



الفصل الأول: نماذج البيانات 1

الصفحة	العنوان
3	1. مقدمة
3	2. تعاريف
3	3. الوحدات الأساسية المكونة لنماذج البيانات
3	1.3 الكيانات Entities
3	2.3 الخصائص Attributes
4	3.3 العلاقات Relationships
4	4.3 أشكال العلاقات
6	4. تصميم قواعد البيانات وسياسات العمل Business Policies
6	5. تطور نماذج البيانات
8	1.5 النموذج الهرمي التسلسلي Hierarchical Model
10	2.5 النموذج الشبكي Network Model
11	3.5 النموذج العلائقي Relational Model
13	4.5 النموذج العلائقي/الغرضي Object/Relational Model
14	6. المراجع

الكلمات المفتاحية

خصائص، صف، سياسات عمل، كائن، بيانات، نمط، علاقة، مخطط، جدول، علاقة واحد لواحد، علاقة واحد لعدة، علاقة عدة لعدة.

ملخص

يركز هذا الفصل على التعريف بنماذج البيانات والمقارنة فيما بينها.

الأهداف التعليمية

يهدف هذا الفصل إلى:

- تحديد أهمية نماذج البيانات
- الوحدات الرئيسية المكونة لنماذج البيانات
- العوامل المؤثرة في تصميم قواعد البيانات
- مساوئ ومحاسن عدد من نماذج البيانات

1. مقدمة

إذا قمت بسؤال أي مطور برمجيات خاصة بالمؤسسات عن الخطوة الأكثر أهمية في عملية التطوير فسيكون الجواب غالباً بأنها مرحلة تحليل متطلبات الزبون Analyzing Requirements. تعتبر نمذجة البيانات إحدى التقنيات الأكثر استخداماً في هذه المرحلة وتهدف إلى تطوير نموذج دقيق أو تمثيل بياني Graphical Representation لمتطلبات الزبون، مما يقودنا بدايةً لمجموعة من التعاريف.

2. تعاريف

1. النموذج Model:

تمثيل مجرد للأغراض والأحداث المرتبطة في العالم الحقيقي، إنه تجريد مبسط للواقع.

2. نموذج البيانات Data Model:

تمثيل مجرد (أو توصيف) للبيانات الخاصة بمؤسسة حيث يوضح هذا النموذج الكيانات Entities والأحداث Events والنشاطات Activities المرتبطة بهذه المؤسسة. نستطيع القول بأن نموذج البيانات يصف المؤسسة بحد ذاتها ويهدف إلى تمثيل البيانات وجعلها أكثر قابلية للفهم.

3. الوحدات الأساسية المكونة لنماذج البيانات

الكيانات Entities

الكيان هو شيء Thing أو غرض Object ذو معنى بالنسبة لعمل المؤسسة Business يمكن توصيفه من خلال مجموعة خصائص. قد تكون الكيانات شخصاً، بناءً أو نشاطاً (كموعد أو عملية)

مثال:

يمكن تمييز أحد الكيانات الرئيسية في مكتبة وهو الكتاب. لاحظ بأنه يجب تمييز كل كتاب في المكتبة عن أمثاله بعنوانه أو رقمه أو مميز آخر وذلك وفقاً لطريقة العمل في المكتبة.

الخصائص Attributes

إن عملية وصف الكيان بعد تمييزه تعني تحديد جميع خصائصه التي نريد معرفتها وتخزينها في قاعدة البيانات.

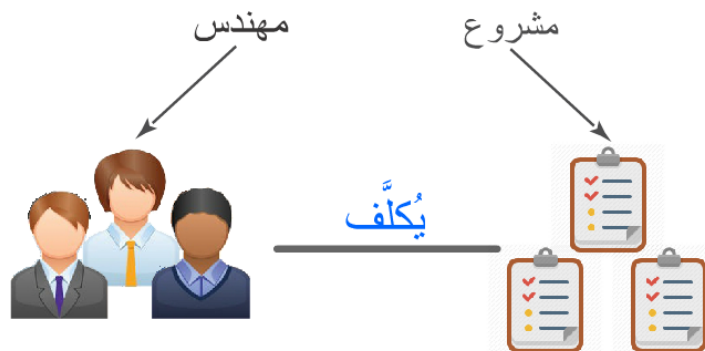
مثال:

إذا عدنا إلى كيان الكتاب نجد أن خصائصه هي: العنوان، الرقم ISBN، الناشر، المؤلف، تاريخ النشر، الخ... لاحظ أن خصائص الكيان يمكن أن تختلف من تطبيق لآخر، فقد يهتم صاحب مكتبة بالخاصية "عدد الصفحات"، بينما لا يهتم آخر بهذه المعلومة، وعندها لا يصبح "عدد الصفحات" أحد الخصائص المطلوبة.

العلاقات Relationships

تشكل العلاقات الروابط Association or Linkage بين الكيانات، كأن نقول مثلاً "يُكلف مهندسي الشركة بالمشاريع".

لاحظ أن الإشارة إلى العلاقة بين الكيانيين مهندس ومشروع تمت باستخدام فعل التكليف.
تُصنف العلاقات بدرجتها Cardinality وفيما إذا كانت اختيارية Optional أم إجبارية Mandatory.



"لاحظ أن الإشارة إلى العلاقة بين الكيانيين
مهندس ومشروع تمت باستخدام فعل التكليف"

مثال:

"لا يمكن تكليف أي مهندس بأكثر من ثلاثة مشاريع في الوقت نفسه، ويجب أن يُكلف بكل مشروع مهندسين اثنين على الأقل".

إن درجة هذه العلاقة من المهندس باتجاه المشروع هي ثلاثة، ومن المشروع باتجاه المهندس هي اثنين، إذاً نقول بأن العلاقة تُصنف بعلاقة عدة لعدة Many-to-Many، وسنتحدث لاحقاً عن أنواع العلاقات.

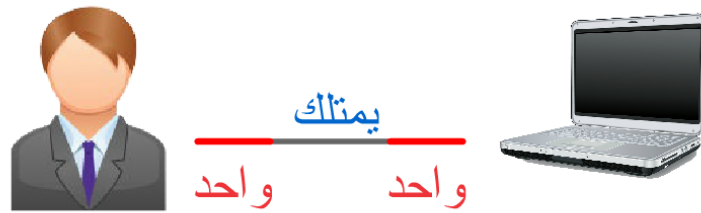
إذا كان هناك احتمال أن تكون درجة العلاقة صفراً تصبح العلاقة اختيارية، وإذا كانت على الأقل واحد تصبح إجبارية. يُعبر عن العلاقات الإجبارية بجملة مثل:

"يجب أن يسجل كل طالب بثلاثة صفوف على الأقل في كل فصل".

أشكال العلاقات

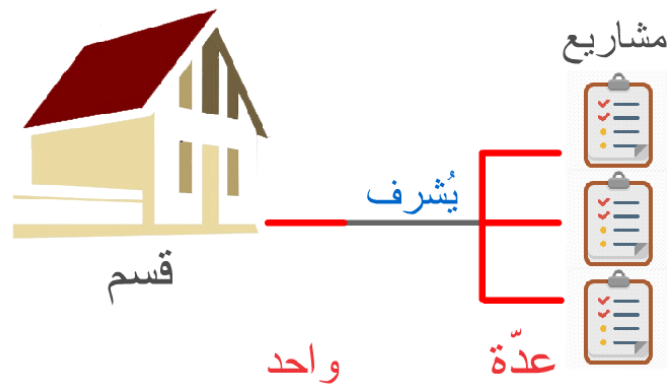
علاقة واحد لواحد One-to-One

"يجب أن يملك كل مهندس حاسوب وحيد"، تعبر هذه الجملة عن علاقة واحد لواحد بين كيان المهندس وكيان الحاسوب في المؤسسة حيث لا يحق للمهندس امتلاك أكثر من حاسوب، ولا يمكن أن يكون الحاسوب ملكاً لأكثر من مهندس. وتُعتبر هذه العلاقة من العلاقات القليلة الاستخدام في نماذج البيانات.



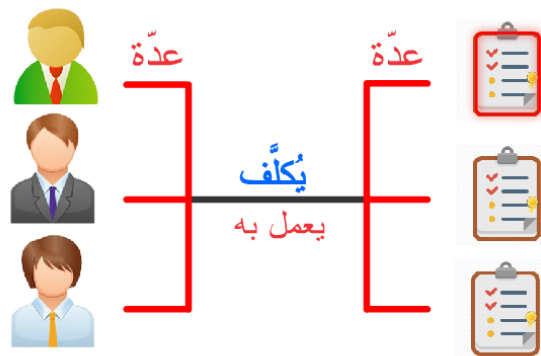
علاقة واحد لعدة One-to-Many

"يمكن أن يشرف كل قسم في المؤسسة على أكثر من مشروع ويجب أن يكون كل مشروع تابعاً لقسم واحد على الأكثر"، تعبر هذه الجملة عن علاقة واحد لعدة بين المشاريع والأقسام في المؤسسة، حيث تكون العدة Many من جهة كيان المشروع، والواحد One من جهة الكيان قسم. نميز في هذا النوع من العلاقات بين الكيان الأب Parent (القسم) والكيان الابن Child (المشروع).



علاقة عدة لعدة Many-to-Many

"يُكلف كل مهندس بعدة مشاريع ويعمل في كل مشروع عدد من المهندسين"، تُعبر هذه الجملة عن علاقة عدة لعدة بين المشاريع والمهندسين.



"يُكلف كل مهندس بعدة مشاريع ويعمل في كل مشروع عدد من المهندسين"

4. تصميم قواعد البيانات وسياسات العمل Business Policies

إن تحديد الكيانات وخصائصها والعلاقات فيما بينها من قبل مصممي قواعد البيانات هو أمر صعب وحساس جداً. قد يتم عمل كهذا بناءً على الفهم الكامل لنوع البيانات المستخدمة ضمن الشركة وكيفية استخدامها، لكنه لا يعطي بالضرورة فهماً واضحاً لطبيعة عمل الشركة وسياساتها.

يتم استخلاص سياسات العمل Business Policies من وصف دقيق ومعمق لعمل الشركة. ويتم استخدامها لإحداث إجراءات جديدة أو لترسيخ إجراءات سابقة.

إن سياسات العمل الموثقة بشكل جيد تساعد بشدة على تعريف الأجزاء الرئيسية لنمط البيانات، من الكائنات إلى خصائص هذه الكيانات والعلاقات فيما بينها والشروط الموضوعة عليها. فالعلاقات الموجودة بين الكائنات تظهر إحدى سياسات العمل التي تحكم العلاقة بين هذه الكائنات.

إن معرفة سياسات العمل تدعم فكرة تصميم وتطوير نموذج البيانات بناءً على طريقة العمل الحقيقية للشركة ودور البيانات فيها. لذلك يجب تحديد سياسات العمل من قبل المصممين لتحديد أثرها على طبيعة ودور ومدى عمل البيانات، وفوق ذلك فإن هذه السياسات تقدم أدوات ليتواصل كل من مصمم ومستخدم قاعدة البيانات بما يسمح بتحديد إجراءات العمل وطريقة تحرك البيانات ضمن النظام، كما أنها تعطي مصمم قواعد البيانات أدوات ليحدد من خلالها العلاقات بين الكائنات والقيود أو المواصفات للكائنات التي تتألف منها قاعدة البيانات.

5. تطور نماذج البيانات

إن السباق من أجل إيجاد إدارة بيانات أفضل قد أوجد عدة نماذج مختلفة حاولت جميعها أن تحل النقص الموجود في بنية نظام الملفات. وبما أن كل نموذج قد تم تقييمه من قبل النموذج الذي سبقه فسنعرض النماذج حسب تسلسلها الزمني الذي ظهرت فيه، الهرمي التسلسلي Hierarchical ثم الشبكي Network ثم العلائقي Relational فالعلائقي/الغرضي Object/Relational وأخيراً الموجهة بالأغراض Object-Oriented.

يوضح الشكل التالي التطور التاريخي لنماذج البيانات:

Timeframe	Technology	Remarks
Pre-1968	File Processing	Predecessor of database processing. Data maintained in lists. Processing characteristics determined by common use of magnetic tape medium
1968-1980	Hierarchical and network models	Era of non-relational database processing. Prominent hierarchical data model was DL/I, part of IBM's first DBMS called IMS. Prominent network data model was CODASYL DBTG model; IDMS was most popular network DBMS.
1980 to present	Relational data model	Relational data model, first published in 1970; began to see commercial application in 1980. IBM endorsed it with DB2; other vendors followed by modifying their DBMS products or by creating new ones. Oracle achieved prominence. SQL became standard relational language.
1982	First microcomputer DBMS products	Ashton-Tate developed dBase products; Microrim created R: Base; Borland followed with Paradox
1985	Interest in object-oriented DBMS (OODBMS) develops	With advent of object-oriented programming, OODBMS were proposed. Little success commercially, primarily because advantages did not justify the cost of converting billions of bytes of organizations data to new format. Under development today.
1991	Microsoft ships Access	Personal DBMS created as element of Windows. Gradually supplanted all other personal DBMS products

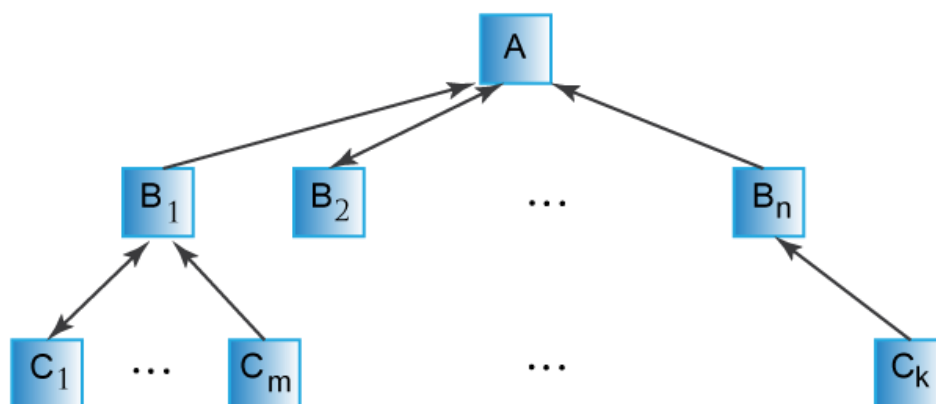
1995	First Internet database applications	Databases become key component of Internet applications. Popularity of the Internet greatly increases need and demand for database expertise.
1997	XML applied to database processing	Use of XML solves long-standing database problems. Major vendors begin to integrate XML into DBMS products.

النموذج الهرمي التسلسلي Hierarchical Model

في نهاية الستينات كانت تعمل شركة في شمال أميركا على نظام الملفات التقليدي وحاولت أن تطور نظام قواعد البيانات الخاص بها، فكتشفت أن لديها أكثر من 60% من البيانات مكررة بلا فائدة، مما دفعها لتطوير نظام ليتعامل مع هذا الحجم من البيانات غير اللازمة، ولذلك تم تطوير نموذج نظام البيانات الذي يركز على مفهوم أن كل الأجزاء الصغيرة للبيانات يمكن أن تكون مع بعضها البعض جزءاً من مكون أكبر للبيانات، وهكذا إلى أن يتم الوصول لوحدة البيانات كاملة. مما أدى إلى ظهور نظام إدارة المعلومات IMS وقد أصبح نظام قواعد البيانات الأعم والأكثر انتشاراً في العالم في السبعينات وبداية الثمانينات.

وقد أدى وجوده إلى بداية التحرك لإيجاد نماذج بيانات أفضل، إلا أن المفاهيم الأساسية لهذا النموذج كانت الأساس في باقي النماذج كما أن محدوديته أدت إلى السعي للوصول إلى نماذج تقدم وجهات نظر أفضل للبيانات. ونستطيع أن نرى بعض المفاهيم الأساسية مازالت موجودة في نماذج البيانات الحديثة المستخدمة في الوقت الحالي.

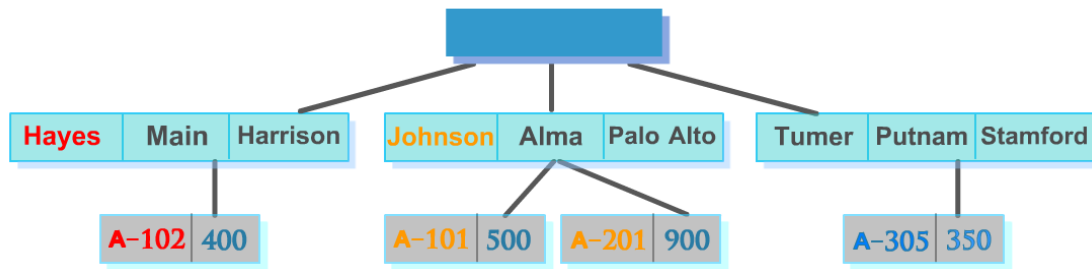
ينظم نموذج البيانات الهرمي التسلسلي بياناته في بنية شجرية Tree Structure كما يوضح الشكل التالي:



مثال:

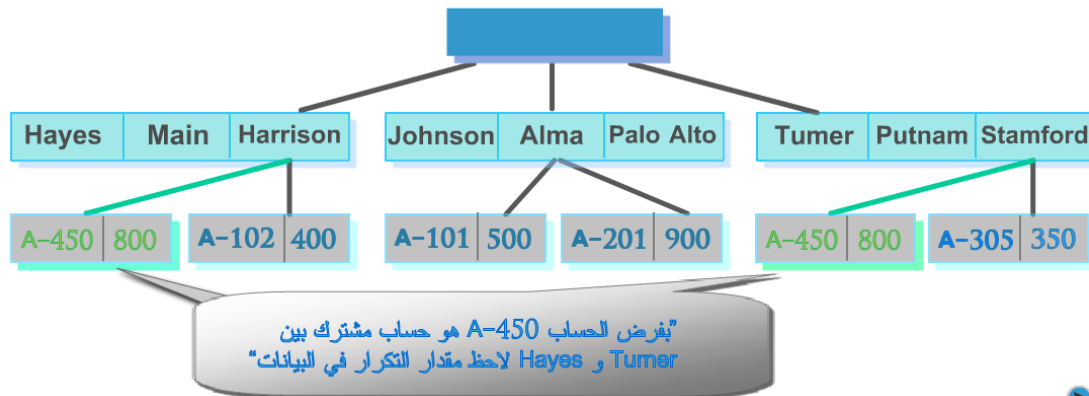
تتألف قاعدة بيانات بنك من نوعين من السجلات: زبون Customer وحساب Account.
يتألف سجل الزبون من ثلاثة حقول: الاسم، الشارع والمدينة.
يتألف سجل الحساب من حقلين: الرقم والقيمة.

يمثل الشكل التالي قاعدة البيانات الهرمية:



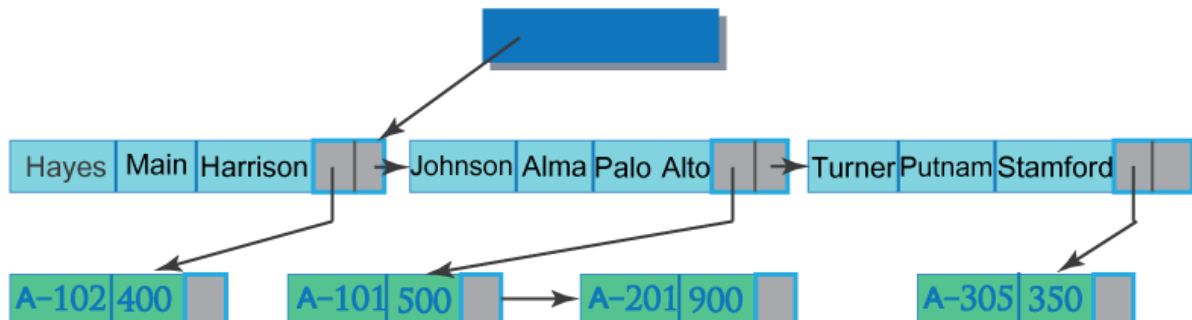
مثال:

لاحظ مقدار التكرار في البيانات عندما يكون أحد الحسابات مشتركاً بين أكثر من شخص.



مثال:

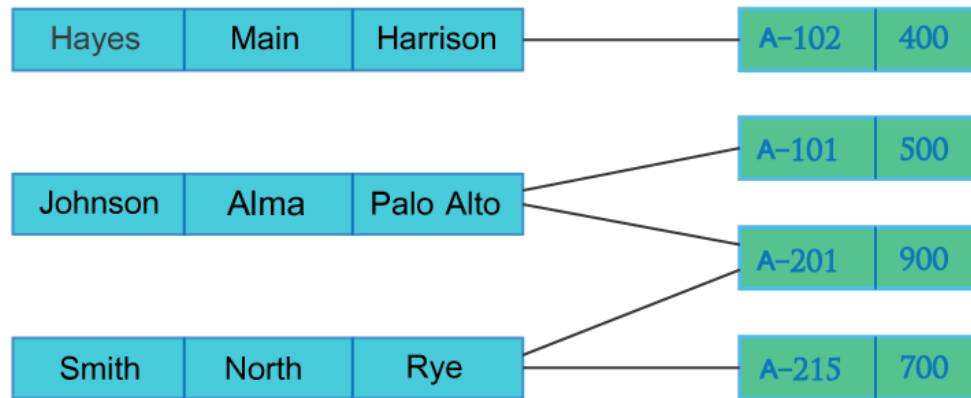
ويوضح الشكل التالي طريقة التخزين في الملف الفيزيائي باستخدام المؤشرات:



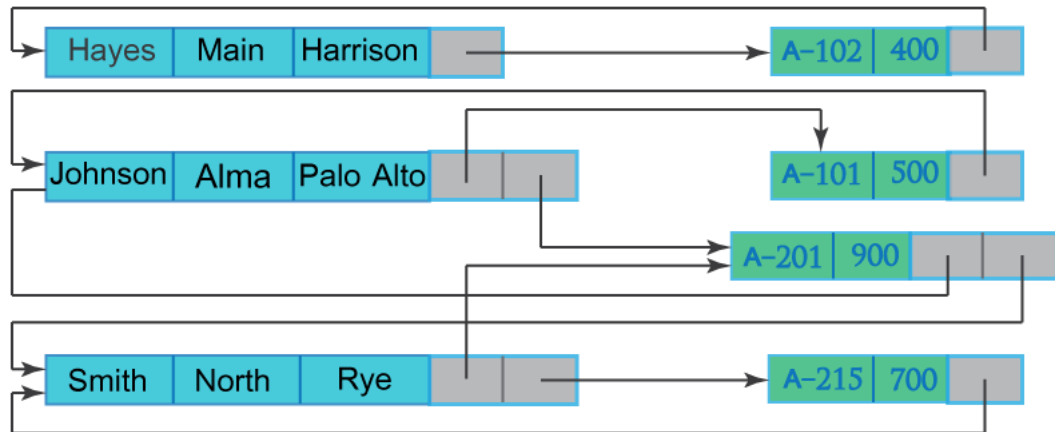
يتميز النموذج الهرمي ببساطة البنية، والفاعلية عندما توجد كمية كبيرة من البيانات التي تكون العلاقة بينها من النوع واحد لعدة One-to-Many. ولكن تطبيقاته كانت معقدة نوعاً ما ومرتبطة ببنية تخزين البيانات، ويزداد هذا التعقيد بشكل كبير عند تمثيل العلاقة عدة لعدة Many-to-Many، وبالإضافة إلى ذلك فإن المشكلة الأهم في هذا النموذج تكرار البيانات.

النموذج الشبكي Network Model

لقد تم تصميم نموذج البيانات الشبكي ليقوم بتقديم علاقات البيانات المعقدة بطريقة فعالة وسهلة أكثر من نموذج البيانات الهرمي التسلسلي، وليقوم بتحسين أداء قواعد البيانات وليقدم ويفرض مقاييس على قواعد البيانات. يتشابه نموذج البيانات الهرمي التسلسلي والنموذج الشبكي في العديد من النواحي، فمثلاً كما في نموذج البيانات الهرمي التسلسلي فإنه في النموذج الشبكي يتم التعامل مع العلاقات على أنها علاقات واحد لعدة (M:1) ولكن على عكس نموذج البيانات الهرمي التسلسلي فإنه يسمح للابن بالحصول على أكثر من أب واحد، لذلك يتم التعامل مع العلاقات بشكل أسهل هنا كما هو واضح بالشكل:



ويوضح الشكل التالي طريقة التخزين في الملف الفيزيائي باستخدام المؤشرات:



لقد حافظت نماذج البيانات الشبكية على العديد من ميزات نماذج البيانات الهرمية التسلسلية، كما أضافت إليها بعض الميزات ولعل أهمها التعامل الأسهل مع العلاقة عدة لعدة.

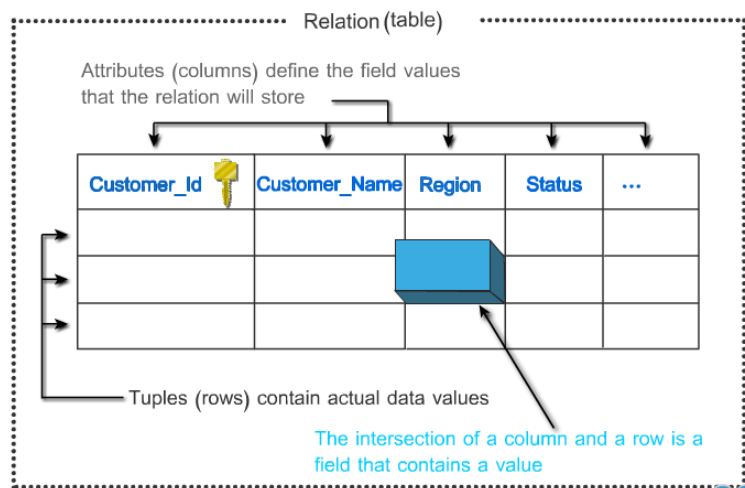
النموذج العلائقي Relational Model

تم استبدال نموذج البيانات الشبكي في بداية الثمانينات بنموذج البيانات العلائقي، وقد بدأ بتطويره في بدايات السبعينات بواسطة IBM حيث قدمت ثورة في تصميم واستخدام قواعد البيانات. إلا أن هذا العمل قد تم اعتباره في السبعينات كنظم متطورة وجيدة ولكن غير عملية حيث أنها تعتمد على قدرة الحواسيب، وفي ذلك الوقت كانت الحواسيب غير متوفرة وتعتمد على قدرات ومجهود كبير، إلا أن التطور الحاصل في مجال الحواسيب وتطور أنظمة التشغيل والبدء بالتقدم في مجال قطع الحواسيب وتوفرها ورخصها كل ذلك أدى إلى البدء في التعامل مع هذا النموذج. ونتيجة التطور الهائل على الحواسيب في الآونة الأخيرة، أصبح هذا النموذج الأكثر انتشاراً، وذلك بتوفر الحواسيب الصغيرة التي تكلف أقل مما كانت تكلفه الحواسيب الأولى، والتي تستطيع أن تشغل برمجيات علائقية متطورة ومعقدة بسهولة.

تتميز أنظمة إدارة قواعد البيانات العلائقية بأنها تسمح للمستخدم والمصمم بالعمل على بيانات عمل منطقية Logical وطبيعية ونقوم هي بالتعامل مع كل التفاصيل الفيزيائية المعقدة.

البنية الأساسية للنموذج العلائقي

أسس العالم Dr. E.F. Codd النظريات الرياضية، التي استند عليها نموذج البيانات العلائقي في عام 1970. وتعتبر البنية الأساسية لهذا النموذج هي Relation أو الجدول Table، وهو عبارة عن مصفوفة تتألف من سلسلة من الأعمدة Columns والأسطر Rows. يمثل الجدول الكيان Entity، والأعمدة خصائص الكيان Attributes، أما الأسطر (أو Tuples) فتمثل حدوث Instances الكيان.



محاسن النموذج العلائقي

• استقلالية البنية التصميمية

وذلك لأن هذا النموذج لا يستخدم المسار للبحث عن البيانات وبذلك تكون التغييرات على البنى لا تؤثر في طرق الوصول للبيانات، وهذا ما كانت تفتقده النماذج السابقة حتى هذا الوقت.

- **أصبحت المفاهيم أسهل وأوضح مع أنها كانت سهلة وواضحة من قبل**
فقد أصبح من الممكن تجاهل مكان وكيفية تخزين البيانات، وأصبح الاهتمام الأكبر بطريقة عرض البيانات، أي طريقة عرضها للأشخاص وليس للحاسب.
- **أصبحت قواعد البيانات أسهل في التصميم والتطبيق والإدارة والاستخدام**
لأن نموذج البيانات العلائقي حقق استقلالية في البيانات واستقلالية في البنى، مما أعطى سهولة في تصميم قاعدة البيانات والتعامل مع محتوياتها.
- **إضافة الاستعلامات اللحظية مما أدى لظهور لغات الاستعلامات البنيوية (SQL) وهي من لغات الجيل الرابع**
ومن أجل استخدامها يجب توفر ثلاثة أمور: واجهة للمستخدم ومجموعة من الجداول في قاعدة البيانات ومحرك بحث للغة الاستفسار.
- **وجود نظام إدارة قواعد بيانات فعال ومتطور أكثر من الموجود في النماذج السابقة**
حيث أن هذه البرامج الجيدة قادرة على التعامل مع التعقيد الموجود ضمن قاعدة البيانات وإخفائه عن المستخدم والمصمم.

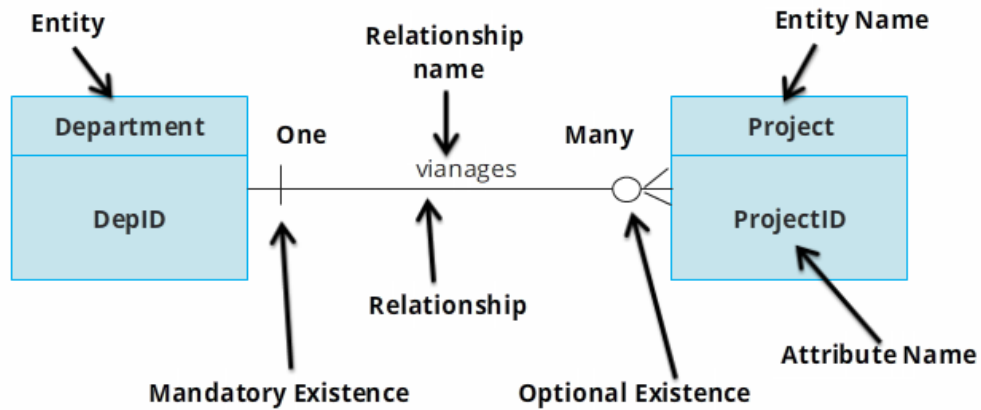
مساوئ النموذج العلائقي

- تحتاج نفس البرمجيات القادرة على إخفاء التعقيدات الموجودة إلى تجهيزات حاسوبية متطورة وتستهلك موارد كبيرة من نظام التشغيل، مما جعل النسخ البدائية من هذا النموذج تجعل عمل الأجهزة بطيئاً نسبياً، ولكن مع تطور الأجهزة وسرعتها تم حل هذه المشكلة تدريجياً.
- شجعت السهولة في الاستخدام العديد من الأشخاص القليلي الخبرة على العمل بها، مما أدى لظهور عدد من التصميمات السيئة والتطبيقات الضعيفة، وبالتالي ظهور نفس الأخطاء التي كانت تظهر في نظام الملفات في بعض الأحيان.

نمذجة الكيانات-العلاقات Entity-Relationship Modeling

قدم العالم تشين في عام 1976 نمذجة الكيانات-العلاقات (ERM)¹، وهي طريقة تصميمية تصف العلاقات بين الكيانات في قاعدة البيانات.

يتم تقديم هذه الطريقة بواسطة مخطط الكائنات العلائقية (ERD)² الذي يستخدم الصور ليقدم وينمذج مكونات قاعدة البيانات من كيانات وعلاقات كما يبين الشكل التالي، وسنتحدث بالتفصيل عن الـ ERD في فصول لاحقة.



النموذج العلائقي/الغرضي Object/Relational Model

أدى التطور الحاصل في طبيعة البيانات المراد تخزينها مثل المقاطع الصوتية، مقاطع الفيديو، والخرائط الجغرافية إلى الحاجة لبنى أكثر تعقيداً من التي توفرها النماذج العلائقية. لذلك تم إدخال المفاهيم الغرضية Object للنموذج العلائقي مع المحافظة على العلاقة Relation أو الجدول Table كبنية رئيسية للبيانات، وكانت النتيجة ظهور النموذج العلائقي/الغرضي الذي يسمح بتخزين أنماط معطيات Data Types معقدة في حقول الجدول. يوضح الشكل التالي جدولاً للممثلين يكون فيه حقل العنوان جدولاً بحد ذاته:

moviestar (name, address (street,city), birth, movies (title,year))					
Name	Address		Birth	Movie	
Fisher	Street	City	9/9/1950	Title	Year
	Maple	Hollywood		Star Wars	1977
	5.Avenue	New York		Empire	1980
Hamill	Street	City	8/8/1962	Title	Year
	Sunset Blvd	LA		Star Wars	1977
				Return	1983

أحد الأشكال لهذه الأنماط المعقدة هو أن يمتلك أحد الحقول في جدول أكثر من معلومة، مثال: تتمثل النقطة الجغرافية بإحداثيات X,Y,Z، لذلك نحتاج لتخزينها كحقل إلى بنية خاصة غير موجودة في النموذج العلائقي.

6. المراجع

- [Article: Data Modeling Finding the Perfect Fit.pdf](#)
- [Article: Why Build A Logical Data Model.pdf](#)
- <http://database.ittoolbox.com>
- <http://www.vtc.com/products/datamodeling.htm>
- <http://www.utexas.edu/its/windows/database/datamodeling/index.html>