

ادارة انظمة قواعد البيانات

Database Management system (DBMS)

عبدالقادر العبدالله

كلية العلوم – تخصص البرمجة

- مقدمة عن قواعد البيانات
- المفاهيم الرئيسية في أنظمة قواعد البيانات
- بناء وهيكلية أنظمة قواعد البيانات
- تصميم أنظمة قواعد البيانات

المخرجات المتوقعة من الدرس

- فهم أساسيات قواعد البيانات وأهميتها مقارنة بالأنظمة التقليدية لتخزين البيانات.
- الإلمام بمصطلحات أساسية مثل البيانات، المعلومات، الجداول، الخصائص (Attribute)، العلاقات (Relationship) واستيعاب مفهوم إدارة قواعد البيانات ودورها في تخزين البيانات واسترجاعها وإدارتها.
- التعرف على مكونات أنظمة إدارة قواعد البيانات مثل الواجهة، محرك قاعدة البيانات، المخططات المعمارية الثلاثة (Three-Schema Architecture).
- تطوير مهارات التفكير المنطقي لتحليل متطلبات البيانات وتحويلها إلى تصميمات عملية
- فهم العلاقة بين تصميم قواعد البيانات وتطوير الأنظمة البرمجية

مقدمة عن قواعد البيانات

• **قاعدة البيانات :** هي نظام يهدف إلى جمع البيانات وتخزينها بطريقة منطقية ومنظمة بحيث يمكن الوصول إليها وإدارتها وتحديثها بكفاءة و تدار باستخدام برامج خاصة تعرف بأنظمة إدارة قواعد البيانات.

تؤدي قواعد البيانات دوراً حيوياً في العديد من المجالات التي تستخدم فيها الحواسيب، مثل الأعمال، التجارة الإلكترونية، الهندسة، الطب، القانون، التعليم، وعلم المكتبات وتشمل قواعد البيانات عدة خصائص منها:

1 - **التكامل والترابط (Integration and Relationship):** البيانات مترابطة ومنظمة بشكل يسمح بربط العناصر المختلفة ببعضها بطريقة منطقية.

2 - **تقليل التكرار (Data Redundancy Reduction):** تكرار البيانات يؤدي إلى زيادة حجم التخزين واحتمالية وجود بيانات غير متسقة قواعد البيانات مصممة لتقليل هذا التكرار باستخدام تقنيات مثل تجزئة البيانات إلى جداول منفصلة مع الحفاظ على العلاقات بينها.

3 - الوصول المتعدد (Multi-User Access): تتيح قواعد البيانات لمستخدمين متعددين الوصول إلى نفس البيانات في نفس الوقت دون تعارض.

4- المرونة (Flexibility) : يمكن تعديل قاعدة البيانات بسهولة لتلبية الاحتياجات المتغيرة.

5 - الدقة والتكامل (Data Accuracy and Integrity): تضمن قواعد البيانات صحة البيانات واتساقها من خلال تطبيق قيود وقواعد محددة.

6 - الأمان (Security): قواعد البيانات تحتوي على آليات للتحكم في الوصول إلى البيانات مثل نظام إدارة الصلاحيات و توفر التشفير لحماية البيانات أثناء النقل والتخزين.

• **خصائص منهج قواعد البيانات :** في نهج قواعد البيانات، يتم الحفاظ على مستودع مركزي للبيانات يعرف مرة واحدة ويمكن للمستخدمين المختلفين الوصول إليه و هذا النهج يتميز بالخصائص التالية :

1 - تنظيم البيانات والطبيعة الذاتية لنظام قاعدة البيانات (Self-describing Nature of a Database System):

تخزين البيانات بطريقة منظمة ومهيكله ويستخدم نظام إدارة قاعدة البيانات لإنشاء البيانات وتصميمها وصيانتها بشكل منطقي ومتسق ويكون هيكل البيانات المخزنة على شكل جداول , اعمدة , انواع البيانات و العلاقات بين الجداول هذا يجعل النظام مستقلا، حيث يمكن لأي مستخدم أو تطبيق فهم قاعدة البيانات.

2 – تجريد واستقلال البيانات (Insulation Between Programs and Data, and Data Abstraction):

يتم فصل البيانات عن البرامج التي تعالجها مما يسمح بتغيير هيكل البيانات دون الحاجة إلى تعديل البرامج المرتبطة بها ولن تؤثر أي تعديلات على بنية قاعدة البيانات على برامج التطبيق والعكس صحيح وتجعل هذه الخاصية تعديل قاعدة البيانات وتحديثها أسهل دون تعطيل عمل أي برنامج وهناك نوعان من استقلال البيانات :

- **الاستقلال المادي للبيانات :** القدرة على تغيير طريقة تخزين البيانات فيزيائيا (مثل نقل البيانات إلى جهاز تخزين جديد) دون الحاجة إلى تغيير التطبيقات أو البرامج التي تستخدم هذه البيانات.

- **الاستقلال المنطقي للبيانات :** يشير إلى القدرة على تغيير الهيكل المنطقي للبيانات (مثل إضافة جدول جديد أو تعديل العلاقات بين الجداول) دون الحاجة إلى تغيير البرامج أو التطبيقات التي تعتمد على قاعدة البيانات.

- 3 - مشاركة البيانات (Data sharing): يمكن مشاركة البيانات بين العديد من المستخدمين والتطبيقات في نفس الوقت كما يقلل من الحاجة إلى تخزين البيانات المكررة في مواقع مختلفة وذلك بفضل ميزات مثل:
- إدارة التزامن : لضمان أن التعديلات التي يجريها مستخدم واحد لا تؤثر على عمل الآخرين.
 - دعم المعاملات : يضمن تنفيذ العمليات بشكل كامل أو عدم تنفيذها إطلاقاً.
- 4 - أمن البيانات (Data Security): يوفر نظام إدارة قاعدة البيانات تدابير أمنية مختلفة مثل المصادقة والتفويض والتشفير لحماية البيانات من الوصول غير المصرح به أو العبث بها أو اختراقها وهذا يجعل من السهل تحديد المستخدم والتحكم في من لديه حق الوصول إلى البيانات وما يمكنه فعله بها كما يوفر نهج قاعدة البيانات أيضاً النسخ الاحتياطي واستعادة البيانات

المفاهيم الرئيسية في أنظمة قواعد البيانات

1. تطور بنية أنظمة إدارة قواعد البيانات (DBMS) :

- الأنظمة القديمة (Monolithic Systems) : كانت تعتمد على تصميم مركزي متكامل، حيث كل وظائف نظام

إدارة قواعد البيانات موجودة في حزمة واحدة كبيرة.

- الأنظمة الحديثة (Modern Systems) : تطورت أنظمة DBMS إلى تصميمات تعتمد على الوحدات

(Modular Design) هذه الأنظمة تستفيد من بنية العميل/الخادم لتوزيع العمل بين المستخدمين والخوادم.

المفاهيم الرئيسية في أنظمة قواعد البيانات

من حيث	الأنظمة الحديثة	الأنظمة القديمة
الوظائف	يتم توزيع الوظائف بين وحدات العميل والخادم	كل الوظائف (تخزين، معالجة، استعلام) تُنفذ في نفس النظام.
الأداء	أفضل: توزيع العمل يُقلل الحمل ويُحسن الأداء.	قد يصبح الأداء بطيئاً مع زيادة الحمل بسبب الطبيعة المركزية
التفاعل مع المستخدم	واجهات رسومية مريحة وسهلة الاستخدام.	واجهات بسيطة مثل النصوص والأوامر.
الاستخدام	مناسب لأنظمة الأعمال الحديثة، الأنظمة الموزعة، وتطبيقات الإنترنت.	مناسب للأنظمة المركزية القديمة فقط.

2. بنية العميل/الخادم في DBMS : تم تقسيم وظائف DBMS الى نوعين رئيسيين من الواحدات :

- وحدة العميل (Client Module): يعمل على أجهزة المستخدمين مثل الحواسيب الشخصية أو محطات العمل
ومن اهم مهام هذه الوحدة :

- التفاعل مع المستخدم من خلال واجهات رسومية مريحة
- تشغيل برامج التطبيقات التي تعتمد على قواعد البيانات
- إرسال الطلبات (مثل الاستعلامات) إلى الخادم

المفاهيم الرئيسية في أنظمة قواعد البيانات

- وحدة الخادم (Server Module): تعمل على أجهزة خوادم متخصصة ومن أهم مهام هذه الوحدة :

- تخزين البيانات بشكل آمن
- معالجة الطلبات والاستعلامات المرسلّة من العميل
- تنفيذ عمليات البحث، الإدراج، التعديل، والحذف
- إدارة قواعد البيانات وتحسين أدائها

❖ الفائدة من بنية العميل/الخادم :

- تقسيم العمل بين أجهزة العميل والخادم مما يؤدي إلى أداء أفضل
- المرونة في ترقية أي من العميل أو الخادم بشكل مستقل دون الحاجة إلى تعديل النظام بالكامل
- دعم الأنظمة الموزعة حيث يمكن أن العملاء والخوادم على شبكات مختلفة

3. النماذج (Data Models) : تستخدم النماذج لتمثيل البيانات وعلاقاتها بشكل منطقي ومن أشهر النماذج :

A. النماذج المفاهيمية (Conceptual Data Models): تقدم مفاهيم قريبة من الطريقة التي يدرك بها

المستخدمون البيانات في العالم الحقيقي و تستخدم :

- الكيان (Entity): يمثل كائنا حقيقيا أو مفهوما في العالم الواقعي مثل "موظف" أو "مشروع".
- السمات (Attributes): تمثل الخصائص التي تصف الكيان مثل "اسم الموظف" أو "راتب الموظف".
- العلاقات (Relationships): تمثل الارتباطات بين الكيانات المختلفة مثل "الموظف يعمل في المشروع A".

❖ مثال : نموذج الكيانات والعلاقات (Entity-Relationship Model)

المفاهيم الرئيسية في أنظمة قواعد البيانات

B. النماذج التمثيلية أو التنفيذية (Representational Data Models): تقدم مفاهيم توازن بين سهولة الفهم من

قبل المستخدمين وطريقة تنظيم البيانات داخل الحاسوب :

- النموذج العلائقي (Relational Model): يعتمد على تمثيل البيانات كجداول تحتوي على صفوف وأعمدة

- النماذج القديمة (Legacy Models) :

- النموذج الشبكي يعتمد على تمثيل البيانات كشبكة من السجلات والعلاقات

- النموذج الهرمي يعتمد على تنظيم البيانات في هيكل هرمي

C. النموذج الكائني (Object-Oriented Data Model): هو نموذج بيانات يعتمد على مفاهيم البرمجة الكائنية

لتنظيم البيانات يتم فيه تمثيل البيانات ككائنات تحتوي على خصائص وسلوكيات

المفاهيم الرئيسية في أنظمة قواعد البيانات

من حيث	النموذج الكائني	النماذج المفاهيمية	النماذج التنفيذية
الهدف الأساسي	تصميم بيانات معقدة تحتوي على سلوكيات إضافية بجانب الخصائص	توفير فهم بسيط وسهل للعلاقات بين البيانات	تسهيل تخزين البيانات وتنظيمها في الأنظمة
الاستخدام	أنظمة إدارة البيانات المعقدة مثل الوسائط المتعددة	تصميم مراحل أولية لقواعد البيانات	أنظمة قواعد البيانات التجارية مثل العلائقية
التعقيد	متوسط إلى مرتفع بسبب الاعتماد على البرمجة الكائنية	منخفض وسهل الفهم للمستخدمين غير التقنيين	متوسط يجمع بين سهولة الاستخدام وقابلية التنفيذ
التنفيذ	يمكن تنفيذه باستخدام لغات البرمجة الكائنية مثل (java)	يستخدم كنموذج نظري ولا ينفذ مباشرة	يتم تنفيذه مباشرة في أنظمة مثل النموذج DBMS العلائقي

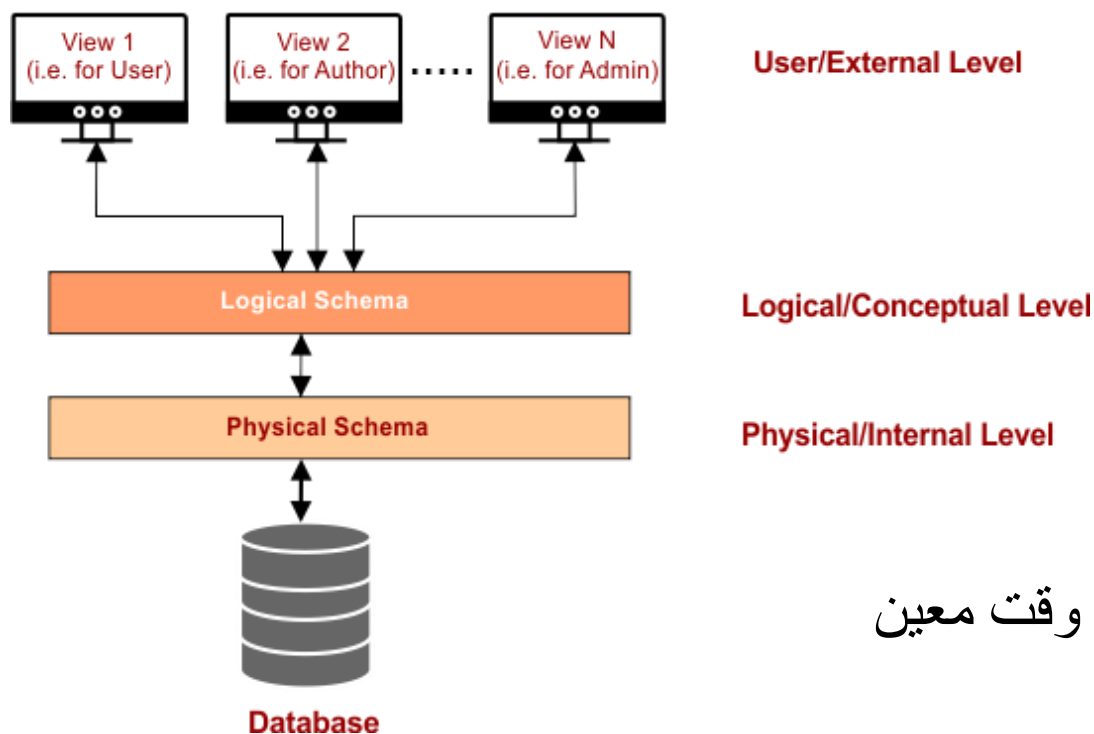
4. المخططات (Schemas) : المخططات هي التصميمات الهيكلية التي تحدد كيفية تنظيم البيانات وتوجد ثلاثة أنواع رئيسية من المخططات :

- المخطط الخارجي (External Schema) : يركز على وجهات النظر الخاصة بالمستخدمين، حيث يمكن لكل مستخدم أن يرى جزءا معينا من البيانات حسب احتياجاته
- المخطط المنطقي (Logical Schema) : يركز على تصميم قاعدة البيانات بشكل عام، مثل الكيانات والعلاقات بين الجداول

المفاهيم الرئيسية في أنظمة قواعد البيانات

- المخطط الفيزيائي (Physical Schema) : يركز على التفاصيل التقنية لتخزين البيانات، مثل كيفية توزيع

البيانات على الأقراص



5. الحالات (Instances) :

الحالات تشير إلى البيانات الفعلية المخزنة في النظام في وقت معين

1. استقلالية البيانات (Data Independence) : القدرة على تغيير المخطط في مستوى معين من قاعدة البيانات دون

الحاجة إلى تعديل المخطط في المستوى الأعلى هذه الخاصية تسهل صيانة وتطوير أنظمة قواعد البيانات.

- استقلالية البيانات المنطقية : تغيير المخطط المفاهيمي دون الحاجة إلى تعديل المخططات الخارجية أو البرامج التطبيقية [إضافة عناصر جديدة , تغيير القيود]

- استقلالية البيانات الفيزيائية : تغيير المخطط الداخلي دون الحاجة إلى تعديل المخطط المفاهيمي أو

المخططات الخارجية [إعادة تنظيم البيانات , إنشاء هياكل وصول إضافية]

❖ كل مستوى من هذه المستويات متصل بالآخر عبر خرائط عند حدوث تغيير في مستوى معين يتم تعديل الخرائط فقط

لتجنب التأثير على المستويات الأخرى

بناء وهيكلية أنظمة قواعد البيانات

استقلالية البيانات المادية\الفيزيائية	استقلالية البيانات المنطقية
تهتم بشكل رئيسي بتخزين البيانات	تهتم بشكل أساسي بالبنية أو تغيير تعريف البيانات
من السهل استرجاعها	من الصعب استرجاع البيانات لأنها تعتمد بشكل أساسي على البنية المنطقية للبيانات
من السهل تحقيق استقلال البيانات المادية	من الصعب تحقيق استقلال البيانات المنطقية
التبديل إلى هياكل البيانات المختلفة /تغيير موقع البيانات وأجهزة التخزين قاعدة	إضافة/تعديل/حذف سمة جديدة

2. لغات أنظمة إدارة قواعد البيانات (DBMS Languages) : توفر أنظمة إدارة قواعد البيانات لغات متعددة تستخدم لتصميم وإدارة قواعد البيانات وتنفيذ العمليات عليها :

- لغة تعريف البيانات (Data Definition Language - DDL) : هي مجموعة من الأوامر المسؤولة عن التحكم في هيكل قاعدة البيانات في المستويات المختلفة و من أشهر الأوامر :

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Age INT  
);
```

- التعديل (ALTER)
- إنشاء (CREATE)
- حذف (DROP)
- إعادة تسمية (RENAME)

- لغة معالجة/تدوال البيانات (Data Manipulation Language - DML): مجموعة من الأوامر المسؤولة عن التفاعل مع البيانات المخزنة في قواعد البيانات و من أشهر الأوامر :

```
SELECT FirstName, LastName, phone  
FROM Employees  
ORDER BY LastName
```

- للاستعلام (select)

- الإدراج (Insert)

- التحديث (Update)

- الحذف (Delete)

```
INSERT INTO Publishers (PubID, PubName, city, province)  
VALUES ('9900', 'Acme Publishing', 'Vancouver', 'BC')
```

• لغة تعريف التخزين (Storage Definition Language - SDL) : تستخدم لتعريف المخطط الداخلي الذي

يحدد كيفية تخزين البيانات و تستخدم ل :

• ضبط الأداء وتحسين كفاءة التخزين

• تحديد الهيكل المادي لقاعدة البيانات

• تحديد أماكن تخزين البيانات على الأقراص

```
CREATE TABLE Students (articles INT),  
ENGINE = INNODB;
```

• لغة تعريف العروض (View Definition Language - VDL) : تستخدم لتعريف العروض (Views) التي

تعتبر وجهات نظر محددة للمستخدمين على قاعدة البيانات و تحدد كيفية ارتباط العروض بالمخطط المفاهيمي :

• إنشاء العروض (Creating Views)

• تعديل العروض (Modifying Views)

• حذف العروض (Dropping Views)

```
CREATE VIEW StudentNames AS  
SELECT Name FROM Students;
```

• لغة الاستعلام المنظمة **SQL (Structured Query Language)**: تستخدم لإدارة ومعالجة البيانات المخزنة

في قواعد البيانات العلائقية وتتيح للمستخدمين إجراء عمليات مختلفة على البيانات وتعتبر لغة شاملة تجمع بين:

• **DDL** لتعريف هيكل قاعدة البيانات

INSERT INTO Students (StudentID, Name, Age)

VALUES (1, 'Ahmed', 20);

• **DML** لمعالجة البيانات

SELECT * FROM Students WHERE Age > 18;

• **VDL** لتعريف العروض

UPDATE Students SET Age = 21 WHERE StudentID = 1;

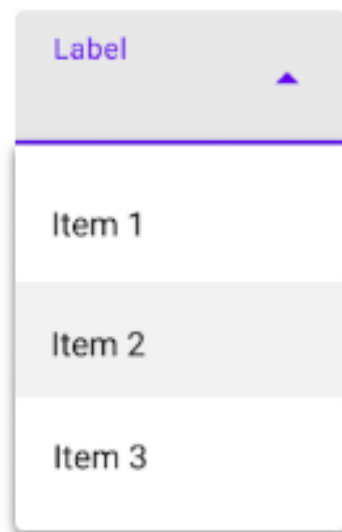
DELETE FROM Students WHERE StudentID = 1;

3. واجهات أنظمة إدارة قواعد البيانات (DBMS Interfaces) : أنظمة إدارة قواعد البيانات توفر واجهات متعددة

لتلبية احتياجات المستخدمين المختلفين بناء على مستوى خبرتهم وطبيعة استخدامهم للنظام :

- واجهات قائمة على القوائم (Menu-Based Interfaces) : تقدم للمستخدم قوائم خيارات تساعد في تكوين

الاستعلامات خطوة بخطوة دون الحاجة إلى معرفة أوامر لغة الاستعلام [قوائم منسدلة في تطبيقات الويب].



- سهولة الاستخدام للمستخدمين غير المتمرسين

- لا حاجة لحفظ الأوامر أو بناء الجمل

بناء وهيكلية أنظمة قواعد البيانات

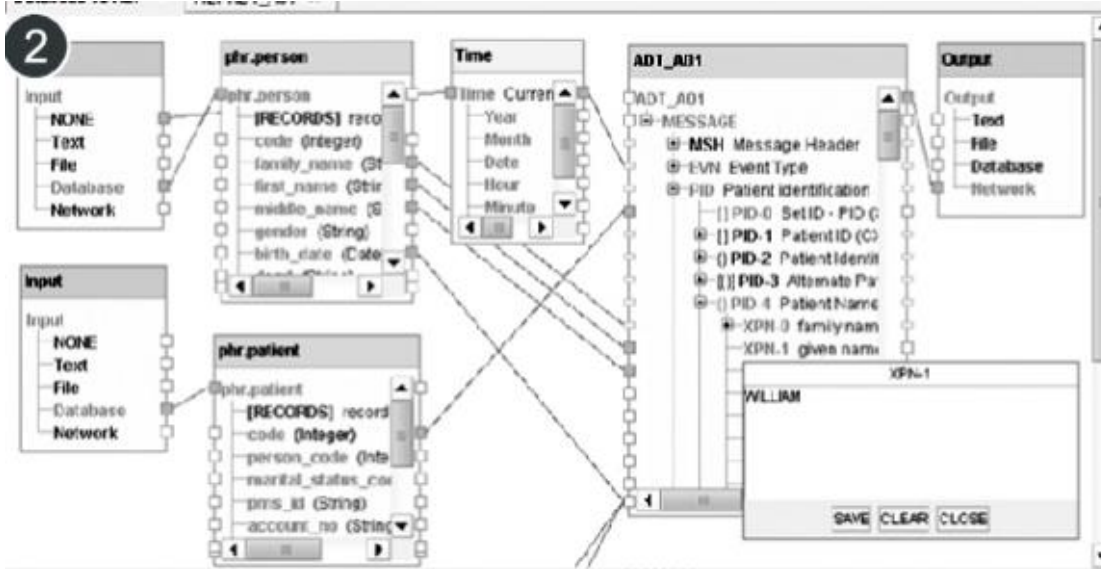
- واجهات تعتمد على النماذج (Forms-Based Interfaces) : تعرض نموذجاً (Form) للمستخدم يمكنه تعبئته لإدخال البيانات الجديدة أو ترك بعض الحقول فارغة لاسترجاع البيانات المطابقة تستخدم لإجراء العمليات

مثل إدخال الطلبات أو استعلامات محددة.

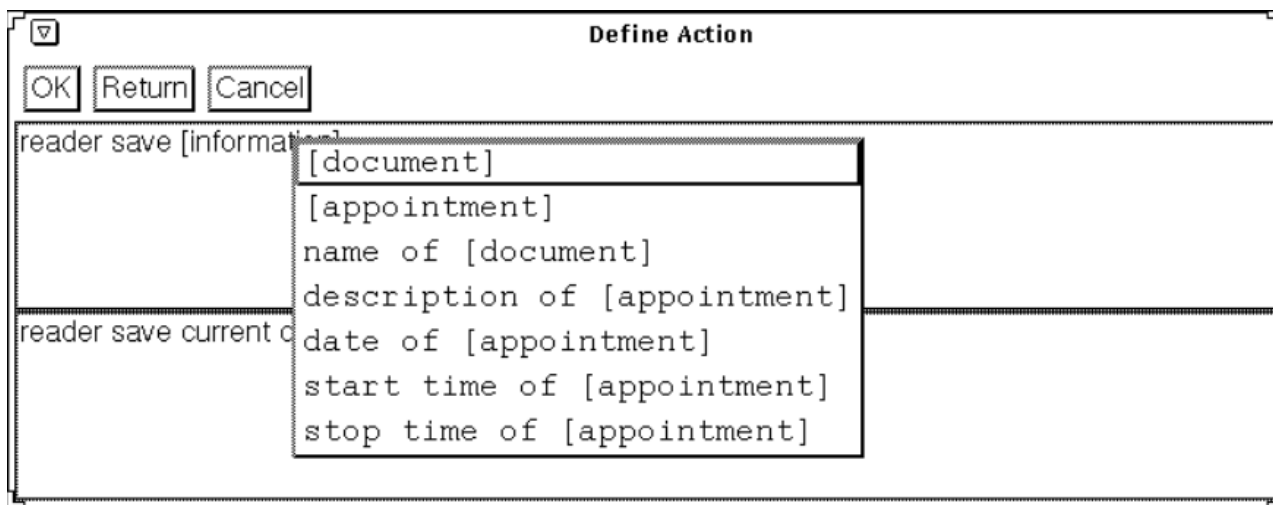
- تصميم بسيط يناسب المستخدمين غير المتخصصين
- بعض الأنظمة تحتوي على لغات خاصة لتحديد النماذج أو أدوات لتصميمها بسهولة

بناء وهيكلية أنظمة قواعد البيانات

- واجهات رسومية (Graphical User Interfaces - GUIs): واجهة تظهر مخططا بيانيا للمستخدم يمكنه من خلاله تكوين الاستعلامات باستخدام الماوس أو أدوات التأشير الأخرى تدمج بين القوائم والنماذج لتسهيل العمليات.
- سهولة الاستخدام وتوفر تجربة تفاعلية
- تتيح رؤية المخطط الكامل لقاعدة البيانات والتفاعل معه



- واجهات اللغة الطبيعية (Natural Language Interfaces) : تتيح كتابة الطلبات بلغة طبيعية مثل الإنجليزية بدلا من كتابة أوامر SQL تحتوي على مخطط داخلي لقواعد البيانات ومعجم للمصطلحات.
- تحاول تفسير الطلب وتحويله إلى استعلام عالي المستوى يمكن معالجته بواسطة DBMS
- في حالة عدم الفهم، يبدأ النظام حوارا مع المستخدم لتوضيح الطلب

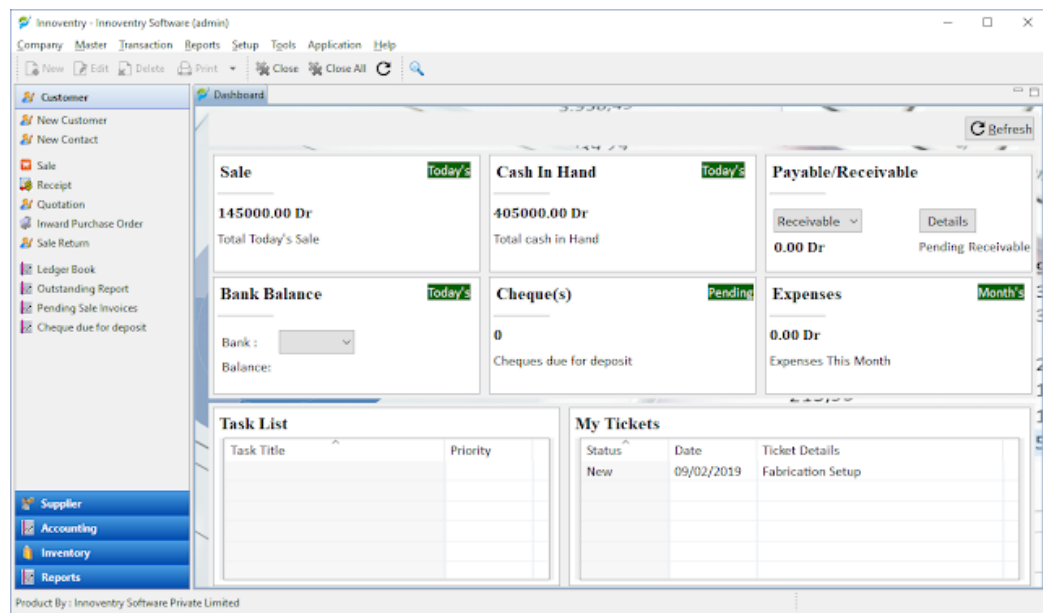


The screenshot shows a 'Define Action' dialog box with three buttons at the top: 'OK', 'Return', and 'Cancel'. Below the buttons is a list of actions and their parameters:

Action	Parameters
reader save [information]	[document] [appointment] name of [document] description of [appointment]
reader save current c	date of [appointment] start time of [appointment] stop time of [appointment]

- سهولة الاستخدام للمستخدمين غير التقنيين
- تستخدم في الأنظمة المعتمدة على الذكاء الاصطناعي

- واجهات المستخدمين المعيارية (Interfaces for Parametric Users) : صمم لفئات محددة من المستخدمين الذين ينفذون مجموعة ثابتة من العمليات المتكررة] تستخدم من قبل موظفي البنوك، موظفي مكاتب الحجز...].



- تصمم لتقليل عدد ضغطات المفاتيح المطلوبة
- يمكن برمجة مفاتيح الوظائف لتنفيذ الأوامر بسرعة

- واجهات مسؤولي قواعد البيانات (Interfaces for the DBA) : تحتوي على أوامر خاصة لمسؤولي قواعد البيانات فقط [إدارة الحسابات، تعديل المخططات، ضبط إعدادات النظام، وإعادة تنظيم البنى التخزينية].
- تتيح صلاحيات متقدمة لاستخدام النظام وإدارته بكفاءة

```
CREATE USER 'user1' IDENTIFIED BY 'password';
```

```
ALTER TABLE Students ADD COLUMN Email VARCHAR(100);
```

```
CREATE INDEX idx_name ON Students(Name);
```



بناء وهيكلية أنظمة قواعد البيانات

العيوب	المزايا	نوع الواجهة
محدودية في إنشاء استعلامات معقدة	سهولة الاستخدام لا حاجة لحفظ الأوامر	القوائم
تتطلب إعدادا مسبقا من قبل المبرمجين	مناسبة لإدخال البيانات او استعلامات بسيطة	النماذج
قد تكون بطيئة في التعامل مع قواعد بيانات كبيرة	تفاعلية وسهلة الاستخدام.	الواجهات الرسومية
قد تفشل في تفسير الطلبات المعقدة	لا حاجة لتعلم لغة استعلام	اللغة الطبيعية
محدودة بالتطبيقات المبرمجة مسبقا	سرعة في الأداء	واجهات المستخدم المعياري
تتطلب خبرة تقنية عالية	صلاحيات شاملة لإدارة النظام	واجهات مسؤولي قواعد البيانات

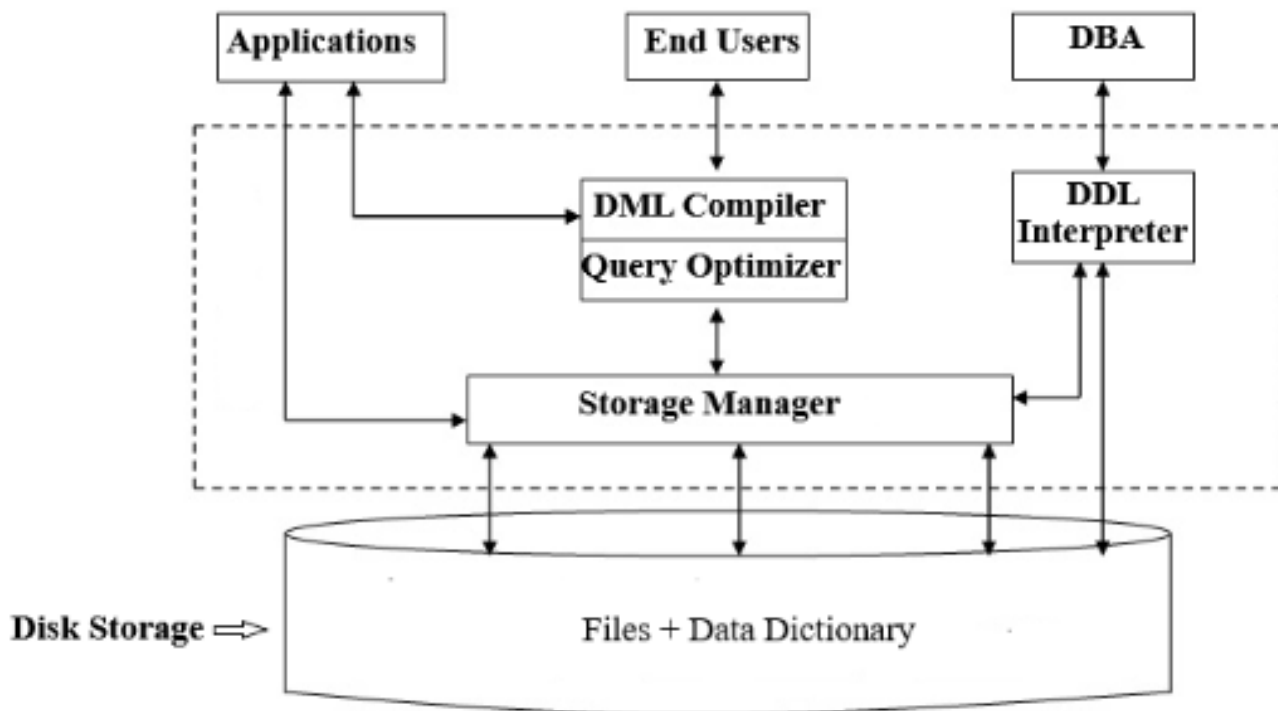
1. مكونات نظام إدارة قواعد البيانات (DBMS Component Modules) : نظام إدارة قواعد البيانات يتكون من مجموعة من الوحدات البرمجية التي تعمل معا لإدارة البيانات بكفاءة كل وحدة لها وظيفة محددة وتعمل هذه الوحدات بتناغم لتلبية احتياجات المستخدمين :
- مدير البيانات المخزنة (Stored Data Manager): يتحكم في الوصول إلى البيانات المخزنة على الأقراص وإدارة عمليات القراءة والكتابة بين الذاكرة الرئيسية والقرص
 - الكتالوج (System Catalog) : يقوم بتخزين البيانات مثل أسماء الجداول, أنواع البيانات, القيود ومعلومات الربط بين المخططات المختلفة.

- **مترجم لغة تعريف البيانات (DDL Compiler):** يقوم بمعالجة أوامر لغة تعريف البيانات لإنشاء أو تعديل المخططات وتخزين الوصف الناتج في الكatalog.
- **معالج قاعدة البيانات وقت التشغيل (Runtime Database Processor):** يقوم بمعالجة العمليات التي تتم أثناء تشغيل قاعدة البيانات و تنفيذ استعلامات القراءة والتحديث على البيانات يعتمد على "مدير البيانات المخزنة" للوصول إلى البيانات المخزنة على القرص.
- **مترجم لغة معالجة البيانات (DML Compiler):** يقوم بترجمة أوامر لغة معالجة البيانات إلى كود تنفيذي يمكن تشغيله و إرسال الكود الناتج إلى "معالج قاعدة البيانات وقت التشغيل".

• محسن الاستعلامات (Query Optimizer): هو مكون أساسي في أنظمة إدارة قواعد البيانات مسؤول عن

تحسين أداء تنفيذ الاستعلامات يعمل على اختيار أفضل خطة تنفيذ لاستعلام معين بناء على العوامل المختلفة

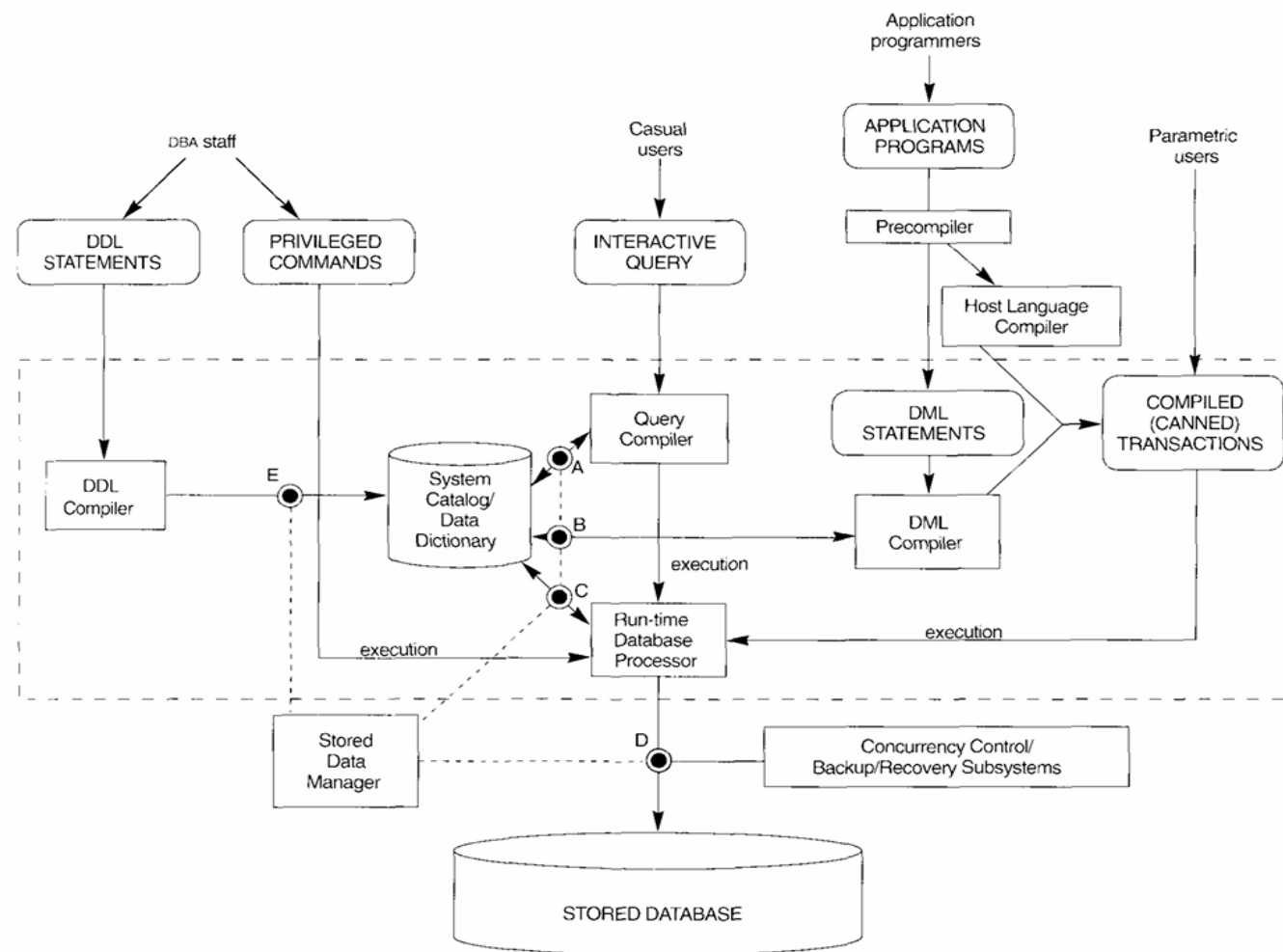
مثل بنية البيانات، الفهارس، وحجم البيانات.



- **مترجم الاستعلامات (Query Compiler):** ترجمة الاستعلامات المكتوبة بلغة (SQL) أو لغات استعلام مشابهة إلى شكل يمكن لنظام إدارة قواعد البيانات فهمه وتنفيذه ويتضمن عدة مراحل [التحليل, إعادة الكتابة, توليد الخطة].
- **ما قبل المترجم (Precompiler):** يقوم بأستخراج أوامر DML من البرامج المكتوبة بلغة البرمجة و إرسالها إلى المترجم لترجمتها ثم دمج الكود الناتج مع الكود الخاص بلغة البرمجة.



تصميم أنظمة قواعد البيانات



2. أدوات أنظمة قواعد البيانات (Database System Utilities) : بالإضافة إلى الوحدات الأساسية الموجودة في

نظام إدارة قواعد البيانات توجد أدوات إضافية تساعد في إدارة النظام بشكل أكثر فعالية.

- أداة التحميل (Loading Utility): تستخدم لتحميل البيانات الموجودة مسبقا (مثل ملفات نصية أو

ملفات متسلسلة) إلى قاعدة البيانات و تدعم نقل البيانات بين DBMS مختلفة

- أداة مراقبة الأداء (Performance Monitoring Utility): تراقب استخدام قاعدة البيانات وتجمع

الإحصائيات و تساعد مسؤول قاعدة البيانات في اتخاذ قرارات لتحسين الأداء و تحديد الاستعلامات

البطيئة أو النقاط العالقة .

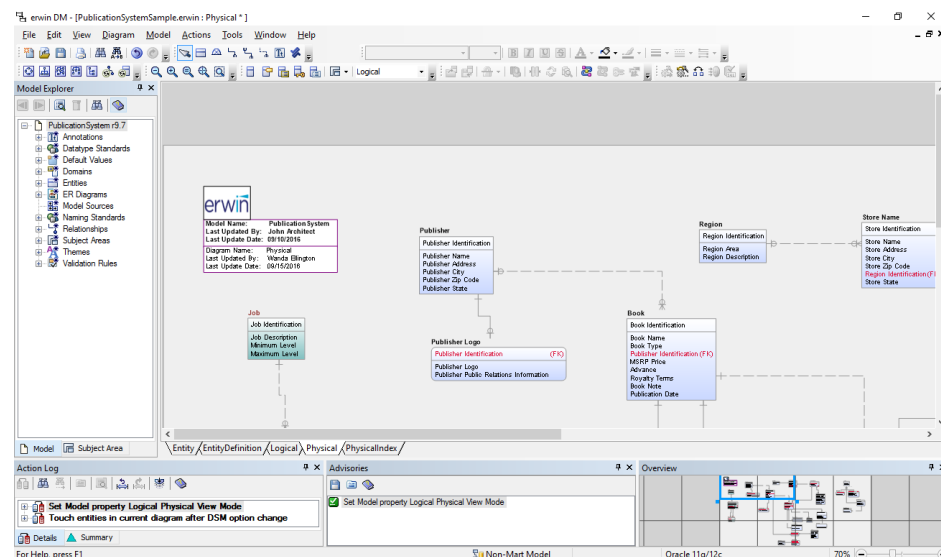
- أداة إعادة تنظيم الملفات (File Reorganization Utility): تستخدم لإعادة تنظيم الملفات المخزنة لتحسين أداء قاعدة البيانات وتحسين سرعة الوصول إلى البيانات و تقليل استهلاك الموارد.
- أداة النسخ الاحتياطي (Backup Utility): تستخدم لإنشاء نسخة احتياطية من قاعدة البيانات لضمان استعادة البيانات في حالة حدوث أعطال وتقسّم الى :
 - نسخة احتياطية كاملة (Full Backup) : نسخة شاملة لجميع بيانات قاعدة البيانات
 - نسخة احتياطية تدرجية (Incremental Backup): تخزن فقط التغييرات التي حدثت منذ آخر نسخة احتياطية

❖ يوجد ايضا بعض الأدوات المساعدة في أنظمة قواعد البيانات منها :

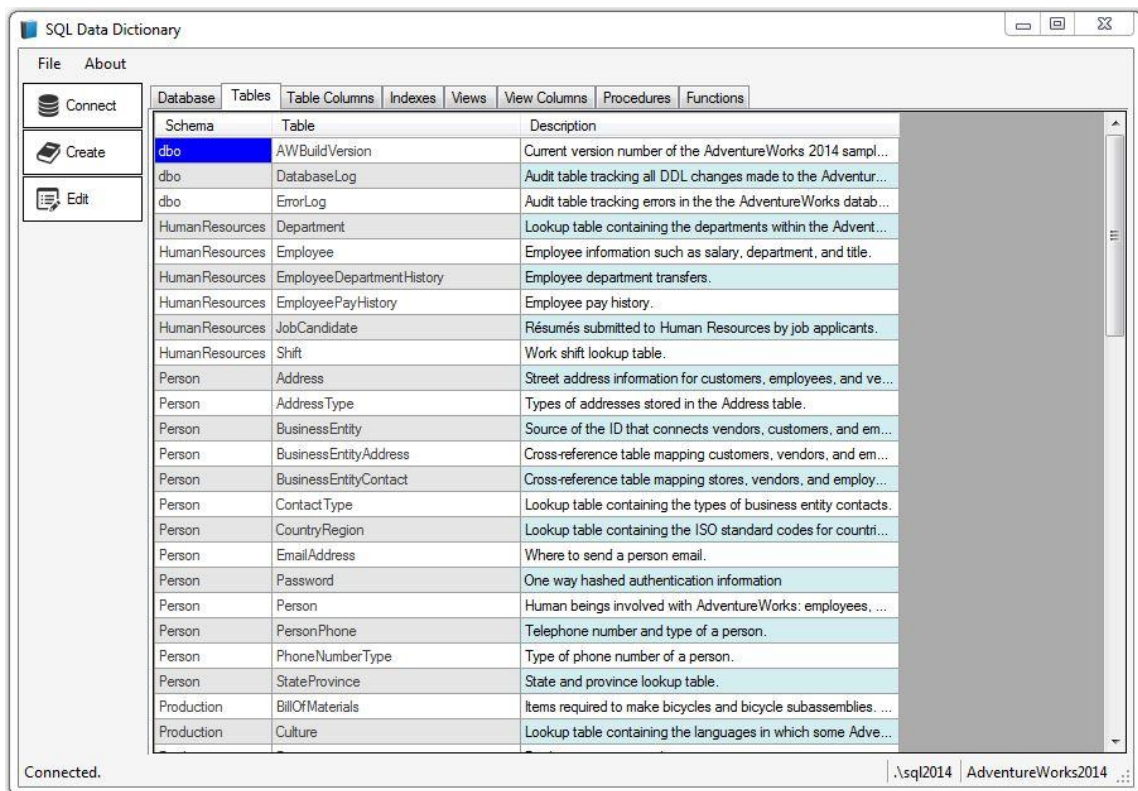
• أدوات هندسة البرمجيات بمساعدة الحاسوب (Computer-Aided Software Engineering): أدوات

تستخدم في مرحلة تصميم قواعد البيانات تساعد المصممين على إعداد المخططات وتحديد العلاقات بين

الجدول والقيود المفروضة [Erwin Data Modeler , IBM Rational Rose]



- قاموس البيانات الموسع (Expanded Data Dictionary): يعتبر مستودع معلومات يشمل معلومات عن المخططات, القيود, قرارات التصميم و معلومات المستخدمين والبرامج التطبيقية.



Schema	Table	Description
dbo	AWBuildVersion	Current version number of the AdventureWorks 2014 sample database.
dbo	DatabaseLog	Audit table tracking all DDL changes made to the AdventureWorks database.
dbo	ErrorLog	Audit table tracking errors in the AdventureWorks database.
HumanResources	Department	Lookup table containing the departments within the AdventureWorks database.
HumanResources	Employee	Employee information such as salary, department, and title.
HumanResources	EmployeeDepartmentHistory	Employee department transfers.
HumanResources	EmployeePayHistory	Employee pay history.
HumanResources	JobCandidate	Résumés submitted to Human Resources by job applicants.
HumanResources	Shift	Work shift lookup table.
Person	Address	Street address information for customers, employees, and vendors.
Person	AddressType	Types of addresses stored in the Address table.
Person	BusinessEntity	Source of the ID that connects vendors, customers, and employees.
Person	BusinessEntityAddress	Cross-reference table mapping customers, vendors, and employees to addresses.
Person	BusinessEntityContact	Cross-reference table mapping stores, vendors, and employees to contacts.
Person	ContactType	Lookup table containing the types of business entity contacts.
Person	CountryRegion	Lookup table containing the ISO standard codes for countries and regions.
Person	EmailAddress	Where to send a person email.
Person	Password	One way hashed authentication information.
Person	Person	Human beings involved with AdventureWorks: employees, customers, and vendors.
Person	PersonPhone	Telephone number and type of a person.
Person	PhoneNumberType	Type of phone number of a person.
Person	StateProvince	State and province lookup table.
Production	BillOfMaterials	Items required to make bicycles and bicycle subassemblies.
Production	Culture	Lookup table containing the languages in which some AdventureWorks content is available.

❖ يختلف عن الكتالوج بأنه يحتوي على معلومات أوسع يتم الوصول إليها بشكل أساسي من قبل المستخدمين ومسؤولي قواعد البيانات.

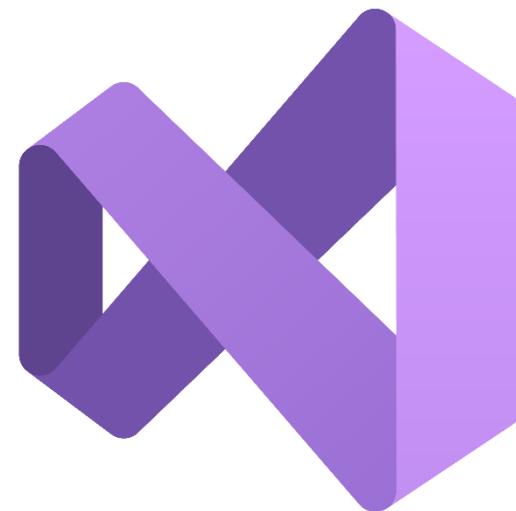
3. بيئات تطوير التطبيقات (Application Development Environments) : بيئات متكاملة تساعد المطورين على إنشاء تطبيقات تعتمد على قواعد البيانات و توفر أدوات تصميم قاعدة البيانات, واجهات المستخدم الرسومية , الاستعلام والتحديث و برمجة التطبيقات .



PowerBuilder



JBuilder



Microsoft Visual Studio

4. تسهيلات الاتصال (Communications Facilities) :

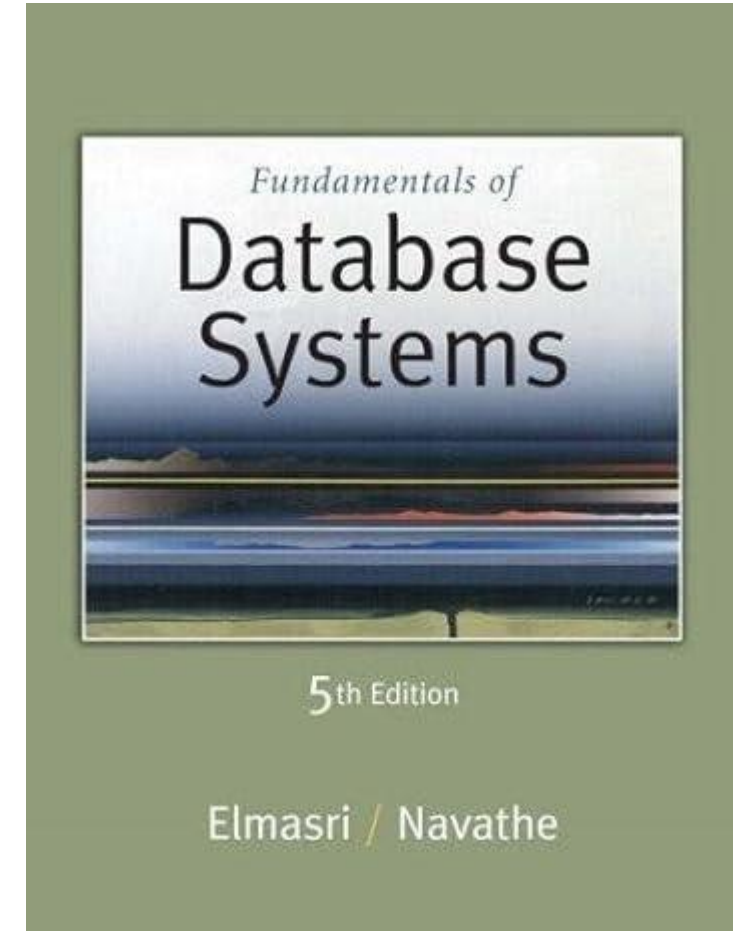
- برامج الاتصالات (Communications Software) : تتيح للمستخدمين الاتصال بقواعد البيانات عن بعد باستخدام أجهزة طرفية أو أجهزة كمبيوتر محلية أو شخصية عن طريق خطوط الهاتف , الشبكات أو الأقمار الصناعية .
- أنظمة قواعد البيانات/اتصالات البيانات (Database/Data Communications Systems): أنظمة تجمع بين نظام إدارة قواعد البيانات و برامج الاتصالات لتسهيل الوصول إلى البيانات تعمل على تحسين تكامل البيانات بين المستخدمين والنظام.

5. أنظمة قواعد البيانات الموزعة (Distributed DBMSs) : قواعد بيانات موزعة فعلياً على أجهزة متعددة و تتطلب شبكات اتصال لربط الأجهزة تستخدم الشبكات المحلية والواسعة للاتصال بين الأجهزة تفيد في تحسين الأداء من خلال توزيع الحمل و توزيع البيانات على أكثر من موقع .



عنوان الفيديو	الرابط
أساسيات قواعد البيانات	https://youtube.com/playlist?list=PL1DUmTEdeA6IWe4-wKJAIf-9_mLyVvcTE&feature=shared
قواعد البيانات	https://youtube.com/playlist?list=PL4n1Qos4Tb6RP_OovpgjoHLkCVaYFy-aj&feature=shared

* Fundamentals of database systems 5th Edition



شكرا لكم