

Relational Database Design

د. مادلين عبود

محتوى مجاني غير مخصص للبيع التجاري



Data Base

01/06/2023

17

8

S.P

1,190

نظري

S.P

1,190

8

S.P

17

01/06/2023

RB Informatcs ;

قواعد معطيات 1

Relational DataBase Design

يتطلب تصميم قاعدة معطيات علاقاتية إيجاد مجموعة من مخططات علاقات جيدة وقادرة على تمثيل جميع المعلومات المطلوبة بأقل تكرار ممكن، قد يؤدي التصميم إلى تكرار المعلومات أو عدم القدرة على تمثيل بعض المعلومات

تتلخص أهداف التصميم بثلاث نقاط:

- إنقاص ظاهرة تكرار المعطيات ما أمكن
- التوثيق من تمثيل العلاقات الموجودة بين الواصفات
- سهولة اختبار التعديلات لاختبار تحقق شروط التكامل المعرفة على قاعدة المعطيات

الأشكال النظامية

1. الشكل النظامي الأول 'FNF' First Normal Form

■ يعد المجال ذري إذا كانت عناصره تعتبر وحدات غير قابلة للتجزئة

أمثلة على مجالات غير ذرية:

- » مجموعة من الأسماء والواصفات المركبة
- » Identification numbers مثل CS101 التي يمكن تقسيمها إلى أجزاء
- يعد المخطط العلاقتي R موجوداً بالشكل FNF إذا كانت مجالات جميع واصفاته ذرية
- إن القيم غير الذرية تعقد آلية التخزين وتشجع تخزين البيانات المكررة

مثال:

» مجموعة من الحسابات المخزنة مع كل عميل ومجموعة من المالكين المخزنة مع كل حساب عندها نفرض أن جميع العلاقات موجودة بالشكل FNF

■ **Atomicity** هي خاصية كيفية استخدام عناصر مجال

مثلاً: عادة ما نعتبر الـ String غير قابلة للتجزئة

لنفترض أن الطلاب يتم إعطاؤهم roll number وهي سلاسل من النموذج CS0012 أو EE1127

❖ إذا تم استخراج أول حرفين من السلسلة للعثور على رقم department يكون مجال الـ roll number غير ذرياً كما أن القيام بذلك هو فكرة سيئة لأنه سيؤدي إلى ترميز Encoding المعلومات في البرنامج المطبق

Application program بدلا من قاعدة المعطيات database

كما نقول عن علاقة أنها من الشكل النظامي الأول FNF إذا كان تقاطع (سطر , عمود) يحوي قيمة واحدة.

مثال: لتكن العلاقة Products:

Products

factory	Products
F_1	$\{P_2, P_4, P_5\}$
F_2	$\{P_1, P_2, P_3\}$

هذه العلاقة ليست من الشكل النظامي الأول

وتصبح من الشكل النظامي الأول

Products

factory	products
F_1	P_2
F_1	P_4
F_1	P_5
F_2	P_1
F_2	P_2
F_2	P_3

في حال كان محتواها كالتالي:

2. Pitfalls في تصميم قاعدة معطيات علاقاتية

كما ذكرنا سابقا أن التصميم الجيد يتطلب إيجاد مجموعة جيدة من مخططات العلاقة

كما أن التصميم السيئ قد يؤدي إلى:

❖ تكرار المعلومات

❖ عدم القدرة على تمثيل بعض المعلومات

كما أن أهداف التصميم:

❖ تجنب البيانات الزائدة (المكررة).

❖ ضمان تمثيل العلاقات بين الواصفات.

❖ تسهيل التحقق من تحديات انتهاك قيود وسلامة قاعدة المعطيات



Example: Consider the relation schema :

Lending-schema = (branch-name, branch-city, assets, customer-name, loan-number, amount)

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

التكرار:

- ❖ بيانات assets , branch-city , branch-name تتكرر لكل قرض يقدمه الفرع.
- ❖ تهدر مساحة.
- ❖ يعقد التحديث وإمكانية عدم تناسق قيم الـ assets.

Null value:

- ❖ لا يمكن تخزين المعلومات حول الفرع branch في حالة عدم وجود قروض loans
- ❖ يمكن عندها استخدام Null value ولكن من الصعب التعامل معها

3. التقسيم decomposition

نقسم علاقة (R) lending-schema إلى:

- ❖ (R_1) Branch – schema(branch – name , branch – city, assets)
- ❖ (R_2) Loan – info – schema(customer – name , loan – number, branch – name, amount)

يجب أن تظهر جميع واصفات العلاقة الأساسية (R) في التقسيم (R_1, R_2) :

$$R = R_1 \cup R_2$$

وهي تجزئة محافظة على المعطيات lossless-join decomposition لجميع العلاقات r الممكنة على المخطط R:

$$r = \prod_{R_1}(r) \bowtie \prod_{R_2}(r)$$

★ إن تحليل R إلى R_1, R_2 هو loss-join إذا وفقط إذا كان أحد التبعية التالية على الأقل في F^+ :

$$R_1 \cap R_2 \rightarrow R_1$$

$$R_1 \cap R_2 \rightarrow R_2$$

مثال على تقسيم غير محافظ على المعطيات:

Decomposition of $R = (A, B)$

$$R_1 = (A), R_2 = (B)$$

r

A	B
α	1
α	2
β	1

A
α
β

B
1
2

 $\prod_A(r)$ $\prod_B(r)$

$$\prod_A(r) \bowtie \prod_B(r)$$

A	B
α	1
α	2
β	1
β	2



4. نظرية theory

- نقرر فيما إذا كانت Relation R في شكل جيد Good form
- في حال أن Relation R ليست في شكل جيد Good form نقوم بعمل decompose لـ مجموعة من العلاقات مثل: $\{R_1, R_2, \dots, R_n\}$
- كل علاقة في شكل جيد Good form
- كما أن التقسيم هو تقسيم محافظ على المعطيات lossless-join decomposition
- إن هذه النظرية معتمدة على:
 - » التبعية الوظيفية: functional dependencies
 - » تبعيات متعددة: Multivalued dependencies

5. التبعية الوظيفية functional dependencies

- تكون القيود على مجموعة العلاقات القانونية
- تتطلب أن تكون قيمة من مجموعة معينة من الواصفات فريدة من نوعها لقيمة لمجموعة أخرى من الواصفات
- التبعية الوظيفية هي تعميم مفهوم المفتاح

❖ Let R be a relation Schema $\alpha \subseteq R$ and $\beta \subseteq R$
 The functional dependencies as $\alpha \rightarrow \beta$
 holds on R

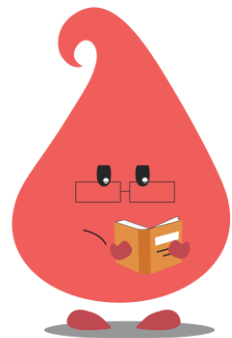
إذا وفقط إذا من أجل أي علاقة قانونية $r(R)$ أيًا يكن الـ two tuples t_1 and t_2 توافق على الوصفة α عندها توافق على الوصفة β

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

Example consider $r(A, B)$ with the following instance of r

1	4
1	5
3	7

On this instance, $A \rightarrow B$ *does not hold*
 but $B \rightarrow A$ *does hold*



- ❖ يعد K مفتاحاً أعلى (super key) لمخطط العلاقة R إذا وفقط إذا: $K \rightarrow R$
- ❖ يعد K مفتاحاً مرشحاً (candidate key) لمخطط العلاقة R إذا وفقط إذا:
 - $K \rightarrow R$ and
 - for no $\alpha \subset K, \alpha \rightarrow R$

❖ تسمح لنا التبعية الوظيفية التصريح عن قيود لا نستطيع التصريح عنها باستخدام المفاتيح العليا super Key

Consider the schema:

Loan- info-schema= (customer-name, loan-number, branch-name, amount)

We expect this set of functional dependencies to hold:

$Loan - number \rightarrow amount$

$Loan - number \rightarrow branch - name$

But would not expect the following to hold:

$loan - number \rightarrow customer - name$

6. استخدام التبعية الوظيفية:

تستخدم التبعية الوظيفية :-

■ اختبار العلاقات لمعرفة ما إذا كانت قانونية بموجب مجموعة معينة من التبعية الوظيفية:

» إذا كانت العلاقة R قانونية بموجب مجموعة F من التبعية الوظيفية، نقول أن R يرضي (satisfies) F

■ تحديد القيود على مجموعة العلاقات القانونية:

» نقول أن F يحمل (holds on) R إذا كانت جميع العلاقات القانونية على R تلبى مجموعة التبعية الوظيفية F

Note: A specific instance of a relation schema may **satisfy** a functional dependency even if the functional dependency **does not hold on** all legal instances

Example: a specific instance of loan-schema may, by chance, **satisfy**

$loan - number \rightarrow customer - name$

■ تكون التبعية الوظيفية عديمة الأهمية trivial إذا كانت راضية satisfied عن جميع حالات العلاقة

(a Functional dependency is **trivial** if it is **satisfied** by all instances of a relation)

مثال:

$Customer - name, loan - number \rightarrow customer - name$

$Customer - name \rightarrow customer - name$

In general $\alpha \rightarrow \beta$ is **trivial** if $\beta \subseteq \alpha$

7. Closure of a set of functional dependencies

■ بالنظر إلى مجموعة F من التبعية الوظيفية هناك بعض التبعية الوظيفية الأخرى التي تنطوي عليها منطقياً بواسطة F

EX: If $A \rightarrow B$ and $B \rightarrow C$, then we can infer that $A \rightarrow C$

■ إن مجموعة التبعية الوظيفية المضمنة بشكل منطقي من قبل F هي ما يسمى

بإغلاق F (The closure of F)

■ نشير إلى إغلاق F بـ F^+

يمكننا إيجاد جميع F^+ عن طريق تطبيق Armstrong's Axioms

- (reflexivity) if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
 (augmentation) if $\alpha \rightarrow \beta$, then $\gamma\alpha \rightarrow \gamma\beta$
 (transitivity) if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$

هذه القواعد هي Complete and sound

complete	Sound
Generate all functional dependencies that hold	Generate only functional dependencies that actually hold

Example:

▪ $R = \{A, B, C, G, H, I\}$

▪ $F = \{A \rightarrow B$
 $A \rightarrow C$
 $CG \rightarrow H$
 $CG \rightarrow I$
 $B \rightarrow H\}$

▪ Some members of F^+ :

$\rightarrow A \rightarrow H$

by transitivity from $A \rightarrow B$ and $B \rightarrow H$

$\rightarrow AG \rightarrow I$

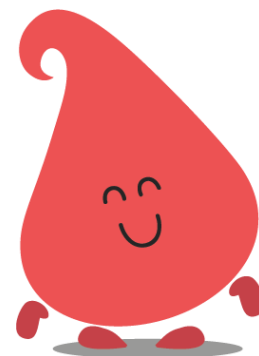
by augmenting $A \rightarrow C$ with G , to get $AG \rightarrow CG$ and then transitivity with $CG \rightarrow I$

$\rightarrow CG \rightarrow HI$

from $CG \rightarrow H$ and $CG \rightarrow I$:

"union rule" can be inferred from:

- ★ Definition of functional dependencies or
- ★ Augmentation of $CG \rightarrow I$ to infer $CG \rightarrow CGI$,
 augmentation of $CG \rightarrow H$ to infer $CGI \rightarrow HI$, and then transitivity



حساب F^+

$F^+ = F$

لحساب إغلاق مجموعة من التبعية الوظيفية F^+ :

نكرّر:



- لكل تبعية وظيفية f في F^+
 - ❖ تطبيق قواعد reflexivity و augmentation على f
 - ❖ نضيف التبعية الوظيفية الناتجة إلى F^+
 - لكل زوج من التبعية الوظيفية f_1, f_2 في F^+
 - ❖ إذا كان من الممكن تجميع f_1, f_2 باستخدام transitivity
 - ثم نضيف الاعتماد الوظيفي الناتج إلى F^+
- حتى لا يتغير F^+ .

يمكننا زيادة تبسيط الحساب اليدوي لـ F^+ باستخدام القواعد الإضافية التالية:

- if $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds (union)
- if $\alpha \rightarrow \beta\gamma$ holds then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds (decomposition)
- if $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds (pseudotransitivity)

يمكننا استنتاج هذه القواعد عن طريق Armstrong's Axiom

8. Closure of attribute sets

- بالنظر إلى مجموعة الواصفات α :
- » يكون تحديد إغلاق α تحت F (يشار إليه بواسطة α^+) كمجموعة من الواصفات التي يتم تحديدها وظيفياً بواسطة α تحت F .
- » $\alpha \rightarrow \beta$ is in $F^+ \iff \beta \subseteq \alpha^+$

{Algorithm to compute α^+ , the closure of α under F }

```

result :=  $\alpha$ ;
while (changes to result) do
  for each  $\beta \rightarrow \gamma$  in  $F$  do
    begin
      if  $\beta \subseteq \text{result}$  then result := result  $\cup \gamma$ 
    end
  
```

شرح الخوارزمية

- $R = \{A, B, C, G, H, I\}$
- $F = \{A \rightarrow B$
 $A \rightarrow C$
 $CG \rightarrow H$
 $CG \rightarrow I$
 $B \rightarrow H\}$
- $(AG)^+$
 1. result=AG
 2. result=ABCG ($A \rightarrow C$ and $A \rightarrow B$)
 3. result=ABCGH ($CG \rightarrow H$ and $CG \subseteq AGBC$)
 4. result= ABCGHI ($CG \rightarrow I$ and $CG \subseteq AGBCH$)



- هل AG هو مفتاح مرشح؟!
 1. هل AG هو مفتاح أعلى؟!
 (1) هل $AG \rightarrow R$ ؟!
 2. هل أي جزء من AG هو مفتاح أعلى؟!
 (1) هل $A^+ \rightarrow R$ ؟
 (2) هل $G^+ \rightarrow R$ ؟

9. استخدامات Attribute closure:

هناك عدة استخدامات لخوارزمية إغلاق الواصفة

- اختبار المفتاح الأعلى Super Key:
 - » لاختبار فيما إذا كان α مفتاحاً أعلى نقوم بحساب α^+ ونتحقق ما إذا كان α^+ يحتوي على جميع سمات R
- اختبار التبعية الوظيفية:
 - » للتحقق مما إذا كان التبعية الوظيفية $\alpha \rightarrow \beta$ holds (أو بعبارة أخرى أنه في F^+) فقط نتحقق مما إذا كان $\beta \subseteq \alpha^+$
 - » نحسب α^+ باستخدام إغلاق الواصفة ومن ثم نتحقق مما إذا كان يحتوي على β
 - » اختبار رخيص وبسيط ومفيد للغاية
- حساب closure of f:
 - » لكل $\gamma \subseteq R$ نوجد الإغلاق γ^+ ولكل $S \subseteq \gamma^+$ نخرج الاعتماد الوظيفي $\gamma \rightarrow S$

10. Canonical cover

- قد يكون لمجموعات التبعية الوظيفية تبعيات زائدة (مكررة) يمكن استنتاجها من الآخرين
 - » مثال: $A \rightarrow C$ is redundant in $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
 - » قد تكون أجزاء من التبعية الوظيفية زائدة عن الحاجة (مكررة)

E.g on RHS : $\{A \rightarrow B, B \rightarrow C, A \rightarrow CD\}$ can be simplified to $\{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$

E.g on LHS : $\{A \rightarrow B, B \rightarrow C, AC \rightarrow D\}$ can be simplified to $\{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$

بشكل حدسي, $F \dashv$ canonical cover هو مجموعة " الحد الأدنى " من التبعية الوظيفية التي تعادل (تساوي) F مع عدم وجود تبعيات زائدة (مكررة) أو مع وجود أجزاء زائدة من التبعية.

إن $F \dashv$ canonical cover هو مجموعة من التبعية F_c

■ يتضمن F منطقياً جميع التبعية في F_c

■ ويتضمن F_c منطقياً جميع التبعية في F_1

■ ولا يوجد تبعية وظيفية في F_c تحتوي على واصفة غريبة Extraneous attribute

■ كل جانب يساري left side من التبعية الوظيفية في F_c فريد

لحساب $F \dashv$ canonical cover

نكرر

■ نستخدم union rule لاستبدال أي تبعية في F

$\alpha_1 \rightarrow \beta_1$ and $\alpha \rightarrow \beta_1$ with $\alpha_1 \rightarrow \beta_1 \beta_2$

■ نبحث عن التبعية الوظيفية $\alpha \rightarrow \beta$ مع واصفة غريبة extraneous attribute إما في α أو في β

■ إذا تم العثور على واصفة غريبة نحذفها من $\alpha \rightarrow \beta$ حتى لا يتغير F

ملاحظة:

قد تصبح union rule قابلة للتطبيق بعد حذف بعض الواصفات الغريبة لذلك يجب إعادة تطبيقها

مثال على حساب canonical cover:

$R = (A, B, C)$

$F = \{A \rightarrow BC$

$B \rightarrow C$

$A \rightarrow B$

$AB \rightarrow C\}$

نجمع بين $A \rightarrow BC$ و $A \rightarrow B$ إلى $A \rightarrow BC$ أصبحت المجموعة هي $\{A \rightarrow BC, B \rightarrow C, AB \rightarrow C\}$

■ A هي واصفة غريبة في $AB \rightarrow C$ لأن $B \rightarrow C$ منطقياً تتضمن $AB \rightarrow C$

أصبحت المجموعة هي : $\{A \rightarrow BC, B \rightarrow C\}$

■ C هي واصفة غريبة في $A \rightarrow BC$ لأن $A \rightarrow BC$ منطقياً مضمنة ضمن $B \rightarrow C$ و $A \rightarrow B$

إذا يكون $F \dashv$ canonical cover : $A \rightarrow B$ و $B \rightarrow C$



11. واصفات غريبة Extraneous Attributes:

بالنظر إلى مجموعة من التبعية الوظيفية والاعتماد الوظيفي

$$\alpha \rightarrow \beta \text{ in } F$$

» الوصفة A غريبة في α إذا كانت $A \in \alpha$ و F تتضمن بشكل منطقي:

$$(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$$

» الوصفة A غريبة في β إذا كانت $A \in \beta$ ومجموعة التبعية الوظيفية تتضمن بشكل منطقي F:

$$(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$$

التضمنين بالاتجاه المعاكس هو عديم الأهمية trivial في كل حالة من الحالات المذكورة أعلاه لأن التبعية الوظيفية "الأقوى" تتضمن دوماً واحداً أضعف

Example: Given $F = \{A \rightarrow C, AB \rightarrow C\}$ B is extraneous in $AB \rightarrow C$

because $A \rightarrow C$ logical implies $AB \rightarrow C$

Example: Given $F = \{A \rightarrow C, AB \rightarrow CD\}$ C is extraneous in $AB \rightarrow CD$

since $A \rightarrow C$ can be inferred even after deleting C

12. اختبار ما إذا كانت واصفة غريبة

Consider a set of F of functional dependencies and the functional dependency $\alpha \rightarrow \beta$ in F

■ To test if attribute $A \in \alpha$ is extraneous in α

1) Compute $(A - \{\alpha\})^+$ using the dependencies in F

2) Check that $(A - \{\alpha\})^+$ contains α ; if it does, A is extraneous

■ To test if attribute $A \in \beta$ is extraneous in β

1) Compute α^+ using only the dependencies in $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$

2) Check that α^+ contains A; if it does, A is extraneous

13. Normalization باستخدام تبعية وظيفية

عندما نحلل مخطط العلاقة R مع مجموعة من التبعية الوظيفية F إلى R_1, R_2, \dots, R_n نريد:

■ تحليل loss less-join

خلاف ذلك قد يؤدي التحليل إلى فقدان المعلومات.

■ لا يوجد تكرار No redundancy

يجب أن تكون العلاقات R_i إما في Boyce codd Normal form أو في Third Normal form

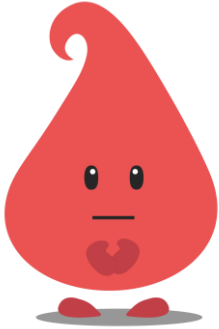
■ الحفاظ على التبعية Depending preservation

لتكن F_i مجموعة التبعية F^+ التي تتضمن فقط واصفات في R_i ويفضل أن يكون التحليل حفاظ على التبعية:

$$(F_1 \cup F_2 \cup \dots \cup F_n)^+ = F^+$$

وخلاف ذلك قد يتطلب التحقق من التحديثات لانتهاك التبعية الوظيفية computing joins وهي مكلفة.

Example:



- $R = \{A, B, C\}$
 - $F = \{A \rightarrow B, B \rightarrow C\}$
 - $R_1 = (A, B), R_2 = (B, C)$
 - loss less – join decomposition
 - $R_1 \cap R_2 = \{B\}$ and $B \rightarrow BC$
 - Depending preserving
 - $R_1 = (A, B), R_2 = (A, C)$
 - loss less – join decomposition
 - $R_1 \cap R_2 = \{A\}$ and $A \rightarrow AB$
 - Not Depending preserving
- (can't check $B \rightarrow C$ without computing $R_1 \bowtie R_2$)

أهداف الـ normalization

- نقرر ما إذا كانت علاقة معينة R في شكل جيد.
- في حال كانت العلاقة R ليست في شكل جيد:
 - » نحللها إلى مجموعة من العلاقات $\{R_1, R_2, \dots, R_n\}$
 - » كل علاقة في شكل جيد
 - » التحليل هو loss less- join decomposition
- تعتمد هذه النظرية على:
 - » تبعية وظيفية
 - » تبعية متعددة القيم

Second Normal Form .14

يكون مخطط العلاقة R في 2NF فيما يتعلق بمجموعة F من التبعية الوظيفية إذا كانت جميع الواصفات غير الأولية تعتمد بالكامل على جميع المفاتيح

Pilot	Pilot_Id	Name	Birth_date	Flight_hours	Address
	1715	John	18-11-1954	200	London
	1825	jack	01-04-1960	150	Amsterdam
	1214	Paul	12-04-1945	300	Paris

Air_Craft	AC_Id	Type	Capacity
	502	AB320	150
	120	B727	140
	211	B747	30
	304	DC10	75

Flights	Flight_Nb	Flight_date	Dept_city	Dept_hr	Destination	Arrival_hr	Gate_Nb	Pilot_Id
	RB405	12-04-1998	Damascus	15:30	Paris	19:30	10	1715
	RB405	14-04-1998	Damascus	16:30	Paris	20:30	10	1825
	AF601	02-05-1998	London	08:30	Paris	09:30	22	1214
	AF601	03-05-1998	London	09:30	Paris	10:30	22	1214
	AF601	04-05-1998	London	07:30	Paris	08:30	22	1214

Boyce-codd Normal Form .15

يكون مخطط العلاقة R موجود في BCNF فيما يتعلق بمجموعة F من التبعية الوظيفية إلا إذا كان لجميع التبعية الوظيفية F^+ من الشكل:

$$\alpha \boxed{?} \rightarrow \beta \text{ where } \alpha \subseteq R \text{ and } \beta \subseteq R$$

تحمل على الأقل واحدة مما يأتي:

$$\alpha \boxed{?} \rightarrow \beta \text{ is trivial (i. e, } \beta \subseteq \alpha)$$

α is a superkey for R

Example:



- $R = \{A, B, C\}$
 $F = \{A \rightarrow B, B \rightarrow C\}$
 $\text{Key} = \{A\}$
- R is **not** in BCNF
- Decomposition $R_1 = (A, B), R_2 = (B, C)$
 R_1 and R_2 in BCNF
 less loss – join decomposition
 dependency preserving

Testing for BCNF .16

للتحقق مما إذا كانت التبعية غير المهمة non-trivial $\alpha \boxed{?} \rightarrow \beta$ تسبب انتهاكا لـ BCNF:

- حساب α^+ (the attribute closure of α)
- نتحقق من أنه يتضمن جميع واصفات R أي أنه super key لـ R.

للتحقق مما إذا كان مخطط العلاقة مع مجموعة معينة من التبعية الوظيفية F في $BCNF$ فإنه يكفي التحقق من التبعية الوظيفية فقط في المجموعة F المحددة لانتهاك $BCNF$ بدلا من التحقق من جميع التبعية في F^+ .
يمكننا أن نظهر أنه إذا لم يسبب أي من التبعية في F انتهاكاً لـ $BCNF$ فإن أيًا من التبعية في F^+ لن يؤدي إلى انتهاك لـ $BCNF$.
 ومع ذلك فإن استخدام F فقط غير صحيح عند اختبار علاقة في تحليل R .

Example:

Consider $R(A, B, C, D)$ with $F = \{A \rightarrow B, B \rightarrow C\}$

- Decompose R into $R_1(A, B)$ and $R_2(A, C, D)$
- Neither of dependencies in F contain only attributes from (A, C, D) so we might be misled into thinking R_2 satisfies $BCNF$.
- In fact, dependency $A \rightarrow C$ in F^+ shows R_2 is not in $BCNF$.

17. BCNF decomposition algorithm

result := {R};
 done := false;
 compute F^+ ;

while (not done) do

if (there is a schem R_i in result that is not in $BCNF$
 then begin

let $\alpha \boxed{?} \rightarrow \beta$ be a nontreivial functional
 dependency that holds on R_i

such that $\alpha \boxed{?} \rightarrow R_i$ is not in F^+ ,
 and $\alpha \cap \beta = \emptyset$;

result := (result - R_i) \cup ($R_i - \beta$) \cup (α, β);

end

else done := true;



■ Note: each R_i is in $BCNF$, and decomposition is loss – join

مثال على BCNF decomposition:

$R = (\text{branch} - \text{name}, \text{branch} - \text{city}, \text{assets},$
 $\text{customer} - \text{name}, \text{loan} - \text{number}, \text{amount})$
 $F = \{\text{branch} - \text{name} \rightarrow \text{assets} \text{ branch} - \text{city}$
 $\text{loan} - \text{number} \rightarrow \text{amount} \text{ branch} - \text{name}\}$
 $\text{key} = \{\text{loan} - \text{number}, \text{customer} - \text{name}\}$

$R_1 = (\text{branch} - \text{name}, \text{branch} - \text{city}, \text{assets})$
 $R_2 = (\text{branch} - \text{name}, \text{customer} - \text{name}, \text{loan} - \text{number}, \text{amount})$
 $R_3 = (\text{branch} - \text{name}, \text{loan} - \text{number}, \text{amount})$
 $R_4 = (\text{customer} - \text{name}, \text{loan} - \text{number})$

بعد التحليل الأخير

 R_1, R_3, R_4

18. BCNF and Dependency Preservation

ليس من الممكن دائماً الحصول على تحليل BCNF وهذا هو الحفاظ على التبعية.

$$\begin{aligned} \diamond R &= (J, K, L) \\ F &= \{ JK \rightarrow L \\ &\quad L \rightarrow K \} \end{aligned}$$

هناك مفتاحين Candidate هما JK و JL.

R ليس في صيغة (شكل) BCNF.

إنَّ أيَّ تحليل لـ R سيفشل في الحفاظ على التبعية $JK \rightarrow L$.



19. Third Normal Form: Motivation

هناك بعض الحالات حيث:

BCNF يحافظ على التبعية.

من المهم الفحص الفعال لانتهاك FD على التحديثات.

فيكون الحل: نعرف شكل طبيعي أضعف يدعى Third Normal Form بحيث:

يسمح ببعض التكرار (مع مشاكل ناتجة، سنرى أمثلة لاحقاً).

ولكن يمكن فحص FDs على كل العلاقات الفردية بدون حساب أي انضمام.

هناك دائماً lossless-join, dependency-preserving decomposition into 3NF.

يكون مخطط العلاقة R في 3NF إذا كان يحقق من أجل $\alpha \rightarrow \beta$ in F^+ تضمين واحدة على الأقل مما يلي:

- $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
- α is a super key for R
- Each attribute A in $\beta - \alpha$ is contained in a candidate key for R

Note: Each attribute may be in a different candidate key

- إذا كانت العلاقة في الشكل BCNF فستكون أيضاً في الشكل 3NF (حيث في BCNF يجب أن يحمل أحد الشرطين الأوليين أعلاه).
- الشرط الثالث هو الحد الأدنى من الاسترخاء من BCNF لضمان الحفاظ على التبعية.

Example:

- $R = \{ J, K, L \}$
- $F = \{ JK \rightarrow L, L \rightarrow K \}$
- Two candidate keys: JK and JL
- R is in 3NF

$$JK \rightarrow L$$

$$L \rightarrow K$$

JK is a superkey

K is contained in a candidate key



- BCNF decomposition has (JL) and (LK)
- Testing for $JK \rightarrow L$ requires a join

■ هناك بعض التكرار في هذا المخطط.

Equivalent to example in book:

Banker – Schema = (branch – name, customer – name, banker – name)

banker – name \rightarrow branch – name

branch – name, customer – name \rightarrow banker – name

Testing for 3NF .20

- التحسين (optimization): نحتاج إلى التحقق من FDs فقط في F ولا نحتاج إلى التحقق من جميع FDs في F^+ .
- نستخدم attribute closure للتحقق من أجل كل تبعية $\alpha \rightarrow \beta$ إذا كانت α هي super key.
- إذا لم تكن α هي super key فيجب علينا التحقق مما إذا كانت كل واصفة في β محتواة في المفتاح المرشح لـ R (Candidate key):
- A. هذا الاختبار أكثر تكلفة لأنه ينطوي على العثور على المفاتيح المرشحة.
- B. إن اختبار 3NF هو NP-Hard.
- C. يمكن إجراء التحليل إلى الشكل الطبيعي الثالث 3NF في وقت متعدد الحدود.

3NF Decomposition Algorithm .21

Let F_c be a canonical cover for F ;

$i := 0$;

for each functional dependency $\alpha \rightarrow \beta$ in F_c do


```

if none of the schemas  $R_j, 1 \leq j \leq i$  contains  $\alpha \beta$ 
  then begin
     $i := i + 1$ ;
     $R_i := \alpha \beta$ ;
  end
if none of the schemas  $R_j, 1 \leq j \leq i$  contains a candidate key for R
  then begin
     $i := i + 1$ ;
     $R_i :=$  any candidate key for R;
  end
return (R1, R2, ..., Ri)

```

**Example:**

مخطط العلاقة

Banker – info – schema = (branch – name, customer – name, banker – name, office – number)

إن التبعية الوظيفية لمخطط العلاقة هذا:

banker – name \rightarrow branch – name, office – number
customer – name, branch – name \rightarrow banker – name

المفتاح هو:

$\{ \text{customer – name, branch – name} \}$

22. Applying 3NF to Banker-info-schema

■ إن حلقة for في الخوارزمية تجعلنا ندرج المخططات التالية في تحليلنا:

Banker – office – schema = (banker – name, branch – name, office – number)
Bank – schema = (customer – name, branch – name, banker – name)

■ بما أن Banker-schema يحتوي على مفتاح مرشح ل Banker-info-schema بهذا نكون قد انتهينا من عملية التحليل.

23. Comparison of BCNF and 3NF

- من الممكن دائماً تحليل العلاقة إلى علاقات في 3NF: ويكون التحليل lossless ويحافظ على التبعية.
- من الممكن دائماً تحليل العلاقة إلى علاقات في BCNF: ويكون التحليل lossless وقد لا يكون محافظاً على التبعية.

مثال على مشاكل بسبب التكرار في 3NF:

$$R = (J, K, L)$$

$$F = \{ JK \rightarrow L, L \rightarrow K \}$$



J	K	L
J1	K1	L1
J2	K1	L1
J3	K1	L1
null	K2	L2

■ مشاكل المخطط الذي يكون في الشكل 3NF وليس BCNF:

A. تكرار في المعلومات (e.g., the relationship L1, K1).

B. الحاجة لاستعمال قيم Null

(e.g., to represent the relationship L2, K2 where there is no corresponding value for J)

24. Design Goals

■ إنّ هدف تصميم قاعدة البيانات العلائقية هو:

- BCNF
- Lossless-join
- Dependency Preservation

■ إذا لم تتمكن من تحقيق ذلك فيمكن تواجد إحدى ما يلي:

- Lack of dependency preservation
- Redundancy due to use of 3NF

■ إنّ SQL لا توفر طريقة مباشرة لتحديد التبعية الوظيفية بخلاف مفاتيح super keys.

■ يمكن تحديد FDs باستخدام التأكيدات لكنها مكلفة للاختبار.

■ حتى لو كان لدينا تحليل محافظ للتبعية فإنه باستخدام SQL لن نتمكن من اختبار التبعية الوظيفية بكفاءة للذي لا يكون جانبه الأيسر مفتاحاً.

لا يصل الناس إلى حديقة النجاح دون أن يمروا بمحطات التعب والفشل واليأس، وصاحب الإرادة القوية لا يطيّل الوقوف في هذه المحطات.

واعلموا بأننا سنبلغ حلمنا ولو بعد حين فنحن بحار عزم إن أردنا.

والى هنا نصل وإياكم إلى ختام محاضرات القسم النظري من مادة قواعد المعطيات 1 راجين أن نكون قد قدمنا كل الفائدة مع تمنياتنا لكم بالتوفيق في امتحاناتكم القادمة.

سامحونا إن أخطئنا فجلّ من لا يخطئ وإلى اللقاء في القريب العاجل مع مادة قواعد المعطيات 2 في السنة الرابعة. كل الحب من فريق مادة قواعد المعطيات 1 ولا تنسونا من صالح دعائكم.

#database1 #we_carry_your_o2 #viva_rbcS



نهاية المقرر
RB Informatics;