1433

قـــــــواعـــــــــد البـــــيانـــات

تعديل وتلخيص : ام يزن 1920

د. مصلح العضايلة

1433

المحاضرة الأولى

مقدمة في أنظمة قواعد البيانات

- أصبحت قواعد البيانات و تطبيقاتها عنصرا جوهرياً في تسيير أمور الحياة اليومية في المجتمع المعاصر ، حيث إن جميع الأنشطة التي يمارسها أفراد المجتمع من تسجيل مواليد ووفيات و نتائج دراسية و وثائق السفر و العمليات البنكية و غيرها الكثير يجب فيها التعامل مع احد قواعد البيانات
- كافة الأنشطة السابقة تدخل في نطاق التطبيقات التقليدية لقاعدة البيانات .
- توجد حالياً تطبيقات متقدمة لقواعد البيانات مثل استخدام الذكاء الاصطناعي و التجارة الالكترونية

خواص قواعد البيانات

- تمثل بعض مظاهر العالم الحقيقي. أي إنها تمثل حالة من حالات البيانات التي تصف موضوع حقيقي
- تمثل مجموعة من البيانات المتلاصقة منطقيا وتحتوي على معنى ضمني
- يتم تصميمها و تخزين البيانات فيها من أجل غرض معين

مفهوم قواعد البيانات

* **قواعد البيانات (Database):** هي أسلوب محدد لتنظيم المعلومات يبسط كيفية إدخالها و تعديلها و استخراجها في صورة ملائمة و مفهومة للمستخدم لمجموعة مشتركة من البيانات المترابطة والمتجانسة منطقيا .
* هي مجموعة من عناصرِ البيانات المنطقية المرتبطة مع بعضها البعض بعلاقة معينة، وتتكون قاعدة البيانات من جداول (واحد أو أكثر). ويتكون الجدول أعمدة (حقول Fields) ومن صفوف (سجلات Record).

نظم ملفات البيانات

- استخدام الملفات في تخزين البيانات.
- استخدام المبرمجون ملفات البيانات في تخزين المعلومات لفترة طويلة.
- أدى استخدام الملفات إلى ظهور بعض المشاكل والعيوب .
- أدى إلى تطوير أسلوب التعامل مع الملفات وبذل الجهد والوقت
- في نظام معالجة البيانات كان كل برنامج يصمم لأداء غرض معين وله الملفات الخاصة به ، دون وجود إطار عام يربط جميع البرامج أو يسمح بإضافة برامج جديدة بسهولة.

أنواع الملفات:

* ملف تتابعي: يتم تخزين سجلات البيانات بشكل تتابعي بنفس ترتيب وصولها للملف سجل بعد سجل. لاسترجاع البيانات تجري عملية قراءة السجلات من أول سجل إلى آخر سجل و بشكل تتابعي.
* ملف عشوائي : يتم تخزين سجلات البيانات بشكل عشوائي مع معرفة موقع أو عنوان كل سجل بيانات، و تتم قراءة البيانات مباشرة عن طريق العنوان.
* ملف مفهرس: يستخدم فهرس أشبه بفهرس الكتاب من خلاله يتم الوصول إلى أي سجل بيانات، يتم عمل الفهرس من خلال احد حقول البيانات.

مشاكل الملفات:

* تكرار البيانات: تكرار البيانات في أكثر من ملف مما يضيع حيز التخزين و الجهد و الوقت .
* عدم تجانس أو توافق البيانات: نفس المعلومة تكون مخزنه في أكثر من ملف عند تعديلها قد لا نعدلها في الملفات الأخرى.
* عدم المرونة:عملية التعديل و الحذف تتطلب جهد و وقت و كلفة عالية.
* الافتقار إلى المواصفات القياسية.
* معدل منخفض لإنتاج البرامج
* مشاركة محدودة جداً بين البرامج المختلفة و ملفات البيانات.
* صعوبة الصيانة أي تعديل لملف يلزم تعديل كافة البرامج الخاصة به .
* امن سرية المعلومات تكون على نطاق محدود.

نظم قواعد البيانات

* نشأت قواعد البيانات و نظم قواعد البيانات من اجل إيجاد بديل لملفات البيانات و نظم معالجتها بحيث تحل كافة المشكلات و القيود و الصعوبات التي يواجهها المستخدمون في تعاملهم مع الملفات.
* البيانات : هي كافة البيانات المطلوب إدخالها أو الاستعلام عنها ، حيث كل بيان يمثل عنصر مستقل مثل (اسم المريض ، رقم الغرفة ، العنوان ، ....)
* المعلومات : هي البيانات التي تمت معالجتها و وضعها في صورة ملائمة و مفهومة للمستخدم.
* نظم قواعد البيانات هي أسلوب محدد لتنظيم البيانات يبسط كيفية إدخالها و تعديلها و استخراجها إما بنفس الشكل المدخل أو مجمعة في صورة إحصائية أو تقارير أو شاشات استعلام مع التحكم في كل عملية.
* تصميم قاعدة البيانات يشمل تحديد أنواع البيانات و التراكيب و القيود على كافة البيانات.
* بناء قاعدة البيانات هو عملية تخزين البيانات نفسها في وسط تخزين تتحكم به نظم قواعد البيانات .
* عند تصميم قاعدة بيانات يجب تحديد المستخدمين و التطبيقات الذين سيستخدمون قاعدة البيانات.

أمثلة نظم إدارة قواعد البيانات

يمكن إنتاج و معالجة قاعدة البيانات باستخدام الحاسب الآلي بواسطة مجموعة من البرامج التطبيقية المصممة خصيصاً لهذا الغرض أو بواسطة نظم إدارة قواعد البيانات (DBMS) مثل:

- MS-Access
- Oracle
- Sybase

- Power Builder
- Informix

الفرق بين نظم قواعد البيانات ونظم الملفات التقليدية

يوجد العديد من الخواص التي تفرق بين قواعد البيانات ونظم الملفات التقليدية وهي:

- **الوصف الذاتى للبيانات (Self-Description Nature):**

تحتوى قواعد البيانات علي البيانات ووصف البيانات وذلك عن طريق إنشاء فهرس البيانات والذي يحتوي على ما يسمى (Meta-data)

- **الفصل بين البرامج والبيانات(Program/Data Insulation):**

لا تحتوى البرامج على وصف البيانات بل يوجد فصل بينهما مما يتيح إمكانية تعديل شكل البيانات بدون الحاجة لتعديل البرامج

- **المشاركة في البيانات والتعامل مع العديد من المستخدمين (Data Sharing and Multi-user system):**

تتيح قواعد البيانات المشاركة في استخدام البيانات وكذلك تعطي إمكانية تعامل العديد من المستخدمين مع نفس قواعد البيانات في نفس الوقت بدون مشاكل

قواعد البيانات و نظم إدارة قواعد البيانات

- نظام إدارة قواعد البيانات: هو مجموعة من البرامج التي يمكن استخدامها في إنشاء و معالجة قاعدة بيانات .
- نظام إدارة قواعد البيانات هو نظام برامجي متعدد الأغراض يسهل تعريف و بناء و معالجة قواعد البيانات التطبيقية.
- يمكن أن تصمم قاعدة بيانات واحدة تستخدم مع العديد من البرامج والتطبيقات.

خواص أخرى لقواعد البيانات

- يمكن أن تكون قاعدة البيانات في أي حجم فيمكن أن تحتوي على القليل من السجلات أو المئات منها ويمكن أن تحتوي على مئات الملايين من السجلات
- يمكن أن يتم إنشائها و التعامل معها يدويا أو باستخدام الحاسبات الآلية
- إذا تم استخدام الحاسب الآلي لإدارة قواعد البيانات فإن ذلك يتم عن طريق مجموعة من البرامج التي تصمم خصيصا لذلك أو عن طريق استخدام نظم إدارة قواعد البيانات (Database Management System DBMS)

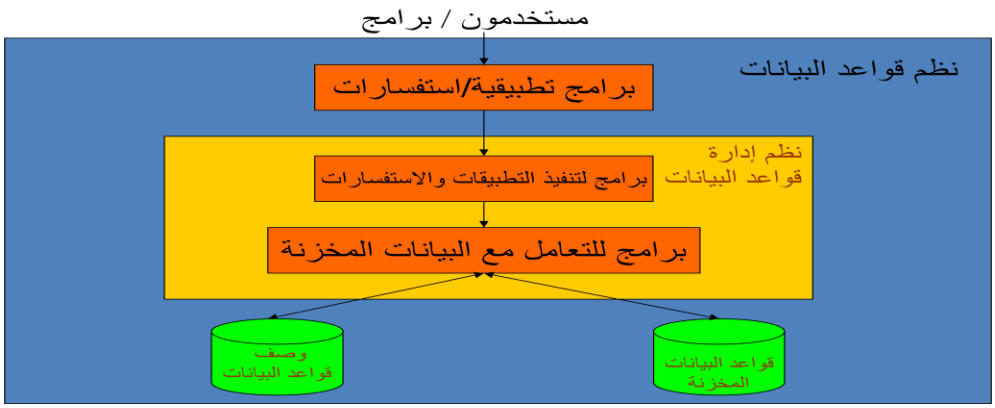
نظام إدارة قواعد البيانات (Database Management System DBMS)

- تسمى قواعد البيانات + نظم إدارة قواعد البيانات بنظم قواعد البيانات

(قواعد البيانات + نظم إدارة قواعد البيانات ← نظم قواعد البيانات)
(Database (DB) + DBMS → Database System (DBS))

مميزات استخدام قواعد البيانات

- ندرة التكرار و إمكانية التحكم في تكرار البيانات
- امن و سرية البيانات عالية جداً
- فرض القيود على المستخدمين الذين ليس لهم صلاحيات معينة
- توفير بيئة تخزين مناسبة و صعوبة فقد البيانات
- السماح باستنباط معلومات من البيانات المتواجدة
- توفير واجهات متعددة لتعامل المستخدم مع البيانات
- تمثيل العلاقات المعقدة بين البيانات بسهولة
- تكامل البيانات بشكل عالي و متناسق.
- سهولة الصيانة حيث اي تعديل يتم بكل سهولة و من مكان واحد
- توفير طرق متعددة للحصول على النسخ الاحتياطية و كذلك معالجة البيانات فى حالات الأعطال التى قد تحدث لقواعد البيانات
- تساعد على وضع معايير قياسية للتعامل مع البيانات
- تقليل زمن تطوير البرامج
- المرونة الشديدة في استخدام وتعديل البيانات
- توفير بيانات على درجة عالية من التحديث
- اقتصادية الاستخدام
- المرونة العالية في مشاركة البيانات و بكل سهولة



مستخدم قواعد البيانات

- **مدير قواعد البيانات(DBA):**
  - هو الذي يقوم بإدارة قواعد البيانات والتحكم في صلاحيات العمل ومراقبة النظام وتحسين أداء قواعد البيانات
- **مصمم قواعد البيانات (DB Designer):**

•يقوم بتصميم قواعد البيانات ليتم إنشائها وبنائها بطريقة ذات كفاءة عالية طبقا لمتطلبات المستخدم

•مستخدم قواعد البيانات (End User):

- بعض المستخدمين يكون لديهم الخبرة الكافية لإعداد الاستفسارات المطلوبة بلغة الاستفسارات، وبعض المستخدمين ليس لديهم الخبرة فيتم إنشاء برامج خاصة لهم يقومون بتشغيلها للحصول على المطلوب

•محلل النظم ومبرمج النظم(Analyst & Programmer):

•يقوم محلل النظم بتحديد متطلبات المستخدم وتطوير هذه المواصفات المطلوبة لتحديد المطلوب من قواعد البيانات

•بينما يقوم مبرمج النظم بتنفيذ المتطلبات لإنشاء التطبيقات المناسبة

•هندسة النظم هي عملية تحليل النظام بالإضافة لعملية إنشاء البرامج التطبيقية

(محلل النظم + مبرمج النظم ← مهندس النظم)

(Analyst + Programmer → Software Engineer)

متى لا نستخدم قواعد البيانات؟

•إذا كانت تكلفة الإعداد عالية بالنسبة لحجم المشروع

•إذا كانت قاعدة البيانات و التطبيقات بسيطة و سهلة

•إذا كان المشروع يحتاج لسرعة استجابة عالية جدا وبشكل ضروري

- إذا كان العمل لا يحتاج إلى بيئة ذات عدة مستخدمين

أشخاص يتعاملون مع قواعد البيانات بطريقة غير مباشرة

- هؤلاء الأشخاص لا يهتمون بقواعد البيانات ذاتها ولكنهم يقدمون لمستخدم قواعد البيانات البيئة اللازمة لهم وهم:

•مصممو ومنفذو نظم إدارة قواعد البيانات:

•هم الذين يقومون بتصميم وتنفيذ نظم إدارة قواعد البيانات نفسها

•مطورو البرامج المساعدة:

•الذين يقومون بتطوير البرامج المساعدة مثل برامج تحليل النظم، تصميم النظم، إنشاء وتطوير التطبيقات، إنشاء التقارير وواجهات التطبيق

•المشغلون وأفراد الصيانة:

•الذين يقومون بتشغيل النظم وإدارتها وصيانتها وكذلك صيانة البرامج والأجهزة المستخدمة في إنشاء وتطوير قواعد البيانات

المحاضرة الثانية

هيكلية نظم إدارة قواعد البيانات

Architecture of DBMS

مكونات بيئة نظم قواعد البيانات

- المكونات المادية: المكونات المادية من حواسيب وخوادم وأجهزة ومعدات.
- المكونات البرمجية: نظم البرمجة الخاصة بقواعد البيانات
- البيانات: هي العنصر المركزي لقواعد البيانات
- الإجراءات والعمليات: هي التعليمات التي تحكم التصميم واستخدام قواعد البيانات بالشكل الأفضل.
- المستخدمون: الأشخاص الذين يتعاملون مع قواعد البيانات

مبادئ قواعد البيانات

- قاعدة البيانات هي أسلوب محدد لتنظيم المعلومات يبسط كيفية الإدخال و الإخراج في إطارات مختلفة مع التحكم في كل عملية.
- أنواع قواعد البيانات
  - ✓ قواعد البيانات الشبكية (Network Database)
  - ✓ قواعد البيانات الهرمية (Hierarchical Database)
  - ✓ قواعد البيانات العلاقية (Relational Database)

هيكلية نظم إدارة قواعد البيانات

(DBMS Architecture)

- **النظام المركزي (Centralized system):**

  وفيه تتواجد جميع وظائف قواعد البيانات والنظم التطبيقية وواجهات التعامل مع المستخدم وغيرها من البرامج في نظام واحد مركزي

- **نظام الخادم - العميل (Client-Server):**

  وفيه يحتوي العميل (يكون عادة عبارة عن حاسب شخصي) النظم التطبيقية وواجهات التعامل مع المستخدم بينما يقوم الخادم بوظائف قواعد البيانات (وفي بعض النظم الحديثة قد يقوم العميل ببعض وظائف قواعد البيانات)

نماذج البيانات (Data Models)

- هو وصف للبيانات أو أنشطة أو إحداث في مكان ما لجعل البيانات منظمة و مفهومة.
- بناء البيانات هو تحديد نوع البيانات و العلاقات بين البيانات والقيود المفروضة عليها
- يمكن أن يحتوي نموذج البيانات على بعض العمليات الأساسية (مثل كيفية تعديل أو استرجاع البيانات)
- في نظم البيانات الشيئية يمكن أن يحتوي النموذج علي مجموعة من العمليات التي يعرفها المستخدم علي البيانات

حالات قواعد البيانات (Instances)

البيانات المتواجدة داخل قواعد البيانات في لحظة معينة تسمي "حالة قواعد البيانات أو الوضع الحالي لقواعد البيانات

(DB State or Current Set of Occurrence or Instance)

- يتم إنشاء الوضع الابتدائي لها عند إدخال البيانات لأول مرة ثم يتغير وضعها عند إجراء العمليات المختلفة على البيانات (إضافة – حذف – تعديل)
- تسمى حالة البيانات هذه "Extension"

هيكلية نظم قواعد البيانات (DB System)

- تحتوي نظم قواعد البيانات على ثلاث مستويات من المخططات وذلك لدعم الخواص التي يجب أن تقدمها نظم إدارة قواعد البيانات:
  - مستوى البيانات الخارجي(The External or View Level)
  - المستوى المفاهيمي (The Conceptual Level)
  - المستوي الداخلي (Internal Level)

مستوى البيانات الخارجي The External or View Level:

- هو الجزء الذي يستهدف المستخدمين
  للتخاطب والاتصال واسترجاع البيانات
  يستخدم برامج تطبيقية وبرامج رسومية أو مباشر
  مرحلة التحليل

المستوى المفاهيمي The Conceptual Level:

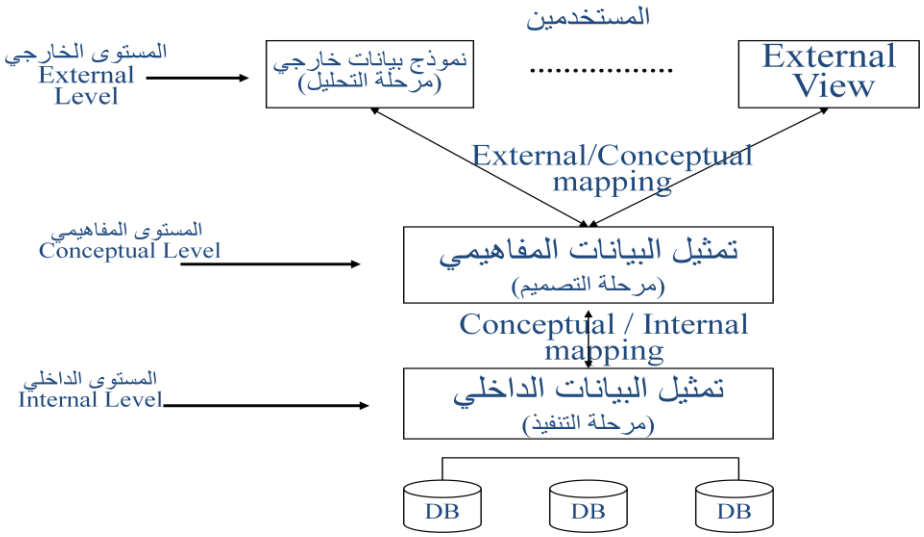
- يحتوي علي Conceptual Schema التي تصف بناء البيانات في قواعد البيانات – نموذج البيانات المنطقي
- تقوم بإخفاء التفاصيل الخاصة بالبناء الفعلي للبيانات
- تقوم بوصف الكيانات، نوع البيانات، العلاقات، القيود و كذلك العمليات التي يعرفها المستخدم

يمكن استخدام Conceptual data model أو Representation data model في بناء هذا المستوي و يطلق عليه مرحلة التصميم

المستوى الداخلي (Internal Level):

- وهو يحتوي علي المخطط الداخلي والذي يقوم بوصف التخزين الفعلي لقواعد البيانات و عملية إنشاء قاعدة البيانات.
- مرتبط بالأجهزة و البرامج
- هذا المخطط الداخلي يتم وصفه باستخدام نموذج (Physical Data Model) الذي يركز على تمثيل و إنشاء قواعد البيانات المصممة على جهاز الحاسب و يطلق عليه مرحلة التنفيذ.

شكل يوضح الثلاث مستويات لمخططات قواعد البيانات



(The Three-Schema Architecture)

ملاحظات علي الثلاث مستويات لمخططات نظم قواعد البيانات

- تعتبر طريقة مناسبة وأداة سهلة للمستخدم ليفهم و يتخيل مستويات مخططات البيانات داخل نظم قواعد البيانات
- معظم نظم إدارة قواعد البيانات لا تفصل تماما بين المستويات الثلاث
- تقوم نظم إدارة قواعد البيانات بتحويل المخططات بين المستويات الثلاث وتحويل البيانات بين هذه المستويات (mapping)
- التحويل بين المخططات (mapping) يعتبر عملية مستهلكة للوقت ولذلك فإن بعض نظم إدارة قواعد البيانات لا تدعم المستوي الثالث (External Level)
- معظم نظم إدارة قواعد البيانات تحتوي علي المستوي الثالث (External level) داخل المستوي الثاني (Conceptual level)
- بعض نظم إدارة قواعد البيانات تحتوي علي التفاصيل الفعلية (Physical details) داخل المستوي الثاني (Conceptual level)

استقلالية البيانات (Data Independence)

- هي المقدرة علي تغيير مخطط البيانات في مستوي معين بدون وجوب تغير المخطط في المستويات الأخرى
- عند تغيير المخطط في مستوى معين فإن الذي يتغير هو طرق التحويل (mapping) بين المستويات
- يوجد نوعان من استقلالية البيانات و هما :
  - ✓ الاستقلال المنطقي (Logical Data Independence)
  - ✓ الاستقلال الفعلي (Physical Data Independence)

الاستقلال المنطقي (Logical Data Independence):

هي المقدرة علي تغيير مخطط البيانات في المستوي الثاني (Conceptual Level) بدون الحاجة إلى تغيير المخطط في المستوي الثالث (External Level) وكذلك بدون تغيير البرامج التطبيقية

- يكون التغيير في المستوي الثاني لكي تستوعب قواعد البيانات التغيرات التي قد تحدث في المخطط نتيجة زيادة أو حذف عناصر بيانات
- التطبيقات التي تتعامل مع العناصر التي تغيرت هي فقط التي يتم تعديلها أما باقي التطبيقات فلا تتغير

الاستقلال الفعلي (Physical Data Independence):

هي المقدرة علي تغيير مخطط البيانات في المستوي الأول (Internal Level) بدون الحاجة إلى تغير المخطط في المستوي الثاني (Conceptual Level)

- يكون التغيير في المستوي الأول (Internal Level) بسبب التغيرات التي قد تحدث نتيجة استخدام أساليب جديدة في تنظيم الملفات من أجل تحسين أداء النظام
- التطبيقات التي تتعامل مع العناصر التي تغيرت هي فقط التي يتم تعديلها أما باقي التطبيقات فلا تتغير

لغات نظم إدارة قواعد البيانات

لغة وصف البيانات (Data Definition Language DDL):

- ✓ تستخدم بواسطة مدير قواعد البيانات (DBA) وكذلك مصمم قواعد البيانات لتعريف بناء قواعد البيانات
- ✓ يوجد مترجم للغة التعريف (DDL Compiler) وذلك لترجمة هذه اللغة وإنتاج برامج يتم تنفيذها لتقوم بإنشاء مخططات البيانات وتخزينها داخل فهرس قواعد البيانات (DB Catalog)

لغة تعريف الأشكال(View Definition Language VDL):

- ✓ تستخدم في بعض نظم إدارة قواعد البيانات التي تستخدم هيكل قواعد البيانات الثلاثي بطريقة حقيقية وذلك لتعريف مخطط البيانات في المستوي الخارجي (External Level)
  النماذج Forms الرسومية GUI التفاعل من خلال القوائم menu

- **لغة التعامل مع البيانات (Data Manipulation Language DML):**

تستخدم لاسترجاع وإدخال وحذف وتعديل البيانات

ملاحظات على لغات قواعد البيانات

- ✓ نظم إدارة قواعد البيانات الحالية تستخدم لغة واحدة شاملة تحتوي علي لغات DDL,VDL,DML
- ✓ لغة الاستفسار الهيكلية (SQL) هي لغة تستخدم مع نموذج البيانات العلائقي و تحتوي علي لغات DDL,VDL,DML وكذلك الجمل الخاصة بتعديل مخطط البيانات

بيئة نظم قواعد البيانات

نظم إدارة قواعد البيانات هي نظم معقدة وتحتوي على العديد من الوحدات التي تدعم ما يحتاجه المستخدم من وظائف ومنها:

- **مترجم لغة تعريف البيانات (DDL Compiler):**

لترجمة تعريف مخطط البيانات والتأكد من صحته ثم تخزين هذا التعريف داخل فهرس النظام

- **منفذ قواعد البيانات (Run-Time DB processor):**

يقوم بالتعامل مع قواعد البيانات عند تشغيل أي أمر خاص بقواعد البيانات

- **مترجم لغة الاستفسارات (Query Compiler):**

يتعامل مع الاستفسارات عن طريق فهم الأوامر وترجمتها ثم إرسالها إلى منفذ قواعد البيانات لتنفيذها

خدمات تقدمها نظم إدارة قواعد البيانات

تقوم بعض نظم إدارة قواعد البيانات بتقديم خدمات إضافية تساعد المستخدم في إدارة نظم قواعد البيانات مثل:

- **تحميل البيانات (Loading):**

وهي عبارة عن عملية تحويل البيانات الموجودة سابقا في النظم القديمة الي شكل ملائم للتصميم الجديد بدون الحاجة إلى إعادة إدخالها يدويا والذي يكون غير ممكن عمليا في كثير من الحالات. ويوجد بعض الأدوات المساعدة والتي تقوم بتحويل البيانات من الشكل القديم التي كانت عليه الي الشكل الجديد و الملائم لقواعد البيانات المصممة حديثا

- **النسخ الاحتياطية (Backup):**

وهي عملية إنشاء نسخ احتياطية للبيانات الموجودة بهدف تأمين البيانات من الأعطال التي قد تؤدي لضياعها

- **تنظيم الملفات (File reorganization):**

هي عملية إعادة تنظيم الملفات علي أسطوانات التخزين بهدف تحسين أداء النظام

- **مراقبة الأداء (Performance monitoring):**

تستخدم لمراقبة وتسجيل أداء قواعد البيانات وبذلك تقدم لمدير قواعد البيانات (DBA) الإحصائيات اللازمة لتحليل أداء النظام ودراسة كيفية تحسينه (بعض النظم تقدم أيضا حلول لرفع الأداء)

أدوات تدعم عمل مستخدم قواعد البيانات

- **CASE tools (أدوات مساعدة هندسة النظم):**

تستخدم في مراحل تصميم قواعد البيانات ويوجد العديد من الأدوات التي تقوم بتنفيذ الكثير من المراحل التي يمر بها تصميم النظام

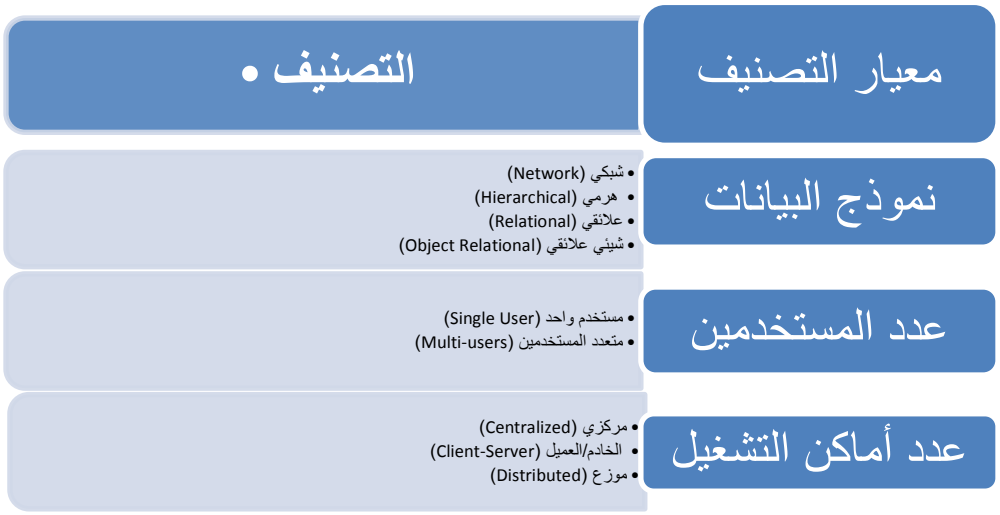
- **أدوات تطوير النظم:**

تستخدم عند تطوير نظم قواعد البيانات سواء أكانت لتصميم قواعد البيانات أو واجهات التعامل مع المستخدم أو تعديل وإنشاء الاستفسارات علي البيانات وكذلك أثناء إنشاء البرامج التطبيقية

- **برامج الاتصال عبر الشبكات:**

وتستخدم لتقديم إمكانية التعامل مع قواعد البيانات عبر الشبكا ت

تصنيف قواعد البيانات



المحاضرة الثالثة

قواعد البيانات العلاقية

Relational Databases

مبادئ قواعد البيانات العلاقية

نظراً لقوة الوحيد Relational Database Management System (RDMS)أصبحت هي النوع المستخدم حالياً، لما تقدمه من قوه و كفاءة و أدوات مساعدة للمبرمجين

تعتمد قاعدة البيانات العلاقية في تصميمها على المفاهيم الطبيعية الموجودة في بيانات نموذج العالم المصغر الذي تمثله قاعدة البيانات.

أساس قواعد البيانات العلاقية هو العلاقات الرابطة بين البيانات و التي تعتبر الجزء الاهم و الذي يمثل اغلب التعاملات مع قاعدة البيانات.

مبادئ قواعد البيانات العلاقية

مثال : نموذج قاعدة بيانات بسيطة (قاعدة بيانات لمستشفى)

Patient (مريض)

رقم المريض	الاسم	الجنس	رقم الغرفة	الطبيب
10	ناصر	1	100	احمد
20	نهى	2	200	سمير
30	عبدالله	1	100	احمد

Medicine (دواء)

رقم الدواء	اسم الدواء	المصنع
A1	بنادول	مصر
B2	اسبرين	الاردن
C2	السولين	السعودية

Treated_By (يعالج بواسطة)

رقم المريض	رقم الدواء	الكمية
10	A1	1
30	C2	3
20	A1	4

نموذج قاعدة البيانات (Database Model) هو نموذج يبين لنا صورة كاملة لنظام المعلومات و الوظائف و القيود الموجودة داخل قاعدة البيانات و يركز على التكامل بين البيانات

لتعريف قاعدة البيانات يجب تحديد تركيب السجلات التي يمكن تخزينها في كل ملف و تحديد الانواع المختلفة لعناصر البيانات.

كل سجل يحتوي على بيانات تمثل مثلا رقم المريض و اسم المريض و الطبيب و الجنس و رقم الغرفة (المثال السابق).

يجب أن نحدد نوع البيانات لكل عنصر بيانات داخل السجل مثل:-

اسم المريض سلسلة حروف

رقم المريض يكون رقم صحيح

يجب أن يكون هنالك رابط بين السجلات المختلفة حيث نجد معلومات عن مريض محدد مثلاً احمد في ملف "مريض" و معلومات عن الدواء و من يعالجه في ملفي "دواء" و "يعالج بواسطة".

الكثير من العلاقات في قواعد البيانات تربط انوع مختلفة من السجلات مع بعضها البعض و يطلق عليها العلاقات الرابطة " Relationships.

كذلك تتضمن معالجة قواعد البيانات الاستعلام و التعديل مثل:-

استخرج أسماء المرضى الذين يعالجهم د. عبدالله.

عدّل اسم المريض احمد إلى محمد.

مقدار تشاركية العلاقات – درجة العلاقة

ارتباط الجداول والعلاقات بعضها ببعض.

درجة العلاقة إما:

- واحد - واحد مسافر - تذكرة
- واحد – متعدد طالب – كتب مستعارة
- متعدد – واحد مؤلفون - كتاب
- متعدد – متعدد طلاب – نشاطات

مقدار تشاركية العلاقات – درجة العلاقة

واحد – واحد :

جدول المسافر

رقم السجل المدني	الاسم
1	احمد
2	منى
3	سعيد

جدول التذكرة

رقم التذكرة	من	الى	رقم المقعد	رقم السجل المدني
ت1	عمان	الدمام	20	1
ت2	عمان	الدمام	35	2
ت3	عمان	الدمام	15	3
ت4	عمان	الدمام	10	4

واحد – متعدد :

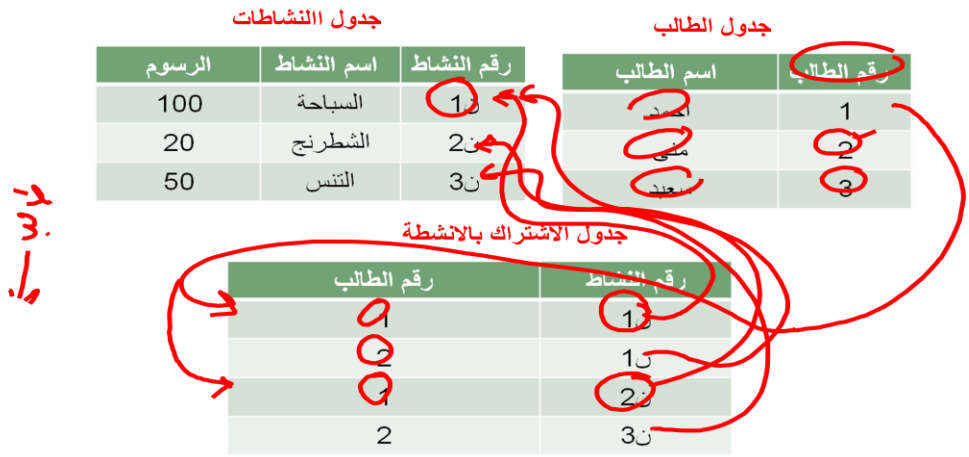
جدول الطالب

رقم الطالب	اسم الطالب
1	احمد
2	منى
3	سعيد

جدول الكتب في المكتبة

رقم الكتاب	عنوان الكتاب	تاريخ الاعارة	رقم الطالب
ب1	قواعد البيانات		1
ب2	الرياضيات		2
ب3	الحاسوب		1
ب4	التربية		3

متعدد – متعدد :



درجة الجدول

عدد الأعمدة التي يحتويها الجدول

جدول الطالب

رقم الطالب	اسم الطالب
1	احمد
2	منى
3	سعيد

درجة الجدول ٢

جدول االنشاطات

رقم النشاط	اسم النشاط	الرسوم
ن1	السباحة	100
ن2	الشطرنج	20
ن3	التنس	50

درجة الجدول ٣

مفاهيم أساسية في قواعد البيانات العلاقية

البياناتData:

هي أي حدوث للبيانات التي تصف أي كائن.

البيانات الوصفيةMetadata:

هي البيانات التي تصف البيانات المخزنة وصفاً دقيقاً و يطلق عليها **Data about**

الكينونةEntity:

هي وحدة معلومات تمثل فئة أو مجموعة من الأشياء أو الكائنات أو الأنشطة، هذه الوحدة لها مواصفات (خصائص) تصفها و تخصصها و تعبر عن مجموعة الكائنات التي تنتمي إليها، هذه المجموعة هي أمثلة أو حالات أو نماذج أو كائنات تتبع هذا الكيان.

و في اغلب الأحيان يكون اسم الكيان اسماً مفرداً.

أمثلة على الكيان من الأمثلة السابقة : مريض ، دواء ، يعالج بـ .

العلاقة الرابطة Relationship:

هي العلاقة التي تربط بين الكيانات و تمثل رابطة العالم المصغر الذي تمثله قاعدة البيانات.

تعبر العلاقات الرابطة عن الروابط بين البيانات في الواقع و تمثل في اغلب الأحوال بفعل مضارع أو فعلاً مبني للمجهول

أمثلة على العلاقات الرابطة

- الكيان طالب و الكيان مدرس و مقرر دراسي يوجد بينهم عدة علاقات رابطه منها

الطالب يدرس مقرر درسي

المدرس يُدرس المقرر الدراسي.

المدرس يُدرس الطالب المقرر الدراسي .

المدرس يرشد الطالب إلى المقرر المناسب.

الطالب يُرشد بواسطة المدرس .

الخاصية أو الحقلAttribute:

هي صفه تصف كيان معين و قيمتها هي احد مكونات سجلات البيانات مثل رقم الطالب و اسم الطالب في الجدول (العلاقة) طالب.

عنصر البيانات Data Item:

هو اقل وحدة بيانات مثل قيمه مخصصة مثلا رقم الطالب 1000 ، اسم الطالب احمد حيث احمد و الرقم 1000 هي عناصر بيانات.

عنصر بيانات مجمع Data aggregate :

هو عنصر بيانات يتكون من عناصر بيانات بسيطة اصغر مثال اسم الطالب (محمد احمد عبدالله) حيث أن الاسم هنا مجمع من ثلاث بيانات اصغر هي الاسم الأول و اسم الأب و العائلة.و يمكن تقسيمها إلى ثلاث حقول مختلفة تمثل جميعها الاسم الكامل للطالب.

سجلRecord :

هو تجميع لعناصر بيانات تمثل احد أمثلة أو حالات كيان محدد .مثل كل طالب له (اسم و رقم و تخصص)

وبالتالي مثال لسجل طالب:
(احمد ، 1000 ، حاسوب)
(عبدالله ، 2500 ، علوم)

كل قيمة من قيم السجل تمثل عناصر بيانات لخاصية من خواص الكيان.

Keyالمفتاح:

هو خاصية واحدة أو (عدة خصائص مجتمعه) من خصائص الكيان تستخدم لاختيار سجل أو أكثر من سجلات ذلك الكيان و يوجد منها ثلاث أنواع:-

Primary Keyالمفتاح الرئيسي

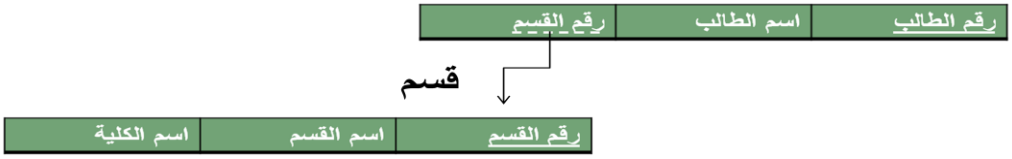
المفتاح الأساسي لكيان هو احد خصائص هذا الكيان و قيمته تكون وحيدة في كل سجل و لا تتكرر في أي سجل أخر من نفس الكيان ، و يجب كذلك أن تحتوي على قيمة و لا يجوز تركها فارغة مثل رقم الطالب في جدول طلاب ، حيث كل طالب يجب أن يكون له رقم مختلف عن زملائه ، و يوضع خط مستقيم أسفل الحقل للدلالة على انه مفتاح رئيسي.

رقم الطالب ▪	اسم الطالب ▪	التخصص ▪	الكلية ▪

المفتاح الأجنبيForeign Key :

هو عبارة عن خاصية عادية من ضمن خواص الكيان و موجودة كخاصية مفتاح أساسي في كيان آخر ، نميز هذا المفتاح بوضع خط متقطع اسفل اسم الخاصية.

طالب

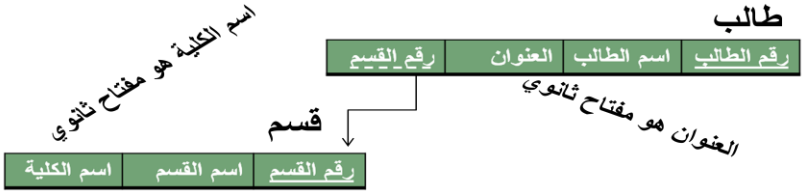


ليس بالضرورة أن يكون اسمي الحقلين متشابهين في الكيانين

المفتاح الخارجي يستخدم للربط بين الكيانات لاستخراج بيانات ذات صلة بين كيانين مختلفين، مثل استخراج اسم القسم و الكلية التي يدرس فيها الطالب.

Secondary Keyالمفتاح الثانوي

هو أي خاصية يمكن استخدامها لاختيار سجلات معينة من بين السجلات الموجودة في الكيان .



جدول المسافر

رقم السجل المدني	الاسم
١	احمد
٢	منى
٣	سعيد

جدول التذكرة

رقم التذكرة	من	الى	رقم المقعد	رقم السجل المدني
ت١	عمان	الدمام	٢٠	١
ت٢	عمان	الدمام	٣٥	٢
ت٣	عمان	الدمام	١٥	٣
ت٤	عمان	الدمام	١٠	٤

وصف قواعد البيانات يسمى "مخطط قواعد البيانات" (Schema)

يستخدم المخطط عند تصميم قواعد البيانات

هذا المخطط لا يتوقع تغيره بشكل تكراري

يتم عادة تمثيل هذا المخطط باستخدام شكل أو رسم هندسي

يوضح هذا المخطط بعض الأشياء مثل أسماء السجلات وأسماء الحقول وقد لا تظهر فيه نوع البيانات المستخدمة أو العلاقات بين البيانات

يسمى هذا المخطط Intension

- "هذا المخطط يتم تخزين وصفه داخل قواعد البيانات وهذا ما يعرف باسم meta-data"

مخطط لبيانات جامعة(Schema)

الطالب

الاسم	رقم الطالب	الفصل	التخصص

المقرر

رقم المقرر	اسم المقرر	عدد الساعات	القسم

المتطلب

رقم المقرر	رقم المتطلب السابق

الشعبة

رقم الشعبة	رقم المقرر	الفصل	السنة	المحاضر

كشف-الدرجات

رقم الطالب	رقم الشعبة	الدرجة

المحاضرة الخامسة

مبادئ قواعد البيانات العلاقية

Fundamentals of Relational Databases

أنواع العلاقات الرابطة بين الكيانات (السجلات)

•العلاقات الرابطة بين الكيانات هي أهم ما يميز قاعدة البيانات العلاقية ، حيث تتوقف قاعدة البيانات التي نصممها و ننفذها بشكل كبير على أنواع العلاقات الرابطة

•تقسم إلى ثلاث أنواع:-

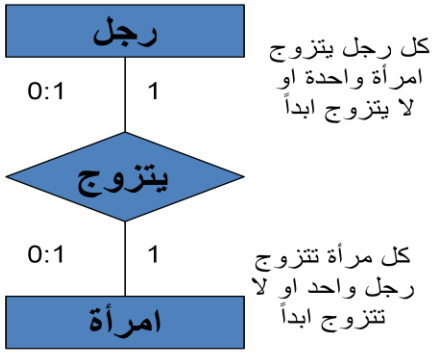
•علاقة سجل واحد مع سجل واحد

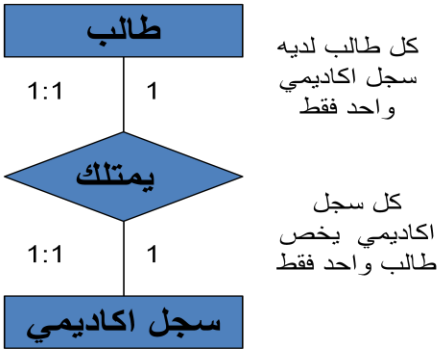
•علاقة سجل واحد مع عدة سجلات

•علاقة عدة سجلات مع عدة سجلات

- **علاقة سجل واحد مع سجل واحد(One to one)**:

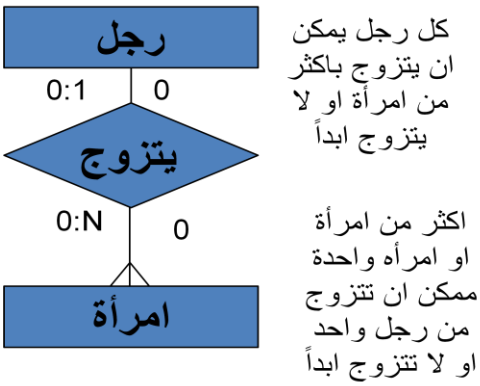
تعني هذه العلاقة أن أي سجل يرتبط مع سجل واحد على الأكثر من الكيان الثاني

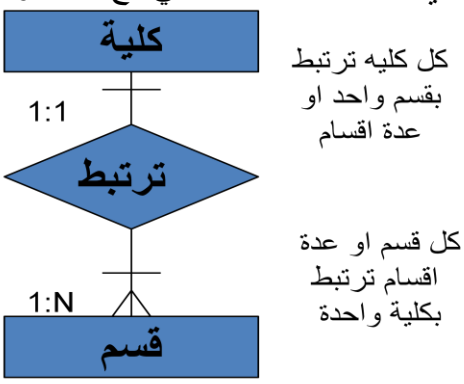




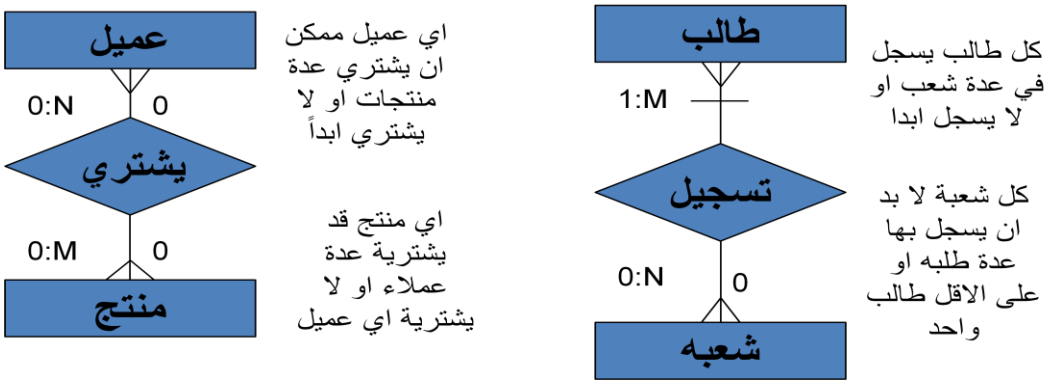
- **علاقة سجل واحد مع عدة سجلات(one to many)**:

تعني هذه العلاقة أن أي سجل من الكيان الأول يمكن أن يرتبط مع عدة سجلات من الكيان الثاني ، و يرتبط أي سجل من الكيان الثاني مع سجل واحد على الأكثر من الكيان الأول





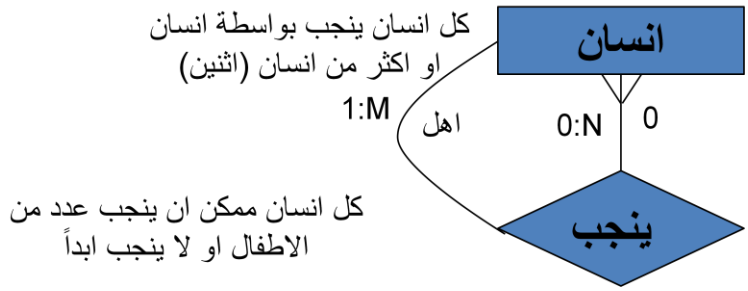
- **علاقة عدة سجلات مع عدة سجلات (Many to many)**: أي سجل من الكيان الأول يرتبط مع عدة سجلات من الكيان الثاني و كذلك يرتبط أي سجل من الكيان الثاني مع عدة سجلات من الكيان الأول.



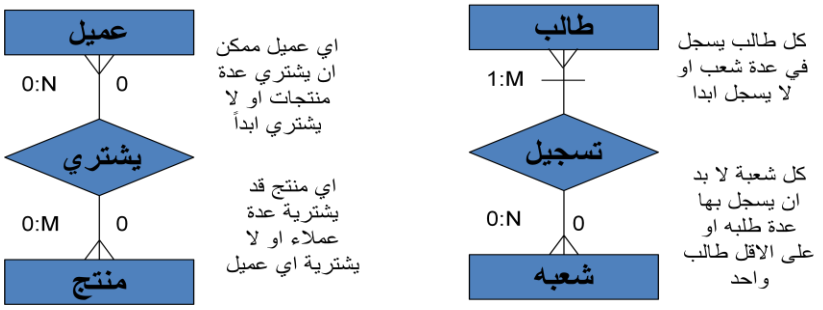
درجة العلاقة الرابطة بين السجلات (الكيانات)
degree of association between Entities

- **تحدد درجة العلاقة الرابطة عدد الكيانات المشاركة في العلاقة الرابطة او بمعنى اخر عدد الكيانات التي تربط بينهما العلاقة.**

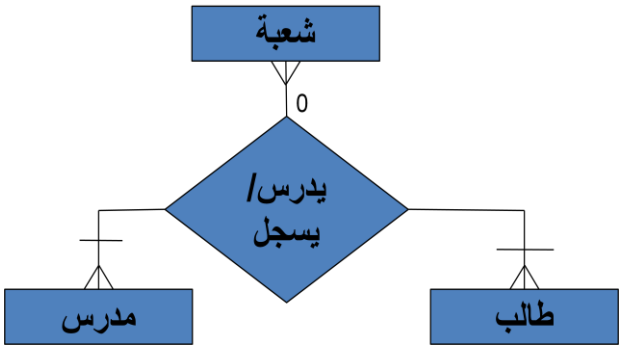
✓ **علاقة أحادية (Unary Relationship):** تكون درجة العلاقة الرابطة احادية اذا كانت العلاقة الرابطة بين الكيان و نفسه ، اي تربط سجلات من نفس الكيان



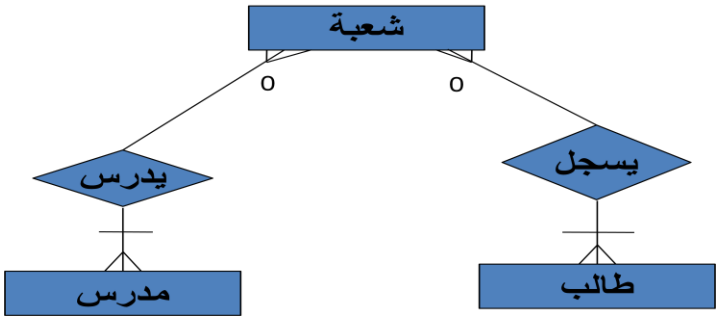
✓ **علاقة ثنائية (Binary Relationship):** تكون درجة العلاقة الرابطة ثنائية اذا كانت العلاقة الرابطة بين كيانين مختلفين.



✓ **علاقة ثلاثية (Ternary Relationship):** تكون درجة العلاقة الرابطة ثلاثية اذا كانت العلاقة الرابطة بين ثلاث كيانات مختلفة. في اغلب قواعد البيانات تحول الرابطة الثلاثية الى ثنائية.



✓ **علاقة ثلاثية (Ternary Relationship) و التحويل الى علاقة ثنائية**



النموذج العلاقي (Relational Model)

لبناء قواعد بيانات يفضل مبرمجو و مصممو قواعد البيانات استخدام نموذج البيانات العلاقي. هذا النموذج مبني على العلاقات بين الكيانات و البيانات.

•اغلب البرامج و النظم المستخدمة في مجال قواعد البيانات صممت لبناء هذا النوع.

•مميزات النموذج العلاقي:

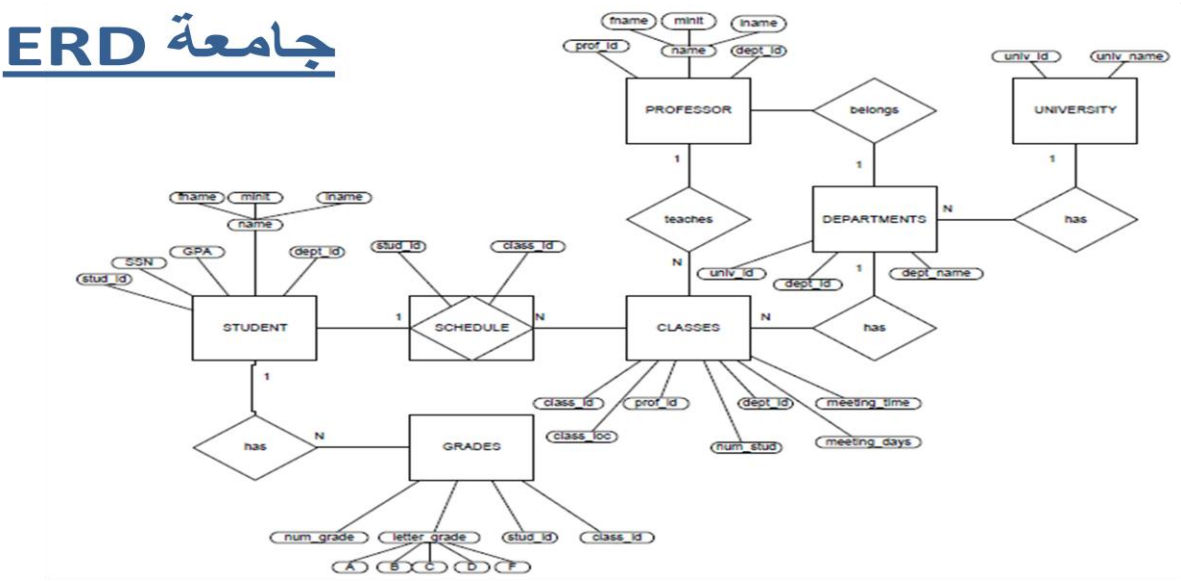
يتميز النموذج العلاقي عن غيره من نظم قواعد البيانات للأسباب التالية:

•له الأدوات و الخصائص التي تميزه عن غيره و الخاصة به.
•يمثل منطقياً كافة الكيانات و العلاقات و خصائصها.
•يعد تطويراً و امتداداً لشكل ملف البيانات التقليدي.
•يستخدم المفاتيح الأساسية و الأجنبية للربط بين الكيانات.
•اعتماداً عليه بنيت خصائص قواعد البيانات التي تجعلها لا تتأثر بمشاكل الصيانة.
•يمكن تطبيق كافة العمليات الحسابية و المنطقية على مكوناته.

نموذج الكيان و العلاقة الرابطة (Entity Relationship Model)

•احد أشهر و أهم طرق تمثيل و تصميم قواعد البيانات هو نموذج الكيان و العلاقة الرابطة (Entity-Relationship Data Model (ERD)).•

- ✓ **نموذج (ERD) هو النموذج الذي يتم استخدامه لانشاء قواعد البيانات على الحاسب الآلي و له قواعد و اشكال محددة تصف الكيانات الموجودة في تطبيق معين و العلاقات الرابطة بين تلك الكيانات و خصائصها و كذلك القيود المفروضة على كل منها.**
- ✓ **يمثل تصميم قاعدة البيانات.**
- ✓ **جميع الاشكال التي تم دراستها هي نماذج مبسطة من نماذج (ERD)**



أسلوب تصميم نموذج الكيان والعلاقة الرابطة

يتم تصميم قواعد البيانات باستخدام إحدى طريقتين:
1) استخدام الرسم البياني للكيان والعلاقة الرابطة (Entity Relationship diagram)
2)تطبيع قواعد البيانات (Database Normalization)

•يتكون تصميم ERD من مجموعة من الكيانات (Entity) تربط بعضها ببعض علاقات رابطة (Relationship).
•السجلات التي تتبع الكيانات عبارة عن بيانات شبه ثابتة ،ونادرا ما تحتاج الى التعديل(Static Data)
•يتم تحديد خصائص كل كيان
- الخاصية التي تميز كل سجل يتبع الكيان ولا تتكرر هي خاصية المفتاح الرئيسي(Primary Key)

•السجلات التي تصف العلاقات الرابطة فهي عبارة عن بيانات تتجدد وتتغير وتضاف وتحذف بشكل متواصل(Dynamic Data).
•يجب تحديد لكل علاقة رابطة الخصائص التي تساعد على وصف العلاقة الرابطة بين كل كيانين
•يجب تحديد نوع العلاقة:
•واحد إلى واحد(One-to-One)
•واحد إلى كثير(One-to-Many)
•كثير إلى كثير(Many-to-Many)

•يجب تحديد نسبة المشاركة (0أو1)

تطبيق قاعدة بيانات الكلية المصغر

في قاعدة بيانات الكلية نهتم بـ:

•تخزين بيانات الطلبة(الرقم الجامعي، الاسم، العنوان، التخصص)
•بيانات المقررات التي يسجلها الطالب(رمز المقرر،اسم المقرر، عدد الساعات المعتمدة،العام الدراسي، الفصل الدراسي، العلامة)
•بيانات أعضاء هيئة التدريس(الرقم،الاسم،الهاتف، القسم ، المقررات)

•التصميم:

1)الكيانات :هي وحدة معلومات لها خصائص تصفها تخصها وأنها تكون أسماء.وقد تم تحديد الكيانات التالية:

•**الطالب** ،وخصائصه هي: (رقم الطالب، الاسم، العنوان، التخصص)
•**المدرس**،وخصائصه هي: (رقم المدرس، الاسم،الهاتف،القسم)
•**المقرر**،وخصائصه هي: (رمز المقرر، اسم المقرر ،عدد الساعات المعتمدة)

- ملاحظة(1):المعلومات المذكورة هي معلومات شبه ثابتة (Static)

ـملاحظة(2):لم يتم ذكر معلومات عن المواد التي يدرسها الطالب في جدول الطالب، ولم يتم ذكر معلومات عن المواد التي يدرسها المدرس في جدول المدرس ، ولم يتم ذكر معلومات الفصل الدراسي في جدول المقرر، فهي كلها معلومات متغيرة (Dynamic)لا تذكر في الكيانات.

ـملاحظة (3): تم تحديد الصفة المميزة لكل كيان بوضع خط تحتها.

2) العلاقات الرابطة:هي عبارة عن فعل يمثل العلاقة بين كيان ونفسه، أو كيانين، أو ثلاثة كيانات معا. وقد تم تحديد العلاقات التالية:

•**يُسجِّل :**

ـهي علاقة تربط الطالب بالمقررات التي يسجلها للدراسة
ـلها الخصائص(الفصل الدراسي، العام الدراسي، العلامة)

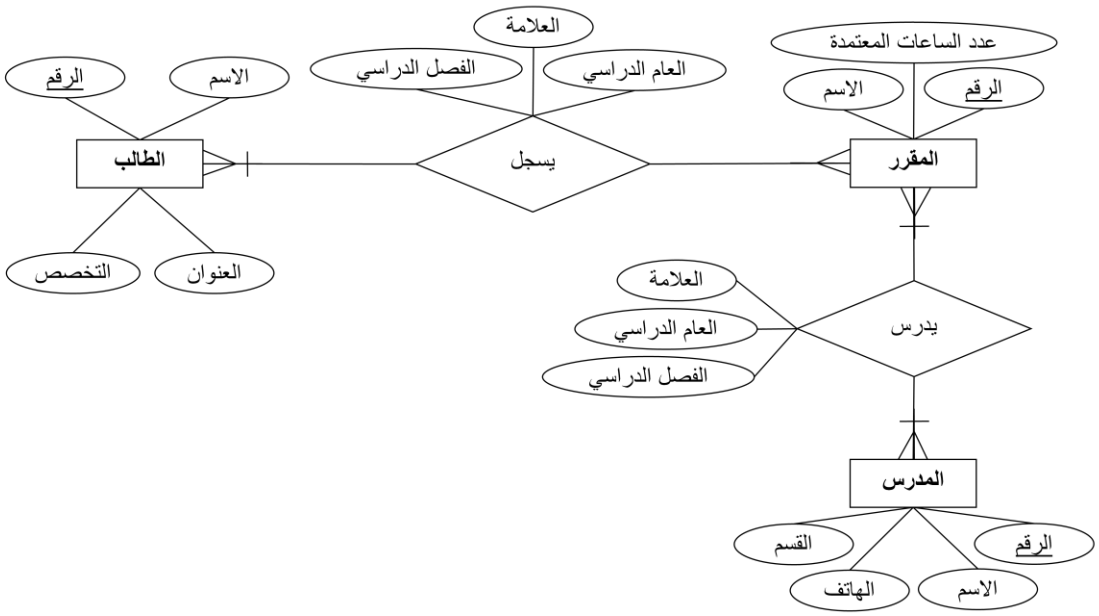
- كل طالب يمكنه تسجيل عدة مقررات،وكل مقرر يمكن ان يسجله اكثر من طالب، أي ان نوع العلاقة كثير إلى كثير(M:N(Many –to – Many

•**يُدرِّس :**

ـهي علاقة تربط عضو هيئة التدريس بالمقرر الدراسي
ـلها الخصائص(الفصل الدراسي، العام الدراسي)

- كل مدرس يمكنه تدريس عدة مقررات،وكل مقرر يمكن ان يدرسه اكثر من مدرس ، أي ان نوع العلاقة كثير إلى كثير(M:N(Many –to – Many

- التصميم:٣)نموذج الكيان العلاقة:



تطبيق قاعدة بيانات المستشفى المصغر

1)الكيانات:

- **المريض** (الرقم، الاسم)
- **الطبيب** (الرقم ، الاسم، الهاتف، التخصص)
- **الدواء** (الرقم ، الاسم)
- **الغرفة** (الرقم ، الهاتف، عدد الأسرة)

تطبيق قاعدة بيانات المستشفى المصغر

2) العلاقات الرابطة :

- يعالج :

ـ علاقة رابطة بين الطبيب والمريض

ـ المريض يتبع دكتور واحد،أما الدكتور فيتبعه عدة مرضى ويمكن ألا يتبعه أي مريض ،وبالتالي تكون الكاردينالياتي (1:1)من جهة الطبيب،و تكون الكاردينالياتي (0:N)من جهة المريض، وعليه يكون نوع العلاقة واحد إلى كثير(1:N)

- ينام في :

ـ علاقة رابطة بين المريض والغرفة

ـ كل غرفة يقيم فيها أكثر من مريض، أو لا يقيم فيها احد، لذلك فالكاردينالياتي من جهة المريض هي (0:N)

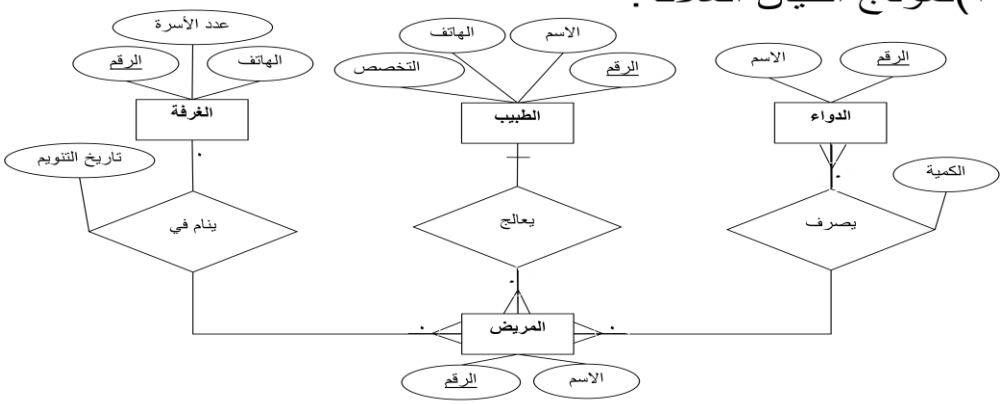
ـ كل مريض يمكن أن ينام في غرفة واحدة، أو ليتم تنويمه في حالة العيادة الخارجية، لذلك فالكاردينالياتي من جهة الغرفة هي (0:1)

- نوع العلاقة واحد إلى كثير (1:N)
ـلها علاقة (تاريخ التنويم)

•يصرف :
ـعلاقة رابطة بين المريض والدواء
ـالمريض يمكن ان يصرف اكثر من دواء او لا يصرف دواءا ، وبالتالي الكارديناليتي تكون (0:N)من ناحية الدواء
ـكل دواء يمكن صرفه من اكثر من مريض، ويمكن ان لايصرفه أي مريض ، وبالتالي الكارديناليتي تكون (0:N)من ناحية المريض
- وعليه يكون نوع العلاقة كثير إلى كثير(N:M)
ـهذه العلاقة لها الخاصية (الكمية)

3)نموذج الكيان العلاقة:

٣)نموذج الكيان العلاقة:

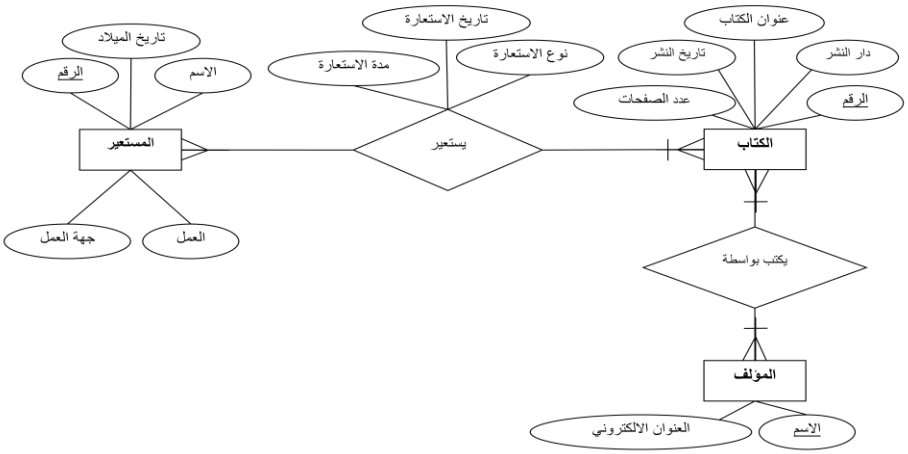


تطبيق الاستعارة الإلكترونية
•الكيانات:
•**المستعير** (الرقم، الاسم، العمل، جهة العمل، تاريخ الميلاد)
•**الكتاب** (الرقم، عنوان الكتاب، عدد الصفحات، دار النشر، سنة النشر)
- **المؤلف** (اسم المؤلف،العنوان الالكتروني)

2) العلاقات الرابطة:
•يستعير:
ـعلاقة رابطة بين المستعير والكتاب
ـلها الخصائص (نوع الاستعارة، مدة الاستعارة، تاريخ الاستعارة)
ـيمكن لأي مستعير أن يستعير عدة كتب، وعلى الاقل كتاب واحد،لذلك الكارديناليتي (1:N)من ناحية الكتاب
ـيمكن ان يستعير الكتاب عدة مستعيرين،أولا احد ، لذلك تكون الكاردينالتي (0:N)من ناحية المستعير.
ـهذه العلاقة من النوع كثير إلى كثير(N:M)
•كُتِبَ بواسطة:

ـعلاقة رابطة بين المؤلف والكتاب
ـيمكن لأي مؤلف أن يكتب عدة كتب، وعلى الأقل كتاب واحد،لذلك الكارديناليتي (1:N)من ناحية الكتاب
ـيمكن أن يشترك في تأليف أي كتاب عدة مؤلفين،أو على الأقل مؤلف واحد ، لذلك تكون الكارديناليتي (1:N)من ناحية المؤلف.
ـهذه العلاقة من النوع كثير إلى كثير (N:M)

٣) نموذج الكيان العلاقة:



المحاضرة السادسة

تصميم نموذج الكيان والعلاقة الرابطة - تابع

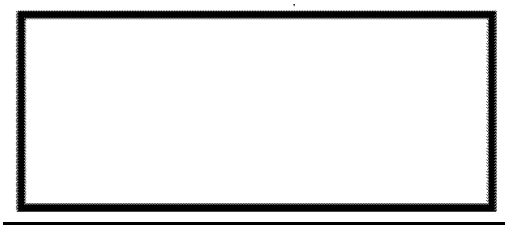
Design of Entity-Relationship Diagram (ERD)

ما هو مخطط الكيان العلاقة؟

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

مكونات مخطط الكيان العلاقة

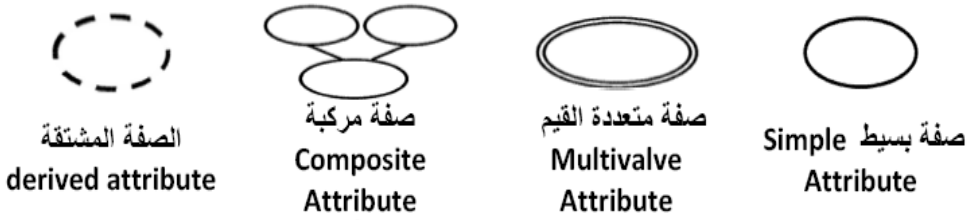
- **الكيان أو الكينونة (Entity):** هو الوحدة الأساسية التي يتم تمثيلها بنموذج الكينونة/العلاقة ويشير هذا الكيان إلي "شيء" حقيقي في الحياة سواء كان له وجود فعلي مثل (طالب – موظف – سيارة - ... الخ) أو وجود منطقي مثل (شركة – وظيفة – مقرر -... الخ).
- ويتم تمثيل الكيان باستخدام شكل مستطيل يكتب داخله اسم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



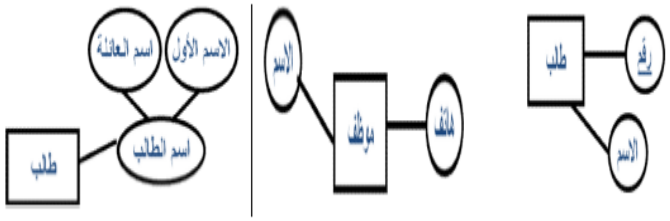
- **الصفة (Attribute):** هي صفة معينة تصف الكيان وتكون تابعة له في المخطط ، مثل اسم الموظف ، عمر الطالب، مرتب موظف، درجة طالب، عدد الساعات الدراسية لمقرر، لاحظ أن كل صفة تتبع كيانا معينا.
- الصفة يمكن أن تكون صفة بسيطة – أو صفة مركبة
- الصفة يمكن أن تكون أحادية القيمة – أو صفة متعددة القيم
- يوضع خط 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

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



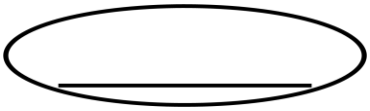
شكل تمثيل الصفات المختلفة للكيان



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

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

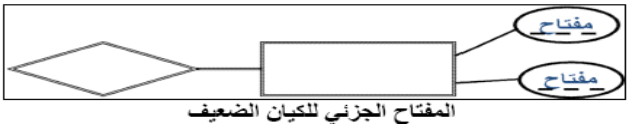
صفة المفتاح الرئيسي(Primary Key Attribute) : هي تلك الصفة المميزة للكيان ، ويتم تمثيلها بشكل بيضاوي مع خط تحت إسم الصفة، وقد يكون للكيان أكثر من صفة لتمثل معا المفتاح الرئيسي
مكونات مخطط الكيان العلاقة



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

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 تقويه، ويتم تمثيل الكيان الضعيف بمستطيل مزدوج

•يرتبط المفتاح الجزئي بالكيان الضعيف، ليساعد فيما بعد بتكوين مفتاح رئيسي للكيان الضعيف.

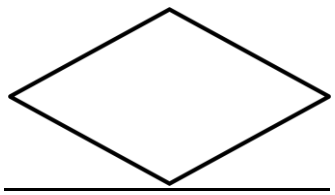


المفتاح الجزئي للكيان الضعيف

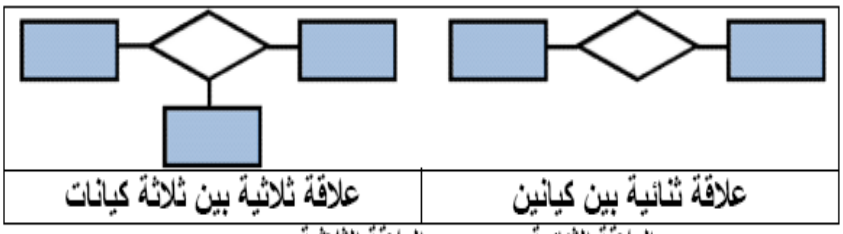
العلاقات(Relation): العلاقة (R) بين مجموعة من الكيانات (E1,E2, ...En) هي مجموعة تمثل الارتباطات بين هذه الكيانات، كل وحدة في العلاقة (R) هي عبارة عن اتحاد أو ارتباط بين الكيانات المرتبطة بهذه العلاقة، بحيث أن هذه الوحدة تمثل بصف واحد من كل كيان مشارك في العلاقة.

•في نموذج الكيان/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

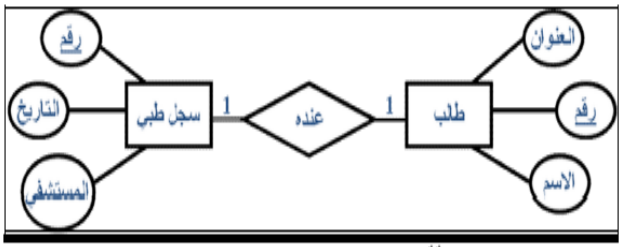


- **درجة العلاقة:** لكل علاقة درجة، وتتحدد هذه الدرجة بعدد الكيانات المرتبطة بهذه العلاقة (ثنائية – ثلاثية - ....)



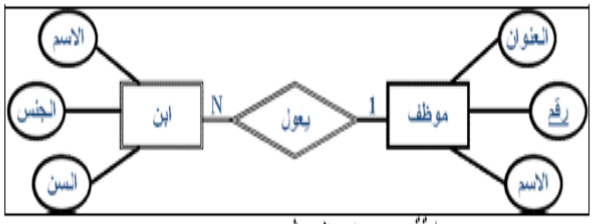
العلاقة الثنائية Binary والعلاقة الثلاثية Ternary

- **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:
  - **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
  - **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
  - **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



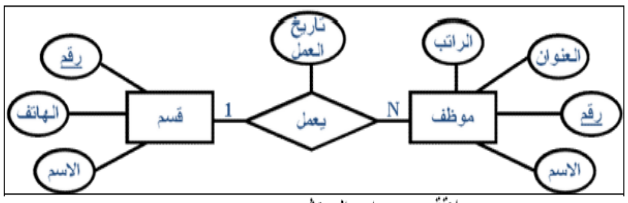
علاقة ١:١ واحد-الى-واحد (one-to-one)

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



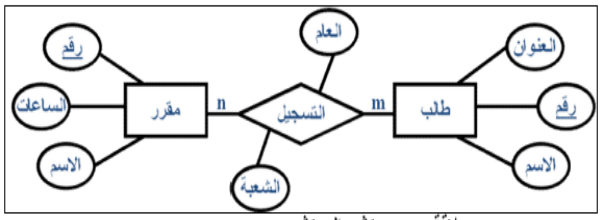
علاقة 1:N واحد-الى-كثير (one-to-many)

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



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.



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

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

•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.

1.الاشتراك الكلي(Total participation) :

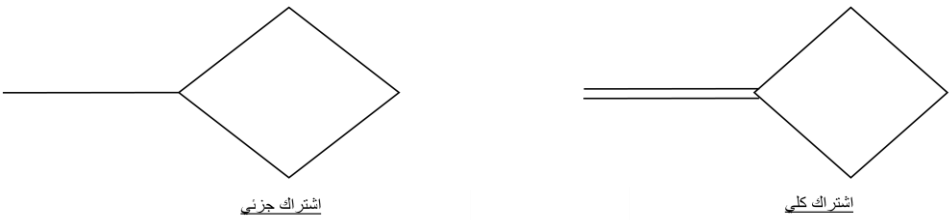
نقول أن العلاقة علاقة الاشتراك الكلي ، إذا كان كل وحدة في الكيان ألأول يجب أن ترتبط بوحدة من الكيان الآخر ضمن العلاقة، يسمي هذا القيد بقيد "ارتباط الوجود" ، أي أن وجود وحدة من كيان ما

يستلزم ارتباطها بوحدة من كيان آخر،ومثال ذلك كيان (سجل الاكاديمي) 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.

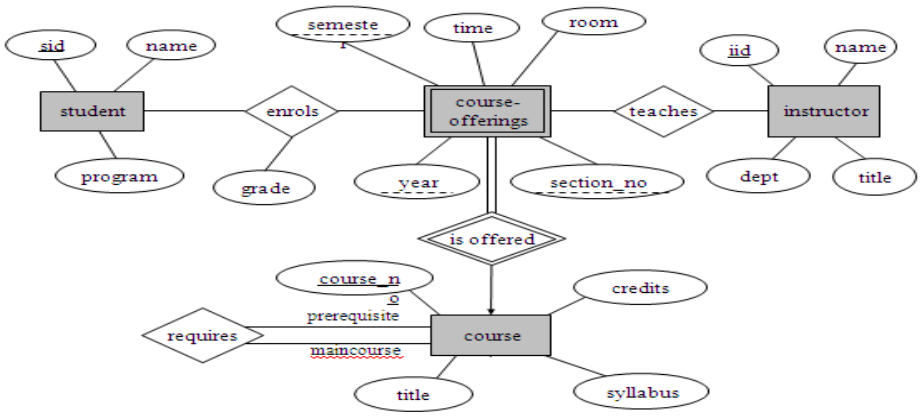
- ويتم تمثيل قيد الاشتراك الكلي، برسم خط مزدوج، يربط الكيانات المرتبطة بهذه العلاقة، من جهة الكيان المعتمد على الاشتراك الكلي

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.

- **أنواع القيود على العلاقاتRelationship Constraints:**

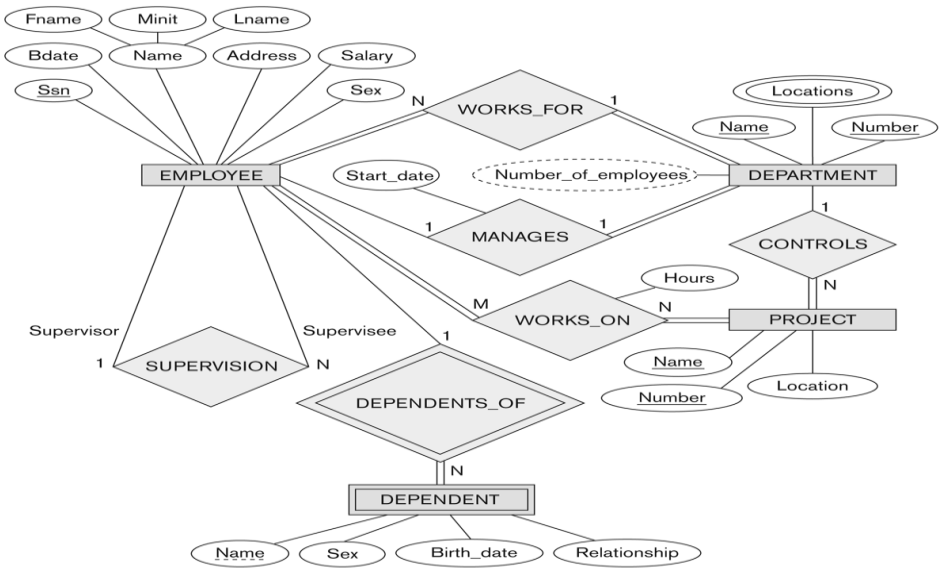


مثال1



E-R diagram for a university.

مثال2

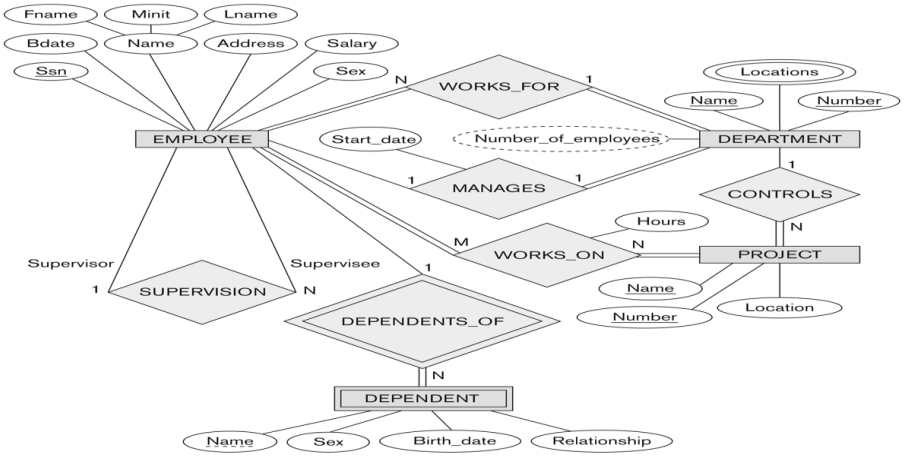


المحاضرة السابعة – الجزء الثاني

تصميم قواعد البيانات العلاقية

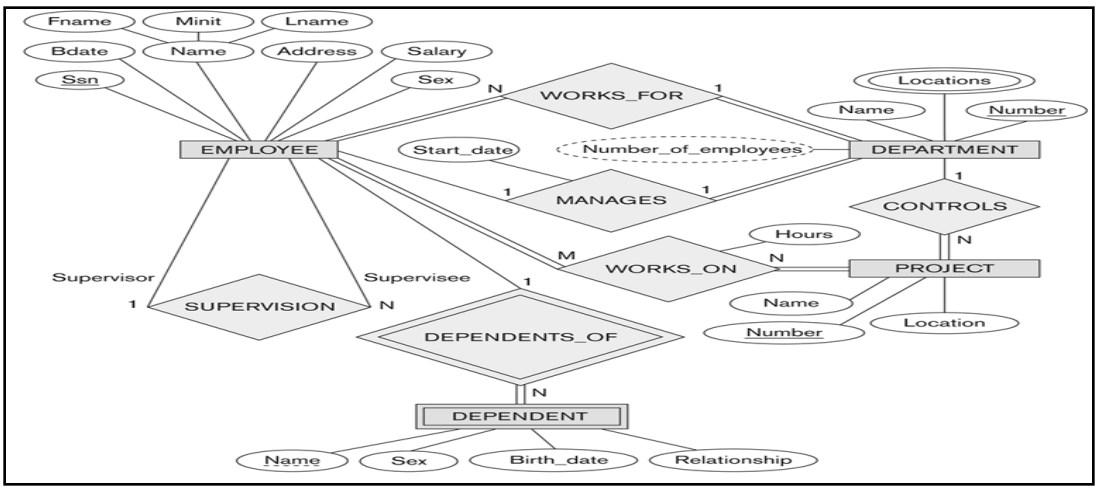
Design of Relational Database

مثال2



تحويل مخطط الكيان العلاقة لشركة إلى ما يقابله من مخطط قواعد البيانات (جداول)

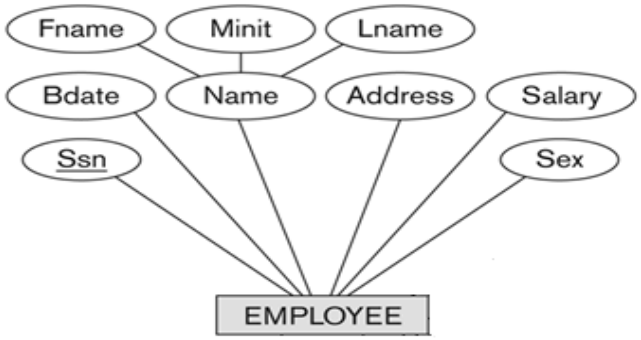
•نبدأ بتحويل الكيان العادي:



- يحتوي مخطط الكيان العلائقي السابق على ثلاث كيانات عادية هي:

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

1.أ) كيان الموظف (Employee)

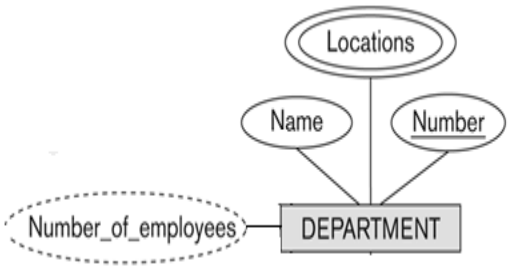


- نلاحظ في هذا الكيان وجود صفة مركبة هي صفة الإسم(Name)والتي تتكون من الصفات الجزئية الاسم الأول (Fname) ، و حرف الاسم الأوسط (Minit) ، والاسم الأخير (Fname). وكما اوضحنا سابقا ، فإن الصفة المركبة في عملية التحويل تدرج بصفاتها الجزئية فقط

- وينتج عن عملية التحويل الجدول التالي:

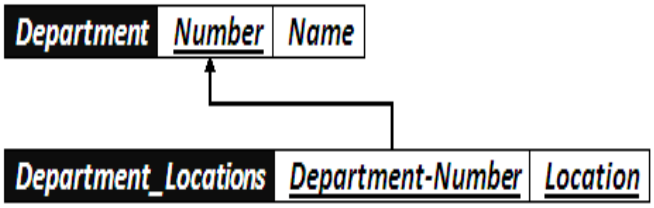
EMPLOYEE	Ssn	Fname	Minit	Lname	Bdate	Address	Salary	Sex

1.ب) كيان القسم (Department)

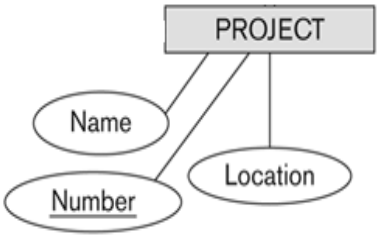


- نلاحظ في هذا الكيان وجود صفة مشتقة هي عدد الموظفين (Number of employees)، وذكرنا سابقا أننا في عملية التحويل نتجاه هذه الصفة ،ليتم بناؤها لاحقا بجملة إستعلام
- كما نلاحظ وجود صفة متعددة القيمة وهي المواقع (locations)وذكرنا سابقا أن الصفة متعددة القيمة يتم تحويلها إلى جدول مستقل مع المفتاح الرئيسي للكيان ،ويكون إسم الجدول مكون من إسم الكيان مضافا إليه إسم الصفة متعددة القيمة.

- وينتج عن عملية التحويل الجدولين التاليين:



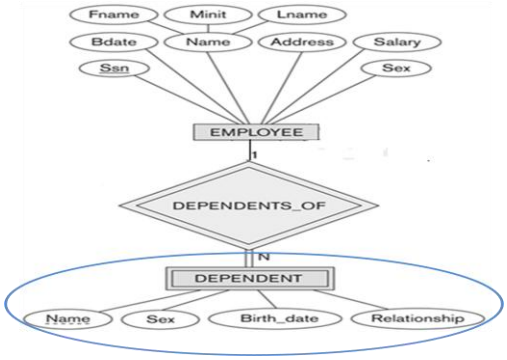
1.ج) كيان المشروع:



- 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:

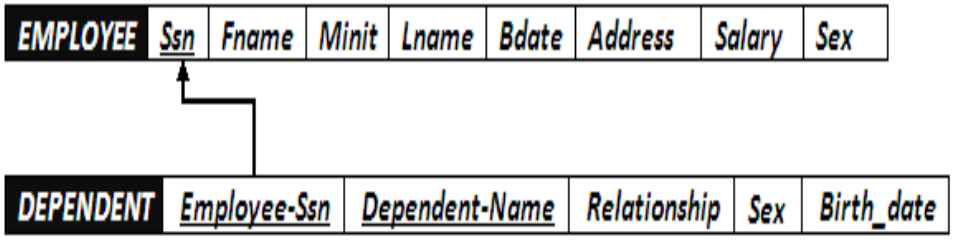
- كيان المعتمد على (Dependent)، حيث يصف هذا الكيان أفراد عائلة الموظف المعتمدين عليه في إعالتهم.

- ويحتوي هذا الكيان على صفة المفتاح الجزئي الاسم (Name)،التالية:
- و يرتبط 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). ويكون المفتاح الرئيسي للجدول هو مفتاحا للكيان القوي(Employee-Ssn) بالإضافة إلى المفتاح الجزئي في الكيان الضعيف (Dependent-Name).

- وينتج عن عملية التحويل الجدول التالي:

DEPENDENT	Employee-Ssn	Dependent-Name	Relationship	Sex	Birthdate

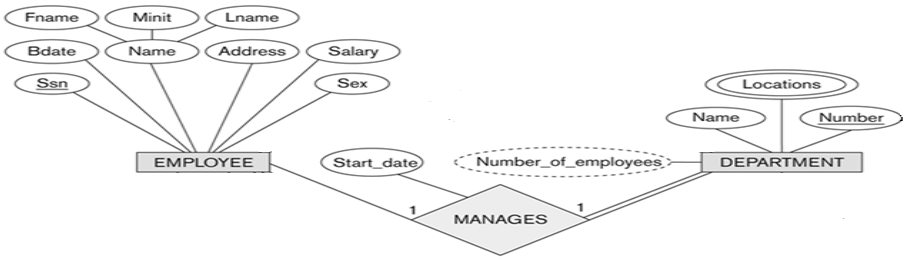
- وبذلك يرتبط جدول (Dependent) بجدول (Employee) بوجود المفتاح الأجنبي رقم الموظف (Employee-Ssn).



2.تحويل العلاقات:

3.أ) تحويل علاقة واحد إلى واحد:

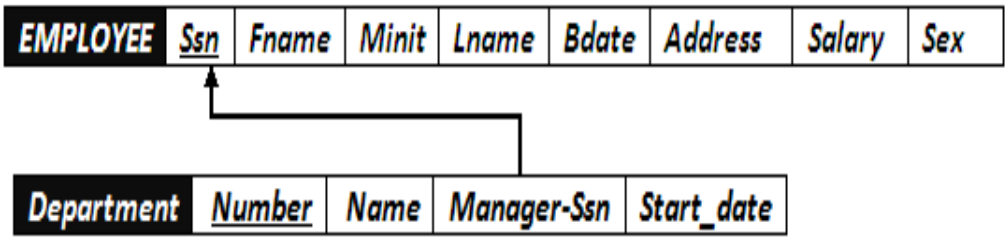
في هذا المثال توجد علاقة من النوع واحد- إلى- واحد ،وهي علاقة (Manages)بين كيان الموظف،وكيان القسم، والعلاقة ذات اشتراك كلي من جهة القسم ، مع وجود الصفة (start_date) على العلاقة.



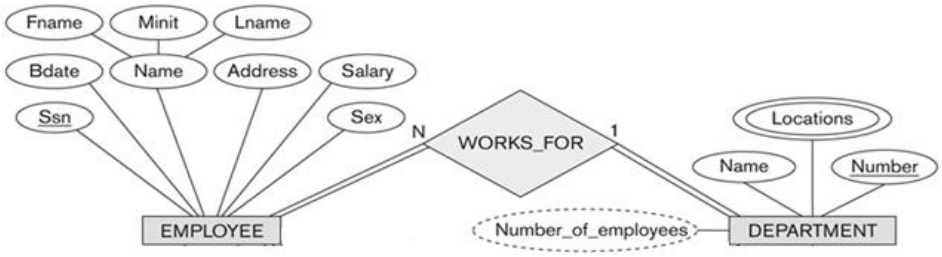
- وبناءا علي النقطة السابقة ، يتم أخذ نسخة من المفتاح الرئيسي لجدول الموظف (Employee_Ssn)، مضافا إليها الصفة على العلاقة (start_date)، ووضعهما كتعديل على جدول القسم(department)، ويفضل إعادة تسمية المفتاح الرئيسي ليدل على العلاقة وهي علاقة ادارة ، فهو مدير (Manager)، وبالتالي بدلا من التسمية Employee_Ssnتصبح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



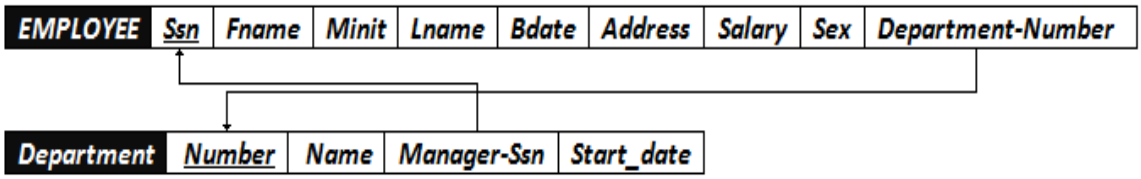
○ في هذا المثال لدينا ثلاث علاقات من نوع واحد إلى كثير وهي:

1.علاقة موظف يعمل في قسم(Works_for)

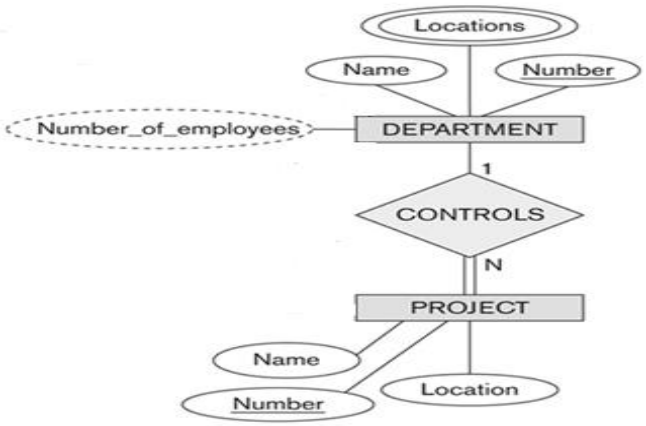
•نتيجة لهذه العلاقة يتم التعديل على جدول الموظف(Employee) ليحتوي على رقم القسم(Department_Number) الذي يعمل فيه، ليصبح جدول الموظف (Employee) بالشكل التالي:

EMPLOYEE	Ssn	Fname	Minit	Lname	Bdate	Address	Salary	Sex	Department-Number

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



2)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:

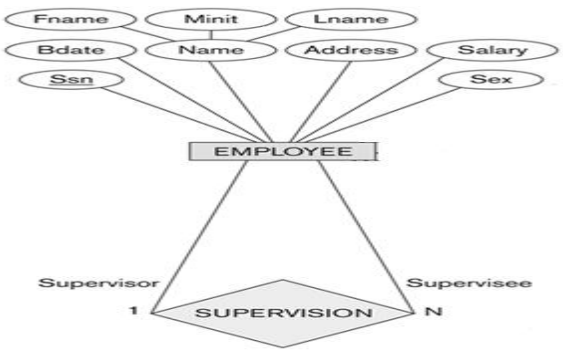
Project	Number	Name	Location	Department_Number

•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

3)علاقة إشراف موظف على موظف (Supervision):

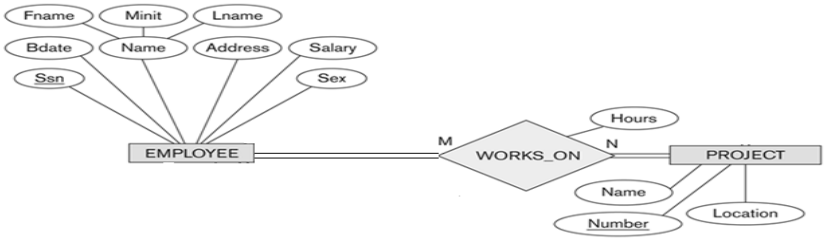


•هذه العلاقة هي علاقة أحادية ذات تغذية راجعة،أي علاقة كيان على نفسه.
•نتيجة هذه العلاقة أن يتم إضافة حقل جديد هو حقل المشرف (Supervisor)،وهو حقل يعبر عن رقم الموظف، مع إعطائه خصوصية كونه مشرفا على غيره من الموظفين.الحقل يتم إضافته إلى جدول الموظف (Employee) ليصبح بالشكل التالي:

EMPLOYEE	Ssn	Fname	Minit	Lname	Bdate	Address	Salary	Sex	Department-Number	Supervisor

3.ج) تحويل علاقة كثير إلى كثير:

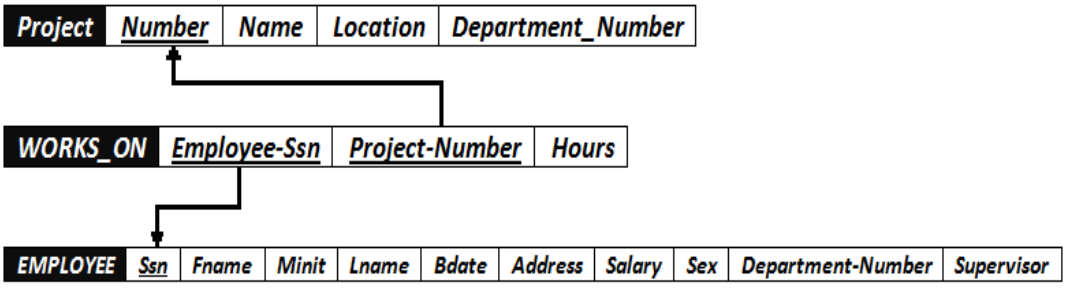
○ في هذا المثال، يوجد لدينا علاقة واحدة من نوع كثير إلى كثير، هي علاقة موظف يعمل على مشروع (Works_on)



○وفي هذه الحالة يتم إنشاء جدول جديد تحت إسم 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 طرق 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: