



مدخل إلى قواعد المعطيات

د. مادلين عبود

قواعد معطيات 1

مقدمة:

تكون قواعد المعطيات فرعاً أساسياً في فروع المعلوماتية، وتعتبر المفاهيم التي تعتمد عليها والأدوات التي تقدمها العمود الفقري لتطوير أنواع عديدة من التطبيقات المعلوماتية الواسعة الانتشار، خاصة في التطبيقات الإدارية والعالية والنظم المساعدة في اتخاذ القرار. وقد بدأ الاهتمام بهذا النوع من التطبيقات مع ظهور الحاجة إلى إدارة حجم كبير من المعطيات، إذ تبين عجز طرق البرمجة التقليدية مثل الـ Files عن مواكبة التطور الحاصل في حجومات هذه المعطيات، وهذا ما استدعى إنشاء نظم عامة تهدف في المقام الأول، إلى تحسين وتسهيل طرق التعامل مع حجومات كبيرة من المعطيات، من قبل عدد كبير من المستخدمين مع تحقيق أمن المعطيات. ان ما سنتناوله في هذه المحاضرات من معلومات كافية لتغطي مفاهيم المادة وتم التوسع ببعض منها من أجل تقديم محتوى كاف باذن الله راجين الله التوفيق لنا ولكم.

مفردات المقرر

سيتم في هذا المقرر التطرق إلى أربع مفاهيم (فصول) أساسية وهي:

✓ النموذج العلاقائي.

✓ لغة SQL.

✓ مدخل إلى قواعد المعطيات.

✓ المخطط المفاهيمي لقاعدة المعطيات.

المراجع:

Fundamentals of database systems – Elmasri and Navathe.

قواعد المعطيات (1) - د. راكان رزوق و د. مادلين عبود - جامعة دمشق - كلية الهندسة المعلوماتية.

شفافيات المادة.

If you fail to plan, you plan to fail



الفصل الأول: مدخل إلى قواعد المعطيات

سيتم إعطاء هذا الفصل على محاضرتين: في محاضرتنا هذه سنتعلم مفهوم قواعد المعطيات وما هي مبررات استخدامها والهدف من نظم إدارتها والوظائف التي توفرها تلك الأنظمة من إدارة للمعلومات، بالإضافة إلى نمذجة وتصميم قواعد المعطيات بشكل عام ومستويات تعريفها.

مبررات استخدام قواعد المعطيات:

سنبين لكم مبررات استخدامنا لقواعد المعطيات وذلك عبر عرض مبسط للطرق التقليدية لتطوير الأنظمة المعلوماتية مع أهم مشكلاتها ومبدأ حلها باستخدام قواعد المعطيات وذلك عبر المثال التالي:

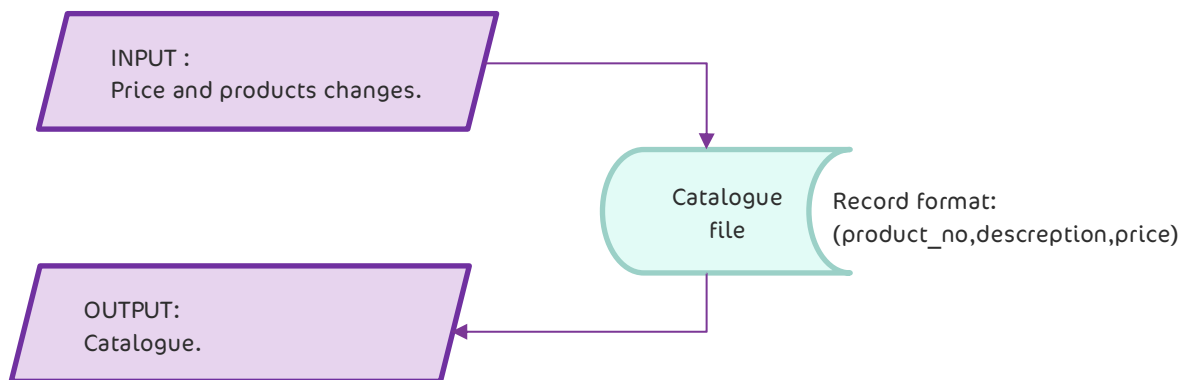
■ الطرق التقليدية لتطوير الأنظمة المعلوماتية:

مثال: مؤسسة صناعية تنتج وتسوق أجهزة كهربائية منزلية

تطبيق أول:

نظام معلوماتي لإصدار دليل تجاري لمنتجات المؤسسة

الشرح: سيكون دخل هذا البرنامج (The input) هو عبارة عن تغيرات المنتجات وأسعارها حيث يوجد سجل (Record) لكل منتج يضم (رقم المنتج، الوصف، السعر). تخزن كل هذه السجلات في ملف وهذا الملف يمثل الدليل المطلوب حيث تتم طباعته كل فترة معينة حسب الطلب. يمكن تمثيله بالمخطط التالي:



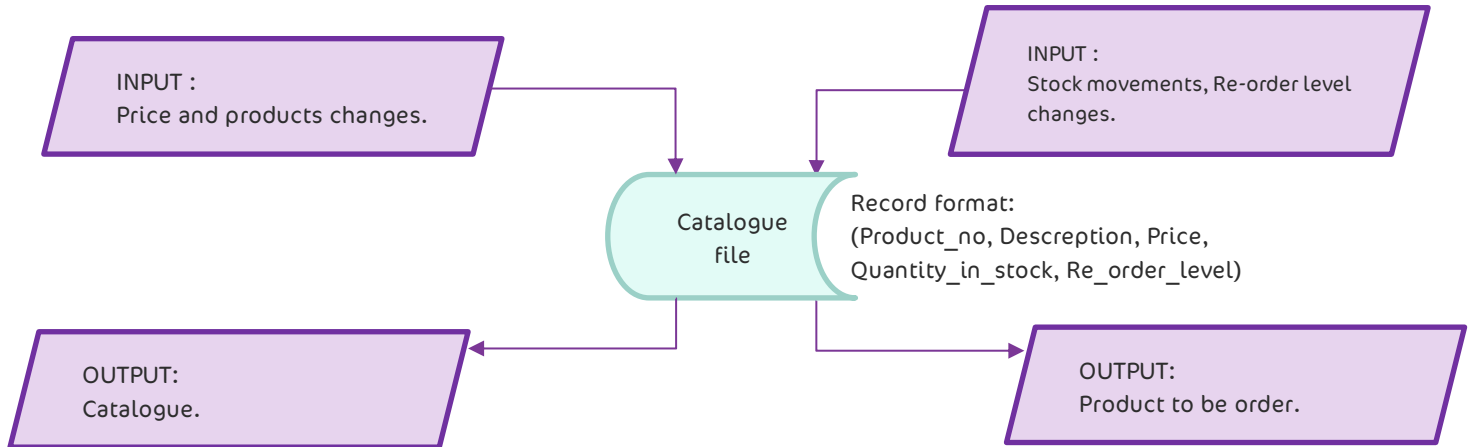
نلاحظ أن هذا البرنامج بسيط جداً ولا يحوي عمليات سوى التغييرات الحاصلة على المنتجات أو على أسعارها التي اعتبرناها كدخل. وبشكل مبدئي ستكون مهمة التطبيق عرض جدول بالمواد الموجودة مع أسعارها (دليل Catalogue) في الخرج .OUTPUT

« If it's meant to be. It will be. »

نظام معلوماتي لإدارة مستودعات المؤسسة

تطبيق ثاني:

الشرح: عندما أرادت المؤسسة تطوير برنامج يقوم بإدارة مستودعاتها، فإننا نضيف خاصيتين على سجلات المنتجات للبرنامج السابق ونعتبرها كمدخلات إضافية في ملف الدليل (Catalogue) وهي (الكمية الموجودة في المستودع، الكمية الدنيا التي يجب توافرها). يمكن تمثيل ذلك عبر المخطط التالي:



وبالتالي فإن دخل البرنامج السابق هو عبارة عن تغيرات المنتجات مع أسعارها بالإضافة إلى التغيرات الحاصلة على الكمية الدنيا المتوفرة والكمية الكلية للبضائع في المستودع وسيتم طباعة الدليل مع المنتجات التي يتم طلبها في الخرج.

■ إن طريقة الملفات Files التي استخدمناها في المثال السابق تعتبر تقليدية بحد ذاتها وهنا يجب مراعاة ما يلي:

- ← ضرورة كتابة إجراءات لتنظيم الوصول إلى الملفات المشتركة لأن عدم التنسيق أو عدم المزامنة قد يؤدي إلى أخطاء لا يمكن التراجع عنها مثال: التعديل على نفس المعلومة في نفس الملف.
 - ← احتمال حدوث أخطاء أثناء الاستثمار أي قراءة خاطئة للمعلومات مثال قراءة محرف عوضاً عن رقم قد يؤدي إلى أخطاء كثيرة.
 - ← تكرار لبعض الوظائف في النظامين.
 - ← أي تعديل على بنية أو شكل تخزين المعلومات داخل الملف يستوجب تعديل على كيفية قراءة المعلومات من الملف مثال: إذا كان لدينا معلومة مسجلة بالطريقة Ahmed like: cake, Ahmed like: fast cars فإن تبديلها إلى الشكل Ahmed like: cake, fast cars يستوجب تعديل في طريقة القراءة.
 - ← كما أن مدير النظام مسؤول عن متابعة استثمار كل الأنظمة وعن تنسيق عمليات التعديل فيها.
- إن وجود العديد من الملفات لنفس الـ Data وتكرار المعلومة في ملفات متعددة يخلقان مشكلة تضارب المعطيات، وعليه فإن النقاط السابقة يجب الانتباه لها تفادياً لحدوث تلك المشكلة.

وإذا كنا للتو تحدثنا عن برنامجين بسيطين فقط فماذا لو كان البرنامج يتعامل مع الكثير من الملفات ويقوم بالكثير من المهمات؟

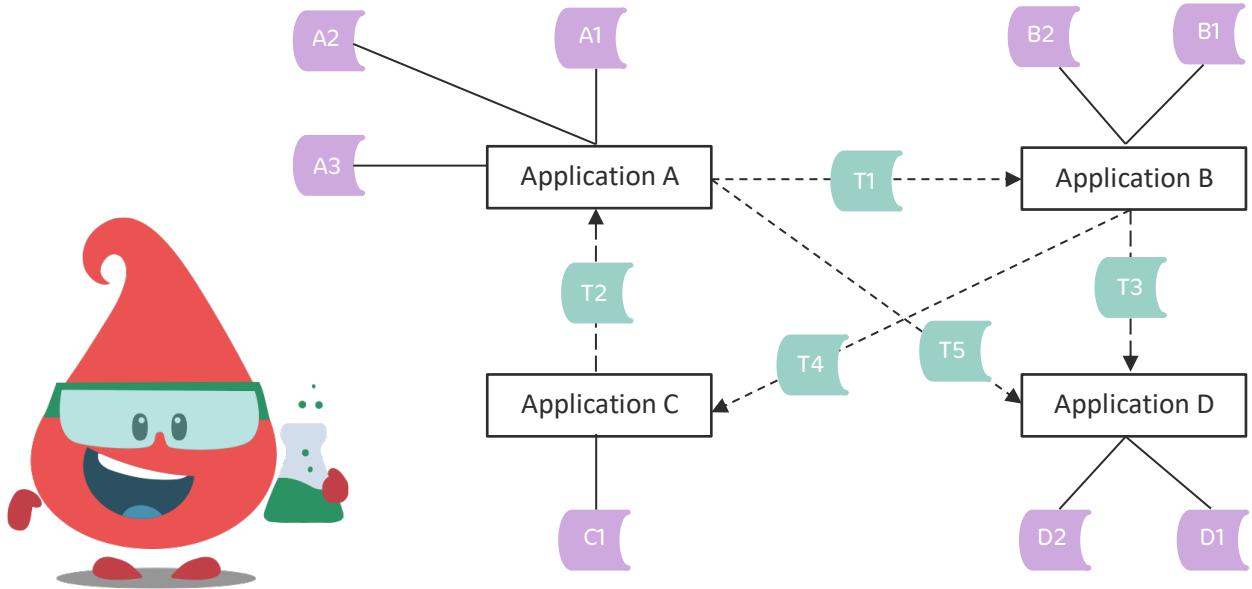


لو أن المؤسسة طلبت من صاحب البرنامج إتاحة عمليات جديدة مثل: إدخال أصناف جديدة - فاتورة مشتريات - مبيعات - مردود مبيعات - جرد أصناف - إتاحة بطاقة صنف - الخ...

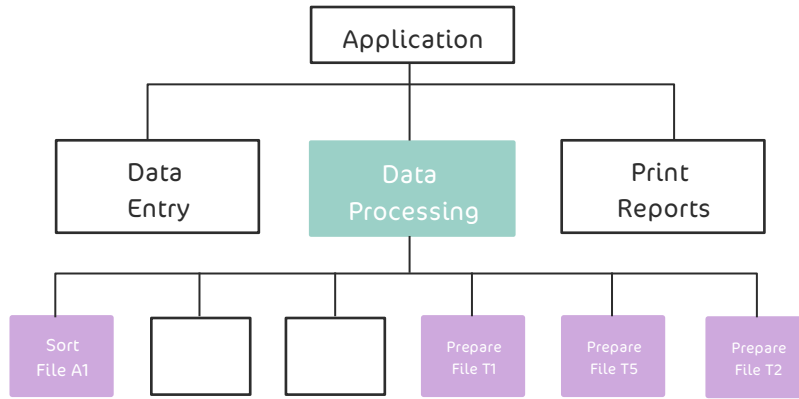
نلاحظ أنه أصبحنا بحاجة إلى التعديل على المعطيات بشكل كبير، والـ file system قد يحتاج إلى إعادة هيكلة وإعادة ترتيب المعطيات لديه، أو في حال كانت العملية صعبة سنحتاج إلى تكرار بعض المعطيات وبالتالي تكرار بعض العمليات، وفي حال أصبحت المعطيات ضخمة جداً لن يكون هذا الحل كافياً.

لذلك نبحث عن حل أفضل تبني فكرته على استقلال ملفات كل تطبيق أو برنامج عن ملفات الآخر حيث تكون خاصة به. ومن الطبيعي أن تكون البيانات مشتركة بين تطبيقات نظام معين فبطبيعة الأحوال نضع نسخ مختلفة من البيانات في ملفات خاصة بكل تطبيق وذلك لتجنب التضارب.

إن حل مشكلة التضارب بين التطبيقات يكون عبر إنشاء ملفات مؤقتة يتم من خلالها إرسال البيانات إلى التطبيقات حيث تقوم كل منها بقراءة تلك البيانات وتعديل ملفاتها. يمكن تمثيل ذلك عن طريق المخطط التالي:



حيث أن T1, T2, T3, T4, T5 كل منها يمثل ملف مؤقت. وبالطبع فإن هذه الطريقة غير فعالة وليست عملية وذلك لأن هرمية وظائف أي برنامج معين ضمن النظام فيها جزء كبير من الأعباء مخصص للملفات وحدها وبالتالي يصبح الجزء الخاص فعلاً بالبرنامج حجمه ضخم على الشكل الآتي:



مما سبق نستخلص نتيجة لأهم مساوئ التخزين بطريقة الملفات Files ومنها:

- ← أنظمة منفصلة يهتم كل منها بوظيفة محددة.
- ← تكرار في المعلومات.
- ← ضرورة تحديد نسق الملفات المؤقتة.
- ← يحوي كل نظام جزءاً مخصصاً لإعداد الملفات المؤقتة وتبادلها ومعالجتها.
- ← وظائف متكررة في كل الأنظمة (فرز الملفات والبحث ضمنها).

■ مبدأ عمل ال Database :

جميع المساوئ السابقة الناتجة عن استخدام الأنظمة التقليدية لتطوير الأنظمة المعلوماتية ولدت ضرورة إنشاء طريقة أفضل وهي تخزين كل البيانات في مكان واحد وهو ما يسمى بقواعد المعطيات Database. وهنا يأتي دور نظام قواعد المعطيات في تأمين سلاسة التعامل مع المعطيات، من خلال فصل المعلومات Data بشكل كامل عن البرنامج Application ووضعها بشكل مركزي بحيث يمكن إجراء أي عملية تعديل عليها (Update, Delete....) دون حدوث تضارب أو فقدان لأي منها، ودراسة المخطط العلاقتي قبل تمثيله كبنية معطيات يتيح إمكانية لتوسيع ميزات البرامج في المستقبل.



“Shoot for the **Moon** even if you miss, You’ll land among the **Stars.**”

تعريف قاعدة المعطيات وإدارة المعلومات

من جميع ما سبق يمكن استنتاج المفاهيم التالية:

قاعدة المعطيات Database:

هي مجموعة من المعطيات **المهيكلّة** غير **المتكررة**، المسجلة على وسط تخزين يسمح **بالوصول** إليها من قبل عدة برامج تطبيقية.

نظام إدارة قواعد المعطيات DBMS – Database Management System:

تجمع من **المعطيات** المرتبطة فيما بينها، ومجموعة من **البرامج** التي توفر الوصول إلى هذه المعطيات. الهدف الأساسي لنظم إدارة قواعد المعطيات هو توفير **محيط عمل ملائم وفعال** يمكن من **تخزين** المعلومات ضمن قاعدة المعطيات واسترجاعها لاحقاً.

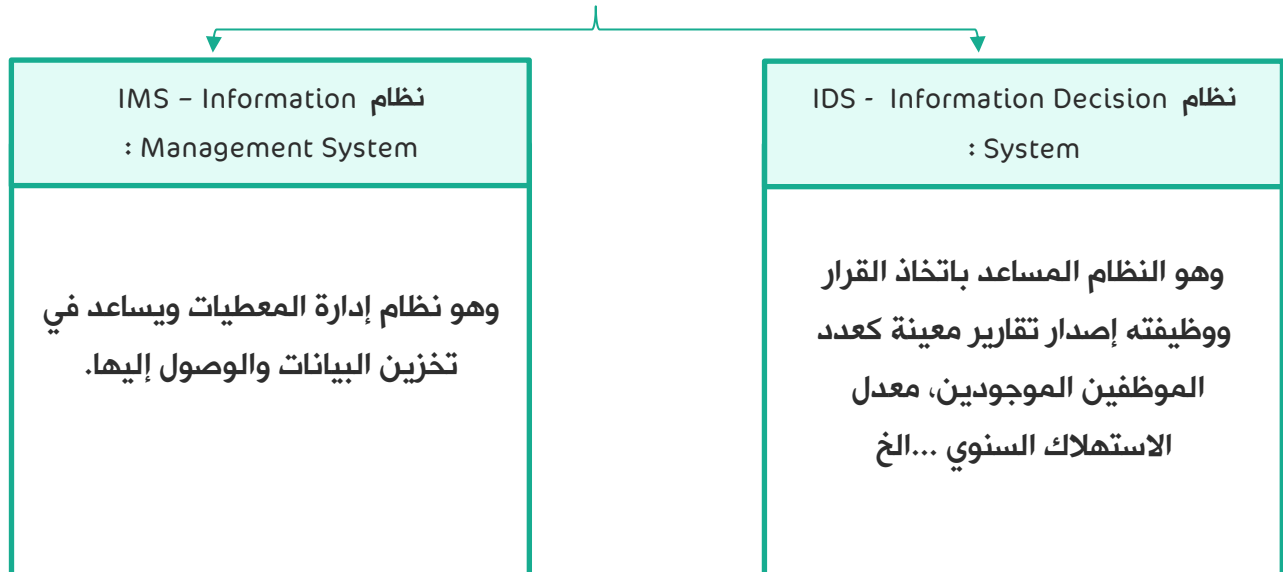
إدارة المعلومات:

هي عملية تعريف بنى تخزين المعلومات وإيجاد التقنيات الملائمة للتعامل مع المعلومات المخزنة وتقديم نظم أمان لحماية المعلومات المخزنة من الوصول غير المشروع وتجنب التضارب في المعلومات المخزنة نتيجة تشارك عدة مستثمرين في الوصول إلى المعلومات.

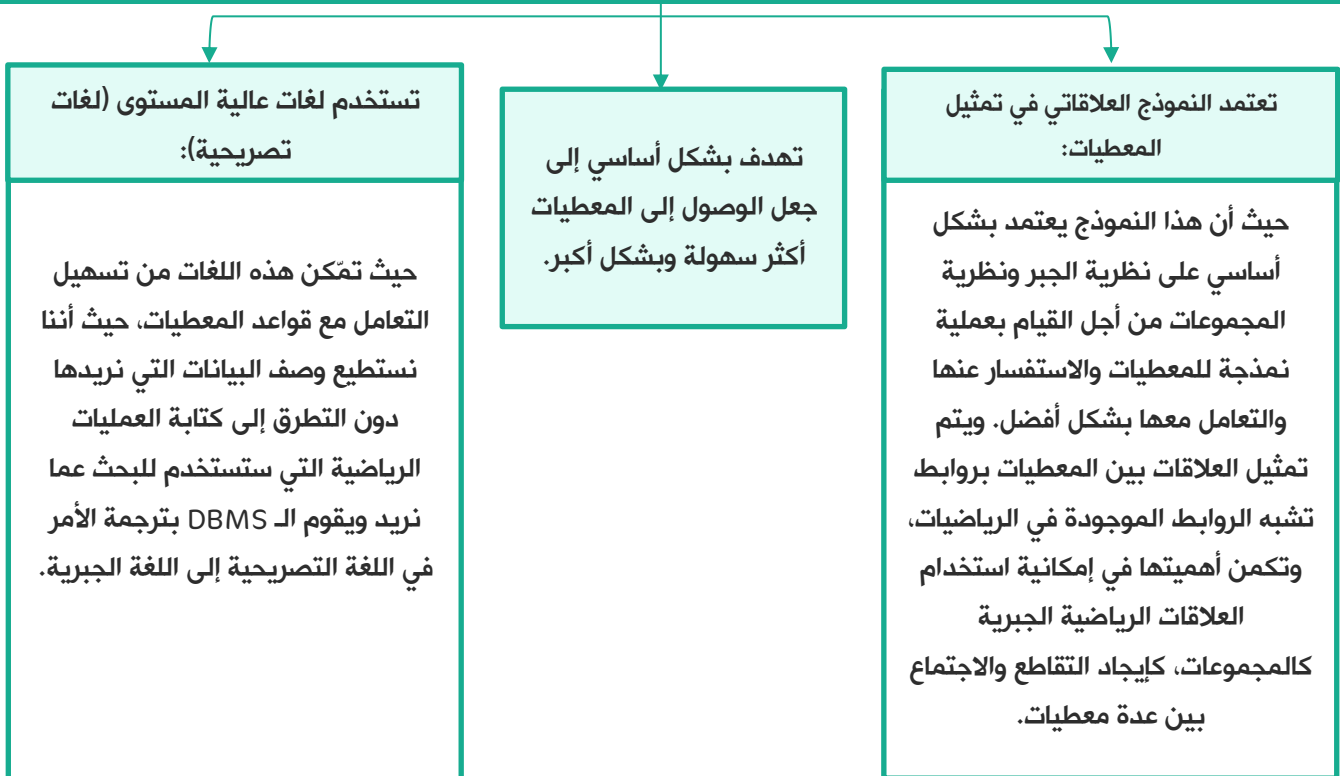
تاريخ تطور قواعد المعطيات

ظهر مفهوم قواعد المعطيات بشكل أساسي، من حيث تخزين المعطيات بشكل مركزي في مكان محدد والوصول إليها في فترة الستينات من القرن الماضي. وقد مرت قواعد المعطيات أثناء تطورها بعدة مراحل من خلال ثلاثة أجيال وهي:

■ **الجيل الأول: فترة الستينات**، وقد ظهر خلال تلك الفترة نوعين من أنظمة المعلومات:



الجيل الثاني: بداية السبعينات وهي ممتدة حتى الآن، وتتميز بالخصائص التالية:

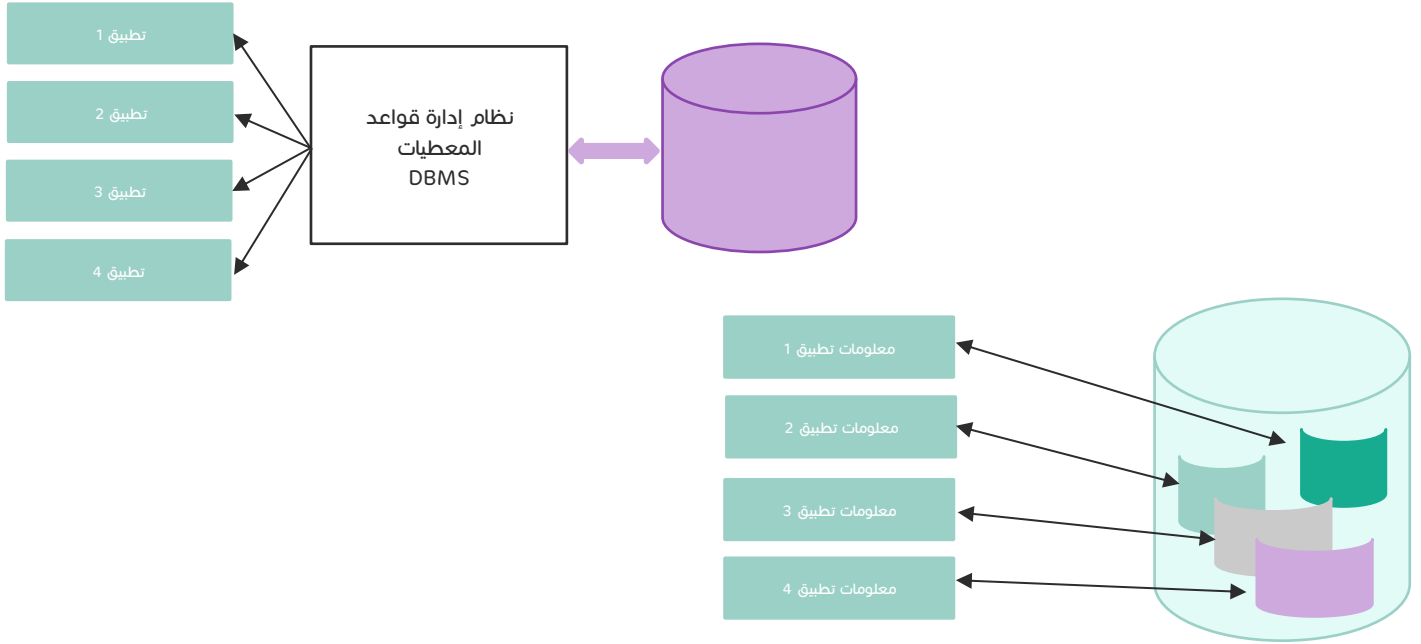


من أكثر أنظمة قواعد المعطيات العلاقائية انتشارا في الوقت الحالي هي:
 . ORACLE, INGRES, SYBASE, INFORMAX, DB2
 وقد تم تسويق أنظمة هذا الجيل منذ عام 1982م.

الجيل الثالث: نهاية الثمانينات وهي ممتدة حتى الآن.

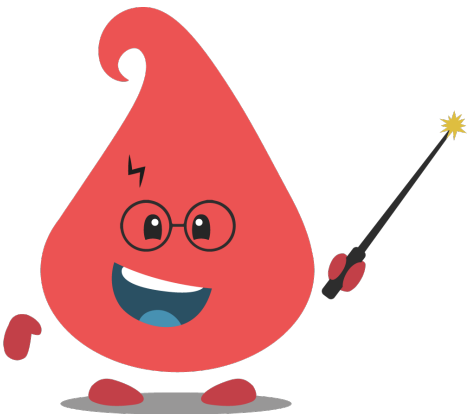


الهدف من نظم إدارة قواعد المعطيات



يتجلى دور نظام إدارة قواعد المعطيات والذي يعرف ب DBMS في العديد من الأهداف ومنها:

- ← إنشاء أنظمة معلوماتية تعنى بتخزين وإدارة حجوم كبيرة نسبياً من المعلومات بطريقة منظمة وفعالة.
- ← توفير محيط عمل ملائم وطريقة لاسترجاع البيانات لاحقاً بسهولة وبشكل سريع ومريح وفعال.
- ← تسهيل عملية تعريف وإنشاء قواعد المعطيات والتعامل معها وجعل هذه المعطيات متاحة لأكثر من مستخدم بنفس الوقت.
- ← الأمان: أي حماية قواعد المعطيات وحماية الوصول للمعلومات بشكل صحيح.
- ← تأمين عملية الوصول إلى المعطيات Data بأقل كلفة ممكنة ومعالجتها تتم عن طريق ال DBMS .



“Happiness can be found even in the darkest of times, if one only remembers to turn on the lights.”

- Albus Dumbledore.

الوظائف التي تقدمها أنظمة إدارة قواعد المعطيات

❧ سيتم توضيح تلك الوظائف عبر مثال لتطبيق معلوماتي بسيط:

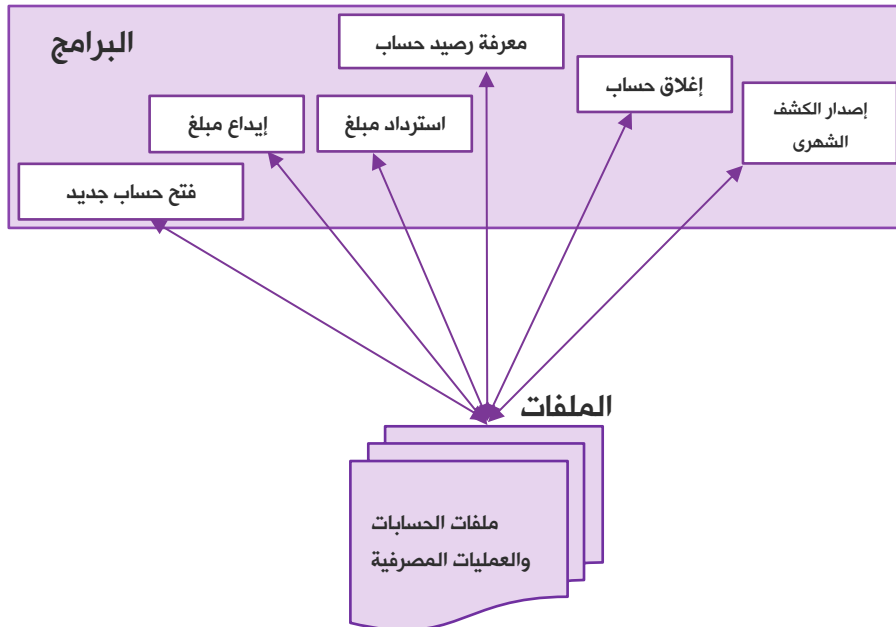
مؤسسة مصرفية تحتفظ بمعلومات عن زبائنها وحساباتهم المصرفية في نظام ملفات تقليدي.

■ يمكن تعريف عدة عمليات على تلك المعطيات مثل:

- ← فتح حساب جديد.
- ← إيداع مبلغ في حساب مصرفي موجود.
- ← استرداد مبلغ في حساب مصرفي موجود.
- ← معرفة رصيد حساب مصرفي.
- ← إغلاق حساب مصرفي (حذف الحساب).
- ← إصدار الكشف الشهري للحسابات.

■ يمكن تخزين تلك المعطيات بشكل تقليدي عن طريق الملفات. وهناك ملفات مشتركة بين الأنظمة مثل:

- ← ملف الحسابات.
- ← ملف العمليات المصرفية.
- ← ملفات مؤقتة التي تستخدم في عمليات الفرز والترتيب.
- ← ملفات دائمة تستخدم لأرشفة المعلومات التاريخية.



إن تطور احتياجات المؤسسة وذلك عبر صدور قوانين جديدة (أنواع جديدة من الحسابات) وتطور بيئة العمل (استخدام آلات الصراف الآلي) سيؤدي إلى ظهور أنواع جديدة من الملفات ومجموعة إضافية من البرامج التطبيقية لإدارة هذه الملفات.

تطوير التطبيقات المعلوماتية باستخدام نظم إدارة الملفات:

عند بناء نظام لهذه المؤسسة باستخدام طريقة الملفات Files التقليدية سوف نضطر لبناء نظام مستقل لكل مهمة وكل نظام يدير ملفاته الخاصة. ولكن سوف تظهر مشاكل مختلفة وهي:

1. تكرار المعطيات وتضاربها:

ويعني ذلك وجود العديد من الملفات لنفس الـ Data وتكرار المعلومة في ملفات متعددة. بالإضافة إلى عدم التوافق فيما بينها عند تطور حاجات المؤسسة.

2. صعوبة الوصول إلى المعطيات:

وهذا يعني أننا نريد كتابة برنامج من أجل تنفيذ كل ميزة جديدة.

3. صعوبة عزل المعطيات:

أي لا يمكننا عزل البيانات عن التطبيق (Application) مما يؤدي إلى صعوبة في التعديل والوصول إلى البيانات. وتتجسد هذه المشكلة في أننا نحتاج إلى إنشاء العديد من الملفات، لما كانت المعطيات موزعة في عدة ملفات ذات بنى مختلفة، فإنه من الصعب كتابة تطبيق جديد لاسترجاع المعطيات وفق أشكال معينة. إن سبب هذه المشكلة هو أن تعريف المعطيات يجري ضمن البرامج التي تدير هذه المعطيات، ومن ثم فإن أي تعديل في بنى التخزين يجب أن يواكبه تعديل كل البرامج التي تتعامل مع هذه المعطيات.

4. تعارض في الوصول المتزامن:

تنتج هذه المشكلة عند وجود أكثر من مستخدم يتعاملون مع البيانات بنفس الوقت، حيث أحدهم يقرأ هذه البيانات وآخر يعدل عليها، وهذه التعديلات لا تصل بشكل مباشر للمستخدمين. حتى يستطيع أكثر من مستخدم الوصول إلى الـ Data والتعديل عليها فإن هذا الأمر يحتاج إلى إدارة معقدة أي أنها تحتاج إلى نظام يقوم بمنع عملية فصل المعطيات عندما يكون أحد المستخدمين يعمل عليها وذلك بغية الحصول على زمن استجابة أقصر وزيادة مردود هذا النظام ولكن ذلك يزيد من أخطار التعديل المتزامن للمعطيات ويزيد تضارب المعطيات الناتجة عنه.

5. صعوبة تحقيق أمن المعطيات:

إن نظام إدارة الملفات يؤمن نوعين من الأذونات للوصول للملفات، إما وصول كامل للملف أو عدم الوصول له، أي أنه لا يقوم بتأمين الوصول الجزئي للملف. وتتجسد هذه المشكلة في عملية الـ access الخاصة بالمستخدم، ففي الـ file system إما يكون لدينا وصول كامل للمعطيات أو لا يكون هناك وصول أبداً.

6. عدم إمكانية التأكد من تكامل المعطيات:

يقصد بتكامل المعطيات بالشروط التي تكون دوماً محققة مهما تغيرت المعطيات، وغالباً ما تكون هذه الشروط محددة ضمن التعليمات البرمجية المسؤولة عن إدارة الملفات، ويجب أن تكون قابلة للتعديل. إذا كنا نكتب برنامج بالطريقة التقليدية فإن هذا الشرط سيكون ضمن الكود، لكن يمكن أن يكون هناك شروط متغيرة مع الزمن، ولتغيير هذه التعديلات البسيطة سنضطر إلى الدخول إلى الكود والتعديل في الجزء المكتوب فيه هذه الشروط، وهذه العملية ليست بسيطة كما تبدو خاصة إذا كانت الـ Data مترابطة مع بعضها وبالتالي فإن التحقق من التكامل بين المعطيات في الأنظمة التقليدية صعب.

■ أهم الوظائف التي توفرها أنظمة إدارة قواعد المعطيات:

1. مركزية المعلومات:

وتتمتع ب خصائص مثل: الإدارة القياسية وإلغاء التكرار وسهولة إدخال وتحديث المعلومات بالإضافة إلى مركزية التحكم والمراقبة.

2. استقلال المعطيات:

أي التعامل مع المعطيات بواسطة برامج تقوم بالوصول إلى هذه المعطيات من مستوى عال من التجريد، لا يظهر الطريقة الفعلية للتخزين، والذي يحتاج إلى معرفة كافة التفاصيل المتعلقة ببنية القاعدة ومحتوياتها الشاملة. وأيضا تخفيف الأعباء الملقة على عاتق المبرمجين. كما يتم من خلالها متابعة العمل على المعطيات في حال حدوث تغير في بنى التسجيلات. الجدير بالذكر إلى أن البرامج التطبيقية لا تتأثر بتغير طرق الوصول.

3. معالجة المعطيات بواسطة لغات غير إجرائية:

واللغة الغير إجرائية هنا هي لغة يستطيع المبرمج بواسطتها أن يسأل قاعدة المعطيات أو يعدّل تلك المعطيات دون تحديد خوارزمية الوصول إلى المعطيات.

4. التسهيلات الخاصة بإدارة المعطيات:

من خلال طريقة تعريف المعطيات وتخزينها والوصول إليها وعرضها. يجري عادةً حصر بعض هذه الأدوات بشخص واحد يدعى مدير النظام.

5. الوصول إلى المعطيات بفعالية:

عبر زيادة عدد الإجراءات التي تنفذ في ثانية واحدة وزيادة عدد المستخدمين الذين يستطيعون الوصول إلى المعطيات بأن واحد بالإضافة إلى إنقاص زمن الاستجابة.

6. التحكم في تكرار المعطيات:

أي يجب أن يكون هناك نسخة وحيدة من المعطيات عبر إعطاء الانطباع بأن كل مستخدم من مستخدمي قاعدة المعطيات يتعامل مع نسخة مستقلة من قاعدة المعطيات. كما يتم توفير الأدوات التي تنسق بين العمليات التي يجريها المستخدمون على النسخة الوحيدة من المعطيات.

7. تكامل المعطيات:

يتم من خلال تكامل المعطيات تحقيق أنواع عديدة من شروط التكامل. في مثال المصارف السابق يتم تحقيق ما يلي:

شروط التكامل المعرفة من قبل المستخدم:

الرصيد أكبر من 250 ل.س.

التكامل المرجعي:

لا يمكن إجراء عمليات مصرفية على حساب قبل فتح الحساب.

تكامل وحدات المعطيات:

لا يمكن فتح حساب مصرفي لزبون دون معرفة عنوانه.

كما يتم من خلالها أيضا تعريف شروط التكامل مع كشف وإيقاف جميع العمليات التي قد تؤدي إلى الإخلال بهذه الشروط.

8. تقسيم المعطيات:

أي تقسيم معطيات قاعدة ما بين عدة تطبيقات، بحيث يستطيع كل منها الوصول إلى المعطيات دون أن ينتظر تطبيقاً آخر.

9. أمن المعطيات:

أي حماية بعض المعطيات الخاصة، بحيث أن مجموعة محددة هي فقط التي تستطيع الوصول إلى تلك المعطيات.

10. تتضمن أنظمة إدارة قواعد المعطيات خوارزميات وطرقاً خاصة لتقسيم المصادر (الوحدة المركزية، وحدات الدخل/الخروج) بين المستثمرين تقسيماً عادلاً.

تصميم قواعد المعطيات

عند تصميم قواعد المعطيات ونمذجتها سنقوم بتعريف المفاهيم الأساسية التالية:

قاعدة المعطيات:

هي مجموعة من البرامج التي تسمح بالوصول إلى المعطيات المخزنة فيها واسترجاعها وتعديلها. وتجمع من الملفات المترابطة.

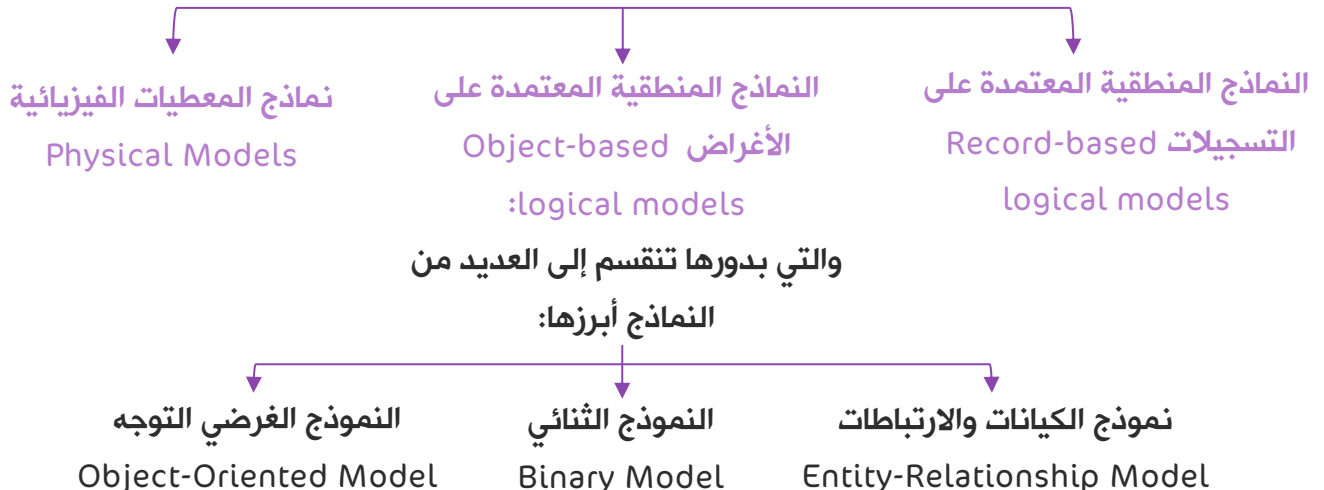
مستويات التجريد (النمذجة):

تستخدم ثلاثة مستويات من التجريد تسمى مخططات (Schema) لتوصيف أي قاعدة معطيات. يجري في كل مستوى توصيف القاعدة ببعض التفصيل الإضافي، عن المستوى الأعلى. كما يقدم نظام إدارة قواعد المعطيات الوسائل الكفيلة بإيجاد الترابط بين هذه المستويات المختلفة.

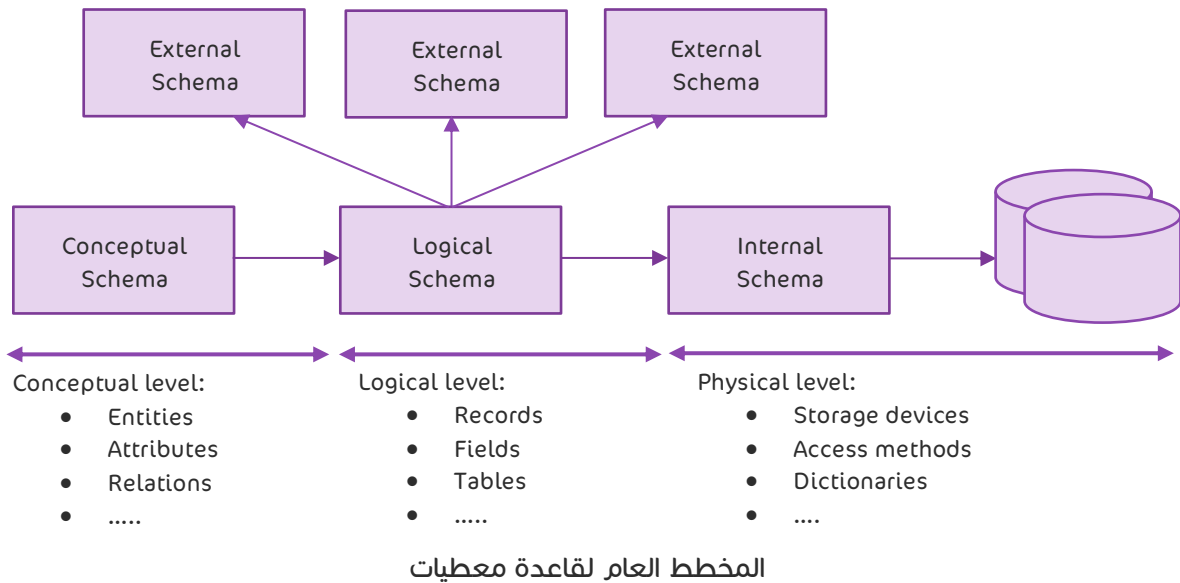
نماذج المعطيات Data Models

نمذجة المعطيات:

هي عملية استخدام مجموعة الأدوات التصميمية التي تساعد على وصف المعطيات، والعلاقات المتبادلة فيما بينها، ودلالة المعطيات، وشروط تناسقها.

تقسم نماذج المعطيات إلى ثلاث مجموعات:

مستويات تعريف قاعدة المعطيات



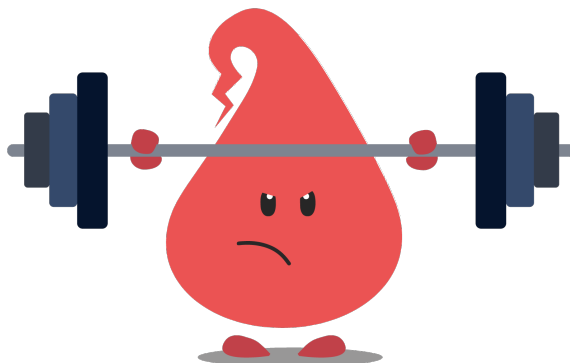
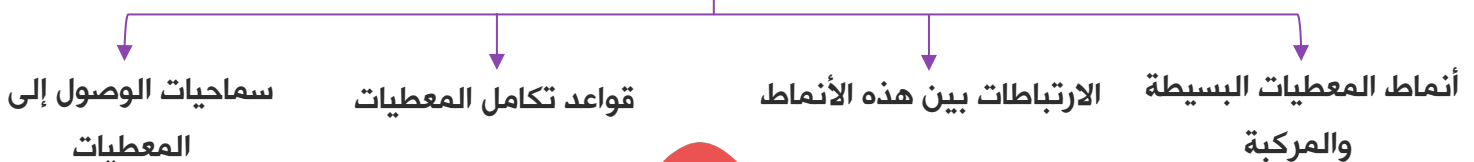
تتم عملية تعريف قاعدة المعطيات على أربعة مستويات أساسية (سيتم شرحها بالتفصيل في المحاضرة القادمة) وهي:

1. المخطط المفاهيمي Conceptual Schema:

يعتبر المخطط المفاهيمي كتجريد للواقع يبين عناصر المعلومات التي ستحتويها قاعدة المعطيات. يبنى هذا المخطط بعد دراسة وتحليل واقع العمل والمعلومات (الوثائق) المتداولة والمخزنة وتحديد ارتباطات هذه المعلومات بالإجراءات المتبعة في المؤسسة. هناك طرق معيارية لتمثيل المخطط المفاهيمي والتي سيتم شرحها بشكل مفصل في المحاضرة القادمة.

2. المخطط المنطقي Logical Schema:

يعد المخطط المنطقي بمثابة نواة قاعدة المعطيات. ويعبر عن تمثيل معياري للمعطيات يعكس طبيعتها وخصائصها وارتباطاتها. يعتمد المخطط المنطقي على النموذج (Model) الذي يتيح نظام إدارة قواعد المعطيات بأنواعه المختلفة (هرمي، شبكي، علاقتي، غرضي التوجه). كما يتضمن ما يلي:



3. المخطط الداخلي Internal Schema:

ويمثل بنية التخزين التي ستحتوي المعطيات فعلياً ويحوي على:

طرق الوصول إلى التسجيلات:

الفهرسة، ربط السجلات، ...

طرق التعامل مع وسائط التخزينالمختلفة

(Segments, Blocks, Buffers)

توصيف ملفات المعطيات:

أنماط الحقول وسعاتها، الحقول المركبة، ...

■ ملاحظة:

في معظم الأحيان لا يحتاج المطور للتدخل في المستوى الداخلي لأن نظام إدارة قواعد المعطيات يقوم بذلك.

4. المخطط الخارجي External Schema:

هو مخطط منطقي جزئي خاص بتطبيق معين يصف جزءاً من قاعدة المعطيات كما يراه التطبيق. كما يصف طريقة تعامل التطبيق مع قاعدة المعطيات ويحدد الجزء الذي يستطيع التطبيق التعامل معه.

■ ملاحظة:

الأصح أنه يوجد 3 مستويات أساسية لمخطط قواعد المعطيات وهي (المخطط المفاهيمي + المخطط المنطقي) = المخطط الخارجي / المخطط الداخلي (الفيزيائي).

The End.

The greatest sin you can commit against yourself
your own life is to know what you want and not to
act.

