الأكاديمية العربية الدولية

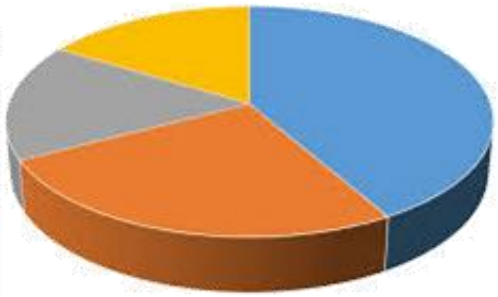
الأكاديمية العربية الدولية

المقررات الجامعية

www.aiacademy.info | care@aiacademy.info

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة الأولى

1. مقدمة Introduction

1.1. مفهوم النظام System concept:

- **النظام** هو مجموعة متفاعلة من العناصر بينها مبادئ عامة حاكمة بغض النظر عن طبيعتها والعلاقات الحاكمة بينها.
- وفي تعريف آخر:

النظام هو العملية التي تنفذها مجموعة من العناصر المتحدة في الوظيفة والتشغيل لتحقيق أهداف معينة.

- وفي تعريف آخر:

النظام هو مجموعة من الأهداف المرتبطة بعلاقات منظمة لتنفيذ وظيفة معينة.

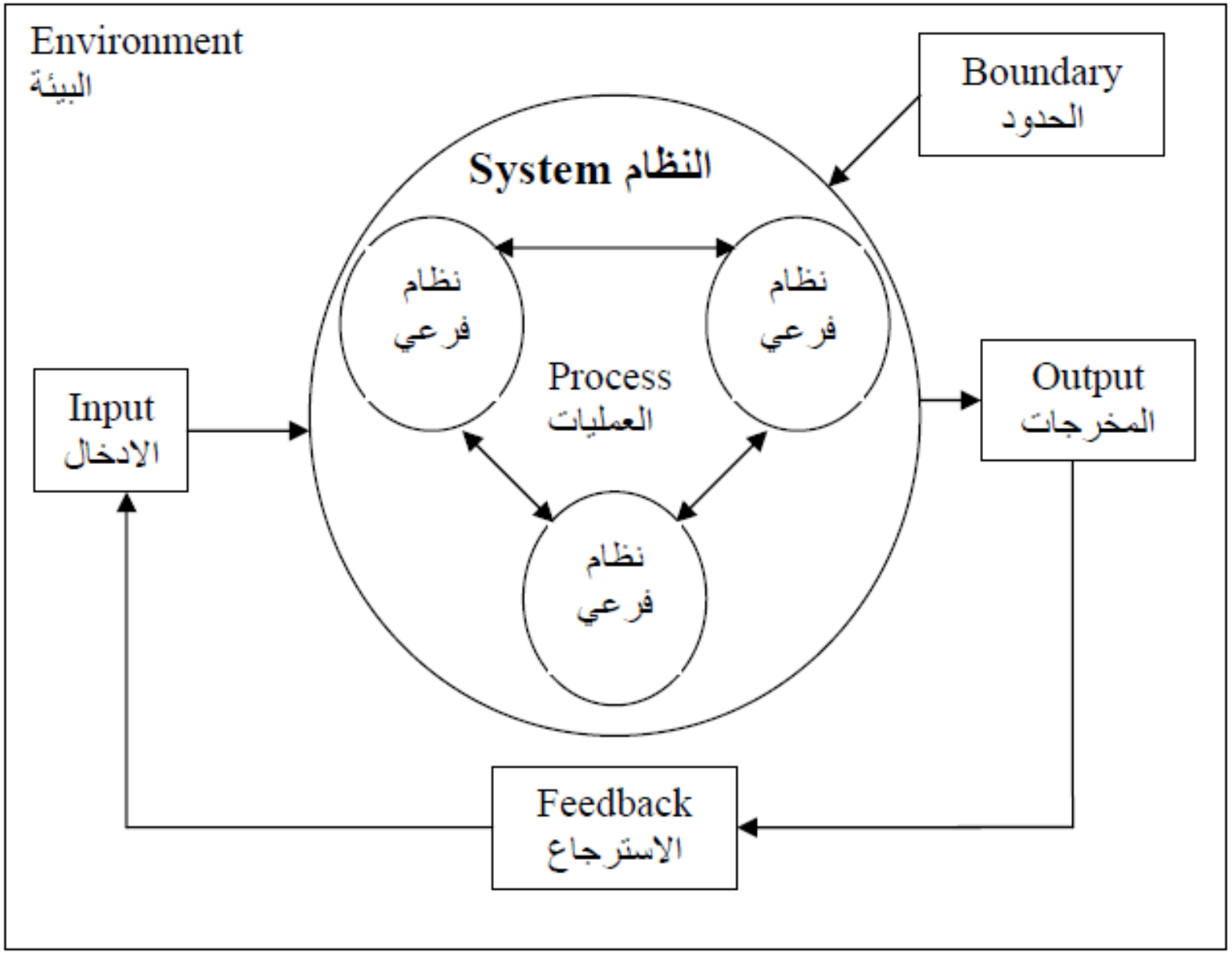
- ومن الممكن أن يتكون النظام من مجموعة من الأنظمة الفرعية الأصغر من حيث المهام والوظائف التي تم توزيعها على مجموعة من الأنظمة الفرعية لتكون فيما بينها نظام أشمل ليحقق هدف أكبر.

- وكما هو موضح بالشكل 1 فإن المكونات الرئيسية لنظام هي:

البيئة Environment، الإدخال Input، العمليات Processes، النظام الفرعي Subsystem،

العلاقات Relationships، الإخراج Output، الحدود Boundary و الاسترجاع Feedback.

- في الجزء التالي سنأتي لشرح كل جزء من هذه الأجزاء.



الشكل 1: النظام System.

2.1. خصائص النظام System characteristics:

يتكون النظام من العناصر التالية:

- **البيئة Environment:**

- بيئة النظام هي مجموعة من عناصر ومكونات بينها علاقات والتي لا تكون عناصرها جزءاً من النظام ولكن أي تغيير في أي عنصر من عناصرها ينتج عنه تغيير في حالة النظام.
- وبذلك فإن بيئة النظام تتألف من جميع المتغيرات التي تؤثر على الحالة, فهي التي تحتوي النظم الفرعية التي ليست جزءاً من النظام ولكنها تتأثر به وتؤثر فيه.

- **الحدود Boundary:**

- هي نطاق العمل المطلوب مثلاً قد تحوي المنظومة الجديدة على معلومات عن الزبون والمبيعات ولكن دون أن تشمل على المخزون.

2.1.خصائص النظام System characteristics....

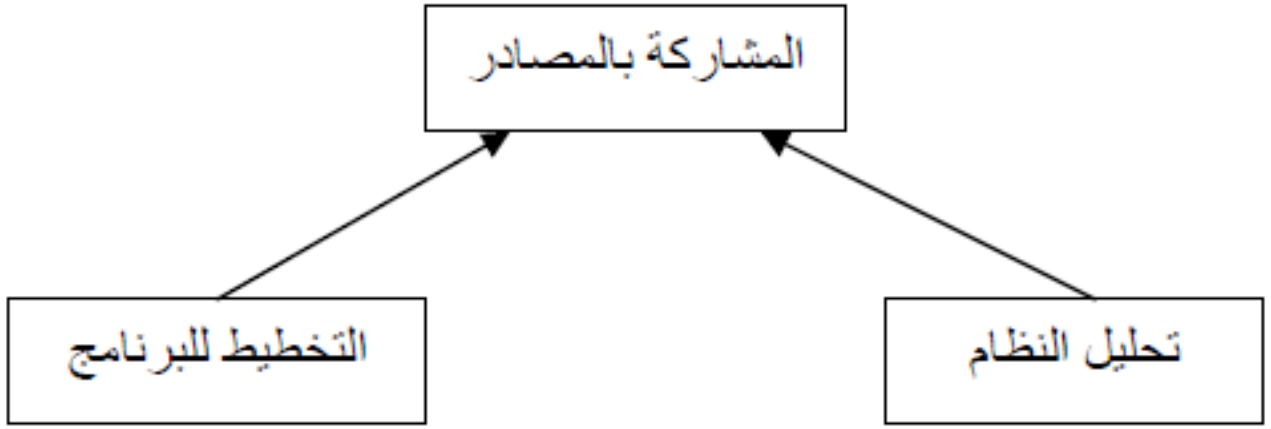
- **الإدخال Input:**
- هي مجموعة من المتطلبات (بيانات) والأوامر التي يتم إدخالها من خارج النظام و يحتاجها لتنفيذ العمليات داخل النظام.
- وتتم عمليات الإدخال إما بواسطة مستخدم النظام أو نظام فرعي خارج النظام.
- **العمليات Processes:**
- هي مجموعة من الاجراءات التي تتحكم فيها علاقات محددة لتنفيذ مجموعة مهام والتي تحقق الهدف العام للنظام.
- هذه المهام أو الوظائف التي يقوم بتنفيذها النظام لإنجاز هدف محدد يعتمد في التنفيذ على المدخلات (بيانات) لمعالجتها تحت عملية تحكم معينة لكي يضمن الدقة في الوصول إلى المعلومات التي سيستفاد منها في اتخاذ القرار السليم وتنظيم العمل.
- **النظام الفرعي Subsystem:**
- هو عبارة عن نظام ويحمل مواصفات النظام العادي لكنه ينفذ مهمة جزء من عملية كاملة لنظام شامل أو أكبر.

2.1.خصائص النظام System characteristics....

- **العلاقات Relationships:**
- هي الصلات التي تربط بين النظم الفرعية للنظام أو البيئة حيث لا يمكن لأي نظام فرعي أن يؤدي وظيفته بمعزل عن غيره لأنه لابد من اعتماده على بيانات الواردة إليه من نظم فرعية أخرى حتى يستطيع تأدية وظيفته, كما أن ما ينتجه من معلومات هي الأخرى تعتبر مدخلات أي بيانات لنظم فرعية أخرى.
- وللعلاقات أنواع مختلفة تحدد المراد منها وهي:

أ) العلاقات المتبادلة المجمعة:

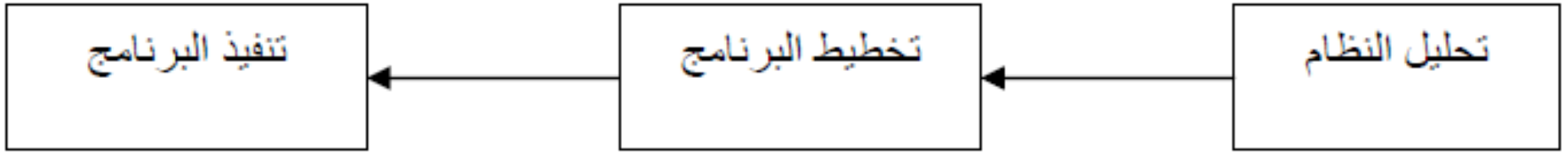
ويراد بها استخدام أفراد أو إدارات النظم الفرعية لمجموعة من المصادر المشتركة.



شكل 2: العلاقات المتبادلة المجمعة.

ب) العلاقات المتبادلة المتعاقبة:

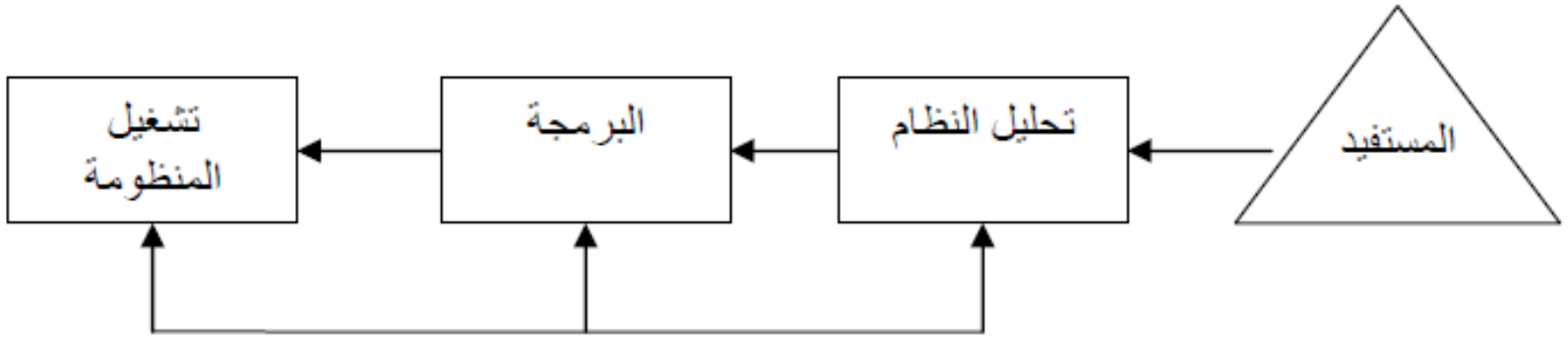
وتعني أن مخرجات أحد النظم هي مدخلات لنظم أخرى.



شكل 3: العلاقات المتبادلة المتعاقبة.

ت) العلاقات المتبادلة التبادلية:

حيث أن تنفيذ عمل نظام فرعي يؤثر بالتبادل على عمل نظام آخر.



شكل 4: العلاقات المتبادلة التبادلية.

2.1.خصائص النظام System characteristics....

- **المخرجات Output:**
- هي المعلومات التي تم الحصول عليها من المنظومة بعد معالجة المدخلات، وقد تأخذ هذه المخرجات شكل تقارير أو استعلامات.
- **الاسترجاع Feedback:**
- بعض الأنظمة تتمتع بعامل التحكم وذلك عن طريق التغذية الاسترجاعية والتي تتم بإعادة ادخال المخرجات كمدخلات جديدة لإعادة المعالجة وإنتاج مخرجات جديدة، لغرض تحسين أو تغيير المخرجات حسب قياسات معينة.
- في تحليل النظم، يتم الاستفادة من التغذية الاسترجاعية لتوضيح أو تنقيح متطلبات المستخدم.

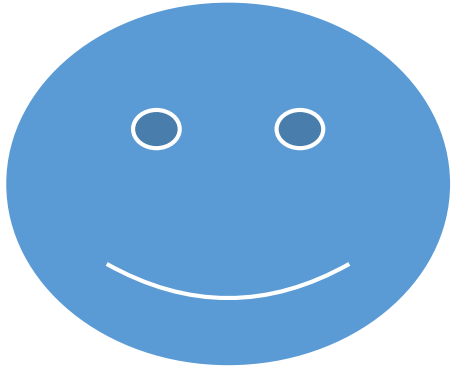
3.1. أنواع الأنظمة Systems classifications:

- اعتماداً على مفهوم البيئة وعلاقتها بالنظام تم تقسيم النظم إلى نظم مغلقة (Closed System) ونظم مفتوحة (Open System).

- **النظم المغلقة Closed System:**
- النظام المغلق هو النظام الذي ليس له علاقة من أي نوع مع البيئة المحيطة به ويعمل بمعزل عنها.
- بمعنى لا يحتاج النظام إلى مدخلات أو عملية اخراج من وإلى البيئة لكي يعمل.
- من أمثلة الأنظمة المغلقة البيوت الزجاجية، الساعة، الكتاب، المحركات التي تعمل بالبطاريات، التفاعلات الكيميائية التي لا تحتاج إلى مؤثر خارجي.
- **النظم المفتوحة Open System:**
- النظام المفتوح عكس النظام المغلق فهو يتفاعل مع بيئته ويعتمد عليها في اتمام عملياته عن طريق المدخلات التي تتم معالجتها ويقوم بإخراج المعلومات للبيئة كنتائج عمل معين، كذلك يتأثر بكل العوامل الخارجية المحيط به.
- الأمثلة على هذا النوع من الأنظمة عديدة مثل جسم الإنسان ، نظام إداري ، نظام الحاسوب...الخ.

4.1. أمثلة على الانظمة:

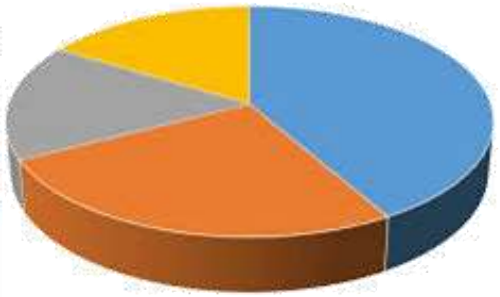
- نظام الحاسوب: هو مجموعة من الأجهزة والبرمجيات تحت نوع معين من التحكم لمعالجة بيانات وإنتاج معلومات.
- نظام التسوق: مجموعة من الناس والبضائع والمعدات والإجراءات تعد أو تنتج وتوزع بضاعة او خدمات للمستخدم.
- نظام الخطوط الجوية: ويتكون هذا النظام من موظفين وطائرات ورحلات ومسافرين بالإضافة لعمليات شحن، و يعمل هذا النظام على تنظيم رحلات للمسافرين وتقديم خدمات لهم خلال الرحلات بالإضافة إلى تقديم خدمات شحن جوي.
- النظام المصرفي: يتكون من موظفين، اجراءات وعملة ويقوم النظام على أساس تقديم خدمات ايداع و صرف وتقديم قروض الى الزبائن بحيث تحكم هذه المعاملة قوانين ولوائح محددة لتسيير السياسة المالية للمصرف.



The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



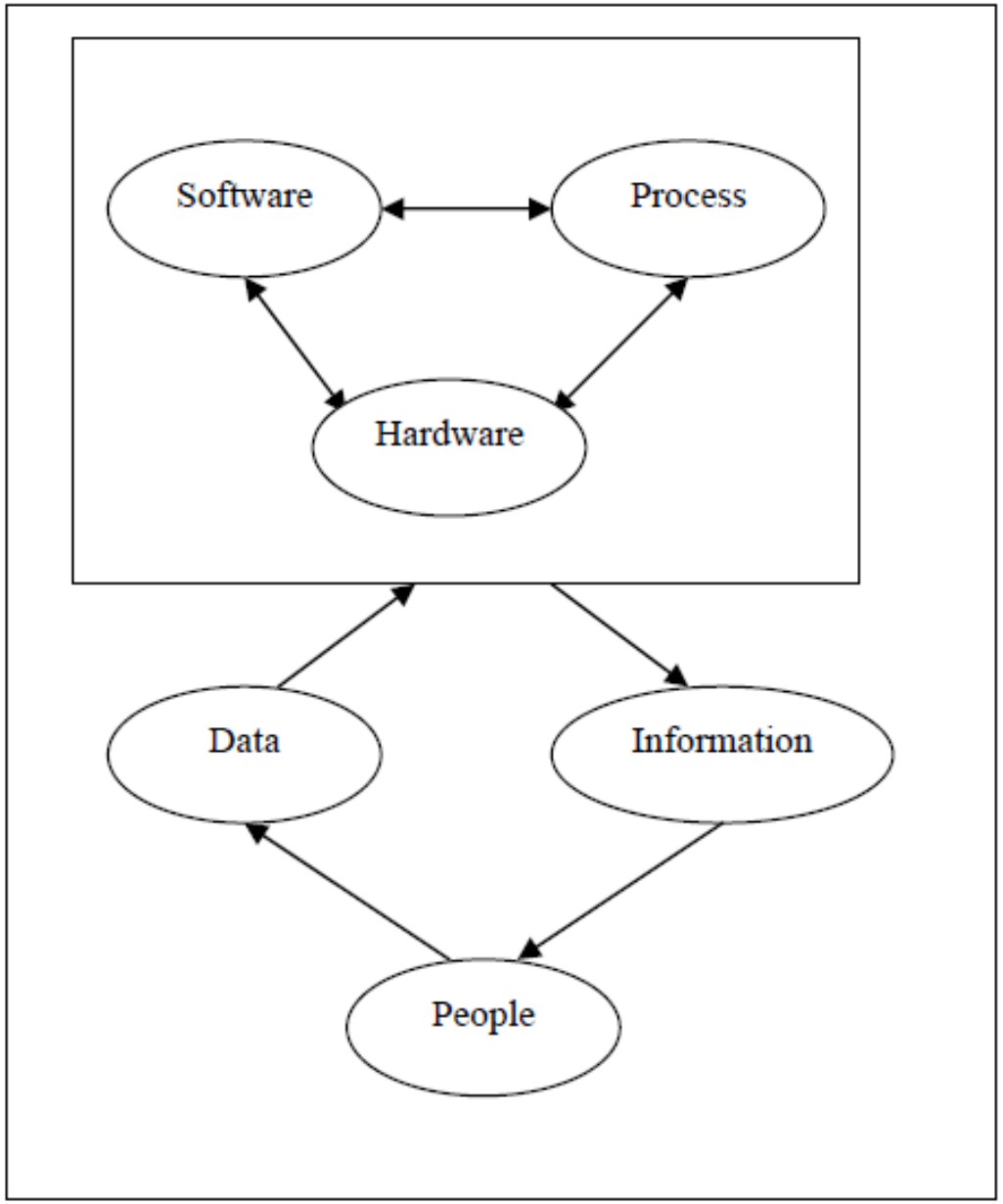
المحاضرة الثانية

2. أنظمة المعلومات Information Systems

1.2. مفهوم نظام المعلومات:

- تعرف أنظمة المعلومات بأنها أنظمة تتكون من بيانات وبرمجيات وأجهزة وأشخاص وعمليات معالجة. و كما هو مبين في الشكل 5 تتكون من الأجزاء التالية:

1. البرمجيات Software: وتشمل البرامج والبيانات والتوثيق وهي تنقسم الى نوعين:
   - **برمجيات تطبيقية Application software:** مثل برمجيات الرواتب ومنظومات ادارة المخازن.
   - **برمجيات النظم System software:** مثل انظمة التشغيل والمترجمات وبرامج النسخ الاحتياطي وبرامج الحماية من الفيروسات.
2. العتاد او الأجهزة Hardware: و هي تشمل الحاسوب و الأجهزة الملحقة له مثل الطابعة والماسحة، بالإضافة إلى أجهزة الاتصال لربط الشبكات.
3. الأشخاص People: هم المستخدمين للمنظومات والموظفين أو المستفيدين مثل الزبائن والموردين والمصدرين.
4. البيانات Data: هي سيل من الحقائق في صورتها الأولية قبل معالجتها وتحويلها إلى معلومات مفيدة مثل الأسماء، العناوين، الأسعار، الكميات، أسماء المواد والدرجات.
5. المعالجة Process: هي مجموعة الاجراءات التي تتم على البيانات لتحويلها إلى معلومة مفيدة.



شكل 5: انظمة المعلومات Information System.

2.2. أقسام نظم المعلومات:

تنقسم نظم المعلومات حسب وظائفها كالتالي:

- منظومة معالجة المعاملات Transaction Processing Systems:
  - وتسمى أيضاً منظومات العمليات مثل منظومة طلبية مبيعات ومنظومة جباية الفواتير.
  - ومهمة هذه المنظومات معالجة البيانات واجراء حسابات وتصنيفات معينه عليها لإصدار فواتير وطلبيات.
- منظومة إدارة المعلومات Management Information System:
  - هذا النوع من المنظومات يساعد المدراء في اتخاذ قرارات أفضل،
  - فمثلاً باستخدام بيانات المبيعات المأخوذة من منظومة العمليات يمكن لمنظومة إدارة المعلومات أن تشير إلى الأصناف الأكثر رواجاً والأصناف الأقل مبيعاً.
  - وبهذه الطريقة يستطيع متخذ القرار أو المسؤول أن يأخذ القرار الصحيح بجلب الأصناف الأكثر مبيعاً وأرباحاً إلى المخزن قبل نفادها.

2.2. أقسام نظم المعلومات...

- المنظومات الخبيرة Expert Systems:
  - هي المنظومات التي تستخدم تقنية الذكاء الاصطناعي سواء كان استعمالها في التحكم بآلات أو أنظمة معلومات مثل التجارة الإلكترونية أو أنظمة طبية معينة.
  - أصبحت المنظومات الخبيرة يشار إليها باسم منظومات إدارة المعرفة Knowledge Management Systems بعد أن استخدمت في الإنترنت لإدارة كم هائل من المعلومات على الشبكة للبحث عن المعرفة والوصول إلى المعلومة المطلوبة.
  - تملك منظومة ادارة المعرفة قاعدة بيانات كبيرة لتكون تحت سيطرة المستخدم للوصول إلى المعلومة التي يحتاجها باستخدام محركات بحث داخل المنظومة.

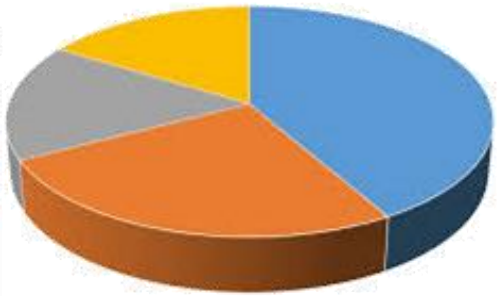
2.2. أقسام نظم المعلومات....

- منظومات تحسين إنتاجية المستخدم Employee Productivity Enhancement Systems:
  - وتسمى أيضاً منظومة ميكنة المكاتب وتتميز بسهولة استخدامها وهي منظومات يستخدمها الموظف لمساعدته في انجاز اعماله الروتينية (مثل الطباعة) بكفاءة عالية.
  - ومن أمثلة هذه المنظومات:
    - نظم البريد الالكتروني Email system
    - معالجة النصوص Word processing
    - الجداول الإلكترونية Spreadsheets
    - قواعد البيانات Databases

The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة الثالثة

3. دورة حياة وإعداد النظام System Development Life Cycle

من الأهمية أن يتم التخطيط و الاعداد الجيد لأي منظومة حتى لا تحدث مشاكل و أخطاء قد تكلف الكثير بسبب سوء الاعداد والتخطيط. لذلك فإن المنظومات يجب أن يتم اعدادها بطريقة منظمة باستخدام منهجية (طريقة) سليمة ويوجد ثلاثة منهجيات تستخدم في تحليل وتصميم نظم المعلومات هي:

1. الطريقة غير الهيكلية Unstructured Methodology:

- هي طريقة قديمة في اعداد المنظومات بدأت في الستينات مستخدمة المخططات الانسيابية كوسيلة للتحليل والتصميم.
- لا تستخدم حالياً إلا نادراً نظراً لتعقد المنظومات المستخدمة.

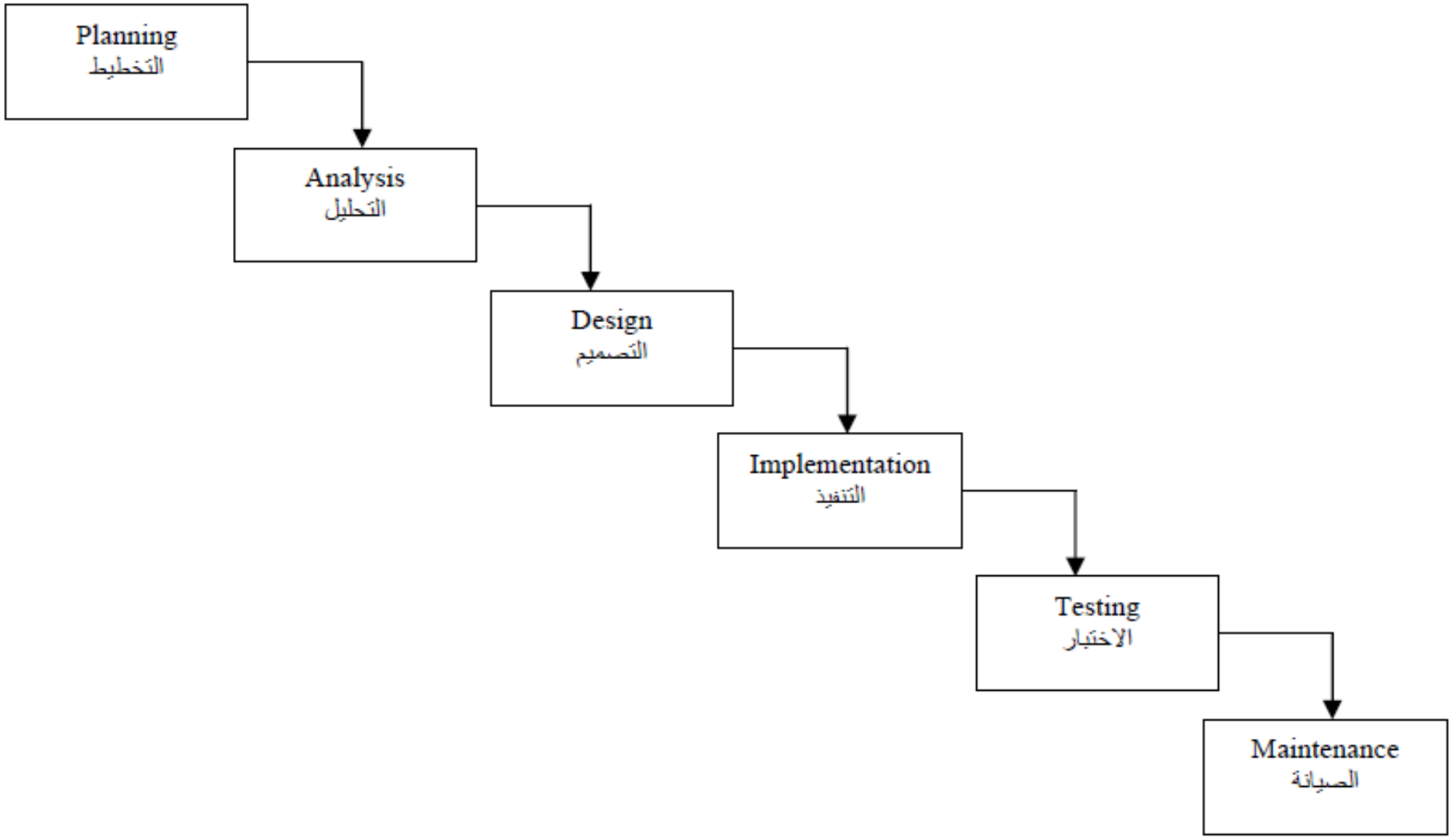
2. الطريقة الهيكلية Structured Methodology:

- بدأ استخدام هذه الطريقة في السبعينات وتتميز بالأدوات والتقنيات الهيكلية المنظمة والسهلة الاستعمال. هذا النوع من الطرق يناسب إعداد المنظومات الكبيرة والمعقدة نظراً لأنها تقسم المنظومة إلى أجزاء صغيرة لتبسيطها.
- كمثال على هذه الطريقة (مخطط انسياب البيانات و المخطط الهيكلي).

3. الطريقة الشيئية (الكائنية) Object_Oriented Methodology:

- الطريقة الشيئية (الكائنية) تستعمل تقنيات متقدمة مثل إعادة استعمال الأجزاء البرمجية لتقليص وقت إعداد المنظومات.
- و بعض الأدوات في هذه الطريقة هي مخطط حالة الاستعمال (Use Case Diagram) و مخطط الفصيلة Class Diagram)..... الخ.

- إن أبسط نموذج لدورة حياة إعداد منظومة هو ما يعرف بالنموذج التتابعي أو النموذج التدفقي الذي يعني أننا نقوم بإعداد المنظومة في تسلسل أي مرحلة بعد مرحلة. يتكون هذا النموذج كما هو مبين بالشكل 6 من عدد من المراحل المتعاقبة بحيث تعرف كل مرحلة بأنها عبارة عن عدد من الأنشطة يجب ان تنجز في فترة زمنية وقبل الانتقال الى المرحلة التي تليها.



شكل 6: دورة حياة وإعداد النظام System Development Life Cycle

ويمكن توضيح هذا النموذج بشيء من التفصيل كما هو مبين بالجدول التالي:

Phase المرحلة		Activities النشاطات
Planning	التخطيط	1- طلب المستخدم User request 2- دراسة الجدوى Feasibility Study 3- خطة المشروع Project Plan 4- مقترح المشروع Project Proposal
Analysis	التحليل	1. إيجاد الحقائق Fact find 2. تحليل المتطلبات Requirement analysis
Design	التصميم	1. التصميم المبدئي Initial Design 2. التصميم المفصل Detailed Design
Implementation	التنفيذ	1. التشفير Coding 2. اكتشاف الأخطاء Debugging 3. اختبار الوحدة Unit test
Testing	الاختبار	1. اختبار التكامل Integration test 2. اختبار النظام System test 3. اختبار القبول Acceptance test
Maintenance	الصيانة	1. التحسين Enhancement 2. التكيف Adaptation 3. التصحيح Correction 4. إعادة الهندسة Re-engineering

جدول 1: نشاطات مراحل دورة حياة النظام

عند البدء في دورة حياة تطوير النظام يجب ملاحظة التالي:

1. كل مرحلة تحتاج إلى مستلزمات (برمجيات، أجهزة و بشر) لإكمال كل عملياتها.
2. كل مرحلة تحتوي على مدخلات ومعالجة ومخرجات.
3. يجب اجراء مراجعة في نهاية كل مرحلة.
4. يجب إعداد وثيقة المواصفات ومراجعتها في نهاية كل مرحلة.
5. يجب التحقق من اكتمال وصحة ووضوح وثيقة المواصفات في كل مرحلة.

في الجزء التالي سنبدأ في شرح تفصيلي لكل مرحلة من مراحل تطوير واعداد النظام.

1.3. مرحلة التخطيط Planning Phase :

- في هذه المرحلة يوجد أربعة أنشطة رئيسية يقوم بها المحلل وهي:
- طلب المستخدم User request
- دراسة الجدوى Feasibility study
- خطة المشروع Project plan
- مقترح المشروع Project proposal

User request طلب المستخدم

- الهدف من هذه الخطوة هي معرفة احتياجات المستخدم والمشاكل التي يواجهها في نظامه القائم. تستخدم نماذج خاصة يملأها المستخدم لغرض تحديد المشاكل الموجودة في النظام الحالي وتعريف الحدود والأهداف المرجوة من النظام المقترح.
- و في هذا النموذج ، يمكن أن يطلب المستخدم واحداً من الاختيارات التالية:
  - المطلوب نظام جديد.
  - المطلوب تحسين النظام الحالي.
  - المطلوب تصحيح أخطاء المنظومة الحالية.
- في المثال التالي نموذج مبسط لطلب المستخدم، هذا النموذج تسلمه لإدارة الشركة لاتخاذ القرار بخصوص البدء في التحليل وذلك بتكليف محلل نظم للقيام بالمهمة.

من: رئيس قسم المبيعات

إلى: المدير العام لشركة

الموضوع: نطام المبيعات

يقوم قسم المبيعات ببيع الاثاث. يدفع الزبون إما نقداً أو بالدين. يستقبل القسم أكثر من 50 زبون في اليوم الواحد منهم 50% تقريباً الدفع نقداً. ويتوقع القسم زيادة في المبيعات نظراً لجودة المنتجات التي تقدم للزبائن. ومع ذلك أود ان افيدكم بأننا في حاجة لتحسين أداء قسم المبيعات. فالعمليات في القسم تجرى يدوياً. والنظام الحالي يعاني بعض المشاكل في الحصول على المعلومة عند الحاجة إليها. ونحن نود أن نطور خدماتنا و التفوق على المنافسين في السوق. ويمكن أداء ذلك بربط جميع فروع المبيعات و بهذه الطريقة نحصل على رضا الزبائن الحاليين وجلب زبائن جدد.

نموذج طلب المستخدم

دراسة الجدوى Feasibility study

هذه الدراسة يجب أن تتم لأغلب المشاريع المتوسطة والكبيرة الحجم وهي تتكون من:

- الجدوى الاقتصادية
- الجدوى الفنية
- الجدوى التشغيلية

الجدوى الاقتصادية

- هي تحليل التكاليف والمزايا لمعرفة ما إذا كانت المزايا تفوق التكاليف المتوقعة. لتحقيق هذا الهدف يقوم المحلل بإعداد نموذج يذكر فيه المزايا المتوقعة مثل : تحسين الخدمات ، تقليص التأخير ، تحسين رضا الزبائن ، اتخاذ قرارات أفضل ، تحسين صورة الشركة ، إعداد موقع لشركة على شبكة الأنترنت.
- ثانيا يقوم المحلل بحساب التكلفة التقديرية للمشروع كالتالي:

  تكاليف العتاد والمعدات (-------- ل.س)

  تكاليف البرمجيات (-------- ل.س)

  تكاليف القوة العاملة (-------- ل.س)

  تكاليف برامج التدريب (-------- ل.س)

  تكاليف التجهيزات (-------- ل.س)

  تكاليف الاستشارات (-------- ل.س)

  مصاريف اخرى (-------- ل.س)

  المجموع (-------- ل.س)

- بالإضافة الى ذلك يجب دراسة الفترة الزمنية وتقدير الوقت اللازم لكي يبدأ في الحصول على عائد من المشروع يفوق التكاليف. تتم العملية بمقارنة التكاليف المتوقعة مع العوائد المتوقعة في آل سنة وحساب الفترة الزمنية التي يسترد فيها المشروع تكاليفه. وبذلك يمكن معرفة متى يبدأ المشروع في استرداد التكاليف ومتى تبدأ مرحلة تحصيل الارباح.

الجدوى الفنية

- من هذه الدراسة نتوقع الاجابة على الاستفسارات التالية:

❑هل للشركة المعدات والبرمجيات و الشبكة اللازمة للمشروع؟ إذا كان ذلك غير متوفر ، هل يمكن الحصول على هذه المستلزمات بسهولة؟

❑هل للشركة الخبرات الفنية اللازمة ؟ إن كان غير ذلك فهل يمكن توفيرها؟

❑هل سيتمكن النظام من معالجة حجم المعاملات المتزايد في المستقبل؟ إذا كانت الإجابة لهذه الاسئلة بنعم فإن النظام المقترح يعتبر ذو جدوى فنية.

الجدوى التشغيلية

- هذه الدراسة لتجيب عن الاستفسارات التالية:
  - ❑هل ستكون المنظومة بعد إعدادها سهلة الاستخدام وتعمل بكفاءة؟
  - ❑هل سيتقبل المستخدمون والمدراء المشروع الجديد، وهل سيكون لديهم الشعور بأن المنظومة الجديدة ستحقق احتياجاتهم؟
  - ❑هل سيتم قبول المنظومة بعد استلامها من قبل المستخدم؟
  - ❑هل ستتطلب المنظومة الجديدة تدريب؟ وهل سيتم التدريب قبل عملية التسليم؟
  - ❑إذا كانت الاجابة عن هذه الاستفسارات بنعم فإن النظام ذو جدوى تشغيلية.

دراسة الجدوى هي عبارة عن تحليل مبدئي يتم قبل البدء في المشروع. وينتج عن هذه الدراسة تقرير كما هو مبين بالجدول التالي:

1. اسم الشروع:
2. وصف النظام الحالي:
3. الحدود:
4. الافتراضات والشروط:
5. حلول بديلة:

   الخطوط العريضة للنظام
   الجدول الزمني
   القيود
   المستلزمات
   المزايا والعيوب
   الفوائد والتكاليف
6. التوصيات والخلاصة
7. الملحقات

محتويات تقرير دراسة الجدوى

Project plan وثيقة خطة المشروع

يجب أن تعد هذه الوثيقة بعد الموافقة على دراسة الجدوى واختيار الحل المطلوب.

وتتضمن وثيقة خطة المشروع النقاط التالية:

- ملخص المشروع
- جدول المشروع
- قيود المشروع
- الأهداف المرجوة
- الوظائف المطلوبة
- نطاق المشروع
- التكلفة التقديرية
- المستلزمات المطلوبة
- الأدوات اللازمة لمرحلة التحليل والتصميم
- خصائص الجودة المطلوبة
- تحليل الأخطار

وثيقة مقترح المشروع Project proposal

يجب أن تعرض هذه الوثيقة البنود التالية:

- وصف المشكلة
- مشاكل النظام الحالي
- شرح للنظام المقترح وتبرير اختيار النظام مقارنة بين الحلول البديلة
- الجدوى الاقتصادية و الفنية والتشغيلية للنظام المقترح
- الأدوات التي ستستخدم في المشروع
- الأفراد ودورهم في المشروع
- تأثير المشروع المقترح على عمليات النظام الحالي
- نسخة من خطة المشروع التي تم عرضها في البند السابق.

2.3. مرحلة التحليل Analysis Phase:

- الهدف الرئيسي من التحليل هو تعريف متطلبات المستخدم. بتعبير آخر يجب أن نحدد في هذه المرحلة ما هو متوقع من النظام المقترح حتى نحقق متطلبات المستخدم. يمكن أن نقسم متطلبات المستخدم إلى ما يلي:

1. متطلبات وظيفية: الوظائف التي ستؤديها المنظومة.
2. متطلبات الأداء: ماهي السرعة المطلوبة من المنظومة لإنجاز الوظائف ويقاس هذا الأداء بوقت الاستجابة.
3. متطلبات الإدخال: وهي البيانات التي تدخل المنظومة لغرض المعالجة.
4. متطلبات الإخراج: هي المعلومات التي تخرج من المنظومة بعد إدخال المدخلات و معالجتها. وقد تكون المخرجات على شكل كشوفات ، تقارير أو استعلامات.
5. متطلبات الجودة: الاعتمادية والأمن وسهولة الاستعمال وقابلية الصيانة.

لكي نحدد متطلبات المستخدم بالكامل ، هناك بعض البنود التي يجب أن يحددها محلل النظم وهي:

ملخص عن المشروع: تعريف لنشاط المستخدم والمشاكل التي تحتاج إلى حل. مثلاً قد يوجد لدينا متجر لبيع الكتب نقداً أو على الحساب ويشعر صاحب المتجر بالحاجة إلى تطوير نظام سجلات الزبائن لزيادة الدخل.

الأهداف: شرح الغاية والغرض من المنظومة الجديدة مثلاً يجب على المنظومة أن تسرع معاملات معينة بنسبة 50 %. وفي مثال اخر المستهدف من المنظومة الجديدة توفير تقارير فورية للإدارة من خلال الشبكة.

الحدود: أي نطاق العمل المطلوب أي توضيح المعلومات والعمليات التي ستشملها المنظومة. مثلا قد تحوي المنظومة الجديدة على معلومات عن الزبون والمبيعات ولكن دون أن تشمل على المخزون.

البيئة: هي كل ما يحيط بالمنظومة من عمليات و إجراءات وسياسات قد تؤثر على المنظومة أو تتأثر بالمنظومة. مثلاً قد يكون نظام المخزن بيئة لنظام المبيعات.

القيود: هي الضوابط التي تفرض على المنظومة المقترحة مثل الوقت المتوفر و الميزانية المرصودة. فمثلاً قد تحدد الميزانية بمبلغ محدد بالأرقام مع تحديد المدة الكافية لإنجاز المنظومة وتحديد المدة الزمنية لضمان عمل المنظومة.

وكما ذكرنا في جدول 1 تشمل هذه المرحل على نشاطين هما ايجاد الحقائق (جمع المتطلبات) و تحليل المتطلبات.

1. إيجاد الحقائق (جمع المتطلبات):

يقوم المحلل في هذه المرحلة بجمع كل المعلومات التي تساعده على فهم طبيعة العمل وآلياته ومشاكله القائمة و يمكن جمع المتطلبات بعدة طرق وهي كالتالي:

- **البحث العام:** ويستخدم فيه المراجع من المكتبة وشبكة المعلومات (الأنترنت).
- **الاستبيان:** ويقوم المحلل بتصميم استبيان يضع فيه آل الاسئلة والاستفسارات المناسبة لكي يصل إلى المعلومة الصحيحة والمهمة.
- **المقابلات:** تنسيق اجتماعات مع الموظفين أو مستخدمي النظام و المسؤولين على اتخاذ قرارات مهمة داخل النظام القائم. يقوم المحلل بالاستفسار حول النظام الحالي ومتطلبات المنظومة الجديدة.
- **العرض التجريبي:** يمكن ان يقوم المحلل بتصميم عرض تجريبي للمنظومة لعرضها على المستخدمين والمسؤولين للوصول لكي تساعد على وضع تصور واضح للمنظومة الجديدة و تجنب اي أخطاء أو سلبيات غير متوقعة.
- **مشاريع مشابهة:** من المفيد جداً للمحلل الحصول على مشاريع مشابه نفذت لدراستها لكي يتمكن من الاستفادة من أخطاء الأخرين.
- **عينات نماذج وتقارير:** من المهم ان يتحصل المحلل على نماذج و تقارير وفواتير النظام الحالي نظراً لأنها تحتوي على البيانات المهمة في عمليات الادخال والاخراج مع وجود تفاصيل حول العمليات التي ستنفذ داخل المنظومة.

يجب على المحلل عند القيام بهذه النشاطات العمل بالإرشادات التالي:

- ✓المقابلات الشخصية يجب تحديدها بتاريخ ومكان وفترة معينة.
- ✓أهداف المقابلة يجب أن تحدد مسبقاً.
- ✓يجب أن يكون الاستبيان قصيراً.
- ✓عند تصميم الاستبيان يجب استعمال أسئلة ذات إجابات قصيرة مثل (نعم و لا) او إجابات اختيارية.
- ✓من المفيد تنسيق الزيارات مع المدراء والموظفين المتواجدين.
- ✓اكتب ملاحظات قصيرة وراجع ما كتبت مع المستخدمين.

2. تحليل المتطلبات:

- يبدأ المحلل في هذه الخطوة بعمل تنظيم للمتطلبات لغرض التوصل لفهم واضح للنظام القائم والنظام المقترح.
- يوجد مجموعة من الأدوات يستخدمها المحلل لتنفيذ عملية تحليل المتطلبات مثل:
  - المخطط الانسيابي للبيانات Data Flow Diagram
  - قاموس البيانات Data Dictionary
  - جدول القرارات Decision Table
  - شجرة القرارات Decision Tree
  - الانجليزية المركبة Structured English
  - المخطط System Analysis and Design Technique (SADT)
  - مخطط العلاقات Entity Relationship Diagram (ERD)

في نهاية هذه المرحلة يقوم المحلل بإعداد وثيقة توصيات المتطلبات والتي تتضمن النقاط التالية:

- ملخص المشكلة
- مخطط انسياب البيانات
- قاموس البيانات
- وظائف المنظومة
- متطلبات الأداء
- البيئة
- القيود
- حدود المنظومة
- معيار القبول
- لغة البرمجة المستخدمة
- جدول وشجرة القرارات
- مخطط العلاقات

- ومن المهم ملاحظة أن هذه الوثيقة تصف المشاكل وليس الحلول وهي نتائج الدراسة وليس عملية المعالجة،
- كما أنها وثيقة بين الزبون والمحلل وسوف تستخدم فيما بعد في التصميم.
- بالإضافة إلى ذلك فهي تقوم بتحويل الاحتياجات إلى متطلبات وتبين ما هو المتوقع من المنظومة وليس كيف تعمل.
- من المهم مراجعتها جيداً مع المستخدم لتفادي أي أخطاء مستقبلية.

ملاحظة: الفرق بين الاحتياجات والمتطلبات هو أن مصطلح متطلبات أدق وأكثر قابلية للاختبار مثلاً:

الاحتياجات: الملف يحمل 100000 سجل موظف

المتطلبات: الملف يجب ان يحمل أسماء كل الموظفين في الشركة.

3.3. مرحلة التصميم Design Phase:

تقوم عملية التصميم بترجمة المتطلبات إلى تمثيل الحل بحيث التصميم يركز على التالي:

تصميم معماري Software architecture design:

يتم تجزئة النظام البرمجي إلى مكونات بحيث كل مكونة يمكن أن تجزأ أكثر إلى وحدات (أجزاء برمجية) باستخدام المخطط الهيكلي كأداة تصميم من أعلى إلى أسفل.

تصميم هياكل البيانات Data structures design:

هي عملية وصف للبيانات من حيث نوعها وطولها أو حجمها و كذلك وصف للملفات (قواعد البيانات) من حيث وصف محتويات كل ملف والحقول البيانية التي يحتويها مع وصف العلاقات التي تربط بين الملفات.

تصميم الخوارزميات Algorithms design:

هي عملية كتابة الخطوات المنطقية لأجزاء البرمجيات في المنظومة وتستعمل طريقة التشفير المرمزة لهذه الكتابة.

تصميم واجهة المستخدم User interface design:

في هذه الخطوة يقوم المحلل بإعداد تصميم للشاشات الرئيسية للإدخال في المنظومة والتي سيستعملها المستخدم ولهذا من المهم أن يكون هذا التصميم بالتشاور مع المستخدم حتي يسهل عليه استعمال المنظومة ولتفادي أي أخطاء تنتج بسبب سوء الاستعمال للمنظومة. بالإضافة إلى ذلك يقوم المصمم بتصميم شاشات الاخراج والتقارير التي تطبع على الورق بتحديد البيانات المهمة في كل شاشة او تقرير او كشف.

4.3. مرحلة التنفيذ Implementation Phase:

الهدف الرئيسي من مرحلة التنفيذ هي انتاج جميع البرامج باستخدام إحدى لغات البرمجة للحصول على منظومة تقوم بتنفيذ جميع الوظائف التي تم إعداد وصف لها في المرحلة السابقة (مرحلة التصميم).

تعتمد هذه المرحلة على **المبرمج** بالذات (ليس بضرورة أن يقوم المحلل بمهمة البرمجة) وهو يقوم بتنفيذ النشاطات التالية:

- ترجمة مواصفات التصميم لكل جزء برمجي إلى برنامج بلغة البرمجة (شفرة المصدر).
- ترجمة البرنامج و البحث و تصحيح الأخطاء.
- اجراء اختبار لكل برنامج على حدى ثم اختباره في نطاق المنظومة كاملة وباستخدام بيانات ومدخلات حقيقية للتأكد من صحة المخرجات حسب طلب المستخدم. وهنا قد يكتشف أخطاء في التصميم.
- كتابة كتيب التشغيل لمساعدة المستخدم في استعمل المنظومة بكفاءة.

عند البرمجة يجب على المبرمج أن يتقيد بالنصائح التالية لمساعدته في إعداد منظومة ذات جودة عالية:

- **استعمال لغة البرمجة التي تناسب التطبيق المطلوب:**

يجب اختيار لغة البرمجة التي لها جميع الإمكانيات المطلوبة.

- **اجعل قابلية قراءة المخرجات أفضل ما يمكن باستعمال الواجهة الرسومية:**

يوجد برمجيات تساعد في إعداد نماذج رسومية لشاشات يمكن استخدمها لإعداد تصميم مبدئي لها في وقت اقصر و بما يناسب المستخدم .

- **اجعل قابلية قراءة نص البرنامج (شفرة المصدر) أفضل ما يمكن باستعمال أسماء ذات معنى وجمل تعليقية لشرح وظيفة كل جزء من البرنامج.**
- **اجعل وقت اعداد المنظومة أقل ما يمكن باستخدام لغة برمجة ذات كفاءة عالية:**

يجب عدم اختيار لغة البرمجة التي تحتاج لوقت أطول في البرمجة ،ومن المعروف ان لغات البرمجة تختلف من حيث درجة التعقيد في البرمجة والزمن المطلوب للإعداد البرامج .

- **اجعل عدد جمل شفرة المصدر أقل ما يمكن:**

بعض الوظائف والعمليات معقدة وطويلة من حيث الإجراءات وعدد خطوات التنفيذ وفي هذه الحالة يجب ان تقسيم هذه الوظائف والعمليات الى عمليات اصغر وبالتالي يصبح عدد جمل شفرة كل جزء برمجي أقل ما يمكن.

- **اجعل الذاكرة المطلوبة أقل ما يمكن.**

يجب أن يحقق الجزء البرمجي ما يلي:

- ✓أن يكون له مدخل واحد ومخرج واحد.
- ✓تجنب استخدام جملة (أمر) اذهب إلى (GOTO).
- ✓أن يحتوي على أقل من 30 جملة (صفحة تقريباً).
- ✓أن يتم تنسيق البرنامج بطريقة هيكلية باستعمال المسافة والأسطر الخالية.
- ✓أن لا يحتوي البرنامج على حلقات دوران متداخلة كثيرة (مثلاً لا يزيد عن 5 حلقات دوران متداخلة).
- ✓أن يحتوي على توثيق كافي حتى يسهل عملية تصحيح وتطوير أي برنامج في المنظومة.

5.3. مرحلة الاختبار Testing Phase:

الهدف الرئيسي من الاختبار هو تحديد إمكانية الحصول على النتائج المطلوبة عند تشغيل المنظومة. وهذا يعني أن اختبار المنظومة الجديدة يجب ان يحقق:

- التخلص من الأخطاء اللغوية
- التخلص من الأخطاء المنطقية
- التخلص من الأخطاء التنفيذية
- تحديد أخطاء مدخلات المستخدم
- تقييم سرعة أداء المنظومة
- تقييم أمن المنظومة
- اكتشاف أي وظيفة مفقودة
- تحقيق متطلبات المستخدم
- تقييم توثيق المنظومة

لاختبار منظومة جديدة ، عادة ما نستعمل عينة من البيانات و نقارن المخرجات (النتيجة) مع النتيجة المتحصل عليها من نفس العينة يدوياً.

يجب أن تحتوي هذه العينة على جميع الحالات المحتملة للتأكد من عدم وجود أخطاء منطقية.

كما ذكرنا سابقاً فإن كل جزء برمجي (برنامج فرعي) من المنظومة يتم اختباره منفصلاً قبل اختبار المنظومة ككل.

الاختبار يشمل:-

- اختبار التكامل
- اختبار النظام
- اختبار المستخدم

في اختبار التكامل يتم ربط واختبار الأجزاء البرمجية (التي تم اختبارها منفردة مسبقاً) كنظام متكامل (وحدة واحدة) للتأكد من أن المتطلبات كما عرّ فها المستخدم –قد تم تحقيقها. يوجد استراتيجيتان شائعتان لاختبار التكامل هما:

✓ *استراتيجية من الأسفل إلى الأعلى*

✓ *استراتيجية من الأعلى إلى الأسفل*

اختبار النظام يتحقق من جميع المتطلبات باستخدام عتاد بيانات حقيقية.

اختبار القبول يتم تنفيذه من قبل الزبون أو المستخدم لإثبات أن المتطلبات المدونة في وثيقة المتطلبات قد تم انجازها. وبذلك تعتبر المنظومة جاهزة للعمل.

6.3. مرحلة الصيانة Maintenance Phase:

الصيانة هي عملية جعل المنظومة تعمل بطريقة صحيحة في مواجهة العوامل التالية التي قد تؤثر على عملها:

- التغيرات بسبب أخطاء تحدث بعد تسليم المنظومة.
- استخدام تقنية جديدة.
- منع أي مشاكل قد تحدث في المنظومة بسبب رداءة في التصميم (إعادة هندسة المنظومة).
- تحسين قدرات المنظومة بإضافة وظائف جديدة.
- تبدأ الصيانة عندما يتم تسليم المنظومة و تركيبها (تحميلها) في موقع الزبون والبدء في تشغيلها.

النشاطات التالية يتم انجازها في هذه المرحلة:

تصحيح الأخطاء: يتعلق بتصحيح الأخطاء التي تظهر أثناء عملية تشغيل المنظومة و التي لم يتم اكتشافها في مرحلة الاختبار.

التكييف: نظراً للتغيرات السريعة في التقنية (برمجيات و معدات) قد تحتاج المنظومة إلى التكيف مع بيئة جديدة مثل منظومة تشغيل أو معدات حاسوب جديدة.

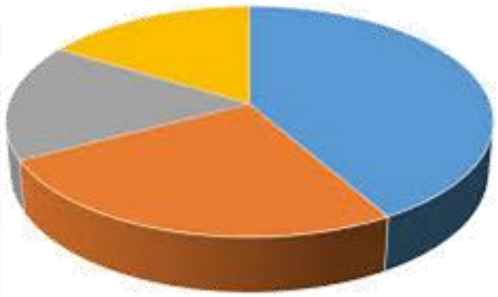
التحسين: ويشمل إما تطوير امكانيات المنظومة أو توفير وظائف أكثر للمنظومة عند اكتشاف متطلبات جديدة.

إعادة الهندسة: نقوم هنا بإعادة تصميم النظام وبرمجته إذا لزم الأمر لمنع أي مشاكل متوقعة من التصميم السيئ.

The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة الرابعة

4. تخطيط المشروع Project Planning

مفهوم التخطيط:

- التخطيط هو كل الأنشطة الإدارية التي تجرى لغرض تعريف الأهداف لمشروع ما والسبل اللازمة لتحقيق هذه الأهداف.
- يعتمد نجاح المشروع إلى حد كبير على التخطيط الجيد
- و التخطيط السيء عامل رئيسي في فشل المشروع.

1.4. أنشطة التخطيط Planning Activities:

1. تحديد الأهداف:

مثال: قد يكون أحد أهداف المشروع هو تقليص الوقت الذي تستغرقه معاملة ما بنسبة 40 % (مثل سحب صك مصرفي).

2. تحديد نطاق او حدود المشروع:

مثال قد يكون نطاق منظومة برمجية للمعهد محصورة في قسم الدراسة والامتحانات فقط.

3. تحديد مستلزمات وقيود المشروع:

مثال: قد تكون مستلزمات مشروع إعداد الرواتب كما يلي:

العنصر البشري: محلل نظم و عدد إثنين مبرمجين.

الأجهزة: 10 أجهزة حاسب شخصي و عدد إثنين طابعة ليزرية.

البرمجيات: VBASIC + Windows + SQL

القيود: في هذه الحالة تكون ما يلي:

التكلفة: الميزانية محددة بقيمة 10000 ل.س.

الوقت: يجب ألا يزيد زمن المشروع عن 5 أشهر.

1.4. أنشطة التخطيط ...Planning Activities

4. تجهيز جدول زمني للمشروع:

مثال: قد يكون الجدول الزمني لمشروع طوله 10 أشهر على النحو التالي

- التحليل شهران
- التصميم شهران
- البرمجة شهران
- الاختبار أربع أشهر

5. تحديد وظائف المنظومة:

مثال: وظائف منظومة المخازن يمكن أن تكون على النحو التالي:

- اضافة أصناف جديدة للمخزون
- الاستفسار عن صنف معين
- تحديث بيانات صنف معين
- تكوين تقارير

1.4. أنشطة التخطيط ....Planning Activities

6. تحديد معايير المشروع:

مثال: قد تكون بعض معايير منظومة مستشفى كما يلي:

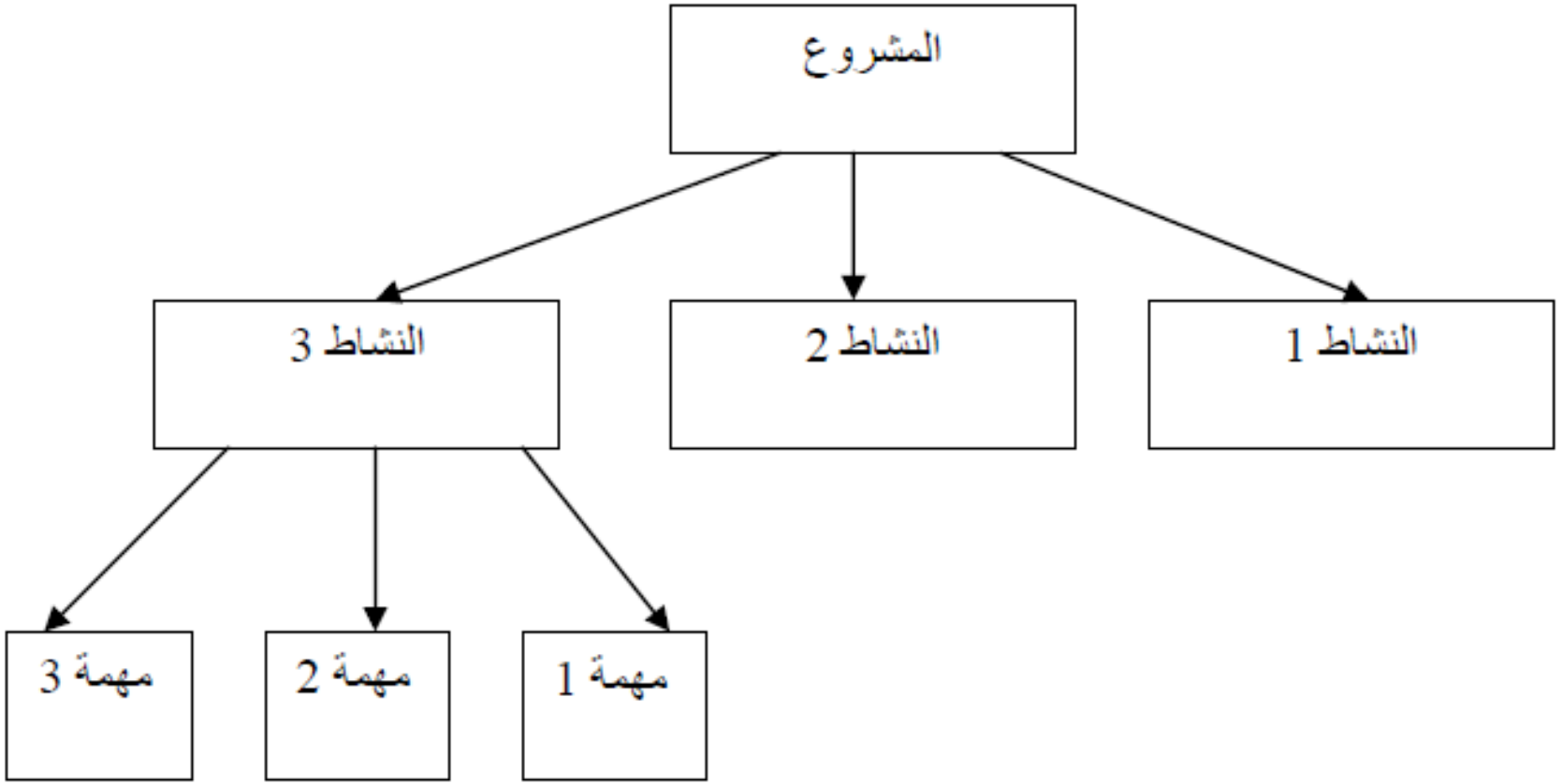
- استخدام لغة البيسك المرئية ولغة الاستفسار SQL كأدوات برمجية.
- يجب أن تعمل المنظومة في بيئة شبكة.
- يجب ألا تزيد عدد أسطر كل برنامج فرعي عن 30.

7. تعريف خصائص الجودة المطلوبة:

مثال: منظومة المصرف يجب أن تتمتع بالاعتمادية ، والكفاءة ، وقابلية الصيانة ، و سهولة الاستعمال.

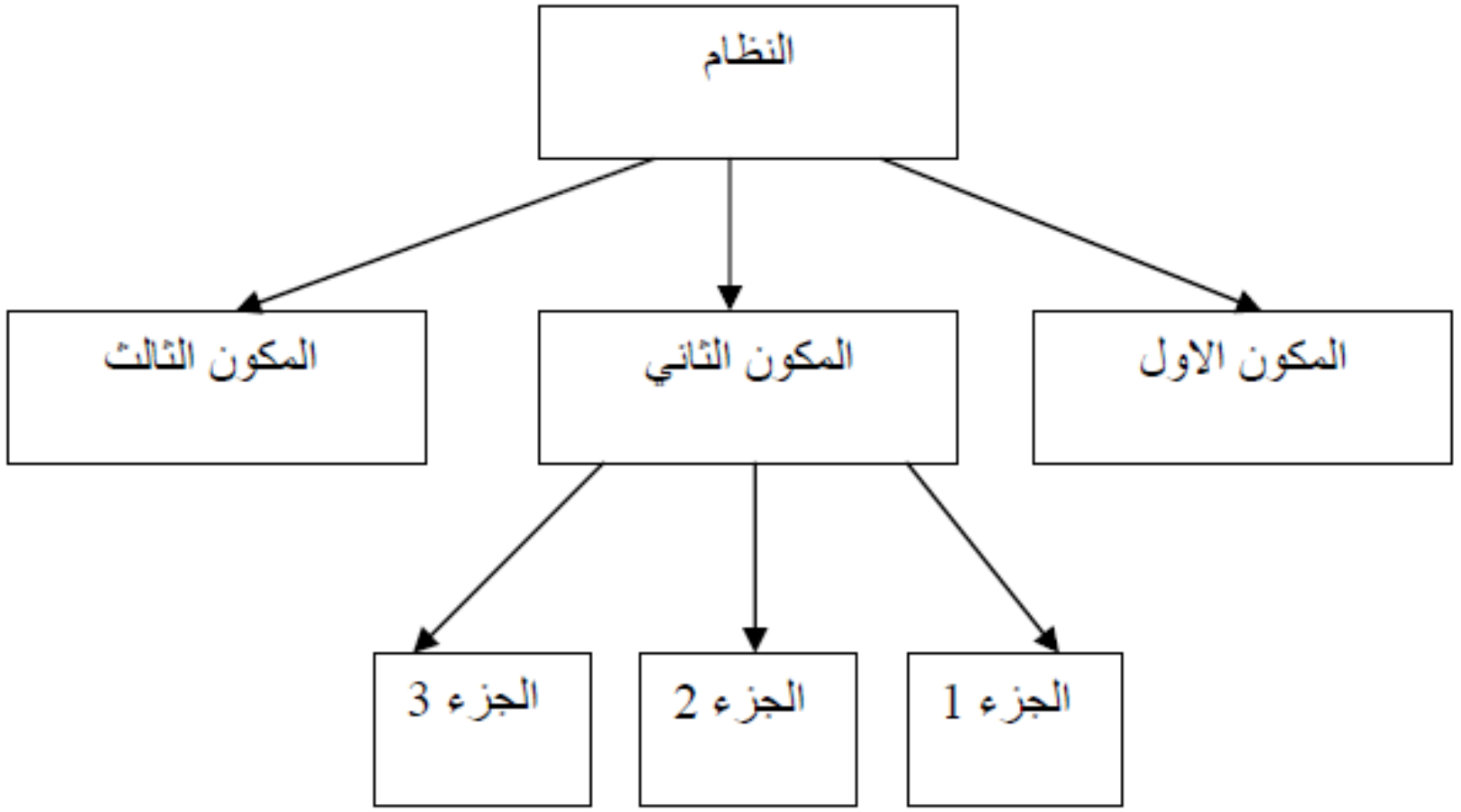
2.4. مخطط تركيبة تجزئة العمل:

- هذه الأداة تسمى Work Break_down Structure و اختصارها (WBS) وهي عبارة عن أداة تخطيط تبين حدود المشروع كمجموعة من المهام بتركيبة هرمية من أعلى إلى أسفل.
- في الشكل 7 التالي نجد أن شكلاً عاماً لـWBS حيث يقسم المشروع أولاً إلى عدد من النشاطات ، ثم يمكن أن يقسم كل نشاط إلى مجموعة مهام.



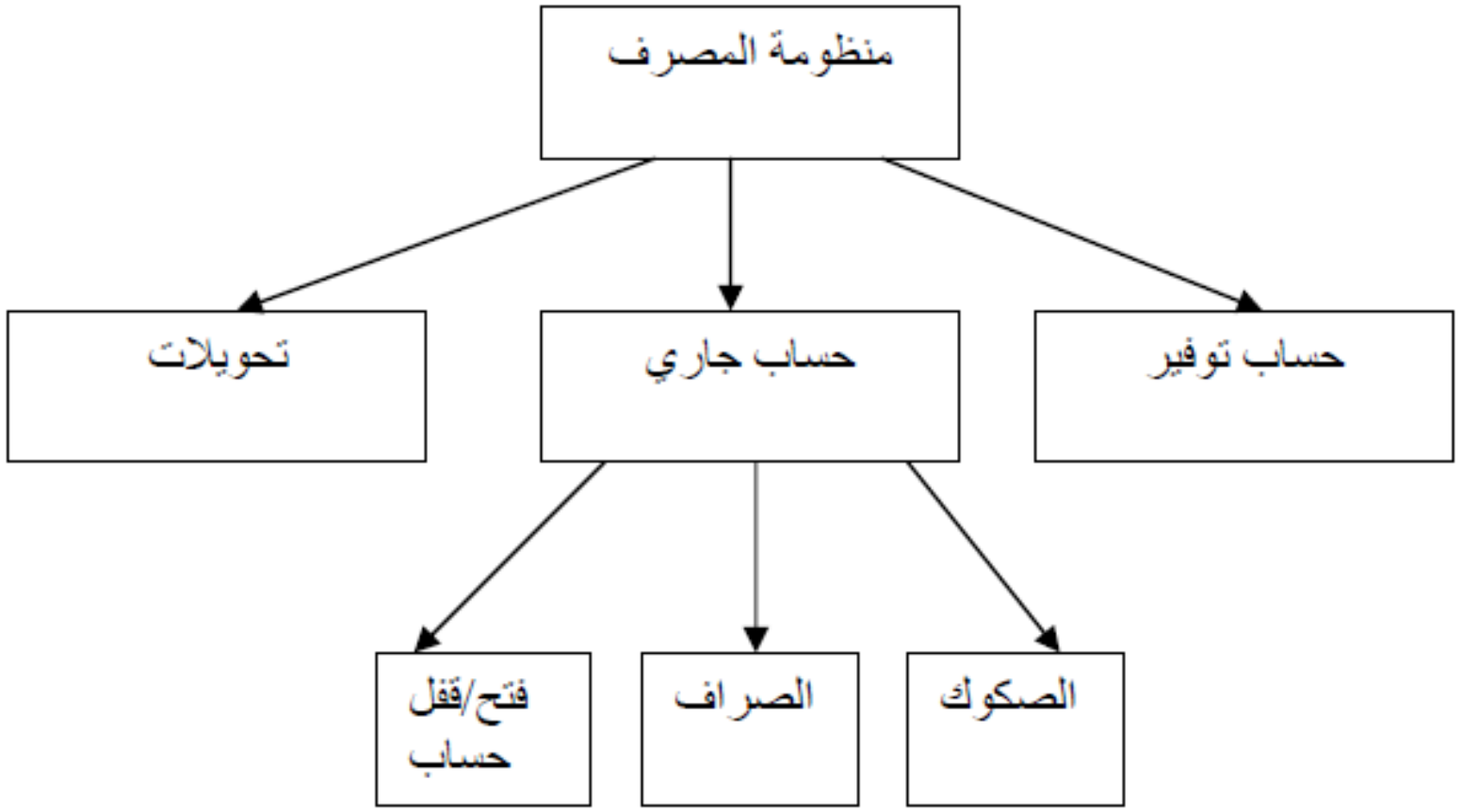
شكل 7: مخطط WBS لتجزئة عمل المشروع

- يمكن أن يستعمل مخطط WBS لرسم أجزاء المنظومة بحيث يبين مجال النظام كمجموعة من الأجزاء الفردية بطريقة من أعلى إلى أسفل.
- الشكل 8 يوضح تقسيم المنظومة المطلوب إعدادها إلى مجموعة من المكونات ثم إلى مكونات فرعية.



شكل 8: مخطط WBS لقسيم المنظومة

- مثال: ارسم مخطط من نوع WBS لمنظومة مصرفية ذات ثلاث مكونات فقط هي (حساب التوفير ، حساب جاري ، تحويلات) حيث حدود المنظومة يخص جزئية الحساب الجاري فقط وهي ذات 3 مكونات جزئية هي (الصكوك ، الصراف ، فتح و قفل حساب).
- انظر شكل 9.



شكل 8: مثال على مخطط WBS لمنظومة مصرف

ميزات مخطط WBS

1. أداة تخطيط قوية لتوضيح حدود المشروع.
2. أداة متابعة قوية لإعداد تقارير حول سير المشروع للإدارة والزبون.
3. يمكن استخدام WBS لتقدير التكلفة ومدة المشروع.
4. توفر آلية لقياس الأداء.

3.4. الجدولة وتقدير زمن المشروع:

✓الجدولة وتقدير الوقت للمشاريع يعتبران الجزء الصعب في التخطيط.

✓ويعتمد التقدير الزمني لأي مشروع على الفترة الزمنية التي يستغرقها انجاز كل مهمة.

✓عندما نقدر الزمن يجب ان نستعمل أيام العمل كوحدة قياس للزمن.

✓لاحظ أن ساعات الدوام قد تختلف من مكان إلى آخر، وفي هذه الحالة تستخدم الساعة كوحدة قياس للزمن.

ارشادات بخصوص تقدير الوقت:

- يجب التعرف على المهام التي يمكن أن تنجز على التوازي.
- استعمل أقل قوى ممكنة.
- خذ في الاعتبار أن بعض الناس قد يتركوا العمل أو يتعرضوا للمرض.
- يتم التقدير كما لو سيكون كل شيء على ما يرام ، ثم يتم زيادة 10 % مثلا للمشاكل الغير متوقعة.

مثلا إذا كان الجزء البرمجي يستغرق يوماً واحداً لإنهائه , وكان لدينا 30 جزء برمجي في المشروع ، فإننا نقدر الزمن لكتابة كل الأجزاء بمدة 30 يوماً. يضاف إلى ذلك 3 أيام (أي 10 %) للأحداث غير المتوقعة (مرض ، إجازة ، ....) فيصبح التقدير هو 33 يوماً.

يجب ملاحظة أن مواصفات المتطلبات والتصميم تتطلب ضعف الوقت اللازم لكتابة شفرة البرنامج.

4.4. أدوات الجدولة:

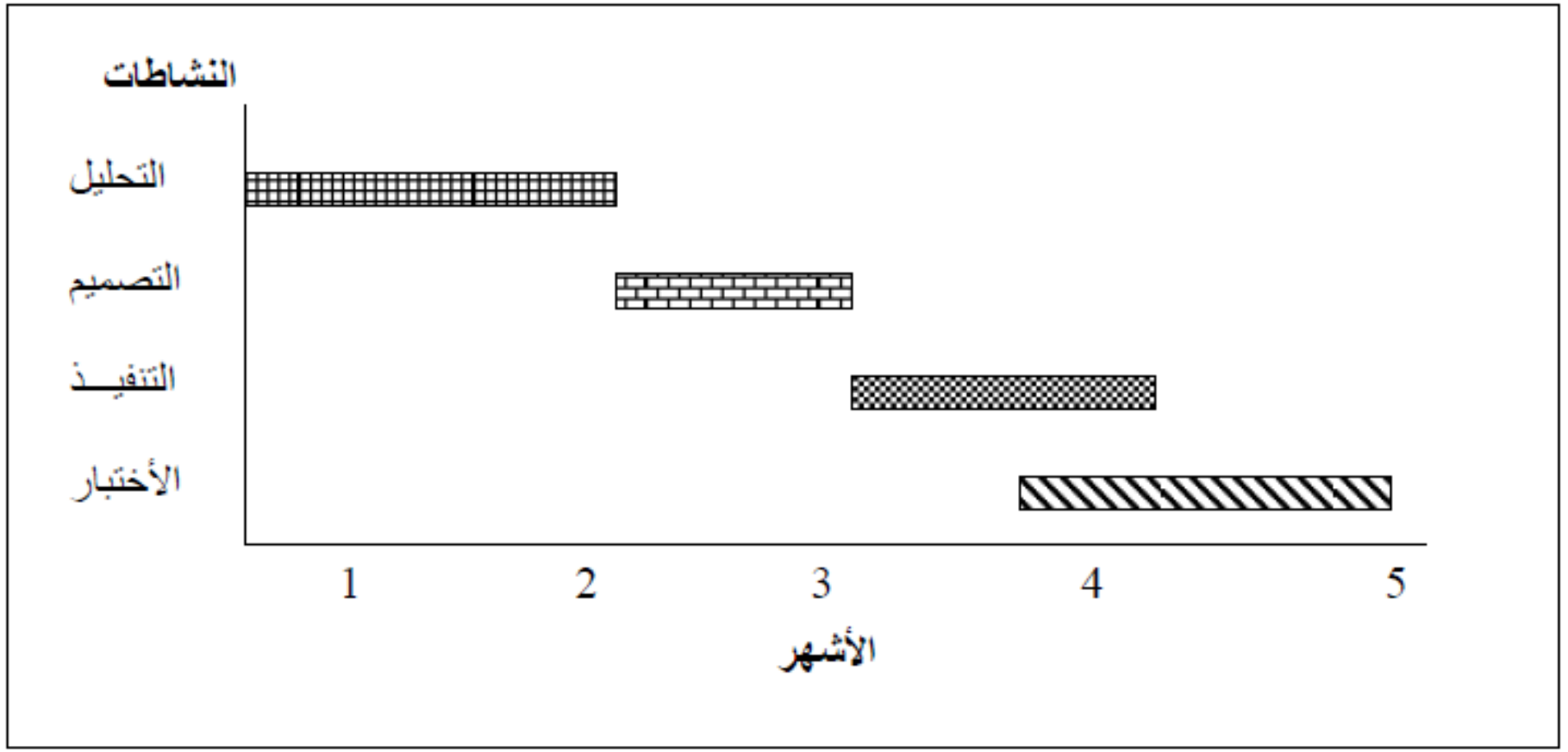
أدوات الجدولة يجب أن تبين:

- المهام
- العلاقة بين المهام
- العنصر البشري المعين لكل مهمة

- المخططات إما أن تنجز يدوياً ، أو بأدوات خاصة مثل برنامج "مدير المشاريع" الذي أعدته شركة مايكروسوفت.
- هذه الأدوات تفترض أن المشروع يتكون من مجموعة من النشاطات أو المهام المعتمدة على بعض. فمنها مهام لا يمكن أن تبدأ قبل أن تنتهي مهام أخرى ، ولكن بعض المهام يمكن أن تنجز بالتوازي مع نشاطات أخرى.
- في عملية الجدولة ، يستخدم مديرو المشاريع أدوات مثل: مخطط غانت، جدول الخط الزمني. هذه الأدوات تستخدم في تخطيط ومتابعة المشاريع.

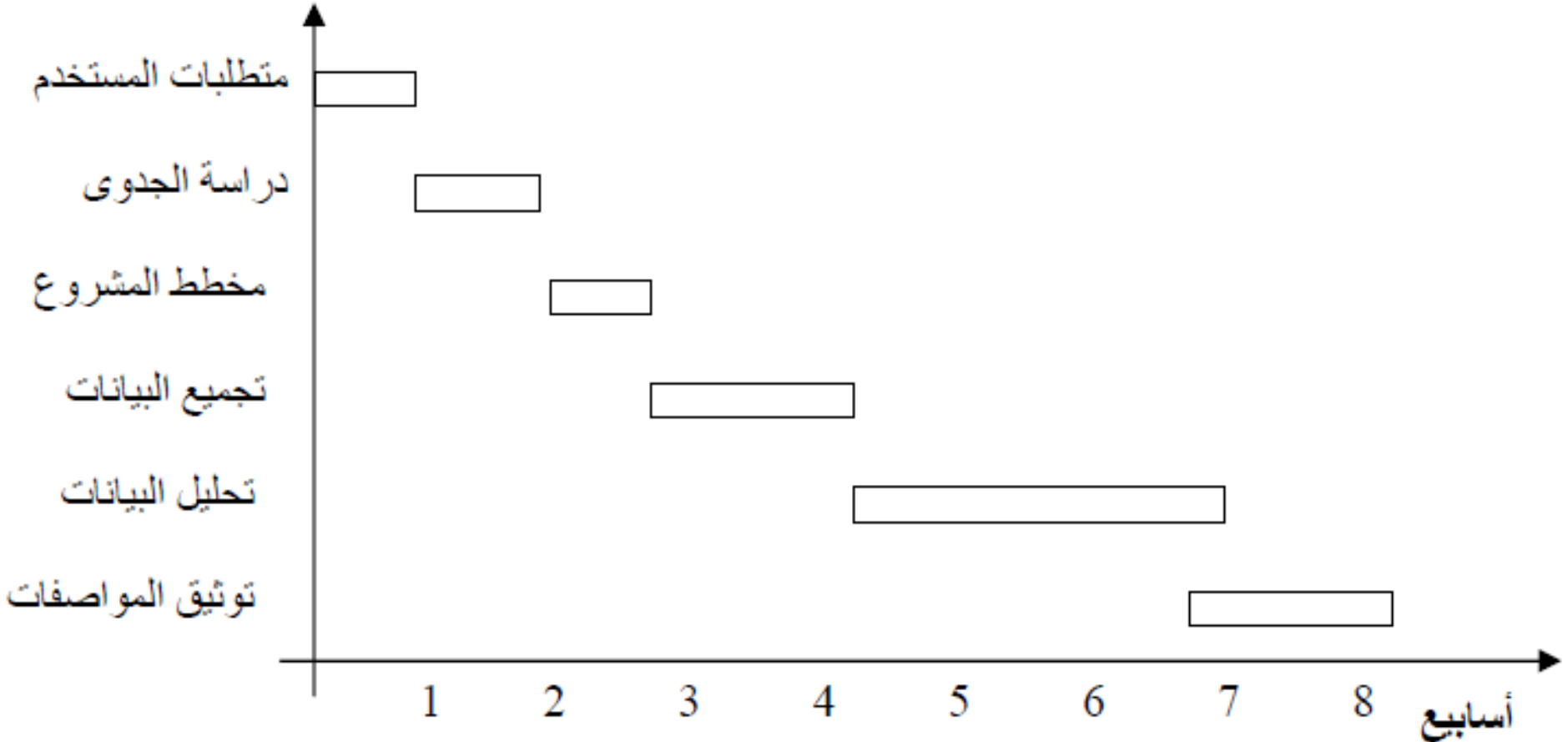
5.4. مخطط غانت Gantt Chart:

- مخطط غانت هو أداة تخطيط رسومية تستخدم لعرض مهام المشروع ومدتها الزمنية.
- يمكن رسم مخطط غانت يدوياً أو بواسطة برنامج رسم.
- المحور الأفقي للمخطط يعبر عن الفترة الزمنية (ممثلاً بوحدات الأيام أو الأشهر أو الأسابيع)
- أما المحور العمودي فيمثل المراحل (أو المهام أو النشاطات) كما في الشكل 9.



شكل 9: مخطط غانت

- كما نرى في الشكل 9 فإن مخطط غانت يتسم بالمراحل المتداخلة مثل مرحلتي التنفيذ والاختبار.
- في الشكل 10 نرى مثالاً على مخطط يدوي لمشروع يتكون من 6 نشاطات حيث نعبر عن مدة كل نشاط بالأسابيع.



شكل10: مثال أخر على مخطط غانت

- لاحظ أن في المثال الثاني لمخطط غانت أنه يشمل على نشاطات مرحلتي التخطيط والتحليل فقط.
- لو مثلنا مهام النشاطات الأخرى لأصبح المخطط طويلاً جداً.
- لحل هذه المشكلة يمكن استخدام جدول الخط الزمني ليمثل مهام المشروع بالتفصيل.
- مثلاً يمكن أن يبين الفترة الزمنية بالأيام بدلاً من أسابيع و أشهر.

6.4. جدول الخط الزمني Time-Line Table:

- هذا الجدول هو عبارة عن أداة تخطيط و متابعة ذات صفوف للمهمات ، و أعمدة لتواريخ البداية والنهاية المتوقعة والفعلية , مع امكانية إضافة عمود لمن يقومون بكل مهمة.

المهمة	البداية المتوقعة	النهاية المتوقعة	البداية الفعلية	النهاية الفعلية

جدول الخط الزمني Time-Line Table

- يمكن إضافة عمودين لهذا الجدول الأول للتكلفة والثاني للتكلفة الفعلية.
- في جدول الخط الزمني نقوم بملء أعمدة البداية والنهاية المتوقعة قبل بداية المشروع.
- البداية والنهاية الفعلية لكل مهمة يمكن أن تعرف فقد أثناء سير عمل المشروع.
- يقوم مدير المشروع بملء البيانات الفعلية عند وقوعها. هده العملية تساعد في إعادة التخطيط للمشروع ومعرفة أسباب التأخير.

7.4. تحليل المخاطر Risk Analysis:

- تحليل المخاطر أصبح من المواضيع المهمة في تخطيط المشاريع ، فهو ذو علاقة وطيدة مع نجاح المشروع أو فشله.
- ومع أن أدوات وطرق إعداد المشاريع قد تطورت تطوراً هائلاً في الوقت الحالي ، إلا أن المشاكل و المخاطر قد تكون موجودة في المشروع مما يؤثر سلباً عليه.
- هذه المشاكل أو المخاطر قد تكون ذات علاقة بجدولة المشروع أو تقدير التكاليف أو الجودة.
- بعض من هذه المشاكل قد لا يمكن تجنبها و لكن يمكن التقليل من مخاطرها.
- المخاطر هي عبارة عن مشكلة يمكن أن تسبب في خسارة أو تهديد نجاح المشروع. بعبارة أخرى ، إنها مشكلة كامنة سيكون لها تأثير سيء على المشروع.

هذا التأثير قد يكون على:

- تكلفة المشروع.
- الجدول الزمني للمشروع.
- جودة المنتج.

تحليل المخاطر هو عبارة عن عملية تحديد و معالجة المخاطر لمشروع ما. لماذا يجب علينا أن نعرف مخاطر المشروع قبل تنفيذه؟

- ✓أولاً لأن المخاطر تهدد نجاح المشروع.
- ✓ثانياً لأن هذه المخاطر المحتملة إذا تحققت فإن تسليم المشروع سيتأخر وتكلفته ستزداد.

لإنجاز تحليل المخاطر يجب أن:

- نعد قائمة بالمخاطر.
- نحدد احتمال كل مخاطرة(نسبة حدوث المخاطرة).
- نحدد توابع كل مخاطرة.
- نحدد استراتيجية الحل لكل مخاطرة.

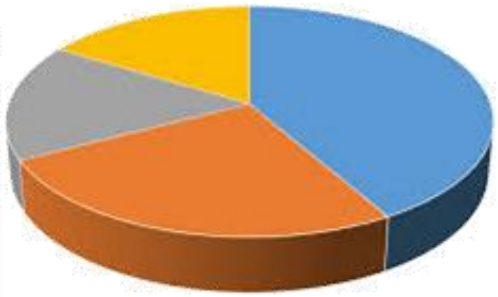
الجدول التالي يبين نموذجاً لتحليل المخاطر:

المخاطرة	الاحتمال	التأثير	الحل
تغير المتطلبات	%80	حرج	• استخدام العرض المبدئي
التدريب على الأدوات غير كافي	%70	سهل التدبير	• التدريب. • التركيز على الدراسة.
غياب العاملين بالمشروع	%60	حرج	• المزيد من التدريب • تحسين اوضاع العاملين
التأخر في تسليم المنظومة	%60	حرج	• إعادة استعمال الوحدات البرمجية. • استعمل أدوات مناسبة لتقدير الزمن.

The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة الخامسة

5. العرض التجريبي Prototyping

مفهوم العرض التجريبي

- العرض التجريبي هو عبارة عن عملية اعداد وعرض منظومة كعينة للزبون تمثل المنظومة المطلوبة ولكن بشكل مبدئي.
- باستخدام العرض التجريبي (المبدئي)، يعرض معد المنظومة على الزبون منظومة ذات وظائف محددة في مرحلة مبكرة وإن كان ذلك في مستوى منخفض من الأداء و الاعتمادية.
- الفكرة هنا هي جعل الزبون يقرر شكل المنظومة النهائي في مرحلة مبكرة.
- يقوم الزبون بمشاهدة شاشات (نماذج) وتقارير وحوارات تفاعلية، والأهم من ذلك بعض الوظائف المطلوبة للنظام. ثم تؤخذ ملاحظاته و اقتراحاته بعناية فائقة.
- الميزة هنا أن الزبون يعطي فرصة لتغيير متطلباته في مرحلة مبكرة من المشروع حيث تعتبر تكلفة التغيير قليلة نسبياً.
- تستخدم طريقة العرض التجريبي كحل لأهم مشكلة في النموذج التتابعي (نموذج شلال المياه)، ألا و هي مشاكل نقص المتطلبات أو عدم دقتها.

1.6. لماذا نحتاج إلى العرض التجريبي؟

1. لتحديد وتنقيح متطلبات المستخدم بوضوح أكثر.
2. لنعطي الزبون صورة واضحة عن شكل المنظومة بعرض عينات منها مثل شاشات وتقارير وحوارات تفاعلية.
3. يعتبر العرض التجريبي عملي أكثر عندما تكون متطلبات الزبون غير واضحة.
4. لعرض أو تسويق او بيع منتوج برمجي وتوضح الفوائد التي يمكن جنيها منها.
5. لتوضيح الجدوى الفنية للمنظومة.

2.6. أنواع العرض التجريبي:

العرض التجريبي الساكن:

- في هذا النوع من العرض التجريبي يتم عرض الشاشات ونماذج مطبوعة من التقارير على الزبون ولكن دون أن يتمكن من إدخال البيانات أو الحصول على تقارير مباشرة من جهاز الحاسب.
- بعبارة أخرى فإن نماذج العرض التجريبي هي ببساطة تصاميم وليست منظومات قابلة للتنفيذ باستخدام لغة برمجة مرئية ذات شاشات وتقارير مثل لغة البيسك المرئي.
- بعد دراسة هذه الشاشات والتقارير، يستطيع الزبون أن يطلب أي تغييرات لهذه الشاشات وتصاميم التقارير.

2.6. أنواع العرض التجريبي...

العرض التجريبي المحذوف:

- الغرض من العرض التجريبي هو تحديد متطلبات المستخدم ففي البداية يتم إعداد منظومة ذات وظائف محددة في وقت قصير باستخدام لغات البرمجة (فيجوال بيسك مثلا) ودون الاهتمام بجودة المنظومة ثم يتم تنقيحها مع الزبون.
- ولكن يتم التخلص من هذه العينة البرمجية بعد وضوح متطلبات المستخدم ، ثم يتم الانتقال إلى المرحلة التالية (مرحلة التصميم) من النموذج التتابعي (الشلالي).

نموذج العرض التجريبي التطوري

- الغرض من هذا النموذج هو الوصول إلى المنظومة النهائية ابتداء من نسخة أولية بوظائف محدودة ثم إعداد عدة إصدارات متعاقبة بوظائف أكثر حتى يتم تلبية كل المتطلبات ويتم إعداد المنظومة النهائية.
- وذلك يعني أن هذا النموذج يمكن استخدامه كبديل للنموذج التتابعي.

نشاطات اعداد نموذج العرض التجريبي:

يمكن تلخيص نشاطات اعداد نموذج العرض التجريبي في الخطوات التالية:

1. يتم التعرف على متطلبات الزبون من قبل معد المنظومة بمساعدة المستخدم و الزبون.
2. يتم تصميم وتنفيذ منظومة العرض التجريبي للزبون لتقوم بأداء الوظائف المهمة (وظائف محددة). ومن المناسب هنا استعمال لغة برمجة مرئية مثل فيجوال بيسك لإعداد هذه المنظومة.
3. يتم تجربة طريقة عمل منظومة العرض التجريبي من قبل المستخدم.
4. يعطى الزبون فرصة استعمال منظومة العرض التجريبي ونتوقع منه في المقابل أن يعطي ملاحظات او يقترح تغييرات.
5. من التغييرات المطلوبة في الخطوة 4 يمكن لمعد المنظومة البرمجة أن يجهز النسخة اللاحقة.
6. يتم تكرار الخطوات 3 و 4 و 5 حتى يتم تحقيق متطلبات المستخدم تماماً و نحصل على المنظومة النهائية.

3.6. مزايا العرض التجريبي:

1. يقلل من مشاكل تحديد المتطلبات.
2. يوفر مشاركة ودعم أآبر من طرف الزبون في المراحل المبكرة من المشروع (التحليل و التصميم).
3. يقلل من المخاطرة في تأخير تسليم المشروع.
4. يمكن استعماله كقاعدة لكتابة المواصفات بجودة عالية.
5. يمكن استعماله لتدريب المستخدمين قبل تسليم المنظومة.
6. يقلل من وقت اعداد المنظومات.
7. يقلل من التكلفة الإجمالية لإعداد المنظومات
8. يقلل من احتمال فشل المشروع.
9. طريقة أفضل للتواصل بين المستخدم و معد المنظومة.

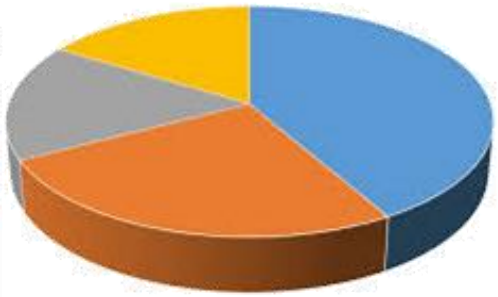
4.6. عيوب العرض التجريبي:

1. أحياناً يحدث التباس للزبون بأن العرض التجريبي هو المنتوج النهائي. وفي هذه الحالة يكون من الصعب إقناع الزبون أن منظومة العرض التجريبي هي للعرض فقط (منظومة تجريبية)، وقد يطلب بالإسراع في استلام المنظومة بعد اجراء تعديلات بسيطة عليها.
2. للحصول على منظومة العرض التجريبي بسرعة، يستخدم معد المنظومة عادة أدوات وخوارزميات بسيطة وقليلة الكفاءة وهذه الأدوات والخوارزميات عادة ما تصبح هو الأدوات الأساسية في إعداد المنتوج النهائي مما يؤدي إلى إعداد منظومة قليلة الجودة.
3. تخلق إحساسا خادعاً من التفاؤل عن المنظومة.
4. لا ينتج عنها توثيق وهذا يجعل من صيانة المنظومة أمراً صعباً.

The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة السادسة

6. تحليل المتطلبات Requirement Analysis

تحليل المتطلبات:

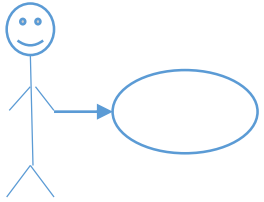
- يبدأ المحلل في هذه الخطوة بعمل تنظيم للمتطلبات لغرض التوصل لفهم واضح للنظام القائم والنظام المقترح.
- يوجد مجموعة من الأدوات يستخدمها المحلل لتنفيذ عملية تحليل المتطلبات، سنقوم بدراسة الأداتين:

➢مخطط حالات الاستخدام USECASE

➢مخطط العلاقات Entity Relationship Diagram (ERD)

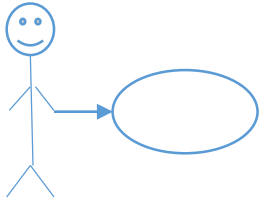
USE CASE DIAGRAM

مخطط حالات الاستخدام



ما هو USECASE ؟

- هو أحد مخططات لغة النمذجة الموحدة UML (Unified Modeling Language).
- يستخدم لوصف سلوك النظام من وجهة نظر المستخدم. فهو ذو فائدة خلال مراحل التحليل و التطوير، ويساعد في فهم المتطلبات.
- هو وصف لسيناريو النظام على اعتبار المتطلبات هي الأحداث والفاعلين هم الممثلون.
- يكون المخطط سهلاً للاستيعاب. مما يمكّن كلا من المطوّرين (محلّلون، مصمّمون، مبرمجون، مختبرون) والمستفيدين (الزبون) من الاشتغال عليه.
- تستنتج حالات الاستخدام من المتطلبات الوظيفية من وثيقة المتطلبات حيث يمثل كل متطلب وظيفي حالة استخدام ترتبط بمفعل معين كما تفيد القيود في استنتاج حالات استخدام مثل التحقق وغيرها..

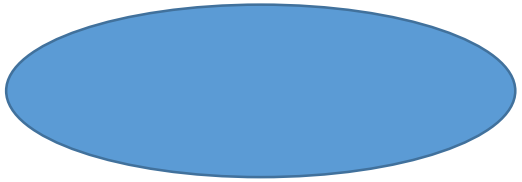


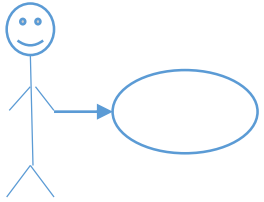
مكونات USECASE :

إن UseCase يتكون من ثلاث أشياء رئيسية هي:

1- الحالة Case :

- **وهي عبارة عن مجموعة عمليات تنفذ عن طريق المستخدم تخدم في النهاية عمل واحد فقط.**
- **وتمثل في الرسم بشكل بيضاوي يكتب فيه اسم العملية التي تنفذها الحالة Case.**
- **يشترط باسم العملية أن تكون فعل أو مصدر.**

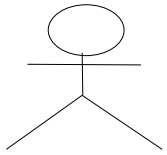




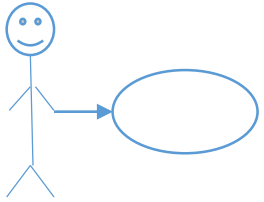
مكونات USECASE:

2- الفاعل أو الممثل Actor :

- **هو أي شيء يتعامل مع النظام من الخارج قد يكون المستخدم المباشر أو شخص متأثر بالنظام أو حتى نظام آخر يتعامل مع النظام.**
- **هو الذي يمكنه بدء أو تفعيل الحالة case.**
- **ويرمز له برمز شخص ويكتب تحته اسم الدور الذي يلعبه.**



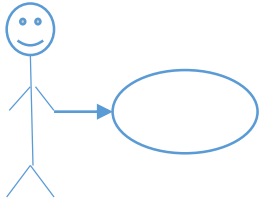
User



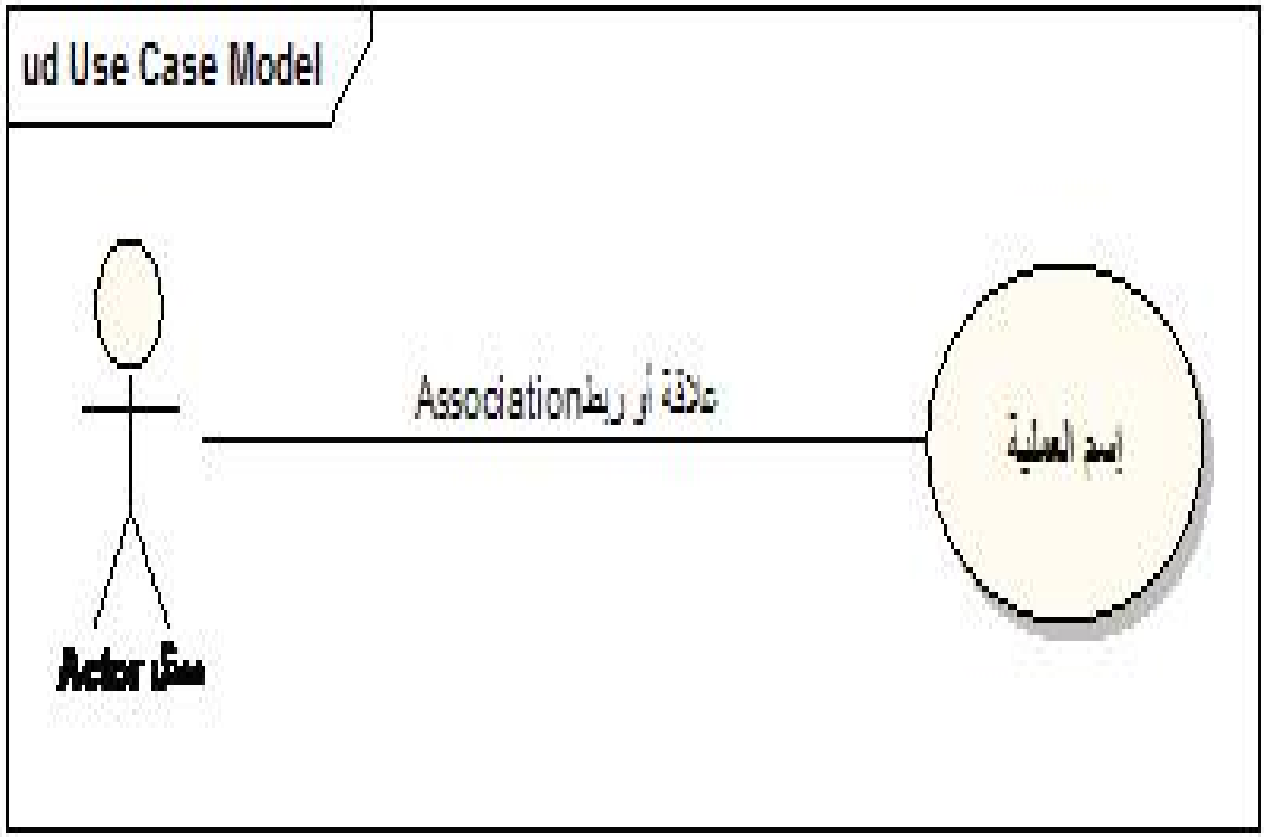
مكونات USECASE:

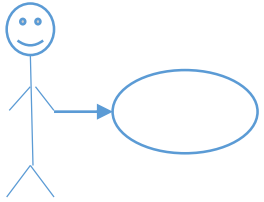
3 - العلاقة Association or Relation :

- **وهي ما يربط بين الحالة وبين الممثل .**
- **ويرمز لها بخط وله أنواع تجدها في وثائق UML.**



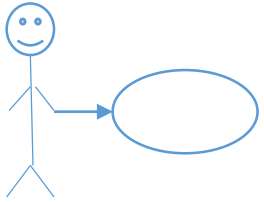
مثال عن المكونات:



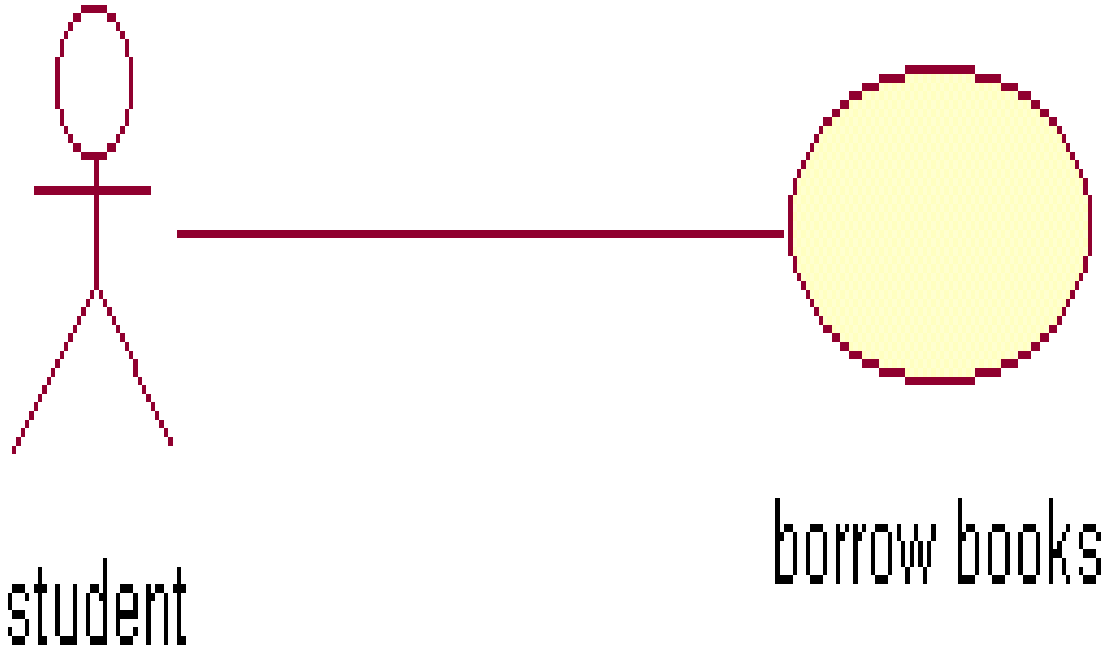


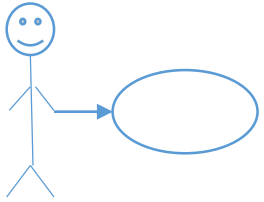
توصيف حالة الاستخدام:

اسم الحالة	
المفعلين (actors)	
الوصف (description)	
الشروط السابقة (precondition)	
التدفق الرئيسي (mainflow)	
الشروط اللاحقة(postcondition)	



مثال: نظام مكتبة:

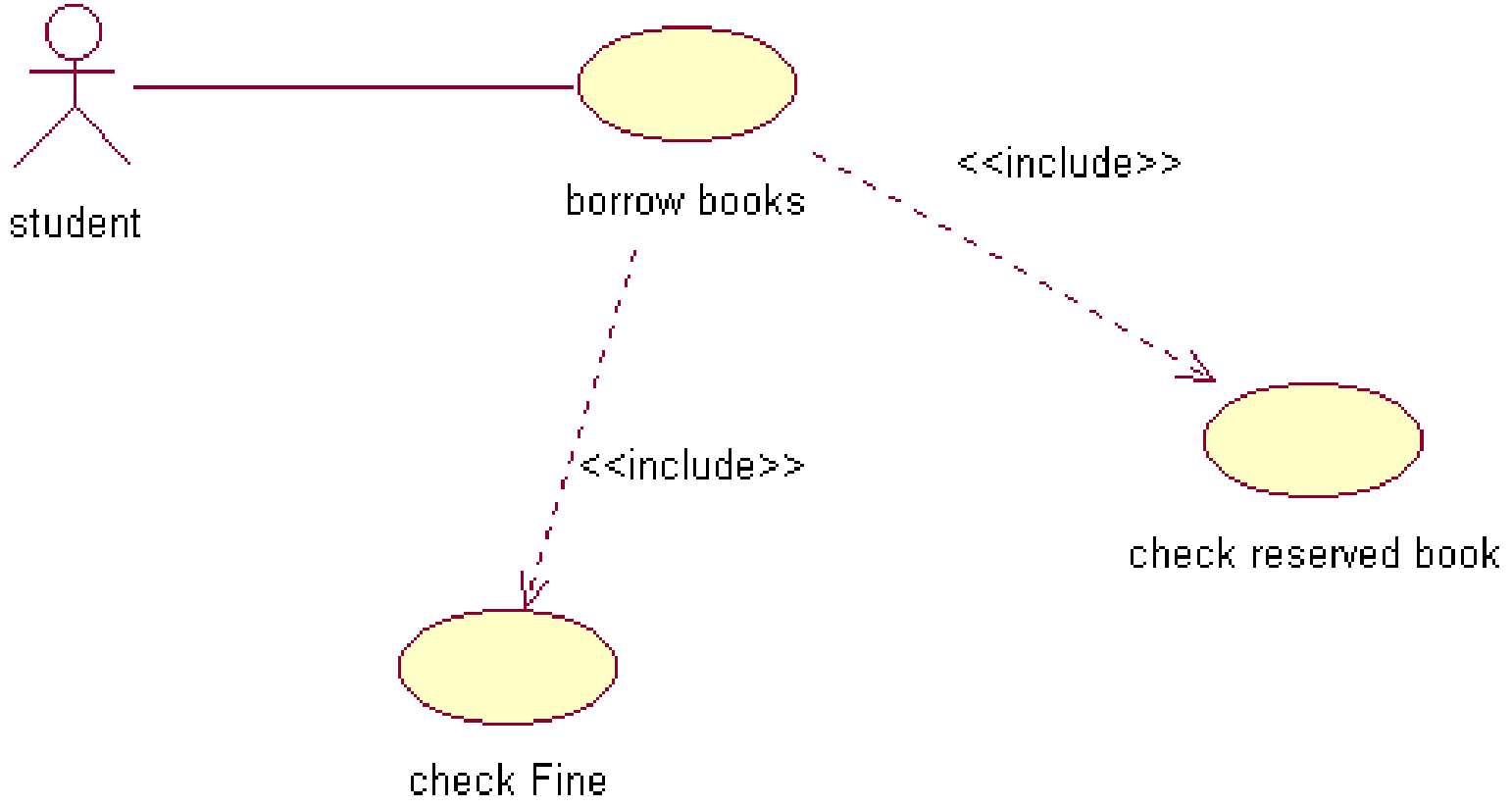


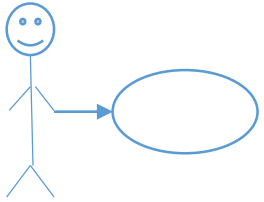


علاقة التضمين Including Relationship:

- علاقة تصل بين حالة case أساسية مع حالة case متضمنة.
- هذه العلاقة تحدد كيف أن السلوك في الحالة المضمنة مستخدم في الحالة الأساسية.
- لا تنتهي الحالة الأساسية إلا بانتهاء الحالات المضمنة فيها.
- كمثال على هذه العلاقة نعود إلى مثال المكتبة.

Including Relationship

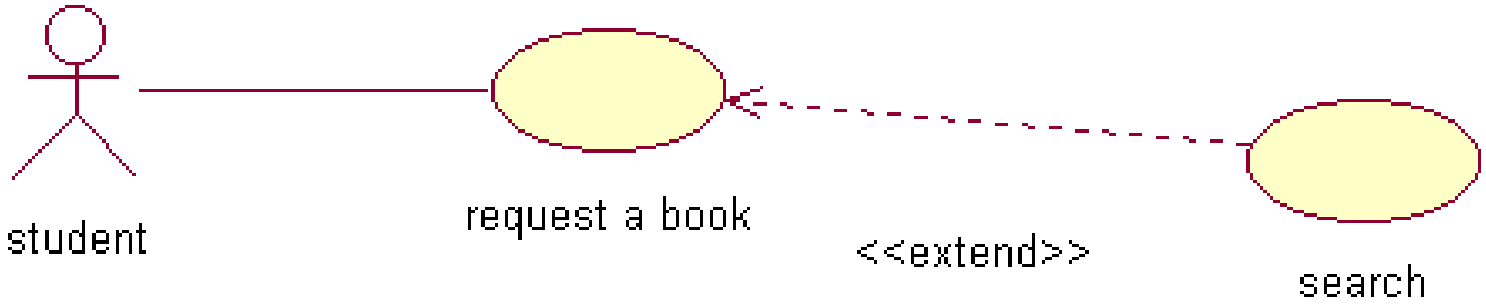




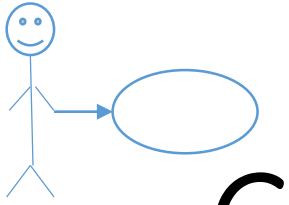
علاقة الامتداد Extend Relationship:

- هذه العلاقة تبين كيف أنه من الممكن إدراج وظيفة أو نشاط حالة في حالة أخرى.
- لا يشترط أن تتم الحالة الأساسية بإتمام الحالة الممتدة عنها.
- الحالة الممتدة يمكن أن تحدث أو لا، فهي تحدث فقط في حالات خاصة استثنائية أو بتحقق شروط معينة.
- كمثال على هذه العلاقة نعود إلى مثال المكتبة.

Extend Relationship



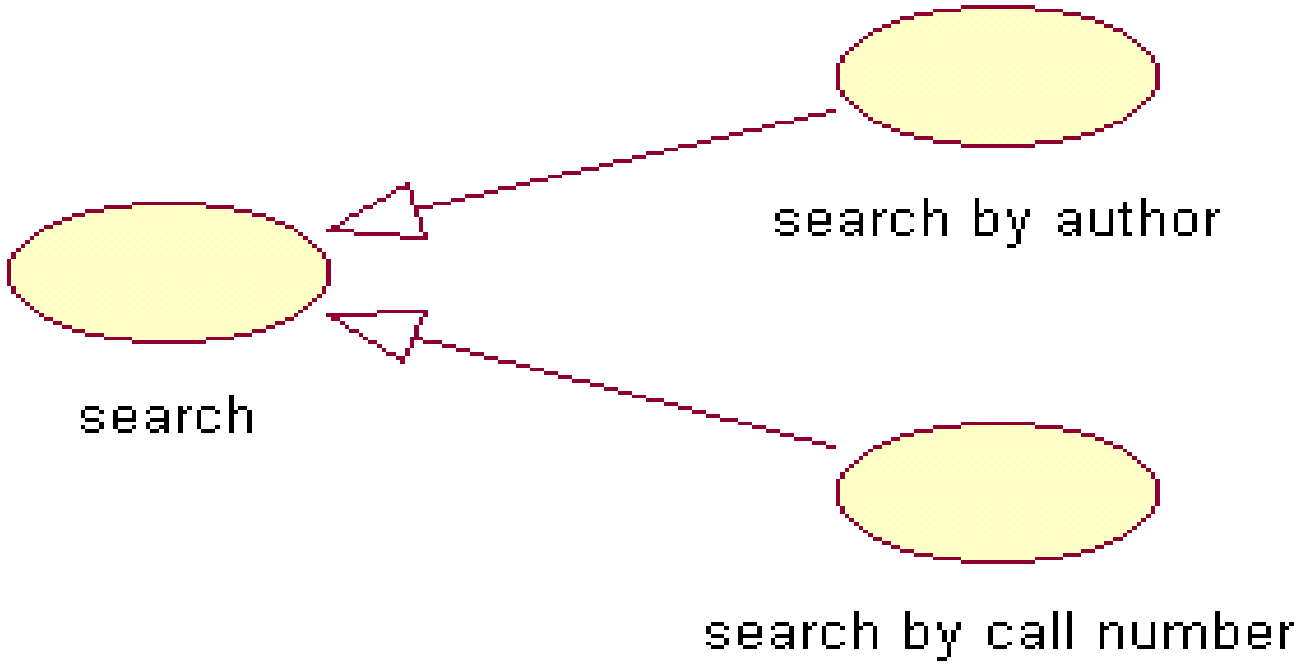
نلاحظ أنه في هذا النوع من العلاقة ليس من الضروري كل عملية طلب لكتاب القيام بعملية البحث فمن الممكن معرفة الطالب لرقم الكتاب و الاستغناء عن عملية البحث

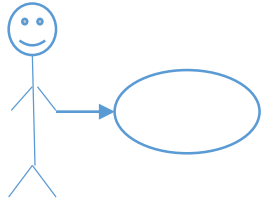


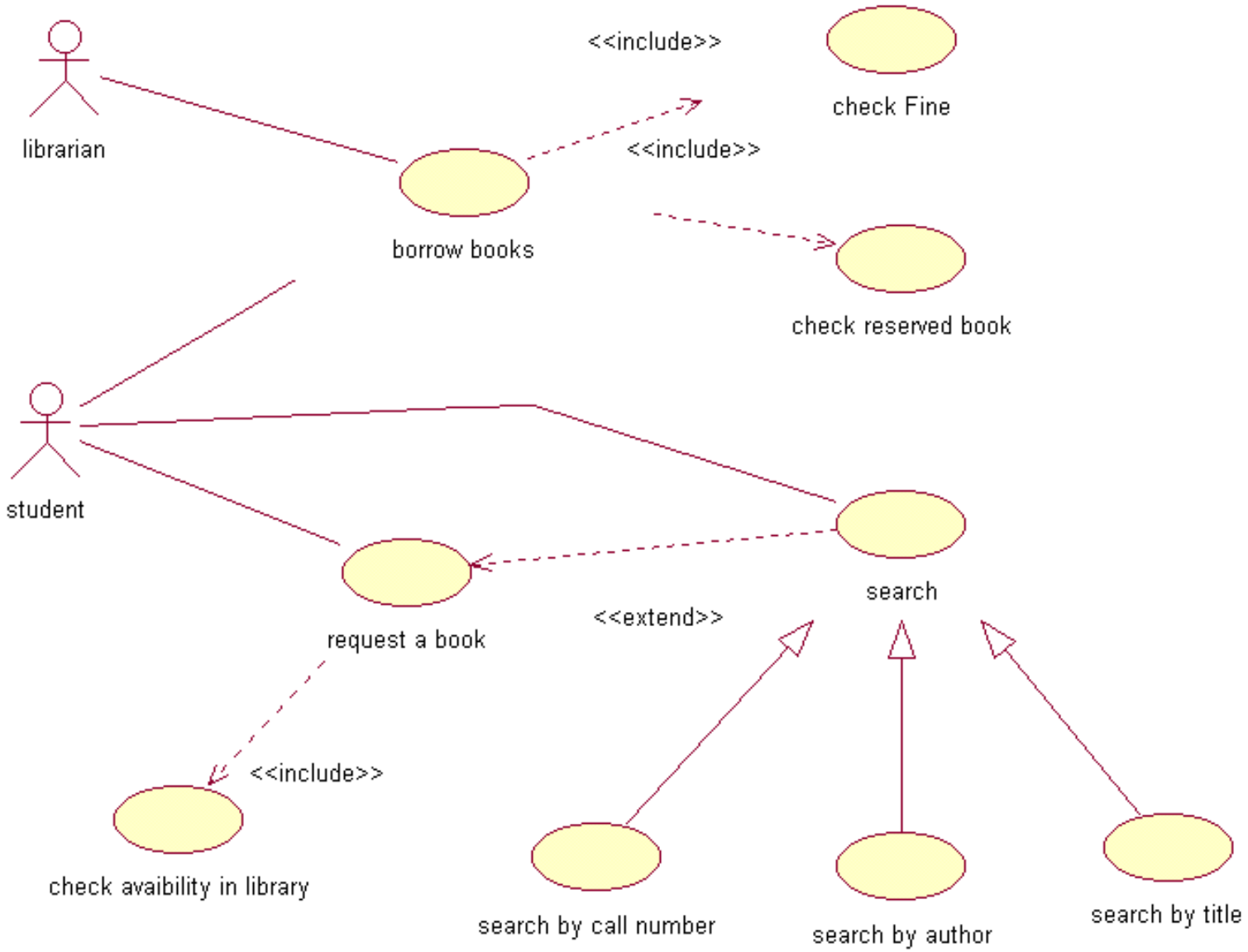
علاقة التعميم أو الوراثة *Generalization relationship*

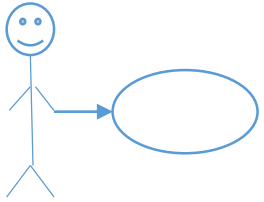
- هذه العلاقة تعني أن الحالة case الابن ترث سلوك ومعنى الحالة case الأب.
- ومن الممكن للابن أن يضيف أو يعيد كتابة نشاط أو سلوك الأب.
- أي أن الحالة الابن هي تخصيص لحالة الأب العامة.
- كمثال على هذه العلاقة نعود إلى مثال المكتبة.

Generalization relationship



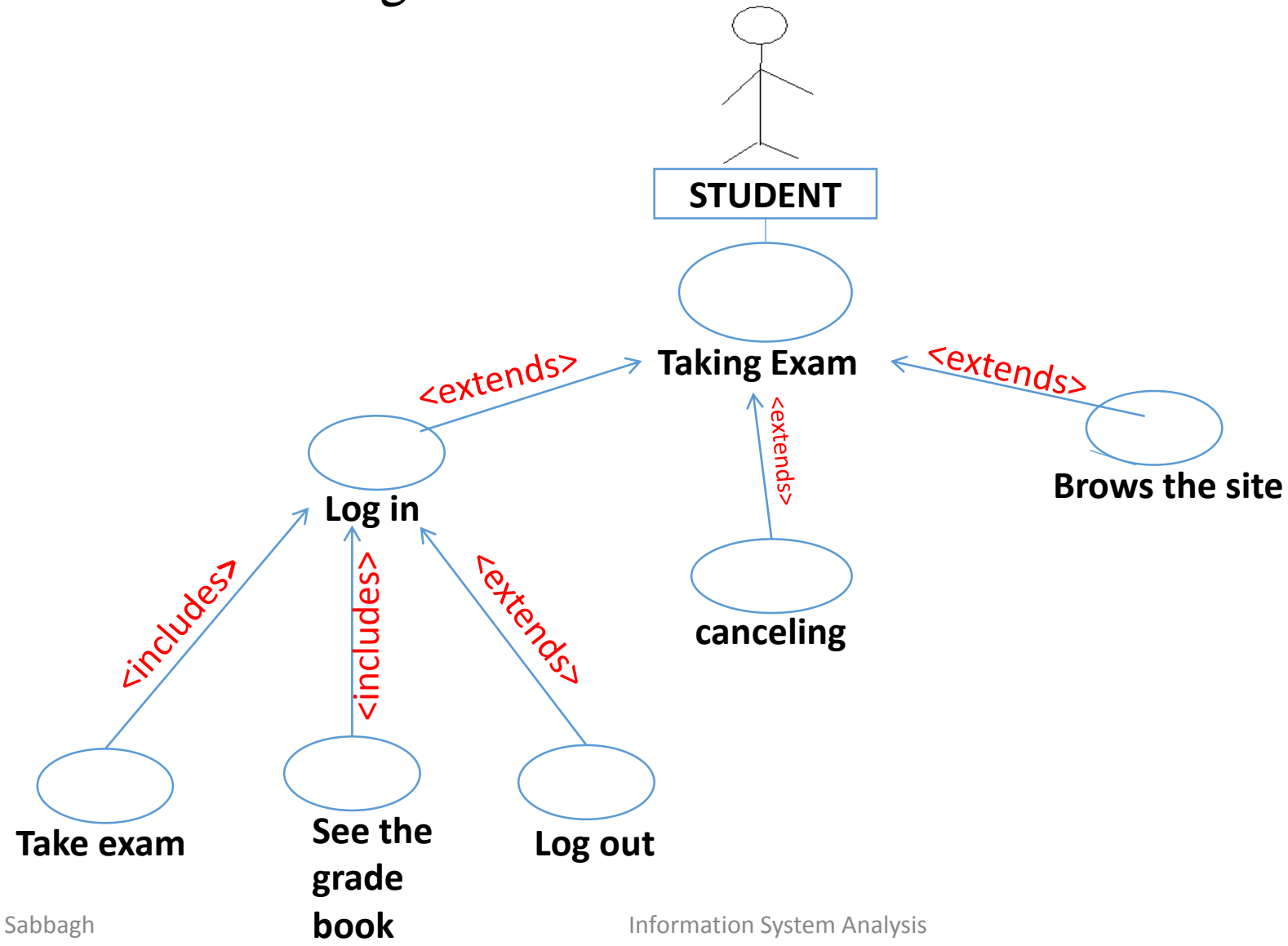


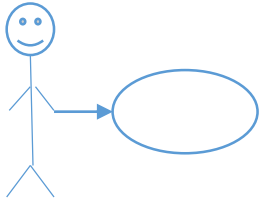

librarian
<<include>>
check Fine
<<include>>
borrow books
check reserved book
student
search
request a book
<<extend>>
<<include>>
check avaibility in library
search by call number
search by author
search by title



Use Case Diagram:

مثال:

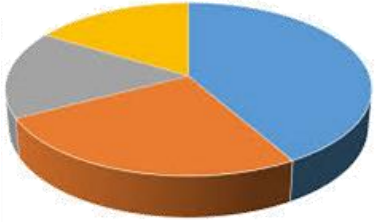




The End

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



المحاضرة السابعة

7. تحليل متطلبات البيانات Data Requirements Analysis

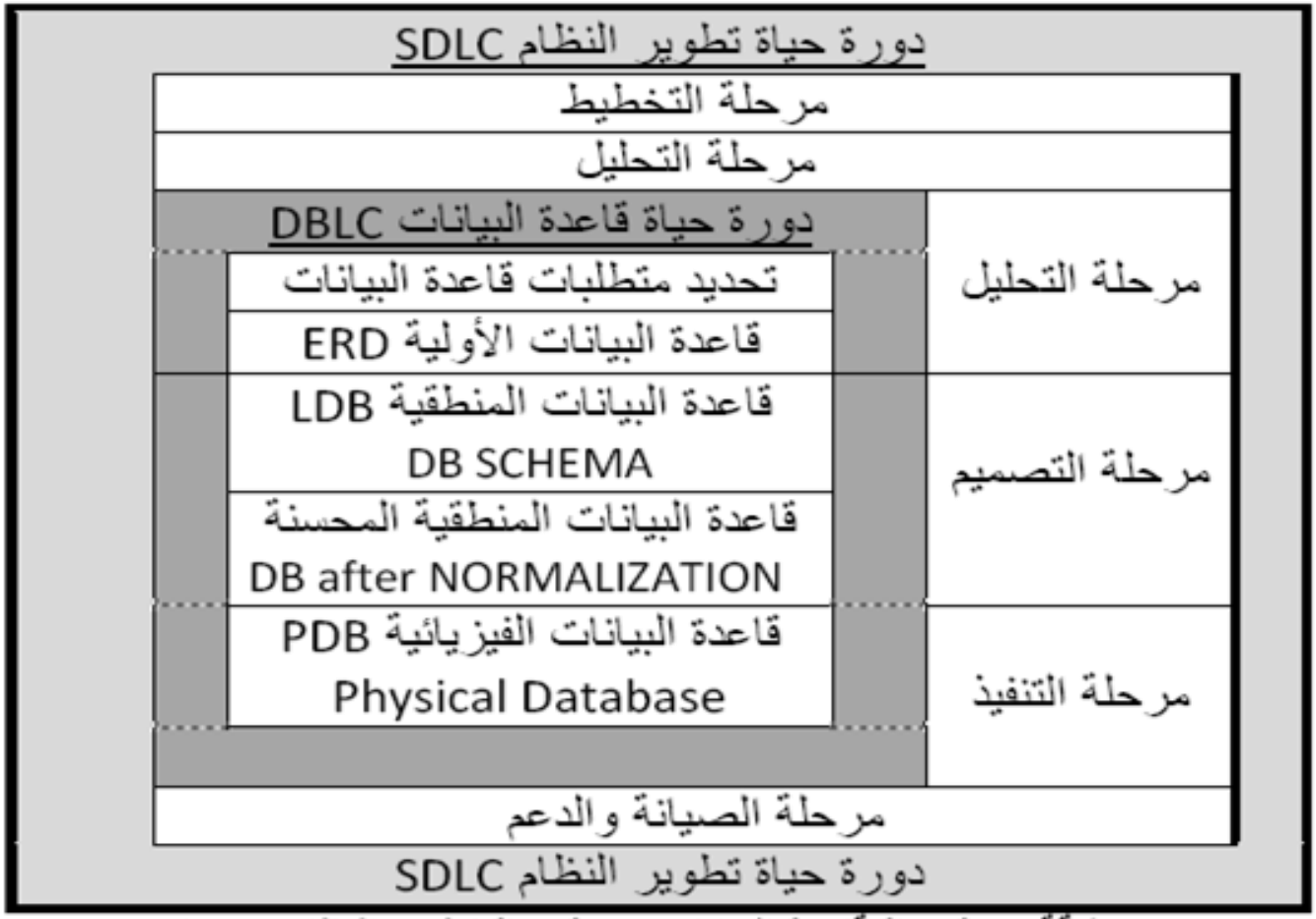
في هذ المحاضرة

- دورة حياة قاعدة البيانات Database Life Cycle (DBLC)
- ما هو مخطط الكينونة العلائقي؟
- مكونات مخطط الكينونة العلائقي
  - الكيان
  - الصفة وانواعها
  - المفتاح الرئيسي والمفتاح الجزئي
  - الكيان الضعيف
  - العلاقات و أنواع العلاقات
  - أنواع القيود على العلاقات
- مثال1(مخطط كينونة علائقي لجامعة)
- مثال2 (مخطط كينونة علائقي لشركة)

دورة حياة قاعدة البيانات Database Life Cycle (DBLC)

- إن عملية تطوير قاعدة البيانات تمر بمجموعة من المراحل، هذه المراحل المتتالية تسمى بدورة حياة قاعدة البيانات.
- هذه المراحل أو دورة الحياة تمر بصورة متزامنة ضمن مراحل دورة حياة نظام المعلومات، كما يوضح الشكل التالي:

Database Life Cycle (DBLC) دورة حياة قاعدة البيانات



علاقة دورة حياة قاعدة البيانات بدورة حياة تطوير النظام عموما

دورة حياة قاعدة البيانات Database Life Cycle (DBLC)

تتكون **دورة حياة قاعدة البيانات من المراحل التالية:**

1. **تحديد المواصفات والمتطلبات الخاصة بقاعدة البيانات،** وهي مرحلة جزئية ضمن جمع مواصفات ومتطلبات نظام المعلومات في مرحلة التحليل.
2. **إعداد قاعدة البيانات الأولية،** وفيها يتم تصميم نموذج اولي للبيانات بواسطة مخططات علاقة-كينونة (E-RD).
3. **تصميم قاعدة البيانات المنطقية،** تحويل قاعدة البيانات الأولية، أو مخطط علاقة-كينونة إلى مخطط الاسكيما، وذلك بإتباع قواعد التحويل

دورة حياة قاعدة البيانات Database Life Cycle (DBLC)

تتكون دورة حياة قاعدة البيانات من المراحل التالية:

4. **تحسين قاعدة البيانات المنطقية**، وذلك بتطبيق قواعد تطبيع البيانات Normalization التي تهدف إلى تقليل تكرارية البيانات، من أجل رفع كفاءة قاعدة البيانات ما أمكن.
5. **تنفيذ قاعدة البيانات الفيزيائية physical database**: وفي هذه المرحلة يتم كتابة أكواد إنشاء قاعدة البيانات بلغةSQL ، ويحدد فيها بنية الجداول ونوع بيانات الحقول والمفاتيح الاساسية والاجنبية وباقي شروط تصميم قاعدة البيانات، ثم تنفيذ ذلك ضمن مدير قاعدة بيانات DBMS مناسب، مثل (oracle, access, sqlserver, mysql .... etc)

مخطط الكينونة العلائقي ERD

ما هو مخطط الكينونة العلائقي؟

هو نموذج عالي المستوي يقوم بعرض بناء البيانات، ويتم استخدام هذا النموذج أثناء مرحلة التصميم المفاهيمي للنموذج الأولى،وينتج عن ذلك النموذج ألأولي ، لقاعدة البيانات، والذي عن طريقة نقوم بتصميم مخطط قاعدة البيانات، ويتم تمثيل بناءالبيانات والقيود المطلوبة عليها باستخدام إشكال رسومية سهلة ومحددة.

مكونات مخطط الكينونة العلائقي

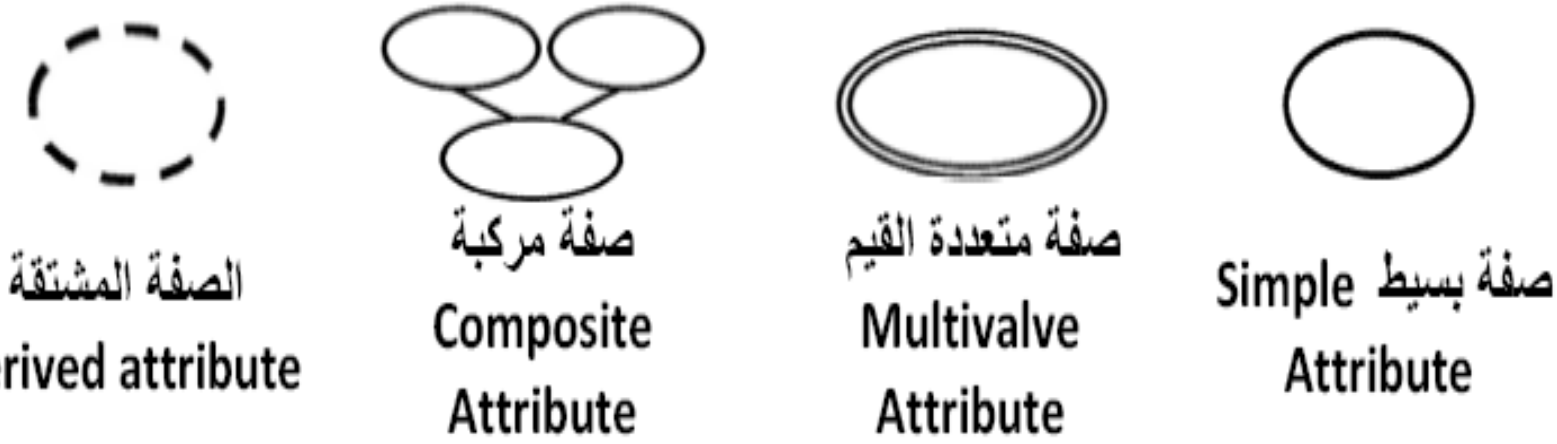
- **الكيان أو الكينونة(Entity):** هو الوحدة الأساسية التي يتم تمثيلها بنموذج الكينونة/العلاقة ويشير هذا الكيان إلي "شيء" حقيقي في الحياة سواء كان له وجود فعلي مثل (طالب – موظف – سيارة - ... الخ) أو وجود منطقي مثل (شركة – وظيفة – مقرر -... الخ).
- ويتم تمثيل الكيان باستخدام شكل مستطيل يكتب داخله اسم الكيان أو الكينونة

مكونات مخطط الكينونة العلائقي

- **الصفة (Attribute):** هي صفة معينة تصف الكيان وتكون تابعة له في المخطط ، مثل اسم الموظف ، عمر الطالب، مرتب موظف، درجة طالب، عدد الساعات الدراسية لمقرر، لاحظ أن كل صفة تتبع كينونة أو كيانا معينا.
- الصفة يمكن أن تكون صفة بسيطة – أو صفة مركبة
- الصفة يمكن أن تكون أحادية القيمة – أو صفة متعددة القيم
- يوضع خط تحت اسم الصفة أو الصفات التي تمثل مفتاح الكيان
- **الصفة المشتقة** :وهي صفة بسيطة يمكن الحصول عليها من صفة بسيطة أخرى، مثل الصفة عمر الطالب التي يمكن الحصول عليها من تأريخ الميلاد

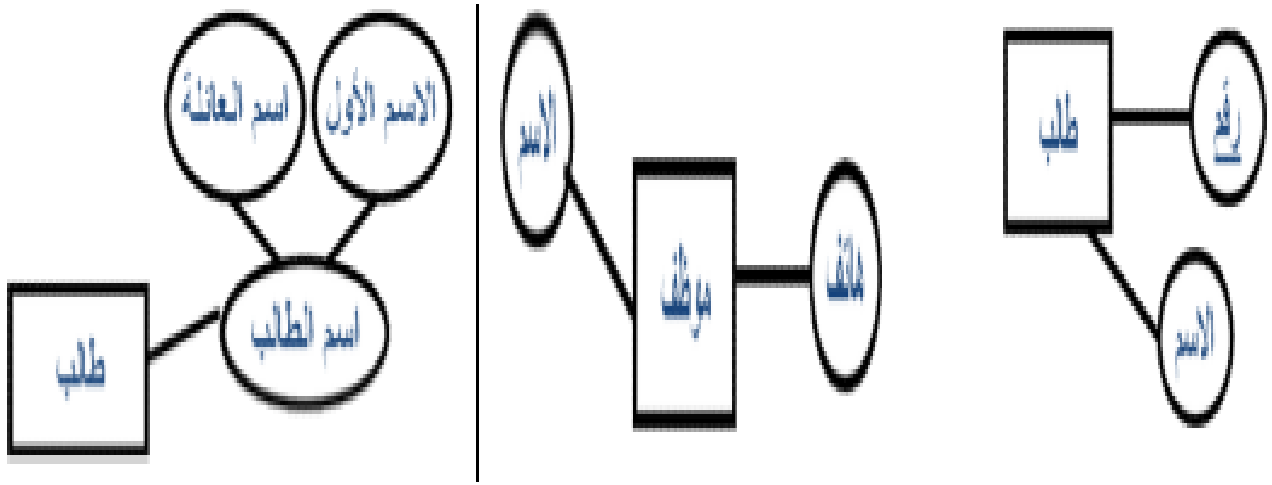
مكونات مخطط الكينونة العلائقي

- يتم تمثيل الصفة باستخدام شكل بيضاوي، والصفة متعددة القيم يتم تمثيلها بالشكل البيضاوي المزدوج



شكل تمثيل الصفات المختلفة للكيان

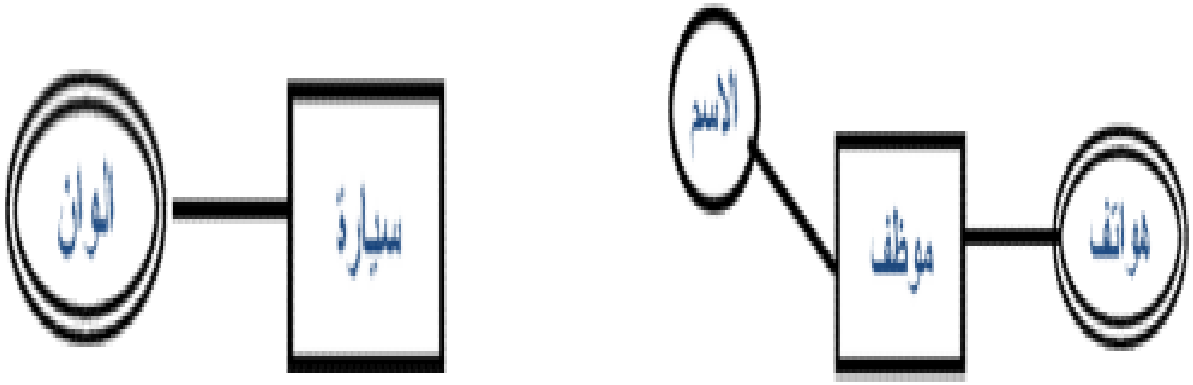
مكونات مخطط الكينونة العلائقي



الصفة البسيطة مثل اسم طالب، هاتف موظف. الصفة المركبة مثل اسم الطالب(الاسم الأول – اسم العائلة)

شكل مثال توضيحي

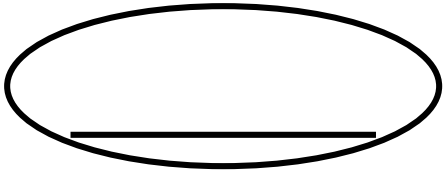
مكونات مخطط الكينونة العلائقي



الصفة متعددة القيم مثل ألوان وهواتف، فالكيان موظف (قد يكون له هاتف أو اثنين أو أكثر)
والكيان سيارة (قد تكون من لون واحد أو أي عدد من الألوان)
شكل مثال توضيحي

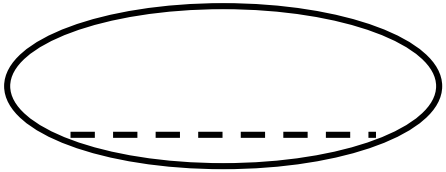
مكونات مخطط الكينونة العلائقي

- **صفة المفتاح الرئيسي(Primary Key Attribute) :** هي تلك الصفة المميزة للكيان ، ويتم تمثيلها بشكل بيضاوي مع خط تحت إسم الصفة، وقد يكون للكيان أكثر من صفة لتمثل معا المفتاح الرئيسي



مكونات مخطط الكينونة العلائقي

- **صفة المفتاح الجزئي (Partial Key Attribute) :** هي تلك الصفة التي لم ترقى لتكون مميزة للكيان ،ولكنها صفة قد تساعد في تكوين صفة مميزة إذا تم ضمها إلى صفة مميزة من كيان آخر، ويتم تمثيلها بشكل بيضاوي مع خط متقطع تحت إسم الصفة

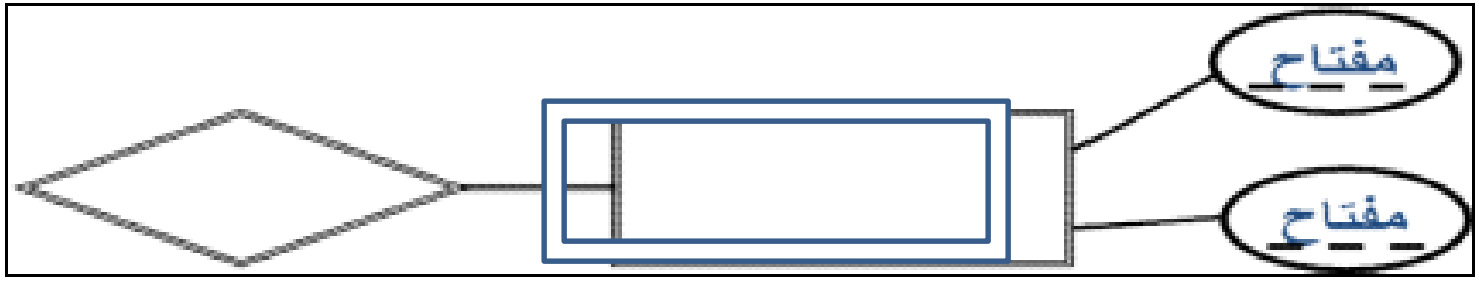


مكونات مخطط الكينونة العلائقي

- **الكيان الضعيف(Weak Entity) :** هو ذلك الكيان الذي ليس لديه مفتاح رئيسي يميز بياناته عن بعضها البعض، وعادة ما يقترن الكيان الضعيف بكيان قوي عن طريق علاقة تعريف تقوي، ويتم تمثيل الكيان الضعيف بمستطيل مزدوج

مكونات مخطط الكينونة العلائقي

- يرتبط المفتاح الجزئي بالكيان الضعيف، ليساعد فيما بعد بتكوين مفتاح رئيسي للكيان الضعيف كما سنرى في محاضرة لاحقة.



المفتاح الجزئي للكيان الضعيف

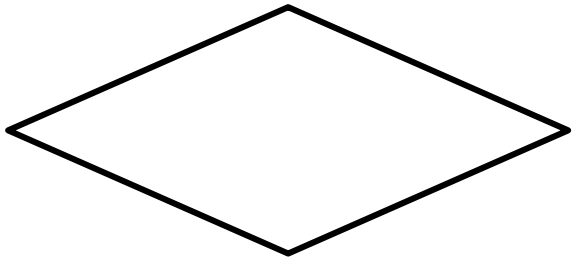
مكونات مخطط الكينونة العلائقي

- يرتبط المفتاح الجزئي بالكيان الضعيف، ليساعد فيما بعد بتكوين مفتاح رئيسي للكيان الضعيف كما سنرى في محاضرة لاحقة.

مثال توضيحي

مكونات مخطط الكينونة العلائقي

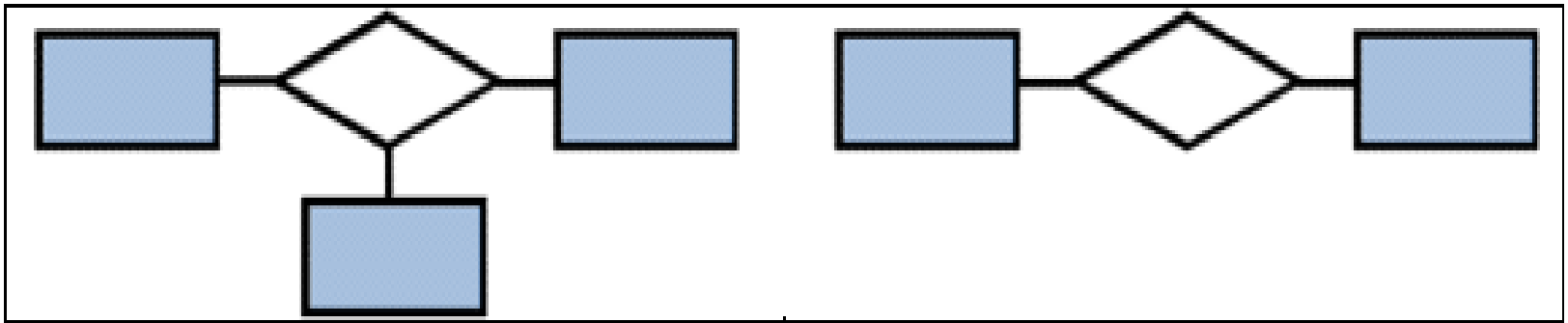
- **العلاقات (Relation):** العلاقة (R) بين مجموعة من الكيانات (E1,E2, ...En) هي مجموعة تمثل الارتباطات بين هذه الكيانات، كل وحدة في العلاقة (R) هي عبارة عن اتحاد أو ارتباط بين الكيانات المرتبطة بهذه العلاقة، بحيث أن هذه الوحدة تمثل بصف واحد من كل كيان مشارك في العلاقة.
- في نموذج الكينونة/العلاقة، يجب أن يتم تمثيل المرجعية من كيان إلى كيان آخر، باستخدام "علاقة" وليس كصفة في الكيان
- يتم تمثيل العلاقة في نموذج الكينونة/العلاقة باستخدام شكل المعين



.

مكونات مخطط الكينونة العلائقي

- **درجة العلاقة:** لكل علاقة درجة، وتتحدد هذه الدرجة بعدد الكيانات المرتبطة بهذه العلاقة (ثنائية – ثلاثية - ....)



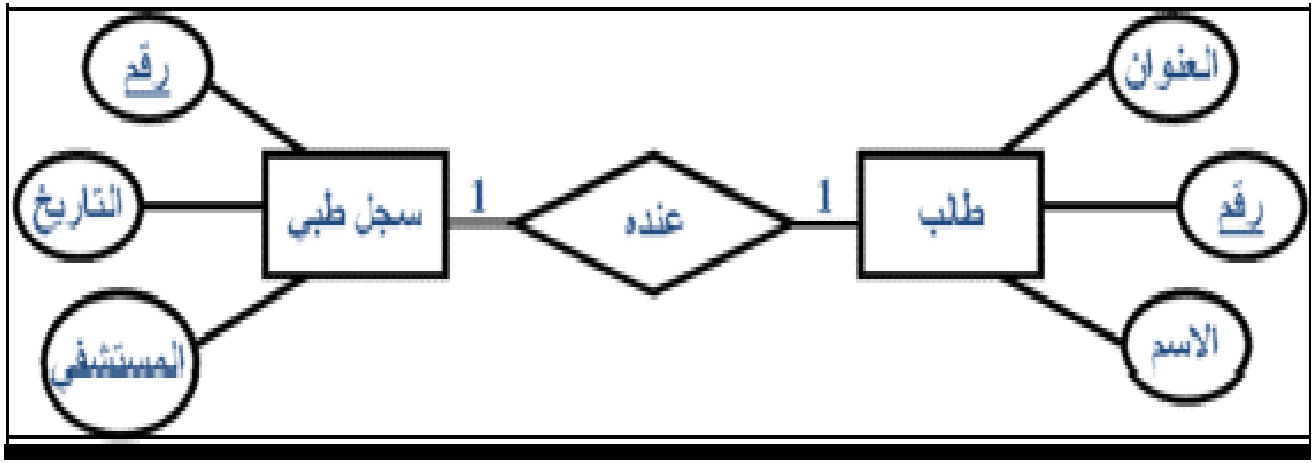
العلاقة الثنائية Binary والعلاقة الثلاثية Ternary

مكونات مخطط الكينونة العلائقي

- **نوع العلاقة(Cardinality Ratio):**المصطلح يعني نسبة الارتباط بين وحدات الكيان، التي ترتبط بنفس العلاقة، وفي العلاقة الثنائية بين كيانين، نوع العلاقة هو عدد الوحدات (Instances) في العلاقة التي يمكن أن يشترك فيها الكيان وهي ثلاثة أنواع:

1. **علاقة واحد- الى-واحد(one-to-one):** وفيها ترتبط وحدة واحدة من الكيان الأول بوحدة واحدة من الكيان الآخر على الأكثر، ويرمز لها بالرمز1:1
2. **علاقة واحد- الى-كثير(one-to-many) :** وفيها يمكن أن ترتبط وحدة واحدة من أحد الكيانات بأكثر من وحدة في الكيان الآخر، والعكس غير صحيح، ويرمز لها بالرمز 1:N
3. **علاقة كثير- الى-كثير(many-to-many) :** وفيها يمكن أن ترتبط أكثر من وحدة من الكيان الأول بأكثر من وحدة في الكيان الآخر، والعكس، أي يمكن لأي وحدة في الكيان الآخر أن ترتبط بأي وحدة في الكيان الأول، ويرمز لها بالرمز M:N

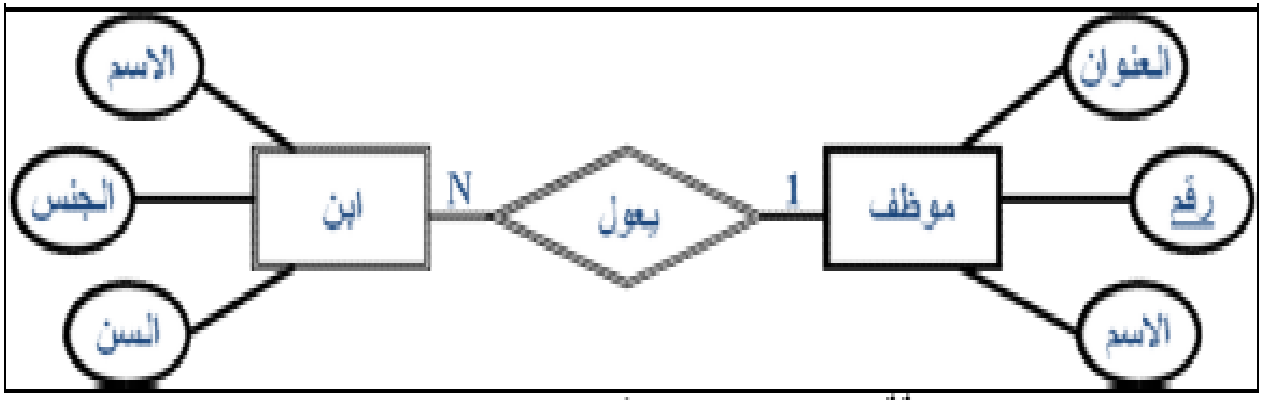
مكونات مخطط الكينونة العلائقي



علاقة ١:١ واحد-الى-واحد (one-to-one)

لاحظ أنه لكل طالب سجل طبي واحد (نوع العلاقة ١)، والسجل يكون لطالب واحد (نوع العلاقة ١).

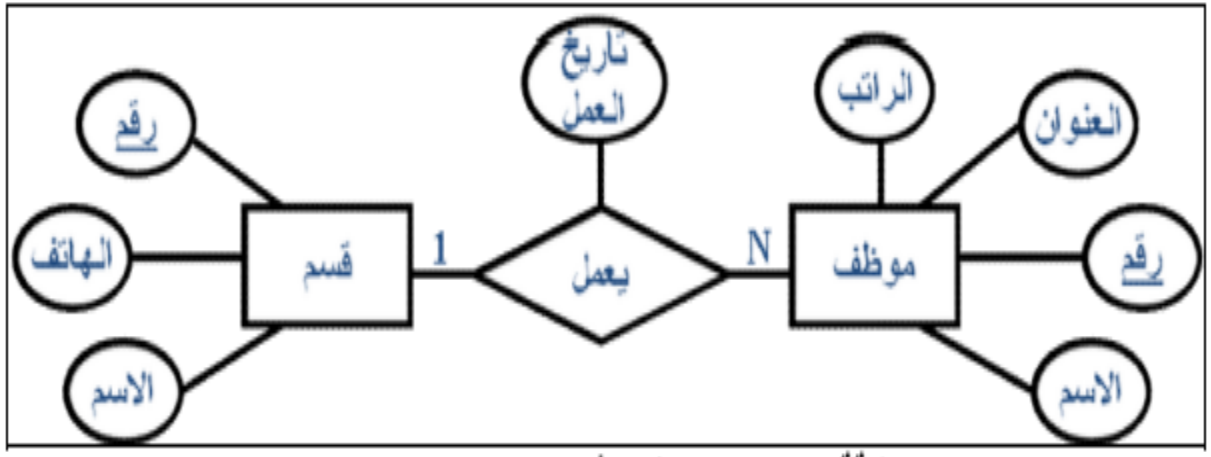
مكونات مخطط الكينونة العلائقي



علاقة 1:N واحد-الى-كثير (one-to-many)

لا حظ أنه كل ابن يتبع لموظف واحد، لأنه لكل إبن أب واحد!، ولكن الموظف قد يكون له عدة أبناء.

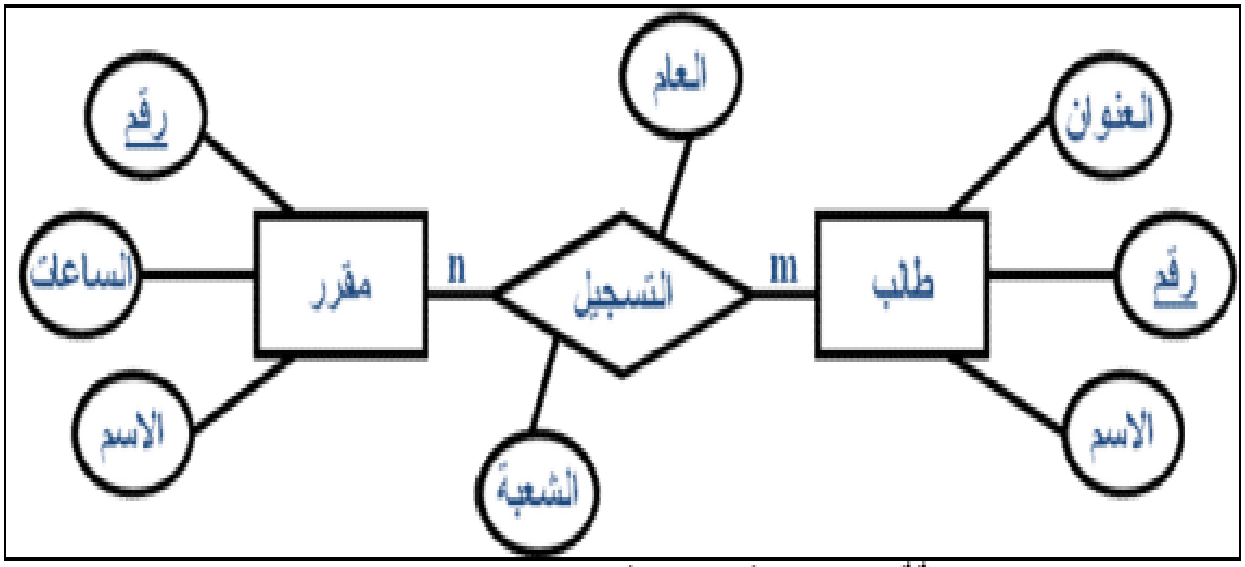
مكونات مخطط الكينونة العلائقي



علاقة 1:N واحد-الى-كثير (one-to-many)

لاحظ أنه لكل موظف قسم واحد، فالموظف لا يمكن أن ينتمي لأكثر من قسم إداري واحد، ولكن القسم قد يكون فيه عدة موظفين.

مكونات مخطط الكينونة العلائقي



علاقة M:N كثير-الى-كثير (many-to-many)

لاحظ أن الطالب قد يكون له عدة مقررات، وكذلك المقرر يمكن أن يسجله عدة طلبة.

مكونات مخطط الكينونة العلائقي

- **أنواع القيود على العلاقات Relationship Constraints**:كما أوضحنا سابقا فالعلاقات تختلف فيما بينها، وكذلك شروط العلاقة تختلف حسب طبيعة العلاقة، وحسب الكيانات المرتبطة، وقيد العلاقة هو القيد الذي يعتمد على طبيعة اشتراك كيانين في علاقة ما، هل هو اشتراك كلي أم جزئي؟، ويحدد **نوع الاشتراك** ما إذا كان وجود الكيان يعتمد على كونه مرتبط بكيان آخر عن طريق العلاقة.

مكونات مخطط الكينونة العلائقي

- **أنواع القيود على العلاقات Relationship Constraints:**

1. **الاشتراك الكلي(Total participation) :** نقول أن العلاقة علاقة الاشتراك الكلي ، إذا كان كل وحدة في الكيان ألأول يجب أن ترتبط بوحدة من الكيان الآخر ضمن العلاقة، يسمي هذا القيد بقيد "ارتباط الوجود" ، أي أن وجود وحدة من كيان ما يستلزم ارتباطها بوحدة من كيان آخر،ومثال ذلك كيان (سجل طبي) لطالب في نظام معلومات الجامعة، إذ لا بد أن يرتبط أي قيد في السجل الطبي بطالب ما في كيان الطلاب، وإلا فلا معنى للعلاقة هذه.

- ويتم تمثيل قيد الاشتراك الكلي، برسم خط مزدوج، يربط الكيانات المرتبطة بهذه العلاقة، من جهة الكيان المعتمد على الاشتراك الكلي

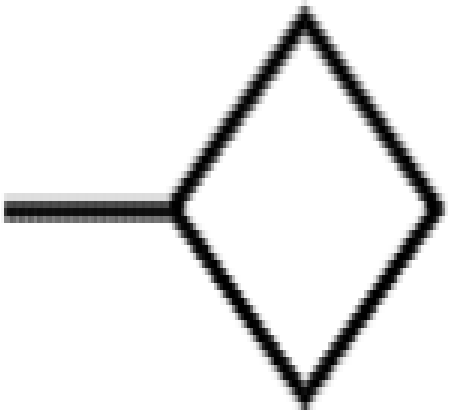
مكونات مخطط الكينونة العلائقي

- **أنواع القيود على العلاقاتRelationship Constraints :**

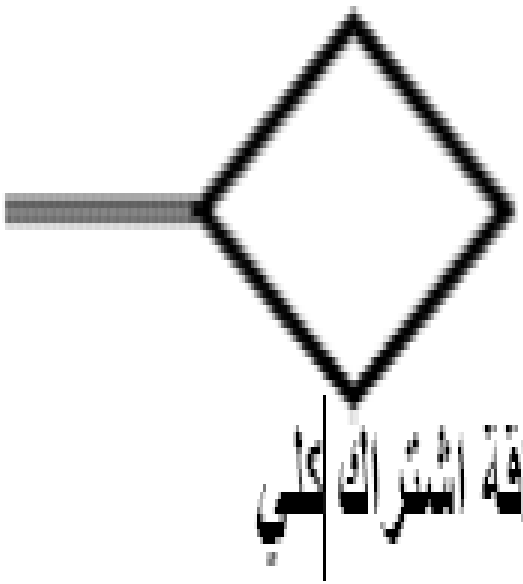
2.الاشتراك الجزئي(Partial participation) :نقول أن العلاقة علاقة اشتراك جزئي، إذا كانت بعض الوحدات في الكيان المشترك بالعلاقة ترتبط ببعض الوحدات في الكيان الآخر ضمن العلاقة، ويتم تمثيل قيد الاشتراك الجزئي برسم خط مفرد يربط الكيانات المرتبطة.

مكونات مخطط الكينونة العلائقي

- **أنواع القيود على العلاقاتRelationship Constraints:**



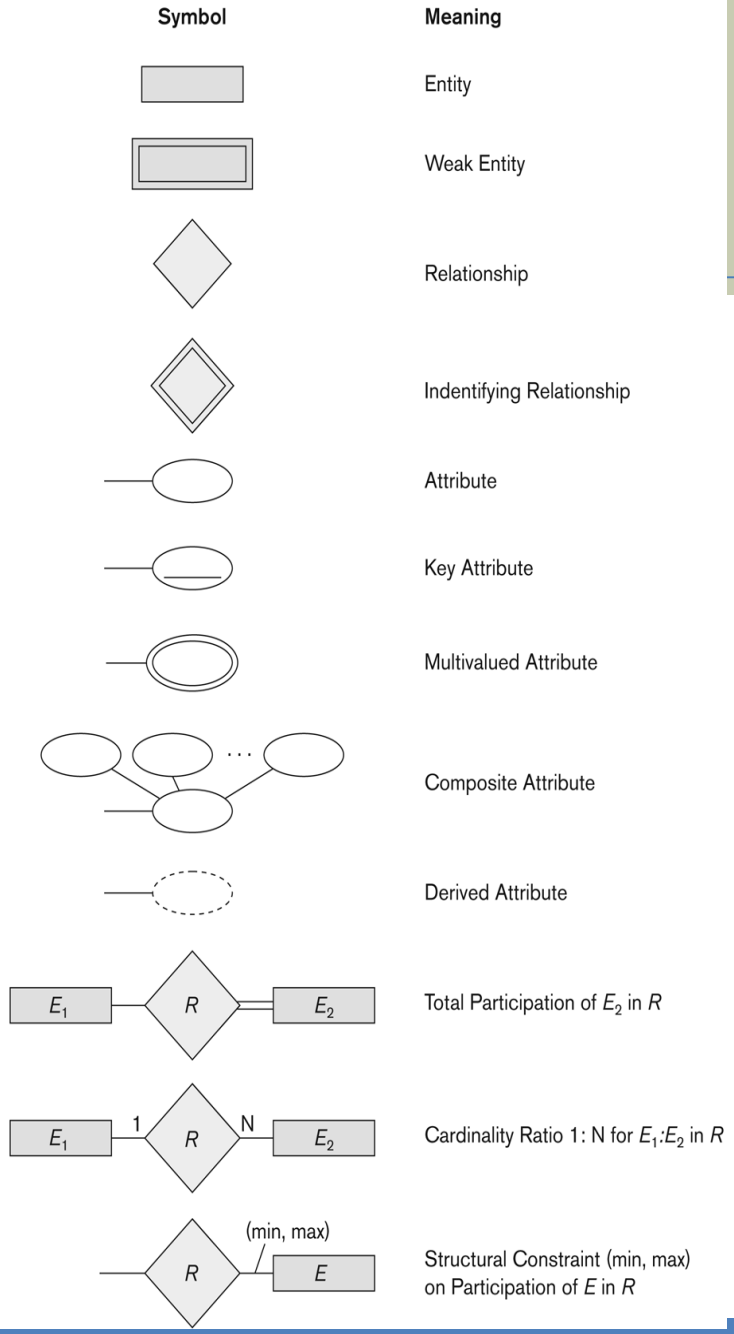
شكل علاقة اشتراك جزئي



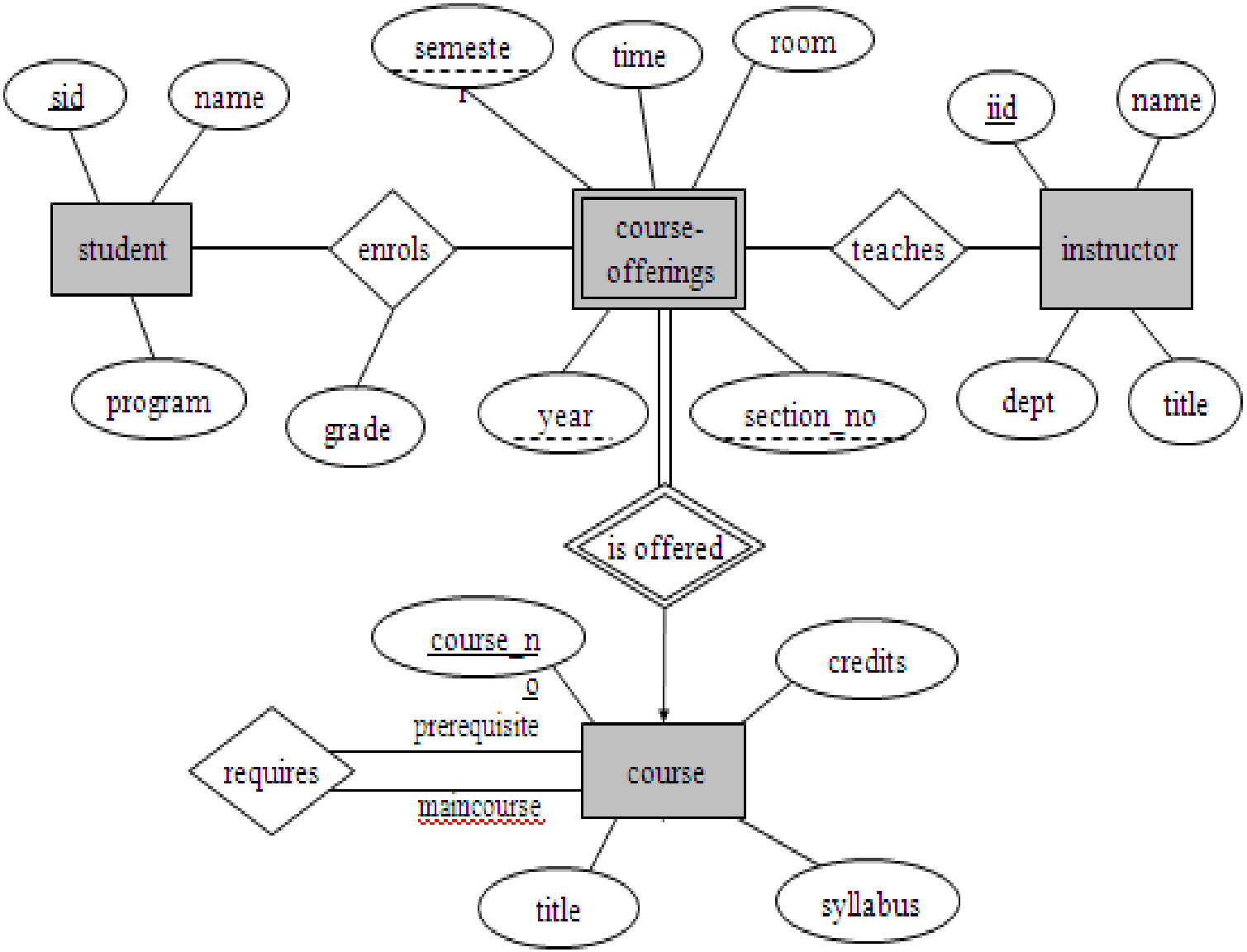
شكل علاقة اشتراك كلي

مكونات مخطط الكينونة العلائقي

Figure 3.14
Summary of the notation for ER diagrams.

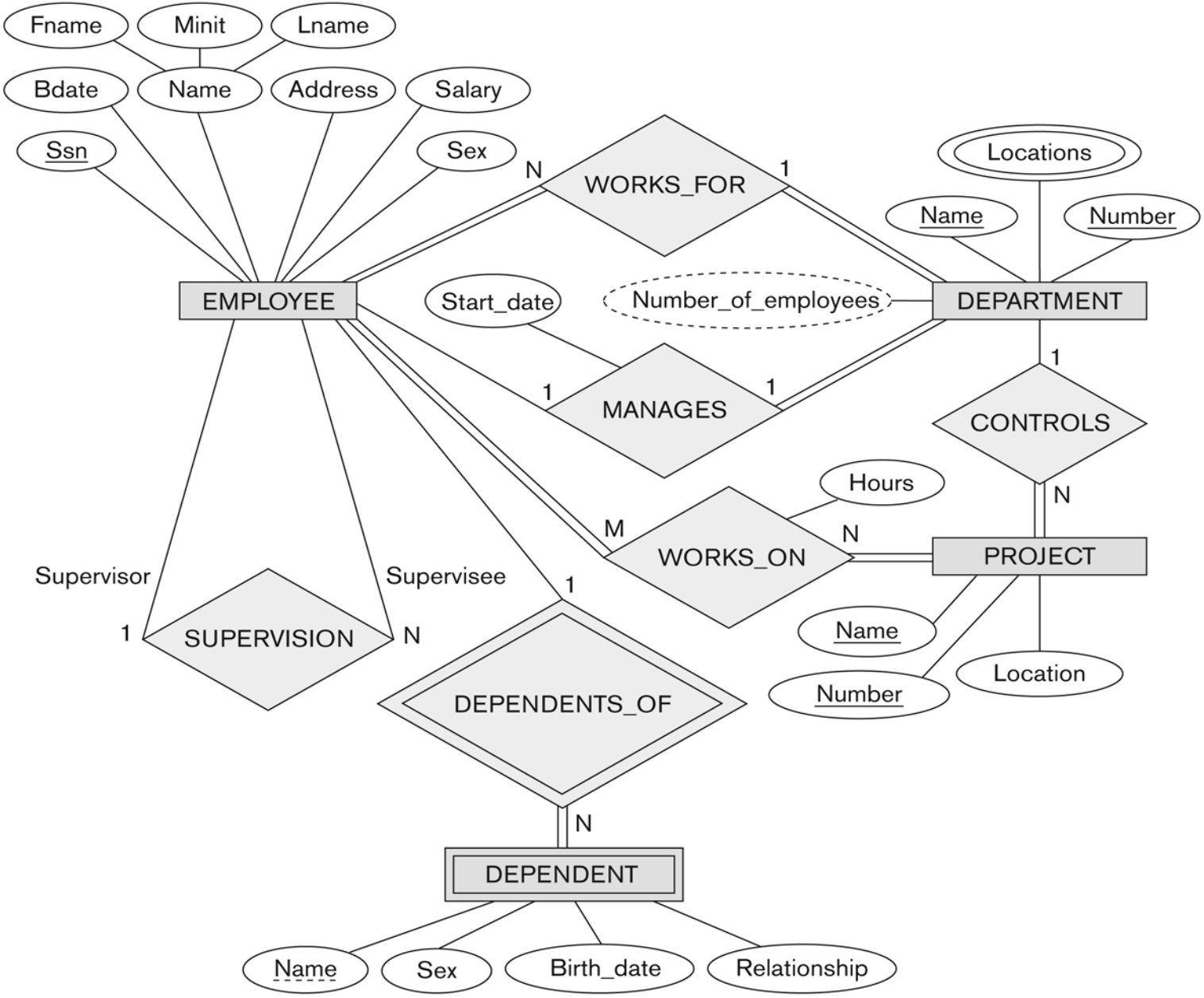


مثال 1



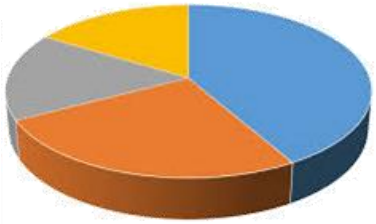
E-R diagram for a university.

مثال2


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
WORKS_ON
N
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ



8

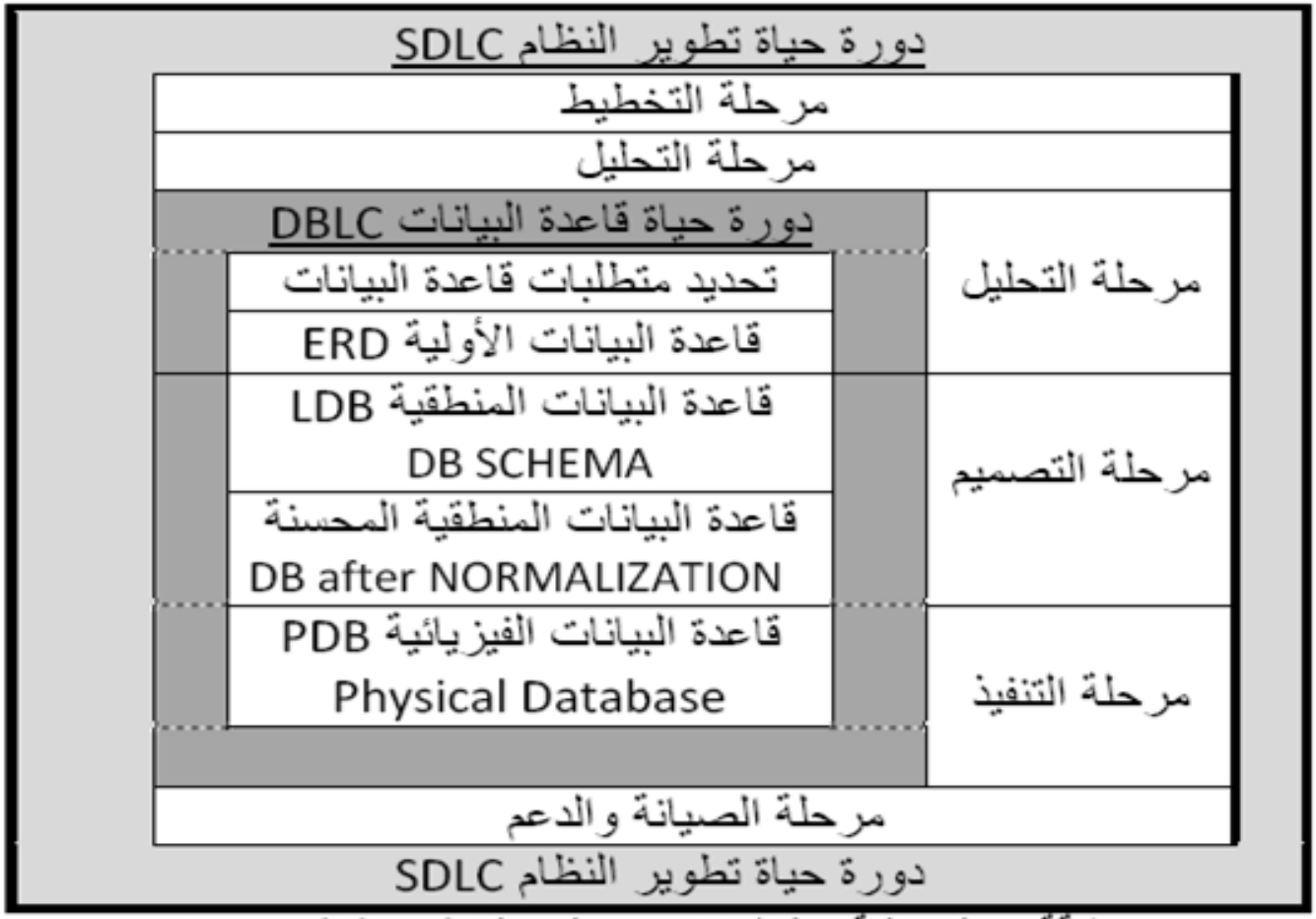
المحاضرة الثامنة

8. تصميم قاعدة البيانات DataBase Design

في هذ المحاضرة

- دورة حياة قاعدة البيانات Database Life Cycle (DBLC)
- مخطط قواعد البيانات Database Schema
- التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
  - تحويل الكيان
  - تحويل الصفة المركبة والصفة متعددة القيمة
  - تحويل الكيان الضعيف
  - تحويل العلاقات
- مثال1(تحويل مخطط كينونة علائقي لجامعة إلى مخطط قواعد بيانات)
- مثال2 (تحويل مخطط كينونة علائقي لشركة إلى مخطط قواعد بيانات)

Database Life Cycle (DBLC) دورة حياة قاعدة البيانات



علاقة دورة حياة قاعدة البيانات بدورة حياة تطوير النظام عموما

مخطط قاعدة البيانات DB Schema

مخطط قواعد البيانات Database Schema

- مخطط قواعد البيانات Database Schema: هو مخطط يصف قاعدة البيانات بشكل رسومي تمهيدا لبنائه على شكل جداول في نظام إدراة قواعد بيانات DBMS
- مخطط قواعد البيانات هو مخطط ينتج عن عملية إخضاع مخطط الكيان العلائقي لخوارزمية التحويل Mapping Algorithm

مخطط قواعد البيانات Database Schema

- **لوصف مخطط قواعد البيانات ،نستخدم المصطلحات التالية:**
- **العلاقة (جدول السكيما) relation:** أو يمكن أن نطلق عليها اسم الجداول، وهي مكونات مخطط قاعدة البيانات الناتجة من إجراء عمليات تحويل مخطط كينونة- علاقة.
- **الحقل field:** هو العمود column الذي يشكل جزء من مكونات الجدول، ويتكون من مجموعة من الأعمدة أو الحقول التي تتميز بتجانس بيانات كل حقل، على أنه يمكن أن يكون نوع بيانات كل حقل مختلفا عن بيانات النوع الآخر
- **السجل Record:** هو الصف row الذي يمثل وحدة instance من وحدات الكيان، بعد تحويله إلى جدول، ويتكون الصف من الخلايا الناتجة عن تقاطعه مع الأعمدة المكونة للجدول.

مخطط قواعد البيانات Database Schema

- **لوصف مخطط قواعد البيانات ،نستخدم المصطلحات التالية:**
- **المفتاح الرئيسي Primary Key(PK):** هو حقل في جدول يتميز بأن قيمه وحيدة في جميع صفوف الجدول، وتكون قيمتة مميزة لكل صف عن أي صف آخر.
- **المفتاح الأجنبي foreign key(FK):** هو حقل موجود في جدول وهو لا يمثل واحدة من صفاتة، ولكنه يعتبر مفتاح أجنبيا لأنه يمثل جدولا آخر، ويجب أن يكون هو نفسه المفتاح الرئيسي في ذلك الجدول، أو على الأقل تكون قيمتة وحيده unique value ويقوم المفتاح الأجنبي بتمثيل العلاقة relationship والربط بين جدولين.

AUTHOR

au_id	au_lname	au_fname	address	city	state
172-32-1176	White	Johnson	10932 Bigge Rd.	Menlo Park	CA
213-46-8915	Green	Marjorie	309 63rd St. #411	Oakland	CA
238-95-7766	Carson	Cheryl	589 Darwin Ln.	Berkeley	CA
267-41-2394	O'Leary	Michael	22 Cleveland Av. #14	San Jose	CA
274-80-9391	Straight	Dean	5420 College Av.	Oakland	CA
341-22-1782	Smith	Meander	10 Mississippi Dr.	Lawrence	KS
409-56-7008	Bennet	Abraham	6223 Bateman St.	Berkeley	CA
427-17-2319	Dull	Ann	3410 Blonde St.	Palo Alto	CA
472-27-2349	Gringlesby	Burt	PO Box 792	Covelo	CA
486-29-1786	Locksley	Charlene	18 Broadway Av.	San Francisco	CA

TITLE

title_id	title	type	price	pub_id
BU1032	The Busy Executive's Database Guide	business	19.99	1389
BU1111	Cooking with Computers	business	11.95	1389
BU2075	You Can Combat Computer Stress!	business	2.99	736
BU7832	Straight Talk About Computers	business	19.99	1389
MC2222	Silicon Valley Gastronomic Treats	mod_cook	19.99	877
MC3021	The Gourmet Microwave	mod_cook	2.99	877
MC3026	The Psychology of Computer Cooking	UNDECIDED		877
PC1035	But Is It User Friendly?	popular_comp	22.95	1389
PC8888	Secrets of Silicon Valley	popular_comp	20	1389
PC9999	Net Etiquette	popular_comp		1389
PS2091	Is Anger the Enemy?	psychology	10.95	736

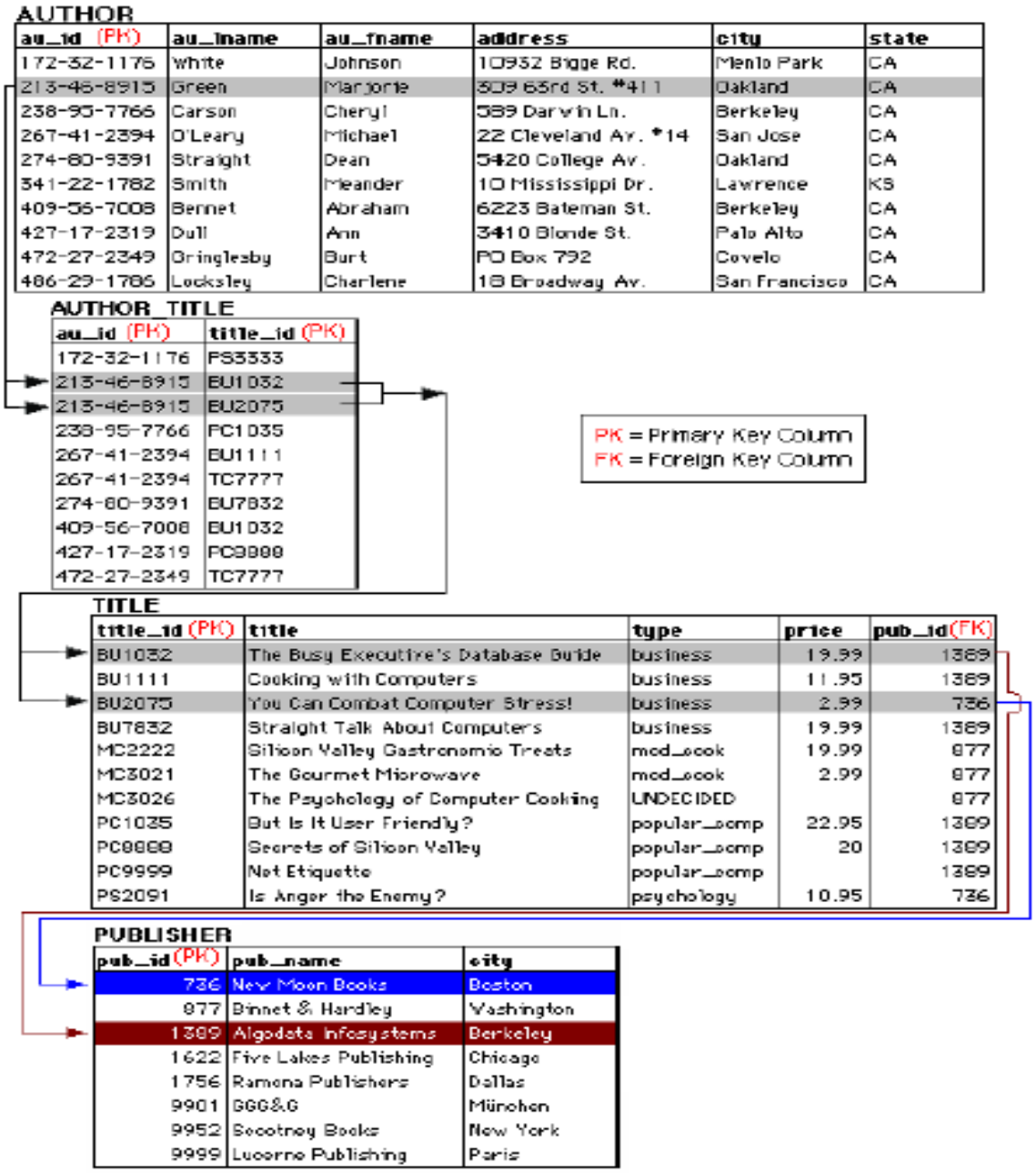
PUBLISHER

pub_id	pub_name	city
736	New Moon Books	Boston
877	Binnet & Hardley	Washington
1389	Algodata Infosystems	Berkeley
1622	Five Lakes Publishing	Chicago
1756	Ramona Publishers	Dallas
9901	GGG&G	München
9952	Scootney Books	New York
9999	Lucerne Publishing	Paris

AUTHOR_TITLE

au_id	title_id
172-32-1176	PS3333
213-46-8915	BU1032
213-46-8915	BU2075
238-95-7766	PC1035
267-41-2394	BU1111
267-41-2394	TC7777
274-80-9391	BU7832
409-56-7008	BU1032
427-17-2319	PC8888
472-27-2349	TC7777

AUTHOR	(au_id, au_lname, au_fname, address, city, state, zip)
TITLE	(title_id, title, type, price, pub_id)
PUBLISHER	(pub_id, pub_name, city)
AUTHOR_TITLE	(au_id, title_id)


AUTHOR
au_id (PK) | au_lname | au_fname | address | city | state
172-32-1176 | White | Johnson | 10932 Bigge Rd. | Menlo Park | CA
213-46-8915 | Green | Marjorie | 309 63rd St. #411 | Oakland | CA
238-95-7766 | Carson | Cheryl | 589 Darwin Ln. | Berkeley | CA
267-41-2394 | O'Leary | Michael | 22 Cleveland Av. #14 | San Jose | CA
274-80-9391 | Straight | Dean | 5420 College Av. | Oakland | CA
341-22-1782 | Smith | Meander | 10 Mississippi Dr. | Lawrence | KS
409-56-7008 | Bennet | Abraham | 6223 Bateman St. | Berkeley | CA
427-17-2319 | Dull | Ann | 3410 Blonde St. | Palo Alto | CA
472-27-2349 | Gringlesby | Burt | PO Box 792 | Covelo | CA
486-29-1786 | Locksley | Charlene | 18 Broadway Av. | San Francisco | CA
AUTHOR_TITLE
au_id (PK) | title_id (PK)
172-32-1176 | PS3333
213-46-8915 | BU1032
213-46-8915 | BU2075
238-95-7766 | PC1035
267-41-2394 | BU1111
267-41-2394 | TC7777
274-80-9391 | BU7832
409-56-7008 | BU1032
427-17-2319 | PC9999
472-27-2349 | TC7777
PK = Primary Key Column
FK = Foreign Key Column
TITLE
title_id (PK) | title | type | price | pub_id (FK)
BU1032 | The Busy Executive's Database Guide | business | 19.99 | 1389
BU1111 | Cooking with Computers | business | 11.95 | 1389
BU2075 | You Can Combat Computer Stress! | business | 2.99 | 736
BU7832 | Straight Talk About Computers | business | 19.99 | 1389
MC2222 | Silicon Valley Gastronomic Treats | mod_cook | 19.99 | 877
MC3021 | The Gourmet Microwave | mod_cook | 2.99 | 877
MC3026 | The Psychology of Computer Cooking | UNDECIDED | | 877
PC1035 | But Is It User Friendly? | popular_comp | 22.95 | 1389
PC8888 | Secrets of Silicon Valley | popular_comp | 20 | 1389
PC9999 | Net Etiquette | popular_comp | | 1389
PS2091 | Is Anger the Enemy? | psychology | 10.95 | 736
PUBLISHER
pub_id (PK) | pub_name | city
736 | New Moon Books | Boston
877 | Binnet & Hardley | Washington
1389 | Algodata Infosystems | Berkeley
1622 | Five Lakes Publishing | Chicago
1756 | Ramona Publishers | Dallas
9901 | GGG&G | München
9952 | Scootney Books | New York
9999 | Lucerne Publishing | Paris

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

تتم عملية تحويل مخطط ERD، بتطبيق مجموعة من الخطوات البسيطة، تسمى خوارزمية التحويل Mapping Algorithm، وتتكون هذه الخطوات من جميع الحالات البسيطة المحتملة، التي قد تكون موجودة في النموذج الأولى، ويتم تطبيق هذه الخوارزمية كاملة، مع تجاوز الحالات التي لم تظهر في النموذج الأولي

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

خوارزمية التحويل Mapping Algorithm:

Step 1: Mapping of Regular Ent y Types	١. تحويل الكيانات العادية (القوية).
Step 2: Mapping of Weak Ent y Types	٢. تحويل الكيانات الضعيفة.
Step 3: Mapping of Binary 1:1 Relaon Types	٣. تحويل العلاقات الثنائية من النوع ١:١.
Step 4: Mapping of Binary 1:N Relaonshi p Types.	٤. تحويل العلاقات الثنائية من النوع ١:N.
Step5: Mapping of Binary M:N Relaonshi p Types.	٥. تحويل العلاقات الثنائية من النوع N:M.
Step 6: Mapping of Mulval ued ari bute s.	٦. تحويل الصفات متعددة القيم.
Step 7: Mapping of N-ary Relationship Types.	٧. تحويل العلاقات فوق الثنائية.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

١. **تحويل أنواع الكيانات العادية:** يتم هنا تحويل جميع الكيانات العادية، أي الكيانات غير الضعيفة، بإنشاء جدول يتكون من الحقول التي تقابل صفات ذلك الكيان. ويتم تحديد أحد مفاتيح الكيان، وتسميته بالمفتاح الرئيسي primary key(PK) وإذا كانت الصفة التي تمثل المفتاح من النوع المركب فإن المفتاح الرئيسي سيكون مجموعة الحقول التي تنشأ من الصفة المركبة.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٢. **تحويل الكيانات الضعيفة:** يتم تحويل كل واحدة من الكيانات الضعيفة، بإنشاء جدول يتكون من الحقول التي تقابل صفات ذلك الكيان، كما يجب إضافة المفتاح الرئيسي للكيان القوي الذي يتبعه ذلك الكيان الضعيف، ويكون المفتاح الرئيسي PK للجدول الجديد، عبارة عن مفتاح مركب مكون من المفتاح الأجنبي FK بالإضافة إلى المفتاح الجزئي (Partial Key) الخاص به.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٣. **تحويل العلاقات الثنائية من النوع (1:1):** إذا كانت العلاقة بين الكيانين علاقة واحد-إلى-واحد فإن عملية التحويل تتم وفق عدة خيارات أشهرها، خيار يسمى **بطريقة المفتاح الأجنبي، وفيه يتم إضافة المفتاح الرئيسي** لأحد الجدولين إلى الجدول الآخر كمفتاح أجنبي ويفضل أن يكون الجدول الذي يحتوي على المفتاح الأجنبي، هو الجدول الذي يكون نوع قيد اشتراكه في العلاقة، من نوع (الاشتراك الكلي)

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٤. **تحويل العلاقات الثنائية من النوع (1:N):** يتم هنا إنشاء جدولين لتمثيل الكيانين المرتبطين، على أن يتم تطبيق **طريقة المفتاح الأجنبي السابقة، وذلك بإضافة** المفتاح الرئيسي للجدول من جهة العلاقة (1) إلى الجدول الآخر المرتبط بالعلاقة (M)، بغض النظر عن نوع قيد الاشتراك.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٥. **تحويل العلاقات الثنائية من النوع (N:M):** في هذا النوع من العلاقات، يتم استحداث جدول جديد، فيكون الناتج من هذه العلاقة ثلاثة جداول، جدولين لتمثيل الكيانين المرتبطين بالعلاقة ويضم الجدول الثالث حقلين كمفتاحين أجنبيين يمثلان المفتاحين الرئيسيين في الجدولين، ويمكن إضافة أي حقل آخر يكون له مغزى، كأن تكون العلاقة لها صفة بذاتها، فتتحول الصفة إلى حقل في الجدول الجديد.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٦. **تحويل الصفات متعددة القيم:** يتم في هذه الحالة، عادة، إنشاء جدول جديد يضم الصفة المتعددة القيم كحقل، ويضاف إلى الجدول مفتاح أجنبي FK يكون ممثلا للمفتاح الرئيسي في الجدول الناتج من الكيان الذي يحتوي على الصفة متعددة القيم. أما الصفات المركبة فتتحول إلى صفات بسيط، فحقول عادية كما أوضحنا أعلاه، والصفات ذات القيم المشتقة تلغى من الجدول، لأنها صفات قابلها للاشتقاق من صفات أخرى، فلا داعي لوجودها

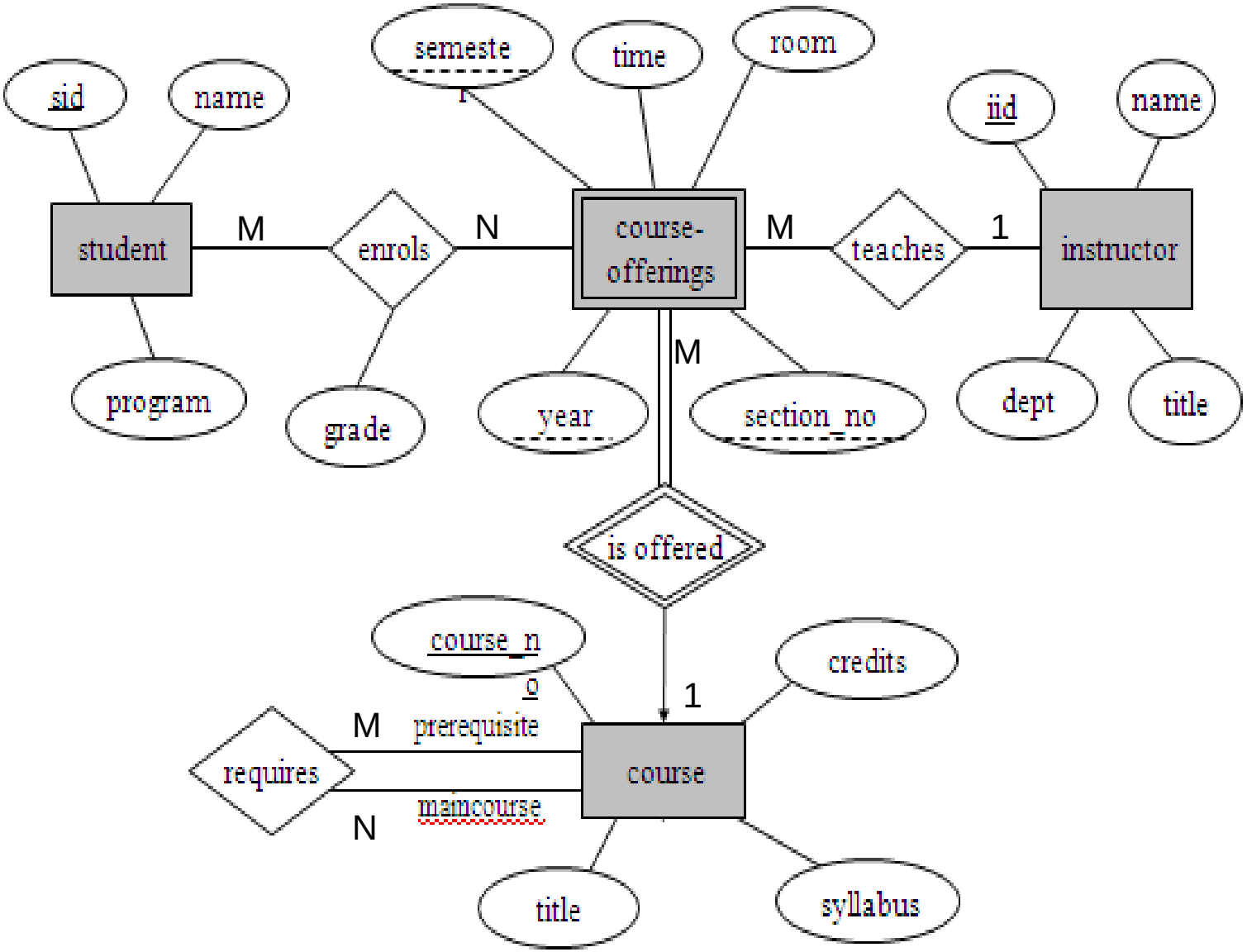
التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

٧. **تحويل العلاقات غير الثنائية، كالعلاقة الثلاثية وما فوقها:**
في حالات نادرة تظهر لدينا علاقات معقدة، كالعلاقة الثلاثية (بين ثلاثة كيانات) والرباعية وما فوقها، وتعالج هذه الحالة بطريقة معالجة الحالة الخامسة (حالة تحويل العلاقات الثنائية من النوع (N:M) حيث يتم إنشاء جدول جديد، وإضافة المفاتيح الرئيسية للجداول المشتركة، حسب عددها، إلى الجدول الجديد كمفاتيح أجنبية مكونة بمجموعها، مفتاحا مركبا يمثل المفتاح الرئيسي للجدول.

التحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات
Mapping ERD to DB schema

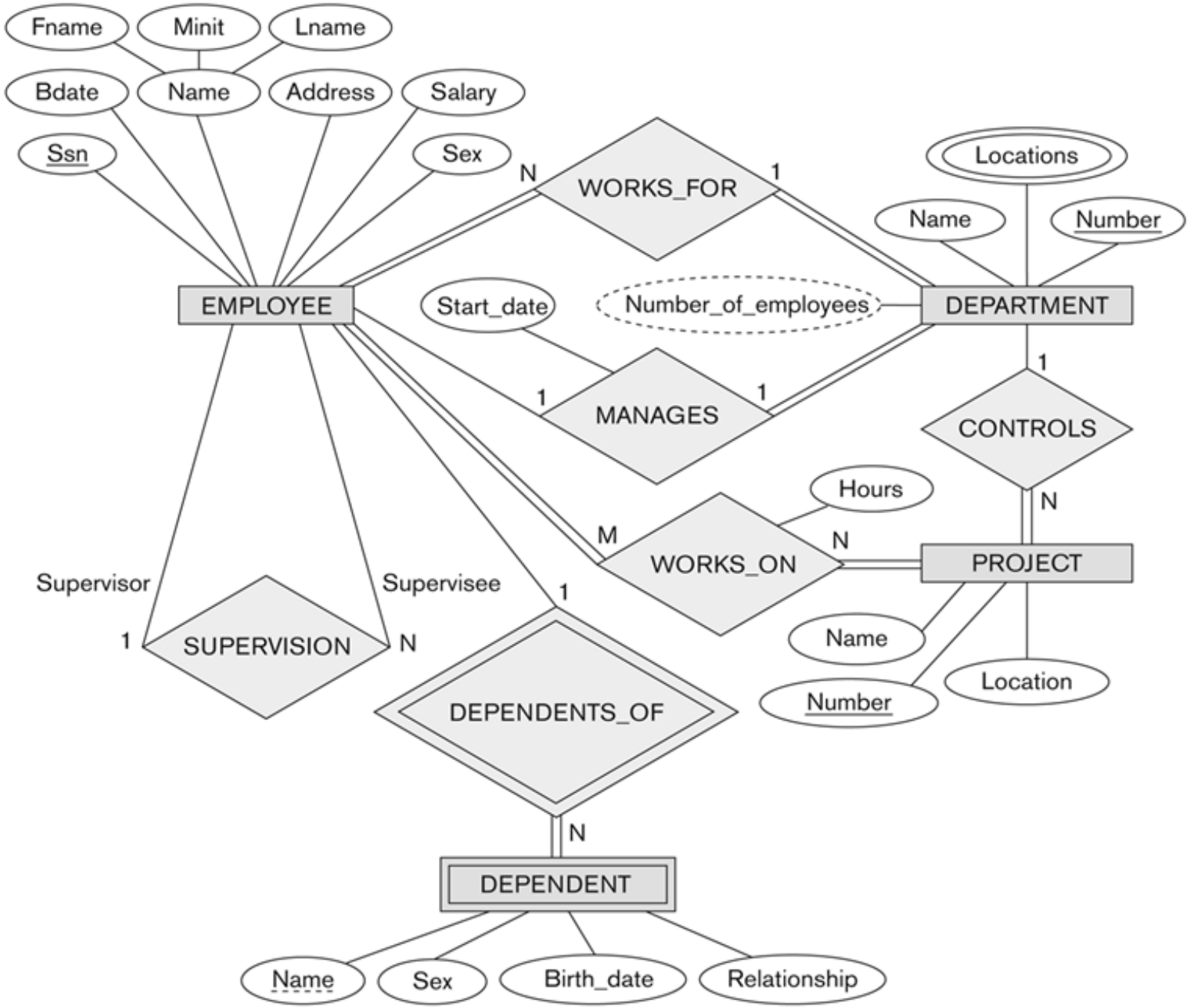
تمرين: حول كل من نماذج الكيان العلائقي التاليين إلى ما يقابلها من مخطط قواعد البيانات Database Schema

مثال 1



E-R diagram for a university.

مثال2


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
WORKS_ON
N
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ

9

Eng. Hana Sabbagh

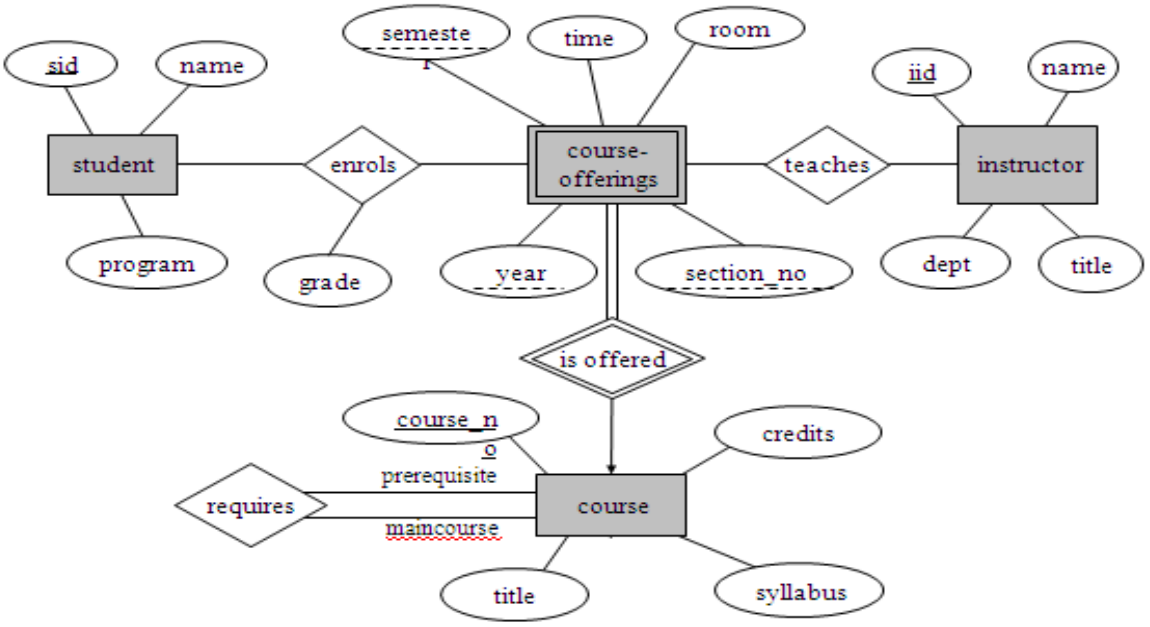
المحاضرة التاسعة

9. أمثلة على تصميم قاعدة البيانات DataBase Design Example

Eng. Hana Sabbagh

أمثلة للتحويل من مخطط الكيان العلائقي إلى مخطط قواعد بيانات

مثال 1 (مخطط الكيان العلائقي لجامعة)



E-R diagram for a university.

حول كل من نموذج الكيان العلائقي السابق إلى ما يقابله من مخطط قواعد البيانات Database

1. **نبدأ بتحويل الكيان العادي:**

يتم تمثيل الكيان العادي (القوي) بشكل المستطيل أحادي الإطار، ويحتوي مخطط الكيان العلائقي السابق على ثلاثة كيانات هي:

- الطالب (Student)
- المحاضر (Instructor)
- المقرر (Course)

نلاحظ في هذا المثال أن كل الصفات من النوع البسيط

ويكون المفتاح الرئيسي (Primary Key) للجدول هو مجموعة صفات المفتاح الرئيسي المرتبطة بالكيان

وينتج عن عملية التحويل الجداول التالية:

STUDENT	sid	name	program

INSTRUCTOR	iid	name	dept	title

COURSE	courseno	title	syllabus	Credits

2. **تحويل الكيان الضعيف:**

يتم تمثيل الكيان الضعيف بشكل مستطيل مزدوج الإطار، وسبب ضعف الكيان، ينتج من عدم وجود صفة مفتاح رئيسي له، ولكن يحتوي على صفة مفتاح جزئي، ولدينا في هذا المثال كيان ضعيف واحد هو:

كيان الشعب الفصلية المقترحة (Course-Offerings)

ويحتوي هذا الكيان على صفات المفتاح الجزئي التالية:

- السنة (Year)
- الفصل (Semester)
- رقم الشعبة (Section-no)

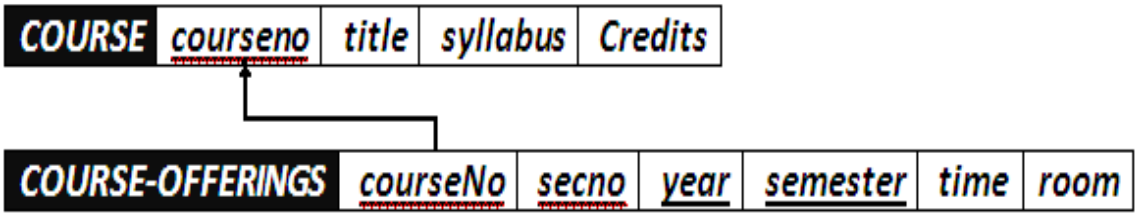
ويجب أن يرتبط الكيان الضعيف بكيان قوي بواسطة علاقة تعريف تمثل علاقة التعريف بشكل معين مزدوج الإطار، وذلك في سبيل تقوية الكيان الضعيف.

وتتم عملية تحويل الكيان الضعيف بتحويله إلى جدول يحمل اسم الكيان الضعيف، ويحتوي حقولاً من الصفات المرتبطة به، بالإضافة إلى حقل المفتاح الرئيسي من جدول الكيان القوي المرتبط معه بعلاقة تعريف، وفي حال وجود أي صفة على علاقة التعريف، يتم تمثيلها بحقل في الجدول ويكون المفتاح الرئيسي للجدول هو مفتاح الكيان القوي بالإضافة الى المفاتيح الجزئية في الكيان الضعيف.

وينتج عن عملية التحويل الجدول التالي:

COURSE-OFFERINGS	courseno	secno	year	semester	time	room

وبذلك يرتبط جدول الشعب الفصلية المقترحة (Course-Offerings) بجدول المقرر (Course) بوجود المفتاح الأجنبي (Foreign Key) رقم المقرر (courseno)

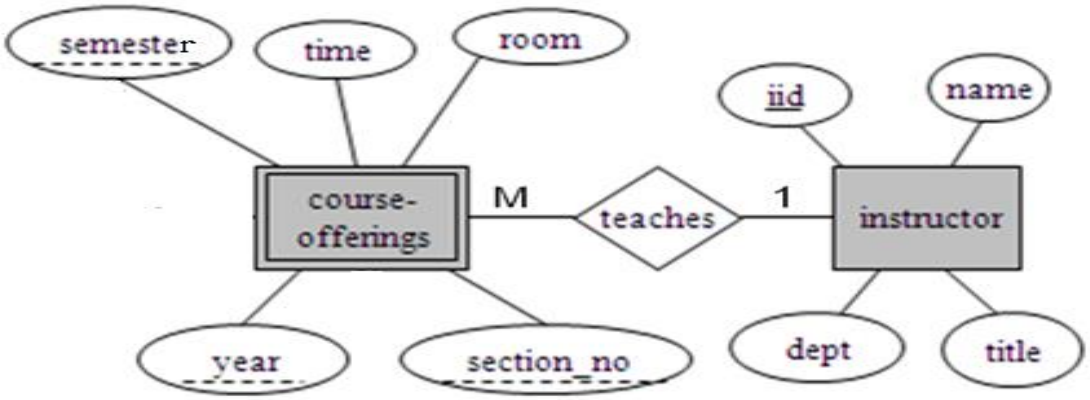


3. **تحويل العلاقات :**

يتم تمثيل العلاقة بشكل معين أحادي الإطار. في هذه الحالة يتم التعامل مع العلاقة حسب نوعها كل على حده. ويتم تجاهل علاقة التعريف (شكل معين مزدوج الإطار)، لأنه تم بناؤها مسبقا عند تحويل الكيان الضعيف.

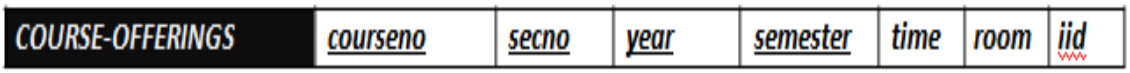
أ. تحويل علاقة واحد إلى كثير:

في هذا المثال لدينا علاقة يُدرس(teaches) بين كيان المحاضر (Instructor) وكيان الشعب الفصلية المقترحة(Course-Offerings)،

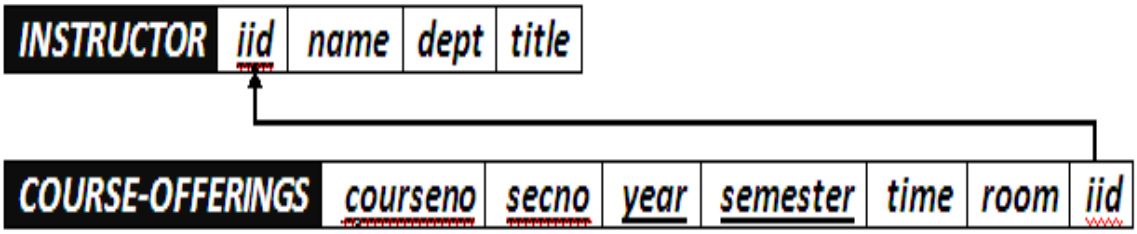


وعليه يتم أخذ المفتاح الرئيسي (iid) من جدول Instructor كونه طرف العلاقة واحد، ويضاف كمفتاح أجنبي في جدول course-offerings، وبالتالي يتم تعديل جدول

Course- Offerings ليصبح بالشكل التالي:



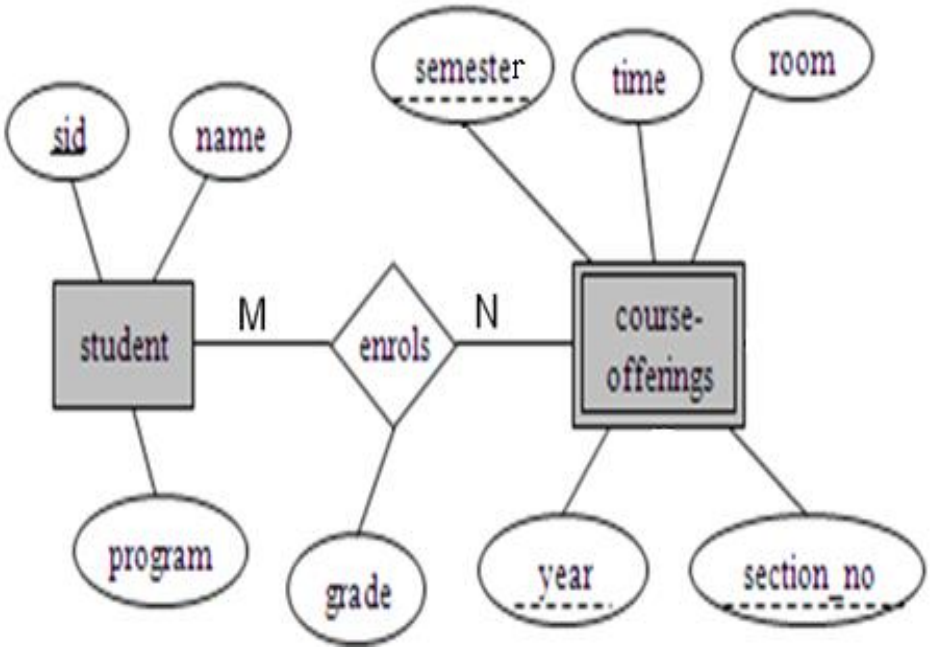
وبذلك يرتبط جدول الشعب الفصلية المقترحة (Course-Offerings)بجدول المحاضر (Instructor) بوجود المفتاح الأجنبي رقم المحاضر (iid)



ب. تحويل علاقة كثير إلى كثير:

في هذا المثال، يوجد لدينا علاقتين من نوع كثير إلى كثير، هما:

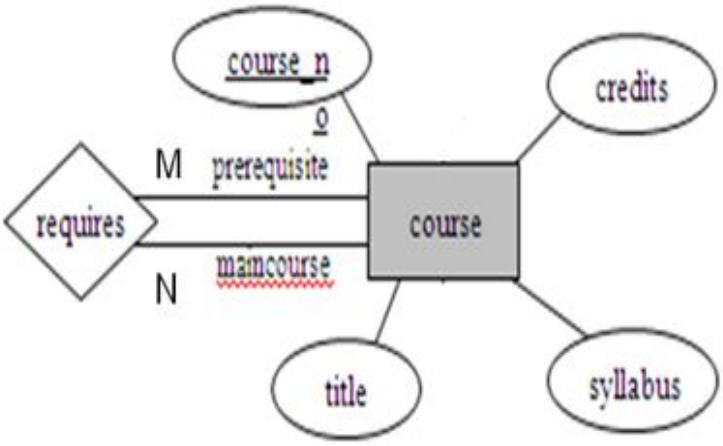
- علاقة يُسَجِّل (enrolls)



ينتج عن هذه العلاقة الجدول التالي:

ENROLS	sid	courseno	secno	year	semester	grade

- علاقة يتطلب (Requires)

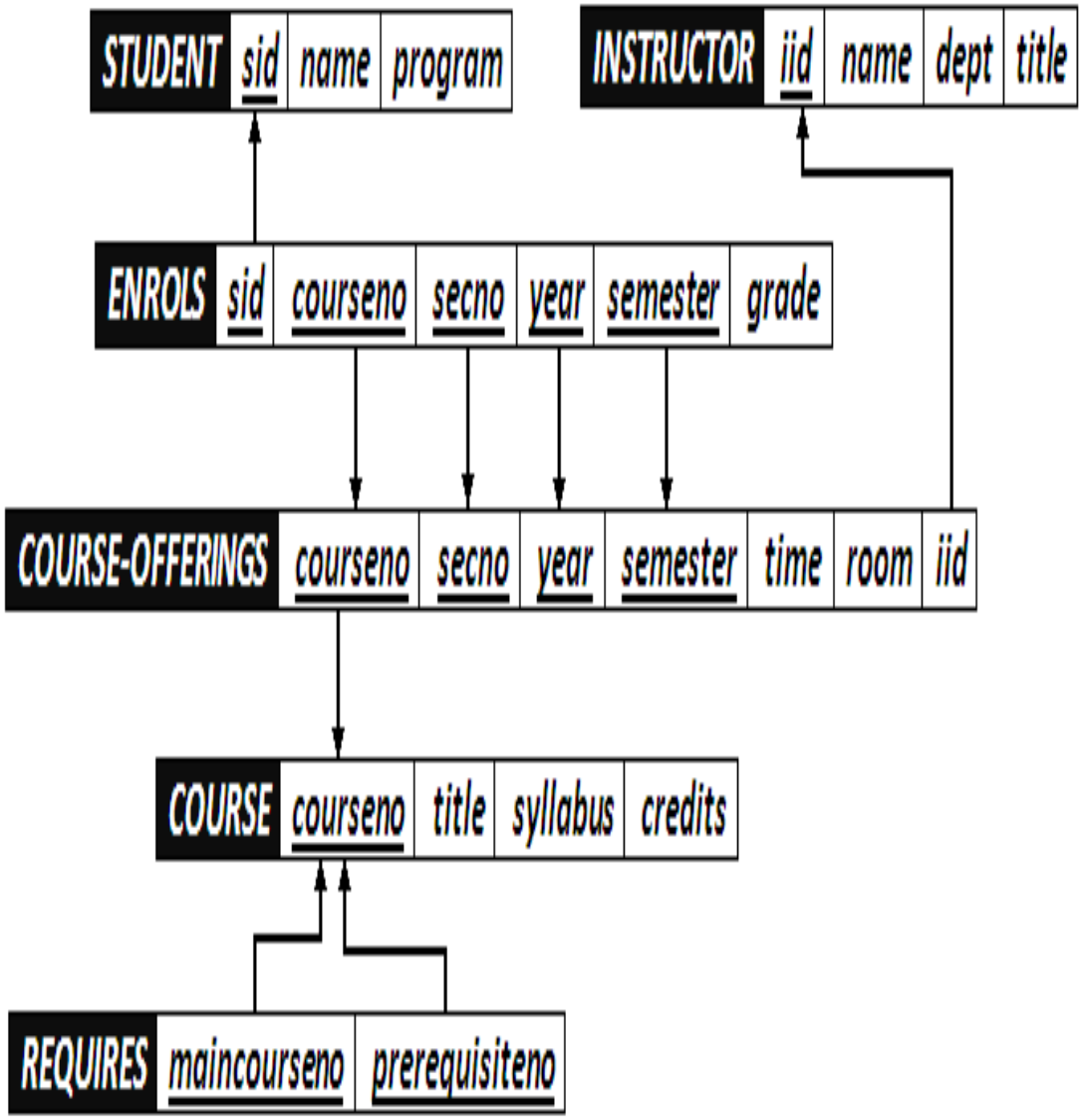


وينتج عن هذه العلاقة الجدول التالي:

REQUIRES	maincourseno	prerequisiteno

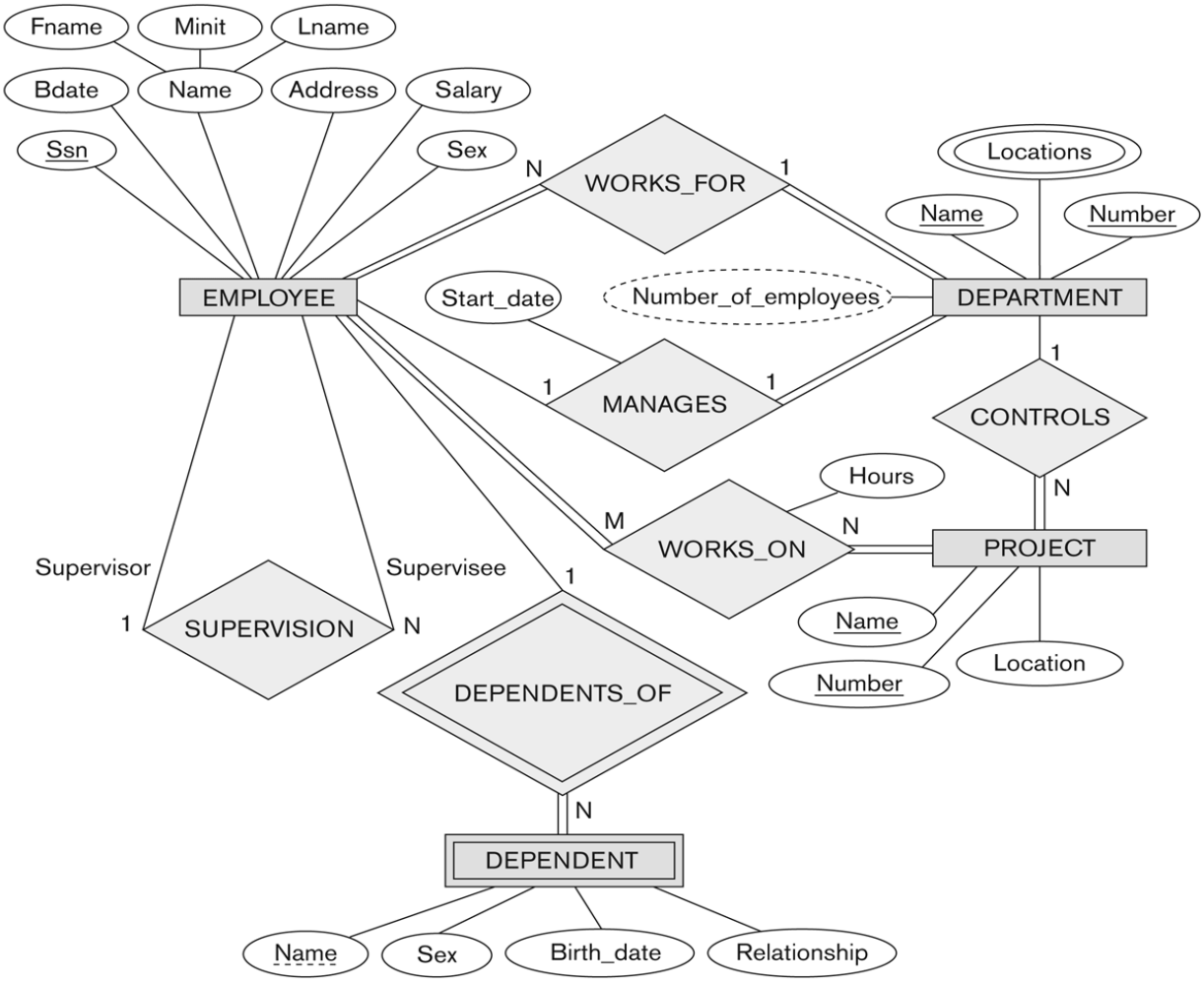
وبذلك يرتبط جدول يتطلب (Requires) مع جدول المقرر (course)

وبذلك نكون قد أنشأنا مخطط قواعد البيانات الناتج عن التحويل للمثال رقم (1)، وهو على الشكل التالي:



نلاحظ الارتباط الوثيق بين جداول قاعدة البيانات، فلا يجوز أن يكون هناك جدول دون علاقة بباقي الجداول، أو بمعزل عن الجداول الأخرى في قاعدة البيانات.

مثال 2 (مخطط الكيان العلائقي لشركة)



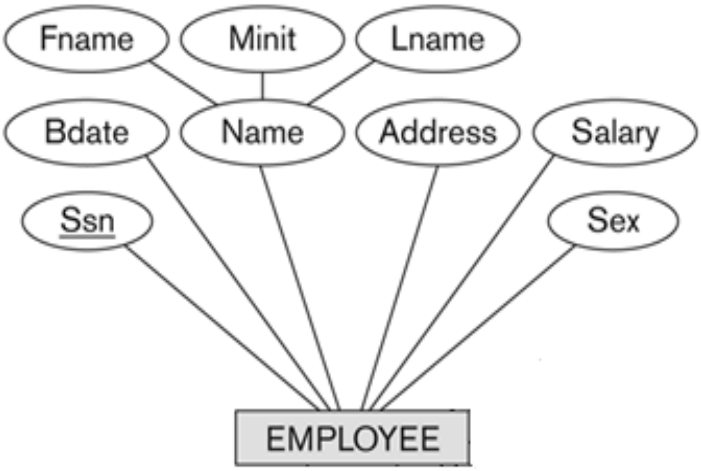
1. **نبدأ بتحويل الكيان العادي:**

يحتوي مخطط الكيان العلائقي السابق على ثلاثة كيانات عادية هي:

- الموظف (Employee)
- القسم (Department)
- المشروع (Project)

أ. **كيان الموظف (Employee)**

نلاحظ في هذا الكيان وجود صفة مركبة هي صفة الاسم (Name) والتي تتكون من الصفات الجزئية الاسم الأول(Fname)، وحرف الاسم الأوسط (Minit)، والاسم الأخير (Fname) وكما اوضحنا سابقا، فإن الصفة المركبة في عملية التحويل تدرج بصفاتها الجزئية فقط.



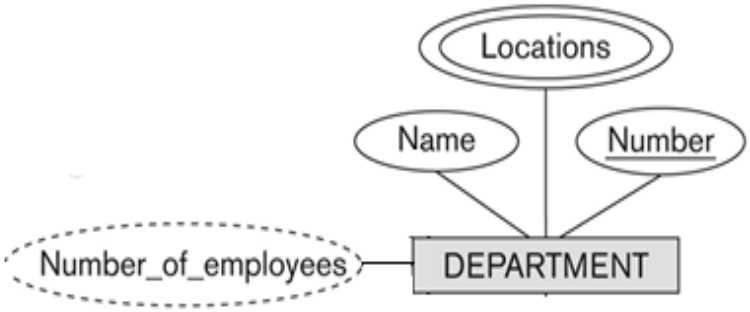
وينتج عن عملية التحويل الجدول التالي:

EMPLOYEE	Ssn	Fname	Minit	Lname	Bdate	Address	Salary	Sex

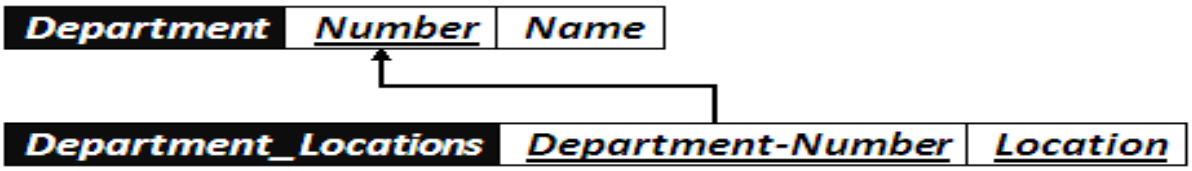
ب. كيان القسم (Department)

نلاحظ في هذا الكيان وجود صفة مشتقة هي عدد الموظفين (Number of employees)، وذكرنا سابقا أننا في عملية التحويل نتجاهل هذه الصفة، ليتم بناؤها لاحقاً بجملة استعلام.

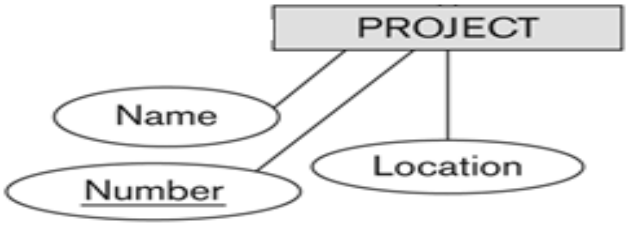
كما نلاحظ وجود صفة متعددة القيمة وهي المواقع (locations) وذكرنا سابقا أن الصفة متعددة القيمة يتم تحويلها إلى جدول مستقل مع المفتاح الرئيسي للكيان، ويكون اسم الجدول مكون من اسم الكيان مضافا إليه اسم الصفة متعددة القيمة.



وينتج عن عملية التحويل الجدولين التاليين :



ت. كيان المشروع (Project)



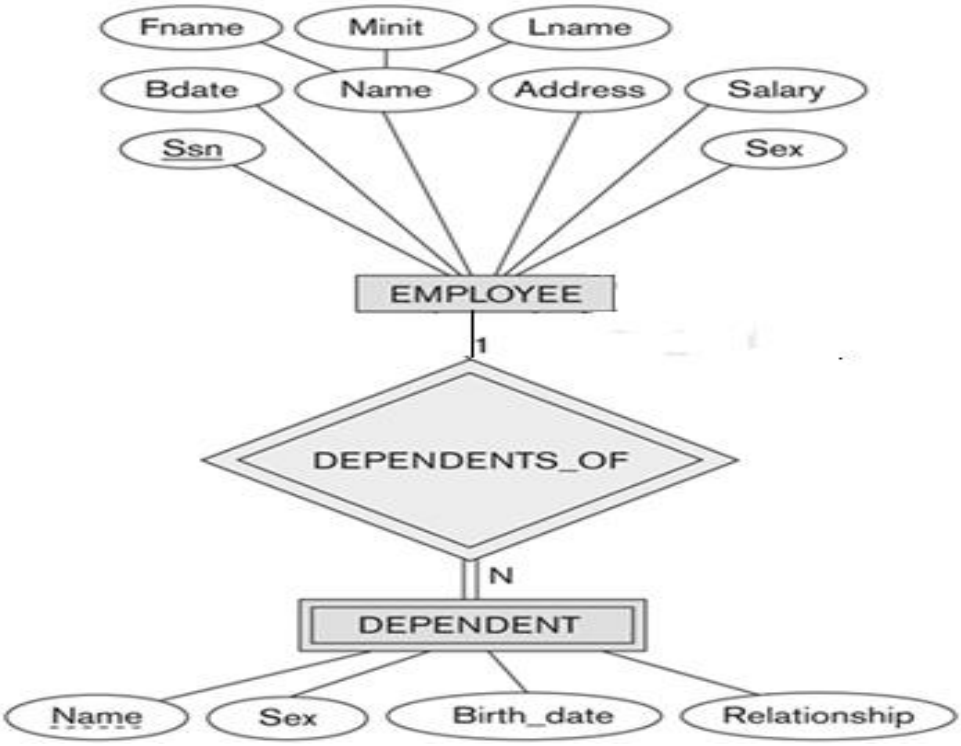
هذا الكيان يحتوي على صفات بسيطة فقط وبالتالي ينتج عن عملية التحويل الجدول التالي:

Project	Number	Name	Location

2. تحويل الكيان الضعيف:

في هذا المثال كيان ضعيف واحد هو:

كيان المعتمد على (Dependent)، حيث يصف هذا الكيان أفراد عائلة الموظف المعتمدين عليه في إعالتهم.



ويحتوي هذا الكيان على صفة المفتاح الجزئي الاسم (Name)، التالية:

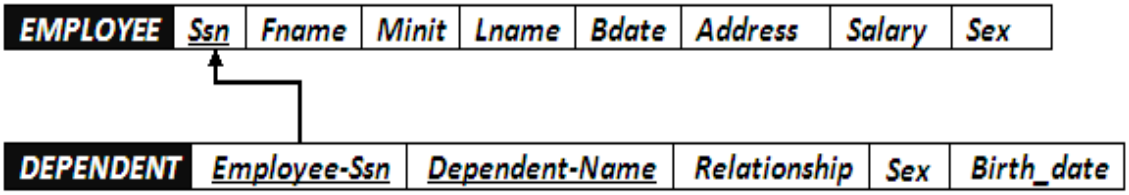
ويرتبط هذا الكيان الضعيف بكيان قوي هو كيان الموظف (Employee) بواسطة علاقة التعريف يعتمد على (Dependent_of).

وتتم عملية تحويل الكيان الضعيف بتحويله إلى جدول يحمل اسم الكيان الضعيف(Dependent)، ويحتوي حقولا من الصفات المرتبطة به، بالإضافة إلى حقل المفتاح الرئيسي من جدول الكيان القوي (Employee)المرتبط معه بعلاقة التعريف (Dependent–of) يكون المفتاح الرئيسي للجدول هو مفتاحا للكيان القوي (Empoyee–Ssn) بالإضافة إلى المفتاح الجزئي في الكيان الضعيف (Dependent–Name).

وينتج عن عملية التحويل الجدول التالي:

DEPENDENT	Employee-Ssn	Dependent-Name	Relationship	Sex	Birth_date

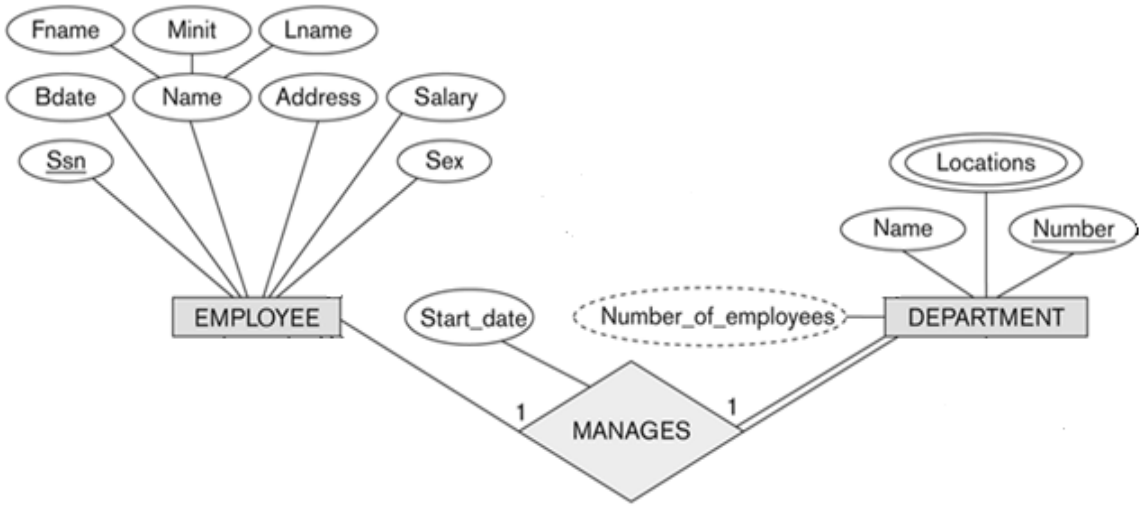
وبذلك يرتبط جدول (Dependent) بجدول (Employee) بوجود المفتاح الأجنبي رقم الموظف (Employee–Ssn).



3. **تحويل العلاقات :**

3.أ. تحويل علاقة واحد إلى واحد:

في هذا المثال توجد علاقة من النوع واحد–إلى–واحد، وهي علاقة (Manages) بين كيان الموظف، وكيان القسم، والعلاقة ذات اشتراك كلي من جهة القسم، مع وجود الصفة (start_date)على العلاقة.

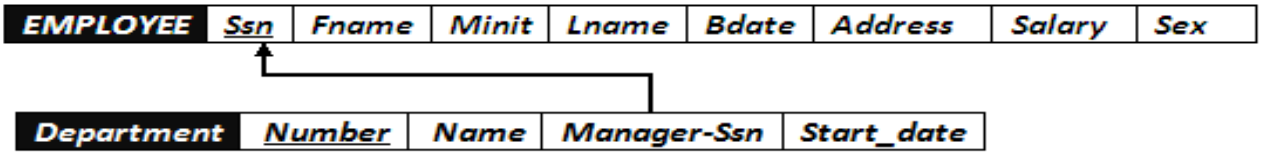


وبناءً على النقطة السابقة، يتم أخذ نسخة من المفتاح الرئيسي لجدول الموظف (Employee_Ssn)، مضافا إليها الصفة على العلاقة (start_date)، ووضعهما كتعديل على جدول القسم (department)، ويفضل إعادة تسمية المفتاح الرئيسي ليدل على العلاقة وهي علاقة ادارة ، فهو مدير (Manager)، وبالتالي بدلا من التسمية Employee_Ssnتصبح التسمية Manager_Ssn.

وعليه يتم التعديل على جدول القسم(Department)ليصبح بالشكل التالي:

Department	***Number***	***Name***	***Manager–Ssn***	***Start_date***

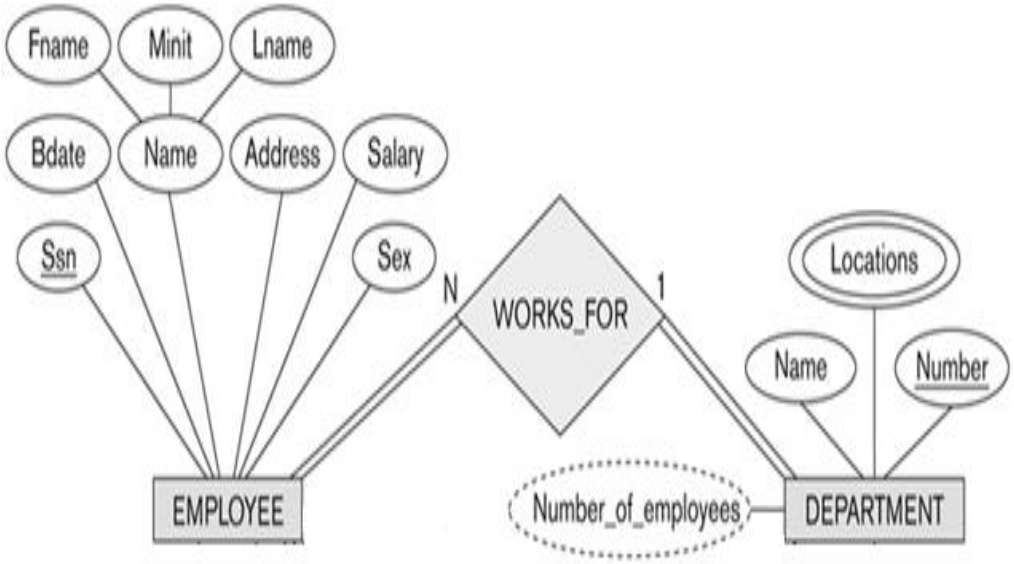
وبذلك يرتبط جدول (Department) بجدول (Employee) بوجود المفتاح الأجنبي رقم الموظف (Employee–Ssn).



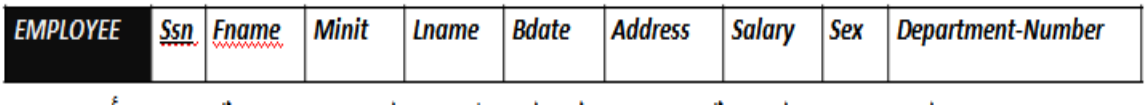
3.ب. تحويل علاقة واحد إلى كثير:

في هذا المثال لدينا ثلاث علاقات من نوع واحد إلى كثير وهي:

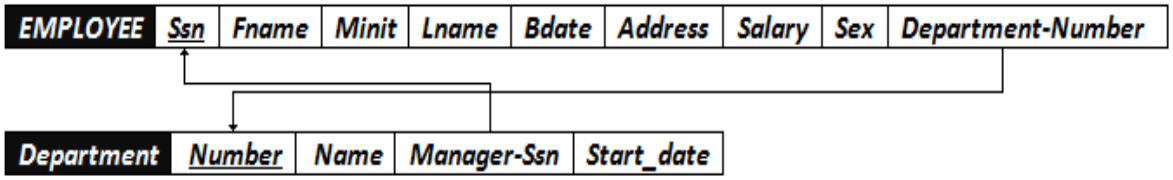
- **علاقة موظف يعمل في قسم (Works_for)**



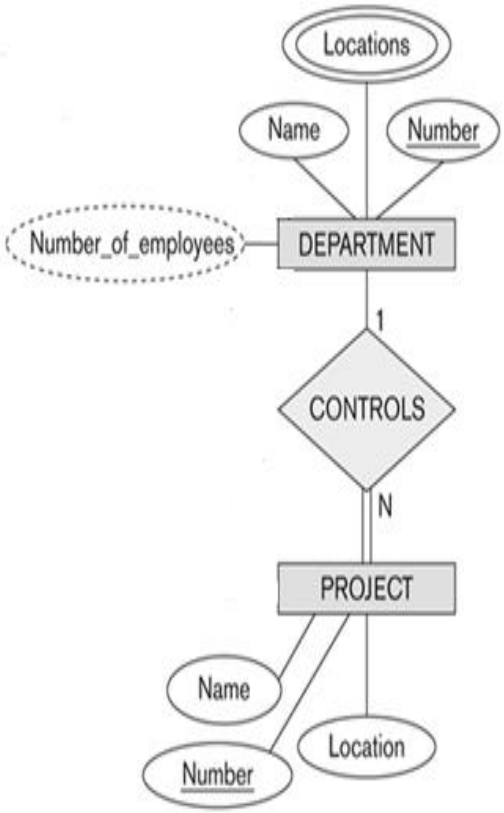
نتيجة لهذه العلاقة يتم التعديل على جدول الموظف (Employee) ليحتوي على رقم القسم (Department_Number) الذي يعمل فيه، ليصبح جدول الموظف (Employee) بالشكل التالي:



وعليه تتوثق العلاقة بين جدول الموظف والقسم، بإضافة مفتاح أجنبي جديد، وهو رقم القسم في جدول الموظف، كما يظهر في الشكل التالي:



- **علاقة قسم يتحكم بمشروع (Controls)**



نتيجة لهذه العلاقة يتم التعديل على جدول المشروع (Project) ليحتوي على رقم القسم (Department_Number) الذي يتحكم في المشروع، ليصبح جدول المشروع (Project) بالشكل التالي:

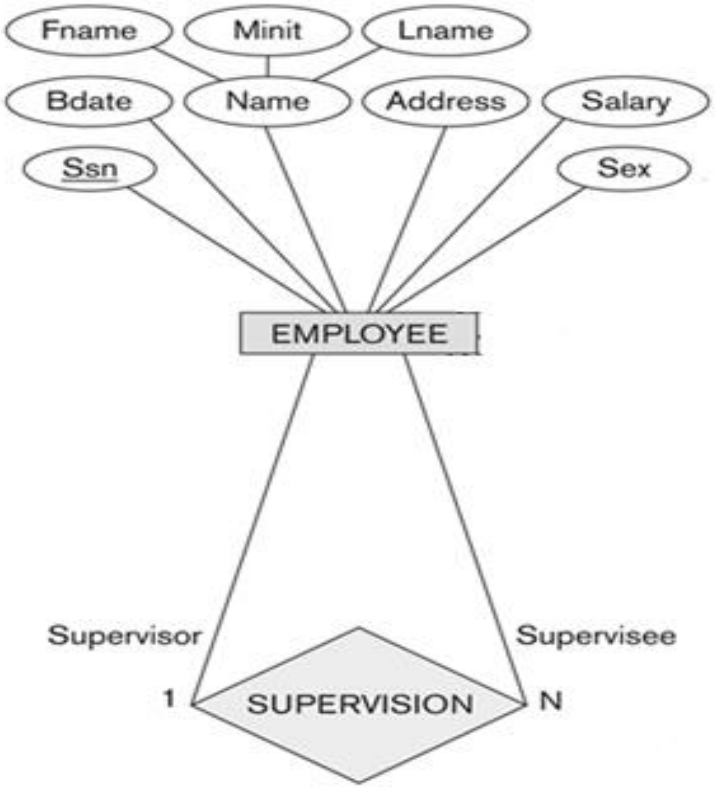
Project	***Number***	***Name***	***Location***	***Department_Number***

وعليه يرتبط جدول القسم بجدول المشروع بواسطة المفتاح الأجنبي رقم القسم (Department–Number) الذي تم إضافته إلى جدول المشروع، كما يظهر في الشكل التالي:

Department	***Number***	***Name***	***Manager-Ssn***	***Start_date***

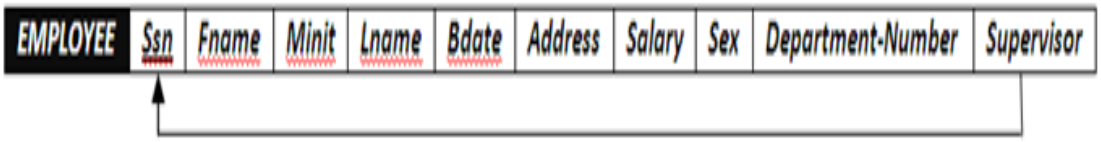
Project	***Number***	***Name***	***Location***	***Department_Number***

- **علاقة إشراف موظف على موظف (Supervision)**



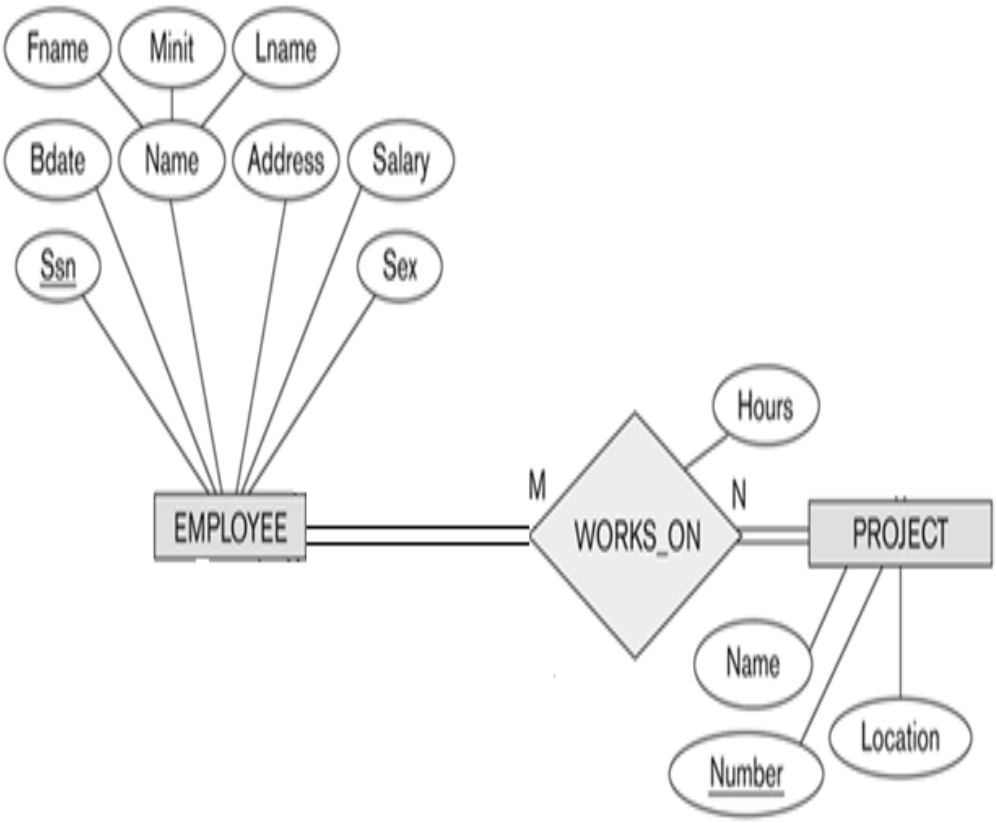
هذه العلاقة هي علاقة أحادية ذات تغذية راجعة أي علاقة كيان على نفسه.

نتيجة هذه العلاقة يتم إضافة حقل جديد هو حقل المشرف (Supervisor)، وهو حقل يعبر عن رقم الموظف، مع إعطائه خصوصية كونه مشرفا على غيره من الموظفين. الحقل يتم إضافته إلى جدول الموظف (Employee) ليصبح بالشكل التالي:



3.ج. تحويل علاقة كثير إلى كثير :

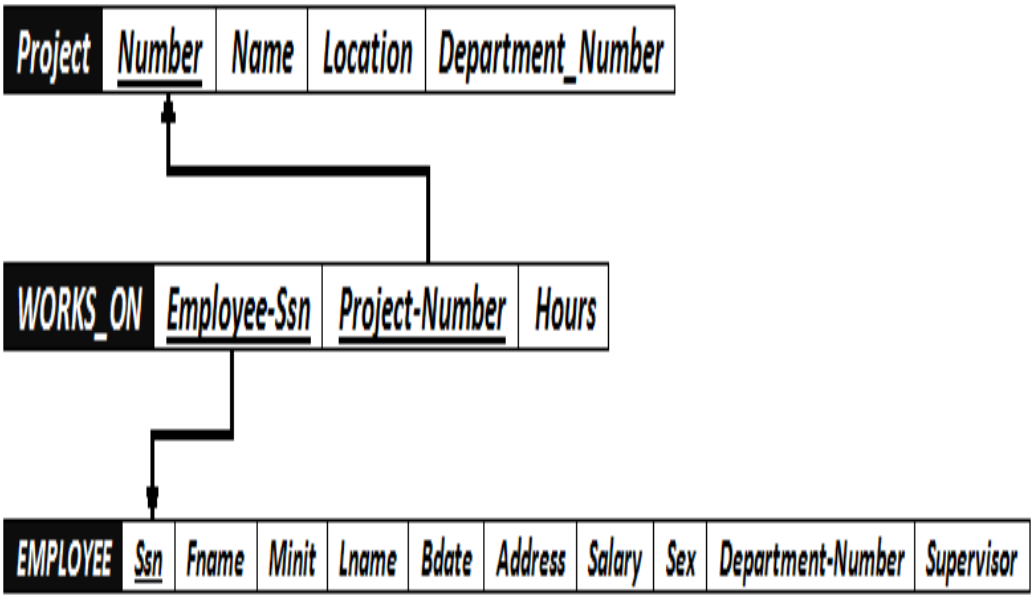
في هذا المثال، يوجد لدينا علاقة واحدة من نوع كثير إلى كثير، هي علاقة موظف يعمل على مشروع (Works_on).



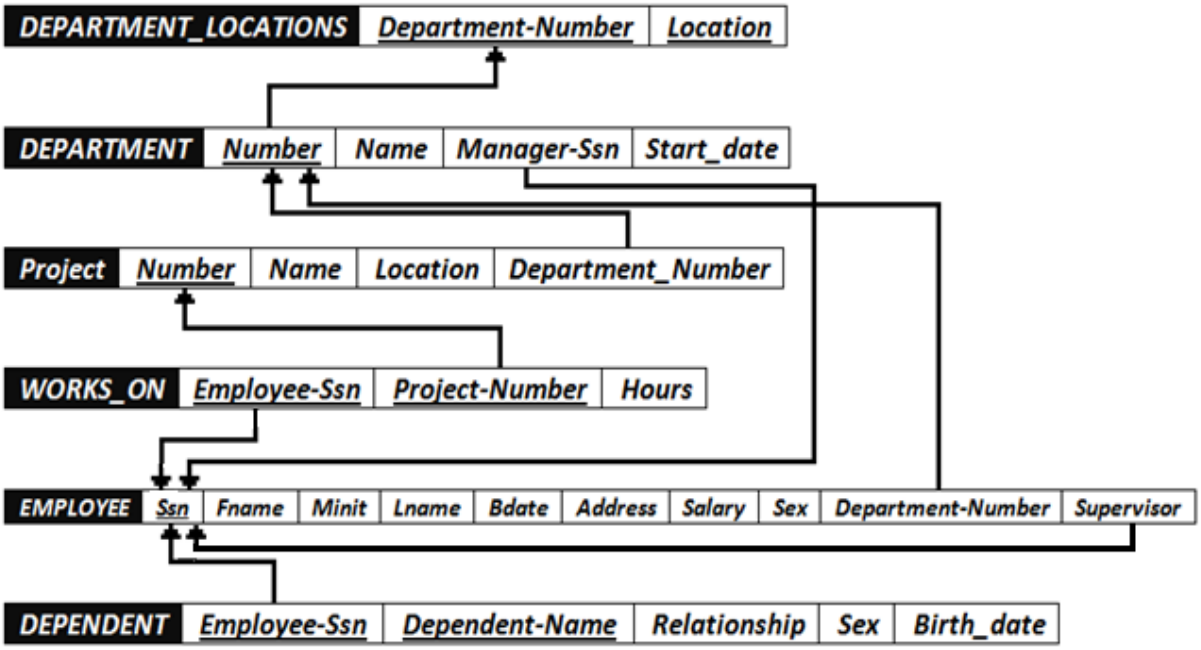
وفي هذه الحالة يتم إنشاء جدول جديد تحت اسم العلاقة (Works–on)، ونقوم بأخذ المفتاح الرئيس من كل الجدولين، الموظف (Employee) والمشروع (Project)، بالإضافة إلى الصفة عدد الساعات (Hours)المرتبطة بالعلاقة، ويصبح لدينا الجدول التالي:

WORKS_ON	Employee-Ssn	Project-Number	Hours

وبذلك يرتبط جدول الموظف (Employee) وجدول المشروع (Project) عن طريق الجدول الجديد (Works_on) عن طريق المفاتيح الأجنبية رقم الموظف (Employee–Ssn) من جدول الموظف Employee، ورقم المشروع (Project–Number) من جدول المشروع (Project).



وبذلك نكون قد أنشأنا مخطط قواعد البيانات الناتج عن التحويل للمثال رقم (2)، وهو على الشكل التالي:



تحليل نظم المعلومات
Information Systems Analysis

إعداد: م. هناء الصباغ

10

المحاضرة العاشرة

10. قواعد التطبيع Normalization Rules

قواعد التطبيع Normalization rules

هي عملية تقسيم البيانات إلى مجموعة من العلاقات التي تتسم بالتجانس والاعتمادية المتبادلة والخالية من التكرار.
وبالطبع فإن القضاء على التكرارية تماما أمر غير ممكن، وما يتم دائما هو تقليل التكرارية ما أمكن كما سيأتي، إن قواعد التطبيع هي مجموعة المعايير المتبعة في تصميم قواعد البيانات، وهي خطوات لتجزئة العلاقات relations (الجداول)، إلى علاقات أصغر.
ورغم اهمية عملية نمذجة البيانات في مخطط كينونة علاقة، أو ما يسمى بقاعدة البيانات الأولية، فإن تطبيع البيانات أو قواعد تسوية البيانات لا تقل أهمية عنها في نظرية قواعد البيانات. إن تطبيع البيانات كما هو واضح من الاسم هو عملية جعل البيانات طبيعية أكثر مقارنة بالنموذج الرياضي المثالي.

لماذا نقوم بتطبيع البيانات ?Why Normalization:

- لمعرفة وفهم المعنى الكامن في البيانات التي تحتويها قاعدة البيانات.
- لمعرفة وفهم العلاقات والإعتماديات المتبادلة بين البيانات، ويقصد بالإعتماديات المتبادلة كون معالجة بيانات معينة (إدخال تعديل أو حذف) تعتمد على معالجة بيانات أخرى.
- لتقليل تكرارية البيانات (data redundancy) والتناقض الذي قد يحصل بينها.
- لحماية البيانات من المشاكل التي قد تطرأ عند معالجتها، أي عند إجراء إدخال بيانات جديدة أو تعديل بيانات موجودة أو حذفها.

مميزات تطبيق قواعد التطبيع Normalization Advantages :

يمكن إجابة السؤال السابق عن طريق إجابة سؤال آخر، ما هي مميزات تطبيق قواعد التطبيع؟، كما يلي:

- تقليل تكرارية البيانات من خلال تصميم جداول قاعدة البيانات الفيزيائية، وذلك يؤدي إلى تقليل مساحة التخزين على القرص الصلب.
- تقليل التضارب الحاصل بين البيانات.(التضارب هو عدم التجانس inconsistency)
- تقليل تكلفة تحديث البيانات ومعالجتها.
- إلغاء العلاقات التي من نوع many-to-many، وذلك بتحويلها إلى علاقات من نوع one-to-many .
- زيادة وتطوير مرونة النظام.

عيوب تطبيع البيانات:

و كأي شيء في الحياة، فكما أن لتطبيق قواعد التطبيع مميزات، فإن له عيوبا منها:

- الاضطرار (أحيانا) إلى استخدام الفهارس التي تعقد عملية المعالجة وإن كانت ترفع من كفاءتها.
- زيادة تعقيدات النظام.
- تقليل كفاءة بعض أوامر استرجاع البيانات، خاصة العلاقات المعقدة، التي يتم عمل ارتباطات كثيرة لها عند الاسترجاع.

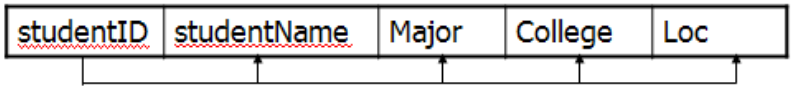
وبسبب هذه العيوب، يتم في بعض الأحيان، تطبيق قواعد التطبيع العكسية، أو ما يسمى Demoralization.
تمرين للبحث: ما هي قواعد عكس التطبيع Demoralization؟ وضح بمثل؟

الاعتمادية الوظيفية Functional Dependencies:

هي وجود قيد بين صفتين two attributes (عمودين)، أو بين مجموعتين من الصفات.
فعندما نقول أن العمود B يعتمد وظيفيا على العمود A في نفس الجدول فإننا نرمز لذلك بـ:

$A \longrightarrow B$ (العمود A يحدد العمود B، أو العمود B يعتمد على A)

في المثل التالي، الجدول student يحتوي على الحقول أو الأعمدة الأربعة التي تعتمد على العمود studentID، كما هو موضح أدناه.



العمود studentID يحدد مجموعة الأعمدة كما هو موضح بالصيغة:

studentID ⟶ { studentName,Major,College,Log }

وبطبيعة الحال يمكن كتابة الجملة السابقة بطريقة اعتمادات وظيفية منفصلة، بحيث نشير إلى أن رقم الطالب يحدد اسم الطالب، وهكذا..

الاعتمادية الوظيفية الجزئية:

عندما يكون هناك عمود معتمد وظيفيا على مفتاح رئيسي مركب، فإن الاعتمادية الجزئية تعني أن العمود معتمد على جزء من الفتاح الرئيسي، وليس كله.

ويعبر عن ذلك كالتالي:

إذا كان العمود A مفتاحا رئيسيا مركبا، بحيث أن A=BC، وكان لدينا العمود D بحيث أن:

A→D، فإننا نقول أن D يعتمد جزئيا على A، إذا كان العمود C فقط هو الذي يحدد D.

ونقول أن العمود D يعتمد كليا على A، إذا كان كلا من العمودين B و C يحددان D معا.

الاعتمادية الوظيفية الكلية أو الكاملة:

تكون هناك اعتمادية كاملة في حالتين:

- كان المفتاح الذي يحدد الأعمدة الأخرى غير مركبا، أي مفردا.
- أو كان المفتاح مركبا، وكل مكوناته تحدد الأعمدة الأخرى معا.

الاعتمادية الوظيفية المتعدية:

إذا كان العمود A يحدد العمود B وكان العمود B يحدد C، فإننا نسمي الجملة A→C إعتمادية متعدية، لأنها نتجت عن الجملة A→B، و B→C.

الأشكال الطبيعية Normal Forms:

هي حالات أو حالة خاصة بالعلاقة relation (الجدول)، والتي تنتج من تطبيق قواعد بسيطة متعلقة بالاعتمادية الوظيفية على تلك العلاقة.

وقواعد التطبيع هي مجموعة من الشروط المتتالية، بحيث لا يعتبر الثاني محققا إلا بعد تحقق الأول، والثالث يعتمد في تحققه على الثاني، أي على تحقق الأول والثاني معا، وهكذا، تقدم هذه الشروط مجموعة خصائص يفترض تحققها في كل قاعدة، وهذه القواعد هي:

- 1st Normal Form (1NF)
- 2nd Normal Form(2NF)
- 3rd Normal Form(3NF)
- Boyce-Codd Normal Form(BCNF)
- 4th Normal Form(4NF)
- 5th Normal Form(5NF)

القاعدة الأولى - الشكل الطبيعي الأول first normal form (1NF):

تشترط هذه القاعدة من أجل وصول العلاقة (أو الجدول) إلى الصيغة أو الشكل الطبيعي الأول، منع وجود القيم المتعددة في حقل معين ومنع وجود القيم المركبة أو القيم المشتقة في أي حقل، وبالتالي يجب أن تكون القيم في مجال الحقل قيماً ذرية Atomic أولية، بمعنى تكون بسيطة وغير قابلة للتفكك.

القاعدة الثانية - الشكل الطبيعي الثاني second normal form (2NF):

تشترط هذه القاعدة من أجل الوصول الجدول (أو العلاقة) إلى الصيغة الطبيعة الثانية:

١) أن تكون العلاقة (الجدول) في الشكل الطبيعي الأول.

٢) أن لا تكون هناك أي اعتمادية وظيفية جزئية، بمعنى أن أي اعتمادية وظيفية يجب أن تكون كاملة كما سيأتي.

القاعدة الثالثة - الشكل الطبيعي الثالث third normal form (3NF):

تشترط هذه القاعدة من أجل الوصول الجدول (أو العلاقة) إلى الصيغة الطبيعة الثالثة:

١) أن تكون العلاقة في الشكل الطبيعي الثاني.

٢) أن لا تكون هناك أي اعتمادية وظيفية متعدية، بحيث يتم تحويل الاعتمادية المتعدية إلى متعدية كاملة.

القواعد الأخرى:

سنكتفي فيما يلي بدراسة القواعد الثلاث الاساسية مع العلم بوجود قواعد أخرى أكثر تعقيدا هي:

قاعدة بويس- كود BCNF، وقاعدة الشكل الطبيعي الرابع، والشكل الطبيعي الخامس.

ما هي المخرجات الناتجة عن تطبيق قواعد التطبيع؟

المخرجات الناتجة هي قاعدة بيانات منطقية سليمة وخالية من العيوب، على أن تكون على الأقل في الشكل الطبيعي الثالث، وتكون حقول أو أعمدة كل جدول معتمدة كليا على عمود واحد أو عدة أعمدة تمثل المفتاح الاساسي فقط لا غير.

ملاحظة هامة:

قواعد تطبيع البيانات ليست مجرد قواعد رياضية صارمة فقط، فتطبيقها يعتمد أساسا على فهم قاعدة البيانات، وفهم طبيعة كل عمود في العلاقة وطبيعة البيانات التي تخزن فيه، وهذه شيء يعود دائما على مصمم قاعدة البيانات ومدى فهمه لمرحلة تحليل النظام.

أمثلة توضيحية حول قواعد التطبيع

١) القاعدة الأولى 1NF:

- إذا كانت العلاقة غير مطابقة للشكل الطبيعي الأول، فإننا نقول أنها في الشكل الطبيعي صفر ونرمز لذلك بـ **0NF.**
- نقوم بفحص العلاقة في الشكل الطبيعي صفر، ونقوم بتحويل العمود ذو القيم المركبة أو متعددة القيم إلى قيم بسيطة، إما بتحويل العمود الواحد إلى عدة أعمدة، أو بعمل علاقة جديدة تحوي القيم المتعددة أو المركبة، **بالإضافة إلى** مفتاح أجنبي يمثل الجدول الأصلي بالطبع.
- المثال التالي يوضح وجود الحقلين children و gender اللذين يجعلان العلاقة EMP **من الشكل** 0NF، هذا **بالإضافة إلى الحقل** Nun_children الذي يحتوي كما هو واضح على قيمة مشتقة، هي عدد الأطفال.

Emp_ID	Children	Gender	Nun_children
002	Nan, Pat, Lee	F, M, M	3
006	Todd	M	1

- **وللعودة بالعلاقة** EMP **إلى الشكل** الطبيعي الأول 1NF، نقوم بتحويل الحقلين متعددي القيم إلى أكثر من حقل، ونقوم بحذف العمود ذا القيمة المشتقة.

Remove derived attributes

Emp_ID	Ch1	G1	Ch2	G2	Ch3	G3	Ch4	G4	Nun_children
002	Nan	F	Pat	M	Lee	M			3
006	Todd	M							1

- فيصير الجدول EMP **كالتالي:**

Emp_ID	Ch1	G1	Ch2	G2	Ch3	G3	Ch4	G4
002	Nan	F	Pat	M	Lee	M		
006	Todd	M						

- وهكذا نحكم أن الجدول EMP صار في الشكل الطبيعي الأول 1NF.

٢)القاعدة الثانية 2NF:

- علينا دائما قبل التحقق من شروط الشكل الطبيعي الثاني، ان نقوم بالتأكد من تحقق الشكل الطبيعي الأول، أي أن العلاقة تحقق القاعدة 1NF.
- نهدف دائما في الشكل الطبيعي الثاني ان نحول أي اعتمادية جزئية إلى اعتمادية كاملة.
- إذا كان المفتاح الاساسي في العلاقة مكونا من عمود واحد فهذا يعيني بالضرورة عدم وجود إعتمادية جزئية.
- وفي حالة وجود مفتاح رئيسي مركب فإننا نبدأ في التأكد من عدم وجود إعتمادية جزئية كما يوضح المثال التالي.
- الجدول أدناه يمثل اسماء مجموعة من الاشخاص الذي يشترون أسهم تابعة لشركات مختلفة.

Cust_ID	Name	Addr	Stock	Curr_price	Dividend	Shares
003	Smith	LA	IBM	100	3	16
019	Jones	NY	USX	50	2	5
102	Harris	KC	IBM	100	3	10
102	Harris	KC	CVD	1	0	1000

- نلاحظ أن المفتاح الاساسي المناسب للجدول أعلاه، هو المفتاح المركب المكون من الحقلين رقم الزبون cust_ID وجهة السهم stock (الشركة التي تبيع السهم).
- وإذا قمنا بفحص الاعتمادية نلاحظ التالي:

- Cust_ID → Name,Addr (Partial FD)
- العمود (رقم الزبون) يحدد العمودين (الاسم، والعنوان) .....اعتمادية جزئية.
- Stock → Curr_price,Dividend (Partial FD)
- العمود جهة السهم يحدد العمودين (السعر الحالي) و (الربح) ... اعتمادية جزئية.

- Cust_ID,Stock → Shares (full FD)

- المفتاح الأساسي المركب من رقم الزبون وجهة السهم يحدد العمود عدد الاسهم .. وهذه فقط هي الاعتمادية الكاملة.
- وبالطبع نحذف الاعتماديات الوظيفية الجزئية محولين الجدول السابق إلى ثلاثة جداول، يحتوي الجدول الأصلي على الاعتمادية الكاملة، ويظهر جدولين بكل واحد منهما الاعتمادية الجزئية التي تصير عند فصلها اعتمادية كاملة.

Cust_ID	Stock	Shares
003	IBM	16
019	USX	5
102	IBM	10
102	CVD	1000

Cust_ID	Name	Addr
003	Smith	LA
019	Jones	NY
102	Harris	KC
102	Harris	KC

Stock	Curr_price	Dividend
IBM	100	3
USX	50	2
IBM	100	3
CVD	1	0

- وبحذف الأعمدة الجاهزة ينتج لدينا الجداول الثلاثة التالية:

Cust_ID	Stock	Shares
003	IBM	16
019	USX	5
102	IBM	10
102	CVD	1000

Cust_ID	Name	Addr
003	Smith	LA
019	Jones	NY
102	Harris	KC

Stock	Curr_price	Dividend
IBM	100	3
USX	50	2
CVD	1	0

- الجداول أو العلاقات السابقة relations في الشكل الطبيعي الثاني 2NF.

٣)القاعدة الثالثة 3NF:

- بعد أن تحقق من كون العلاقة في الشكل الطبيعي الثاني نبدأ بالتحقق من تحقق الشكل الطبيعي الثالث.
- نهدف في الشكل الطبيعي الثالث القضاء على أي إعتمادية متعدية، وتحويل العلاقة إلى الاعتمادية الكاملة، ويكون ذلك غالبا بفصل أعمدة الاعتمادية المتعدية عن العلاقة الأولى.
- علينا أن نتأكد من كون كل الاعمدة تعتمد على المفتاح الاساسي مباشرة، ولا تعتمد عليه عن طريق عمود (غير مفتاحي) آخر.
- في المثال التالي العلاقة تمثل جدولا يتكون من رقم الصنف كمفتاح اساسي، واسم الصنف والشركة المصنعة وبلد الشركة المصنعة كأعمدة غير مفتاحية.

Product#	Name	Mfr	Mfr_HQ
1001	Walkman	Sony	Japan
1002	Camera	Leica	Germany
1003	VCR	Sony	Japan

- واضح تماما أن المفتاح الاساسي يحدد اسم الشركة المصنعة، وأن اسم الشركة يحدد بلد الشركة، وعليه فإن العلاقة المتعدية هي كون المفتاح الاساسي (رقم المنتج) يحدد بلد الشركة المصنعة.

Product# ⟶ Mfr

Mfr ⟶ Mfr_HQ

Therefore Product# ⟶ Mfr_HQ

- يجب إلغاء العلاقة المتعدية بتحويل الجدول السابق إلى جدولين يتكون كل واحد من إعتمادية كاملة كما يلي:

Product#	Name	Mfr	Mfr_HQ
1001	Walkman	Sony	Japan
1002	Camera	Leica	Germany
1003	VCR	Sony	Japan

Product#	Name	Mfr
1001	Walkman	Sony
1002	Camera	Leica
1003	VCR	Sony

Mfr	Mfr_HQ
Sony	Japan
Leica	Germany

- كما هو واضح من الشكل السابق تم فصل الجدول (المنتج)، إلى جدولين هما المنتج و جدول الشركات فيه المفتاح الاساسي لاسم الشركة Mfr، بحيث يكون مفتاح اجنبيا في جدول المنتج، والذي يظل المفتاح الاساسي به كما كان سابقا.

تمرين للطالب:

قم بتطبيق قواعد التطبيع normalization على الجدول التالي، الذي يظهر أنه في الشكل الطبيعي الصفري 0NF، حتى وصوله إلى الشكل الطبيعي الثالث 3NF، علما أنه يمثل قاعدة بيانات نظام تأجير أقراص مدمجة CDs.

Receipt	Date (M/D/Y)	PID	PName	Phone	City	State	VID	VName	Type	Days	Cost
68395	9/1/01	226	Lindsey Moore	(734)763-4385	ST Ann Arbor	MI	325. 548. 6437	Shrek, Remembering, The Replacement Killers	N, H, H	1, 2, 2	3.00. 2.50. 2.50
68397	9/4/01	224	Helen Baker	(743)763-2138	Chicago	MI	325. 6437	Shrek, The Replacement Killers	N, H	1, 2	3.00. 2.50
69001	10/2/01	226	Lindsey Moore	(734)763-4385	ST Ann Arbor	MI	468	True Lies	N	1	3.00

وأن قاموس بيانات الجدول كالتالي:

اسم العمود	البيان
Receipt	رقم إيصال الإستلام الذي يأخذه المستأجر معه.
Date	تأريخ عملية التأجير.
PID	رقم الزبون الذي قام بعملية الاستئجار.
Pname	اسم الزبون.
Phone	رقم هاتف الزبون.
City	العنوان ـ المدينة
State	العنوان ـ الولاية
VID	أرقام الافلام التي في الأقراص المستأجرة.
Vname	اسماء الافلام المستأجرة.
Type	نوع الفيلم أو نوع القرص؟.
Days	عدد ايام الايجار لكل فيلم.
Cost	تكلفة الاستئجار وهو المبلغ الذي على الزبون دفعه.