التطبيع NORMALIZATION

- تحديد المشكلات التي تعالجها الأشكال والمستويات المختلفة من التطبيع
- كيفية القيام بتطبيع قاعدة بيانات
- تحديد مستوى التطبيع الأفضل لقاعدة البيانات التي يقوم بتصميمها.

• النموذج الطبيعي الأول NF.1

النموذج الثاني 2NF All of the non-key fields depend on all of the key fields

النموذج الثالث 3NF It contains no transitive dependencies

نموذج التطبيع BCNF (Boyce-Codd) Every determinant is a candidate key

المحاضرة التاسعة + العاشرة د كندة أبوقاسم

What is Normalization
ما هو التطبيع

قد تكون قاعدة البيانات العلائقية عرضة لبعض أنواع المشكلات منها:

- قد تحتوي على الكثير من البيانات المكررة، وهذا لا يسبب خسارة في المساحة فقط وإنما يجعل تعديل جميع هذه البيانات المكررة عمالا شاقا مستهلكا للزمن.
- قد تسبب **ربطا خاطئا لقطعتي بيانات غير مرتبطتين** وبالتالي لا يمكن حذف إحداهما من دون حذف الأخرى.
- قد تتطلب **قطعة من البيانات غير موجودة** من أجل تمثيل قطعة أخرى موجودة.
- قد تؤدي إلى **محدودية في عدد القيم** التي تستطيع إدخالها من أجل ما يجب أن يكون قطعة بيانات متعددة القيم.

بحسب اصطلاحات قواعد البيانات فهذا الأمر يدعى حالات شاذة anomalies.

التطبيع Normalization هي عملية لإعادة تنظيم قاعدة البيانات لجعلها بشكل قياسي (طبيعي) وهذا يمنع مثل هذه الحالات الشاذة.

الصيغ المعيارية NORMALIZATION

الصيغ المعيارية Forms Normal

Normalization is the process of a clearance the database of inappropriate reputation of data by depending on the inference rules and functiona ldependency, although the process of developing the database design in the normal forms is a key building block for a state of art design of the database.

الصيغة المعيارية: هي عملية تخليص قاعدة البيانات من التكرار غير المناسب للبيانات بالاعتماد على قوانين الاستنتاج والاعتمادية الوظيفية، وان عملية وضع تصميم قاعدة البيانات في الصيغة المعيارية يشكل لبنة أساسية في عملية التصميم الجيد لقاعدة البيانات.

مشاكل التصميم السيئ لقاعدة البيانات:

بفرض أن شركة ما تضم مجموعة من مسؤولي المبيعات يعملون في عدة مواقع، تريد الإدارة تخزين البيانات المتعلقة بموظفيها وبيانات مستودعاتها في قاعدة معطيات. المحاولة الأولى لبناء القاعدة نتج عنها الجدول التالي

Id	Name	Address	Title	Store	Store_Address	Store_Phone
Sr1	Jane	E1	Sales Rep	A1	S1	011-1234567
Sr2	Fred	E2	Sales Rep	A1	S1	011-1234567
Sr3	Ed	E3	Manager	A1	S1	011-1234567
Sr4	Ann	E4	Sales Rep	A2	S2	011-7891234
Sr5	Jone	E5	Sales Rep	A2	S2	011-7891234
Sr6	Smith	E6	Manager	A2	S2	011-7891234

لاحظ المشاكل التالية:

1. بفرض أننا نريد إدخال بيانات مستودع جديد قبل توظيف أحد فيه، هذا يتطلب إدخال قيم Null في بيانات الموظف ومن ضمنها الحقل المفتاح (Id) مما يخترق شرط تكامل البيانات. هذا ما يتدعى بإشكالية الادخال . (Insertion anomaly)

2- بفرض أننا حذفنا تسجيل الموظف الأخير في أحد المستودعات، سيؤدي ذلك إلى فقدان بيانات مستودع موجود في الشركة. هذا ما يدعى بإشكالية الحذف (anomaly Deletion)

3 إذا أردنا تغيير رقم هاتف أحد المستودعات، عندها يجب تغيير رقم الهاتف في كل تسجيلات موظفي ذلك المستودع، فإذا تم تغيير جزء من هذه التسجيلات دون جزء آخر سيؤدي ذلك إلى بيانات متناقضة في قاعدة البيانات، وهذا ما يدعى بإشكالية التعديل .(anomaly Update)

يتم حل المشاكل السابقة من خلال تجزئ الجدول السابق إلى جدولين، أحدهما يحمل بيانات الموظفين ويستورد مفتاح الجدول الثاني الذي يحمل بيانات المستودعات:

Id	Name	Address	Title	Store
Sr1	Jane	E1	Sales Rep	A1
Sr2	Fred	E2	Sales Rep	A1
Sr3	Ed	E3	Manager	A1
Sr4	Ann	E4	Sales Rep	A2
Sr5	Jone	E5	Sales Rep	A2
Sr6	Smith	E6	Manager	A2

Store	Store Address	Store phone
A1	S1	011-1234567
A2	S2	011-7891234

- **حل مشكلة الادخال:** عند إدخال بيانات موظف جديد، يتم تستجيل مفتاح المستودع الذي يعمل فيه، أما بقية بيانات المستودع فهي مخزنة في جدول مستقل وبالتالي لن تظهر بيانات متناقضة في قاعدة البيانات بشأن المستودعات، وإذا أردنا إدخال بيانات مستودع جديد فيمكن إدخالها بصرف النظر عمن يعمل في المستودع، مما يجنبنا إشكالية الادخال.

- **حل مشكلة الحذف:** عند حذف الموظف الأخير في مستودع، تبقى بيانات المستودع في جدول مستقل.

- **حل مشكلة التعديل:** عند تعديل رقم هاتف المستودع، يتم التعديل على تسجيله واحدة في جدول المستودعات.

المشاكل التي يمكن ان تتعرض لها قواعد البيانات

Employee_department

Empno	Ename	Job	Salary	Deptno	Dname	Loc
10	Smer	Clerk	300	1	Accounting	Baghdad
20	Ali	Manager	500	1	Accounting	Baghdad
30	Khalid	Salesman	400	2	Sales	Mosul
40	Saeed	Salesman	450	2	Sales	Mosul
50	Salem	Clerk	350	3	Operation	Basrah

1- مشاكل تكرار البيانات Anomalies Data

نلاحظ في الجدول التالي أن معلومات الموظف والقسم الذي يعمل فيه موجودة في جدول واحد،
ونتيجة لذلك فأن تكرار بعض البيانات مثل اسم وموقع القسم في كل سجل سوف يؤدي إلى عدة مشاكل.

2- **مشكلة الإضافة : Anomaly Insertion** وهي عدم القدرة على إضافة قيمة بيانية في أحد السجلات حيث أننا لا نستطيع أن نضيف قسما السجلات؛ لعدم وجود قيمة بيانية أخرى في مكان آخر. جديدا لأن المفتاح الرئيسي للجدول هو رقم الموظف.

3 .**مشكلة التعديل : Anomaly Updating** عند التعديل على قيمة بيانية في أحد المواقع، يجب تغيير جميع القيم المشابهة لها في المواقع الأخرى.
حيث إذا قمنا بتعديل موقع ( ) loc القسم فلابد من إجراء عملية التعديل لجميع الموظفين ستؤدي هذه العملية إلي عدم في هذا القسم؛ وتوافقية البيانات، أي أن نفس رقم القسم ولكن أكثر من موقع،
وكذلك إذا تمت عملية تغيير موقع القسم عند أي موظف عن طريق الخطأ، فإنه لو قمنا بعملية استرجاع لجميع الموظفين الذين يعملون في هذا القسم فإن هذا الموظف لن يظهر بين الموظفين.

4 .**مشكلة الحذف** : Anomaly Deletionوهي مشكلة ترابط البيانات التي تحذف بأكملها عند حذف قيمة بيانية متصلة بها.
حيث إن القسم 3 يحتوي على موظف واحد فقط، ولو قمنا بحذف هذا الموظف فإن معلومات القسم 3 سوف تختفي من الجدول.

5- الاعتمادية لوظيفية Functional Dependency (FD)

وهي اعتماد قيمة إحدى صفات الكيان على قيمة صفة (صفات) أخر ى، ويرمز لها بالرمز ⟶

حيث من خلال تحديد هذه الاعتمادية سوف نستطيع أن نحدد المكان الذي يجب أن توضع فيه الصفة، وهذا بالتالي يؤدي إلى وضع البيانات في المكان الصحيح والتخلص من عملية تكرار البيانات.

- **مثال1 :** $A \longrightarrow B$
- يعني أن B تعتمد اعتماداً وظيفياً على A ، ونستطيع أن نقول هنا أن قيمة A تحدد قيمة B. وأن A ترجع لنا قيمة واحدة فقط لـ B.
- ويمكن ايضاً أن نقول أننا نستطيع أن نصل إلى البيانات في العمود B بمعرفتنا للبيانات في العمود A.
- **مثال2:** لكل موظف اسم واحد فقط ولكل موظف قسم واحد يعمل فيه إذاً:

FD1 : Empno ⟶ Ename
FD2 : Empno ⟶ Deptno

- أي يمكن معرفة اسم الموظف من رقمه وكذلك القسم.
- ويمكن أن نعيد كتابة هذه الاعتمادية على الشكل التالي:

FD1 : Empno ⟶ Ename , Deptno

- قواعد الاستنتاج **Inference Rules**
- Inference Rules is a set of rules that are used in the process of identifying the functional dependency.
- هي عبارة عن مجموعة من القواعد تستخدم في عملية تحديد الاعتمادية الوظيفية وتتلخص هذه القواعد كما يلي:
- **1- قاعدة الانعكاسية (Reflexive Rule)** : إذا كانت y جزء من x (y محتواه في x) فإن x تحدد y

$$X \geq Y : X \longrightarrow Y$$

- **مثال:**

$$\mathbf{Empno \longrightarrow Deptno}$$

2- قاعدة الزيادة أو الإضافة (Augmentation Rule) : إذا كانت x تحدد y فإن إضافة z إلي x تعني أنه بالإمكان إضافة z إلى y

$$\{ X \longrightarrow Y \} \models XZ \longrightarrow YZ$$

- **مثال:**

$$\mathbf{Empno \longrightarrow Deptno}$$
$$\mathbf{Empno,Dname \longrightarrow Deptno,Dname}$$

3- قاعدة التعدي (Transitive Rule) : إذا كانت x تحدد y وكانت y تحدد z فإن x تحدد

$$\{ X \longrightarrow Y , Y \longrightarrow Z \} \models X \longrightarrow Z$$

- مثال:

$$Book \longrightarrow Author$$
$$Author \longrightarrow Auth_Addr$$
$$Book \longrightarrow Auth_Addr$$

4- قاعدة الاتحاد (Union Rule): إذا كانت x تحدد y و x تحدد z إذاً x تحدد yz

$$\{ X \longrightarrow Y , X \longrightarrow Z \} \models X \longrightarrow YZ$$

- مثال:

$$Empno \longrightarrow Ename$$
$$Empno \longrightarrow Gender$$
$$Empno \longrightarrow Ename,Gender$$

5- قاعدة التقسيم (Decomposition Rule) : وهي عكس قاعدة الاتحاد، إذا كانت x تحدد yz إذاً x تحدد y و تحدد z

$$\{ X \longrightarrow YZ \} \models X \longrightarrow Y , X \longrightarrow Z$$

- مثال:

$$Empno \longrightarrow Ename,Gender$$
$$Empno \longrightarrow Ename$$
$$Empno \longrightarrow Gender$$

هناك خمس مستويات مختلفة للتطبيع، يتضمن كل مستوى المستويات التي قبله. بالإضافة لتحقيقها بعض المزايا الإضافية الخاصة بالمستوى الجديد.

هذا يعني أن قاعدة بيانات بمستوى معين من التطبيع تملك بشكل تلقائي حسنات المستويات الأدنى.

المستويات المختلفة من التطبيع مرتبة من الأضعف إلى الأقوى هي :

- النموذج الطبيعي الأول1FN.
- النموذج الطبيعي الثاني2NF.
- النموذج الطبيعي الثالث N.3F
- نموذج Codd-Boyce الطبيعي BCNF(نوع محسن من المستوى الثالث).
- النموذج الطبيعي الرابع NF.4
- النموذج الطبيعي الخامس NF.5
- النموذج الطبيعي مجال/مفتاح DKNF

تمتلك قاعدة بيانات من النموذج DKNFقوة مذهلة في الحماية من الحالات الشاذة كما تتضمن جميع أشكال القوة الأخرى لقواعد البيانات.

مستويات التطبيع (التنظيم) :

التنظيم هو عملية ترتيب وتوزيع جداول قاعدة المعطيات العلائقية، للتقليل من تكرار المعطيات وتقليص حجم التخزين المطلوب ولحل إشكاليات الادخال والحذف والتعديل مع المحافظة على تكامل البيانات.

مستوى التنظيم	عملية التنظيم
1NF	حذف المجموعات المكررة. أنشئ جدولاً مستقلاً لكل مجموعة واصفات مترابطة، وعيّن مفتاحاً أساسياً لكل جدول.
2NF	حذف البيانات المكررة. إذا وجدت أي واصفة تعتمد فقط على جزء من مفتاح متعدد-القيمة، انقل الواصفة إلى جدول مستقل.
3NF	حذف الأعمدة غير المرتبطة على مفتاح. إذا وجد مجموعة من الواصفات لا تساهم في وصف المفتاح، انقلها إلى جدول مستقل.
BCNF	Boyce-Codd Form. إذا لم تكن العلاقة بين المفاتيح المرشحة علاقة واضحة، افصل هذه المفاتيح في جداول مختلفة.
4NF	عزل العلاقات العديدة المستقلة عن بعضها. لا وجود لجدول يحوي علاقات 1:n أو n:m غير المترابطة بشكل مباشر.
5NF	عزل العلاقات العديدة المترابطة دلالياً. وجود قيود عمليّة على البيانات تستدعي الفصل المنطقي للعلاقات عديد لعديد.

تنظيم قاعدة البيانات

يكون النموذج الطبيعي الأول بشكل **أساسي محققا لمعظم القيود** عند بناء قاعدة البيانات بأحدي برامج قاعدة البيانات. بالإضافة إلى زوج من الخصائص الإضافية من أجل جعل قاعدة البيانات أكثر فائدة، إلا أن أغلب هذه القواعد أساسية

المزايا الرسمية للنموذج الطبيعي الأول هي :

1. كل عمود يجب أن يكون له اسم مميز (غير مكرر).
2. ترتيب الأعمدة والصفوف لا أهمية له.
3. كل عمود يجب أن يكون له نوع بيانات وحيد.
4. لا يمكن لصفين أن يحتويا قيما متطابقة **(تكرار)**.
5. كل عمود يجب أن يحتوي قيمة وحيدة غير **(متعددة Multi value)**.
6. لا يمكن للأعمدة أن تتضمن مجموعات مكررة **(nested relations)**

القاعدتان الأولى والثانية تحصل عليهما مجانا لدى استخدامك لبرنامج قواعد بيانات علائقي مثل Access، SQL server أو MySQL. فهذه البرامج تتطلب منك أن تعطي الأعمدة أسماء مختلفة، كما أنها لا تكترث لترتيب الأعمدة والصفوف، على الرغم من أنك عندما تبحث عن بيانات فإنك قد ترغب بتحديد الترتيب التي ترد فيه من أجل تحقيق الانسجام.

1الشكل النظامي الأول (1NF):

يقال عن جدول في قاعدة البيانات أنه من الشكل النظامي الأول إذا كان **تقاطع كل سطر وعمود فيه (أي الخلايا)، يتضمن قيمة وحيدة غير قابلة للتجزئة.**

الخطوات التي يتكون منها الشكل النظامي الأول:

1. **إزالة المجموعات المكررة**: تبدأ هذه المرحلة من خلال عرض المعطيات ضمن جدول بحيث تتكون كل خلية فيه من قيمة وحيدة ولا توجد هناك أية مجموعات مكررة. يتم إزالة المجموعات المكررة من خلال إزالة كافه القيم الفارغة NULL والتأكد من أن كافة الواصفات تحتوي على قيم مناسبة.
2. **تعريف المفتاح الرئيسي.**
3. **تعريف كافة التبعيات.**

القاعدة 3 تعني أنه لا يمكن لصفين أن يتضمنا قيما من أنواع مختلفة لنفس العمود.
على سبيل المثال، حقل القيمة في جدول ما لا يمكن أن تحتوي على سلسلة محرفيه في أحد الصفوف، وقيمة رقمية في صف آخر وعملة في صف ثالث. وهذه غالبا ما تكون خاصية نحصل عليها مجانا لأن برامج قواعد البيانات لن تسمح لك بذلك، " الحقل يجب أن يحتوي على نوع واحد

القاعدة 4 مهمة لأنه إذا كان بإمكان صفين احتواء قيم متطابقة، فكيف سيتم التمييز بينهما ؟.
السبب الوحيد الذي يسمح لك بكسر هذه القاعدة هو في حال كونك لا ترغب بالتمييز بينهما، على سبيل المثال، بفرض أنك تسجل طلبية لشراء قلم رصاص، بعض الأوراق وغيرها من القرطاسية، ثم استنتجت أنك بحاجة لقلم رصاص إضافي فإنك تضيفه إلى نهاية الطلبية

القاعدة 5 قد تكون أكثر قاعدة ترغب باختراقها. في بعض الأحيان قد يحتوي عنصر البيانات على صفة تتطلب عدة قيم **متعددة**. تخزين عدة قيم في حقل واحد يحد من فائدة ذلك الحقل
.الحل هو بتجزئة القيم المتعددة ونقلها إلى جدول جديد وربط تلك السجلات مع الجدول الأول من خلال المفتاح الرئيسي.

القاعدة 6 تعني أننا لا نستطيع أن يكون لدينا عدة أعمدة تحتوي قيما لا يمكن التمييز بينها.

تمرين : تنظيم البيانات بالنموذج الطبيعي الأول

City	City	Connections
Beirut	Oman	1
Algeria	Damascus	Tunisia, Tripoli, Cairo
Algeria	Damascus	Tunisia, Tripoli, Cairo
Beirut	Oman	1
Algeria	Damascus	Tunisia, Tripoli, Cairo

عينة بيانات من جدول رحلات الطيران.

يحتوي الجدول التالي على معلومات حول رحلات الطيران، حيث يحتوي على جزء من معلومات رحلتين من Beirut إلى Omanوجزء من معلومات ثالث رحلات من Algeriaإلى .Damascus العمودان الأول والثاني يحددان مدينتي الانطلاق والوصول ويحوي العمود الأخير المدن المرتبطة (إن وجدت) أو عدد الارتباطات.

المهمة المطلوبة هي تحويل هذا النموذج الرديء إلى النموذج الطبيعي الأول.

الحل : ان الجدول ليس من الشكل القياسي الأول لأنه في الحقل الواحد عدة قيم

1 **تحقق من أن كل عمود يملك اسما فريدا** هذا الجدول يحوي عمودين يحملان الاسم city، لحل هذه المشكلة، قم بتسمية العمودين كما يلي : StartCityو DestinationCity. تصبح وضعية جدول البيانات بعد إنجاز الخطوة الأولى من الشكل التالي:

StartCity	DestinationCity	Connections
Beirut	Oman	1
Algeria	Damascus	Tunisia, Tripoli, Cairo
Algeria	Damascus	Tunisia, Tripoli, Cairo
Beirut	Oman	1
Algeria	Damascus	Tunisia, Tripoli, Cairo

جدول رحلات الطيران بعد تنفيذ الخطوة الأولى

2 **تحقق أن ترتيب الأسطر والأعمدة لا يؤثر**. إذا كان ترتيب الأسطر يؤثر، قم بإضافة عمود لتسجيل المعلومات الخاصة بموقعه. في مثالنا هذا قم بإنشاء عمود خاص بالأولوية priority وخزن به بشكل صريح أولوية رغبات المسافرين

تصبح وضعية جدول البيانات بعد إنجاز الخطوة الثانية من الشكل التالي:

StartCity	DestinationCity	Connections	Priority
Beirut	Oman	1	**2**
Algeria	Damascus	Tunisia, Tripoli, Cairo	**1**
Algeria	Damascus	Tunisia, Tripoli, Cairo	**1**
Beirut	Oman	1	**2**
Algeria	Damascus	Tunisia, Tripoli, Cairo	**1**

تحقق من أن كل عمود يحوي نوع بيانات وحيد، في حال كون أحد الأعمدة يحوي بيانات من أكثر من نوع بيانات ، قم بتجزئته إلى عدة أعمدة، عمود لكل نوع بيانات. في حالتنا هذه، قم بذكر المدن المرتبطة ولا تسجل عدد المدن.

جدول رحلات الطيران بعد تنفيذ الخطوة الثالثة a

StartCity	DestinationCity	Connections	Priority	Cust_ID
Beirut	Oman	Baghdad	2	**4637**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**12878**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**2871**
Beirut	Oman	Baghdad	2	**9287**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**9287**

3 تحقق من أنه لا يوجد صفان يمكن أن يحتويا قيما متطابقة. إذا احتوى صفان قيما متطابقة قم بإضافة حقل للتمييز بينهما. في حالتنا هذه قم بإضافة عمود CustomerId بحيث يمكن أن تحدد الزبائن بشكل منفصل

StartCity	DestinationCity	Connections	Priority	Cust_ID
Beirut	Oman	Baghdad	2	**4637**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**12878**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**2871**
Beirut	Oman	Baghdad	2	**9287**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	**9287**

4 تحقق من أن كل عمود يحتوي قيمة وحيدة. إذا احتوى عمود ما عدة قيم، قم بتجزئتها إلى جدول جديد. قم بإنشاء جدول جديد يدعى Connections وقم بنقل معلومات المدن المرتبطة إليه. ولربط هذه السجلات إلى صفوفها الأصلية في هذا الجدول، قم بإضافة أعمدة تعبر عن أعمدة المفتاح الأساسي في الجدول الأصلي .(CustomerId،Date)

5.تحقق من أنه ال توجد عدة أعمدة تحوي مجموعات مكررة من المعلومات. في مثالنا هنا، يجب التفكير في الحقلين Connectionsو Dateوالقرار فيما كانا يحويان قيما متمايزة

StartCity	DestinationCity	Connections	Priority	Cust_ID	Date
Beirut	Oman	Baghdad	2	4637	**4/1/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	12878	**6/21/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	2871	**6/21/14**
Beirut	Oman	Baghdad	2	9287	**4/1/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	9287	**6/21/14**

لاحظ أنه تمت إضافة حقلين على التوالي للتمييز بين السجلات، CustomerId و Date

StartCity	DestinationCity	Connections	Priority	Cust_ID	Date
Beirut	Oman	Baghdad	2	4637	**4/1/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	12878	**6/21/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	2871	**6/21/14**
Beirut	Oman	Baghdad	2	9287	**4/1/14**
Algeria	Damascus	Tunisia, Tripoli, Cairo	1	9287	**6/21/14**

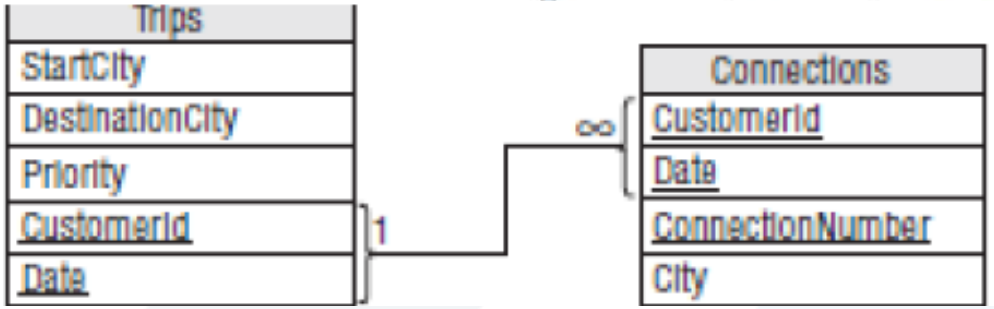
جدول رحلات الطيران بعد تنفيذ الخطوة الخامسة.

StartCity	Des_City	Priority	CustID	Date
Beirut	Oman	2	4637	4/1/14
Algeria	Damascus	1	12878	6/21/14
Algeria	Damascus	1	2871	6/21/14
Beirut	Oman	2	9287	4/1/14
Algeria	Damascus	1	9287	6/21/14

CustID	Date	conNum	City
4637	4/1/14	1	Baghdad
12878	6/21/14	1	Tunisia
12878	6/21/14	2	Tripoli
12878	6/21/14	3	Cairo
2871	6/21/14	1	Tunisia
2871	6/21/14	2	Tripoli
2871	6/21/14	3	Cairo
9287	4/1/14	1	Baghdad
9287	6/21/14	1	Tunisia
9287	6/21/14	2	Tripoli
9287	6/21/14	3	Cairo

بالتالي يصبح النموذج العلائقي لقاعدة البيانات الخاصة بتسجيل بيانات رحلات الطيران كما يلي:

لنموذج العلائقي النهائي لقاعدة البيانات



2 الشكل النظامي الثاني Second norm (2FN)

نقول عن جدول أنه من النموذج الطبيعي الثاني إذا :

1. كان من النموذج الطبيعي الأول.
2. جميع الحقول غير المفتاحية مرتبطة بجميع الحقول المفتاحية (لا تملك ارتباطات جزئية Partial Dependency)

. All of the non-key fields depend on all of the key fields

2 الشكل النظامي الثاني :(2FN)

يقال عن جدول أنه من الشكل النظامي الثاني إذا حقق ما يلي:

- هو من الشكل النظامي الأول.
- كل الواصفات التي لا تشكل جزءاً من المفتاح الأساسي، تعتمد وظيفياً وبشكل كلي على المفتاح الأساسي

الخطوات التي يتكون منها الشكل النظامي الثاني:

1. تعريف كافة مكونات المفاتيح: ينبغي أولا كتابة كل جزء من المفتاح الرئيسي (المركب) على حدة ثم كتابة المفتاح ككل على سطر آخر. و بعد ذلك سيتم اعتبار كل سطر مما سبق مفتاح رئيسي في جدول خاص.
2. تعريف الواصفات ذات التبعيات

التبعية الوظيفية :(functional dependency)

تعتمد الواصفة Bعلى الواصفة Aإذا كانت كل قيمة لـ A تقابلها قيمة وحيدة لـ B، فمثلًا رقم الزبون يحدد اسمه، وكل قيمة لـ c_id تقابلهما قيمة وحيدة ل c_name لذلك يقال أن c_name تتبع وظيفيا لـ c_id أو يقال أن c_id تحد دc_name. (Determine)

B تعتمد وظيفياً بشكل كلي على مفتاح مركب (functional fully dependency)، إذا كان المفتاح المركب يحدد B، و B لاتعتمد وظيفيا على جزء منه

.

يبدأ وضع الجداول في الشكل النظامي الثاني بتحديد جميع علاقات التبعية بين الواصفات، وممن ثم تجزيء الجداول (decomposition).
بتطبيق ذلك على مثالنا نجد العلاقات التالية آخذين بعين الاعتبار التبعية الكاملة والجزئية

C_id	P_num	C_name	P_address	R_start	R_end	Rent	O_num	Owner
01 01	Pr3 Pr22	Jane Jane	A B	1-1-96 2-1-98	12-1-98 3-30-00	785 1200	Po23 Po44	Jones Jan
02	Pr17	Fred	C	2-1-88	1-11-90	1000	Po32	Jill
03 03	Pr32 Pr22	Ed Ed	D B	6-1-90 4-1-00	3-1-95 null	950 1200	Po32 Po44	Jill Jan

1. C_id + p_num → r_start, r_end.
2. C_id → c_name.
3. P_num→ p_address, rent, owner_num, owner.
4. Owner_num→ owner.
5. C_id + r_start→ p_num, p_address, r_end, r_end, rent, owner_num, owner.
6. P_num + r_start→ c_id, c_name, r_end.

يبدأ تطبيق معيار NF2 بتجزيء الجدول انطلاقا من علاقة التبعية الكاملة (a)

Rental

C_id	P_num	R_start	R_end
01	Pr3	1-1-96	12-1-98
01	Pr22	2-1-98	3-30-00
02	Pr17	2-1-88	1-11-90
03	Pr32	6-1-90	3-1-95
03	Pr22	4-1-00	Null

من ثم نشكل الجداول اللازمة لحقول علاقات التبعية الجزئية:

Customer

C_id	C_name
01	Jane
02	Fred
03	Ed

Property

P_num	P_address	rent	Owner_num	Owner
Pr3	A	785	Po23	Jones
Pr22	B	1200	Po44	Jan
Pr17	C	1000	Po32	Jill
Pr32	D	950	Po32	Jill

- **الصيغة المعيارية الثانية (2NF) Second Normal Form**

- The table will be in 2NF:

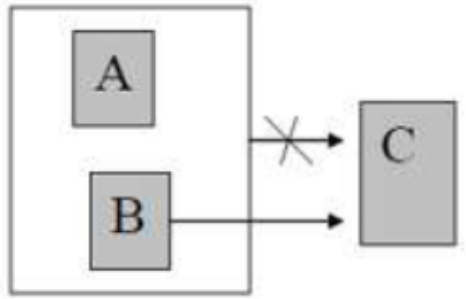
1. The table in the 1NF.
2. The table does not contain a partial dependency.

- إن الجدول يكون في الصيغة المعيارية الثانية إذا:

1. كان في الصيغة المعيارية الأولى.
2. لا يحتوي الجدول على اعتمادية جزئية.

- **الاعتمادية الجزئية Partial Functional Dependency:**

- Partial Functional Dependency is some of the columns (attributes) functionally dependent on the part of the primary key (composite).
- هي أن تعتمد بعض الأعمدة (الصفات) اعتماداً وظيفياً على جزء من المفتاح الرئيسي (المركب)، كما في الشكل:



- نلاحظ أن A, B تحدد C أي أن C تعتمد اعتماد وظيفياً على A,B، وكذلك أن B تحدد C اي أن C تعتمد اعتماداً وظيفياً على B، وفي هذه الحالة يمكن ان نقول أن هذا الجدول يحتوي على اعتمادية جزئية.

- يوضح الجدول الآتي درجات الطلاب: **Stud**

Sno	Cno	Grades
10	1	70
10	2	88
20	1	100
20	2	94

- فمثلا لو أردنا أن نعرف درجة الطالب رقم 10 ستظهر لنا درجتان وهما 70–88.
- وهذا يدل على أن الدرجات لا تعتمد اعتماداً وظيفياً كاملاً **Full Functional Dependency** على رقم الطالب، بل أنها تعتمد أيضاً على رقم المادة.
- ويمكن توضيحها كالآتي:

Sno , Cno → Grades

- اذن فالدرجات تعتمد وظيفياً كاملاً على رقم الطالب + رقم المادة.

- فمثلا لو أردنا أن نعرف درجة الطالب رقم 10 ستظهر لنا درجتان وهما 70-88.
- وهذا يدل على أن الدرجات لا تعتمد اعتماداً وظيفياً كاملاً **Full Functional Dependency** على رقم الطالب، بل أنها تعتمد أيضاً على رقم المادة.
- ويمكن توضيحها كالآتي:

Sno , Cno ⟶ Grades

- اذن فالدرجات تعتمد وظيفياً كاملاً على رقم الطالب + رقم المادة.
- ومثلاً في حالة أضفنا إلى الجدول اسم الطالب واسم المادة:

Stud

Sno	Cno	Grades	St_name	Course
10	1	70	Ahmed	DB
10	2	88	Ahmed	OS
20	1	100	Salem	DB
20	2	94	Salem	OS

- **مثال:**

Sno , Cno ⟶ St_name

- هنا خطأ لأن رقم الطالب يكفي لنعرف اسمه. وهذه الحالة تسمى اعتماد وظيفي جزئي.
- إذن فإن اسم الطالب يعتمد وظيفياً على رقم الطالب فقط. وهو اعتماد وظيفي جزئي.
- **مثال أخر:**

Sno , Cno ⟶ Course

- هنا خطأ لأن رقم المادة يكفي لنعرف اسمها. وهذه الحالة تسمى اعتماد وظيفي جزئي.
- في الصيغة المعيارية الثانية لا ينفع أن يكون هناك صفة تعتمد اعتماد وظيفي جزئي على المفتاح الرئيسي، ولكن يجب أن يكون اعتماداً وظيفياً كاملاً كما في حالة الدرجات فهي تعتمد على المفتاح المركب (رقم الطالب + رقم المادة).
- اما في حالة اسم الطالب، اسم المادة فهو اعتماد وظيفي جزئي.
- الحل في هذه الحالة أن نضيف جدول جديد لكل صفة مع المفتاح الرئيسي المعتمدة عليه اعتماداً وظيفياً كاملاً . وتحذف الصفات من الجدول الأساسي.

1) ننشأ جدول جديد به المفتاح الرئيسي + الصفة المعتمدة عليه اعتمادية وظيفية كاملة.

2) نحذف الصفة من الجدول الأساسي.

جداول الشكل النظامي الثاني تحل إشكاليات الحذف والإضافة ولا تحل إشكالية التعديل، فمثلا إذا أردنا تعديل بيانات أحد المالكين فيجب تعديل مجموعة من التسجيلات، مما يتفق مع إشكالية التعديل سابقة الذكر

1الشكل النظامي الثالث :(3NF)

يكون الجدول من الشكل النظامي الثالث إذا حقق ما يلي:

- من الشكل النظامي الثاني.
- لا توجد فيه واصفات لا تشكل جزءاً من المفتاح الأساسي وتعتمد بالتعد ي على المفتاح الأساسي

الخطوات التي يتكون منها الشكل النظامي الثالث:

1. تعريف كل المحددات الجديدة: ينبغي اعتبار كل محدد في تبعية متعدية مفتاحا أساسيا لجدول جديد، بحيث يعبر المحدد عن الواصفة التي تحدد قيمتها قيما أخرى ضمن السطر نفسه.
2. تعريف الواصفات ذات التبعيات: يتم في هذه الخطوة تحديد كافة الواصفات المرتبطة بكل محدد معّرف في الخطوة الأولى، كما يتم هنا تعريف عالقات التبعية المتشكلة.
3. إزالة الواصفات ذات التبعيات من التبعيات المتعدية: إزالة كافة الواصفات ذات التبعيات من علاقات التبعيات المتعدية لكل جدول من جداول العلاقة المتعدية.

الجداول Rental و Customer لا تحوي واصفات تعتمد بالتعدي على المفتاح الأساسي، بينما الواصفة owner في الجدول Property يعتمد على owner_ num الذي بدوره يعمد على P_num ولوضع هذا الجدول في الشكل النظامي الثالث يجب تجزئيه إلى الجدولين التاليين:

Property

P_num	P_address	rent	Owner_num
Pr3	A	785	Po23
Pr22	B	1200	Po44
Pr17	C	1000	Po32
Pr32	D	950	Po32

Owner

Owner_num	Owner
Po23	Jones
Po44	Jan
Po32	Jill

التشكل النظامي الثالث يحل إشكاليات التعديل والحتف والاضافة، بالإضافة إلى حل مشكلة تكرار البيانات،
وعندما يطلب تنظيم قاعدة بيانات فيقصد بذلك عادة وضعها في الشكل النظامي الثالث

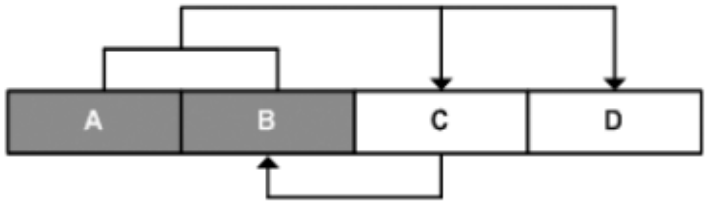
الشكل النظامي Boycee-Codd (BCNF)

يطبق الشكل النظامي BCNFعلى الجداول الحاوية على:

- عدة مفاتيح مرشحة.
- مفاتيح مرشحة مركبة.
- مفاتيح مرشحة متقاطعة.

يكون الجدول من الشكل BCNFإذا كان كل محِّدّد من محدداته يشّكل مفتاحا مرشحا

. وبالتالي إذا كان الجدول يحتوي على مفتاح مرشح وحيد سيتطابق الشكل النظامي Codd-Boyce مع الشكل النظامي الثالث.



A+B⇨C, D
C⇨B

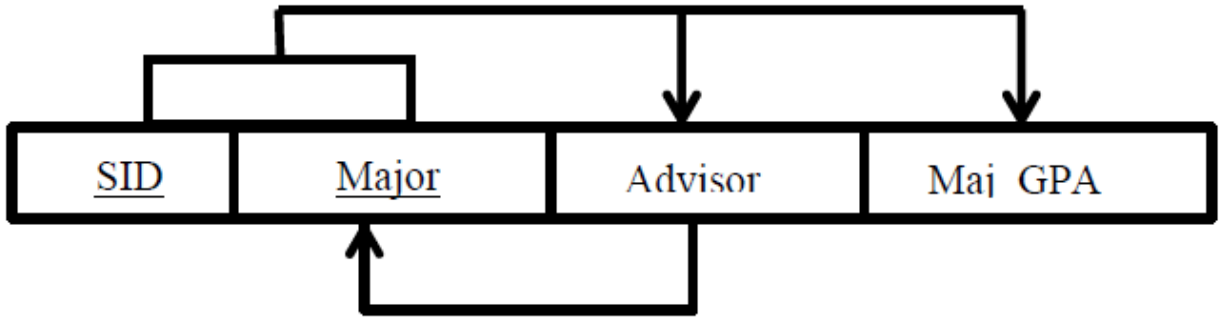
ليكن الشكل التالي، الذي يمثل مخطط تبعيات لجدول معين

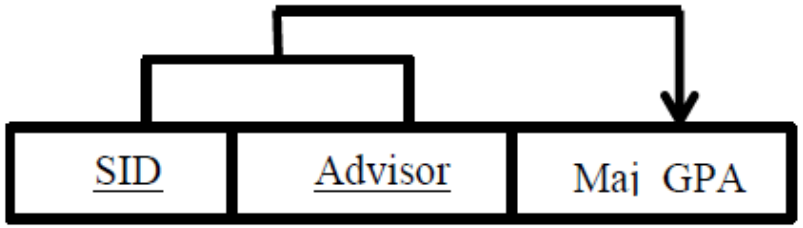
يمكن أن نستنتج من المخطط التبعيات التالية:

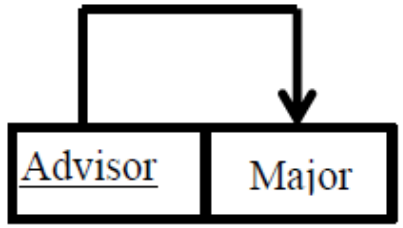
لا يحتوي المخطط على تبعيات جزئية أو متعدية، بالتالي فهو يحقق الشكل النظامي الثالث ولكن الجدول السابق لا يحقق الشكل النظامي Codd-Boyceبسبب وجود علاقة التبعية من Cإلى B

- أولاً ينبغي تحويل المفتاح الرئيسي إلى الحقلين C+A وذلك على اعتبار أن C مفتاح مرشح وهو الشرط الذي نفترض تحققه على كافة واصفات الجدول، وبالتالي فان العلاقة من Cإلى B تتحول الان إلى علاقة تبعية جزئية، وعند هذه النقطة تندرج بنية الجدول تحت الشكل النظامي الأول؛

ليكن المثال التالي: الصفة Majorهي جزء من المفتاح ولكنها تعتمد على Advisor وهو بدوره ليس مفتاح، إلا أن العلاقة من النموذج الثالث وفي BCNF لا تسمح لأي صفه أن تعتمد إلا على SK, CK, PK والحل نخرج Advisorو Majorفي علاقة جديدة يكون المفتاح Advisorوباقي العلاقة يكون Advisorو SIDهما المفتاح.







تمرين : تنظيم البيانات بالشكل BCNF

بفرض أن لدينا الجدول Employee Assignments ويحوي الحقول التالية

EmployeeId , FirstName , LastName و project .

كل موظف يمكن أن يعين في عدة مشاريع وكل مشروع يمكن أن يحوي عدة موظفين.
تجاهل الحالات الشاذة كأن يكون هناك موظفان لهما نفس الاسم.

يجب القيام بما يلي :

1- تبيان سبب كون هذا الجدول ليس من الشكل BCNF أوجد determinant لا يكون أيضا key candidate .

2-تحديد مشكلات البيانات التي يمكن أن تحصل في هذا الجدول.

جعل هذا الجدول من الشكل BCNF بنقل الحقول FirstName، LastName إلى جدول جديد يدعى EmployeeData. وإضافة حقل لربط سجلات الجدول الجديد بالجدول الأصل

الخطوة الأولى- تبيان سبب كون هذا الجدول ليس من الشكل BCNF:

المشكلة في هذا الجدول هي أنه يحتوي مفهومين مختلفين. حيث يحوي عدة أجزاء من البيانات الخاصة بالموظف (EmployeeId، FirstName، LastName) بالإضافة إلى بيانات المشاريع المسندة للموظف.

candidate key لهذا الجدول هي Project/EmployeeId

الخطوة الثانية- تحديد مشكلات البيانات التي يمكن أن تحصل في هذا الجدول:

الخطو ة الثالثة- جعل هذا الجدول من الشكل :BCNF

الحل يكون بنقل أحد الحقول المرتبطة إلى جدول جديد، في هذه الحالة يمكن نقل الحقول FirstName، LastName إلى جدول جديد يدعى EmployeeData. قم بإضافة حقل لربط سجلات الجدول الجديد بالجدول الأصلي.