



الأكاديمية العربية الدولية
Arab International Academy

إسم المادة: إدارة أنظمة قواعد البيانات

إسم المحاضر: م. خليل المحمد

الأكاديمية العربية الدولية – منصة أعد

تعريف نظم قواعد البيانات

نظم قواعد البيانات هي نظم مصممة لتخزين وإدارة البيانات بشكل منظم يمكن الوصول إليه بسهولة. تشمل نظم قواعد البيانات عدة أنواع، منها:

- 1. نظم قواعد البيانات العلائقية (RDBMS):** تعتمد على تنظيم البيانات في جداول مترابطة، حيث يُمكن تنفيذ الاستعلامات المعقدة باستخدام لغة SQL. أمثلة على هذه النظم تشمل MySQL و PostgreSQL.
- 2. نظم قواعد البيانات غير العلائقية (NoSQL):** تُستخدم لتخزين البيانات التي لا يمكن تمثيلها بسهولة في جداول. تشمل هذه النظم:
 - نموذج الوثائق: مثل MongoDB ، حيث يتم تخزين البيانات في مستندات JSON.
 - نموذج الكائنات: مثل ObjectDB ، حيث يتم تخزين البيانات ككائنات.
 - نموذج الرسوم البيانية: مثل Neo4j ، حيث يتم تنظيم البيانات في شكل رسوم بيانية لتسهيل استعلامات العلاقات.
- 3. نظم قواعد البيانات السحابية:** تقدم خدمات تخزين البيانات عبر الإنترنت، مما يتيح الوصول إليها من أي مكان.
- 4. نظم قواعد البيانات الزمنية:** تُستخدم لتخزين البيانات التي تتعلق بالوقت، مثل قياسات الأداء والبيانات الزمنية.

أهمية نظم قواعد البيانات في التطبيقات الحديثة

- 1. تخزين منظم وآمن للبيانات:** توفر نظم قواعد البيانات بيئة منظمة لتخزين البيانات، مما يسهل الوصول إليها وإدارتها. تتيح هذه النظم تخزين البيانات في جداول مترابطة، مما يقلل من احتمالية فقدان البيانات أو تكرارها.
- 2. تحسين الأداء:** تستخدم نظم قواعد البيانات تقنيات مثل الفهارس (Indexes) لتسريع عمليات البحث والاستعلام. تُساعد الفهارس في تقليل الوقت اللازم لاسترجاع البيانات، مما يؤدي إلى تحسين أداء التطبيق بشكل عام.
- 3. الدعم للاستعلامات المعقدة:** يمكن استخدام لغات الاستعلام مثل SQL لإجراء استعلامات معقدة وفعالة. يسمح هذا للمستخدمين بجمع وتحليل البيانات بسهولة، مما يسهل اتخاذ قرارات مبنية على بيانات دقيقة.
- 4. التوافر العالي والموثوقية:** توفر نظم قواعد البيانات ميزات مثل النسخ الاحتياطي والاسترداد، مما يضمن عدم فقدان البيانات في حالة حدوث أعطال. كما يمكن تكوينها للعمل بشكل متزامن عبر عدة خوادم لضمان التوافر المستمر.
- 5. التحكم في الوصول والأمان:** تتيح نظم قواعد البيانات إعداد سياسات تحكم دقيقة للوصول إلى البيانات، مما يضمن حماية المعلومات الحساسة من الوصول غير المصرح به. تُعتبر هذه الميزة ضرورية في التطبيقات التي تتعامل مع البيانات الشخصية أو المالية.

الفرق بين نظم قواعد البيانات العلائقية وغير العلائقية

1. نظم قواعد البيانات العلائقية: (RDBMS)

- **الهيكالية:**
تعتمد على الجداول (Tables) لتخزين البيانات، حيث تُنظم البيانات في صفوف وأعمدة، وكل جدول يمثل نوعًا معينًا من البيانات.
- **لغة الاستعلام:**
تستخدم لغة (SQL (Structured Query Language لتنفيذ الاستعلامات وإدارة البيانات، مما يوفر وسيلة قوية للتعامل مع البيانات المعقدة.
- **العلاقات:**
تدعم العلاقات بين الجداول من خلال المفاتيح الرئيسية (Primary Keys) والمفاتيح الأجنبية (Foreign Keys)، مما يتيح ربط البيانات بطريقة منظمة.
- **التطبيع:**
تتطلب نظم قواعد البيانات العلائقية عملية التطبيع لتقليل التكرار وضمان سلامة البيانات.
- **الاستخدامات:**
تُفضل في التطبيقات التي تتطلب مستوى عالٍ من التعقيد في البيانات والعلاقات، مثل أنظمة إدارة المؤسسات والأنظمة المالية.

الفرق بين نظم قواعد البيانات العلائقية وغير العلائقية

2. نظم قواعد البيانات غير العلائقية: NoSQL

- الهيكلية: لا تعتمد على الجداول، وتستخدم هياكل مختلفة مثل المستندات (Documents) أو الرسوم البيانية (Graphs) أو القيم المفتاحية (Key-Value) أو الأعمدة (Column-Family).
- المرونة: توفر مرونة أكبر في تخزين البيانات، حيث يمكن تخزين أنواع مختلفة من البيانات دون الحاجة إلى هيكل محدد مسبقاً.
- الاستعلام: تستخدم طرق استعلام مختلفة، مثل RESTful APIs، ولا تتطلب استخدام SQL، مما يسهل تكامل البيانات مع التطبيقات الحديثة.
- الأداء: غالباً ما توفر أداءً أفضل عند التعامل مع كميات كبيرة من البيانات، خاصة عند الحاجة إلى التوسع السريع.
- الاستخدامات: تُفضل في التطبيقات التي تتطلب أداءً عاليًا ومرونة في تخزين البيانات، مثل تطبيقات الويب الكبيرة، ووسائل التواصل الاجتماعي، وتحليل البيانات الضخمة.

متى يُفضل استخدام كل نوع؟

• نظم قواعد البيانات العلائقية:

- عندما تكون هناك حاجة لتعامل مع بيانات معقدة ذات علاقات متداخلة.
- في الحالات التي تتطلب تنفيذ استعلامات معقدة وإدارة قواعد بيانات صارمة.
- عندما تكون سلامة البيانات والتطبيع ضرورية، مثل في التطبيقات المالية أو أنظمة إدارة المؤسسات.

• نظم قواعد البيانات غير العلائقية:

- عندما تحتاج إلى تخزين كميات ضخمة من البيانات ذات الأنواع المتنوعة.
- في الحالات التي تتطلب أداءً عالياً وسرعة في التوسع.
- عندما تكون المرونة في تخزين البيانات أو التعامل مع البيانات المتغيرة بسرعة أساسية، مثل في تطبيقات الإنترنت الحديثة أو تطبيقات تحليل البيانات.

مكونات نظم قواعد البيانات

- 1. الجداول:(Tables) :** هي وحدات التخزين الأساسية في نظم قواعد البيانات العلائقية، حيث تُنظم البيانات في صفوف (Rows) وأعمدة.(Columns)
 - **الصفوف والأعمدة:** كل عمود يمثل خاصية أو حقل (Field) من البيانات، بينما يمثل كل صف سجلاً منفصلاً يحتوي على معلومات متعلقة بمادة معينة. على سبيل المثال، في جدول "الطلاب"، قد يحتوي العمود الأول على أسماء الطلاب، والعمود الثاني على أرقام الهواتف، وهكذا.
- 2. الفهارس:(Indexes) :** الفهارس هي هياكل بيانات تُستخدم لتسريع عمليات البحث والاستعلام في الجداول. تعمل الفهارس كقوائم مرجعية تسمح للوصول السريع إلى الصفوف التي تحتوي على بيانات معينة.
 - **الأنواع:** هناك عدة أنواع من الفهارس، مثل الفهارس الفريدة (Unique Indexes) التي تمنع تكرار القيم في عمود معين، وفهارس النصوص الكاملة (Full-Text Indexes) المستخدمة لتحسين البحث عن النصوص في البيانات.
- 3. العلاقات:(Relationships)**
 - **التعريف:** العلاقات تُشير إلى كيفية ارتباط الجداول المختلفة مع بعضها البعض. هذه العلاقات تُساعد في تنظيم البيانات بشكل يضمن عدم التكرار وبتيح سهولة الوصول إليها.

مكونات نظم قواعد البيانات

• أنواع العلاقات:

- علاقة واحد إلى واحد: **(One-to-One)** : حيث يمكن لسجل واحد في جدول A أن يرتبط بسجل واحد فقط في جدول B. مثال: كل طالب لديه بطاقة هوية واحدة.
 - علاقة واحد إلى متعدد: **(One-to-Many)** : حيث يمكن لسجل واحد في جدول A أن يرتبط بعدة سجلات في جدول B. مثال: يمكن لمعلم أن يُدرّس عدة طلاب.
 - علاقة متعدد إلى متعدد: **(Many-to-Many)** : حيث يمكن لسجل واحد في جدول A أن يرتبط بعدة سجلات في جدول B ، والعكس صحيح. مثال: الطلاب والدورات الدراسية حيث يمكن لكل طالب التسجيل في عدة دورات، وكل دورة يمكن أن تضم عدة طلاب.
4. **المخططات (Schemas)** : المخطط هو الهيكل التنظيمي الذي يحدد كيفية تنظيم الجداول والعلاقات بينها في قاعدة البيانات.
5. **القيود (Constraints)** : القيود هي قواعد تُفرض على البيانات لضمان جودتها وسلامتها. تشمل القيود مثل:
- **قيود المفتاح الأساسي (Primary Key)** : يُستخدم لضمان أن كل سجل في جدول يكون فريداً.
 - **قيود المفتاح الخارجي (Foreign Key)** : يُستخدم لربط الجداول والتأكد من أن القيم المرتبطة صحيحة.
 - **قيود القيمة الافتراضية (Default Constraint)** : يُحدد قيمة افتراضية لحقل معين إذا لم يتم إدخال قيمة.

نماذج البيانات - النموذج العلائقي

هيكل النموذج العلائقي:

- **الجدول:** تُعتبر الجداول هي الوحدات الأساسية لتخزين البيانات في النموذج العلائقي. كل جدول يتكون من:
 - **الصفوف (Rows):** تمثل كل صف سجلاً منفصلاً يحتوي على مجموعة من البيانات المرتبطة.
 - **الأعمدة (Columns):** تمثل كل عمود خاصية أو حقل (Field) من البيانات. يُحدد نوع البيانات المخزنة في كل عمود (مثل نصوص، أرقام، تواريخ).
- **المفتاح الأساسي (Primary Key):** هو عمود أو مجموعة من الأعمدة في الجدول تُستخدم لتمييز كل سجل بشكل فريد.
- **المفتاح الخارجي (Foreign Key):** هو عمود في جدول يُستخدم لإنشاء علاقة مع جدول آخر.
- **العلاقات بين الجداول:** يُعتبر النموذج العلائقي مثالاً لإدارة العلاقات بين الجداول. يُمكن أن تكون العلاقات:
 - **واحد إلى واحد (One-to-One):** حيث يتواجد سجل واحد في جدول مرتبط بسجل واحد في جدول آخر.
 - **واحد إلى متعدد (One-to-Many):** حيث يتواجد سجل واحد في جدول مرتبط بعدة سجلات في جدول آخر. على سبيل المثال، يمكن لمعلم واحد تدريس عدة طلاب.
 - **متعدد إلى متعدد (Many-to-Many):** حيث يُمكن أن يرتبط عدة سجلات في جدول بسجلات متعددة في جدول آخر. على سبيل المثال، يمكن لعدة طلاب التسجيل في عدة دورات.

نماذج البيانات - النموذج العلائقي

- جدول الطلاب:

Major	Age	Name	StudentID
علوم الحاسب	20	أحمد	1
هندسة	22	ليلى	2
علوم البيانات	21	محمد	3

- جدول الدورات:

Credits	CourseName	CourseID
3	برمجة تطبيقات الويب	101
3	قواعد البيانات	102
4	الشبكات	103

- جدول التسجيل (لربط الطلاب بالدورات):

CourseID	StudentID
101	1
102	1
103	2
101	3

مثال على النموذج العلائقي:

لنأخذ مثلاً بسيطاً على نموذج علائقي يحتوي على جدولين: "الطلاب" و"الدورات".

في هذا المثال، يُظهر جدول "التسجيل" كيفية الربط بين الطلاب والدورات، حيث يمكن لطالب واحد أن يسجل في عدة دورات، ويمكن لكل دورة أن تضم عدة طلاب.

نماذج البيانات - نموذج كائن-علاقة

- **تعريف:** يمثل نموذج كائن-علاقة كائنات البيانات ككائنات في برمجة الكائنات، مما يعني أن البيانات تُعتبر وحدات كاملة تتضمن كل من البيانات (السمات) والسلوك (الطرق أو الوظائف) المرتبطة بها.
- **الكائنات:** في هذا النموذج، تُعتبر الكائنات هي الوحدات الأساسية، والتي تتكون من:
 - **السمات (Attributes):** وهي البيانات الخاصة بكل كائن. **الطرق (Methods):** وهي الوظائف أو الإجراءات التي يمكن تنفيذها على الكائن.
- **تمثيل البيانات المعقدة:**
- **الأنواع المعقدة (Complex Types):** يدعم نموذج كائن-علاقة الأنواع المعقدة، مما يسمح بتخزين هياكل بيانات غير بسيطة.
- **العلاقات المتقدمة:** يُتيح نموذج كائن-علاقة أيضًا تمثيل العلاقات الأكثر تعقيدًا بين الكائنات. يُمكن أن تكون العلاقات بين الكائنات:
 - **علاقات وراثية (Inheritance):** حيث يمكن أن ترث الكائنات صفات وسلوكيات من كائنات أخرى. على سبيل المثال، يمكن أن يكون هناك كائن "طالب دراسات عليا" يرث من كائن "طالب".
 - **علاقات متعددة إلى متعددة:** يسمح هذا النموذج بتمثيل العلاقات بين الكائنات بشكل أكثر تعقيدًا، مثل العلاقة بين الطلاب والدورات، حيث يمكن لكل طالب أن يسجل في عدة دورات، ويمكن لكل دورة أن تضم عدة طلاب.

نماذج البيانات - نموذج كائن-علاقة

مثال على نموذج كائن-علاقة:

لنأخذ مثالاً على كيفية تمثيل كائن "طالب" في نموذج كائن-علاقة:

• كائن الطالب:

كائن: طالب

السمات:

الاسم: نص -

العمر: عدد صحيح -

التخصص: نص -

الدورات: قائمة من كائنات الدورة (مع كائن دورة يحتوي على: اسم الدورة، ورقم الدورة، والرصيد) -

الطرق:

عرض معلومات الطالب -

تسجيل في دورة -

• كائن الدورة

كائن: دورة

السمات:

اسم الدورة: نص -

رقم الدورة: عدد صحيح -

الرصيد: عدد صحيح -

نماذج البيانات - نموذج المستندات

```
{
  "student_id": "12345",
  "name": "أحمد",
  "age": 20,
  "major": "علوم الحاسب",
  "courses": [
    {
      "course_id": "101",
      "course_name": "برمجة تطبيقات الويب",
      "credits": 3
    },
    {
      "course_id": "102",
      "course_name": "قواعد البيانات",
      "credits": 3
    }
  ]
}
```

تعريف المستند: هو كائن يحتوي على مجموعة من القيم (Values) التي تمثل السمات (Attributes). يمكن أن يحتوي المستند على سمات بسيطة مثل النصوص والأرقام، بالإضافة إلى هياكل بيانات أكثر تعقيداً مثل القوائم (Arrays) والأشياء (Objects) المضمنة.

كيفية تخزين البيانات: هيكل المستند:

يتم تنظيم البيانات داخل المستندات بشكل مشابه للكيانات في نموذج البيانات الكائني. يمكن أن تكون المستندات متداخلة، حيث يحتوي مستند واحد على مستندات أخرى، مما يعكس العلاقات بين البيانات بشكل طبيعي.

مثال على مستند: لنأخذ مثالاً على مستند يمثل معلومات عن طالب:

في هذا المثال، يتم تمثيل الطالب كمستند يتضمن معلوماته الشخصية بالإضافة إلى قائمة بالدورات التي يتبعها.

نماذج البيانات - نموذج الرسوم البيانية

مفهوم نموذج الرسوم البيانية: يعتمد نموذج الرسوم البيانية على مفهوم الرسم البياني، حيث يتكون من **العُقد (Nodes)** و**الحواف (Edges)**. تمثل العُقد الكيانات (مثل الأشخاص، الأماكن، الأشياء)، بينما تمثل الحواف العلاقات بين هذه الكيانات.

- **العُقد**: كل عُقدة تمثل كائنًا أو كيانًا يمكن أن يحمل خصائص معينة. على سبيل المثال، في نموذج رسوم بياني للموظفين، يمكن أن تمثل كل عُقدة موظفًا وتحتوي على معلومات مثل الاسم، والوظيفة، والعمر.
- **الحواف**: تمثل الحواف العلاقات بين العُقد، ويمكن أن تكون موجهة أو غير موجهة. على سبيل المثال، يمكن أن تشير حافة من عُقدة "موظف" إلى عُقدة "قسم"، مما يدل على أن هذا الموظف ينتمي إلى ذلك القسم.

دور نموذج الرسوم البيانية في نظم قواعد البيانات:

- **تحليل العلاقات المعقدة**: يتميز نموذج الرسوم البيانية بقدرته على تحليل العلاقات المعقدة بين الكيانات بسهولة.
- **الأداء العالي**: يسمح هذا النموذج بعمليات استعلام فعالة، حيث يمكن استرجاع المعلومات المتعلقة بالعلاقات بسرعة كبيرة.
- **التعامل مع البيانات غير المهيكلة**: يُعتبر نموذج الرسوم البيانية مثاليًا للتعامل مع البيانات غير المهيكلة أو البيانات التي يصعب تمثيلها في نماذج أخرى مثل النماذج العلائقية.

نماذج البيانات - نموذج الرسوم البيانية

فوائد نموذج الرسوم البيانية:

- سهولة التوسع: يمكن بسهولة إضافة عُقد جديدة وعلاقات جديدة دون الحاجة إلى تعديل الهيكل الأساسي للبيانات، مما يجعل النموذج مرناً.
- تمثيل البيانات الديناميكية: يمكن أن تعكس الرسوم البيانية التغييرات في البيانات بشكل لحظي، مما يُعد ميزة قوية للتطبيقات التي تتطلب تحديثات سريعة.
- تحسين استعلامات البيانات: يُمكن استخدام خوارزميات متقدمة لتحليل الرسوم البيانية، مثل خوارزميات البحث، وخوارزميات إيجاد المسارات الأقصر، مما يحسن من أداء استعلامات البيانات.

أمثلة على التطبيقات:

- الشبكات الاجتماعية: تُستخدم نماذج الرسوم البيانية لتحليل الروابط بين الأصدقاء، وتوصيات الأصدقاء، ومراقبة الأنماط السلوكية.
- نظم التوصية: تُستخدم في توصيات المنتجات أو المحتوى بناءً على علاقات المستخدمين والمنتجات.
- تحليل الشبكات: تُستخدم في دراسة الشبكات مثل شبكات الاتصالات، وشبكات النقل، لتحليل كيفية تفاعل العناصر المختلفة.

نظم إدارة قواعد البيانات العلائقية

تعريف: نظم إدارة قواعد البيانات العلائقية (Relational Database Management Systems - RDBMS) هي نوع من نظم إدارة قواعد البيانات التي تعتمد على نموذج البيانات العلائقي. يُستخدم هذا النموذج لتنظيم البيانات في جداول (Tables) حيث تُعتبر كل جدول مجموعة من السجلات (Records) التي تحتوي على معلومات ذات صلة. تمثل RDBMS القاعدة الأساسية لتخزين واسترجاع البيانات في العديد من التطبيقات الحديثة، بما في ذلك تطبيقات الأعمال، والأنظمة المالية، والبرمجيات التجارية.

1. الخصائص الرئيسية لنظم RDBMS:

- **الهيكل الجدولي:** تُخزن البيانات في شكل جداول تحتوي على صفوف (Rows) وأعمدة (Columns) كل صف يمثل سجلاً فريداً، بينما كل عمود يمثل خاصية أو سمة من سمات ذلك السجل.
- **الروابط بين الجداول:** يُمكن الربط بين الجداول باستخدام المفاتيح (Keys)، حيث يُستخدم المفتاح الأساسي (Primary Key) لتحديد كل سجل بشكل فريد، بينما يُستخدم المفتاح الخارجي (Foreign Key) لإنشاء علاقات بين الجداول.
- **لغة الاستعلام البنوية: (SQL):** يتم استخدام لغة SQL لاستعلام البيانات وإجراء العمليات المختلفة على الجداول، مثل الإضافة (INSERT)، التحديث (UPDATE)، الحذف (DELETE)، والاستعلام (SELECT).

نظم إدارة قواعد البيانات العلائقية RDBMS

2. مكونات RDBMS:

- الجداول (Tables): تُعتبر الجداول هي البنية الأساسية لنظام RDBMS ، حيث تُخزن البيانات في صفوف وأعمدة. مثال على جدول "الموظفين":

- المفاتيح (Keys):

○ المفتاح الأساسي (Primary Key)

يُستخدم لتحديد كل سجل بشكل فريد في الجدول.

○ المفتاح الخارجي (Foreign Key): يُستخدم لإنشاء علاقة بين جدولين.

على سبيل المثال، يمكن أن يحتوي جدول "الموظفين" على مفتاح خارجي يشير إلى جدول "الأقسام".

- العلاقات (Relationships): تُحدد العلاقات بين الجداول، مثل العلاقات واحد إلى واحد

(One-to-One) أو واحد إلى متعدد (One-to-Many).

Department	Position	Name	Employee_ID
تقنية	مطور	أحمد	1
إدارة	مديرة	ليلى	2

Phone	Email	Name	CustomerID
0123456789	ahmed@example.com	أحمد	1
0987654321	leila@example.com	ليلى	2

أمثلة على نظم إدارة قواعد البيانات العلائقية RDBMS

MySQL.1: هو نظام إدارة قواعد بيانات مفتوح المصدر، يُستخدم على نطاق واسع في تطبيقات الويب، بما في ذلك أنظمة إدارة المحتوى مثل WordPress و Drupal.

- **المميزات: الأداء العالي:** يتميز MySQL بأداء جيد في استعلامات القراءة، مما يجعله مناسبًا للتطبيقات التي تحتاج إلى استجابة سريعة.
- **سهل الاستخدام:** يحتوي MySQL على واجهة سهلة الاستخدام، بالإضافة إلى العديد من أدوات الإدارة مثل phpMyAdmin.
- **دعم واسع:** يدعم مجموعة واسعة من أنظمة التشغيل ولغات البرمجة، مما يجعله خيارًا شائعًا بين المطورين.
- **المرونة:** يدعم أنواع متعددة من التخزين، مما يمنح المطورين القدرة على تخصيص أداء قاعدة البيانات.

PostgreSQL.2: هو نظام إدارة قواعد بيانات مفتوح المصدر يُعتبر من أكثر الأنظمة تقدمًا، ويدعم ميزات متقدمة مثل المعاملات المعقدة.

- **المميزات: دعم أنواع البيانات المتقدمة:** يدعم PostgreSQL أنواع بيانات مثل JSON و XML و Array، مما يجعله مناسبًا لتطبيقات البيانات المعقدة.
- **الامتداد:** يسمح بإنشاء أنواع بيانات مخصصة، مما يوفر للمطورين مرونة أكبر في تصميم قواعد البيانات.
- **التوافق مع ACID:** يلتزم PostgreSQL بمبادئ (Atomicity, Consistency, Isolation, Durability) ACID، مما يضمن سلامة البيانات.
- **الأداء العالي في الاستعلامات المعقدة:** يتميز بأداء قوي في التعامل مع الاستعلامات المعقدة، مما يجعله خيارًا جيدًا للتطبيقات التي تتطلب تحليل البيانات.

أمثلة على نظم إدارة قواعد البيانات العلائقية RDBMS

3. Oracle Database : هو نظام إدارة قواعد بيانات تجاري يُستخدم بشكل واسع في المؤسسات الكبيرة ويعتبر من الأقوى في السوق.

- **المميزات: الأداء المتقدم:** يُعرف بأدائه العالي وقدرته على معالجة كميات ضخمة من البيانات بكفاءة.
 - **الأمان المتقدم:** يحتوي على ميزات أمان متقدمة مثل التشفير والتحكم في الوصول المتقدم، مما يجعله مناسباً لتخزين البيانات الحساسة.
 - **إمكانية التوسع:** يدعم تدرجاً عاليًا في الأداء، مما يسمح بالتوسع بسهولة عند الحاجة.
 - **الميزات المتقدمة:** يوفر ميزات مثل تحليل البيانات المتقدم وإدارة البيانات غير المهيكلة.
- **4. Microsoft SQL Server :** هو نظام إدارة قواعد بيانات يُستخدم بشكل شائع في بيئات الأعمال، ويتميز بالتكامل مع أدوات Microsoft الأخرى.
 - **المميزات:**

- **التكامل:** يتكامل بسلاسة مع أدوات Microsoft مثل Excel و Power BI، مما يسهل تحليل البيانات.
- **الأمان والامتثال:** يقدم ميزات أمان قوية ومتطلبات التوافق مع المعايير التنظيمية.
- **التحليل المتقدم:** يوفر أدوات تحليل البيانات المتقدمة مثل Analysis Services.
- **واجهة المستخدم:** يتميز بواجهة مستخدم سهلة الاستخدام، مما يجعله مناسباً للمستخدمين من مختلف الخلفيات التقنية.

أمثلة على نظم إدارة قواعد البيانات العلائقية RDBMS

.5. SQLite: هو نظام إدارة قواعد بيانات خفيف الوزن ومضمّن، يُستخدم بشكل شائع في تطبيقات الهواتف المحمولة والمشاريع الصغيرة.

المميزات:

- **خفة الوزن:** لا يتطلب تثبيتًا منفصلاً ويعمل في ملف واحد، مما يجعله مثاليًا للتطبيقات الصغيرة.
- **سهولة الاستخدام:** سهل الإعداد والاستخدام، مما يجعله خيارًا جيدًا للمطورين الجدد.
- **أداء جيد:** يقدم أداءً جيدًا في عمليات القراءة، مما يجعله مناسبًا لتطبيقات الهواتف المحمولة.
- **متعدد المنصات:** يعمل على مجموعة متنوعة من أنظمة التشغيل والأجهزة.

.6. MariaDB: هو نظام إدارة قواعد بيانات مفتوح المصدر تم تطويره كمشروع بديل لـ MySQL بعد استحواذ Oracle على MySQL.

المميزات:

- **توافق مع MySQL:** متوافق مع MySQL ، مما يسهل التبديل بين النظامين.
- **أداء محسّن:** يقدم تحسينات في الأداء مقارنة بـ MySQL ، بما في ذلك ميزات تخزين جديدة.
- **الميزات المتقدمة:** يوفر ميزات إضافية مثل المحركات التخزينية الجديدة والعمليات الموزعة.

هيكل البيانات في RDBMS

إليك تفصيلاً لكيفية تنظيم البيانات داخل RDBMS:

1. **الجدول:(Tables):** تُعتبر الجداول هي الوحدة الأساسية في RDBMS ، حيث تُخزن البيانات في شكل صفوف وأعمدة.

• هيكل الجدول:

- **الصفوف:(Rows):** تمثل كل صف سجلاً فريداً يتضمن معلومات معينة عن الكيان. على سبيل المثال، يمكن أن يمثل كل صف معلومات عن عميل معين.
- **الأعمدة:(Columns):** تمثل كل عمود سمة أو خاصية من خصائص الكيان. على سبيل المثال، قد يكون لدى جدول العملاء أعمدة مثل CustomerID, Name, Email, وPhone.

2. **المفاتيح:(Keys)**

- **المفتاح الأساسي:(Primary Key):** يُستخدم لتحديد كل سجل بشكل فريد في الجدول. يجب أن تكون قيمته فريدة و غير فارغة (NOT NULL) في جدول العملاء، يمكن أن يكون CustomerID هو المفتاح الأساسي.
- **المفتاح الخارجي:(Foreign Key):** يُستخدم لإنشاء علاقة بين جدولين. يشير المفتاح الخارجي إلى المفتاح الأساسي في جدول آخر. على سبيل المثال، يمكن أن يحتوي جدول الطلبات على مفتاح خارجي CustomerID للإشارة إلى جدول العملاء.

هيكل البيانات في RDBMS

3. العلاقات (Relationships): تُحدد العلاقات بين الجداول من خلال المفاتيح. هناك أنواع مختلفة من العلاقات:

- علاقة واحد إلى واحد: (One-to-One) : كل سجل في جدول A يرتبط بسجل واحد فقط في جدول B. مثال: يمكن أن يكون لكل عميل حساب بنكي واحد.
- علاقة واحد إلى متعدد: (One-to-Many) : كل سجل في جدول A يمكن أن يرتبط بعدة سجلات في جدول B. مثال: يمكن أن يكون لكل عميل عدة طلبات.
- علاقة متعدد إلى متعدد: (Many-to-Many) : يمكن أن يرتبط كل سجل في جدول A بعدة سجلات في جدول B ، والعكس صحيح. مثال: يمكن أن يكون لدى الطلاب عدة دورات، ويمكن أن تحتوي كل دورة على عدة طلاب.

4. التطبيع (Normalization): هو عملية تنظيم البيانات داخل الجداول لتقليل التكرار وتحسين الكفاءة. تتضمن عملية التطبيع تقسيم الجداول إلى جداول أصغر مع إنشاء علاقات بينهما. يتم تصنيف التطبيع إلى عدة نماذج، مثل:

- النموذج الأول: (1NF) : يجب أن تكون جميع قيم الأعمدة ذرية (غير قابلة للتجزئة) وأن تكون هناك قيمة فريدة لكل صف.
- النموذج الثاني: (2NF) : يجب أن يكون كل عمود غير رئيسي مرتبطاً بالمفتاح الأساسي فقط.
- النموذج الثالث: (3NF) : يجب ألا تكون هناك تبعيات وظيفية غير مباشرة بين الأعمدة.

5. القيود (Constraints): القيود هي قواعد تُطبق على الجداول للتحكم في البيانات المخزنة. تشمل القيود الشائعة: **NOT NULL**: تمنع القيم الفارغة في عمود معين. **UNIQUE**: تضمن أن كل قيمة في عمود معين فريدة. **CHECK**: تضمن أن القيم تلبى شروطاً معينة. **FOREIGN KEY**: تحدد العلاقات بين الجداول وتضمن سلامة البيانات.

خصائص RDBMS:

1. مبادئ: ACID

- **التقابل: (Atomicity)** : تضمن هذه الخاصية أن تتم المعاملات ككتلة واحدة. إذا فشلت جزء من المعاملة، يتم التراجع عن جميع العمليات المرتبطة بها.
- **التماسك: (Consistency)** : تضمن هذه الخاصية أن تبدأ كل معاملة في حالة ثابتة وتصل إلى حالة ثابتة أخرى. يجب أن تلتزم جميع البيانات بقواعد معينة، مثل القيود (constraints) والمفاتيح.
- **العزل: (Isolation)** : تشير هذه الخاصية إلى أن معاملات متعددة يمكن أن تحدث في نفس الوقت دون التأثير على بعضها البعض.
- **الدوام: (Durability)** : تعني هذه الخاصية أن البيانات المدخلة في النظام ستظل محفوظة حتى في حالة حدوث فشل أو انقطاع.

2. دعم البيانات الهيكلية:

- يتم تنظيم البيانات في جداول، مما يسهل الوصول إليها واستعلامها. كل جدول يمثل كياناً محدداً، مما يجعل من السهل تنظيم المعلومات وفهمها.

3. القدرة على تنفيذ استعلامات معقدة:

- تدعم RDBMS استخدام لغة الاستعلام البنوية (SQL) لتنفيذ استعلامات معقدة. يمكن للمستخدمين كتابة استعلامات متعددة الجداول، واستخدام الدوال، وتجميع البيانات بطرق مختلفة.

خصائص RDBMS:

4. تعدد المستخدمين: يمكن للعديد من المستخدمين الوصول إلى قاعدة البيانات في نفس الوقت، مما يسمح بوجود بيانات عمل متعددة دون تعارض.
5. السلامة والأمان: تقدم RDBMS ميزات أمان متقدمة، بما في ذلك التحكم في الوصول، والتشفير، وخصائص الأمان الأخرى لضمان حماية البيانات.
6. النسخ الاحتياطي والاسترداد: توفر RDBMS آليات النسخ الاحتياطي والاسترداد لضمان استعادة البيانات في حالة الفشل أو الكوارث.
7. التوسع: يمكن توسيع RDBMS بسهولة لتلبية احتياجات الأعمال المتزايدة. تدعم العديد من الأنظمة عمليات التوزيع والنسخ المتعددة لتلبية متطلبات الأداء العالية.
8. تطبيع البيانات: تدعم RDBMS عملية التطبيع، التي تساعد في تقليل التكرار وتحسين سلامة البيانات من خلال تنظيم الجداول والعلاقات بينها.
9. التكامل مع الأنظمة الأخرى: يمكن دمج RDBMS بسهولة مع أنظمة أخرى، مما يسمح بتبادل البيانات بين التطبيقات المختلفة، مما يعزز من كفاءة العمليات.
10. دعم المعاملات: تدعم RDBMS المعاملات، مما يسمح بتنفيذ مجموعة من العمليات في خطوة واحدة. يضمن ذلك أن جميع العمليات تتم بنجاح أو يتم التراجع عنها بالكامل.

خطوات تصميم قاعدة البيانات

1. تحديد متطلبات النظام:

- جمع المعلومات: يجب على المصمم جمع المعلومات من المستخدمين النهائيين وأصحاب المصلحة لفهم متطلبات النظام.
- تحديد الأهداف: تحديد الأهداف الرئيسية التي يجب أن تحققها قاعدة البيانات، مثل تحسين أداء البحث، وضمان سلامة البيانات، ودعم العديد من المستخدمين.

2. تحديد الكيانات والعلاقات:

- تحديد الكيانات: حدد الكيانات الأساسية التي يجب أن تمثلها قاعدة البيانات. الكيان هو شيء يمكن أن يكون له بيانات مرتبطة، مثل "العميل" أو "المنتج".
- تحديد العلاقات: حدد كيف ترتبط الكيانات المختلفة مع بعضها. يمكن أن تكون العلاقات من نوع واحد إلى واحد، أو واحد إلى متعدد، أو متعدد إلى متعدد.

3. إنشاء نموذج الكيانات والعلاقات: (ER Model)

- رسم النموذج: استخدم أدوات تصميم مثل ERD (مخطط الكيانات والعلاقات) لإنشاء تمثيل بصري للكيانات والعلاقات.
- تحديد الخصائص: تحديد الخصائص الرئيسية لكل كيان (مثل الاسم، تاريخ الميلاد للعميل) وتحديد المفاتيح الأساسية والخارجية.

خطوات تصميم قاعدة البيانات

4. تطبيع البيانات:

- تحليل النموذج: تحليل نموذج ER والتأكد من أنه يتبع مبادئ التطبيع لتقليل التكرار وتحسين سلامة البيانات.
- تطبيق مستويات التطبيع: تطبيع البيانات إلى النموذج الأول (1NF) ، الثاني (2NF) ، والثالث (3NF) حسب الحاجة، مع تحديد متى يمكن أن يكون هناك تكرار مقبول.

5. تصميم الجداول:

- إنشاء الجداول: بناء الجداول بناءً على الكيانات المعروفة وتحديد الأعمدة اللازمة لكل جدول.
- تحديد الأنواع: تحديد نوع البيانات لكل عمود (مثل النص، الرقم، التاريخ) وتطبيق القيود اللازمة (مثل NOT NULL ، UNIQUE).

6. تحديد القيود والفهارس:

- تطبيق القيود: إضافة القيود اللازمة مثل القيود على المفاتيح الأساسية والخارجية، والقيود الأخرى (مثل CHECK).
- إنشاء الفهارس: إنشاء فهارس لتحسين أداء الاستعلامات، مع الأخذ بعين الاعتبار الأعمدة الأكثر استخدامًا في عمليات البحث.

خطوات تصميم قاعدة البيانات

7. إنشاء مخطط قاعدة البيانات:

- الرسم النهائي: تجميع كل العناصر في مخطط قاعدة البيانات النهائي، الذي يظهر جميع الجداول، العلاقات، والقيود بطريقة مرئية.
- مراجعة المخطط: استعرض المخطط مع الأطراف المعنية للحصول على التعليقات والملاحظات.

8. تطوير الوثائق:

- توثيق التصميم: إنشاء وثائق مفصلة حول التصميم، بما في ذلك وصف الجداول، العلاقات، القيود، والأهداف العامة.
- إعداد دليل المستخدم: إذا لزم الأمر، إعداد دليل مستخدم يشرح كيفية استخدام قاعدة البيانات.

9. اختبار التصميم:

- تنفيذ الاختبارات: اختبار التصميم من خلال تنفيذ استعلامات تجريبية للتحقق من أن قاعدة البيانات تعمل كما هو متوقع.
- تقييم الأداء: تقييم أداء قاعدة البيانات من حيث السرعة والقدرة على التعامل مع الاستعلامات.

10. التنفيذ والصيانة:

- تنفيذ قاعدة البيانات: إنشاء قاعدة البيانات الفعلية بناءً على التصميم والتأكد من إدخال البيانات بشكل صحيح.
- الصيانة والتحديث: تطوير خطة للصيانة الدورية والتحديثات لضمان استمرار عمل قاعدة البيانات بكفاءة.

التوجهات المستقبلية في إدارة قواعد البيانات

1. التحول إلى السحابة:

- زيادة الاعتماد على قواعد البيانات السحابية: يُتوقع أن تستمر الشركات في الانتقال إلى قواعد البيانات السحابية، مما يوفر لهم مرونة أكبر وموارد متاحة عند الطلب.
- تحسين الأداء والتقليل من التكاليف: ستساعد قواعد البيانات السحابية في تقليل التكاليف التشغيلية وتعزيز الأداء من خلال توفير بنية تحتية مرنة وسهلة التوسع.

2. الذكاء الاصطناعي والتعلم الآلي:

- تحليل البيانات الذكي: من المتوقع أن تُدمج تقنيات الذكاء الاصطناعي والتعلم الآلي في نظم إدارة قواعد البيانات لتحسين عمليات التحليل واستخراج البيانات.
- التنبؤ وتحسين الأداء: ستستخدم هذه التقنيات للتنبؤ بسلوك المستخدم وتحسين الأداء بناءً على البيانات التاريخية.

3. قواعد البيانات الموزعة:

- زيادة استخدام قواعد البيانات الموزعة: مع الحاجة إلى التعامل مع كميات ضخمة من البيانات، ستكون قواعد البيانات الموزعة أكثر شيوعاً، مما يسمح بتوزيع البيانات عبر عدة مواقع جغرافية.
- تحسين الاعتمادية: ستسهم هذه القواعد في تحسين الاعتمادية من خلال توفير إمكانية الوصول إلى البيانات حتى في حالة تعطل جزء من النظام.

التوجهات المستقبلية في إدارة قواعد البيانات

4. قواعد البيانات الزمنية:

- تطوير قواعد البيانات الزمنية: ستستمر قواعد البيانات الزمنية في النمو، حيث يُستخدمها الكثير من التطبيقات التي تحتاج إلى تتبع البيانات على مر الزمن، مثل أنظمة الرصد والتحليل المالي.
- تحليل البيانات الزمنية: ستزداد أهمية أدوات تحليل البيانات الزمنية لتحليل الاتجاهات والتغيرات على مدار الزمن.

5. زيادة التركيز على الأمان:

- تعزيز أمان البيانات: في ظل زيادة الهجمات الإلكترونية، سيستمر التركيز على تحسين أمان قواعد البيانات من خلال تقنيات مثل التشفير المتقدم وإدارة الهوية.
- الامتثال للمعايير: ستبقى الشركات ملزمة بالامتثال لمعايير حماية البيانات، مما سيؤثر على تصميم وإدارة قواعد البيانات.

6. تكامل البيانات:

- توجه نحو نظم تكامل البيانات: سيتزايد الطلب على نظم تكامل البيانات التي تجمع بين مصادر البيانات المتعددة، مما يسهل تحليل البيانات واستخراج الرؤى القيمة.
- الأدوات الذكية للتكامل: ستظهر أدوات جديدة تعتمد على الذكاء الاصطناعي لتسهيل عملية تكامل البيانات وتحليلها.



الأكاديمية العربية الدولية
Arab International Academy

خاتمة

أمل ان تكونوا قد حققتم الفائدة