3.2. أهم مفاهيم قواعد البيانات العلائقية

3.2.1. المفتاح الأساسي للجدول: Primary Key

الجدول يتألف من صفوف (أو سجلات Records)، و في قواعد البيانات العلائقية يجب ألا يتكرر الصف كاملاً بل يجب أن توجد قيمة فريدة تميز كل صف من الصف الآخر. هذا ما نسميه بالمفتاح الأساسي للجدول. كمثال على ذلك في جدول المستعيرين يمكن أن يكون لدينا أشخاص لهما الإسم نفسه وتاريخ الميلاد نفسه وكل البيانات عنهما متماثلة لذلك نميز بينها عن طريق رقم الهويه مثلاً ولكن هذا التمييز صعب بعض الشيء وأحياناً غير ممكن. لذلك سهلت لنا قواعد البيانات ميزة الترقيم التلقائي بحيث عندما نضيف سجلاً جديداً فإنه يأخذ رقماً خاصاً به تلقائياً ودون أن ندرج له رقماً في عمود المفتاح الأساسي و بالتالي نميز بين كل صف في الجدول عن طريق رقمه الذي يميزه.

3.2.2. المفتاح الغريب(الأجنبي) Foreign Key

في قواعد البيانات هناك مصطلح اسمه التكامل المرجعي Referential Integrity أي إنه لو فرضنا أنه يُوجد لدينا جدولان مثل Addresses، Employees يتم الربط بينهما عن طريق وضع عمود في أحدهما، هذا العمود له اسم عمود المفتاح الأساسي نفسه في الجدول الآخر، وله نوع البيانات نفسها Data Type والصفات نفسها Attributes. هذا العمود نسميه (مفتاح غريب) بحيث يصبح هذا العمود هو صلة الوصل بين الجدول الأول والجدول الثاني. هذه العملية نسميها بالتكامل المرجعي.

إذاً كتعريف نهائي للتكامل الرجعي: التكامل المرجعي هو علاقة بين جدولين يحوي الجدول الأول على مفتاح أساسي والجدول الثاني يرتبط مع الجدول الأول بمفتاح غريب له اسم المفتاح الأساسي نفسه وبياناته نفسها وصفاته نفسها. أما المفتاح الغريب فهو عمود في جدول يشير إلى عمود في جدول آخر هو المفتاح الأساسي للجدول الأخير و يكون لكل من العمودين الأساسي والغريب الاسم نفسه ونوع البيانات المدخلة فيهما والصفات نفسها. فعند إضافة بند إلى الجدول ذي المفتاح الغريب وليس موجوداً

Scanned by CamScanner

في الجدول ذي المفتاح الأساسي فسيقوم ملقم قواعد البيانات بتنبيهي إلى أنني أدرج شيئاً غير موجود في الجدول الأساسي.

من الأشياء المفيدة التي يقدمها لنا التكامل المرجعي هي عملية الحذف المتسلسل Deletes Cascading. ففرضاً أن لديك موظفاً مسجلاً في الجدول Employees و لدى هذا الموظف ثلاثة عناوين مسجلة في الجدول Addresses، ثم قدم هذا الموظف طلب استقالة ووافقت أنت عليه عندها ستقوم بحذف سجله من قاعدة البيانات الخاصة بشركتك. تخيل مدى الصعوبة والإرباك الذي ستتعرض له عند كل عملية حذف و ولاسيما مع الزبائن أو مع الأشخاص الذين يسجلون على بريد إلكتروني مجاني ثم لا يعودوا إلى البريد والكثير من العمليات المشابهة. وهنا تبرز أهمية الحذف المتسلسل فعندما أطلب من ملقم قواعد البيانات أن يحذف سجل هذا الموظف من الجدول Employees فإنه سيقوم آلياً بحذف جميع السجلات المقابلة في الجدول الذي يحتوي على مفتاح غريب

3.2.3. القيمة Null

في قواعد البيانات العلائقية سنتعرض لكثير من القيمة Null وهي تعني اللاقيمة (أو عديم القيمة) أي لا شيء وهي بالطبع تختلف عن الصفر Zero

3.2.4. الفهارس Indexes

الفهرس في قواعد البيانات كالفهرس المعروف للكتب ويؤدي وظيفته نفسها. ففي الكتاب يساعدنا الفهرس على سرعة الانتقال إلى موضوع معين في الكتاب. كذلك الأمر في قواعد البيانات، فالفهارس تساعدنا على سرعة الانتقال و الحصول على معلومات لاستعلام ما بسرعة أكبر.

Employee

Emp N°	EmpName	Job	Adress	Depn°

Department

Depn°	Location	Tel

Scanned by CamScanner

3.3. أنواع العلاقات

3.3.1. علاقة واحد لواحد

One-to-One Relationships

في علاقة واحد لواحد كل سجل من الجدول الأول يرتبط بسجل واحد فقط من الجدول الثاني وكل سجل من الجدول الثاني يرتبط بسجل واحد فقط من الجدول الأول أي إن كلا الجدولين يمكن أن يكون سجل واحد فقط على جانبي العلاقة كما في الشكل

1. نلاحظ من هذه القاعدة أن قيمة المفتاح الأساسي في كل سجل من الجدولين هي نفسها، في غياب مثل هذه القاعدة، يمكن الجمع بين كلا الجدولين في جدول واحد دون خرق قواعد التنظيم (سنقوم بدراسة مفهوم التنظيم في الفصل الرابع من هذا الكتاب).

. One to One

الشكل 1 علاقة راس برأس أو علاقة واحد إلى واحد

في علاقة واحد إلى واحد بين الجدول A و الجدول B عدد الصفوف في الجدول A يجب أن يساوي عدد الصفوف في الجدول B.

سيكون من الواضح أن العلاقات واحد إلى واحد ليست مفيدة جداً عند مصمم قاعدة البيانات، ببساطة يمكن دمج كلا الجدولين في جدول واحد. مع ذلك، هناك بعض الحالات التي تساعد فيها علاقة واحد إلى واحد في تحسين الأداء. على سبيل المثال: إذا كان جدول قاعدة بيانات يحتوي على عدد قليل من أعمدة البيانات التي يكثر استخدامها والأعمدة المتبقية التي يتم استخدامها بشكل غير منتظم، ومصمم قاعدة البيانات قد يقسم جدولاً واحداً إلى جدولين مرتبطين من خلال علاقة واحد إلى واحد. مثل هذا التصميم يقلل من النفقات العامة اللازمة لاسترداد الأعمدة المستخدمة نادراً عندما يتم تنفيذ الاستعلام على محتويات جدول قاعدة البيانات. مثال على ذلك البطاقة الشخصية تحتوي البيانات التعريفية بالاضافة الى الصورة الشخصية والبصمة، في هذه

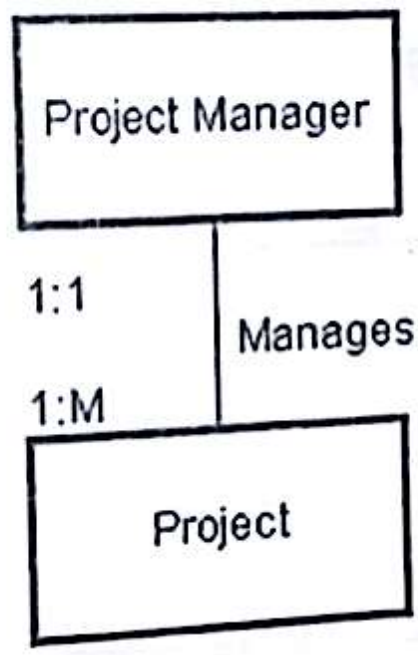
Scanned by CamScanner

الحالة يتم إنشاء جدولين الأول يحوي المعلومات التعريفية والجدول الثاني يحوي الصورة والبصمة ويرتبط بعضهما ببعض بعلاقة واحد إلى واحد .

3.3.2. علاقة واحد لمتعدد One-to-Many Relationships

كل سجل من الجدول الأول يرتبط بمجموعة سجلات من الجدول الثاني وكل سجل من الجدول الثاني يرتبط بسجل واحد فقط من الجدول الأول كما في الشكل 2. هذه العلاقة هي مماثلة لتلك التي بينك و بين أحد الوالدين. لديك أم واحدة فقط، ولكن والدتك قد يكون لديها عدة أولاد. أو كما في حال جدول الدول مع جدول المدن كل دولة لها مجموعة مدن وكل مدينة لها دولة واحدة فقط .

ONE-TO-MANY RELATIONSHIP



الأب ← أولاد
الأب له عدة أولاد
ولكن الأولاد لهم
أب واحد

مدينة ← محافظات

الشكل 2: علاقة رأس باطراف.

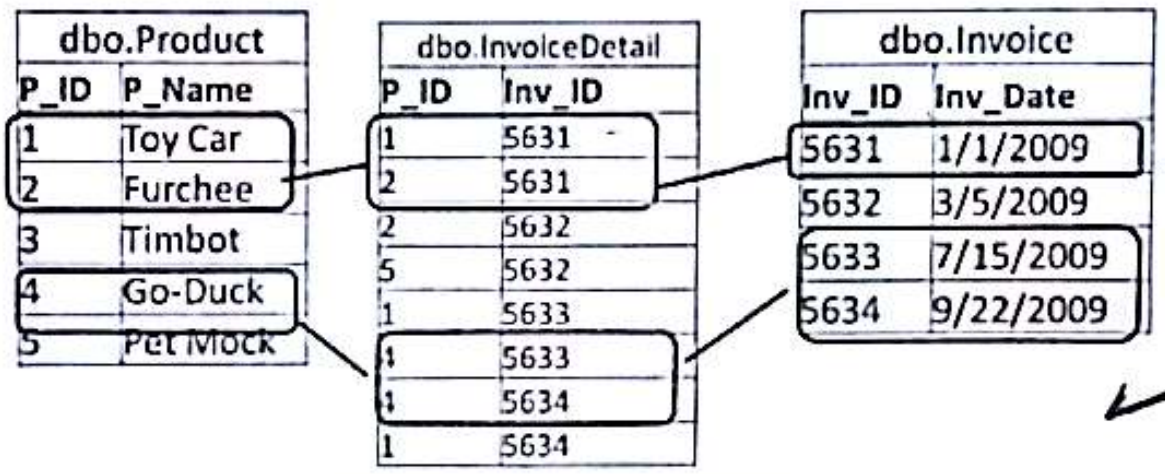
Scanned by CamScanner

3.3.3. علاقة متعدد لمتعدد

Many-to-Many Relationships

كل سجل من الجدول الأول يرتبط بمجموعة سجلات من الجدول الثاني وكل سجل من الجدول الثاني يرتبط بمجموعة سجلات من الجدول الأول كما في الشكل 3. مثال على ذلك جدول الكتب وجدول المؤلفين فكل كتاب قد يكون له مجموعة مؤلفين و كل مؤلف له مجموعة كتب .

إن تمثيل علاقة متعدد لمتعدد بين جدولين تتم عبر كسر العلاقة بجدول ثالث يحوي المفاتيح الأساسية من الجدولين فقط ويسمى الجدول الوسيط، في حال المثال السابق جدول الكتب وجدول المؤلفين ننشئ جدولاً ثالثاً يربط بينهم وليكن جدول كتب المؤلفين.



Many-to-Many Relationships

الشكل 3: علاقة أطراف بأطراف.

4. المفاهيم الأساسية في نموذج الكينونات العلائقية

(Principle of Entity Relationship Model, ER Model)

إن نموذج الكينونات العلائقية هو وصف للبيانات في البيئة قيد الدراسة والعلاقة بينها. حتى يتم توضيح كيفية تصميم هذا النموذج لا بدُ من توضيح المفاهيم الأساسية في هذا التصميم وهي مفهوم الكينونة ونوع الكينونة وأنواع العلاقات بين الكينونات وكذلك الواصفات (الصفات) المختلفة لهذه الكينونات.

Scanned by CamScanner

4.1. الكينونة (Entity)

تُعرَّف الكينونة أنها شيء(ما) او شخص او مكان أو حدث أو مفهوم في البيئة التي يتم دراستها. مثال على ذلك: شخص معين داخل النظام، أو قسم معين أو سلعة معينة أو طالب معين.

الشكل 4: اسم الكينونة.

4.2. نوع الكينونه (Entity Type)

هي عبارة عن مجموعات كينونية لها الخصائص أو الصفات نفسها فمثلاً:

- جميع الطلاب الذين يدرسون في الجامعة
- جميع المرضى الذين لهم سجلات داخل مستشفى
- جميع الموظفين الذين يعملون في مؤسسة.

يرمز نوع الكينونة داخل نموذج الكينونات العلائقية كما في الشكل 4.

نوع الكينونة قوية ونوع الكينونة ضعيفة

(Strong Entity and Weak Entity Type)

يمكن تصنيف الكينونات داخل مؤسسة معينة على أنها من نوع كينونة قوية أو نوع كينونة ضعيفة. نقصد هنا بقول كينونة قوية هي تلك الكينونات التي لا يعتمد

Scanned by CamScanner

وجودها على كينونة أخرى. أما الكينونات الضعيفة فهي التي يعتمد وجودها على كينونات أخرى.

مثال: في بيئة نظام المحاسبة داخل الجامعة نرى أن الموظفين داخل النظام لا يعتمد وجودهم على كينونات أخرى. أما إذا أخذنا أبناء الموظفين في نظام المحاسبة فيعتمد وجودهم على وجود الآباء (الموظفين).

فوجود الأبناء يعتمد على وجود الآباء (الموظفين) داخل النظام أي إنه لا يمكن أن يتواجد ابن إلا إذا تواجد الأب (الموظف). تُسمى في هذه الحالة كينونة الأبناء بكينونة ضعيفة ويرمز لها كما في الشكل 5.

الشكل 5: الكينونة الضعيفة.

4.3. الواصفات (الصفات)

4.3.1. مفهوم الواصفة (Attribute)

تُمثل الواصفات الخصائص المختلفة للكينونه.

مثال:

نوع الكينونة	الواصفات
الطالب	رقم الطالب، اسمه، العنوان، التخصص
الموظف	رقم الموظف، الإسم، الراتب، الدرجة

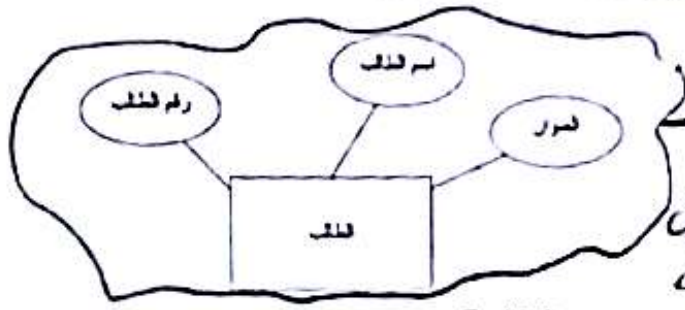
ويرمز لها كما في الشكل 6.

Scanned by CamScanner

الشكل 6: الواصفة أو الخاصية.

مثال:

الشكل 7 يوضح واصفات الكينونة الطالب.



الشكل 7: واصفات كينونة الطالب.

هناك عدة أنواع من الواصفات:

1. الواصفة البسيطة (Simple Attribute)
2. الواصفة المركبة (Composite Attribute)
3. الواصفة ذات القيمة الواحدة (Single Value Attribute)
4. الواصفة ذات القيم المتعددة (Multi Values Attribute)
5. الواصفة المشتقة (Derived Attribute)

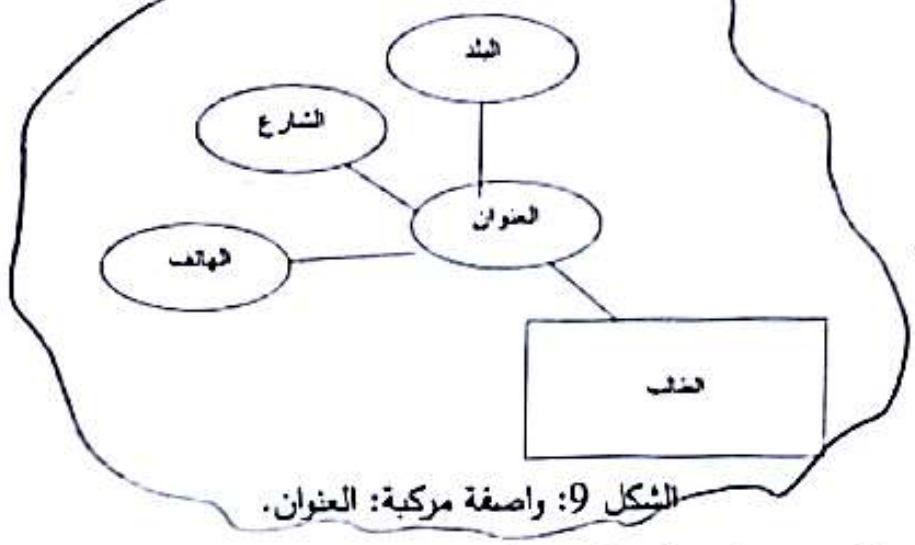
1. الواصفة البسيطة: نقصد هنا بالواصفة البسيطة أي إن الواصفة لا يمكن تجزئتها إلى أجزاء. مثال: كينونة الطالب فإن صفة العمر للطالب لا تُجزأ إلى أجزاء. يمكن تمثيل هذه الواصفة كما هو موضح في الشكل 8.

الشكل 8: واصفة بسيطة.

Scanned by CamScanner

الواصفة المركبة: الواصفة المركبة هي الواصفة التي يمكن تقسيمها إلى أجزاء أخرى ذات دلالة. مثال: الصفة عنوان الطالب يمكن تقسيمها إلى أجزاء أخرى مثل المدينة، الشارع، العنوان البريدي و تمثل كما في الشكل 9.

2. الواصفة ذات القيمة الواحدة: نقصد هنا أن الواصفة تأخذ قيمة واحدة لتلك الكينونة ويمكن أن تدعى بالواصفة فردية القيمة. مثال: في كينونة الطالب صفة العمر تأخذ قيمة واحدة فقط ولا يمكن أن تأخذ أكثر من



الشكل 9: واصفة مركبة: العنوان.

قيمة. نلاحظ أن العمر يمكن أن يكون ذات قيمة واحدة وكذلك بسيطة كما ذكرنا سابقاً.
الشكل 10 يوضح واصفة ذات قيمة واحدة.

الشكل 10: واصفة ذات قيمة واحدة وهي بسيطة.

3. الواصفات ذات القيم المتعددة: هي الواصفات التي تأخذ أكثر من قيمة لنفس الكينونة ويرمز لها كما في الشكل 11.

الشكل 11: شكل الواصفة ذات قيم متعددة.

Scanned by CamScanner

مثال: عنوان البريد الالكتروني لطالب حيث أنه يمكن أن يكون للطالب أكثر من بريد الكتروني و يمكن تمثيل ذلك كما في الشكل 12.

الشكل12: واصفة متعددة القيم: العنوان الإلكتروني.

4. الواصفة المشتقة: وهي الواصفة التي تُحدد من خلال واصفات أخرى.

يرمز للواصفة المشتقة كما هم موضح في الشكل 13.

الشكل 13: واصفة مشتقة.

مثال: صفة العمر لطالب معين يمكن أن تُحسب من خلال معرفة تاريخ اليوم وتاريخ ميلاد ذلك الطالب. يمكن تمثيل ذلك كما هو موضح في الشكل 14.

الشكل 14: مثال واصفة مشتقة: العمر.

Scanned by CamScanner

4.3.2. القيود ومجالات الواصفات

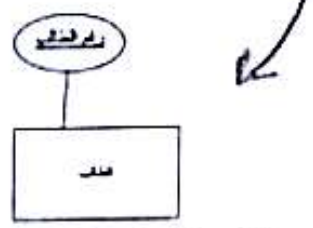
(Domain and Constraints of Attributes)

نقصد بمجال الواصفة مجموعة القيم التي يمكن أن تحدد تلك الواصفة. مثال: لو أخذنا كينونة الطالب وله واصفات رقم الطالب، اسم الطالب، عمر الطالب. يمكن تحديد المجال لكل واصفة كما يأتي:

- واصفة رقم الطالب: يجب ألّا يزيد عن 8 خانات رقمية.
- واصفة اسم الطالب: مكون من 30 حرف، 30 أقصى حد.
- واصفة عمر الطالب: يجب ألّا يقل عن 18 سنة أو لا يزيد على 35 سنة.

أما بالنسبة لخاصية المفتاح لواصفة معينة فإن لكل نوع كينونة هنالك صفة مميزة. نقصد هنا بالواصفة المميزة تلك التي لها قيم لا تتكرر مع الكينونات الأخرى والمكونة لنوع الكينونة وهذه الواصفة تدعى المفتاح.

مثال1: في نوع كينونة الطالب نلاحظ أنه لكل طالب هنالك رقم خاص به لا يتكرر لطالب آخر ولذلك يدعى رقم الطالب بالمفتاح لنوع كينونة الطالب وتمثل كما في الشكل 15.



الشكل 15: مثال واصفة مميزة: رقم الطالب.

مثال2: في نوع كينونة الموظف في شركة معينة نلاحظ أن رقم هوية الموظف لا يتكرر لموظف أخر ولذلك يدعى رقم هوية الموظف بالمفتاح لنوع كينونة الطالب.

مثال3: في نوع كينونة القسم في جامعة ما نلاحظ أن كل قسم له رقم يميزه من قسم آخر ولذلك يدعى رقم القسم بالمفتاح للقسم.

Note: في بعض الأحيان يتم تكوين المفتاح من مجموعة خصائص لنوع كينونة. مثال: في نوع كينونة مشروع نلاحظ أن هنالك خصائص مختلفة مثل اسم المشروع، رقم المشروع، مكان المشروع، القسم المشرف على المشروع. هنا يمكن القول إن اسم

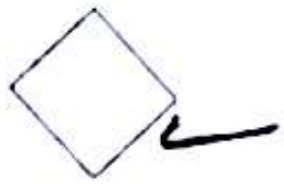
Scanned by CamScanner

يعني إما اسم المشروع و رقم المشروع كلاهما مفتاح؟
أو واحد منهما مفتاح...

المشروع و/أو رقم المشروع هي خصائص مفتاح المشروع لأنه لا يمكن أن يكون هنالك أكثر من مشروع بنفس الاسم وكذلك رقم المشروع .

4.4. أنواع العلاقات (Relationship Type)

نقصد بنوع علاقة الرابط هي مجموعة الارتباطات بين كينونات مختلفة ونستخدم الرمز الموضح في الشكل 16 للتعبير عن علاقة.



الشكل 16: رمز العلاقة (الارتباط).

يوجد عدة أنواع للعلاقات:

1. علاقة واحد . إلى . واحد

(One-to –One Relationship)

2. علاقة واحد . إلى . متعدد

(One-to-Many Relationship)

3. علاقة متعدد . إلى . متعدد

(Many-to-Many Relationship)

4. العلاقة التعاودية

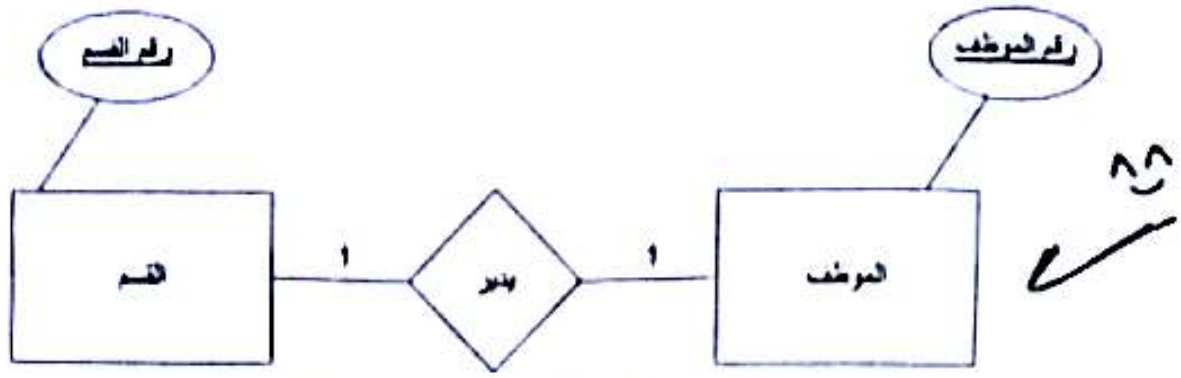
(Recursive Relationship)

علاقة واحد . إلى . واحد

خلال هذه العلاقة يتم ربط كل عنصر في كينونة مع عنصر واحد في كينونة أخرى.

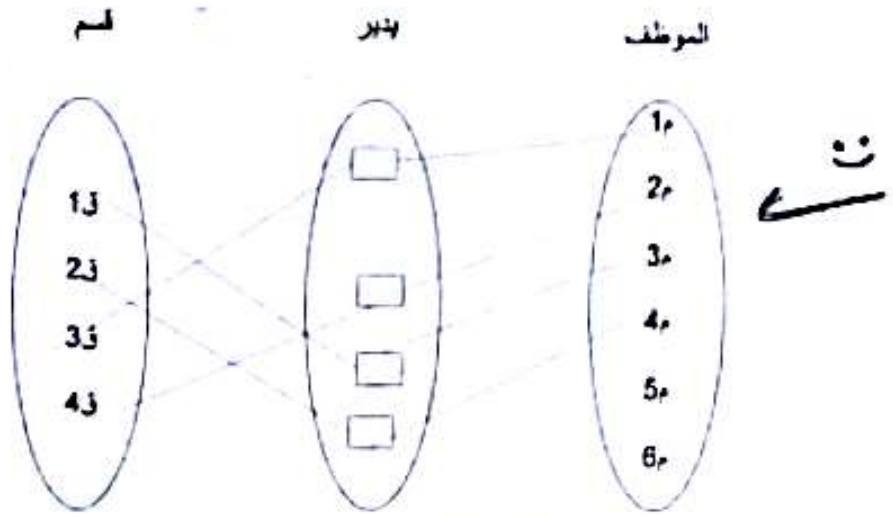
مثال: في شركة ما هنالك مجموعة من الأقسام وكل قسم يُديره موظف واحد. ولا يجوز أن يدير أكثر من موظف أي قسم في الشركة. هنا نحدد الكينونتين؛ الموظف والقسم والعلاقة بينهما (يدير) وفق المخطط الكينوني الموضح في الشكل 17.

Scanned by CamScanner



الشكل 17: موظف واحد يدير قسم واحد.

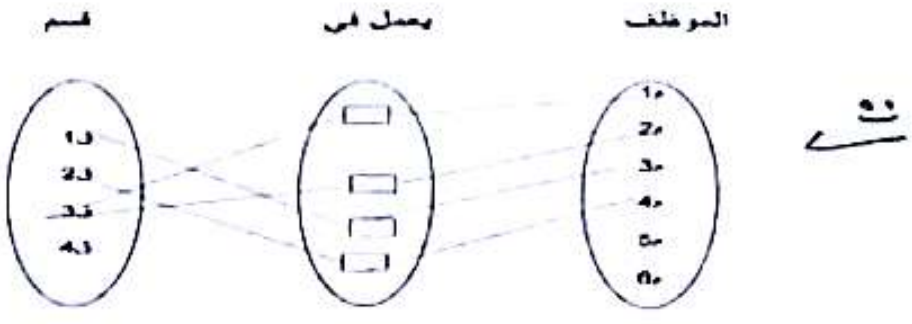
ويمكن توضيح المفهوم بشكل رياضي كما هو موضح في الشكل 18.



الشكل 18: موظف يدير قسم.

علاقة واحد . إلى . متعدد

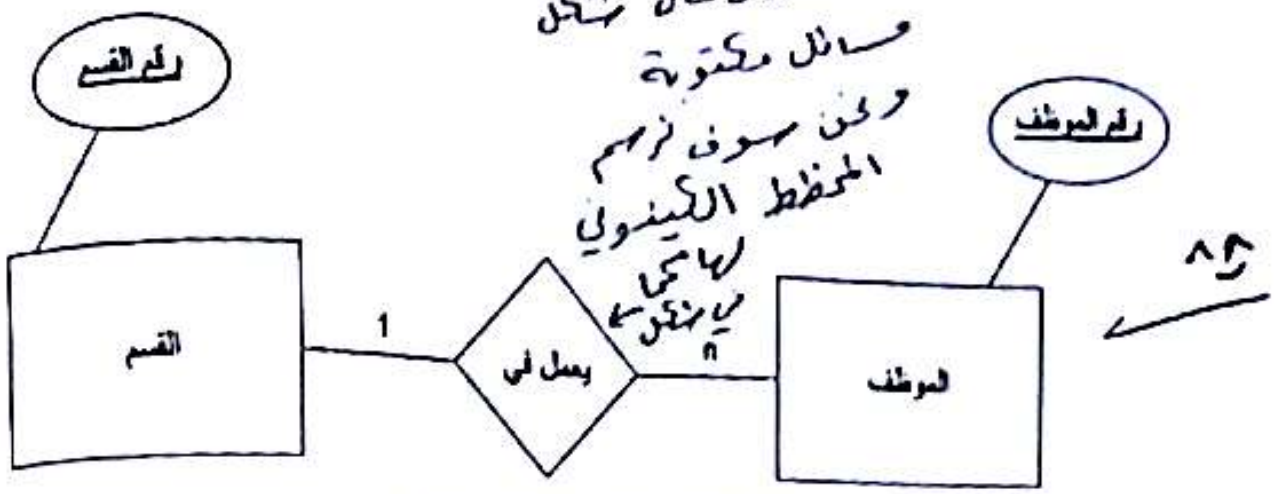
في هذا النوع من العلاقات لدينا أكثر من عنصر في كينونة يرتبط مع عنصر واحد من كينونة أخرى. مثال: في شركة ما هناك أقسام. يعمل في كل قسم مجموعة من الموظفين، وكل موظف يمكن أن يعمل في قسم واحد كما في الشكل 19.



الشكل19: موظف يعمل في قسم.

المخطط الذي يمثل هذه العلاقة هو كما في الشكل 20.

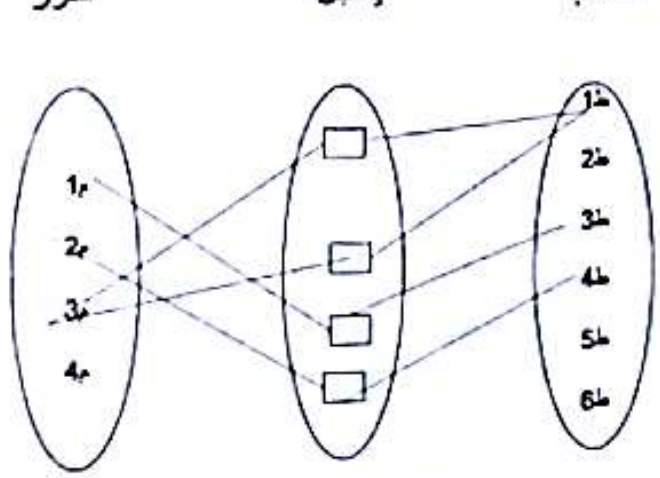
Scanned by CamScanner



الشكل 20: علاقة واحد إلى متعدد.

علاقة متعدد إلى . متعدد

كل عنصر في الكينونة الأولى مرتبطة بأكثر من عنصر في كينونة ثانية. كذلك كل عنصر في الكينونة الثانية مرتبط بأكثر من عنصر في الكينونة الأولى كما في الشكل 21.

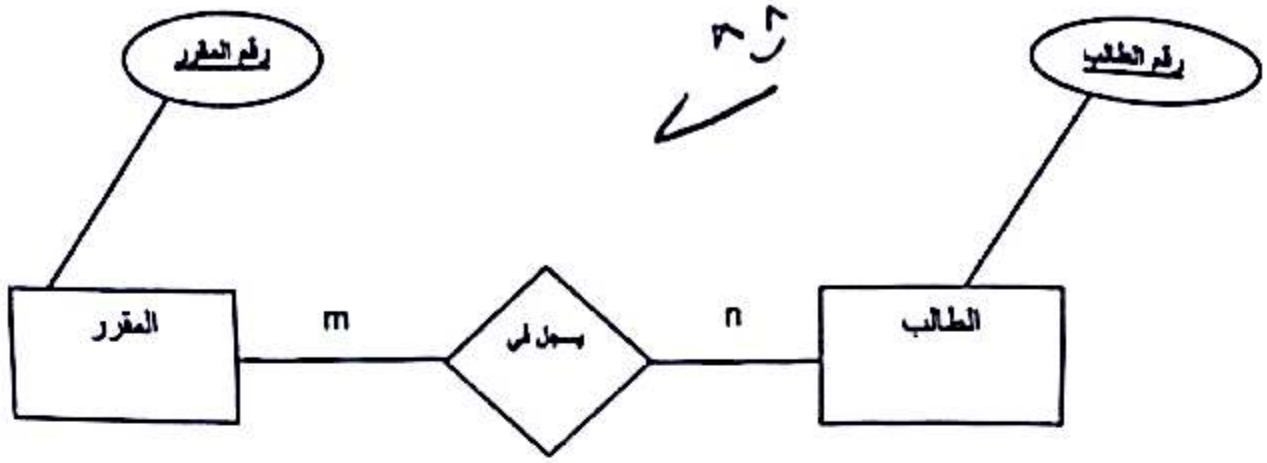


الشكل 21: علاقة متعدد الى متعدد.

مثال: في نظام التسجيل في الجامعة كل طالب يسجل أكثر من مقرر، و المقرر الواحد يسجل فيه أكثر من طالب.

Scanned by CamScanner

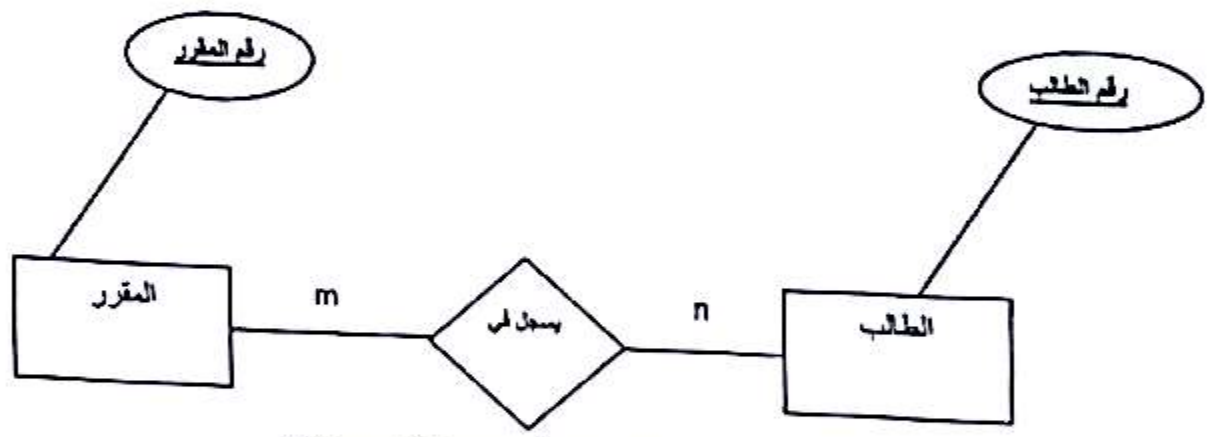
تُمثل العلاقة كما في الشكل 22.



الشكل 22: علاقة متعدد الى متعدد.

ملاحظة:

يمكن أن يكون لدينا نوع الكينونة نفسها من أجل أنواع ارتباطات مختلفة كما في الشكل 23.

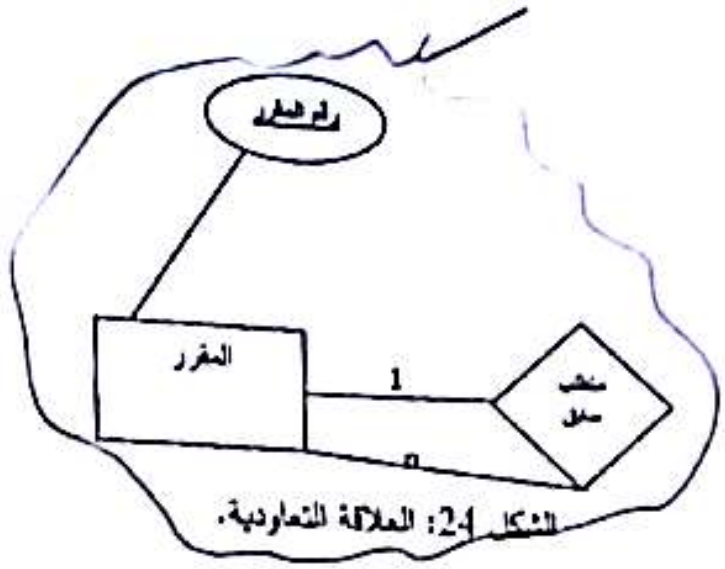


الشكل 23: نفس نوع الكينونة لأنواع ارتباطات مختلفة.

العلاقة التعاودية

كل عنصر في كينونة يمكن أن يرتبط مع عنصر من الكينونة نفسها. مثال: في نظام التسجيل في الجامعة المقرر يمكن أن يكون له مقرر آخر كمتطلب سابق كما في الشكل 24.

Scanned by CamScanner



الشكل 24: العلاقة التعاودية.

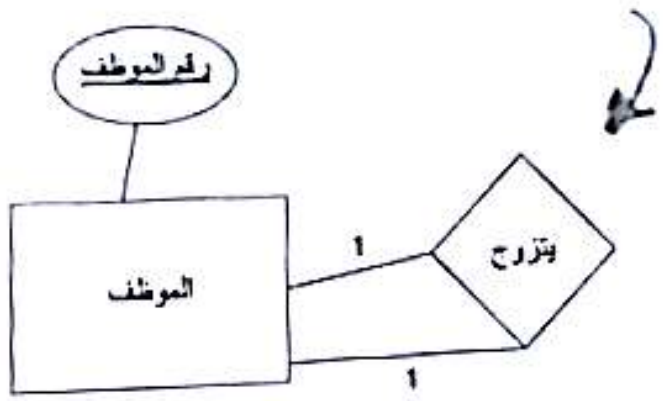
درجة العلاقة (Relation Degree)

تعبر درجة العلاقة عن عدد الكينونات التي ترتبط بعضها ببعض من خلال علاقة واحدة. يوجد هنالك عدة درجات شائعة في نموذج الكينونات العلائقية منها الأحادية (من درجة واحدة) ومنها الثنائية (من درجة 2) و منها الثلاثية (من درجة 3). على الرغم من وجود درجات علائقية أخرى إلا أنها لا تُستخدم بشكل كبير في الواقع العملي.

4.4.1. درجة العلاقة الأحادية

(Unary Relationship)

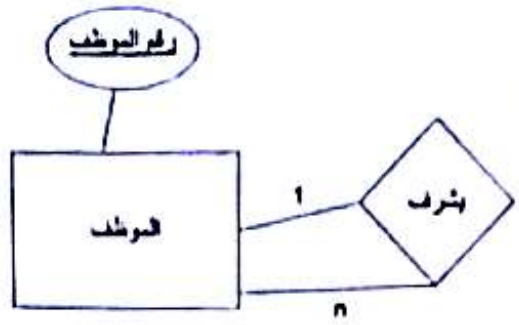
مثال1: موظف في شركة ما يمكن أن يتزوج من موظفة في تلك الشركة كما في الشكل 25.



الشكل 25: مثال 1 على درجة العلاقة الأحادية.

مثال2: كل موظف يشرف على مجموعة من الموظفين.كما في الشكل 26.

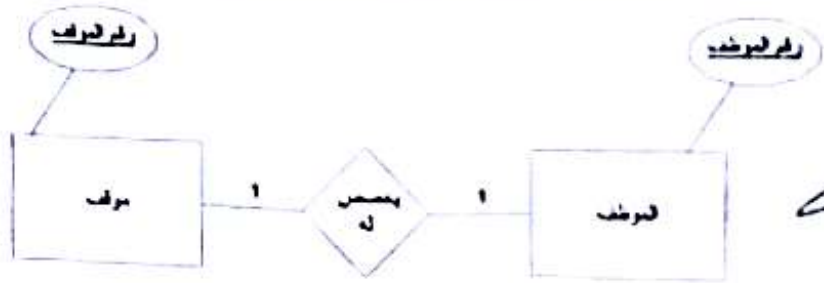
Scanned by CamScanner



الشكل 26: مثال 2 على درجة العلاقة الأحادية.

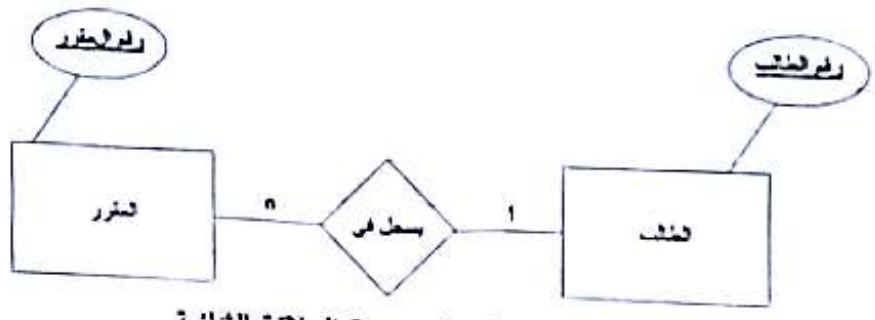
4.4.2. درجة العلاقة الثنائية

مثال1: يُحدد لكل موظف في شركة ما موقف واحد خاص لسيارته كما في الشكل 27.



الشكل 27: مثال 1 على درجة العلاقة الثنائية.

مثال2: كل طالب يسجل في أكثر من مقرر كما هو موضح في الشكل 28.

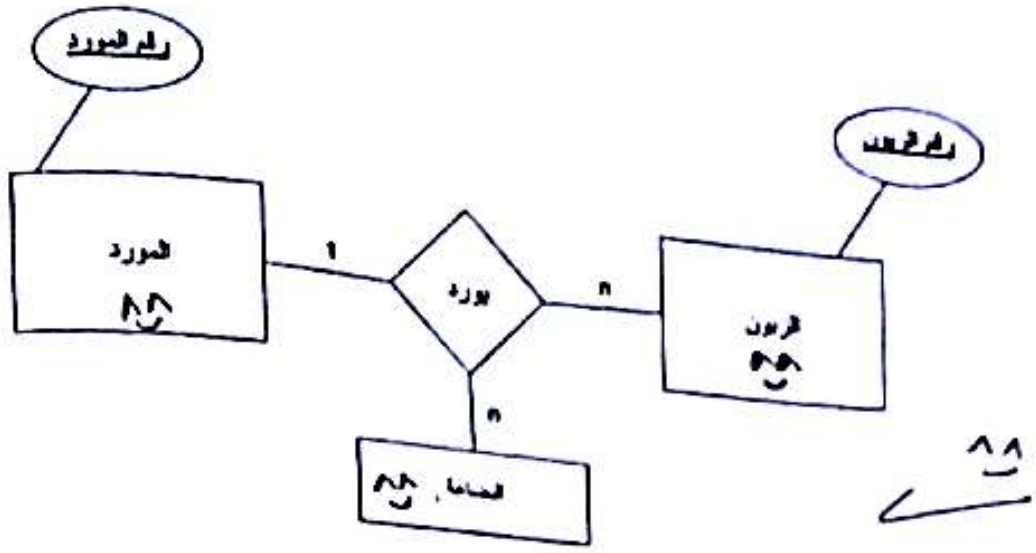


الشكل 28: مثال 2 على درجة العلاقة الثنائية.

4.4.3. درجة العلاقة الثلاثية

مثال: المورد يورد بضاعة إلى زبائنه كما هو موضح في الشكل 29.

Scanned by CamScanner



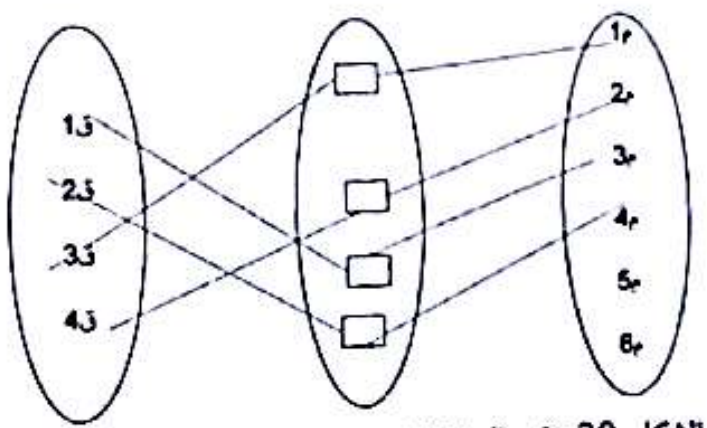
الشكل 29: درجة العلاقة الثلاثية.

4.5. قيد المشاركة (Participation Constraint)

يتم الحديث عن هذا القيد في حال وجود كينونة مرتبطة بكينونة أخرى. يوجد نوعان من قيود المشاركة وهي قيد المشاركة الكلي وقيد المشاركة الجزئي.

4.5.1. قيد المشاركة الكلي (Total Participation Constraint)

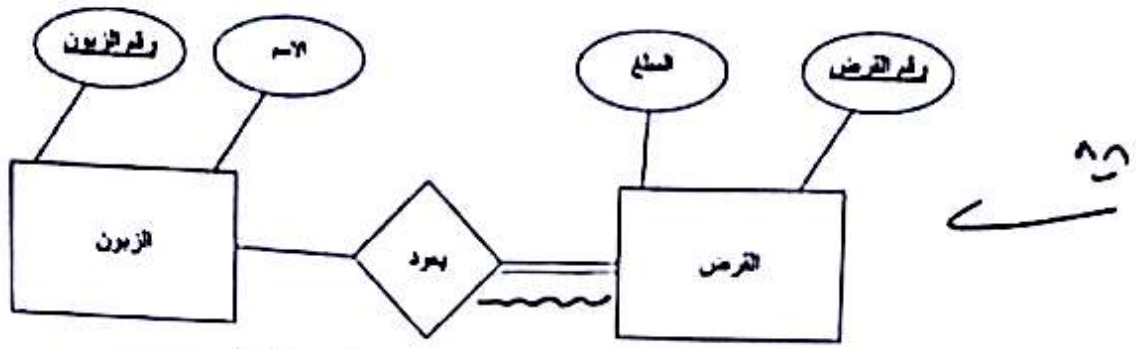
في هذه الحالة كل كينونة يجب أن ترتبط بوحدة (.....) في العلاقة. يتم تمثيل هذا القيد في مخطط الكينونات عن طريق رسم خط مزدوج يربط الكينونات بهذه العلاقة. مثال: في بنك ما إذا فرضنا أن كل قرض يجب أن يعود على الأقل إلى زبون واحد. يمكن توضيح ذلك كما في الشكل 30.



الشكل 30: قيد المشاركة الكلي قرض(ق)، زبون(م).

Scanned by CamScanner

في هذه الحالة نلاحظ أن كل قرض يجب أن يشارك في علاقة يعود مع الزبون وهذه مشاركة كلية. وفي مخطط الكينونات تُمثل كما في الشكل 31.



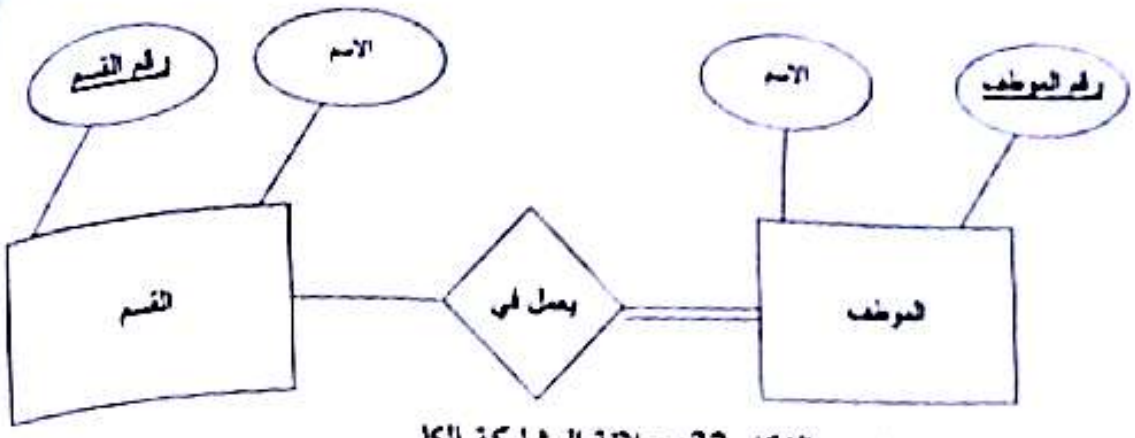
الشكل 31: قيد المشاركة الكلي(يعود لي).

3.1.1.قيد المشاركة الجزئي (Partial Participation)

في هذه الحالة فإن بعض الكينونات ترتبط بوحده (Instance) في العلاقة. يتم تمثيلها برسم خط مفرد يربط الكينونات المرتبطة بهذه العلاقة. أي إن بعض الكينونات لا تشارك في العلاقة. في المثال السابق الزبون ليس شرطاً أن يكون مُقترضاً. وبالتالي فإن بعض الزبائن لا يرتبطون بعلاقة (يعود)، ولهذا مُثلت بخط مفرد.

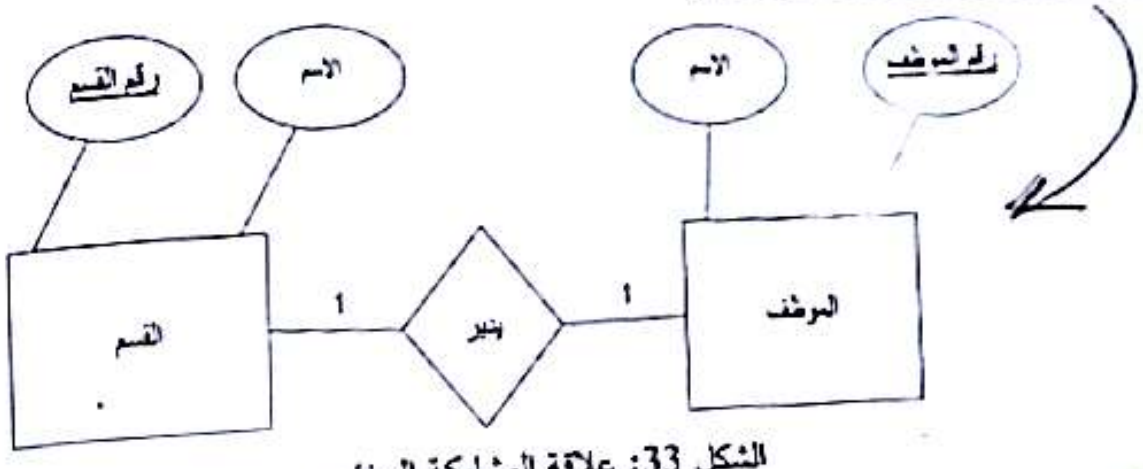
مثال آخر: في شركة ما تضم مجموعة من الموظفين، إذا فرضنا أن كل موظف يجب أن يعمل في قسم واحد فقط، والعلاقة بين الموظف والقسم هي علاقة يعمل في. إذا فرضنا أيضاً أن كل موظف يعمل في قسم. أي لا يمكن أن يكون هنالك موظف لا يعمل في قسم ما. و بالتالي فإن كينونة الموظف ترتبط في علاقة يعمل بعلاقة مشاركة كلية.

Scanned by CamScanner



الشكل 32: علاقة المشاركة الكلي.

و عليه فإن التمثيل يتم بخط مزدوج كما في الشكل 32. أما إذا قلنا إن كل قسم يشيره موظف ما فإن بعض أو جزء من مجموعة كينونات الموظفين ترتبط بكينونة القسم عن طريق العلاقة يمتلك و ليس الكل كما في الشكل 33.



الشكل 33: علاقة المشاركة الجزئي.

مثال: ارسم مخطط الكينونات (ER-Diagram) والمتعلق بقاعدة البيانات لبنك ما والذي متطلباته كما يأتي:

- يوجد للبنك زبائن ويحتفظ البنك بمعلومات عن هؤلاء الزبائن وتشمل رقم الهوية وهو رقم مفرد واسم الزبون وعنوان الزبون.
- يمكن للزبون أن يملك أكثر من حساب، حيث تشمل معلومات الحساب رقم الحساب وهو مفرد والرصيد ونوع الحساب.
- يمكن للزبون أن يحصل على أكثر من قرض وتشمل معلومات القرض رقم القرض وهو مفرد ونوع القرض وقيمة القرض.
- يمكن لأكثر من زبون أن يشتركوا في قرض واحد.
- لا يُشترط أن يكون للزبون حساب في البنك للحصول على قرض.
- لا يُشترط لفتح حساب بالبنك أن يحصل الزبون على أي قرض.

Scanned by CamScanner

لنتحدث عن كيفية عمل مخطط الكينونات كما يأتي:

أولاً: نحدد الكينونات.

1. الزبون (رقم الهوية، الاسم، العنوان)

2. الحساب (الرقم، الرصيد، النوع)

3. القرض (الرقم، النوع، القيمة)

ثانياً: نحدد العلاقة بين الكينونات.

1. يمتلك (الزبون يكون له أكثر من حساب)

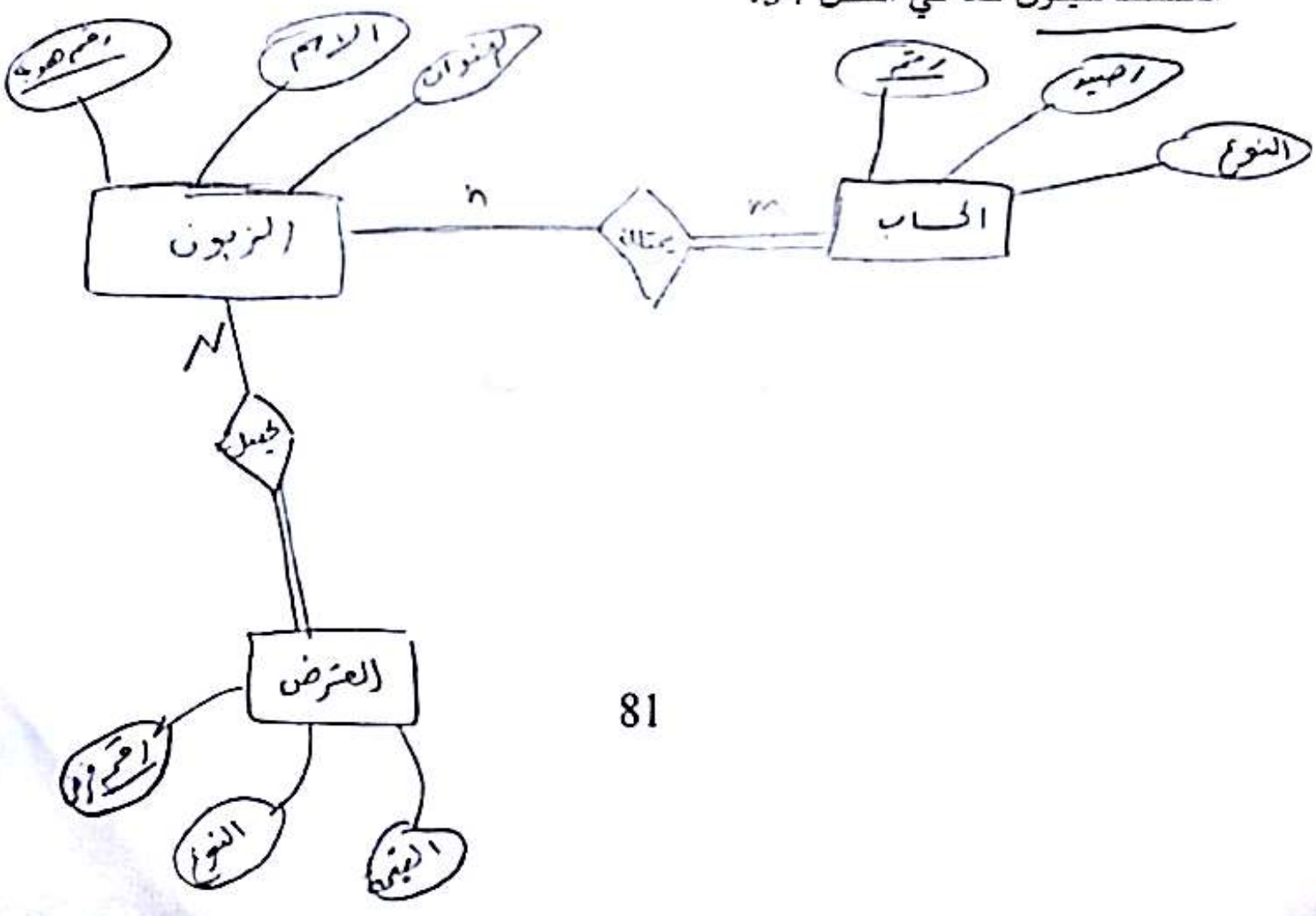
2. يحصل (الزبون يحصل على قرض)

ثالثاً: نحدد القيود.

1. لا يُشترط للحصول على قرض أن يكون الزبون له حساب بالبنك.

2. لا يُشترط لفتح حساب في البنك الحصول على أي قرض.

من القيد الأول نلاحظ أن كل حساب بالبنك لا بد أن يكون له زبون، وهذا يعني أن كل حساب يجب أن يمتلكه زبون، وبالتالي فإن مشاركة كينونة الحساب هي مشاركة كلية في العلاقة. كذلك بالنسبة للقيد الثاني فإن كينونة القرض هي مشاركة كلية في العلاقة يحصل لأنه لا يمكن أن يكون هنالك قرض بدون زبون. بناءً على ما سبق نجد أن المخطط سيكون كما في الشكل 34.



Scanned by CamScanner