==========

Higher-Level Normal Forms أشكال عادية عالية المستوى

The normal forms are as follows:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. Boyce-Codd Normal Form
5. Fourth Normal Form
6. Fifth Normal Form
7. Domain Key Normal Form

1. أول نموذج عادي
2. النموذج العادي الثاني
3. النموذج العادي الثالث
4. نموذج Boyce-Codd العادي
5. النموذج العادي الرابع
6. النموذج العادي الخامس
7. النموذج العادي لمفتاح المجال

The **first three** normal forms are the **most critical** for developing a working database. The other normal forms add refinements that are valuable but **not as critical**.

النماذج الثلاثة الأولى هي الأكثر أهمية لتطوير قاعدة بيانات عاملة. تضيف الأشكال العادية الأخرى تحسينات ذات قيمة ولكنها ليست بالغة الأهمية.

1- THE BOYCE – CODD NORMAL FORM (BCNF)

THE BOYCE - CODD NORMAL FORM (BCNF) -1

A table is in (BCNF) when every determinant in the table is a candidate key. Most designers consider the BCNF to be a special case of the 3NF, so how can a table be in the 3NF and not are in BCNF?

يكون الجدول في (BCNF) عندما يكون كل محدد في الجدول هو مفتاح مرشح. يعتبر معظم المصممين أن BCNF حالة خاصة لـ NF3 ، فكيف يمكن أن يكون الجدول في NF3 وليس في BCNF؟

A table is in 3NF when it is in 2NF and there are no transitive dependencies. But what about a case in which a non-key attribute is the determinant of a key attribute? That condition does not violate 3NF, yet it fails to meet the BCNF requirements because BCNF requires that every determinant in the table be a candidate key.

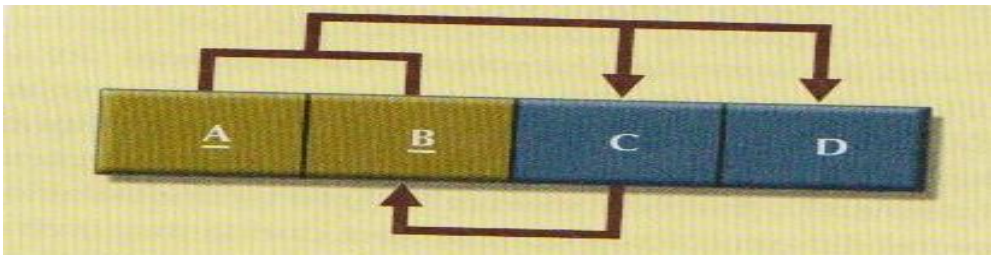
This functional dependency

الجدول في NF3 عندما يكون في NF2 ولا توجد تبعيات متعدية. ولكن ماذا عن الحالة التي تكون فيها السمة غير الرئيسية هي المحدد للسمة الرئيسية؟ هذا الشرط لا ينتهك NF3 ، ومع ذلك فإنه يفشل في تلبية متطلبات BCNF لأن BCNF تتطلب أن يكون كل محدد في الجدول مفتاح مرشح.

هذه التبعية الوظيفية

A+B →C, D

C → B

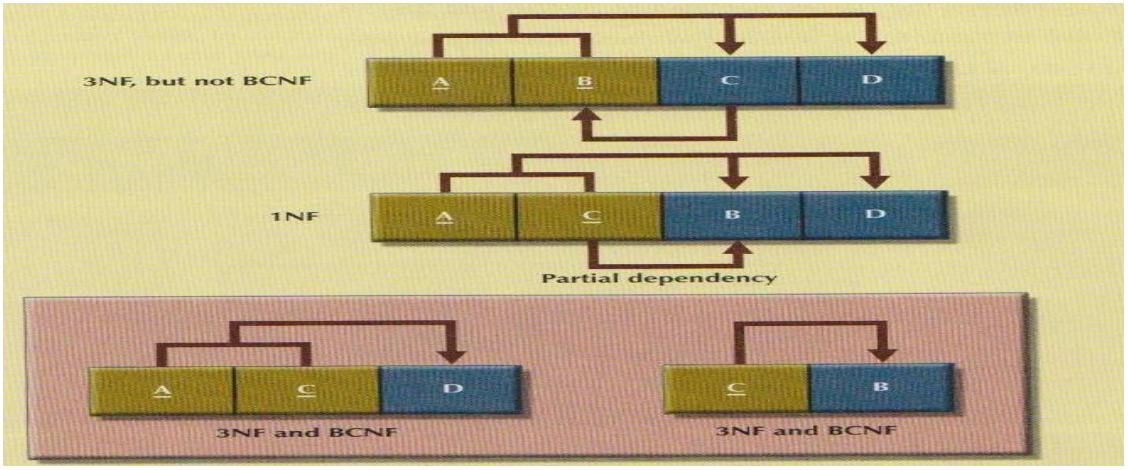


The condition C → B indicates that a non-key attribute determines part of the primary key_ and that dependency is not transitive, thus, the table structure in figure above meets the 3NF requirements. Yet the condition C → B causes the table to fail to meet the BCNF requirements.

يشير الشرط B C إلى أن السمة غير الرئيسية تحدد جزءًا من المفتاح الأساسي _ وأن التبعية ليست انتقالية ، وبالتالي ، فإن بنية الجدول في الشكل أعلاه تلبي متطلبات NF3. ومع ذلك ، فإن الشرط B C يتسبب في فشل الجدول في تلبية متطلبات BCNF.

To convert the table structure above into table structures that are in 3NF and BCNF , first change the primary key to A+C . That is an appropriate action because the dependency C → B means that C is, in effect, a superset partial dependency C→ B. next, follow the standard decomposition procedures to produce the result shown in figure bellow

لتحويل بنية الجدول أعلاه إلى هياكل جدول موجودة في NF3 و BCNF ، قم أولاً بتغيير المفتاح الأساسي إلى A + C. هذا إجراء مناسب لأن التبعية B C تعني أن C هي ، في الواقع ، تبعية جزئية شاملة B C. بعد ذلك ، اتبع إجراءات التحلل القياسية لإنتاج النتيجة الموضحة في الشكل أدناه



To see how this procedure can be applied to an actual problem, examine the sample data in table bellow

لمعرفة كيف يمكن تطبيق هذا الإجراء على مشكلة فعلية ، قم بفحص البيانات النموذجية في الجدول أدناه

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

The table above reflects the following conditions:

- Each CLASS_CODE identifies a class uniquely. This condition illustrates the case in which a course might generate many classes.
- A student can take many classes.
- A staff member can teach many classes but each class is taught by only one staff member.

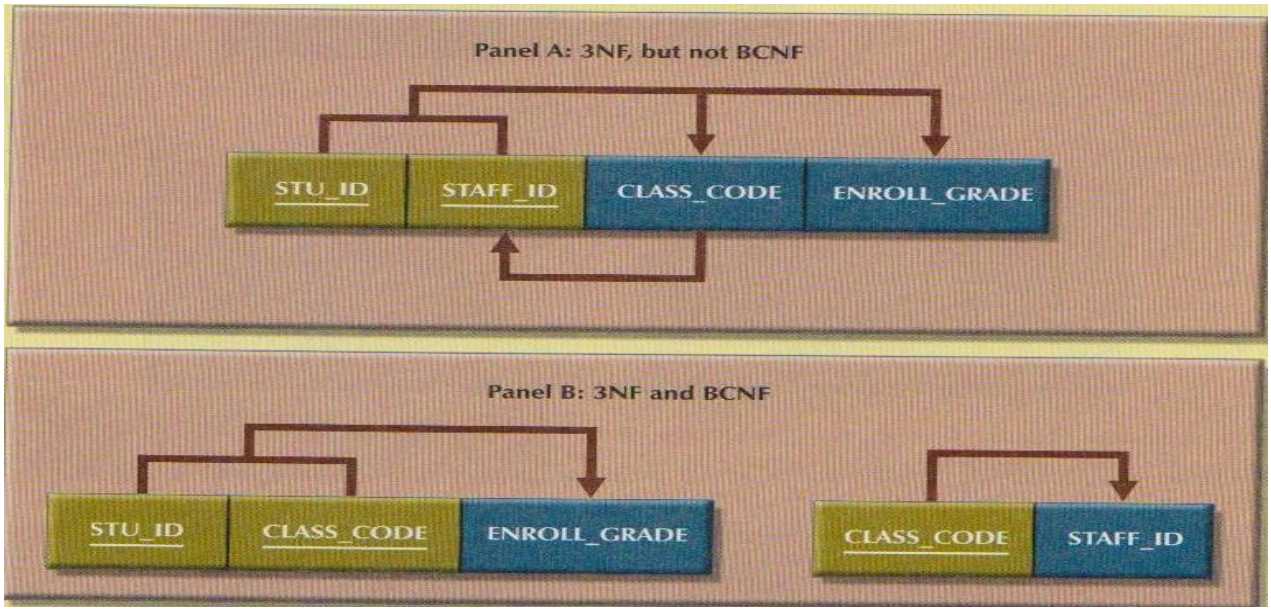
- كل CLASS_CODE يحدد فئة بشكل فريد. يوضح هذا الشرط الحالة التي قد تولد فيها الدورة العديد من الفصول الدراسية.
- يمكن للطالب أن يأخذ عدة فصول.
- يمكن للموظف تدريس العديد من الفصول ولكن يتم تدريس كل فصل بواسطة موظف واحد فقط.

The structure shown in table above is reflected in Panel A of figure bellow

ينعكس الهيكل الموضح في الجدول أعلاه في اللوحة "أ" بالشكل أدناه

==========

STU_ID + STAFF_ID → CLASS_CODE, ENROLLL_GRADE
CLASS_CODE → STAFF_ID



Panel A has a major problem .For example, if a different staff member is assigned to teach class 32456 two rows will require updates thus reducing an update anomaly .And if student 135 drops class 28458 information about how taught that class is lost, thus , producing a deletion anomaly the solution to the problem is in panel B.

توجد مشكلة كبيرة في اللوحة (أ). على سبيل المثال ، إذا تم تعيين موظف مختلف لتدريس الفصل 32456 ، فسيتطلب صفان تحديثات وبالتالي تقليل حدوث شذوذ في التحديث. إنتاج حالة شاذة في الحذف يكون حل المشكلة في اللوحة "ب".

NOTE:

Remember that a table is in BCNF when every determinant in that table is a candidate key. Therefore, when a table contains only one candidate key, 3NF and BCNF are equivalent.

ملاحظة:

تذكر أن الجدول موجود في BCNF عندما يكون كل محدد في هذا الجدول هو مفتاح مرشح. لذلك ، عندما يحتوي الجدول على مفتاح مرشح واحد فقط ، تكون NF3 و BCNF متساويتين.

==========

2-FOURTH NORMAL FORM (4NF)

You might encounter poorly designed database, or you might be asked to convert spreadsheets into a database format in which multiple multivalued attributes exist for Example, consider the possibility that an employee can have many assignments and can also be involved in multiple service organizations

2- النموذج العادي الرابع (NF4)

قد تواجه قاعدة بيانات سيئة التصميم ، أو قد يُطلب منك تحويل جداول البيانات إلى تنسيق قاعدة بيانات حيث توجد سمات متعددة القيم على سبيل المثال ، ضع في اعتبارك إمكانية أن يكون للموظف العديد من التعيينات ويمكن أيضًا أن يشارك في مؤسسات خدمة متعددة

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

There is a problem with the tables in figure above, That is, the tables contain two sets of independent mulivalued dependencies. (One employee can have many service entries and many assignment entries) the tables are likely to contain quite a few null values, in fact the tables do not even have a viable candidate key.(The EMP_NUM values are not unique, so they cannot be PKs .No combination of the attributes in table versions 1 and 2 can be used to create a PK because some of them contain nulls).

توجد مشكلة في الجداول في الشكل أعلاه ، أي أن الجداول تحتوي على مجموعتين من التبعيات متعددة القيم المستقلة. (يمكن أن يكون لموظف واحد العديد من إدخالات الخدمة والعديد من إدخالات التعيين) من المحتمل أن تحتوي الجداول على عدد قليل جدًا من القيم الخالية ، في الواقع لا تحتوي الجداول حتى على مفتاح مرشح قابل للتطبيق. (قيم EMP_NUM ليست فريدة ، لذلك لا يمكن أن تكون لا يمكن استخدام أي مجموعة من السمات في إصدارات الجدول 1 و 2 لإنشاء PK لأن بعضها يحتوي على أصفار).

Version 3 at least has a PK, but it is composed of all of the attributes in the table. In fact, version 3 meets 3NF requirements ,yet it contains many redundancies that are clearly undesirable.

The solution is to eliminate the problems caused by independent multivalued dependencies. You do this by creating the ASSIGNMENT and SERVICE_V1 tables depicted in figure bellow note that neither the ASSIGNMENT nor the SERVICE_V1 table contains independent multivalued dependencies those tables are said to be in 4NF. If you fallow the proper design procedures illustrated, you shouldn't encounter the previously described problem. Specifically, the discussion of 4NF is largely academic if you make sure that your tables conform to the following two rules:

يحتوي الإصدار 3 على الأقل على PK ، ولكنه يتكون من جميع السمات الموجودة في الجدول. في الواقع ، يفي الإصدار 3 بمتطلبات NF3 ، ومع ذلك فهو يحتوي على العديد من التكرار غير المرغوب فيه بشكل واضح.

الحل هو القضاء على المشاكل التي تسببها التبعيات المستقلة متعددة القيم. يمكنك القيام بذلك عن طريق إنشاء جدولي ASSIGNMENT و SERVICE_V1 الموصوفين في الشكل أدناه ، لاحظ أنه لا يوجد جدول ASSIGNMENT ولا جدول SERVICE_V1 يحتوي على تبعيات مستقلة متعددة القيم يُقال إن هذه الجداول موجودة في NF4. إذا تم توضيح إجراءات التصميم المناسبة ، فلن تواجه المشكلة الموضحة مسبقًا. على وجه التحديد ، تكون مناقشة NF4 أكاديمية إلى حد كبير إذا تأكدت من أن جداولك تتوافق مع القاعدتين التاليتين:

1. All attributes must be dependent on the primary key ,but they must be independent of each other
2. No row may contain two or more multivalued facts about an entity.

1. يجب أن تعتمد جميع السمات على المفتاح الأساسي ، لكن يجب أن تكون مستقلة عن بعضها البعض
2. لا يجوز أن يحتوي أي صف على حقيقتين أو أكثر من الحقائق متعددة القيم حول كيان ما.

NOTE:

A table is in 4NF when it is in 3NF and has no multiple sets of multivalued dependencies.

ملاحظة:

يوجد الجدول في NF4 عندما يكون في NF3 ولا يحتوي على مجموعات متعددة من التبعيات متعددة القيم.

==========

Table name: PROJECT

PROJ_CODE	PROJ_NAME	PROJ_BUDGET
1	BeThere	1023245.00
2	BlueMoon	20198608.00
3	GreenThumb	3234456.00
4	GoFast	5674000.00
5	GoSlow	1002500.00

Table name: EMPLOYEE

EMP_NUM	EMP_LNAME
10121	Rogers
10122	O'Leery
10123	Panera
10124	Johnson

Table name: ASSIGNMENT

ASSIGN_NUM	EMP_NUM	PROJ_CODE
1	10123	1
2	10121	2
3	10123	3
4	10123	4
5	10121	1
6	10124	2
7	10124	3
8	10124	5

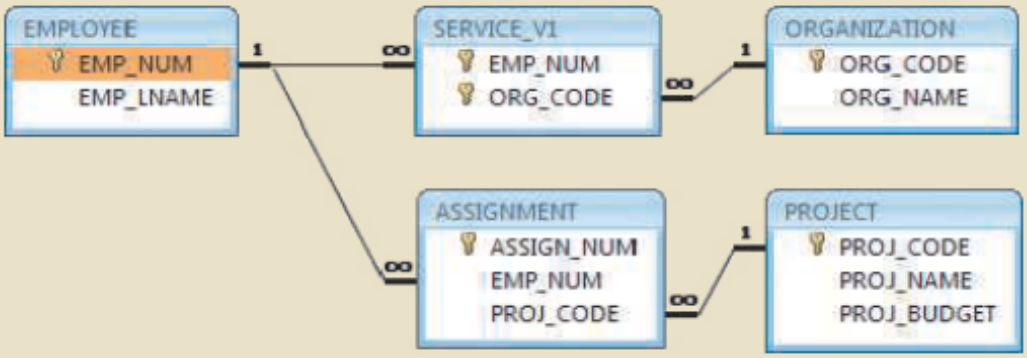
Table name: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

The relational diagram



CODD'S RELATIONAL DATABASE RULES

In 1985, Dr. E. F. Codd published a list of 12 rules to define a relational database system.

قواعد قاعدة البيانات ذات الصلة الخاصة بـ CODD

في عام 1985 ، نشر الدكتور E.F. Codd قائمة من 12 قاعدة لتحديد نظام قاعدة البيانات العلائقية.

RULE	RULE NAME	DESCRIPTION
1	Information	All information in a relational database must be logically represented as column values in rows within tables.
2	Guaranteed Access	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value, and column name.
3	Systematic Treatment of Nulls	Nulls must be represented and treated in a systematic way, independent of data type.
4	Dynamic On-Line Catalog Based on the Relational Model	The metadata must be stored and managed as ordinary data, that is, in tables within the database. Such data must be available to authorized users using the standard database relational language.
5	Comprehensive Data Sublanguage	The relational database may support many languages. However, it must support one well defined, declarative language with support for data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization, and transaction management (begin, commit, and rollback).
6	View Updating	Any view that is theoretically updatable must be updatable through the system.
7	High-Level Insert, Update and Delete	The database must support set-level inserts, updates, and deletes.
8	Physical Data Independence	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	Logical Data Independence	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of column or inserting columns).
10	Integrity Independence	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	Distribution Independence	The end users and application programs are unaware and unaffected by the data location (distributed vs. local databases).
12	Nonsubversion	If the system supports low-level access to the data, there must not be a way to bypass the integrity rules of the database.
	Rule Zero	All preceding rules are based on the notion that in order for a database to be considered relational, it must use its relational facilities exclusively to manage the database.

==========

IMPROVING THE DESIGN

The table structures are cleaned up to eliminate the troublesome partial and transitive dependencies. You can now focus on improving the database's ability to provide information and on enhancing its operational characteristics. In the next few paragraphs, you will learn about the various types of issues you need to address to produce a good normalized set of tables.

تحسين التصميم

يتم تنظيف هياكل الجدول للتخلص من التبعيات الجزئية والمتعددة المزعجة. يمكنك الآن التركيز على تحسين قدرة قاعدة البيانات على توفير المعلومات وتعزيز خصائصها التشغيلية. في الفقرات القليلة التالية ، ستتعرف على الأنواع المختلفة من المشكلات التي تحتاج إلى معالجتها لإنتاج مجموعة جيدة من الجداول المعيارية.

Evaluate PK Assignments

Each time a new employee is entered into the EMPLOYEE table, a JOB_CLASS value must be entered. Unfortunately, it is too easy to make data—entry errors that lead to referential integrity violations. For example, entering DB Designer instead of Database Designer for the JOB_CLASS attribute in the EMPLOYEE table will trigger such a violation. Therefore, it would be better to add a JOB_CODE attribute to create a unique identifier. The addition of a JOB_CODE attribute produces the dependency:

تقييم تكليفات القتل العمد

في كل مرة يتم فيها إدخال موظف جديد في جدول الموظف ، يجب إدخال قيمة JOB_CLASS. لسوء الحظ ، من السهل جدًا ارتكاب أخطاء في إدخال البيانات تؤدي إلى انتهاكات تكامل مرجعي. على سبيل المثال ، سيؤدي إدخال DB Designer بدلاً من مصمم قاعدة البيانات للسمة JOB_CLASS في جدول EMPLOYEE إلى حدوث مثل هذا الانتهاك. لذلك ، سيكون من الأفضل إضافة سمة JOB_CODE لإنشاء معرف فريد. تنتج إضافة سمة JOB_CODE التبعية:

JOB_CODE —> JOB_CLASS, CHG_HOUR

==========

Evaluate Naming Conventions

It is best to adhere to the naming conventions. Therefore, CHG _HOUR will be changed to JOB_ CHG _HOUR to indicate its association with the JOB table. In addition, the attribute name JOB_CLASS does not quite describe entries such as Systems Analyst, Database Designer, and so on; the label JOB_DESCRIPTlON fits the entries better. Also, you might have noticed that HOURS was changed to ASSIGN_HOURS in the conversion from 1NF to 2NF. That change lets you associate the hours worked with the ASSIGNMENT table.

تقييم اصطلاحات التسمية

من الأفضل الالتزام باتفاقيات التسمية. لذلك ، سيتم تغيير CHG _HOUR إلى JOB_ CHG _HOUR للإشارة إلى ارتباطها بجدول JOB. بالإضافة إلى ذلك ، لا يصف اسم السمة JOB_CLASS إدخالات مثل محلل الأنظمة ومصمم قاعدة البيانات وما إلى ذلك ؛ تلائم التسمية JOB_DESCRIPTlON الإدخالات بشكل أفضل. أيضًا ، ربما لاحظت أنه تم تغيير HOURS إلى ASSIGN_HOURS في التحويل من NF1 إلى NF2. يتيح لك هذا التغيير ربط ساعات العمل بجدول التعيين.

Refine Attribute Atomicity

It is generally good practice to pay attention to the atomicity requirement. An atomic attribute is one that cannot be further subdivided. Such an attribute is said to display atomicity. Clearly, the use of the EMP_NAME in the EMPLOYEE table is not atomic because EMP_NAME can be decomposed into a last name, a first name, and an initial.

صقل ذرية السمة

من الممارسات الجيدة عمومًا الانتباه إلى المتطلبات الذرية. السمة الذرية هي السمة التي لا يمكن تقسيمها بشكل فرعي. يقال أن هذه السمة تعرض الذرية. من الواضح أن استخدام EMP_NAME في جدول EMPLOYEE ليس ذريًا لأنه يمكن تحليل EMP_NAME إلى اسم العائلة والاسم الأول والاسم الأولي.

==========

Identify New Attributes

If the EMPLOYEE table were used in a real—world environment, several other attributes would have to be added. For example, An employee hire date attribute (EMP_HIREDATE) could be used to track an employee's job longevity and serve as a basis for awarding bonuses to long—term employees and for other morale—enhancing measures.

تحديد السمات الجديدة

إذا تم استخدام جدول الموظف في بيئة العالم الحقيقي ، فسيتعين إضافة العديد من السمات الأخرى. على سبيل المثال ، يمكن استخدام سمة تاريخ تعيين الموظف (EMP_HIREDATE) لتتبع طول العمر الوظيفي للموظف وتكون بمثابة أساس لمنح المكافآت للموظفين على المدى الطويل ولإجراءات أخرى لتعزيز الروح المعنوية.

Identify New Relationships

According to the original report, the users need to track which employee is acting as the manager of each project. This can be implemented as a relationship between EMPLOYEE and PROJECT.

تحديد العلاقات الجديدة

وفقًا للتقرير الأصلي ، يحتاج المستخدمون إلى تتبع الموظف الذي يعمل كمدير لكل مشروع. يمكن تنفيذ ذلك كعلاقة بين الموظف والمشروع.