تطبيقات رياضية

المحاضرة السابعة

١٣/٤/٢٠١٥

3) الربط الخارجي: (⟗) Outer Join

- لاحظنا في الربط الطبيعي أنه في بعض الأحيان لا يكون لجميع الأسطر في أحد الجداول ما يُربط معه في الجدول الآخر.
- أي أن قيم حقول الربط في أحد الجدولين ليس لها نسخة مقابلة تحقق شرط الربط الطبيعي في الجدول المقابل.
- قد نحتاج في بعض الأحيان إلى إظهار تلك الأسطر، ويمكن الوصول إلى هذه النتيجة باستخدام الربط الخارجي.
- يسمى الربط الخارجي من اليمين ⟖ أو من اليسار ⟕ حسب توضع العلاقة المراد استرجاع كامل أسطرها بالنسبة لعملية الربط.

فمثلاً: الربط من اليسار هو الربط الذي تظهر في نتيجته الأسطر في العلاقة الأولى والتي لا يوجد لها ما يربط معها في العلاقة الثانية، بالإضافة إلى باقي الأسطر التي تحقق شرط الربط.

- يمكن أن نعرّف الربط الخارجي الكامل (التام) full outer join حينما نريد استرجاع جميع أسطر كلا العلاقتين، كما في المثال التالي:

Employee

N°	Name	office N°
23	toto	17
25	Loulou	15
35	jojo	22
23	toto	14

⟗

office

office N°	Buildim
14	A_1
42	B_2
17	A_1
62	C_7

نتيجة الربط الخارجي للعلاقتين السابقتين هي الجدول :

Employee office

N°	Name	office N°	Building
23	toto	17	A_1
25	Loulou	15	⊥
35	jojo	22	⊥
23	toto	14	A_1
⊥	⊥	42	B_2
⊥	⊥	62	C_7

* الربط الجزئي: Semi join

يعطينا الربط الجزئي جميع السجلات الموجودة في العلاقة الأولى (بما فيها السجلات التي ليس لها مقابل) دون أن تظهر حقول العلاقة الثانية والتي ليس لها مقابل في العلاقة الأولى.

وبالتالي عملية الربط الجزئي هي عملية ربط خارجي من اليسار أو من اليمين وهي عملية غير تبديلية.

مثلاً: إذا قمنا بالربط الخارجي من اليسار بين العلاقتين المذكورتين في المثال الأخير سينتج لدينا نفس الجدول ولكن بالأسطر الأربعة الأولى فقط.

أما إذا قمنا بالربط الخارجي من اليمين بينهما فسينتج نفس الجدول ماعدا السطرين الثاني والثالث.

تصميم قاعدة البيانات

التنظيم: Normalization

نقصد بالتنظيم (أو التسوية) تصميم جداول قاعدة البيانات بحيث نتحكم بتكرار البيانات ونتجنب حالات الشذوذ anomaly والتي يمكن أن تنتج عن عمليات الإضافة والحذف والتعديل على تلك البيانات.

إشكاليات التنظيم السيء لقاعدة البيانات:

توجد مشاكل أساسية ينبغي علينا تجنبها خلال عملية تصميم قاعدة البيانات.

من هذه المشاكل:

1) التكرار غير المفيد في تخزين البيانات. redundancy anomaly
2) الإضافة الخاطئة Insertion anomaly
3) الحذف الخاطئ Deletion anomaly
4) التعديل الخاطئ Update anomaly

في الواقع، إن استخدام طريقة منظمة لتصميم جداول قاعدة البيانات يقلل من هذه المشاكل إلى حد كبير.

سنتناول بعض الأمثلة التي توضح الشذوذ الذي يمكن أن يؤثر على تصميم قاعدة البيانات.

* من خلال الجدول المعطى، يمكن أن نناقش كل من المشاكل المذكورة:

1- **التكرار غير المفيد:**

- إن المعلومات المتعلقة بالراتب لكل موظف مكررة عدة مرات بلا فائدة.
- إن المعلومات المتعلقة بالميزانية لكل مشروع مكررة عدة مرات بلا فائدة.
- بعض قيم الحقل Name مكررة ولا يمكن بالتالي اعتباره مفتاحاً للعلاقة
- كذلك بعض قيم الحقل Project مكررة ولا يمكن اعتباره مفتاحاً للعلاقة.
- إن مفتاح هذه العلاقة هو Name, Project

/ /

اسم الموظف	اسم المشروع	العمل	راتب الموظف	ميزانية المشروع
Name	Project	Job	Salary	Budget
Mhd	DB	Assistant	7000	10000
Samer	UML	Designer	1000	20 000
Samer	XML	Designer	1000	60 000
ALi	XML	Manager	3000	60 000
ALi	UML	Consulter	3000	20 000
ALi	DB	Consulter	3000	10 000
Sami	Web	Employee	4000	32 000
Majd	DB	Manager	500	10 000
Majd	XML	Designer	500	60 000
Yaser	XML	Designer	1500	60 000
Yaser	DB	Designer	1500	10 000

2- الإضافة الخاطئة:

- بما أن المفتاح هو Name, Project فإنه لا يمكن إدراج موظف بدون تحديد المشروع الذي يعمل فيه لأنه لا يجوز أن يكون المفتاح عديم القيمة (null)
- كذلك لا يمكن إدراج مشروع بدون تحديد على الأقل اسم الموظف الذي يعمل فيه

3- الحذف الخاطئ:

- بفرض أن موظفاً ما قد أنهى عمله في الشركة، أو أن موظفاً ما قد تم إعفاؤه من الخدمة في مشروع، وأن هذا الموظف هو الموظف الوحيد في المشروع (مثل sami)
- إن حذف سجل ذاك الموظف سيؤدي إلى فقدان معلومات عن المشروع الذي يعمل فيه

4- التحديث الخاطئ:

- إن التكرار في تخزين البيانات سيؤدي إلى تكرار في تحديث تلك البيانات، فمثلاً: في الجدول السابق هناك تكرار لتخزين الراتب لكل موظف، وإذا ما أردنا تحديث راتب موظف ما فإن ذلك يتطلب تحديثه في كل سجلات الجدول، وهذا يقلل من فعالية قاعدة البيانات.
- وبنفس الطريقة، فإن تحديث ميزانية مشروع ما سيؤدي إلى نفس المشكلة.

التبعية الوظيفية: Functional Dependency FD

(الاعتمادية الوظيفية) . (الارتباط الوظيفي)

- يُقال عن الواصفة X إنها تحدد Determine الواصفة Y
أو أن الواصفة Y تعتمد اعتماداً وظيفياً (ترتبط ارتباطاً وظيفياً) على X
إذا كانت كل قيمة من X تحدد قيمة واحدة فقط من Y
- ويُرمز لذلك بالتبعية الوظيفية $X \longrightarrow Y$
- ويمكن التعبير عن ذلك رياضياً كما يلي:

إذا كان هناك سطران t_1, t_2 في الجدول، وكان $t_1.X = t_2.X$

فإن $t_1.Y = t_2.Y$ وذلك: $\forall t_1, t_2 \in R$

مجموعة أسطر الجدول

- إذا كان لدينا الارتباط الوظيفي: $X \longrightarrow Y$

Source عندئذٍ ندعو X مصدر الارتباط الوظيفي

goal وندعو Y هدف الارتباط الوظيفي

مثال: لنأخذ العلاقة التالية:

Emp N°	Emp Name	Emp Adress
112	Sami	E A_1
652	Majd	E A_2
43	Hadi	E A_3
231	ALi	E A_4
43	Hadi	E A_5

فلاحظ من الجدول أن رقم الموظف يحدّد اسمه، لذا يُقال إن Emp Name يتبع وظيفياً لـ Emp N°

أو يُقال إن Emp N° يحدّد Emp Name

ونرمز لذلك بـ : Emp N° ⟶ Emp Name

- نقول إن هناك ارتباطاً وظيفياً بين Emp N° , Emp Name لأنه لا يمكن إدراج الزوج <112, Samer> بدون حذف الزوج <112, Sami>

وبالمقابل لا يوجد ارتباط وظيفي بين اسم الموظف Emp Name وعنوانه Emp Adress

ملاحظة: إن مفهوم التبعية الوظيفية العلائقية هو نفسه مفهوم التابع الرياضي حيث لا يمكن لعنصر في المنطلق أن يرتبط بعنصرين مختلفين في المستقر.

تعاريف:

* نقول عن الواصفة Y إنها تعتمد وظيفياً بشكل تام Full FD على واصفة مركبة إذا كانت تلك الواصفة المركبة تحدد Y، وكانت Y لا تعتمد وظيفياً على جزء منها.

مثال: إذا كان A,B,C ⟶ D ارتباط وظيفي تام، فإن:

$A \not\longrightarrow D$, $B \not\longrightarrow D$, $C \not\longrightarrow D$

$A,B \not\longrightarrow D$, $A,C \not\longrightarrow D$, $B,C \not\longrightarrow D$

* نقول عن الارتباط الوظيفي $A \longrightarrow B$ إنه ارتباط وظيفي مباشر إذا تعذر وجود واصفة C بحيث $A \longrightarrow C \longrightarrow B$.

التبعية الوظيفية والمفتاح:

انطلاقاً من مفهوم التبعية الوظيفية وتعريف مفتاح العلاقة الذي هو واصفة أو أكثر تحدد صف واحد داخل الجدول (بدون تكرار)

يمكن أن نستنتج أن المفتاح هو الواصفة (أو مجموعة الواصفات) التي نستطيع من خلالها تحديد كل الواصفات الأخرى الموجودة في الجدول.

نهاية المحاضرة السابعة.