ذلك يسهل تصميم قواعد المعطيات العلائقية ويتيح المعالجة الفعالة لطلبات المستخدمين .

1- بنية قواعد المعطيات العلائقية

تتألف قاعدة المعطيات العلائقية من جداول لكل منها اسم وحيد مميز . يتألف كل جدول بدوره من مجموعة من الأعمدة وعدد من الأسطر . يمثل كل سطر من الجدول علاقة تربط مجموعة من القيم تشبه العلاقات الرياضية لذلك دعي هذا النموذج العلائقي .

1-1 البنى الأساسية

لدينا الجدول التالي يبين الحسابات account:

balance	account _number	branch_name
500	A-101	Downtown
700	A-215	Mianus

تسمى عناوين الأعمدة واصفات attributes , لكل واصف مجال domain , مثلاً مجال الواصف " اسم الفرع " هو مجموعة كل أسماء الفروع نرمز لهذه المجموعة بالرمز D1 ولمجال الواصف account_number بالرمز D2 والى مجال الواصف balance بالرمز D3 , إن كل سطر من الجدول account يتألف من ثلاثية (V1,V2,V3) حيث V1 هو أسم الفرع وينتمي إلى المجال D1 , و V2 هو رقم الحساب وينتمي إلى المجال D2 و V3 هو الرصيد وينتمي إلى المجال D3. سوف يحوي الجدول account مجموعة جزئية من الجداء D1×D2×D3 وبشكل عام جدول يحوي n واصف يجب أن يكون مجموعة جزئية من الجداء D1×D2×.....×Dn .

نستخدم المتحول t للدلالة على قيمة الواصف في أول سطر مثلاً : t[balance]=500

4-1- لغات الاستعلام

لغة الاستعلام هي اللغة التي يطلب بواسطتها المستخدم معلومات من قاعدة المعطيات وهي عادة من مستوى أعلى من لغات البرمجة القياسية .

ويمكن تصنيف هذه اللغات في نوعين : لغات إجرائية ولغات غير إجرائية

- **اللغات الإجرائية** : يقوم فيها المستثمر بتحديد مجموعة من العمليات التي يجريها النظام على قاعدة المعطيات للوصول إلى المعطيات المطلوبة .
- **اللغات غير الإجرائية** : يصف فيه المستثمر المعلومات المرغوبة دون إعطاء الإجراء المحدد للحصول عليها .

وتقدم معظم قواعد المعطيات كلتا اللغتين .

2- الجبر العلائقي

الجبر العلائقي هو لغة استعلام إجرائية , يتألف من مجموعة من العمليات التي تأخذ علاقة أو اثنتين كدخل وتعطي علاقة جديدة كخرج

العمليات الأساسية في الجبر العلائقي هي :

- الاختيار Select
- الإسقاط Project
- الاجتماع union
- الفرق Difference
- الجداء الديكارتي Cartesian Produce
- إعادة التسمية Rename

بالإضافة إلى العمليات المعرفة ابتدءا من العمليات الأساسية وهي :

- التقاطع Intersection
- الدمج الطبيعي Natural Join
- القسمة Division
- النسب Assignment

سوف نعرض بعض العمليات الأساسية والإضافية في الجبر العلاقاتي موضحين طريقة عملها بأمثلة على قاعدة المعطيات المتعلقة بالمصارف والمعرفة بالمخطط العلاقاتي التالي :

Loan-schema(loan-number , amount , branch-name)

Customer-schema(customer-name , customer-street ,customer-city)

Borrower-schema(customer-name, loan-number)

Employee-schema(employee-name , phone-number)

Loan-officer-schema(banker-name , customer-name , loan-number)

Depositor-schema(customer-name, account-number)

Account-schema(account-number , balance)

Branch-schema(branch-name , branch-city , assets)

1-2- العمليات الأساسية

تسمى العمليات (اختيار , إسقاط , إعادة التسمية) عمليات أحادية لأنها تجري على علاقة واحدة . العمليات الثلاثة الباقية هي ثنائية لأنها تجري على زوج من العلاقات

عملية الاختيار :

تقوم باختيار مجموعة من الحدوديات التي تحقق شرطاً معينا condition من علاقة . يرمز لهذه العلاقة كما يلي :

$$\sigma_{condition}(relation)$$

مثال : لنأخذ مخطط العلاقة التالية :

Loan-schema(loan-number , amount , branch-name)

لاختيار مجموعة الحدوديات القروض للفرع "perryridge" نكتب:

$$\sigma_{branch\text{-}name= \text{“perryridge”}} (loan)$$

يمكن أن تحوي الشروط عمليات مقارنة مثل (=,>=,<,=<, ><,) أو عمليات منطقية مثل and, or , not . مثلاً

لاختيار قروض "perryridge" التي لاتقل عن 1200 نكتب: $\text{branch-name= “perryridge” ^ amount>1200} (loan)$

عملية الإسقاط projection :

وهي عملية وحيدة المعامل تسمح بانتقاء بعض الواصفات من العلاقة . نرمز لهذه العملية بالشكل :

$$\Pi_{\text{selected attributes}}(\text{relation})$$

مثال : للحصول على أرقام القروض ومبالغها دون أن نهتم بأسماء الأفرع نكتب:

$$\Pi_{\text{loan-number , amount}}(\text{loan})$$

فإذا كان جدول علاقة القروض loan كمايلي :

loan-number	branch-name	amount

يكون الجدول الناتج عن عملية الإسقاط :

loan-number	amount

تركيب العمليات العلاقاتية

إن نتيجة العمليات العلاقاتية هي علاقة . هذا ما يسمح لهذه العمليات أن تجتمع لتؤلف عبارات الجبر العلاقاتي . مثلاً لإيجاد أسماء الزبائن الذين يعيشون في مدينة "Harrison" نكتب :

$$\sigma_{\text{customer-name}}(\Pi\ \text{customer-city} = \text{“Harrison”}\ (\text{customer}))$$

أي قمنا بتركيب عمليتين الأولى اختيار الحدوديات التي تحقق الشرط : (المدينة = "Harrison") وإسقاط العلاقة الناتجة من العملية السابقة على العمود customer-name

Selection

- σPR
- Selects the records in relation R that satisfy a predicate P
- Output relation has the same schema as its input

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

$$\sigma_{major='CS'} student$$

Relational algebra

- There are 6 basic operators:
- Selection σ
- Projection π
- Renaming ρ
- Cartesian product ×
- Set difference −
- Union ∪

The operators takes relations as input, and outputs a relation

- Schemas of the input/output schema are fixed
- Operators can be composed

Projection

$\pi_A R$

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

- Retains only the attributes A in the output (i.e., "filters" on columns)
- Schema of the result is exactly A

Projection in relational algebra must eliminate duplicates

- In practice, no for using multi-set relational algebra, unless requested by the user.

$\pi_{major,adm_year} student$

major	adm_year
CS	2021
CE	2020
CS	2020

عملية الجداء الديكارتي Cartesian Product

نرمز لها بالرمز (×) وتسمح بتجميع معلومات من علاقتين ونكتب r1×r2 وبشكل عام إذا كان لدينا العلاقتين r2(R2) , r1(R1) فإن r1×r2 هي علاقة R مخططها العلاقاتي هو تلاصق R1&R2 وتحوي جميع الحدوديات t التي تحقق الشرط :

يوجد t1 من r1 و t2 من r2 بحيث

t1[R]=t1[R1]^t2[R]=t2[R2] R1

مثال : للحصول على جميع الزبائن

الذين اقترضوا من المصرف الفرع

"perryridge" وبالتالي نحتاج إلى المعلومات الموجودة في

العلاقتين loan & borrower وتعطي عملية اختيار الحدوديات المتعلقة بالفرع المطلوب من جداء العلاقتين معلومات عن الزبائن والقروض المأخوذة من الفرع المطلوب ولكن ليست المعلومات المطلوبة وللحصول على المعلومات المطلوبة نكتب :

R=R1× R2

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b2	c1	d1
a2	b2	c2	d2

R1

A	B
a1	b1
a2	b2

R2

C	D
c1	d1
c2	d2

Cartesian product

$R_1 \times R_2$

- Concatenates every pair of tuples $t1 \in R1, t2 \in R2$ into a single tuple $t \in R1 \times R2$
- Output schema is the concatenation of the two input schemas
- There might be naming conflicts, use renaming operator to avoid that

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021

enrollment

sid	semester	cno	grade
100	s22	562	2.0
102	s22	562	2.3
100	f21	560	3.7

$student \times enrollment$

sid	name	login	major	adm_year	sid	semester	cno	grade
100	Alice	alicer34	CS	2021	100	s22	562	2.0
100	Alice	alicer34	CS	2021	102	s22	562	2.3
100	Alice	alicer34	CS	2021	100	f21	560	3.7
101	Bob	bob5	CE	2020	100	s22	562	2.0
More results follows								

عملية الاجتماع Union Operation

نرمز لها U وهناك شرطان لتنفيذ العملية على علاقتين r U s :

1- عدد الواصفات متساوي في العلاقتين .

2- مجال الواصفة رقم i في العلاقة r هو نفسه مجال الواصفة رقم i للعلاقة s .

ويمكن أن تكون العلاقتين r , s ناتجتين عن تعبير في جبر علاقاتي .

مثال : للحصول على جميع الزبائن الذين يتعاملون مع المصرف (المودعين والمقترضين) نحتاج إلى إيجاد مجموعة المودعين وهي معلومات موجودة في علاقة depositor , وإلى إيجاد مجموعة المقترضين وهي موجودة في علاقة borrower ثم إلى إجراء عملية اجتماع بين المجموعتين كمايلي :

$$\Pi_{\text{customer-name}}(\text{depositor}) \cup \Pi_{\text{customer-name}}(\text{borrower})$$

Union

- $R \cup R'$
- Union of two relations of the compatible schema
- Output schema remains the same as inputs

Same number of columns. The ith columns in both relations have the same type for all i

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020

new_students

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
102	Charlie	charlie7	CS	2021
104	Carol	carol20	CS	2021

$students \cup new_students$

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
104	Carol	carol20	CS	2021

Other useful operators

- Set intersection: $R \cap R' = R - (R - R')$

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020

new_students

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
102	Charlie	charlie7	CS	2021
104	Carol	carol20	CS	2021

$students \cap new_students$

sid	name	login	major	adm_year
100	Alic	alicer34	CS	2021

2-4- الدمج الخارجي

لنأخذ العلاقات ذات المخططات العلاقاتية التالية :

Employee (employee-name , street, city)

Ft-works(employee-name , branch-name , salary)

ولنفرض أننا نريد توليد علاقة واحدة تحوي جميع المعلومات حول الموظفين لذا نستخدم عملية الدمج الطبيعي :

employee

employee-name	street	city
A	S1	C1
B	S2	C1
C	S4	C2
D	S5	C3

ft- works

employee-name	branch-name	salary
A	Br1	500
B	Br2	700
D	Br1	1220
F	Br3	2000

ونتيجة الدمج الطبيعي هي :

Employee ⋈ ft-works

employee-name	street	city	branch-name	salary
A	S1	C1	Br1	500
B	S2	C2	Br2	700
D	S5	C3	Br1	1220

Compound operators

- Several useful compound operators
- Join ⋈
- Inner join
- Natural join
- Outer join
- Set intersection ∩
- Division operator /

Inner join

- $R \bowtie_P R' = \sigma_P(R \times R')$
  - Selecting records that satisfy the predicate P from $R \times R'$
- Most common special case is *natural join*

$$R \bowtie R' = \pi_{A(R) \cup A(R')} \sigma_{\forall a \in A(R) \cap A(R'): R.a = R'.a} (R \times R')$$

$A\ R$: attributes of R

- The predicate P is implicitly equality between common attributes of R and R'
- Projecting to all unique attributes of R and R' (only one copy for common attributes)
- Equi-join: P is conjunction of equality predicates
- Useful for denormalization

Natural join

student S

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

enrollment E

sid	semester	cno	grade
100	s22	562	2.0
102	s22	562	2.3
100	f21	560	3.7
101	s21	560	3.3

$$S \bowtie E = \pi_{S.sid,S.name,S.login,S.major,S.adm_{year},E.semester,E.cno,E.grade}\sigma_{S.sid=E.sid}S \times E$$

sid	name	login	major	adm_year	semester	cno	grade
100	Alice	alicer34	CS	2021	s22	562	2.0
100	Alice	alicer34	CS	2021	f21	560	3.7
101	Bob	bob5	CE	2020	s21	560	3.3
102	Charlie	charlie7	CS	2020	s22	562	3.7

نلاحظ ضياع المعلومات غير المشتركة بين الجدولين لتجنب ذلك نستخدم عملية الدمج الخارجي ويوجد ثلاثة أنواع من هذه العملية :

الدمج الخارجي اليساري ويرمز لها بـ

الدمج الخارجي اليميني ويرمز لها بـ

الدمج الخارجي الكامل ويرمز لها بـ

تمثل الأشكال عملية الدمج مع إضافة حدوديات إلى نتيجة عملية الدمج الطبيعي .

ففي عملية الدمج اليساري (أو اليميني) تضاف إلى نتيجة الدمج الطبيعي حدوديات موجودة في العلاقة اليسارية(اليمينية) وغير موجودة في العلاقة اليمينية (اليسارية), بعد وضع null عدم التعيين للواصفات القادمة من العلاقة اليمينية(اليسارية) . أما عملية الدمج الخارجي الكامل فهي اجتماع عمليتي الدمج الخارجي اليميني واليساري

تمثل الجداول التالية عمليات الدمج السابقة :

Employee ⋈ ft-works : عملية الدمج الخارجي الكامل

Employee-name	street	city	Branch-name	salary
A	S1	C1	Br1	500
B	S2	C1	Br2	700
D	S5	C3	Br1	1220
F	Null	Null	Br3	2000
C	S4	C2	Null	Null

جامعة المنارة
MANARA UNIVERSITY

الدمج الخارجي اليساري

Employee ⟕ ft-work

Employee-name	street	city	Branch-name	salary
A	S1	C1	Br1	500
B	S2	C1	Br2	700
D	S5	C3	Br1	1220
C	S4	C2	Null	Null

Employee ⟖ ft-works عملية الدمج الخارجي اليميني

Employee-name	street	city	Branch-name	salary
A	S1	C1	Br1	500
B	S2	C1	Br2	700
D	S5	C3	Br1	1220
F	Null	Null	Br3	2000

Outer join

student S

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

enrollment E

sid	semester	cno	grade
100	s22	562	2.0
102	s22	562	2.3
100	f21	560	3.7
101	s21	560	3.3

$$S \bowtie_{S.sid=E.sid} E$$

S.sid	S.name	S.login	S.major	S.adm_year	E.sid	E.semester	E.cno	E.grade
100	Alice	alicer34	CS	2021	100	s22	562	2.0
100	Alice	alicer34	CS	2021	100	f21	560	3.7
101	Bob	bob5	CE	2020	101	s21	560	3.3
102	Charlie	charlie7	CS	2020	102	s22	562	2.3
103	David	davel	CS	2020	NULL	NULL	NULL	NULL