مقدمة لقواعد البيانات

المحاضرة الأولى

قواعد البيانات ومستخدميها

مقدمة لقواعد البيانات

قواعد البيانات هي مجموعة من البيانات المرتبطة.

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 بها من أجل غرض معين

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)

نظم إدارة قواعد البيانات

(Database Management System DBMS)

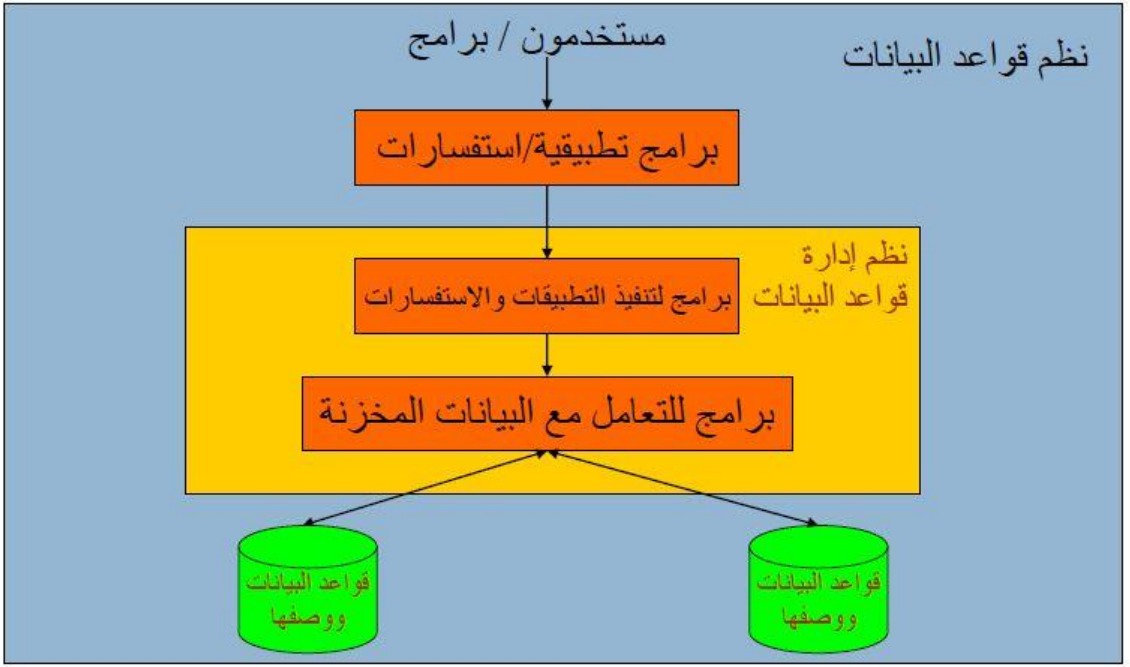
نظم إدارة قواعد البيانات:

- هي مجموعة من البرامج التي تمكن المستخدم من إنشاء ومعالجة قواعد البيانات
- يتم إنشاء هذه النظم لتساعد المستخدم في تعريف وبناء ومعالجة قواعد البيانات:
  - **تعريف البيانات**: المقصود منه هو تعريف البيان ونوعه وبنائه ووصف القيود المفروضة عليه
  - **بناء البيانات**: هو تحديد طريقة تخزينها وكيفية البناء وتنفيذه
  - **معالجة البيانات**: هو تعديل البيانات و إنشاء الاستفسارات و إنشاء التقارير

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))



مخطط يوضح قواعد البيانات ونظم إدارتها

خواص تفرق بين قواعد البيانات ونظم الملفات التقليدية

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

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

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

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

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

- **المشاركة فى 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

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

مستخدم قواعد البيانات

- **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)

أشخاص يتعاملون مع قواعد البيانات بطريقة غير مباشرة

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

- **مصمموا ومنفذوا نظم إدارة قواعد البيانات:**

  هم الذين يقومون بتصميم وتنفيذ نظم إدارة قواعد البيانات نفسها

- **مطوروا 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

مميزات استخدام قواعد البيانات

- التحكم فى تكرار البيانات
- فرض القيود على المستخدمين الذين ليس لهم صلاحيات معينة
- توفير بيئة تخزين لاتفقد البيانات
- السماح باستنباط معلومات من البيانات المتواجدة
- توفير واجهات متعددة لتعامل المستخدم مع البيانات
- تمثيل العلاقات المعقدة بين البيانات بسهولة
- فرض قيود التكامل بين البيانات

مميزات استخدام قواعد البيانات

- 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 فى استخدام وتعديل البيانات
- توفير بيانات على درجة عالية من التحديث
- اقتصادية الاستخدام

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 ضرورى
- اذا كان العمل لا يحتاج الى بيئة ذات عدة مستخدمين

مقدمة لقواعد البيانات

المحاضرة الثانية

مقدمة لنظم إدارة قواعد البيانات

الجزء الأول

هيكلة نظم إدارة قواعد البيانات (DBMS Architecture)

١. **النظام المركزي (Centralized system):**

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

٢. **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)

- نموذج البيانات هو مجموعة من الأفكار (Concepts) والتي تستخدم لوصف بناء البيانات
- بناء البيانات هو تحديد نوع البيانات و العلاقات بين البيانات والقيود المفروضة عليها
- يمكن أن يحتوي نموذج البيانات علي 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

تصنيفات نماذج البيانات

يوجد العديد من نماذج البيانات التي تم اقتراحها ويمكن أن نصنفها حسب نوع بناء البيانات الي هذه الأصناف الثلاثة:

١. **High-Level (Conceptual) Data Model:**

وهو قريب جدا من كيفية إدراك المستخدم للبيانات

ويستخدم هذا النموذج الأفكار الآتية:

- **Entities (كيانات):** وهي تمثل كيان حقيقي يتم التعامل معه مثل: الطلبة – المقررات – المشاريع - .....
- **Attributes (صفات):** وهي تمثل خواص للكيان مثل: الأسماء – الدرجات – أرقام الهواتف _ .....
- **Relationships (العلاقات):** وهي تمثل العلاقات بين الكيانات مثل علاقة الطالب-المقرر - .....

تصنيفات نماذج البيانات

٢. **Low-Level (Physical) Data Model:**

وهو يقوم بوصف كيفية تمثيل البيانات داخل الحاسب

- يستخدمه المتخصصين في الحاسبات
- يهتم هذا النموذج بكيفية تمثيل البيانات وكذلك التعامل معها داخل الحاسبات بطريقة ذات كفاءة عالية

تصنيفات نماذج البيانات

٣. **Representation (Implementation) Data Model:**

- هو نموذج متوسط بين الأول و الثاني
- يحتوي علي المبادئ الذي يفهمها المستخدم كما في النموذج الأول ولكن يمكن أن يحتوي علي بعض التفاصيل الخاصة ببناء البيانات وكيفية التعامل معها بكفاءة مثل النموذج الثاني
- يستخدم هذا النموذج في معظم نظم إدارة قواعد البيانات الموجودة
- يحتوي هذا النموذج علي نماذج البيانات الأكثر استخداما وهو النموذج العلائقي وأيضا النماذج القديمة مثل النموذج الشبكي و النموذج الهرمي

مخططات قواعد البيانات (Schemas)

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

- يستخدم المخطط عند تصميم قواعد البيانات
- هذا المخطط لا يتوقع تغيره بشكل تكرارى
- 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

الشعبة

رقم-الشعبة	رقم-المقرر	الفصل	السنة	المحاضر

كشف-الدرجات

رقم الطالب	رقم الشعبة	الدرجة

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

هيكلة نظم قواعد البيانات (DB System)

تم اقتراح هيكل لنظم قواعد البيانات يحتوي علي ثلاث مستويات من المخططات وذلك لدعم الخواص التي يجب أن تقدمها نظم إدارة قواعد البيانات والتي تم مناقشتها سابقا

وهذه المستويات هي:

١. **المستوى الداخلي (Internal Level):**

- وهو يحتوي علي المخطط الداخلي والذي يقوم بوصف التخزين الفعلي لقواعد البيانات
- هذا المخطط الداخلي يتم وصفه باستخدام النموذج الآتي **(Physical Data Model)**

هيكلة نظم قواعد البيانات (DB System)

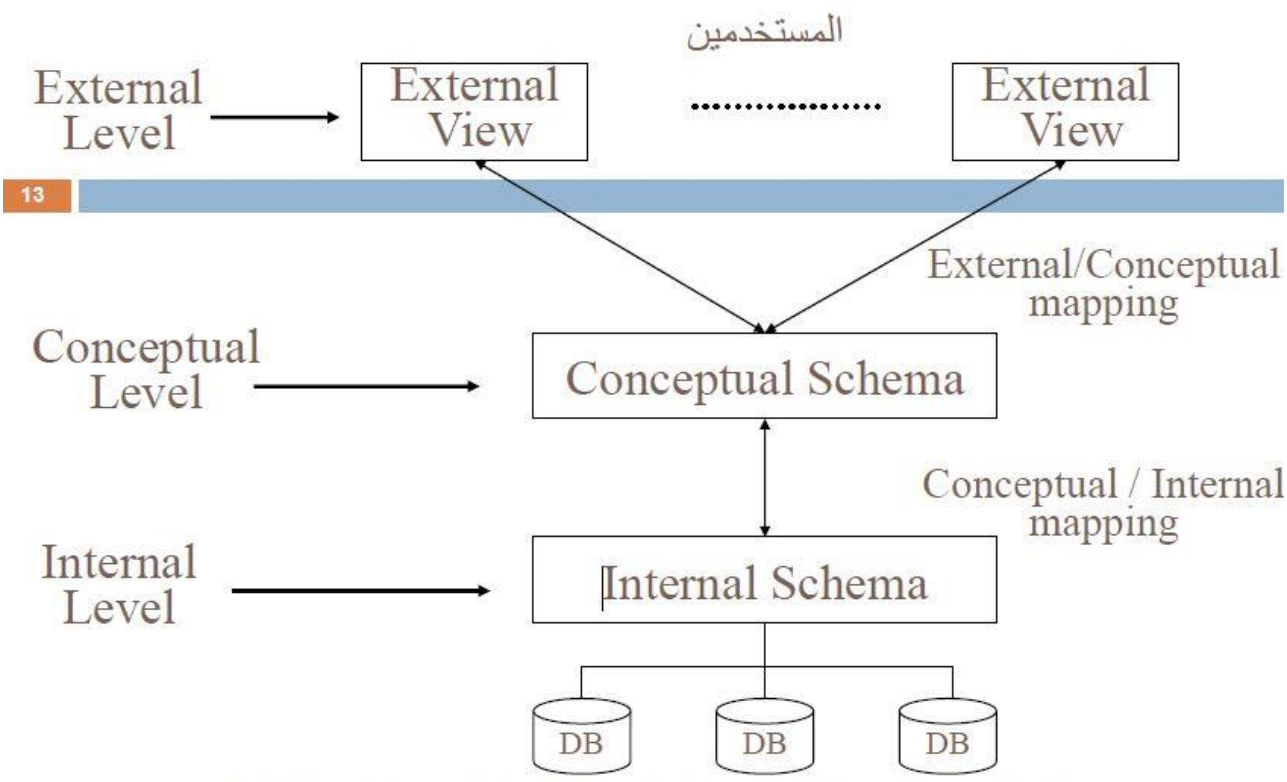
٢. **The Conceptual Level:**

- يحتوي علي Conceptual Schema التي توصف بناء البيانات في قواعد البيانات
- 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

هيكلة نظم قواعد البيانات (DB System)

٣. **The External or View Level:**

- يحتوي علي مجموعة من الأشكال التي يعرفها المستخدم
- تقدم لكل مجموعة من المستخدمين شكل معين للبيانات هو عبارة عن جزء معين من قواعد البيانات هؤلاء المستخدمين لهم اهتمام به
- يمكن أيضا استخدام Conceptual data model أو Representation data model في بناء هذا المستوي



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)

مقدمة لقواعد البيانات

المحاضرة الثالثة

مقدمة لنظم إدارة قواعد البيانات

الجزء الثاني

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)

استقلالية البيانات (Data Independence)

- **الاستقلال المنطقي (Logical Data Independence):**
- هي المقدرة علي تغيير مخطط البيانات في المستوي الثاني (Conceptual Level) بدون الحاجة إلى تغير المخطط في المستوي الثالث (External Level) وكذلك بدون تغيير البرامج التطبيقية
- يكون التغير 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

استقلالية البيانات (Data Independence)

- **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 التغير 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 تعريف 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 الثلاثى بطريقة حقيقية وذلك لتعريف مخطط البيانات في المستوي الثالث (External Level)

لغات نظم إدارة قواعد البيانات

- **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
  - يمكن أن يتم كتابة لغة SQL داخل لغة عالية المستوي مثل .. ,C, Pascal وعند ذلك لابد من وجود ما يسمى (Precompiler) حيث يقوم بفصل لغة SQL وإرسالها الى نظم إدارة قواعد البيانات بينما يتم إرسال اللغة عالية المستوى إلى مترجم اللغة

واجهات التعامل مع نظم إدارة قواعد البيانات (DBMS Interface)

- **التعامل عن طريق القوائم (Menu-based Interface):**
  - يتم التعامل بين المستخدم و النظام عن طريق مجموعة من القوائم التي تحول طلبات المستخدم إلى اللغة التي تتعامل مع نظم إدارة قواعد البيانات والتي تقوم بدورها بتنفيذ تلك الأوامر
- **التعامل عن طريق النماذج (Form-based Interface):**
  - يتم التعامل بين المستخدم و النظام عن طريق مجموعة من النماذج التي يستخدمها المستخدم لإدخال البيانات الجديدة أو استرجاع البيانات المطلوبة
  - تستخدم هذه النماذج عادة من المستخدمين الذين ليس لديهم خبرة في التعامل مع قواعد البيانات

واجهات التعامل مع نظم إدارة قواعد البيانات (DBMS Interface)

- **التعامل عن طريق الرسومات (Graphical User Interface):**
  - وذلك عن طريق عرض مخطط البيانات عن طريق الرسومات ثم استخدام هذه الأشكال وكذلك بعض الأدوات المرسومة والتي يوفرها النظام لكى يقوم المستخدم بتعريف الاستفسارات التي يحتاجها في التعامل مع البيانات
- **التعامل عن طريق اللغات الطبيعية (Natural Language Interface):**
  - وذلك عن طريق توفير نظام يقوم بقبول طلبات المستخدم بلغة قريبة من لغة المستخدم (عربي أو انجليزي) وذلك لتعريف الاستفسارات التي يريدها

واجهات التعامل مع نظم إدارة قواعد البيانات (DBMS Interface)

- **التعامل مع المستخدمين قليلي الخبرة(Parametric User Interface):**
  - وذلك عن طريق توفير واجهات تعامل سهلة يتم استخدام المفاتيح الوظيفية على لوحة المفاتيح أو كلمات مختصرة للأوامر المطلوبة وذلك لتقليل الكلمات التي يكتبها المستخدم
- **التعامل مع مدير قواعد البيانات (DBA Interface):**
  - يكون لمدير قواعد البيانات واجهات تعامل خاصة تمكنه من القيام بوظائفه الخاصة بتعريف البيانات ومراقبة النظام والتحكم في الصلاحيات المعطاة لكل مستخدم

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

بيئة نظم قواعد البيانات

- **قبل المترجم (Pre-Compiler):**

هو برنامج يقوم باستخلاص أوامر التعامل مع البيانات (DML) من داخل البرامج المكتوبة بلغات عالية المستوى ثم إرسال هذه الأوامر إلى المترجم الخاص بها وإرسال البرنامج المكتوب باللغة عالية المستوى إلى مترجمها الخاص بها

- **مترجم لغة التعامل مع البيانات (DML Compiler):**

يقوم بترجمة الأوامر الخاصة بالتعامل مع البيانات (DML) لإنشاء برامج يتم إرسالها إلى منفذ أوامر قواعد البيانات (Run-Time processor) ليتم تنفيذ هذه الأوامر

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 الي 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

خدمات تقدمها نظم إدارة قواعد البيانات

- **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)

مقدمة لقواعد البيانات

المحاضرة الرابعة

الجزء الأول

النموذج العلاقي
(RELATIONAL MODEL)

مبادئ النموذج العلاقي

Relational Data Model Concepts

١. يقوم النموذج العلاقي بتمثيل قواعد البيانات كمجموعة من العلاقات (الجداول)

٢. النموذج العلاقي يتم استخدامه على نطاق كبير بسبب سهولته وبسبب وجود أساس رياضي له

٣. كل جدول في النموذج يحتوي على مجموعة من الصفوف التي تمثل مجموعة من البيانات المترابطة

النموذج العلاقي للبيانات

- كل صف في الجدول يسمى Tuple
- كل عمود في الجدول يمثل قيم صفة معينة (Attribute)
- كل جدول يسمى علاقة
- المدى (Domain) هو وصف لنوع البيانات التي تتواجد في عمود معين

المدى في النموذج العلاقي للبيانات

- **المدى (Domain) هو مجموعة القيم الأولية التي لاتقبل التقسيم لوحدات أصغر في العلاقة التي تستخدم تلك القيم**

مثال:

الاسم بطول ١٠ حروف: هو مجموعة الاسماء التي تحتوي على ١٠ حروف بحد أقصى وتمثل اسماء أشخاص

درجة الطالب اعداد صحيحة بطول ٣ أرقام: تمثل درجات الطلبة على أن تكون اعداد صحيحة وبحد أقصى ٣ أرقام

رقم الهاتف بطول ٧ أرقام : يمثل أرقام الهواتف بحيث يحتوي كل عنصر على ٧ أرقام

يحتوي تعريف كل بيان على نوع البيانات (أرقام – حروف – تاريخ ... الخ) وحجم البيان وكذلك بعض الشروط المطلوبة

تعريفات في النموذج العلاقي للبيانات

يوجد لمخطط البيانات العلاقي $R(A_1, A_2, .. A_n)$ أو [ر(ا١ ، ا٢،...ان)] تلك التعريفات:

١. اسم العلاقة هو (ر) أو(R)

٢. (ا١ ، ا٢، ا٣، ....ان) أو $(A_1, A_2, A_3, .. A_n)$ هي الصفات (Attributes)

٣. كل صفة (Attribute) هي اسم لدور يقوم به مدى معين داخل العلاقة

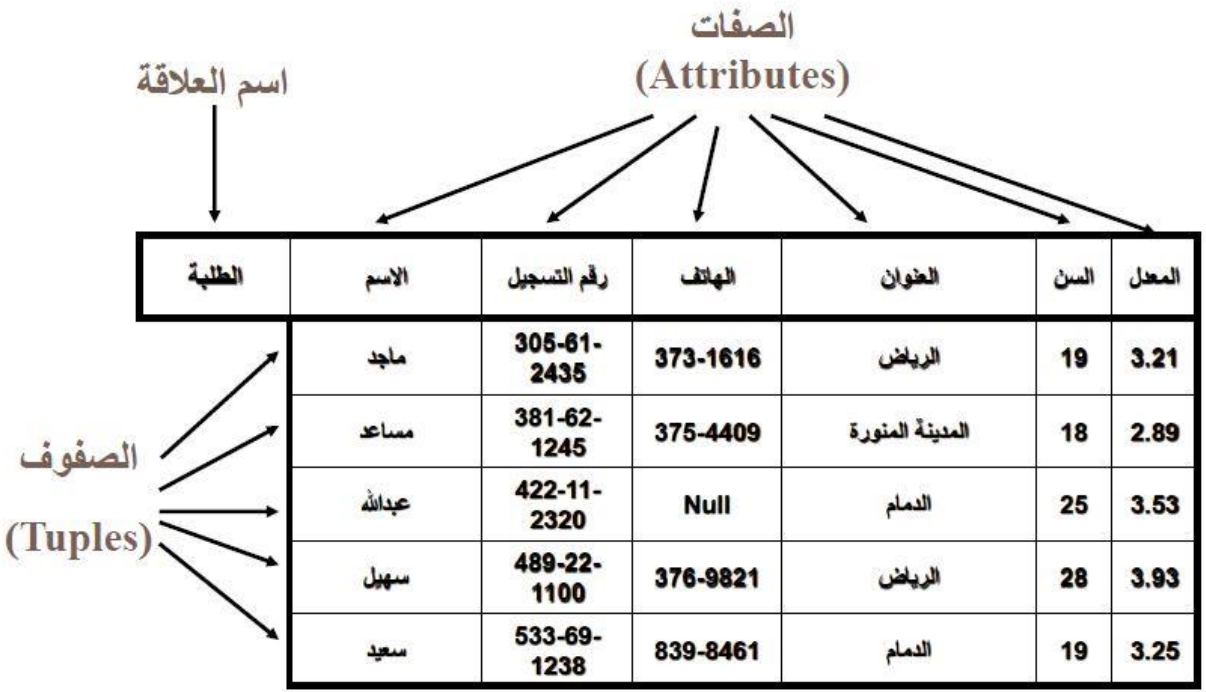
٤. درجة العلاقة [درجة (ر)] أو (deg(R))هي عدد الصفات المتواجدة في علاقة معينة

تعريفات في النموذج العلاقي للبيانات

مثال:

الطلبة(الاسم ، رقم التسجيل ، الهاتف ، العنوان، السن، المعدل)

١. اسم العلاقة هو: الطلبة
٢. الصفات (Attributes) هي: الاسم ، رقم التسجيل ، الهاتف ، العنوان، السن، المعدل
٣. كل صفة (Attribute) هي اسم لدور يقوم به مدى معين داخل العلاقة (الاسم هو اسم الطالب، .... المعدل هو معدل التراكمي)
٤. درجة العلاقة [درجة (الطلبة)] = ٦



الطلبة	الاسم	رقم التسجيل	الهاتف	العنوان	السن	المعدل
	ماجد	305-61-2435	373-1616	الرياض	19	3.21
	مساعد	381-62-1245	375-4409	المدينة المنورة	18	2.89
	عبدالله	422-11-2320	Null	الدمام	25	3.53
	سهيل	489-22-1100	376-9821	الرياض	28	3.93
	سعيد	533-69-1238	839-8461	الدمام	19	3.25

شكل ١: يوضح الصفات والصفوف داخل علاقة الطلبة

تعريفات في النموذج العلاقي للبيانات

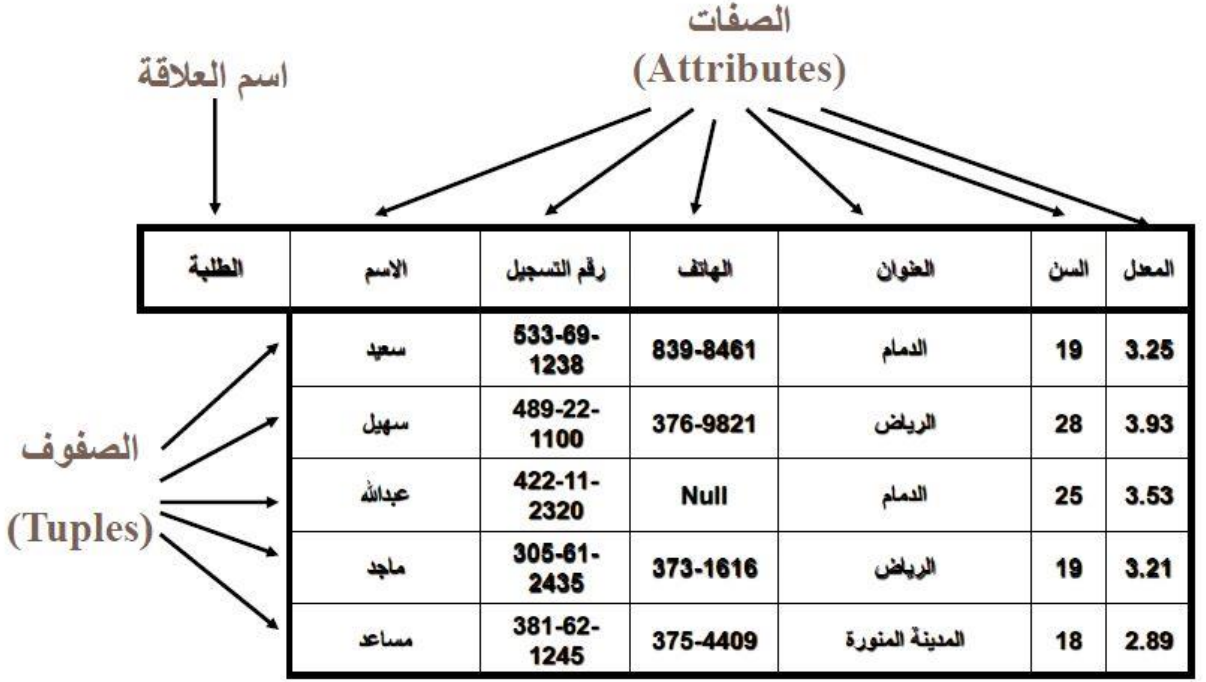
ملاحظات على الشكل السابق:

- تسمى حالة البيانات في وضع معين (مثل الشكل السابق مثلا) بحالة البيانات
- هذه الحالة تتغير من وقت لآخر حسب التعديلات التي تجرى على البيانات
- لابد أن تتوافق البيانات داخل كل عمود مع المدى الذى يصف ذلك العمود
- القيم الغير معرفة والتي يرمز لها بالرمز (Null) تمثل قيم غير معروفة إلى الآن أو قيم غير متواجدة

خواص العلاقات (الجداول)

ترتيب الصفوف:

- ترتيب الصفوف داخل العلاقة لا يعتبر جزء من تعريف العلاقة ذاتها لأن العلاقة هي مجموعة من الصفوف والمجموعات لا تعطي أي اعتبار للترتيب
- لا يوجد أفضلية معينة لترتيب صفوف داخل علاقة على ترتيب أخر بل يتوقف ذلك على التطبيق المطلوب
- أى ترتيب للصفوف داخل العلاقة لا يعتبر علاقة جديدة بل نعتبر العلاقة الأصلية كما هى ولكن فقط بشكل مختلف مع الاحتفاظ بكل التعريفات والخواص
- الشكل التالى يوضح العلاقة السابقة للطلبة ولكن في ترتيب مختلف



الطلبة	الاسم	رقم التسجيل	الهاتف	العنوان	السن	المعدل
	سعيد	533-69-1238	839-8461	الدمام	19	3.25
	سهيل	489-22-1100	376-9821	الرياض	28	3.93
	عبدالله	422-11-2320	Null	الدمام	25	3.53
	ماجد	305-61-2435	373-1616	الرياض	19	3.21
	مساعد	381-62-1245	375-4409	المدينة المنورة	18	2.89

شكل ٢: يوضح علاقة الطلبة مع ترتيب مختلف للصفوف

خواص العلاقات (الجداول)

ترتيب القيم داخل الصفوف:

- ترتيب القيم داخل الصف مهمة ويجب أن يتم التعامل مع القيم داخل الصفوف على هذا النحو
- كل قيمة داخل الصف هي قيمة أساسية لابد وأن تتبع المدى الذي يعرف هذه القيم
- لابد وأن يتم إدخال القيم والتعامل معها بنفس الترتيب المفروض وقت بناء العلاقة (الجدول)

مقدمة لقواعد البيانات

المحاضرة الرابعة

الجزء الثاني

النموذج العلاقي
(RELATIONAL MODEL)

القيود في النموذج العلاقي
(Relational Model Constraints)

يتميز النموذج العلاقي لبناء قواعد البيانات بإمكانية وضع القيود على البيانات وكيفية التعامل معها مما يتيح للمستخدم تعريف هذه القيود عند بناء مخطط البيانات ثم تقوم نظم إدارة قواعد البيانات بتنفيذ هذه القيود على النموذج كلما تم التعامل معه مما يضمن توافق البيانات في أي وقت مع المتطلبات التى يضعها المستخدم وقت البناء.

القيود في النموذج العلاقي
(Relational Model Constraints)

وتنقسم هذه القيود إلى:

١. قيود المدى (Domain Constraints)

٢. قيود المفتاح (Key Constraints)

٣. قيود القيم الغير المتواجدة (Null Constraint)

٤. قيود تكامل الكيان (Entity Integrity Constraint)

٥. قيود التكامل المرجعي (Referential Integrity Constraint)

القيود في النموذج العلاقي
(Relational Model Constraints)

١. **قيود المدى (Domain Constraints)**

كل صفة (Attribute) لابد وأن يكون قيمة أساسية لاتقبل التقسيم وأن تكون من المدى الذي يعرف هذه الصفة

مثال: اسم الطالب:

لابد وأن يكون حرفي ولا يزيد طوله عن الطول المعرف لهذه الصفة
ولابد وأن يكون أسم شخص (ليست أية مجموعة حروف متلاصقة)
وكذلك لا يمكن أن نقسم الاسم ونتعامل مع كل جزء بطريقة مباشرة

القيود في النموذج العلاقي
(Relational Model Constraints)

٢. **قيود المفتاح (Key Constraints)**

- **بما أن العلاقة تم تعريفها على إنها مجموعة من الصفوف ولما كانت المجموعات له صفة أساسية وهي عدم التكرار لمكوناتها وهي الصفوف، فلذلك لا يمكن أن نجد أكثر من صف يحتوي على نفس جميع القيم المتواجدة في الصف الأخر وذلك لجميع الصفات**

القيود في النموذج العلاقي
(Relational Model Constraints)

٢. **قيود المفتاح (Key Constraints)**

- ويوجد عادة مجموعة من الصفات لا تتكرر في أكثر من صف وتسمى بالمفتاح الأكبر (Super key SK)
- هذا المفتاح (SK) لابد وأن يتوافر فيه هذا الشرط:

قيمة المفتاح في أي صف لا يمكن أن تتواجد في صف أخر

أو بطريقة رياضية أوضح: $t_1[SK] \neq t_2[SK]$

حيث أن t_1 هو الصف الأول و t_2 هو الصف الثاني و SK هي قيمة الصفات التي تمثل المفتاح

القيود في النموذج العلاقي
(Relational Model Constraints)

٢. **قيود المفتاح (Key Constraints)**

- المفتاح الأكبر (Super key SK) يمكن أن يحتوي على صفات زائدة غير ضرورية لتعريف المفتاح
- ولذلك تم استخدام تعريف محدد للمفتاح (Key) وهو عبارة عن أصغر مجموعة من الصفات يمكن أن تعرف أي صف بدون أن تتكرر
- **مثال للمفتاح:**

رقم الطالب في الجامعة، رقم الهوية، رقم رخصة القيادة

مع العلم أن أي صفة مع أية من الصفات السابقة تعطي مفتاح ولكنه من النوع (مفتاح أكبر) (Super Key)

القيود في النموذج العلاقي
(Relational Model Constraints)

٢. **قيود المفتاح (Key Constraints)**

- ممكن أن تحتوي العلاقة على عدة مفاتيح كما بالمثال التالي الذي يحتوي على رقم الرخصة أو رقم المسلسل لموتور السيارة كمفتاح (Key)
- تسمى هذه المفاتيح بالمفاتيح المؤهلة (Candidate Keys)
- المفتاح الأساسي هو أحد المفاتيح المؤهلة وهو المفتاح الذي يتم اختياره ليمثل المفتاح الرئيسي للعلاقة (Primary Key)
- المفتاح الرئيسي يلعب دور مهم جدا في العلاقات ويجب تحديده بدقة والتأكد أنه لا يمكن أن يتكرر تحت أي ظرف لأكثر من صف

القيود في النموذج العلاقي
(Relational Model Constraints)

٢. **قيود المفتاح (Key Constraints)**

- **الصفات التي تمثل المفتاح الرئيسي داخل العلاقة يتم وضع خط تحتها لتوضيح أن هذه الصفات تمثل المفتاح الرئيسي المختار**

- **مثال:**

المقررات (رقم المقرر – اسم المقرر – اسم الدبلوم – المستوى – عدد الساعات)

الطلبة (رقم الطالب – الاسم الأول – اسم الأب –اسم العائلة – الهاتف – المدينة)

التسجيل (رقم الطالب – رقم المقرر – العام الأكاديمي – الدرجة)

المفتاح الرئيسي

Primary key

المفاتيح المؤهلة

Candidate keys

سيارات	رقم-الرخصة	رقم-مسلسل-الموتور	الشركة	الموديل	السنة
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

شكل٣: يمثل بيانات علاقة للسيارات بها أكثر من مفتاح

القيود في النموذج العلاقي
(Relational Model Constraints)

٣. **قيود القيم الغير المتواجدة (Null Constraint).**

- يمكن أن نستخدم خاصية (قيم مطلوبة) عند تعريف العلاقة وذلك لضمان إدخال المستخدم لهذه القيمة وعدم قبول نظم إدارة قواعد البيانات لعدم وجود قيمة لهذه الصفة في أي وقت

- **مثال:**

تستخدم مع تعريف اسم الطالب حيث لا يصح أن يتواجد طالب بدون اسم. وكذلك مع أي صفة أخرى لابد من وجود قيمة لها

القيود في النموذج العلاقي
(Relational Model Constraints)

٤. **قيود تكامل الكيان (Entity Integrity Constraint)**

- لا يمكن أن يكون المفتاح الرئيسي غير معرف (Null) لأي صف
- أي أن تعريف الصفة أو الصفات كمفتاح رئيسي يعني أنها لا يمكن أن تكون غير معرفة في أي وقت ولا يلزم أن تقوم بوضع شرط غير معرف عليها
- هذا قيد تكامل يعني أن الكيان متكامل في أي وقت وأن نظم إدارة قواعد البيانات تتحقق من هذا الشرط على الدوام

القيود في النموذج العلاقي
(Relational Model Constraints)

٥. **قيود التكامل المرجعي (Referential Integrity Constraint)**

- هذه القيود تعرف بين علاقتين
- تستخدم هذه القيود للحفاظ على التوافق بين البيانات المتواجدة في الجدولين
- **هذه القيود تعني الأتي:**

الصف الموجود في علاقة معينة عندما يرتبط (يشير) إلي علاقة أخرى لابد وأن يشير لصف موجود في هذه العلاقة

القيود في النموذج العلاقي
(Relational Model Constraints)

- **قيود التكامل المرجعي (Referential Integrity Constraint)**

- يستخدم مبدأ المفتاح الأجنبي (Foreign Key) لتحديد وتعريف التكامل المرجعي.
- يعتبر التكامل المرجعي من أهم القيود التي يجب أن يدرسها مصمم البيانات وأن يعرفها بدقة حيث أنها تساعد على توافق البيانات بشكل كبير جدا وتجعل المستخدم يتعامل مع البيانات بسهولة شديدة تاركاً أمر التكامل على نظم إدارة قواعد البيانات

القيود في النموذج العلاقي
(Relational Model Constraints)

٥. **قيود التكامل المرجعي (Referential Integrity Constraint)**

- **المفتاح الأجنبي (Foreign Key):**

هو صفة أو مجموعة من الصفات (FK) في علاقة معينة (R1) مثلاً بحيث تشير إلي علاقة (R2) لو تحققت الشروط التالية:

١. الصفات المكونة للمفتاح الأجنبي لابد و أن يكون لها نفس المدى للمفتاح الرئيسي (Primary Key) في العلاقة R2 المشار اليها

٢. القيمة في المفتاح الأجنبي في أي حالة من حالات العلاقتين لابد وأن تشير لصف موجود أو أن تكون قيمة صفات المفتاح الأجنبي غير معرفة (Null)

أي رياضيا تكتب مثل: $(t_1[FK] = t_2[PK])$ or is null

القيود في النموذج العلاقي
(Relational Model Constraints)

- **المفتاح الأجنبي (Foreign Key):**
  - المفتاح الأجنبي يرسم على مخطط البيانات عن طريق سهم يخرج من المفتاح الأجنبي وينتهي عند المفتاح الرئيسي في العلاقة المشار إليها
  - قيود التكامل المرجعي لابد وأن يتم تعريفها وقت بناء العلاقات وتقوم نظم إدارة قواعد البيانات بتنفيذ تلك القيود
  - قيود التكامل المرجعي هي جزء من تعريف البيانات ويوجد أن تدرس قبل البدء في عملية الإنشاء
  - المثال التالي يوضح مخطط علاقات قبل و بعد و ضع قيود التكامل

الموظفون

رقم-القسم	الراتب	الجنس	العنوان	تاريخ-الميلاد	رقم-الموظف	اسم-العائلة	اسم-الأب	الاسم

الأقسام

رقم-المدير	رقم-القسم	اسم-القسم

المشاريع

رقم-القسم	مكان-المشروع	رقم-المشروع	اسم-المشروع

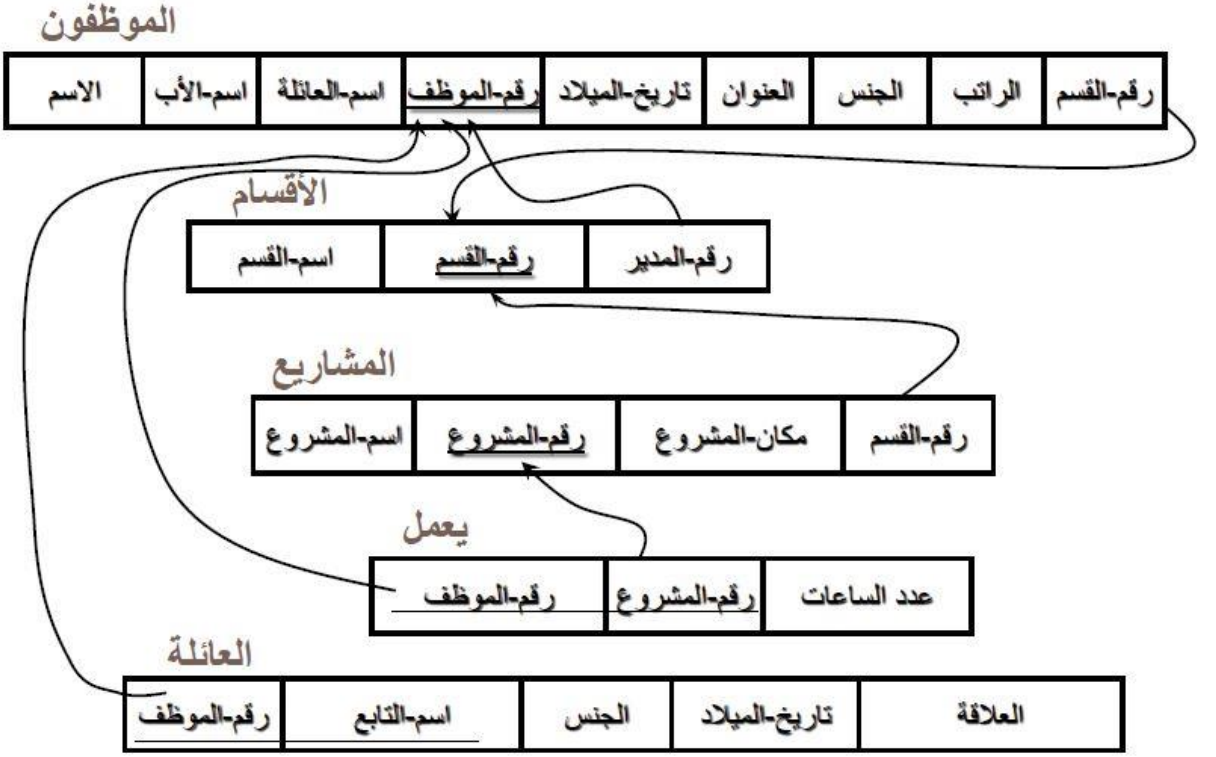
يعمل

عدد الساعات	رقم-المشروع	رقم-الموظف

العائلة

العلاقة	تاريخ-الميلاد	الجنس	اسم-التابع	رقم-الموظف

شكل ٤: يمثل هذا الشكل مخطط بيانات شركة قبل قيود التكامل



شكل ٥: يمثل هذا الشكل مخطط بيانات شركة بعد تعريف قيود التكامل

أمثلة

29

١ـ تسجيل الطلبة لمقررات في جامعة

إذا كانت بيانات الطلبة و المقررات والتسجيل كما هو مبين في التالي:

طالب (رقم الطالب – الاسم الأول – اسم العائلة – اسم الأب – المدينة – تاريخ الميلاد)

مقرر (رقم المقرر – اسم المقرر – الدبلوم – المستوى – عدد الساعات)

التسجيل (رقم الطالب – رقم المقرر – العام الجامعي – الفصل – الدرجة)

المطلوب هو تحديد شكل مخطط البيانات العلاقي موضحاً عليه المفتاح الرئيسي لكل علاقة وكذلك المفاتيح الأجنبية

أمثلة

١- تسجيل الطلبة لمقررات في جامعة

طالب (رقم الطالب – الاسم الأول – اسم العائلة – اسم الأب – المدينة – تاريخ الميلاد)

مقرر (رقم المقرر – اسم المقرر – الدبلوم – المستوى – عدد الساعات)

التسجيل (رقم الطالب – رقم المقرر – الفصل – الدرجة)

٢- تسجيل مبيعات شركة تجارية

إذا كانت بيانات مبيعات شركة تجارية هي:

المنتج (رقم المنتج – الاسم – بلد الصنع – الوحدة)

العميل (اسم العميل – رقم العميل – الهاتف – العنوان)

البائع (اسم البائع ـــ رقم البائع - الجنسية - الهاتف – تاريخ التعين)

الفاتورة (رقم فاتورة البيع – رقم المنتج – رقم العميل - رقم البائع – التاريخ – الكمية – المبلغ)

المطلوب هو تحديد شكل مخطط البيانات العلاقي موضحاً عليه المفتاح الرئيسي لكل علاقة وكذلك المفاتيح الأجنبية

أمثلة

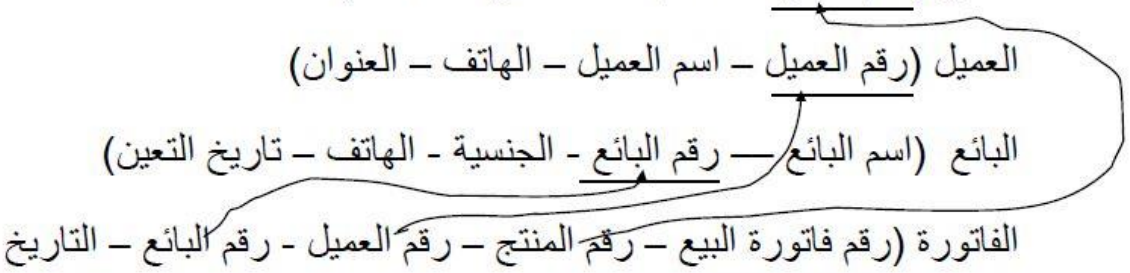
٢- تسجيل مبيعات شركة تجارية

المنتج (رقم المنتج – الاسم – بلد الصنع – الوحدة)

العميل (رقم العميل – اسم العميل – الهاتف – العنوان)

البائع (اسم البائع – رقم البائع - الجنسية - الهاتف – تاريخ التعين)

الفاتورة (رقم فاتورة البيع – رقم المنتج – رقم العميل - رقم البائع – التاريخ – الكمية – المبلغ)



٣- تسجيل حركة الاستعارة في مكتبة الجامعة

إذا كانت بيانات الطلبة و الكتب والتسجيل كما هو مبين في التالي:

الكتاب (رقم الكتاب – عنوان الكتاب – الناشر – سنة النشر – عدد الصفحات)

المستعير (رقم المستعير – اسم المستعير – العنوان – الهاتف)

الاستعارة (رقم الكتاب– رقم المستعير– تاريخ الاستعارة – تاريخ الرجوع)

المطلوب هو تحديد شكل مخطط البيانات العلاقي موضحاً عليه المفتاح الرئيسي لكل علاقة وكذلك المفاتيح الأجنبية

٣- تسجيل حركة الاستعارة في مكتبة الجامعة

الكتاب (رقم الكتاب – عنوان الكتاب – الناشر – سنة النشر – عدد الصفحات)

المستعير (رقم المستعير – اسم المستعير – العنوان – الهاتف)

الاستعارة (رقم الكتاب – رقم المستعير– تاريخ الاستعارة – تاريخ الرجوع)

أمثلة

٤- تسجيل حركة تأجير السيارات في مكتب لتأجير السيارات

إذا كانت بيانات السيارات والعملاء والتسجيل كما هو مبين في التالي:

السيارة (رقم السيارة – الموديل – سنة الصنع – اللون)

المستأجر (رقم المستأجر – الاسم– الهاتف – العنوان)

الإيجار (رقم المستأجر – رقم السيارة – تاريخ الإيجار – المدة - التكلفة)

المطلوب هو تحديد شكل مخطط البيانات العلاقي موضحاً عليه المفتاح الرئيسي لكل علاقة وكذلك المفاتيح الأجنبية

٤- تسجيل حركة تأجير السيارات في مكتب لتأجير السيارات

السيارة (رقم السيارة – الموديل – سنة الصنع – اللون)

المستأجر (رقم المستأجر – الاسم– الهاتف – العنوان)

الإيجار (رقم المستأجر – رقم السيارة – تاريخ الإيجار – المدة - التكلفة)

مقدمة لقواعد البيانات

المحاضرة الرابعة

نمذجة البيانات باستخدام نموذج الكينونة/العلاقة
(DATA MODELING USING THE ENTITY RELATIONSHIP MODEL)

الجزء الأول

- نموذج الكينونة/العلاقة (ERD) هو نموذج عالي المستوي يقوم بعرض بناء البيانات (high-level conceptual data model).

- يتم استخدام هذا النموذج في العادة أثناء مرحلة تصميم المفاهيم (Conceptual Design).

- الناتج من استخدام هذا النموذج هو مخطط البيانات.

- يتم تمثيل بناء البيانات والقيود المطلوبة عليها باستخدام اشكال رسومية سهلة ومحددة.

مفاهيم النموذج ومصطلحاته

- **الكيان (Entity):**

هو الوحدة (الشيء) الأساسية التي يتم تمثيلها بنموذج الكينونة/العلاقة (ER) 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)

- **مفتاح الكيان: لكل كيان مفتاح عبارة عن صفة أو صفات تعرف الكيان بطريقة وحيدة.**
- **يتم تمثيل الكيان باستخدام شكل مستطيل**

كما هو مبين:

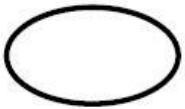
مفاهيم النموذج ومصطلحاته

الصفة (Attribute):

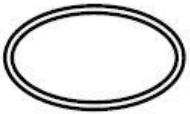
هي صفة معينة تصف الكيان مثل اسم الموظف ، عمر الطالب، مرتب موظف، درجة طالب، عدد الساعات الدراسية لمقرر.

- 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.

- يتم تمثيل الصفة باستخدام شكل بيضاوي كالآتي:



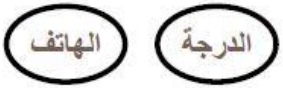
- الصفة متعددة القيم يتم تمثيلها بالشكل البيضاوى المزدوج كما هو موضح بالشكل التالي:



مفاهيم النموذج ومصطلحاته

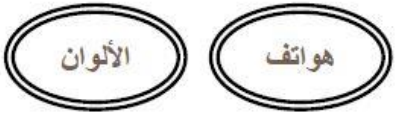
أمثلة توضيحية للصفات (Attributes):

- الصفة البسيطة مثل درجة طالب، هاتف موظف



- الصفة المركبة مثل
  اسم الطالب (الاسم الأول – اسم العائلة)

- الصفة متعددة القيم مثل هواتف موظف (قد يكون له هاتف او اثنين أو أكثر)، الوان سيارة (قد تكون من لون واحد أو أي عدد من الألوان)



مفاهيم النموذج ومصطلحاته

العلاقات

- العلاقة ر (R) بين مجموعة من الكيانات ك١، ك٢، .....كن (E1,E2, ...En) هي مجموعة تمثل الارتباطات بين هذه الكيانات.

- كل وحدة (Instance) في العلاقة ر هي عبارة عن اتحاد أو ارتباط بين الكيانات المرتبطة بهذه العلاقة بحيث أن هذه الوحدة (Instance) تحتوي على صف واحد من كل كيان مشارك في العلاقة.

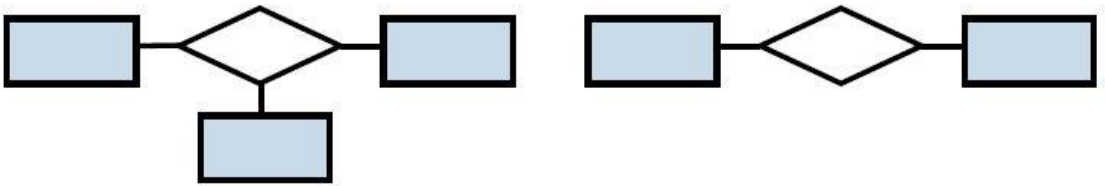
- المرجعية من كيان إلى كيان آخر يجب أن يتم تمثيلها في نموذج الكينونة/العلاقة باستخدام "علاقة" وليس كصفة في الكيان.

مفاهيم النموذج ومصطلحاته

- يتم تمثيل العلاقة في نموذج الكينونة/العلاقة باستخدام شكل المعين (Diamond shape) والذي يرتبط مع الكيانات بخطوط مستقيمة وهذا يتم تمثيله بالشكل التالي:

- لكل علاقة درجة (Degree of Relationship):

وهي عدد الكيانات المرتبطة بالعلاقة (ثنائية – ثلاثية - ....)



أنواع القيود على العلاقات

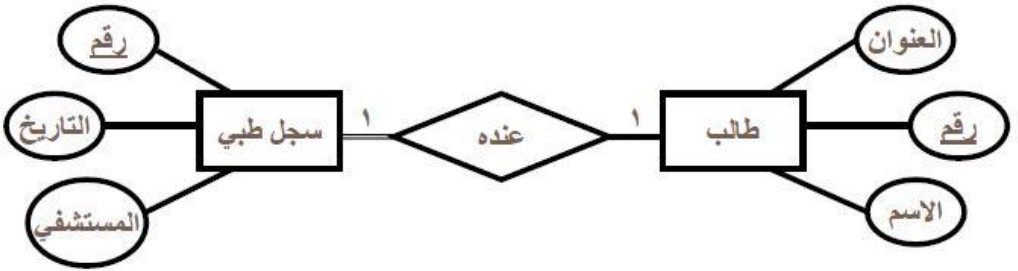
- **نوع العلاقة (Cardinality Ratio) :**

هي عدد الوحدات (Instances) في العلاقة التي يمكن أن يشترك فيها الكيان وهي عبارة عن الأنواع الآتية:

- **١:١ علاقة واحد-الى-واحد (one-to-one)**
- **1:N علاقة واحد-الى-كثير (one-to-many)**
- **M:N علاقة كثير-الى-كثير (many-to-many)**

مثال لأنواع العلاقات

علاقة ١:١ { واحد-الى-واحد (one-to-one)}

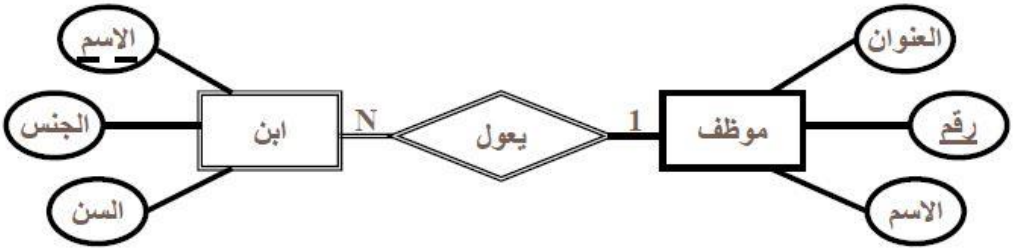


ملاحظات:

- لكل طالب سجل واحد (نوع العلاقة ١).
- السجل يكون لطالب واحد (نوع العلاقة ١).

مثال لأنواع العلاقات

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

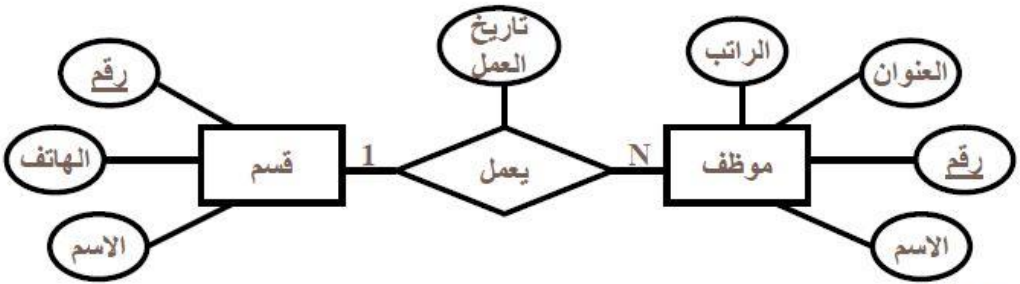


ملاحظات:

- لكل ابن موظف واحد.
- الموظف قد يكون له عدة ابناء.

مثال لأنواع العلاقات

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

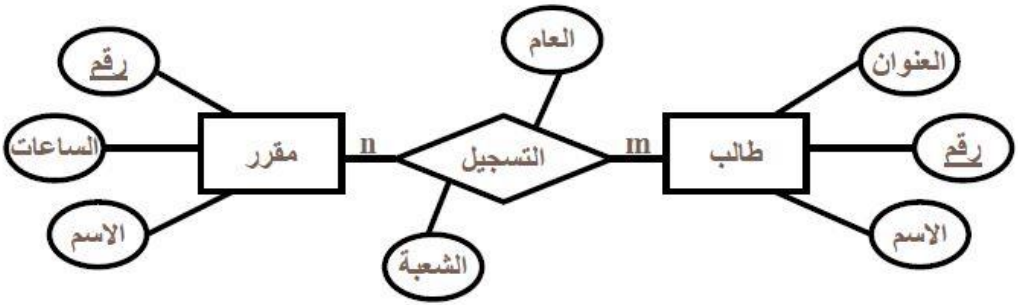


ملاحظات:

- لكل موظف قسم واحد.
- القسم قد يكون فيه عدة موظفون.

مثال لأنواع العلاقات

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



ملاحظات:

- الطالب قد يكون له عدة مقررات.
- المقرر يمكن أن يسجله عدة طلبة.

مقدمة لقواعد البيانات

المحاضرة الرابعة

نمذجة البيانات باستخدام نموذج الكينونة/العلاقة
(DATA MODELING USING THE ENTITY RELATIONSHIP MODEL)

الجزء الثاني

أنواع القيود على العلاقات

- **نوع الاشتراك (Participation Constraint):**

هو يحدد ما إذا كان وجود الكيان يعتمد على كونه مرتبط بكيان آخر عن طريق العلاقة.

أنواع قيود الاشتراك (Participation constraints):

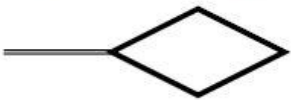
١. **اشتراك كلي (Total participation).**
٢. **اشتراك جزئي (Partial participation).**

أنواع القيود على العلاقات

أنواع قيود الاشتراك (Participation constraints):

١. **اشتراك كلي (Total participation):**

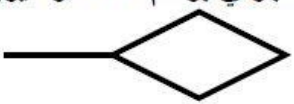
- كل كيان يجب أن يرتبط بوحدة (Instance) في العلاقة.
- يتم تمثيل قيد الاشتراك الكلي برسم خط مزدوج يربط الكيانات المرتبطة بهذه العلاقة مثل:



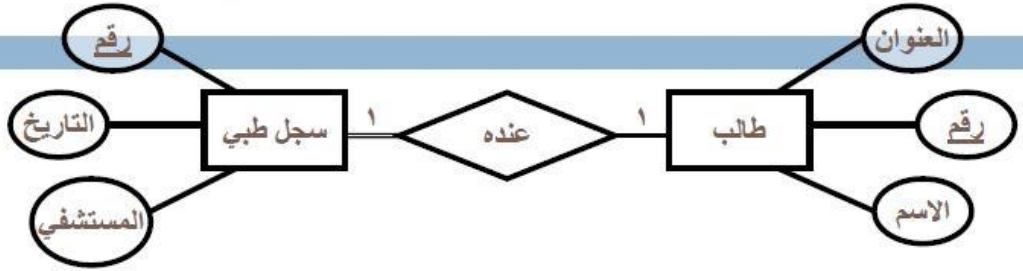
- يسمي هذا القيد بقيد "ارتباط الوجود" (Existence Dependency).

٢. **اشتراك جزئي (Partial participation):**

- بعض الكيانات ترتبط ببعض الوحدات (Instances) في العلاقة.
- يتم تمثيل قيد الاشتراك الجزئي برسم خط مفرد يربط الكيانات المرتبطة بهذه العلاقة مثل:



مثال على أنواع الارتباط



ملاحظات:

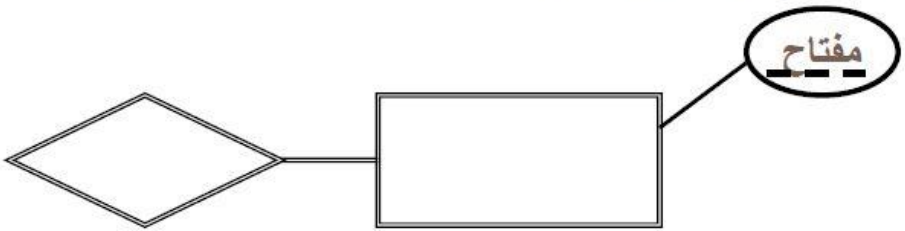
- لكل طالب سجل واحد (نوع العلاقة ١).
- السجل يكون لطالب واحد (نوع العلاقة ١).
- يمكن أن يكون بعض الطلبة ليس لديهم سجلات (اشتراك جزئي).
- كل سجل لابد وأن يكون يتبع طالب معين (اشتراك كلي).

الكيان الضعيف (Weak Entity)

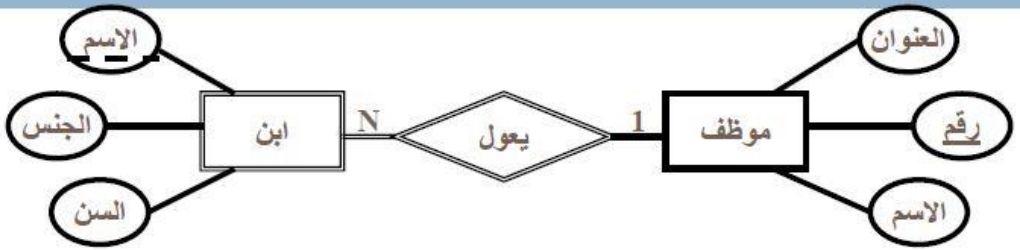
- الكيانات التي لا تحتوي علي صفات تمثل مفتاح لذاتها تسمى كيانات ضعيفة.
- الكيان الضعيف يرتبط بكيان أخر معرف له عن طريق علاقة معرفة لهذا الكيان.
- الكيان الضعيف دائما يرتبط بارتباط كلي مع العلاقة المعرفة له.
- يتم تمثيل الكيان الضعيف باستخدام مستطيل مزدوج الخط كما هو مبين بالشكل التالي:

الكيان الضعيف (Weak Entity)

- يحتوي الكيان الضعيف عادة على مفتاح يسمي المفتاح الجزئي (Partial key) والذي يتكون من مجموعة من الصفات التي تعرف الكيان الضعيف المرتبط بالكيان المعرف له بطريقة وحيدة.
- المفتاح الجزئي يتم تمثيله عن طريق وضح خط متقطع تحت الصفات المكونة له.
- الكيان الضعيف والعلاقة المعرفة له يتم تمثيلهم بخط مزدوج للأشكال المعرفة لهم في نموذج الكينونة/العلاقة.



مثال على الكيان الضعيف



ملاحظات:

- **"ابناء" كيان ضعيف حيث أنه لا يحتوى على مفتاح لذاته (المفتاح الجزئي اسم الابن يمكن ان يتكرر لموظف أخر ولكن لا يتكرر لنفس الموظف).**
- **العلاقة "يعول" هي العلاقة المعرفة للكيان الضعيف "ابناء".**
- **الكيان "موظف" هو الكيان المعرف للكيان الضعيف "ابناء".**
- **الكيان الضعيف "ابناء" يشترك اشتراك كلي مع العلاقة "يعول".**

مثال رقم ١

شركة تجارية لديها مجموعة من الاقسام لتنفيذ اعمال الشركة ولكل قسم (اسم القسم – رقم القسم – هاتف القسم). ولدي الشركة عدد من الموظفين الذين يعملون في الاقسام المختلفة وبياناتهم كالآتي (اسم الموظف – الرقم الوظيفى – العنوان – الراتب). يتم تسجيل تاريخ عمل كل موظف في قسمه في سجل توظيف الموظفون. اقترح نموذج بيانات لتمثيل بيانات هذه الشركة.

الحل

تحديد الكيانات:

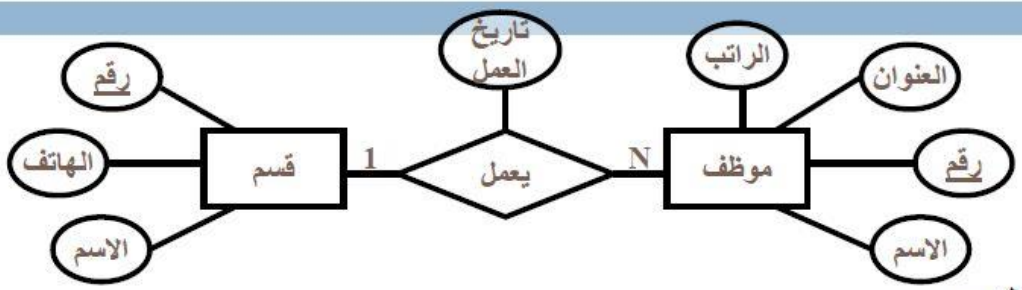
القسم (اسم القسم – رقم القسم – هاتف)

الموظف (اسم الموظف – الرقم الوظيفى – العنوان – الراتب).

تحديد العلاقات:

علاقة عمل الموظفون في الاقسام.

مثال رقم ١



ملاحظات:

- الموظف يعمل في قسم واحد والقسم به عدة موظفون (1:N).
- يمكن أن يكون بعض الموظفون ليس لديهم اقسام (اشتراك جزئي).
- يمكن أن يكون بعض الأقسام ليس لديهم موظفون في (اشتراك جزئي).
- "تاريخ العمل" هي صفة للعلاقة "يعمل" ولذلك اضيفت لها.

مثال رقم ٢

جامعة تقوم بتدريس مجموعة من المقررات الدراسية للطلبة وقررت تسجيل البيانات الخاصة بالطلبة والمقررات وعملية تسجيل الطلبة للمقررات. كل مقرر له البيانات التالية: اسم المقرر – رقم المقرر – عدد الساعات. وبيانات الطالب المطلوبة هي: اسم الطالب – الرقم الجامعى – العنوان. يقوم مسجل الكلية بتدوين العام الجامعي والفصل الدراسي ورقم الشعبة عند تسجيل الطالب لأي مقرر.

اقترح نموذج بيانات لتمثيل بيانات هذه الجامعة.

الحل

تحديد الكيانات:

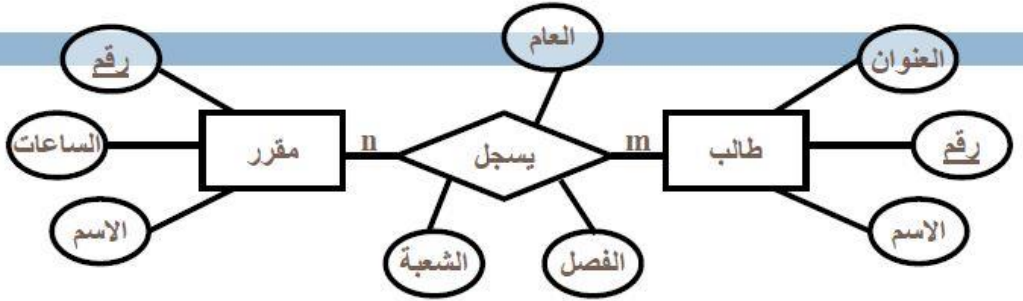
الطالب (اسم الطالب – الرقم الجامعى – العنوان)

المقرر (اسم المقرر – رقم المقرر – عدد الساعات).

تحديد العلاقات:

علاقة تسجيل الطالب لمقرر.

مثال رقم ٢



ملاحظات:

- الطالب يمكن أن يسجل مجموعة من المقررات (نوع العلاقة M).
- المقرر يسجله مجموعة من الطلبة (نوع العلاقة N).
- يمكن أن يكون بعض الطلبة ليس لديهم مقررات (اشتراك جزئي).
- يمكن أن يكون بعض المقررات غير مسجل فيها طلبة (اشتراك جزئي).
- "العام، الفصل، الشعبة" هي صفات للعلاقة "يسجل" ولذلك اضيفت لها.

مثال رقم ٣

23

مركز صيانة أجهزة كهربية يقوم باستقبال اجهزة العملاء وتسجيلها في سجل الإصلاحات. ثم يقوم بتحديد الفني الذي يقوم باصلاح الجهاز وتسجيل قطع الغيار المطلوبة للجهاز والتي طلبها الفني من المخازن. فإذا كانت بيانات الجهاز هي رقمه المسلسل و اسم الصنف وبيانات العملاء هي رقمه واسمه ورقم الهاتف. والبيانات الخاصة بالفني هي رقمه واسمه وتخصصه وبيانات قطع الغيار هي رقم القطعة و سعرها ويتم تسجيل تاريخ دخول الجهاز لمركز الصيانة في سجل الإصلاحات و يتم تسجيل عدد قطع الغيار في سجل احتياجات الأجهزة.

اقترح نموذج بيانات لتمثيل بيانات مركز الصيانة.

الحل

تحديد الكيانات:

العميل (اسم العميل – رقم العميل – الهاتف) - الجهاز (رقم الجهاز – اسم الصنف)

الفني (رقم الفني – اسم الفني – التخصص) - قطعة الغيار (رقم القطعة – السعر)

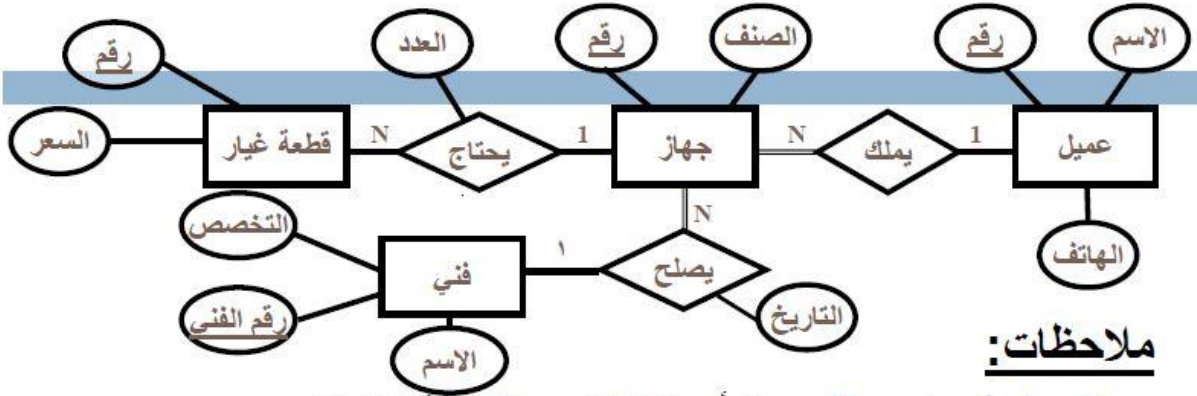
تحديد العلاقات:

١-علاقة ملكية عميل لجهاز.

٢-علاقة اصلاح الفني للجهاز.

٣- علاقة احتياج الجهاز لقطع الغيار.

مثال رقم 3



ملاحظات:

- العميل قد يكون له عدة أجهزة (نوع العلاقة 1:N).
- الفني يمكن أن يصلح عدة أجهزة (نوع العلاقة 1:N).
- الجهاز قد يحتاج لعدة قطع غيار (نوع العلاقة 1:N).
- الجهاز لابد وأن يتبع عميل وأن يتم تعين فني لإصلاحه (اشتراك كلي).
- "التاريخ" هي صفة للعلاقة "يصلح" ولذلك اضيفت لها.

مثال رقم ٤

مركز طبي يقوم باستقبال المرضي ويقوم الطبيب المختص بالكشف علي المريض ومن الممكن أن يطلب الطبيب بعض التحاليل للمريض. بيانات الاطباء هي الاسم والرقم الوظيفي والتخصص بينما يتم تسجيل هذه البيانات للمريض: الاسم ورقم المريض، وعند كل زيارة يتم تسجيل تاريخ الزيارة. وبيانات التحليل هي اسم التحليل ورقمه وتكلفته وأيضا يتم تسجيل تاريخ إجراء التحليل. ويحتوي المركز على أماكن للتنويم ويتم تسجيل تاريخ الدخول والخروج ومسلسل التسجيل ورقم الغرفة.

اقترح نموذج بيانات لتمثيل بيانات المركز الطبي.

الحل

تحديد الكيانات:

الطبيب (اسم الطبيب – الرقم الوظيفي – التخصص) - المريض (رقم المريض – الاسم).

التحليل (رقم التحليل – الاسم - السعر)

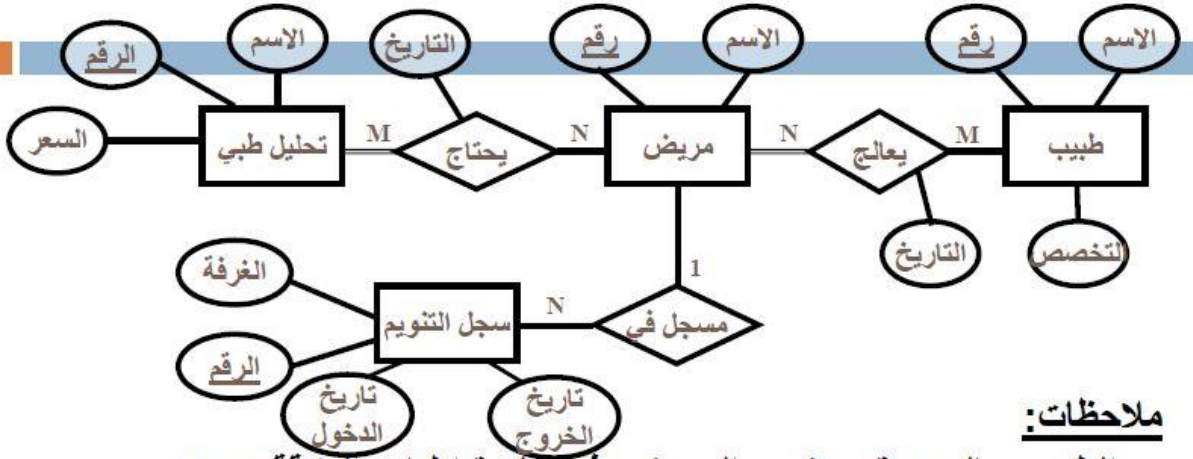
سجل التنويم (رقم التسجيل - تاريخ الدخول – تاريخ الخروج – رقم الغرفة)

تحديد العلاقات:

١-علاقة يعالج (M:N). ٢-علاقة يحتاج تحاليل (M:N).

٣- علاقة مسجل في سجل التنويم (1:N).

مثال رقم 4



ملاحظات:

- **الطبيب يعالج عدة مرضى والمريض يذهب لعدة اطباء (العلاقة M:N).**
- **المريض قد يحتاج عدة تحاليل والتحليل يحتاجه أي عدد من المرضى (العلاقة M:N).**
- **المريض يمكن أن يسجل عدة مرات في سجل التنويم (نوع العلاقة 1:N).**

مقدمة لقواعد البيانات

المحاضرة الخامسة

تحويل نموذج الكينونة/العلاقة إلى قاعدة بيانات علائقية

RELATIONAL DB DESIGN

ER-TO-RELATIONAL MAPPING

Relational DB Design
ER-to-Relational mapping
(التحويل)

عملية التحويل هي عبارة عن خطوات محددة يتبعها مصمم قواعد البيانات لتحويل نموذج الكينونة/العلاقة الي مخطط قواعد البيانات العلائقية.

(Mapping ERD to relational schema)

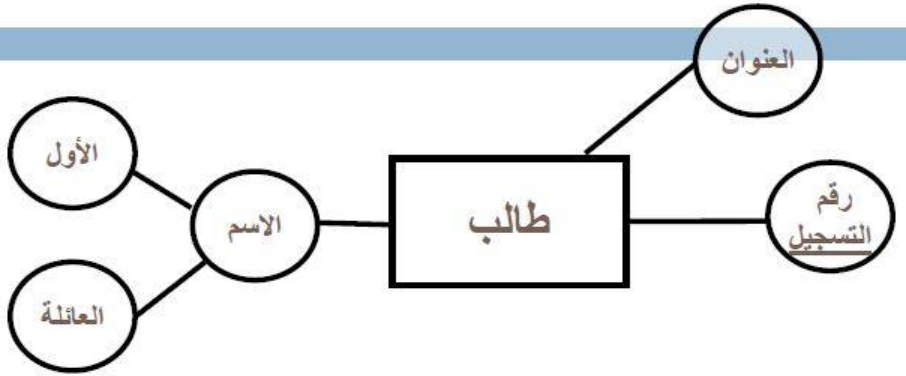
تعتمد عملية التحويل علي فهم المصمم للمبادئ التي تم عرضها ومناقشتها سابقاً والخاصة بالمفتاح الأساسي و المفتاح الأجنبى و أنواع الكيانات والعلاقات وأنواع الارتباطات بين العلاقات.

التحويل – الخطوة الأولى

١. **لكل كيان عادى (ك) فى شكل الكينونة/العلاقة يتم عمل الآتى:**

- إنشاء علاقة (ر) تحتوي علي جميع الصفات البسيطة الموجودة في الكيان (ك).
- الصفات المركبة الموجودة في (ك) يتم تمثيلها في العلاقة (ر) عن طريق تمثيل صفاتها البسيطة التي تكون الصفة المركبة (أي لا تظهر الصفات المركبة للكيان ك في العلاقة ر).
- مفتاح العلاقة (ر) هو مفتاح الكيان (ك) أو أحد المفاتيح المؤهلة في حالة وجود أكثر من مفتاح للكيان (ك).

الخطوة الأولي: مثال



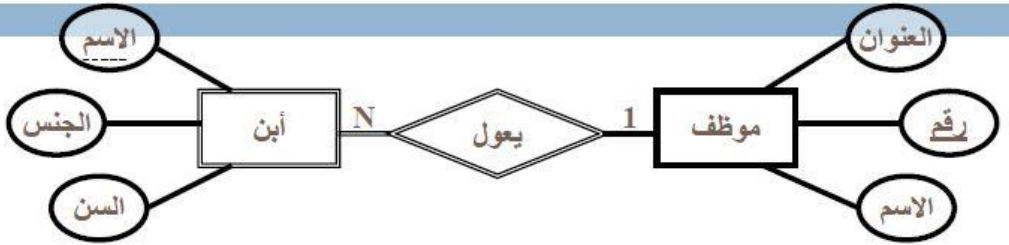
طالب (رقم التسجيل – العنوان – الاسم الأول – اسم العائلة)

التحويل – الخطوة الثانية

٢. **لكل كيان ضعيف (ض) في شكل الكينونة/العلاقة يتم عمل الآتي:**

- إنشاء علاقة (ر) تحتوي علي جميع الصفات البسيطة الموجودة في الكيان (ض).
- الصفات المركبة الموجودة في (ك) يتم تمثيلها في العلاقة (ر) عن طريق تمثيل صفاتها البسيطة التي تكون الصفة المركبة كما تم في الخطوة الأولي.
- إضافة المفتاح الأساسي للكيان المرتبط مع الكيان (ض) والمعرف له الي العلاقة (ر) مع اعتباره مفتاح أجنبي يربط (ض) مع الكيان المعرف له.
- مفتاح العلاقة (ر) هو المفتاح الجزئي للكيان (ض) + المفتاح الأساسي للكيان المرتبط مع الكيان (ض) والمعرف له (الذي تم إضافته كمفتاح أجنبي للكيان ض).

الخطوة الثانية: مثال



موظف (رقم الموظف – العنوان – الاسم)

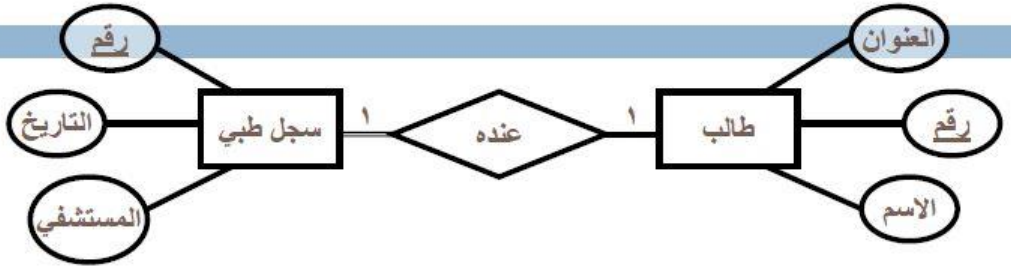
أبن (الجنس - السن - الاسم - رقم الموظف)

التحويل – الخطوة الثالثة

٣. **لكل علاقة (ر) من النوع ١:١ في شكل الكينونة/العلاقة يتم عمل الآتي:**

- تحديد الكيانان س، ت المرتبطان معاً عن طريق العلاقة (ر).
- اختيار واحدة من العلاقتين (س ، ت) وإضافة المفتاح الأساسي للعلاقة الأخرى كمفتاح أجنبي في العلاقة التي تم اختيارها.
- من الأفضل اختيار الكيان المرتبط ارتباطاً كلياً بالعلاقة (ر) ليتم ضم المفتاح الأساسي للعلاقة الأخرى إليه كمفتاح أجنبي.
- إذا كان الكيانان مرتبطان ارتباطا كلياً بالعلاقة (ر) فيمكن اختيار أي كيان لضم المفتاح الأساسي للكيان الأخر إليه كمفتاح أجنبي وإن كان من الممكن دمج الكيانان س،ت معاً ليصبحا كياناً واحداً.

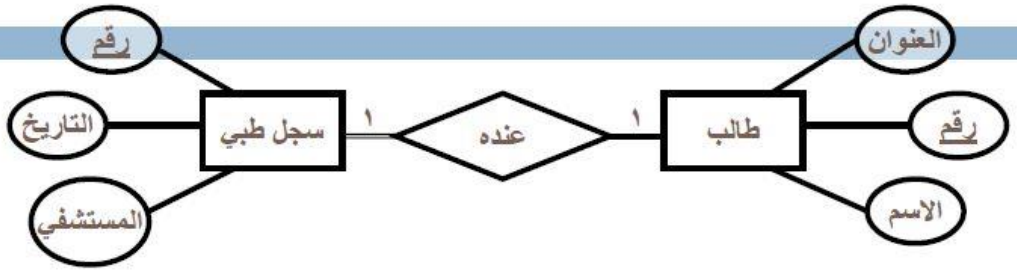
الخطوة الثالثة: مثال (الحل الأمثل)



طالب (الرقم الجامعي – العنوان – الاسم)

سجل طبي (رقم السجل – التاريخ – المستشفي- الرقم الجامعي)

الخطوة الثالثة: مثال (حل أخر ولكنه غير أمثل)

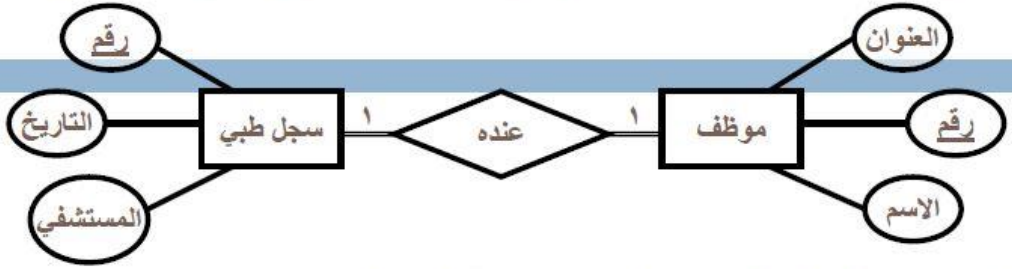


طالب (رقم التسجيل – العنوان – الاسم – رقم السجل الطبي)

سجل طبي (رقم السجل الطبي – التاريخ – المستشفي)

(حقل رقم السجل الطبي سيكون فارغ للطالب الذي ليس له سجل طبي و هذا من الممكن أن يتكرر لكثير من الطلبة)

الخطوة الثالثة: مثال (ارتباط كلي من الطرفين)



موظف (رقم الموظف – العنوان – الاسم)

سجل طبي (رقم السجل الطبي – التاريخ – المستشفي - رقم الموظف)

أو

موظف (رقم الموظف – العنوان – الاسم - رقم السجل الطبي)

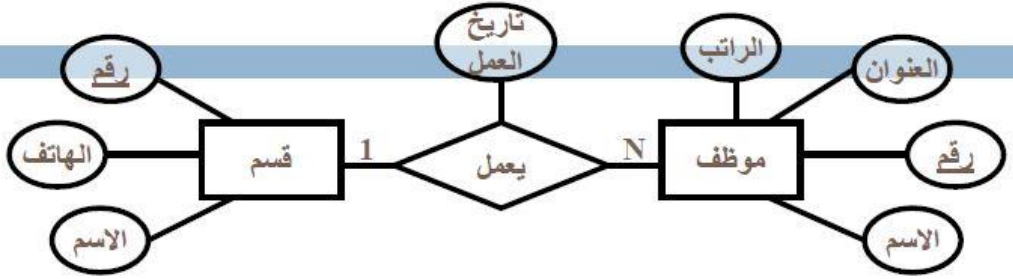
سجل طبي (رقم السجل الطبي – التاريخ – المستشفي)

التحويل – الخطوة الرابعة

٤. **لكل علاقة ثنائية عادية (ر) من النوع 1:N نقوم بعمل الآتي:**

- نحدد العلاقة س والتي تمثل الكيان الموجود عند الجانب **N** في العلاقة (ر).
- إضافة المفتاح الأساسي للكيان الأخر المرتبط بالعلاقة (ر) كمفتاح أجنبي في العلاقة س.
- نضيف أية صفات موجودة علي العلاقة (ر) للعلاقة س.

الخطوة الرابعة: مثال



موظف (رقم الموظف – الاسم – العنوان – الراتب – رقم القسم - تاريخ العمل)

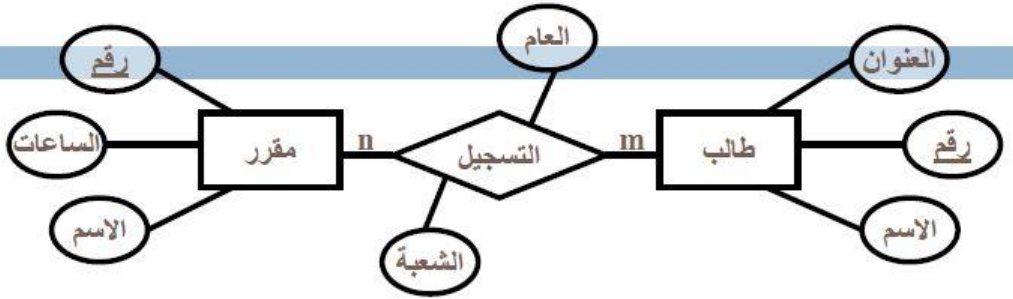
قسم (رقم القسم – اسم القسم – رقم الهاتف)

التحويل – الخطوة الخامسة

٥. **لكل علاقة ثنائية عادية (ر) من النوع M:N نقوم بعمل الآتى:**

- **ننشأ علاقة جديدة س تمثل العلاقة (ر).**
- **إضافة المفاتيح الأساسية للكيانين المرتبطين بالعلاقة (ر) كمفاتيح أجنبية في العلاقة س.**
- **نضيف أية صفات موجودة علي العلاقة (ر) للعلاقة س.**
- **المفتاح الأساسي للعلاقة س هو مجموعة المفاتيح الأجنبية التي تم ضمها إلي س وتمثل المفاتيح الأساسية للكيانين المرتبطين بالعلاقة (ر).**

الخطوة الخامسة: مثال

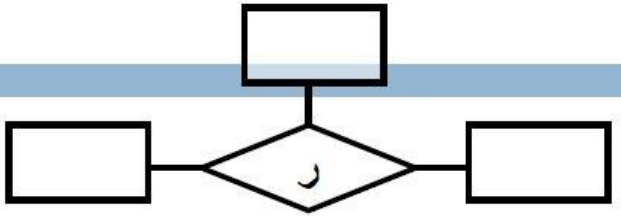


طالب (رقم الطالب – اسم الطالب – العنوان)

مقرر (رقم المقرر – اسم المقرر – عدد الساعات)

التسجيل (رقم الطالب - رقم المقرر – العام – الشعبة)

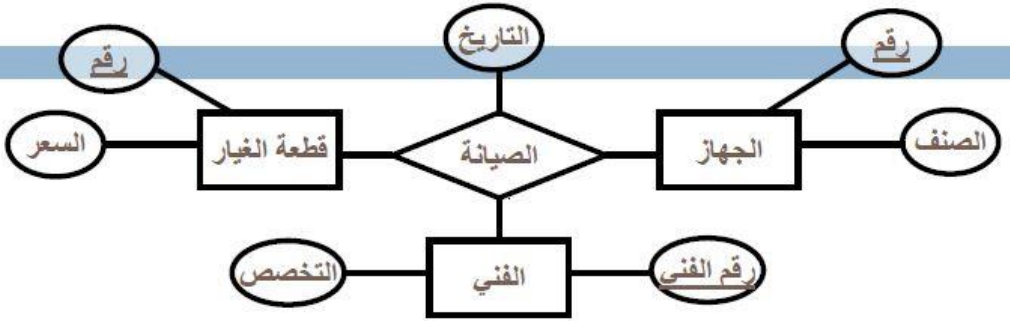
التحويل – الخطوة السادسة



٦. **لكل علاقة (ر) من الدرجة (N) (أى ثلاثية أو أعلى) نقوم بعمل الآتي:**

- **ننشأ علاقة جديدة (س) لهذه العلاقة (ر).**
- **إضافة المفاتيح الأساسية للكيانات المرتبطة بالعلاقة (ر) كمفاتيح أجنبية في العلاقة (س).**
- **نضيف أية صفات موجودة علي العلاقة (ر) للعلاقة س.**
- **المفتاح الأساسي للعلاقة (س) هو مجموعة المفاتيح الأجنبية التي تم ضمها إلي (س) وتمثل المفاتيح الأساسية للكيانات المرتبطة بالعلاقة (ر).**

الخطوة السادسة: مثال



الجهاز (رقم الجهاز – الصنف)

الفني (رقم الفني – التخصص)

قطعة الغيار (رقم القطعة – السعر)

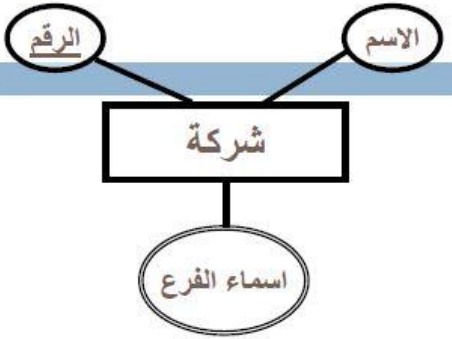
الصيانة (رقم الجهاز - رقم الفني – رقم القطعة – التاريخ)

التحويل – الخطوة السابعة

٧. **لكل صفة A متعددة القيم (multivalued) يتم عمل الأتي:**

- **ننشأ علاقة جديدة (ر) لهذه الصفة A.**
- **إضافة الصفة A إلي العلاقة (ر) كصفة لها.**
- **إضافة المفتاح الأساسي للكيان الذي يحتوي علي الصفة A كمفتاح أجنبي في العلاقة (ر).**
- **المفتاح الأساسي للعلاقة (ر) هو المفتاح الأجنبي الذي تم ضمه إلي (ر) بالإضافة إلي الصفة A.**

الخطوة السابعة: مثال



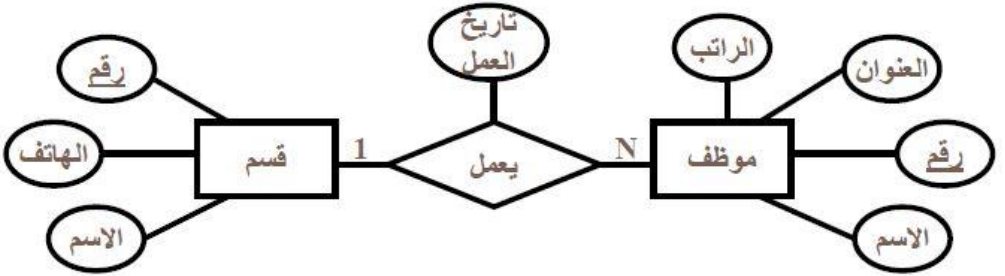
شركة (رقم الشركة – الاسم)

الفرع (اسم الفرع – رقم الشركة)

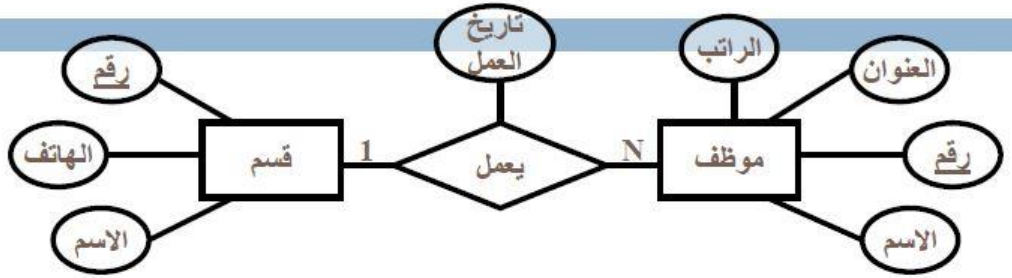
مثال رقم ١

الشكل المعطي يمثل نموذج كينونة/علاقة (ER) لتمثيل بيانات موظفين وأقسامهم في شركة.

المطلوب هو تحويل الشكل إلي النموذج العلاقي المكافئ له.



حل المثال رقم ١



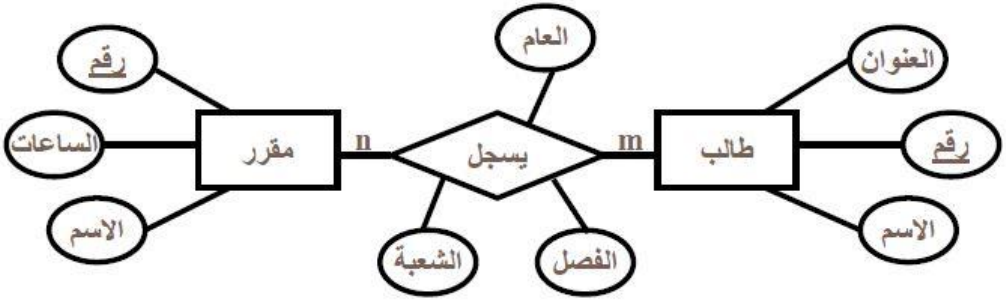
قسم (رقم القسم – الاسم – الهاتف)

موظف (رقم الموظف – الاسم – العنوان - الراتب - رقم القسم - تاريخ العمل)

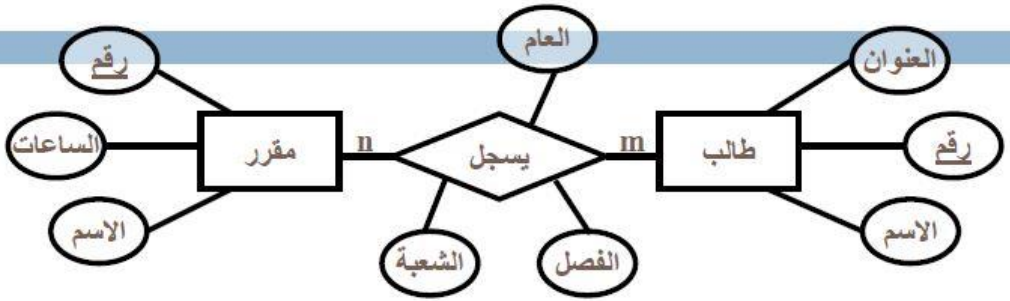
مثال رقم ٢

الشكل المعطي يمثل نموذج كينونة/علاقة (ER) لتمثيل بيانات تسجيل الطلبة لمقررات في جامعة.

المطلوب هو تحويل الشكل إلي النموذج العلاقي المكافئ له.



حل المثال رقم ٢



طالب (الرقم الجامعي – الاسم – العنوان)

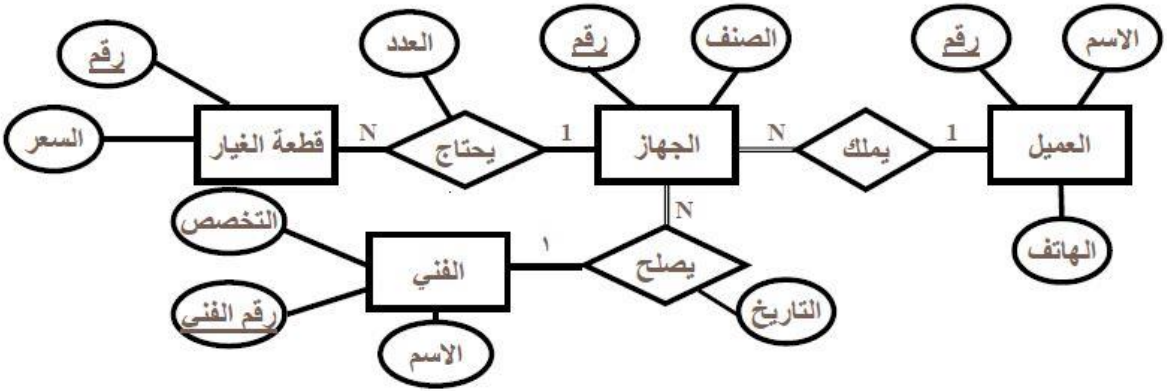
مقرر (رقم المقرر – الاسم – الساعات)

التسجيل (الرقم الجامعي - رقم المقرر – العام – الفصل – الشعبة)

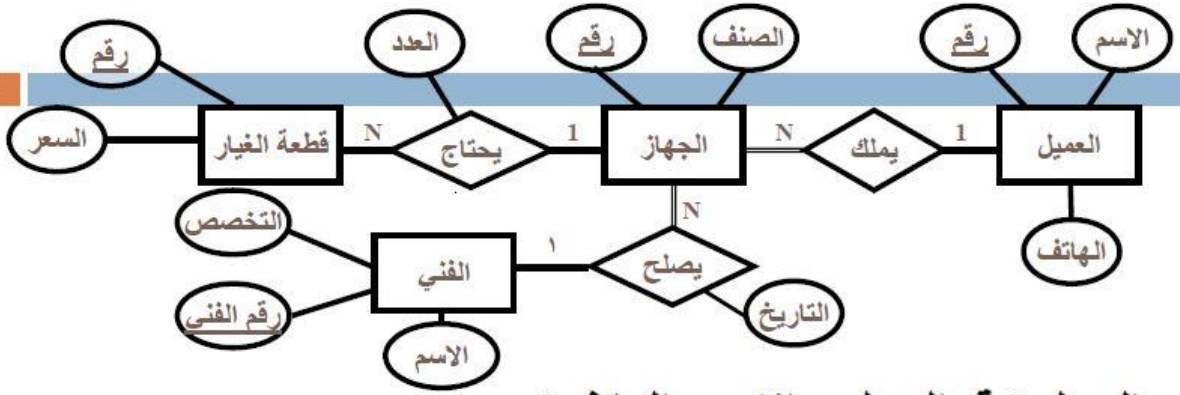
مثال رقم 3

الشكل المعطي يمثل نموذج كينونة/علاقة (ER) لتمثيل بيانات اصلاح أجهزة في مركز صيانة.

المطلوب هو تحويل الشكل إلي النموذج العلاقي المكافئ له.



حل المثال رقم 3



العميل (رقم العميل - الاسم - الهاتف)

الجهاز (رقم الجهاز - الصنف – رقم العميل – رقم الفني – تاريخ الإصلاح)

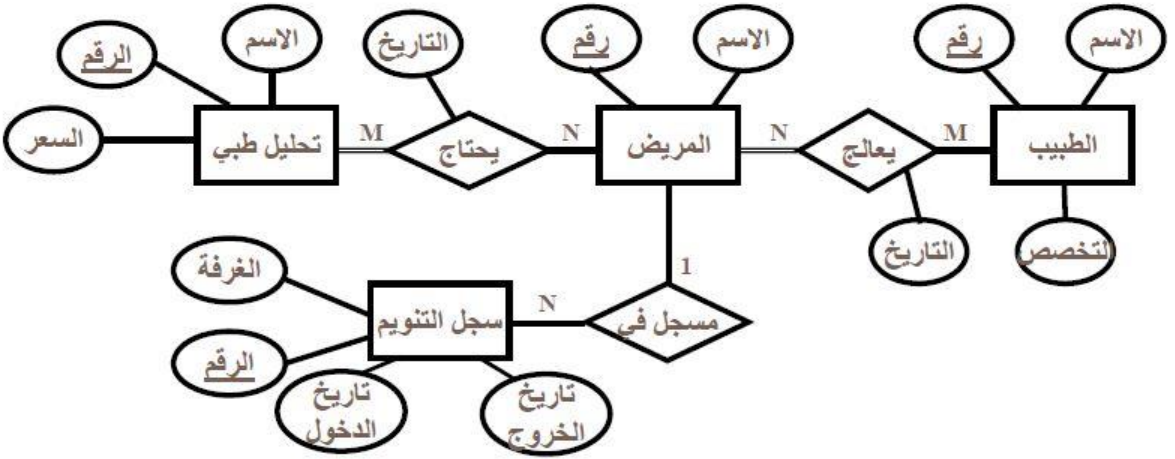
قطعة الغيار (رقم القطعة - السعر – رقم الجهاز – العدد)

الفني (رقم الفني – الاسم – التخصص)

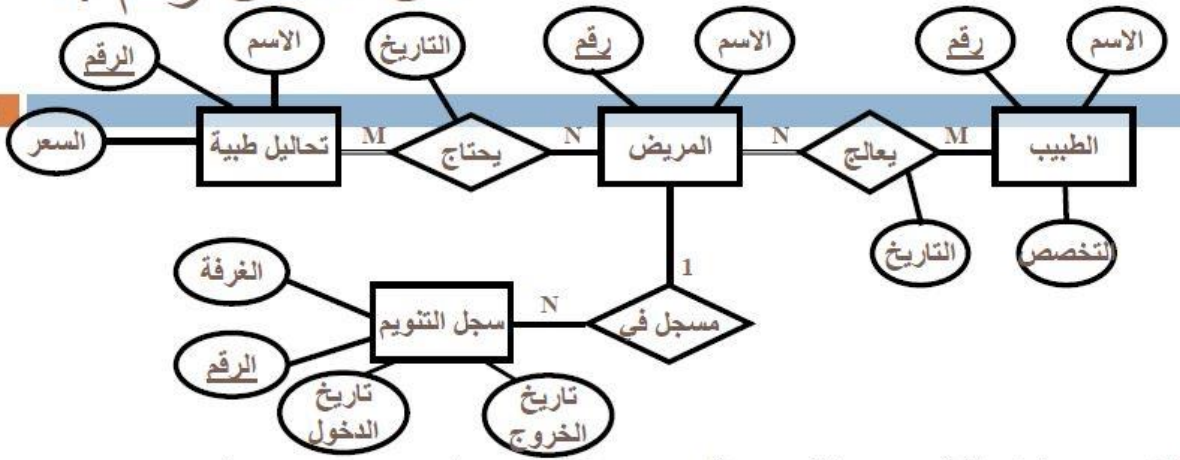
مثال رقم 4

الشكل المعطي يمثل نموذج كينونة/علاقة (ER) لتمثيل بيانات تسجيل المرضى في مركز طبي.

المطلوب هو تحويل الشكل إلي النموذج العلاقي المكافئ له.



حل المثال رقم 4



الطبيب ≡ (رقم الطبيب – الاسم – التخصص)

المريض ≡ (رقم المريض – الاسم)

يعالج ≡ (رقم الطبيب – رقم المريض – التاريخ)

التحليل الطبي ≡ (رقم التحليل – الاسم – السعر)

يحتاج ≡ (رقم التحليل – رقم المريض – تاريخ التحليل)

سجل التنويم ≡ (رقم التسجيل – تاريخ الدخول– تاريخ الخروج – الغرفة – رقم المريض)

مقدمة لقواعد البيانات

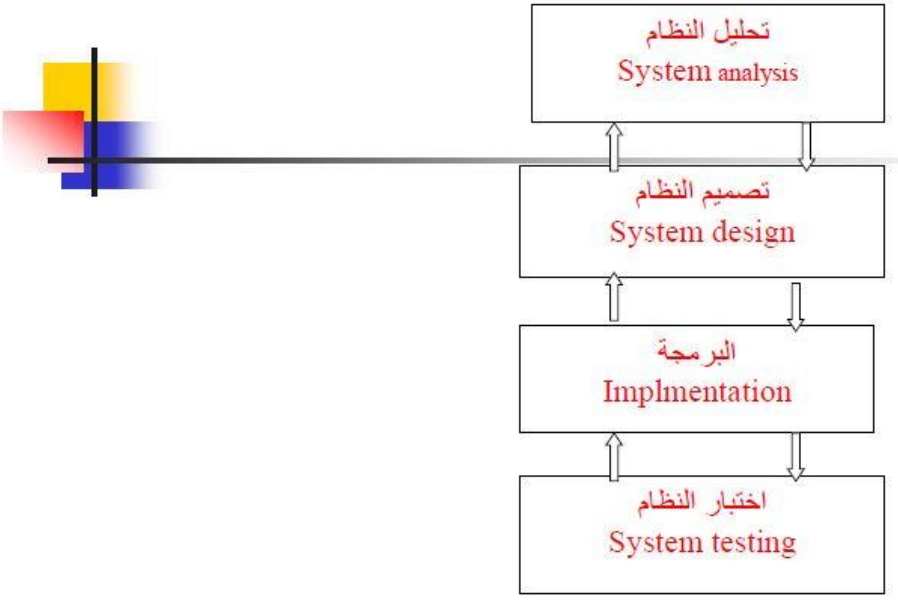
المحاضرة السادسة

دورة حياة نظم المعلومات

INFORMATION SYSTEMS LIFE CYCLE

دورة حياة نظم المعلومات

- كثير ما يعاني نظم المعلومات من مشاكل مستمرة تتطلب إما تطوير النظام أو إحلاله بنظام جديد وهذا الإحلال أو التطوير يتم بعد إجراء حلقه متصلة من الدراسات تشمل عدة مراحل. كما في الشكل التالي: -


تحليل النظام
System analysis
تصميم النظام
System design
البرمجة
Implmentation
اختبار النظام
System testing

تحليل النظام system analysis

يتم تحليل النظام من خلال مرحلتين

- اولاً:- تعريف المشكلة **Problem Definition**

يجب على المحلل أن يفهم المشكلة فهما عميقا ويجب أن تكون واضحة أمامه تماما وان يصنف هذه المشاكل إلى أنواع مشاكل معروفه وهي كتالي:

▪ مشكلة الثقة The Problem Of Reliability

- من خصائص المشكلة والتي توجد في النظام الحالي المؤسس أن الإجراءات تعمل بعض الوقت ولا تستطيع أن تعمل طول الوقت وكذلك فان نفس الإجراء يعطي نتائج مختلفة من وقت لأخر و بالتالي يسبب عدم الثقة في إجراءات النظام الحالي .

▪ مشكلة صحة النتائج The Problem Of Validity

- هذه المشكلة تكمن في أن النظام يخرج أحيانا نتائج كثيرة غير واضحة وغير صحيحة وبالتالي تسبب اضطرابات للمستخدمين.

- **The Problem Of Accuracy** *مشكلة الدقة*
- هذه المشكلة تكون معظمها معتمده على صحة البيانات فان صحت البيانات صحت نتائج النظام وان خطأت أصبح النظام غير الدقيق.
- **The Problem Of Capacity** *مشكلة القدرة أو الاستيعاب*
  - قدرة التشغيل غير كافيه
  - قدرة التخزين غير كافيه
  - قدرة العاملين غير كافيه.
- **The Problem Of Throughput** *مشكلة الانجازأو الإنتاج*
- هذه المشكلة ترتبط بإنتاجيه وقدرة النظام فإذا كان النظام قدرته عاليه و الانتاجيه منخفضة فهذا معناه وجود

ثانياً: ـ تحليل متطلبات النظام System requirements analysis

- في هذه الخطوة يقوم محلل النظام system analyst بالاجتماع مع مستخدم لنظام(user) للتعرف على مايحتاجه المستخدم من تعديل النظام القديم أو إحلاله بنظام أخر.

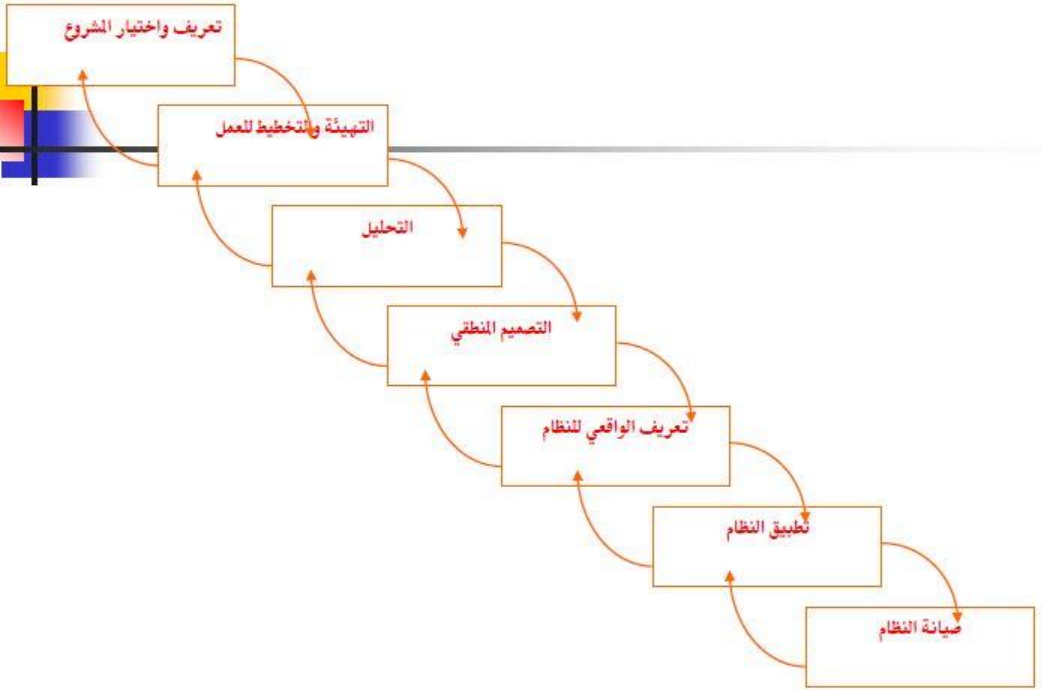
Developing Information systems And The system Developing live cycle

- **منهجية تطوير النظم:**

عملية رتيبة تتبع في المنظمة وذلك لتوصيل جميع الخطوات الضرورية على التحليل والتصميم ثم البرمجة وصيانة نظام المعلومات.

دورة حياة تطوير النظام SDLC

- وهي عبارة عن منهجية تقليدية تستخدم لتطوير وصيانة واستبدال أنظمة المعلومات.
وهذه الدورة مكونة من سبع مراحل يمر بها النظام من بداية في التفكير في إنشاء مشروع إلى مراحل صيانة المشروع كما سوف نشاهد في الشكل التالي وهذه المراحل متسلسلة كما:


تعريف واختيار المشروع
التهيئة والتخطيط للعمل
التحليل
التصميم المنطقي
تعريف الواقعي للنظام
تطبيق النظام
صيانة النظام

دورة حياة النظام SDLC

١. **تعريف واختيار المشروع**

: Project Identification And Selection

هذه المرحلة تعتبر المرحلة الأولى من دورة حياة النظام وخلال هذه المرحلة يتم التعرف على احتياجات نظام المنظمة وتحليلها و وضع أولويات لها وترتيبها.

الناتج من هذه المرحلة: system service request (SSR)

وهو عبارة عن ملخص المشكلة

دورة حياة النظام SDLC

٢. التهيئة والتخطيط للعمل

: Project Identification And planning

تعتبر المرحلة الثانية من دورة حياة النظام ومن خلالها يتم شرح نظام المعلومات المتوقع تنفيذه..

وهل من الأفضل الاستمرار في إكمال العمل في المشروع (النظام) أو إلغاءه.

تخطيط وتهيئة المشروع ومن ثم نقرر دراسة الجدوى من هذا المشروع واتخاذ قرار المتابعة

دورة حياة النظام SDLC

- أهم الخطوات لدراسة الجدوى للمشروع :

١- هل المشروع مربح اقتصادياً ٢- فنياً

٣- عملياً

٤- هل الوقت كافي لعمل المشروع ٥- قانونياً ٦- سياسياً

وكذلك يتم اعتماد الخطة لعمل الأجزاء المتبقية من **SDLC** في حال اتخاذ القرار بالاستمرار في المشروع وعدم إلغاءه.

الناتج من هذه المرحلة : Plan ,detailed problem definition ,cost/benefit analysis

دورة حياة النظام SDLC

٣. التحليل Analysis:

وهذه هي المرحلة الثالثة من **SDLC** ويتم خلالها دراسة النظام الحالي واقتراح النظم البديلة التي سوف تحل محله.

الناتج من هذه المرحلة : Requirement structure

دورة حياة النظام SDLC

٤. التصميم المنطقي Logical Design:

وهذه هي المرحلة الرابعة من **SDLC** ويتم خلالها وصف لكل وظائف النظام الذي وقع عليه الاختيار من بين الأنظمة البديلة وهذا الوصف لا يكون مرتبط بأي نوع من لغات البرمجة أو تطبيقات الحاسب الآلي.

الناتج من هذه المرحلة Logical Data model :

دورة حياة النظام SDLC

٥. التصميم الواقعي للنظام Physical Design:

- وهذه هي الخطوة الخامسة من **SDLC** وفيه يتم تحويل التصميم المنطقي إلى تفاصيل ترتبط بالتقنية التي سوف تتبع (لغات البرمجة وتطبيقات الحاسب) وخلال هذه المرحلة يمكن إنجاز كل مهام البرمجة وتطبيقات الحاسب.
- الناتج من هذه المرحلة :

Physical Data model

دورة حياة النظام SDLC

٦. تطبيق النظام implementation:

- وهذه هي الخطوة السادسة من **SDLC** وفيها نبدأ بكتابة الـ **code** للنظام بعد اختبار لغة البرمجة في التصميم الفعلي للنظام وبعد الانتهاء من كتابة الـ **code** نقوم بعملية اختبار النظم وذلك بإدخال بيانات حقيقية لمعرفة نقاط الضعف في النظام ومن ثم تعديلها.
- ثم نقوم بعملية تحميل **-installation-** للنظام على أجهزة المنظمة ، وبعد ذلك نقوم بعملية الدعم الفني للمستخدمين.
- الناتج من هذه المرحلة : Source Code

دورة حياة النظام SDLC

٧. صيانة النظام Maintenance:

وهذه هي الخطوة السابعة والأخيرة من **SDLC** وفيها يتم إجراء التعديلات على النظم بناءً على طلب المستخدمين.

أنواع دورة حياة تطوير النظم
Type of System Developing Live Cycle

١- دورة حياة تطوير النظام التقليدية Traditional SDLC:

عيوب الطريقة التقليدية:

- من الصعب العودة إلى مرحلة سابقة بمجرد الشروع في المرحلة التي تليها.
- يعتبر وقت التحليل والتصميم قليلاً مقارنة بالوقت المحدد للبرمجة.

أنواع دورة حياة تطوير النظم
Type of System Developing Live Cycle

٢. التحليل المركب والتصميم البنائي
Structured Analysis And Structure Design

* يستخدم تقنيات **DFD** الهدف منها هو تقليل الوقت والجهد.
* من السهل العودة إلى المرحلة السابقة في حال تغير المتطلبات.
* يتم فيه تقسيم العملية (المشكلة) إلى وحدات اصغر يكون من السهل إدارتها والسيطرة عليها.

أنواع دورة حياة تطوير النظم
Type of System Developing Live Cycle

٣.التحليل والتصميم الشيئي Object Oriented Approach:

* مبني على أساس الأهداف العامة للمنطقة وليس على نوع البيانات أو طرف المعالجة.

* **يتميز بثلاث سمات هي:**

١- الوراثة **Inheritance**.

٢- التغليف أو الاحتواء **Encapsulate**.

٣- تعدد الأشكال **Polymorphism**

الصيغ المعيارية Normal Forms

- **Normalization** is the process of a clearance the database of inappropriate reputation of data by depending on the inference rules and functional dependency, although the process of developing the database design in the normal forms is a key building block for a state of art design of the database.

- **الصيغة المعيارية:** هي عملية تخليص قاعدة البيانات من التكرار غير المناسب للبيانات بالاعتماد على قوانين الاستنتاج والاعتمادية الوظيفية.
- وإن عملية وضع تصميم قاعدة البيانات في الصيغة المعيارية يشكل لبنة أساسية في عملية التصميم الجيد لقاعدة البيانات، وتتم هذه العملية على عدة مراحل مثل:

- الصيغة المعيارية الأولى 1NF First Normal Form
- الصيغة المعيارية الثانية 2NF Second Normal Form
- الصيغة المعيارية الثالثة 3NF Third Normal Form

مشاكل تكرار البيانات
(Data Anomalies)

- نلاحظ في الجدول التالي أن معلومات الموظف والقسم الذي يعمل فيه موجودة في جدول واحد، ونتيجة لذلك فأن تكرار بعض البيانات مثل اسم وموقع القسم في كل سجل سوف يؤدي إلى عدة مشاكل.

Employee_department

Empno	**Ename**	**Job**	**Salary**	**Deptno**	**Dname**	**Loc**
10	Smer	Clerk	300	1	Accounting	Baghdad
20	Ali	Manager	500	1	Accounting	Baghdad
30	Khalid	Salesman	400	2	Sales	Mosul
40	Saeed	Salesman	450	2	Sales	Mosul
50	Salem	Clerk	350	3	Operation	Basrah

مشاكل تكرار البيانات
(Data Anomalies)

1. **مشكلة الإضافة Insertion Anomaly :** حيث أننا لا نستطيع أن نضيف قسماً جديداً إلا إذا كان القسم يحتوي على موظف؛ لأن المفتاح الرئيسي للجدول هو رقم الموظف.
2. **مشكلة التعديل Updating Anomaly :** حيث إذا قمنا بتعديل موقع (loc) القسم فلا بد من إجراء عملية التعديل لجميع الموظفين في هذا القسم؛ وإلا ستؤدي هذه العملية إلي عدم توافقية البيانات، أي نفس رقم القسم ولكن أكثر من موقع، وكذلك إذا تمت عملية تغيير موقع القسم عند أي موظف عن طريق الخطأ، فإنه لو قمنا بعملية استرجاع لجميع الموظفين الذين يعملون في هذا القسم فإن هذا الموظف لن يظهر بين الموظفين.
3. **مشكلة الحذف Deletion Anomaly:** إن القسم 3 يحتوي علي موظف واحد فقط، ولو قمنا بحذف هذا الموظف فإن معلومات القسم 3 سوف تختفي من الجدول.

الاعتمادية الوظيفية
Functional Dependency FD

- Functional Dependency is the adoption of one attributes on the value of another attribute(s).Where through the identification of this dependency we will be able to determine where it should be placed, and this consequently leads to put the data in the right place and get rid of the data replication.

- وهي اعتماد قيمة إحدى صفات الكيان على قيمة صفة (صفات) أخرى، ويرمز لها بالرمز (⟶).
- حيث من خلال تحديد هذه الاعتمادية سوف نستطيع أن نحدد المكان الذي يجب أن توضع فيه الصفة.
- وهذا بالتالي يؤدي إلى وضع البيانات في المكان الصحيح والتخلص من عملية تكرار البيانات.

- مثال : A ⟶ B
- يعني أن B تعتمد اعتماداً وظيفياً على A، ونستطيع أن نقول هنا أن قيمة A تحدد قيمة B. وأن A ترجع لنا قيمة واحدة فقط لـ B.
- ويمكن ايضاً أن نقول أننا نستطيع أن نصل إلى البيانات في العمود B بمعرفتنا للبيانات في العمود A.

الاعتمادية الوظيفية
Functional Dependency FD

- وهي اعتماد قيمة إحدى صفات الكيان على قيمة صفة (صفات) أخرى، ويرمز لها بالرمز (⟶).
- حيث من خلال تحديد هذه الاعتمادية سوف نستطيع أن نحدد المكان الذي يجب أن توضع فيه الصفة.
- وهذا بالتالي يؤدي إلى وضع البيانات في المكان الصحيح والتخلص من عملية تكرار البيانات.

- مثال : A ⟶ B
- يعني أن B تعتمد اعتماداً وظيفياً على A، ونستطيع أن نقول هنا أن قيمة A تحدد قيمة B. وأن A ترجع لنا قيمة واحدة فقط لـ B.
- ويمكن ايضاً أن نقول أننا نستطيع أن نصل إلى البيانات في العمود B بمعرفتنا للبيانات في العمود A.

الاعتمادية الوظيفية
Functional Dependency FD

- مثال : لكل موظف اسم واحد فقط ولكل موظف قسم واحد يعمل فيه إذاً:

FD1 : Empno → Ename

FD2 : Empno → Deptno

- أي يمكن معرفة اسم الموظف من رقمه وكذلك القسم.

- ويمكن أن نعيد كتابة هذه الاعتمادية على الشكل التالي:

FD1 : Empno → Ename , Deptno

قواعد الاستنتاج
Inference Rules

- **قواعد الاستنتاج Inference Rules**

- Inference Rules is a set of rules that are used in the process of identifying the functional dependency.

- **<u>قواعد الاستنتاج</u> :** هي عبارة عن مجموعة من القواعد تستخدم في عملية تحديد الاعتمادية الوظيفية وتتلخص هذه القواعد كما يلي:

1- قاعدة الانعكاسية (Reflexive Rule) : إذا كانت y جزء من x (y محتواه في x) فإن xتحدد y

$Y \subseteq X \Rightarrow X : Y$

- مثال:

Empno ⟶ Deptno

<u>ملاحظة</u> : |= تعني أنه إذا تحقق الطرف الأيسر فإننا نستطيع استنتاج الطرف الأيمن.

2- قاعدة الزيادة أو الإضافة (Augmentation Rule) : إذا كانت x تحدد y فإن إضافة z إلي x تعني أنه بالإمكان إضافة z إلى y

$$\{ X \rightarrow Y \} \models XZ \rightarrow YZ$$

- **مثال:**

Empno → Deptno

Empno,Dname → Deptno,Dname

3- قاعدة التعدي (Transitive Rule) : إذا كانت x تحدد y وكانت y تحدد z فإن x تحدد z

$$\{ X \rightarrow Y , Y \rightarrow Z \} \models X \rightarrow Z$$

- **مثال:**

Book → Author

Author → Auth_Addr

Book → Auth_Addr

4- قاعدة الاتحاد (Union Rule) : إذا كانت x تحدد y و x تحدد z إذاً x تحدد yz

$$\{ X \rightarrow Y , X \rightarrow Z \} \models X \rightarrow YZ$$

- **مثال:**

Empno ⟶ Ename

Empno ⟶ Gender

Empno ⟶ Ename,Gender

5- قاعدة التقسيم (Decomposition Rule) : وهي عكس قاعدة الاتحاد، إذا كانت x تحدد yz إذاً x تحدد y و تحدد z

$$\{ X \rightarrow YZ \} \models X \rightarrow Y , X \rightarrow Z$$

- **مثال:**

Empno ⟶ Ename,Gender

Empno ⟶ Ename

Empno ⟶ Gender

العلاقة غير المعيارية
(Unnormalized Form UNF)

- هي العلاقة التي تحتوي على مجموعة مكررة من البيانات، أي وجود أكثر من قيمة بيانية في داخل الخلية، أمثلة:

Customer

Cno	Cname	Address
10	Smer Mohammed Ali	12 abc Mosul
20	Ahmed Ali Salem	343 xyz Baghdad
30	Khalid Saad Ahmed	23 Abc Anbar
40	Noor Ahmed Mohammed	67 xyz Dohok

Employee

Eno	Ename	Proj_code	Hours	Deptno	Dname
210	Ali	P1 P2 P3	12 20 40	10 20 20	Research Operation Operation
201	Salem	P1 P3	30 15	10 20	Research Operation
305	Ali	P2 P3	40 20	20 20	Operation Operation

الصيغة المعيارية الأولى
(First Normal Form 1NF)

- The table will be in the 1NF if all table columns contain simple data (Atomic), that is, any intersection of column with row gives only one value.

- إن الجدول يكون في الصيغة المعيارية الأولى إذا كانت جميع أعمدة الجدول تحتوي على بيانات بسيطة أو مفردة (Atomic غير مركبة) أي إن تقاطع العمود مع الصف يعطي قيمة واحدة فقط.
- مثال : الاسم (يتم تقسيمه إلى الاسم الأول، الاسم الثاني، اسم العائلة)، العنوان (يتم تقسيمه إلى المدينة، الشارع، رقم المنزل) وكل واحد في عمود مستقل.

- أي أن في 1NF هناك قاعدتان أساسيتان:

1- أن لا يكون هناك صفة متعددة القيم : بمعنى أن يكون داخل العمود أكثر من بيان.

2- أن لا يكون هناك صفة مركبة : بمعنى أن يقسم العمود إلى قسمين أو اكثر.

الصيغة المعيارية الأولى
(First Normal Form 1NF)

- مثال 1: في الجدول التالي معلومات الزبون، ونلاحظ أن الاسم يحتوي على ثلاثة قيم بيانية وكذلك العنوان، فبالتالي لا نستطيع أن نخزن قيمة واحدة فقط في عمود الاسم أو العنوان.

Customer

Cno	Cname	Address
10	Smer Mohammed Ali	12 abc Mosul
20	Ahmed Ali Salem	343 xyz Baghdad
30	Khalid Saad Ahmed	23 abc Anbar
40	Noor Ahmed Mohammed	67 xyz Dohok

- ولوضع الجدول في الصيغة المعيارية الأولى 1NF يجب تقسيم الأعمدة المركبة إلى أعمدة بسيطة:

Cno	Fname	Mname	Lname	Sno	Street	City
10	Smer	Mohammed	Ali	12	Abc	Mosul
20	Ahmed	Ali	Salem	343	Xyz	Baghdad
30	Khalid	Saad	Ahmed	23	abc	Anbar
40	Noor	Ahmed	Mohammed	67	xyz	Dohok

- مثال 2: يمثل الجدول التالي سجل ساعات العمل Hours لموظف في عدد من المشاريع Projects والقسم الذي يشرف على تنفيذ المشروع.

Employee

Eno	**Ename**	**Proj_code**	**Hours**	**Deptno**	**Dname**
210	Ali	P1	12	10	Research
		P2	20	20	Operation
		P3	40	20	Operation
201	Salem	P1	30	10	Research
		P3	15	20	Operation
305	Ali	P2	40	20	Operation
		P3	20	20	Operation

- كما هو مبين فأن عدداً من الأعمدة تحتوي على أكثر من قيمة مثل رمز المشروع وساعات العمل والاقسام، ولتحويله يجب أن نقوم بتقسيم الجدول على النحو التالي:

Employee

Eno	**Ename**	**Proj_code**	**Hours**	**Deptno**	**Dname**
210	Ali	P1	12	10	Research
210	Ali	P2	20	20	Operation
210	Ali	P3	40	20	Operation
201	Salem	P1	30	10	Research
201	Salem	P3	15	20	Operation
305	Ali	P2	40	20	Operation
305	Ali	P3	20	20	Operation

- ولكن توجد مشكلة في هذا الجدول وهي ايجاد المفتاح الرئيسي، إذ أصبح رقم الموظف لا يصلح أن يكون مفتاحاً رئيسياً PK، وذلك لأن من شروطه هي عدم التكرار.
- سوف نقوم الأن باستخدام الاعتمادية الوظيفية FD لمحاولة إيجاد المفتاح الرئيسي للجدول:

FD1 : Eno ⟶ Ename

- حيث أن الرقم يحدد الاسم، ولكل موظف رقم واحد. ولا يمكن أن نقول أن الاسم يحدد الرقم لوجود اسماء متشابهة.

FD2 : Proj_code ⟶ Deptno

- حيث أن لكل مشروع قسم واحد يشرف عليه.

FD3 : Deptno ⟶ Dname

- حيث أن لكل قسم اسم واحد.
- أما بالنسبة لبقية العناصر فمثلاً اسم الموظف لا يحدد شيئاً لأنه يوجد أكثر من موظف اسمه Ali فالأسم لا يحدد الرقم، وكذلك فأن Ali يعمل في أكثر من مشروع.
- وكذلك رمز المشروع لا يحدد عدد الساعات ولا الموظفين الذين يعملون فيه، فالمشروع P1 يعمل فيه أكثر من موظف وبساعات مختلفة.
- اما بالنسبة للقسم فلا يحدد الموظفين ولا المشاريع، فمثلاً القسم 20 يشرف على أكثر من مشروع وهذه المشاريع يعمل عليها اكثر من موظف.

- ففي هذه الحالة يجب علينا القيام بمحاولة جديدة لإيجاد المفتاح الرئيسي PK من خلال مفتاح مركب من اكثر من صفة.

- نقوم بربط رقم الموظف مع رمز المشروع:

FD4 : Eno, Proj_code → Ename

FD5 : Eno, Proj_code → Deptno

FD6 : Eno, Proj_code → Hours

FD7 : Deptno → Dname

FD8 : Eno, Proj_code → Ename, Hours, Deptno, Dname

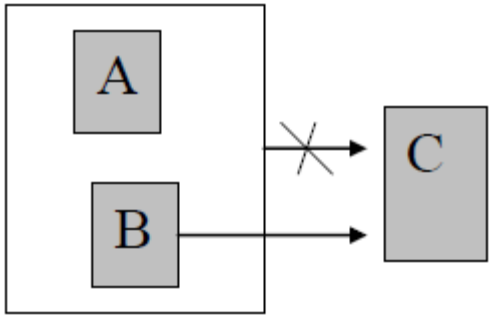
- FD4 و FD5 تنطبق مع FD1 و FD2 حيث ان رقم الموظف وحده يحدد الاسم وكذلك رمز المشروع يحدد القسم.
- اما بالنسبة لـ FD6 فأنها تنطبق لأن رقم الموظف ورمز المشروع يحددان عدد ساعات عمل الموظف في ذلك المشروع.
- وبالتالي نكون قد حصلنا على مفتاح رئيسي لهذا الجدول وكذلك قمنا بوضعه في الصيغة المعيارية الأولى.
- ملاحظة: عندما يكون الجدول بالصيغة غير المعيارية UNFاو 0NF نقوم بتحويل العمود ذو القيم المركبة أو متعددة القيم إلى قيم بسيطة بطريقتين:
- إما بتحويل العمود الواحد إلى عدة أعمدة أو صفوف،
- أو بعمل علاقة جديدة تحوي القيم المتعددة أو المركبة، بالإضافة إلى مفتاح أجنبي يمثل الجدول الأصلي بالطبع.

الصيغة المعيارية الثانية
(Second Normal Form 2NF)

- إن الجدول يكون في الصيغة المعيارية الثانية إذا:

1. كان الجدول في الصيغة المعيارية الأولى.
2. لم يحتوي الجدول علي اعتمادية جزئية.

- **الاعتمادية الجزئية:** هي أن تعتمد بعض الأعمدة (الصفات) اعتماداً وظيفياً على جزء من المفتاح الرئيسي (المركب).



- نلاحظ أن A,B تحدد C أي أن C تعتمد اعتماد وظيفياً على A,B،
- وكذلك أن B تحدد C اي أن C تعتمد اعتماداً وظيفياً على B، وفي هذه الحالة يمكن ان نقول أن هذا الجدول يحتوي على اعتمادية جزئية.

الاعتماد الوظيفي الكامل والجزئي
Full & Partial Dependency

- يوضح الجدول الآتي درجات الطلاب:

Stud

Sno	Cno	Grades
10	1	70
10	2	88
20	1	100
20	2	94

- فمثلا لو أردنا أن نعرف درجة الطالب رقم 10 ستظهر لنا درجتان وهما 70-88.
- وهذا يدل على أن الدرجات لا تعتمد اعتماداً وظيفياً على رقم الطالب فحسب، بل أنها تعتمد أيضاً على رقم المادة أيضاً.
- ويمكن توضيحها كالآتي:

Sno , Cno → Grades

- اذن فالدرجات تعتمد وظيفياً كاملاً على رقم الطالب + رقم المادة.

الاعتماد الوظيفي الكامل والجزئي
Full & Partial Functional Dependency

- مثلاً في حالة أضفنا إلى الجدول اسم الطالب واسم المادة:

Stud

Sno	Cno	Grades	St_name	Course
10	1	70	Ahmed	DB
10	2	88	Ahmed	OS
20	1	100	Salem	DB
20	2	94	Salem	OS

- **مثال:**

Sno , Cno ⟶ St_name

- هنا خطأ لأن رقم الطالب يكفي لنعرف اسمه. وهذه الحالة تسمى اعتماد وظيفي جزئي.
- إذن فإن اسم الطالب يعتمد وظيفياً على رقم الطالب فقط. وهو اعتماد وظيفي جزئي.
- **مثال أخر:**

Sno , Cno ⟶ Course

- هنا خطأ لأن رقم المادة يكفي لنعرف اسمها. وهذه الحالة تسمى اعتماد وظيفي جزئي.

الصيغة المعيارية الثانية
(Second Normal Form 2NF)

- في الصيغة المعيارية الثانية لا ينفع أن يكون هناك صفة تعتمد اعتماد وظيفي جزئي على المفتاح الرئيسي.
- ولكن يجب أن يكون اعتماداً وظيفياً كاملاً كما في حالة الدرجات فهي تعتمد على المفتاح المركب (رقم الطالب + رقم المادة).
- اما في حالة اسم الطالب، اسم المادة فهو اعتماد وظيفي جزئي.
- الحل في هذه الحالة أن نضيف جدول جديد لكل صفة مع المفتاح الرئيسي المعتمدة عليه اعتماداً وظيفياً كاملاً . وتحذف الصفات من الجدول الأساسي.

الصيغة المعيارية الثانية
(Second Normal Form 2NF)

1) ننشأ جدول جديد به المفتاح الرئيسي + الصفة المعتمدة عليه اعتمادية وظيفية كاملة.
2) نحذف الصفة من الجدول الأساسي.

Sno	Cno	Grades	St_name	Course
10	1	70	Ahmed	DB
10	2	88	Ahmed	OS
20	1	100	Salem	DB
20	2	94	Salem	OS

Sno	St_name
10	Ahmed
20	Salem

Sno → St_name

- هنا يعتمد اسم الطالب على رقمه اعتماداً وظيفياً كاملاً.

الصيغة المعيارية الثانية
(Second Normal Form 2NF)

1) ننشأ جدول جديد به المفتاح الرئيسي + الصفة المعتمدة عليه اعتمادية وظيفية كاملة.
2) نحذف الصفة من الجدول الأساسي.

Sno	Cno	Grades	St_name	Course
10	1	70	Ahmed	DB
10	2	88	Ahmed	OS
20	1	100	Salem	DB
20	2	94	Salem	OS

Cno	Course
1	DB
2	OS

Cno → Course

- هنا يعتمد اسم المادة على رقمها اعتماداً وظيفياً كاملاً.

الصيغة المعيارية الثانية
(Second Normal Form 2NF)

- والان هل الجدول التالي هو في الصيغة المعيارية الثانية 2NF؟

Employee

Eno	Ename	Proj_code	Hours	Deptno	Dname
210	Ali	P1	12	10	Research
210	Ali	P2	20	20	Operation
210	Ali	P3	40	20	Operation
201	Salem	P1	30	10	Research
201	Salem	P3	15	20	Operation
305	Ali	P2	40	20	Operation
305	Ali	P3	20	20	Operation

- وللإجابة على ذلك، نجيب على السؤالين التاليين:
- 1- هل الجدول في الصيغة المعيارية الاولى 1NF؟
- نعم، لأنه لا توجد هناك قيم متكررة، كل عمود يحتوي على قيمة واحدة فقط.

2- هل توجد هناك اعتمادية جزئية؟

ولمعرفة ذلك يجب أن نحدد الاعتمادية الوظيفية

FD1 : Eno → Ename

FD2 : Proj_code → Deptno, Dname

FD3 : Eno, Proj_code → Ename, Deptno, Hours

- المفتاح الرئيسي للجدول هو Eno, Proj_code ولكن Eno يحدد Ename إذاً هناك اعتمادية جزئية.
- وكذلك Proj_code يحدد Deptno, Dname وهذه اعتمادية جزئية أخرى.

- وللتخلص من هذه المشكلة يجب أن نقوم بتقسيم الجدول إلى جداول بحيث يضم كل منها الجزء من المفتاح والأعمدة التي تعتمد عليه ونبقي فقط المفتاح المركب مع الاعمدة التي تعتمد عليه.

1- نقوم بنقل اسم ورقم الموظف إلى جدول جديد ونبقي نسخة من رقم الموظف في الجدول الأصلي (لأنه جزء من المفتاح الرئيسي).

2- نقوم بنقل رمز المشروع ورقم القسم واسم القسم إلى جدول جديد ونبقي نسخة من رمز المشروع في الجدول الأصلي (لأنه جزء من المفتاح الرئيسي).

Eno	**Proj_code**	**Hours**
210	P1	12
210	P2	20
210	P3	40
201	P1	30
201	P3	15
305	P2	40
305	P3	20

Eno	**Ename**
210	Ali
201	Salem
305	Ali

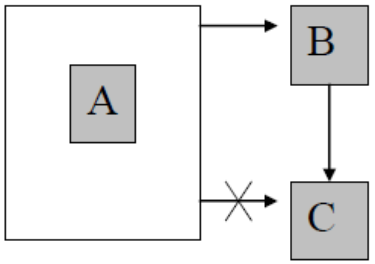
Proj_code	**Deptno**	**Dname**
P1	10	Research
P2	20	Operation
P3	20	Operation

الصيغة المعيارية الثالثة
(Third Normal Form 2NF)

- إن الجدول يكون في الصيغة المعيارية الثالثة إذا:

1. كان الجدول في الصيغة المعيارية الثانية.
2. لم يحتوي الجدول على اعتمادية متعدية.

- **الاعتمادية المتعدية Transitive Dependency:** هي أن تعتمد بعض الأعمدة (الصفات) اعتماداً وظيفياً على صفة غير المفتاح الرئيسي.



- نلاحظ أن A تحدد B,C أي أن B,C تعتمد اعتماد وظيفياً على A،
- وكذلك أن B تحدد C اي أن C تعتمد اعتماداً وظيفياً على B، وفي هذه الحالة يمكن ان نقول أن هذا الجدول يحتوي على اعتمادية متعدية.

الاعتمادية المتعدية
Transitive Functional Dependency (TFD)

- بمعنى انه هناك صفة تعتمد على صفة أخرى والصفة الأخرى تعتمد على المفتاح الرئيسي، في هذه الحالة تكون الصفة التي في المنتصف هي صفة وسيطة.
- يعني أنها تعتمد بطريقة غير مباشرة على المفتاح الرئيسي، أي أن الصفة لا تعتمد على المفتاح الرئيسي مباشرة، وإنما تعتمد على صفة هي التي تعتمد على المفتاح الرئيسي.
- نلاحظ في الجدول التالي أن رقم القسم يعتمد مباشرة على المفتاح الرئيسي (رقم الموظف)، بينما اسم مدير القسم يعتمد على رقم القسم وليس على المفتاح الرئيسي.

Eno	**Deptno**	**Dept_manager**
10	1	Ali
35	2	Hassan
55	3	Noor
20	4	Ahmed

Eno → Deptno → Dept-manager

Deptno → TFD

الصيغة المعيارية الثالثة
(Third Normal Form 2NF)

- في هذه الحالة يكون الحل:

1) نحذف الصفة المتعمدة من الجدول الأساسي.

2) ننشأ جدول جديد به الصفة المعتمدة مع الصفة الوسيطة، ونعين الصفة الوسيطة مفتاح رئيسي.

Eno	Deptno	Dept_manager
10	1	Ali
35	2	Hassan
55	3	Noor
20	4	Ahmed

Deptno	Dept_manager
1	Ali
2	Hassan
3	Noor
4	Ahmed

الصيغة المعيارية الثالثة
(Third Normal Form 2NF)

- والأن هل الجداول التالية في الصيغة المعيارية الثالثة 3NF؟

Eno	**Proj_code**	**Hours**
210	P1	12
210	P2	20
210	P3	40
201	P1	30
201	P3	15
305	P2	40
305	P3	20

Eno	**Ename**
210	Ali
201	Salem
305	Ali

Proj_code	**Deptno**	**Dname**
P1	10	Research
P2	20	Operation
P3	20	Operation

- وللإجابة نجيب على التساؤلين التاليين:
- 1- هل الجداول في الصيغة المعيارية الثانية 2NF؟
- نعم، وذلك لعدم وجود اعتمادية جزيئة.
- 2- هل توجد اعتمادية متعدية؟
- لمعرفة ذلك يجب تحديد الاعتمادية الوظيفية لكل جدول:
- **الجدول الأول:**

FD1 : Eno ⟶ Ename

- لا توجد اعتمادية متعدية.
- **الجدول الثاني:**

FD2 : Eno, Proj_code ⟶ Hours

- لا توجد اعتمادية متعدية.
- **الجدول الثالث:**

FD1 : Proj_code ⟶ Deptno, Dname

FD2 : Deptno ⟶ Dname

- المفتاح الرئيسي Proj_code يحدد Deptno, Dname وفي نفس الوقت فأن Deptno يحدد Dname أي أن هناك اعتمادية متعدية.
- وللتخلص من هذه المشكلة نقوم بتقسيم الجدول إلى جداول بحيث يضم كل منها الاعمدة التي تعتمد على بعض، ونبقي المفتاح مع الأعمدة التي تعتمد عليه وحده فقط مع ابقاء المحدد الجديد Deptno.
- أي نقوم بنقل رقم واسم القسم إلى جدول جديد ونبقي نسخة من رقم القسم في الجدول الأصلي:

Eno	**Proj_code**	**Hours**
210	P1	12
210	P2	20
210	P3	40
201	P1	30
201	P3	15
305	P2	40
305	P3	20

Eno	**Ename**
210	Ali
201	Salem
305	Ali

Proj_code	**Deptno**
P1	10
P2	20
P3	20

Deptno	**Dname**
10	Research
20	Operation

- Ex: Convert the following UNF table to 3NF form?

UNF

St_no	St_name	C_no	Cource	Degree
30	Ahmed Ali	1 2	DB OS	70 80
33	Khaled Hassan	1 2	DB OS	62 75
35	Noor Mohammed	1 2	DB OS	40 50

- للتحويل إلى 1NF يجب أن تحتوي كل خلية على قيمة بيانية واحدة

1NF

St_no	St_name	C_no	Cource	Degree
30	Ahmed Ali	1	DB	70
30	Ahmed Ali	2	OS	80
33	Khaled Hassan	1	DB	62
33	Khaled Hassan	2	OS	75
35	Noor Mohammed	1	DB	40
35	Noor Mohammed	2	OS	50

FD1: St_no → St_name

FD2: C_no → Cource

FD3: St_no, C_no → Degree

اعتمادا على النتائج السابقة يتم وضع مفتاح رئيسي للجدول مكون من رقم الطالب ورقم المادة.

- للتحويل إلى 2NF يجب التحقق من الشرطين:

1- الجدول في الصيغة المعيارية الاولى 1NF.

2- لا توجد هناك اعتمادية جزئية بين الحقول.

- الشرط الاول تحقق لان البيانات مفردة.
- أما الشرط الثاني فانه لم يتحقق وذلك لأن الحقول غير المفتاحية St_name و Course تعتمد على جزء من المفتاح وليس على كل المفتاح وهذا واضح من خلال الاعتمادات الوظيفية:

FD1: St_no ⟶ St_name

FD2: C_no ⟶ Cource

- لتحويلها الى 2NF نقوم بتقسيم الجدول الى علاقات اعتمادا على الاعتمادات الوظيفية التي جعلت من الجدول ليس في الصيغة المعيارية الثانية وهي:

St_no ⟶ St_name

C_no ⟶ Cource

وتكون النتيجة كما يلي:

FD1: St_no ⟶ St_name

FD2: C_no ⟶ Cource

FD3: St_no, C_no ⟶ Degree

2NF

St_no	St_name
30	Ahmed Ali
33	Khaled Hassan
35	Noor Mohammed

C_no	Cource
1	DB
2	OS

St_no	C_no	Degree
30	1	70
30	2	80
33	1	62
33	2	75
35	1	40
35	2	50

- للتحويل إلى 3NF يجب التحقق من الشرطين:

1. كان الجدول في الصيغة المعيارية الثانية.
2. لم يحتوي الجدول على اعتمادية متعدية.

- لمعرفة ذلك يجب تحديد الاعتمادية الوظيفية لكل جدول:
- **الجدول الأول:**

FD1 : St_no → St_name

- لا توجد اعتمادية متعدية.
- **الجدول الثاني:**

FD2 : C_no →Course

- لا توجد اعتمادية متعدية.
- **الجدول الثالث:**

FD3 : St_no, C_no → Degree

- لا توجد اعتمادية متعدية.
- إذن تحقق شروط 3NF فتبقى الجداول كما هي.

- Homework: Convert the following UNF table to 3NF form?

Author	ISBN	Title	Pages	Publisher
Abraham Silberschatz, Henry Korth	0072958863	Database System	1168	Willy
Abraham Silberschatz, Henry Korth	0471694665	Operating System	944	Willy

- Homework: Convert the following UNF table to 3NF form?

Auth_id	**Fname**	**Lname**	**ISBN**	**Title**	**Pages**	**Publisher**
10	Abraham	Silberschatz	0072958863	Database System	1168	Willy
20	Henry	Korth	0072958863	Database System	1168	Willy
10	Abraham	Silberschatz	0471694665	Operating System	944	Willy
20	Henry	Korth	0471694665	Operating System	944	Willy

- Ex: Convert the following UNF table to 3NF form?

Eno	**Ename**	**Age**	**Child_name**	**Company**	**Serv_year**	**Job_title**
30	Ahmed	45	Sami, Layla	IBM	15	Accountant
33	Khaled	50	Majed, Ali, Rami	MIT	20	Programmer

Eno	**Ename**	**Age**	**Child_name**	**Company**	**Serv_year**	**Job_title**
30	Ahmed	45	Sami	IBM	15	Accountant
30	Ahmed	45	Layla	IBM	15	Accountant
33	Khaled	50	Noor	MIT	20	Programmer
33	Khaled	50	Ali	MIT	20	Programmer
33	Khaled	50	Rami	MIT	20	Programmer

- **FD1 : Eno → Ename**

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

1. مقدمة عن الجبر العلائقي Relational Algebra

- الجبر العلائقي بسيط الفهم وعملياته يمكن تطبيقها بفاعلية.

- عمليات الجبر العلائقي ليست ضرورية مثل SQL حتى ولو لها نفس
- الاسم.
- الجبر العلائقي لغة نظرية غير اجرائية.
- إنشاء علاقة جديدة دون تغيير العلاقة أو العلاقات الأصلية، علماً أن السجلات تتأثر دون تكرار.
- قدم في عام ١٩٧٠ E.F.CODD الجبر العلائقي كأساس لغات استعلام قاعدة البيانات.
- يعتبر الجبر العلائقي الأساس النظري لنموذج البيانات العلائقية ولغات الاستعلام.
- يعتقد البعض أن معرفة الجبر العلائقي يسهل عملية كتابة استعلامات صحيحة بلغة SQL.

2. التعرف على المفاهيم والرموز المستخدمة في عمليات الجبر العلائقي

- تعتبر عمليات الجبر العلائقي أحد طرق تمثيل البيانات (الكائنات مع الخصائص) .
- التعامل مع البيانات توفر مجموعة من الإجراءات (العمليات) للتعامل مع علاقة واحده أو علاقتين أو مجموعة من العلاقات (الجداول العلائقية).
- رموز العمليات مجموعه عمليات لمعالجة العلاقات.
- يتضمن الجبر العلائقي شرحاً توضيحياً عن طرق التعامل مع البيانات من خلال الجبر العلائقي.
- يعرف النموذج العلائقي على أنه تلخيص مبسط للبيانات، بينما يعتبر الجبر العلائقي الواجهة "وعادة يدعى النموذج المنطقي".

3. عمليات الجبر العلائقي

تصنف عمليات الجبر العلائقي بالطريقة الأولى إلى صنفين هما:

- **العمليات التي تطبق على علاقة واحدة فقط، ويكون الناتج علاقة واحدة جديدة، والعلاقة الأصلية تبقى كما هي لا تتأثر.**
- **العمليات التي تطبق على علاقتين أو أكثر ويكون الناتج علاقة واحدة جديدة، والعلاقات الأصلية تبقى كما هي لا تتأثر.**

العمليات التي تطبق على علاقة واحدة فقط

- العرض PROJECTION
- الانتقاء SELECTION
- أعادة التسمية RENAMING

العمليات التي تطبق على علاقتين او اكثر

- الاتحاد UNION
- الفرق DIFFERENCE
- التقاطع INTERSECTION
- الضرب الكارتيزي CARTESIAN PRODUCTS
- الضم بأنواعها .JOIN (Equi-join, Semi-join, …..,etc.)
- القسمة DIVISION

ويمكن تصنيف عمليات الجبر العلائقي بطريقة ثانية الى:

- **عمليات رئيسة وهي:**
  - الاتحاد UNION
  - الفرق DIFFERENCE
  - الانتقاء SELECTION
  - العرض PROJECTION
  - الضرب الكارتيزي CARTESIAN PRADUCTS
- **عمليات مشتقة وهي:**
  - الضم JOIN
  - التقاطع INTERSECTION
  - القسمة DIVISION
- **عمليات إجرائية:**

تعبير علائقي يحدد استعلام (سلسلة من العمليات).

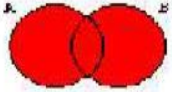
الجبر العلائقي: نظام رياضي يستخدم لوصف ومعالجة البيانات. ويتكون من مجموعة العمليات والتي تعمل على (تعالج) علاقة أو علاقات، وغالباً ما تستخدم في تحسين أو الحصول على نسخة أكثر فعالية لاستعلامات قواعد البيانات.

❖ البناء الأساسي للجبر العلائقي يتكون من:

- العلاقات.
- العمليات على العلاقات.
- كل من المعاملات والنتائج عبارة عن علاقات، ولا تغيير على المجموعات الأصلية.

اسم العملية	رمز العملية	الصيغة العامة
الاتحاد Union	U	R U S

إذا كانت العلاقتان R, S تحتوي على I و J من الصفوف على التوالي فإن ناتج اتحاد العلاقتين يكون علاقة جديدة مثلاً T تحتوي على الأكثر (I+J) صف دون تكرار، أي العلاقة الجديدة تحتوي كل السجلات الموجودة في العلاقتين R و S دون تكرار، (إزالة السجلات المكررة). وهي تمثل عملية المنطق (أو OR) وهي:



A U B

عملية تبادلية: R U S = S U R

وكذلك توزيعية R U (S U T) = (R U S) U T

درجة العلاقتين يجب أن تكون متوافقة،

(عدد الخصائص في العلاقتين متساو ومستمدة من نفس نطاق (مجال) البيانات في الحقول المتقابلة).

مثال:

Table R

A	49
B	54

table S

64	D
49	A

R U S ➔ T

A	49
B	54
D	64

مقدمة لقواعد البيانات

المحاضرة الثامنة

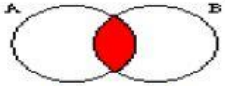
الجبر العلائقي

Relational Algebra

التقاطع Intersection

اسم العملية	رمز العملية	الصيغة العامة
التقاطع Intersection	$\cap$	$R \cap S$

إذا كانت العلاقتان R, S تحتوي على I و J من الصفوف على التوالي فإن ناتج تقاطع العلاقتين يكون علاقة جديدة مثلاً T تحتوي على الصفوف الموجودة (المتكررة) في كلا العلاقتين R, S فقط وهي تمثل عملية المنطق (و and) وهي:



$A \cap B$

عملية تبادلية $R \cap S = S \cap R$

وكذلك توزيعية $R \cap (S \cap T) = (R \cap S) \cap T$

درجة العلاقتين يجب أن تكون متوافقة، (عدد الخصائص في العلاقتين متساو ومستمدة من نفس نطاق (مجال) البيانات في الحقول المتقابلة).

مثال: $R \cap S \rightarrow T$

Table R

A	49
B	54

Table S

D	64
A	49

A	49

مقدمة لقواعد البيانات

المحاضرة الثامنة

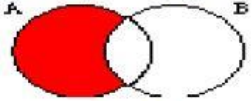
الجبر العلائقي

Relational Algebra

الطرح (الفرق) Difference

اسم العملية	رمز العملية	الصيغة العامة
الطرح Difference	−	R − S

إذا كانت العلاقتان R, S تحتوي على I و J من الصفوف على التوالي فإن ناتج طرح العلاقتين يكون علاقة جديدة مثلاً T تحتوي على الصفوف (السجلات) الموجودة في R وغير موجودة في العلاقة S وهي:



A – B

عملية غير تبادلية $R - S \neq S - R$

درجة العلاقتين يجب أن تكون متوافقة،

(عدد الخصائص في العلاقتين متساو ومستمدة من نفس نطاق (مجال) البيانات في الحقول المتقابلة).

مثال:

Table R

A	49
B	54

Table S

D	64
A	49

R − S ➔ T

B	54

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

القسمة Division

اسم العملية	رمز العملية	الصيغة العامة
القسمة Division	÷	R ÷ S

إذا كانت العلاقتان R, S تحتوي على I و J من الخصائص على التوالي فإن ناتج قسمة العلاقتين R / S تكون علاقة جديدة مثلاً T تحتوي على S ▬ R خاصية (عمود)، وعليه فإن عدد أعمدة (خصائص) العلاقة R يكون أكبر من عدد أعمدة العلاقة S، وإذا كانت X عنصر في العلاقتين R و S، وكانت Y عنصر فقط في العلاقة R فإن R/S يعطي كل السجلات التي تحتوي Y من العلاقة R والتي لها تشابه (تتطابق) مع X لكل قيم X في S.

ويمكن الحصول على عملية القسمة من خلال عمليات العرض، الطرح وعملية الضرب الكارتيزي كما في الشكل الآتي:

$$\pi_{<c>}(R1) - \pi_{<c>}((R2 \times (\pi_{<c>}(R1))) - R1)$$

مثال: ما ناتج تنفيذ العملية التالية **T = R / S**

R	
Empid	COMP
56	IBM
76	GE
56	GE
٦٢	GM

/

S
COMP
GE
IBM

=

T
EmpID
56

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

عملية الضرب الكارتيزي Cartesian product

اسم العملية	رمز العملية	الصيغة العامة
الضرب الكارتيزي	×	$R \times S$

إذا كانت العلاقتان R, S تحتوي على I و J من الصفوف على التوالي وعلى n,m من الحقول (اعمدة) على التوالي فإن ضرب العلاقتين يكون علاقة جديدة مثلاً T تحتوي على الأكثر (I X J) صف و (m+n) عمود، وتشمل كل الأزواج الممكنة للعلاقتين R, S. *(جمع كل سجل في العلاقة R مع كل سجلات العلاقة S).*

درجة النتيجة = مجموع الدرجات للمعاملات.

R(I1,I2,I3,I4,…,In) x S(J1,J2,J3,J4,…,Jm)= T(I1,I2,…In,J1,J2,…Jm)

حيث سجلات T تمثل سجل لكل جمع بين سجلات S,R و n,m خصائص وليس لها معنى إلا إذا استخدمت مع غيرها من العمليات

عملية الضرب الكارتيزي:

مثال: ما ناتج تنفيذ العملية الآتية: **R x S**

R

W	X
A	49
B	54

S

Y	Z
D	64
A	46

R x S ⟶ T

W	X	Y	Z
A	49	D	64
A	49	A	46
B	54	D	64
B	54	A	46

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

عملية الانتقاء Selection

اسم العملية	رمز العملية	الصيغة العامة لكتابتها
الانتقاء Select	σ	$\sigma_{\text{<condition>}}$ (Relation)

تعتبر عملية احادية Unary operation بحاجة إلى جدول (علاقة) واحد لإجرائها .

الشرط Condition يحتوي على أحد رموز عمليات المقارنة الآتية <>,=,<=,>=,>,< بالإضافة إلى:

a اسم حقل موجود في العلاقة.

b اسم حقل موجود أو قيمة ثابتة.

الشرط مثلاً (a <> 5).

صيغ مختلفة للشرط:

$\sigma_{\text{<condition>}}$ (<Relation>)

$\sigma_{\text{<condition> And / Or <condition>}}$ (<Relation>)

$\sigma_{\text{Not <condition>}}$ (<Relation>)

$\sigma_{\text{(<condition> and <condition>) or <condition>}}$ (<Relation>))

إذا كانت العلاقة R تحتوي على عدد من الخصائص $R_{(A,B,C,D,\ldots,K)}$ على التوالي وعلى عدد من الصفوف (I1,I2,….,In) فان ناتج عملية الانتقاء يكون علاقة جديدة مؤقتة تحتوي على كل خصائص العلاقة ولكن يعرض الصفوف التي تحقق (ينطبق عليها) شرط ما على علاقة.

إذاً تستخدم عند الحاجة لانتقاء مجموعة جزئية من الصفوف الأفقية من علاقة والتي تحقق الشرط الانتقاء في أمر الانتقاء ، ويتم حذف ما تبقى من السجلات (الصفوف) التي لا تحقق الشرط دون التأثير على الخصائص. ويتم حذف السجلات المكررة في العلاقة الجديدة. دون أن تتأثر العلاقة الأصلية.

بناءً على العلاقة الآتية Person:

Person		
name	age	weight
Ali	34	80
Issa	28	64
Mosa	34	80
Nader	29	70
Fadi	54	54

ما ناتج تنفيذ الأوامر الآتية:

١. $\sigma_{age>=30}(Person)$

$\sigma_{age>=30}(Person)$		
name	age	weight
Ali	34	80
Mosa	34	80
fadi	54	54

٢. $\sigma_{age>=weight}(Person)$

$\sigma_{age>=weight}(Person)$		
name	age	weight
Fadi	54	54

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

عملية العرض Projection

اسم العملية	رمز العملية	الصيغة العامة لكتابتها
العرض Project	π/pi أو Π	$\pi_{<attribute\text{-}list>}$ (<Relation>)

تعتبر عملية أحادية Unary operation بحاجة إلى جدول (علاقة) واحد لإجرائها .
إذا كانت العلاقة R تحتوي على عدد من الخصائص R (a, b, c, ... ,n) على التوالي فإن ناتج عملية العرض يكون علاقة جديدة مؤقتة تحتوي فقط على الخصائص المحددة في قائمة الخصائص <attribute-list>، وتظهر الخصائص في العلاقة بنفس ترتيب الخصائص في قائمة الخصائص ودون تكرار للسجلات.

إذاً تستخدم عند الحاجة لانتقاء عدد من خصائص علاقة R، واحد أو أكثر من مجموعة الخصائص الكلية، وذلك بتحديد أسماء الخصائص المطلوبة في قائمة الخصائص<attribute-list> ويتم استثناء ما تبقى من الخصائص دون التأثير على الجدول الأصلي، ويتم اختزال الصفوف المكررة في العلاقة الجديدة.

درجة العلاقة = عدد الخصائص في العلاقة.

بناءً على العلاقة الآتية **Person**

Person			
id	name	address	hobby
11	علي	بيت لحم	تجميع طوابع
٢٢	موسى	نابلس	تجميع عملة
٣٣	عيسى	القدس	التسلق
٤٤	نادر	بيت لحم	تجميع طوابع
٥٥	فادي	الخليل	تجميع عملة

أوجد ناتج تنفيذ الأوامر الآتية:

1. $\pi_{\text{name, hobby}}(\text{Person})$

$\pi_{\text{name, hobby}}(\text{Person})$	
name	hobby
علي	تجميع طوابع
موسى	تجميع عملة
عيسى	التسلق
نادر	تجميع طوابع
فادي	تجميع عملة

2. $\pi_{\text{Hobby}}(\text{Person})$

$\pi_{\text{Hobby}}(\text{Person})$
hobby
تجميع طوابع
تجميع عملة
التسلق

ملاحظة:
يمكن تداخل عمليات العرض والانتقاء بأمر واحد، وقد يكون ناتج علاقة مدخل لعلاقة جبرية جديدة. كما في الأوامر الآتية:

1. $\pi_{\text{id, name}} (\sigma_{(\text{address='بيت لحم' or hobby = 'التسلق'})} (\text{Person}))$

id	name
11	علي
٣٣	عيسى
٤٤	نادر

2. $\pi_{\text{sname, rating}} (\sigma_{\text{rating}>8} (\text{Employee}))$

Sname	Rating
Yunes	9
Hamza	10

٣. للحصول على قائمة من أرقام الموظفين الذين يعملون في دائرة رقم واحد.

$$\pi_{\text{empnum}} (\sigma_{\text{deptno} = 1} (\text{Employee}))$$

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

إعادة التسمية Renaming

اسم العملية	رمز العملية	الصيغة العامة لكتابتها
إعادة التسمية Rename	ρ	$\rho_{a/b}(R)$ أو $\rho_R(E)$

تعتبر عملية أحادية وتستخدم لإعادة تسمية الحقول (الخصائص) أو اسم العلاقة نفسها وتظهر على أحد الصور الآتية:

١. $\rho_{a/b}(R)$ الناتج علاقة جديدة نسخة طبق الأصل للعلاقة R باستثناء الحقل b لكل السجلات تم إعادة تسميته إلى a .

٢. $\rho_R(E)$ لإعادة تسمية العلاقة E إلى R، وهي مهمة عند ضم جدول مع نفسه.

٣. $\rho_{R(a/b)}(E)$ لإعادة تسمية علاقة والخصائص يمكن إجراء عملية إعادة التسمية على اسم حقل في قائمة أسماء الحقول أو على اسم علاقة في قائمة أسماء العلاقات وتعتبر أسماء الحقول والجداول الجديدة أسماء مستعارة (أي لا يتم استبدالها في قاعدة البيانات) وتكون مؤقتة لا تخزن بشكل دائم. لإعادة تسمية العلاقة E إلى R وكذلك استبدال اسم الحقل b بالاسم a.

مثال:

استبدل اسم الحقل **name** بالاسم **empname** في العلاقة **Employee** الآتية:

Employee	
name	**Employee-id**
علي	٣٤١٥
موسى	٢٣١٤
رنا	٤٥٢٥

الحل:

نكتب الأمر الآتي:

$$\rho_{empname/name}(Employee)$$

فنحصل على العلاقة الآتية:

empname	**Employee-id**
علي	٣٤١٥
موسى	٢٣١٤
رنا	٤٥٢٥

ملاحظات:

- خصائص العلاقة يجب أن تكون اسماؤها مختلفة .
- هذه العملية مفيدة عند الحاجة لضم علاقة مع نفسها.

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

Relational Algebra

عملية الضم Join

اسم العملية	رمز العملية	الصيغة العامة لكتابتها
الضم Join	$\bowtie$	$R \bowtie_{<\text{join condition}>} S$

إذا كانت العلاقة R تحتوي على عدد من الخصائص R(A1,A2,….An) على التوالي ومن درجة n والعلاقة S تحتوي على عدد من الخصائص S(B1,B2….Bm) m على التوالي، فان ناتج عملية الضم يكون علاقة جديدة مؤقتة تحتوي على خصائص العلاقتين M+N بدون تكرار للخصائص المشتركة. وتدعى عملية الضم أحياناً الضم الداخلي Inner join.

$\{A1,A2….AN\} \cap \{B1,B2….BM\}$

وفيما يتعلق بسجلات العلاقتين R و S فيتم ضم السجلات المشتركة أو المتطابقة في بيانات الحقل المشترك في علاقة جديدة مؤقتة بعد مقارنة بيانات الخصائص المشتركة في كلا الجدولين كي لا يحصل تكرار للسجلات وعدم ضم السجلات التي لا تشترك بالقيمة في كلا الجدولين بعد مقارنة بيانات الحقل المشترك في كلا الجدولين بمعنى ضم البيانات أفقياً من صف بيانات من جدول مع صف جدول أخر عند تحقق شرط ما .

درجة العلاقة الناتجة = مجموع درجات العلاقات مطروح منها الخصائص المكررة.

- عملية الضم عملية تبادلية $R \bowtie S = S \bowtie R$
- عملية الضم عملية تجميعية $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$
- ابسط صورة لعملية الضم هي عملية الضرب الكارتيزي لعلاقتين
- في حالة تشابه أسماء الخصائص فلا يسمح بذلك ويتطلب إعادة تسميتها.
- في حالة عدم وجود أعمده مشتركة عملية الضم الطبيعي تحجم في عملية الضرب الكارتيزي $R \times S$
- عملية الضم يمكن الحصول عليها من حاصل عملية الضرب الكارتيزي وعملية الانتقاء وربما عملية العرض.

Join = R x S + Selection

natural Join = Equi-join + Project

- شرط الضم يمكن ان يكون مجموعة مركبة تستخدم and و or وتستخدم رموز عمليات المقارنة للشرط. (شرط) and (شرط) or (شرط) ... And (شرط).
- الصيغة العامة *لشرط* الضم <join condition> هي <a θ b>.
- يمكن إجراء عملية الضم Join على شروط أخرى غير المساواة Equi-join.
- يمكن تصنيف عملية الضم حسب نوع المقارنة بين الأعمدة وطريقة التعامل مع ناتج المقارنة كما يلي:
- كل عمليات الضم هي عملية Theta ورموز المقارنة (> , < , <> , >= , <=) تدعى Theta Join.
- Equi-Join عملية ضم علائقي عندما يكون شرط الضم هو تعبير المساواة في هذه الحالة تدعى Equi-Join

Natural Join الضم الطبيعي:

عبارة عن عملية ضم لجدولين بخاصية مشتركة، والناتج جدول جديد حيث كل صف فيه عبارة عن (يتكون من) زوج من الصفوف من الجداول المدخلة عند تساوي قيمة الحقل المشترك (أي عندما يتحقق الشرط)، دون تكرار حيث تظهر سجلات الحقل المشترك مرة واحدة فقط في الجدول الناتج.

Natural Join = Equi-Join + Projection

مثال:

عملية ضم الجدولين S و R على اساس المساواة في العمود K في كلا الجدولين

R		
K	X	Y
1	A	2
2	B	4
3	C	6
4	D	8
5	E	10

S	
K	Z
1	20
4	24
5	28
7	32
9	36

Equi-Join				
K	X	Y	K	Z
1	A	2	1	20
4	D	8	4	24
5	E	10	5	28

Natural Join			
K	X	Y	Z
1	A	2	20
4	D	8	24
5	E	10	28

خطوات حساب الضم الطبيعي Natural join لكل من العلاقتين R و S كما يلي:

١. احسب حاصل ضرب العلاقتين R و S (ضرب كارتيزي R×S) .

٢. اختار السجلات التي تتساوى عند قيم الحقل المشترك R . Y = S . Y

٣. عرض الخصائص بدون تكرار (أي عرض R . Y أو S . Y (الناتج من خطوة ٢) وسمها Y ثم اعرض باقي الخصائص.

وهي تعتبر حالة خاصة من عملية Equi-Join والتي يجب ان تحقق الشروط الآتية:

١. شرط المقارنة يجب أن يكون رمز المساواة.

٢. كل الحقول المشتركة يجب أن تشارك في عملية الضم.

٣. واحد من الحقول المشتركة يجب انضمامه لمجموعة السجلات الناتجة.

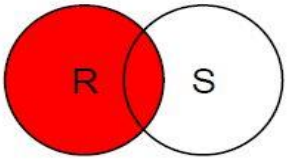
Left Outer Join (⟕)

الناتج مجموعة السجلات التي يتم ضمها من العلاقتين R و S والمتساوية في قيم الحقول المشتركة في العلاقتين بالإضافة إلى سجلات العلاقة R التي ليس لها قيم مشتركة في العلاقة S.

* **Null تعني ان القيم غير معروفة (غير موجودة)**

Employee		
Name	EmpId	DeptName
هاني	٣٤١٥	المالية
سعاد	٢٢٤١	المبيعات
جمال	٣٤٠١	المالية
هديل	٢٢٠٢	المبيعات
تامر	١١٢٣	الإدارة

Department	
deptName	Manager
المبيعات	هديل
الإنتاج	حسن



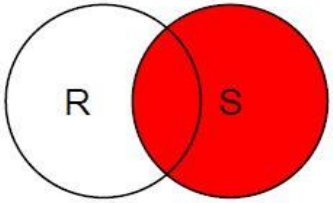
$R - S + R \cap S$

Employee ⟕ Department			
Name	EmpId	DeptName	Manager
هاني	٣٤١٥	المالية	null
سعاد	٢٢٤١	المبيعات	هديل
جمال	٣٤٠١	المالية	null
هديل	٢٢٠٢	المبيعات	هديل
تامر	١١٢٣	الإدارة	null

Right Outer Join (⟖)

الناتج مجموعة السجلات التي يتم ضمها من العلاقتين R و S والمتساوية في قيم الحقول المشتركة في العلاقتين بالإضافة إلى سجلات العلاقة S التي ليس لها قيم مشتركة في العلاقة R .

Employee		
Name	EmpId	DeptName
هاني	٣٤١٥	المالية
سعاد	٢٢٤١	المبيعات
جمال	٣٤٠١	المالية
هديل	٢٢٠٢	المبيعات
تامر	١١٢٣	الإدارة



$R \cap S + S - R$

Department	
deptName	Manager
المبيعات	هديل
الإنتاج	حسن

Employee ⟖ Department			
Name	EmpId	DeptName	Manager
سعاد	٢٢٤١	المبيعات	هديل
هديل	٢٢٠٢	المبيعات	هديل
null	null	الإنتاج	حسن

مقدمة لقواعد البيانات

المحاضرة الثامنة

الجبر العلائقي

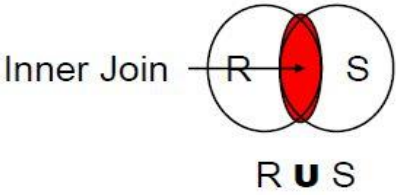
Relational Algebra

Full Outer Join (⟗)

الناتج مجموعة السجلات التي يتم ضمها من العلاقتين R و S والمتساوية في قيم الحقول المشتركة في العلاقتين، بالإضافة إلى سجلات العلاقة S التي ليس لها قيم مشتركة في R وكذلك سجلات العلاقة R التي ليس لها قيم مشتركة في S في الحقول المشتركة.

Employee		
Name	EmpId	DeptName
هاني	٣٤١٥	المالية
سعاد	٢٢٤١	المبيعات
جمال	٣٤٠١	المالية
هديل	٢٢٠٢	المبيعات
تامر	١١٢٣	الإدارة

Department	
deptName	Manager
المبيعات	هديل
الإنتاج	حسن



Employee ⟗ Department			
Name	EmpId	DeptName	Manager
هاني	٣٤١٥	المالية	null
سعاد	٢٢٤١	المبيعات	هديل
جمال	٣٤٠١	المالية	null
هديل	٢٢٠٢	المبيعات	هديل
تامر	١١٢٣	الإدارة	null
null	null	الإنتاج	حسن

Semijoin (⋉)

حيث R و S علاقتان وهذه العملية مشابه لعملية الضم الطبيعي Natural Join وناتجها يكون فقط مجموعة السجلات الموجودة في R التي لها قيمة مساوية في الحقل المشترك في S.

افرض ان لديك الجداول التالية:

Employee

Name	Empid	Deptname
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Hassan	2202	Production

Dept

Deptname	Manager
Sales	Hassan
Production	Hamza

Employee ⋉ dept

Name	Empid	Deptname
Sally	2241	Sales
Hassan	2202	Production

(▷) Antijoin

حيث R و S علاقتان وناتجهما يكون فقط مجموعة السجلات الموجودة في R التي ليس لها قيمة مساوية في الحقل المشترك في العلاقة S.

Employee

Name	Empid	Deptname
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Hassan	2202	Production

Dept

Deptname	Manager
Sales	Hassan
Production	Hamza

Employee ▷ dept

Name	Empid	Deptname
Harry	3415	Finance
George	3401	Finance