

الوحدة التدريبية السادسة

المعرفة بقواعد البيانات العلاقية (SQL) او قواعد البيانات غير العلاقية (NoSQL) بالإضافة إلى نمذجة البيانات وتحسين الاستعلام

المتطلبات السابقة:

لا يوجد

نتائج التعلم:

عند الانتهاء من دراسة هذه الوحدة واكتساب مهاراتها الادائية والاتجاهات السلوكية الصحيحة والعلوم المهنية المرافقة من خلال التفاعل مع أنشطتها وخبراتها المختلفة يصبح المتدرب قادرا على اداء نتائج التعلم الآتية:

1. فهم قواعد البيانات العلائقية (SQL)
2. معرفة مفاهيم قواعد البيانات غير العلائقية (NoSQL)
3. تصميم نماذج فعالة لتخزين البيانات
4. تحسين أداء الاستعلامات، سواء على قواعد البيانات العلائقية أو غير العلائقية
5. مراعاة أمان قواعد البيانات وإدارتها

مصادر التعلم	الأنشطة التعليمية والتدريبية
الوحدة التدريبية	قراءة المعلومات النظرية
الشبكة العنكبوتية	البحث في المواقع الالكترونية التعليمية
المشغل / المختبر	تنفيذ التمارين العملية
سوق العمل	التدريب الميداني

روابط التعلم الإلكتروني

سيتم تزويد المتدربين بروابط التعلم الإلكتروني من خلال المدرب

مقدمة

يعد فهم تعقيدات قواعد البيانات العلائقية (SQL) وقواعد البيانات غير العلائقية (NoSQL) أمرًا أساسيًا لأي شخص مشارك في إدارة قواعد البيانات وتطويرها. تعتمد قواعد بيانات SQL ، المعروفة بتخزين البيانات المنظمة وقدرات الاستعلام القوية ، على مخطط محدد جيدًا وعلاقات بين الجداول للحفاظ على تكامل البيانات. في المقابل ، توفر قواعد بيانات NoSQL نماذج بيانات مرنة ، تستوعب مجموعة متنوعة من أنواع البيانات ومقاييسها ، مما يجعلها مثالية للتعامل مع البيانات غير المنظمة أو شبه المنظمة في التطبيقات التي تتطلب أداءً عاليًا وقابلية للتوسع.

بالإضافة إلى إتقان أنظمة قواعد البيانات هذه ، يعد الكفاءة في نمذجة البيانات وتحسين الاستعلام أمرًا ضروريًا. تضمن نمذجة البيانات الفعالة أن قواعد البيانات مصممة لتحقيق الأداء الأمثل وقابلية التوسع ، مما يقلل من التكرار ويحسن سرعة الوصول. تركز تقنيات تحسين الاستعلام ، سواء لقواعد بيانات SQL أو NoSQL ، على تحسين أداء الاستعلام وتقليل أوقات الاستجابة واستخدام الموارد بكفاءة. تعمل هذه المهارات معًا على تمكين متخصصي قواعد البيانات من بناء حلول تخزين بيانات قوية وفعالة وقابلة للتطوير ومصممة خصيصًا لتلبية احتياجات التطبيقات الحديثة.

فهم قواعد البيانات العلائقية (SQL) Structured Query Language

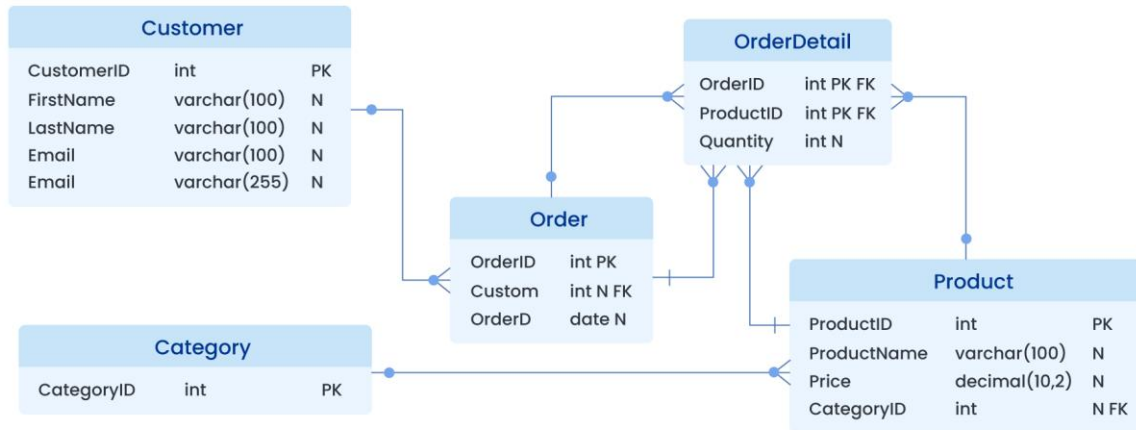
• تعريف قواعد البيانات العلائقية

- أ. شرح قواعد البيانات ودورها في تخزين البيانات.
- ب. تعريف قاعدة البيانات العلائقية: قاعدة بيانات مصممة للتعرف على العلاقات بين عناصر المعلومات المخزنة.
- ج. السياق التاريخي: تطوير قواعد البيانات العلائقية والمساهمين الرئيسيين (على سبيل المثال، EF Codd).

• الغرض من قواعد البيانات العلائقية

- أ. تنظيم البيانات: كيف تنظم قواعد البيانات العلائقية البيانات في الجداول.
- ب. استرجاع البيانات: الاستعلام واسترجاع البيانات بكفاءة.
- ج. سلامة البيانات: ضمان دقة واتساق البيانات من خلال القيود والقواعد.
- د. قابلية التوسع: القدرة على التعامل مع كميات كبيرة من البيانات والمستخدمين.

• المفاهيم الأساسية: الجداول والصفوف والأعمدة والعلاقات



الشكل رقم 1: الجداول والصفوف والأعمدة والعلاقات في قواعد البيانات

— الجداول (Tables)

- أ. التعريف: الجدول عبارة عن مجموعة من البيانات ذات الصلة المحفوظة بتنسيق منظم داخل قاعدة البيانات.
 - ب. المكونات: تتكون الجداول من صفوف وأعمدة.
- مثال: جدول "الموظفون" يحتوي على أعمدة مثل معرف الموظف والاسم والمنصب والقسم.

— الصفوف (Rows)

- أ. التعريف: يمثل الصف (أو السجل) عنصر بيانات واحد منظم ضمنياً في الجدول.
 - ب. الخصائص: يحتوي كل صف على بيانات لكيان واحد ويتبع مخطط الجدول.
- مثال: صف في جدول "الموظفون" يمثل تفاصيل موظف واحد.

— الأعمدة (Columns)

- أ. التعريف: العمود (أو الحقل) هو مجموعة من قيم البيانات من نوع معين، واحدة لكل صف في الجدول.
- ب. أنواع البيانات: يتم تعريف الأعمدة بأنواع بيانات محددة (على سبيل المثال، عدد صحيح، varchar، تاريخ).
عمود "الاسم" في جدول "الموظفون" والذي يحمل اسم كل موظف.

— العلاقات (Relationships)

- أ. التعريف: العلاقات هي ارتباطات بين الجداول في قاعدة بيانات علائقية.
- ب. أنواع العلاقات:
 - واحد لواحد (1:1): كل صف في الجدول (أ) مرتبط بصف واحد فقط في الجدول (ب).
 - واحد لمتعدد (1:M): يرتبط صف واحد في الجدول (أ) بصفوف متعددة في الجدول (ب).
 - متعدد إلى متعدد (M:M): يمكن ربط الصفوف الموجودة في الجدول (أ) والجدول (ب) بصفوف متعددة مع بعضها بعض.
 - التنفيذ: يتم إنشاء العلاقات باستخدام المفاتيح الأساسية والمفاتيح الخارجية.
 - مثال: جدول "الموظفون" وجدول "الأقسام" مع وجود مفتاح خارجي في "الموظفين" مرتبط بالمفتاح الأساسي في "الأقسام".

• أهمية قواعد البيانات في تطوير التطبيقات :

— دور قواعد البيانات في التطبيقات

- أ. تخزين البيانات: مستودع مركزي لتخزين بيانات التطبيق.
- ب. إدارة البيانات: إدارة عمليات البيانات (CRUD - إنشاء، قراءة، تحديث، حذف) داخل التطبيق.
- ج. استرجاع البيانات: الاستعلام عن البيانات واسترجاعها بكفاءة لاستخدام التطبيق.

— التكامل مع الأنظمة الأمامية والخلفية

- أ. قاعدة البيانات باعتبارها العمود الفقري: كيف تدعم قواعد البيانات واجهات المستخدم الأمامية والعمليات الخلفية.
- ب. واجهات برمجة تطبيقات البيانات: استخدام واجهات برمجة التطبيقات للتفاعل مع قواعد البيانات من مكونات الواجهة الأمامية والخلفية.
- مثال: تطبيق ويب بواجهة مستخدم (واجهة أمامية) وخادم (خلفية) يصل إلى قاعدة بيانات علائقية.

• أمثلة لتطبيقات العالم الحقيقي باستخدام قواعد البيانات العلائقية

- أ. مواقع التجارة الإلكترونية: إدارة مخزون المنتجات وبيانات العملاء والمعاملات.
- ب. أنظمة إدارة المحتوى (CMS): تخزين واسترجاع محتوى موقع الويب.
- ج. أنظمة تخطيط موارد المؤسسات (ERP): دمج البيانات عبر وظائف الأعمال المختلفة.
- د. أنظمة إدارة علاقات العملاء (CRM): إدارة معلومات العملاء وتفاعلاتهم.

• أساسيات SQL

- أ. التعريف: SQL (لغة الاستعلام الهيكلية) هي لغة برمجة قياسية لإدارة قواعد البيانات العلائقية ومعالجتها.
- ب. الأهمية: يعد SQL ضروريًا للاستعلام عن البيانات وتحديثها وإدارتها في قواعد البيانات العلائقية.
- ج. جمل SQL الأساسية :

— أمر التحديد (SELECT Command)

أ. بناء الجملة: "SELECT column1, column2, ... FROM table_name;"

ب. توضيح: Retrieves data from one or more columns in a table

مثال:

- Selecting all columns: "SELECT * FROM Employees;" ○
- Selecting specific columns: "SELECT Name, Position FROM Employees;" ○

— إدراج ال أمر (INSERT Command)

ج. بناء الجملة: "SELECT column1, column2, ... FROM table_name;"

د. توضيح: Retrieves data from one or more columns in a table

مثال:

- Selecting all columns: "SELECT * FROM Employees;" ○
- Selecting specific columns: "SELECT Name, Position FROM Employees;" ○

— أمر التحديث (UPDATE Command)

هـ. بناء الجملة:

"INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);"

و. توضيح: Adds new rows of data into a table

مثال :

"INSERT INTO Employees (Name, Position, Department) VALUES ('John Doe', ○
'Manager', 'Sales');"

— أمر الحذف (DELETE Command)

ز. بناء الجملة:

"UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;"

ح. توضيح: تعديل البيانات الموجودة في الجدول.

مثال : "UPDATE Employees SET Position = 'Senior Manager' WHERE EmployeeID = 1;"

• أنواع البيانات والقيود :

— أنواع البيانات :

أ. نظرة عامة: تحدد أنواع البيانات نوع البيانات التي يمكن تخزينها في عمود.

ب. أنواع البيانات الشائعة:

- عدد صحيح: الأعداد الصحيحة (على سبيل المثال، "INT"، "BIGINT").
- الحرف/السلسلة: بيانات نصية (على سبيل المثال، "CHAR"، "VARCHAR").
- التاريخ/الوقت: التواريخ والأوقات (على سبيل المثال، "DATE"، "DATETIME").
- العشري: أرقام النقطة الثابتة (على سبيل المثال، "DECIMAL"، "NUMERIC").

أمثلة:

- "Name VARCHAR(50)"
- "Salary DECIMAL(10, 2)"

— القيود (Constraints) :

أ. نظرة عامة: تفرض القيود قواعد على البيانات الموجودة في الجداول.

ب. القيود المشتركة:

- المفتاح الأساسي (Primary Key): يحدد بشكل فريد كل صف في الجدول.
- المفتاح الخارجي (Foreign Key): يضمن التكامل المرجعي بين الجداول.
- فريد (Unique): يضمن أن جميع القيم الموجودة في العمود فريدة.
- غير فارغ (Not Null): يضمن أن العمود لا يمكن أن يحتوي على قيمة فارغة.
- التحقق (Check): التأكد من استيفاء القيم الموجودة في العمود لشرط معين.

أمثلة:

- "PRIMARY KEY (EmployeeID)"
- "FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)"
- "UNIQUE (Email)"
- "NOT NULL"
- "CHECK (Age >= 18)"

• الإستعلامات في SQL ومجموعات النتائج :

— كتابة الاستعلامات الأساسية

- أ. استرجاع البيانات: استخدام "SELECT" للاستعلام عن البيانات.
- ب. تصفية البيانات: استخدام "WHERE" لتحديد الشروط.
- ج. فرز البيانات: استخدام "ORDER BY" لفرز مجموعات النتائج.

مثال:

- Basic retrieval: "SELECT * FROM Employees;"
- Filtering: "SELECT * FROM Employees WHERE Department = 'Sales';"
- Sorting: "SELECT * FROM Employees ORDER BY Name;"

• ربط الجداول (Joining Tables)

أ. التعريف: تجمع الصلات بين صفوف من جدولين أو أكثر بناءً على عمود ذي صلة.
ب. أنواع الصلات:

- الانضمام الداخلي (Inner Join): إرجاع صفوف ذات قيم متطابقة في كلا الجدولين.
 - الانضمام إلى اليسار (Left Join): إرجاع كافة الصفوف من الجدول الأيسر والصفوف المتطابقة من الجدول الأيمن.
 - الانضمام إلى اليمين (Right Join): إرجاع كافة الصفوف من الجدول الأيمن والصفوف المتطابقة من الجدول الأيسر.
 - الانضمام الكامل: (Full Join) إرجاع الصفوف عند وجود تطابق في أحد الجداول.
 - بناء الجملة والأمثلة:
- Inner join: "SELECT Employees.Name, Departments.DepartmentName FROM Employees INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;"
 - Left join: "SELECT Employees.Name, Departments.DepartmentName FROM Employees LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;"

• تجميع البيانات (Aggregating Data)

أ. استخدام الدوال التجميعية: "COUNT()", "SUM()", "AVG()", "MAX()", "MIN()".
ب. تجميع البيانات: استخدام "GROUP BY" لتجميع الصفوف التي تشترك في خاصية ما.
ج. تصفية المجموعات: استخدام "HAVING" لتصفية المجموعات بناءً على الوظائف المجمعة.
أمثلة:

- Counting rows: "SELECT COUNT() FROM Employees;"
- Summing values: "SELECT SUM(Salary) FROM Employees;"
- Grouping: "SELECT Department, COUNT() FROM Employees GROUP BY Department;"
- Filtering groups: "SELECT Department, COUNT() FROM Employees GROUP BY Department HAVING COUNT() > 5;"

• إدارة مجموعات النتائج (Managing Result Sets)

أ. ترقيم الصفحات: استخدام "LIMIT" و "OFFSET" لترقيم الصفحات النتائج.
- مثال:

"SELECT * FROM Employees LIMIT 10 OFFSET 20;"

• تقنيات SQL المتقدمة :

— مقدمة إلى عمليات الانضمام (Joins)

أ. التعريف: تجمع الصلات بين صفوف من جدولين أو أكثر بناءً على عمود مرتبط بينهما.
ب. الأهمية: يعد فهم الصلات أمرًا ضروريًا للاستعلام عن البيانات من جداول متعددة بكفاءة.

- الانضمام الداخلي (INNER Join)

أ. بناء الجملة: اختر أعمدة من الجدول 1 `INNER JOIN table2 ON table1.column = table2.column` ;
ب. توضيح: يقوم بإرجاع الصفوف ذات القيم المطابقة في كلا الجدولين فقط.
مثال:

```
sql
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

- الانضمام الأيسر ((LEFT Join (or LEFT OUTER Join))

أ. بناء الجملة `SELECT columns FROM table1 LEFT JOIN table2 ON table1.column = table2.column;`
ب. توضيح: يقوم بإرجاع كافة الصفوف من الجدول الأيسر والصفوف المطابقة من الجدول الأيمن. الصفوف غير المتطابقة من الجدول الأيمن سيكون لها قيم فارغة.
مثال:

```
sql
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

- الانضمام الأيمن ((RIGHT Join (or RIGHT OUTER Join))

أ. بناء الجملة `SELECT columns FROM table1 RIGHT JOIN table2 ON table1.column = table2.column;`
ب. توضيح: يقوم بإرجاع كافة الصفوف من الجدول الأيمن والصفوف المطابقة من الجدول الأيسر. الصفوف غير المتطابقة من الجدول الأيسر سيكون لها قيم فارغة.
مثال:

```
sql
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
RIGHT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

- الانضمام الكامل ((FULL Join (or FULL OUTER Join))

أ. بناء الجملة `SELECT columns FROM table1 FULL JOIN table2 ON table1.column = table2.column;`
ب. توضيح: يقوم بإرجاع الصفوف عند وجود تطابق في أحد الجداول. سيكون للصفوف غير المتطابقة قيم NULL في أعمدة الجدول بدون تطابق.

مثال:

```
sql
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
FULL JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

• الاستعلامات الفرعية والاستعلامات المتداخلة :

— مقدمة إلى الاستعلامات الفرعية

- أ. التعريف: الاستعلام الفرعي هو استعلام متداخل داخل استعلام آخر.
ب. الأنواع: استعلامات فرعية ذات صف واحد، واستعلامات فرعية متعددة الصفوف، واستعلامات فرعية مرتبطة.

— الاستعلامات الفرعية ذات الصف الواحد (Single-row Subqueries)

- أ. بناء الجملة: `SELECT columns FROM table WHERE column = (SELECT column FROM table`
`:(WHERE condition`
ب. توضيح: تقوم بإرجاع قيمة واحدة لاستخدامها في الاستعلام الرئيسي.
مثال:

```
sql
SELECT Name
FROM Employees
WHERE DepartmentID = (SELECT DepartmentID FROM Departments WHERE
DepartmentName = 'Sales');
```

— الاستعلامات الفرعية متعددة الصفوف (Multiple-row Subqueries)

- أ. بناء الجملة : `SELECT columns FROM table WHERE column IN (SELECT column FROM table`
`WHERE condition);`
ب. توضيح: تقوم بإرجاع قيم متعددة لاستخدامها في الاستعلام الرئيسي.

مثال:

```
sql
SELECT Name
FROM Employees
WHERE DepartmentID IN (SELECT DepartmentID FROM Departments WHERE Location =
'New York');
```

— الاستعلامات الفرعية المرتبطة (Correlated Subqueries)

- أ. بناء الجملة : `SELECT columns FROM table1 WHERE column1 = (SELECT`
`aggregate_function(column2) FROM table2 WHERE table1.column = table2.column);`
ب. توضيح: يشير الاستعلام الفرعي إلى أعمدة من الاستعلام الخارجي.

مثال:

```
sql
SELECT Name, Salary
FROM Employees e1
WHERE Salary > (SELECT AVG(Salary)
FROM Employees e2
WHERE e1.DepartmentID = e2.DepartmentID);
```

– الاستعلامات المتداخلة (Nested Queries)

توضيح: يمكن تداخل الاستعلامات الفرعية ضمن استعلامات فرعية أخرى.

مثال:

```
sql
SELECT Name
FROM Employees
WHERE DepartmentID = (SELECT DepartmentID
FROM Departments
WHERE Location = (SELECT Location
FROM Locations
WHERE LocationID = 1));
```

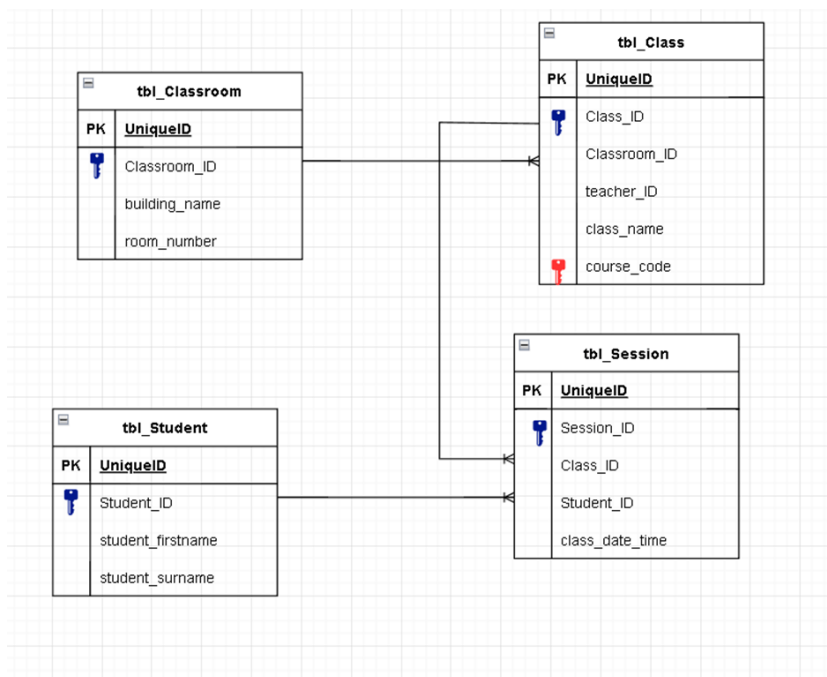
• الفهرسة لتحسين الأداء (Indexing for Performance Optimization)

– مقدمة إلى الفهارس

أ. التعريف: الفهارس هي كائنات قاعدة بيانات تعمل على تحسين سرعة عمليات استرجاع البيانات.

ب. الأنواع: الفهارس الأولية، والفهارس الفريدة، والفهارس غير الفريدة.

– إنشاء الفهارس



الشكل رقم 2: الفهارس في قواعد البيانات

- أ. بناء الجملة: إنشاء فهرس اسم على اسم الجدول (column1, column2, ...):
 ب. الشرح: أمر أساسي لإنشاء فهرس على أعمدة محددة.
 مثال:

```
sql
CREATE INDEX idx_employee_name ON Employees (Name);
```

- فوائد الفهرسة
 - أ. تحسين أداء الاستعلام: بحث واسترجاع وفزر أسرع.
 - ب. انخفاض عمليات الإدخال/الإخراج: وصول أقل إلى القرص لاستعادة البيانات.
 - اعتبارات الفهرسة (Considerations for Indexing)
 - أ. فهرسة النفقات العامة: زيادة تكاليف التخزين والصيانة.
 - ب. التأثير على عمليات الكتابة: عمليات الإدراج والتحديث والحذف أبطأ بسبب صيانة الفهرس.
 - أفضل ممارسات الفهرسة (Best Practices for Indexing)
 - أ. أعمدة الفهرس المستخدمة في عبارات WHERE و JOIN و ORDER BY.
 - ب. تجنب الإفراط في الفهرسة: اقتصار عدد الفهارس على الفهارس الأساسية.
 - ج. استخدام الفهارس المركبة: فهارس على أعمدة متعددة للاستعلامات المعقدة.
 - مراقبة الفهارس وتحسينها (Monitoring and Optimizing Indexes)
 - أ. مراقبة استخدام الفهرس بانتظام: تحديد الفهارس غير المستخدمة أو قليلة الاستخدام.
 - ب. إعادة بناء الفهارس أو إعادة تنظيمها: الحفاظ على كفاءة الفهرس.
- مثال: تحليل استخدام الفهرس

```
sql
SELECT
FROM sys.dm_db_index_usage_stats
WHERE database_id = DB_ID('your_database_name');
```

• مبادئ تصميم قواعد البيانات :


- التطبيع وعدم التطبيع
- ✓ التطبيع (Normalization)
 - أ. التعريف: التطبيع هو عملية تنظيم البيانات في قاعدة بيانات لتقليل التكرار وتحسين سلامة البيانات.
 - ب. أهداف التطبيع: التخلص من البيانات الزائدة عن الحاجة، والتأكد من أن تبعيات البيانات منطقية، وتبسيط الصيانة.
 - ج. الأشكال العادية:
- النموذج العادي الأول (1NF): يضمن أن كل عمود يحتوي على قيم ذرية وأن كل سجل فريد من نوعه.
- النموذج العادي الثاني (2NF): يتم تحقيقه عندما يكون الجدول في 1NF وتكون جميع الأعمدة غير الرئيسية معتمدة بشكل كامل على المفتاح الأساسي.
- النموذج العادي الثالث (3NF): يتم تحقيقه عندما يكون الجدول في 2NF وجميع الأعمدة تعتمد فقط على المفتاح الأساسي.

— نموذج Boyce-Codd العادي (BCNF): نسخة أكثر صرامة من 3NF، مما يضمن عدم وجود أية حالات شاذة.

✓ إلغاء التطبيع (Denormalization)

- أ. التعريف: إزالة التطبيع هي عملية إدخال التكرار عمدًا في قاعدة البيانات لتحسين أداء القراءة.
- ب. متى يتم إلغاء التطبيع:
- في السيناريوهات التي يكون فيها أداء القراءة أمرًا بالغ الأهمية، ويكون تناسق البيانات أقل أهمية.
 - المقايضات: الموازنة بين تحسين الأداء والأخطاء المحتملة في البيانات.
- أمثلة:

- إضافة جداول ملخصة لإعداد التقارير.
- دمج الجداول لتقليل عدد الصلات في الاستعلامات.



- استكشف أنواع مختلفة من قواعد بيانات NoSQL مثل مخازن المستندات، ومخازن القيمة الأساسية، وقواعد بيانات الرسوم البيانية.
- كيف تتعامل قواعد البيانات هذه مع البيانات بشكل مختلف مقارنة بقواعد البيانات العلائقية؟

التعرف على مفاهيم قواعد البيانات غير العلائقية (NoSQL)

تعريف قواعد بيانات NoSQL

- أ. التعريف: قواعد بيانات NoSQL هي قواعد بيانات غير علائقية مصممة لتخزين البيانات على نطاق واسع ولمعالجة البيانات الضخمة وتطبيقات الويب في الوقت الفعلي.
- ب. الخصائص: عدم وجود مخططات، والقياس الأفقي، والمرونة في نماذج البيانات.

أنواع قواعد بيانات NoSQL

- أ. مخازن المستندات (Document Stores)
- الوصف: تخزين البيانات في مستندات JSON أو BSON أو XML.
 - الميزات الرئيسية: مخطط مرن، سهل لتخزين البيانات الهرمية.
- أمثلة: CouchDB ، MongoDB .
- ب. متاجر القيمة الرئيسية (Key-Value Stores)
- الوصف: تخزين البيانات كمجموعة من أزواج القيمة الرئيسية.
 - الميزات الرئيسية: عمليات بحث بسيطة وسريعة ومثالية للتخزين المؤقت وإدارة الجلسة.
- أمثلة: ريديس ، رياك .
- ج. متاجر العمود العائلية (Column-Family Stores)
- الوصف: تخزين البيانات في أعمدة وصفوف، بشكل مشابه للجداول ولكن مع تحسين أداء القراءة والكتابة.
 - الميزات الرئيسية: فعال لمجموعات البيانات الكبيرة، ومصمم للتخزين الموزع.
- أمثلة: كاساندر، HBase .

د. قواعد بيانات الرسم البياني

- الوصف: تخزين البيانات كعقد وحواف، مثالية لتمثيل العلاقات.
- الميزات الرئيسية: ممتاز للاستعلامات المعقدة حول العلاقات، مثل الشبكات الاجتماعية.
- أمثلة: Neo4j، أمازون نيتون.

المقارنة مع قواعد البيانات العلائقية

- مرونة المخطط (Schema Flexibility)
 - أ. NoSQL: بدون مخطط، يمكنه التعامل مع هياكل البيانات المختلفة دون تعديل.
 - ب. SQL: مخطط ثابت، مطلوب بنية محددة مسبقًا.
- قابلية التوسع (Scalability)
 - أ. NoSQL: القياس الأفقي، وإضافة المزيد من الخوادم لتوزيع التحميل.
 - ب. SQL: القياس الرأسي، إضافة المزيد من الموارد إلى خادم واحد.
- اتساق البيانات (Data Consistency)
 - أ. NoSQL: الاتساق النهائي (AP من نظرية CAP).
 - ب. SQL: اتساق قوي (معاملات ACID).
- نموذج البيانات (Data Model)
 - أ. NoSQL: نماذج مختلفة (المستند، القيمة الرئيسية، عائلة الأعمدة، الرسم البياني).
 - ب. SQL: نموذج علائقي يحتوي على جداول وعلاقات محددة مسبقًا.
- لغة الاستعلام (Query Language)
 - أ. NoSQL: لغات استعلام مختلفة حسب نوع قاعدة البيانات.
 - ب. SQL: لغة الاستعلام القياسية (SQL).
- حالات الاستخدام (Use Cases)
 - أ. NoSQL: تطبيقات البيانات الضخمة، والتحليلات في الوقت الفعلي، وإدارة المحتوى، وإنترنت الأشياء، والشبكات الاجتماعية.
 - ب. SQL: التطبيقات التقليدية والاستعلامات المعقدة وأنظمة المعاملات.

حالات الاستخدام والفوائد في تطوير التطبيقات

- حالات الاستخدام (Use Cases)
 - أ. متاجر المستندات: أنظمة إدارة المحتوى، منصات التدوين، مواقع التجارة الإلكترونية.
 - ب. مخازن القيمة الأساسية: التخزين المؤقت، وإدارة الجلسة، وتخزين تفضيلات المستخدم.
 - ج. متاجر عائلة الأعمدة: التحليلات في الوقت الفعلي، وبيانات السلاسل الزمنية، وتسجيل الأحداث.
 - د. قواعد بيانات الرسم البياني: الشبكات الاجتماعية، محركات التوصية، كشف الاحتيال.

- الفوائد في تطوير التطبيقات

- أ. المرنة: يمكن لقواعد بيانات NoSQL استيعاب التغيرات في بنية البيانات وحجمها بسهولة، مما يجعلها مناسبة لبيئات التطوير السريعة.
- ب. قابلية التوسع: تم تصميمها للتوسع عن طريق إضافة المزيد من الخوادم، فهي تتعامل مع كميات كبيرة من البيانات وأحمال حركة المرور العالية بكفاءة.
- ج. NoSQL لحالات استخدام محددة، وغالبًا ما توفر عمليات قراءة وكتابة أسرع مقارنة بقواعد البيانات العلائقية التقليدية.
- د. فعالة من حيث التكلفة: يمكن أن يكون التوسع الأفقي أكثر فعالية من حيث التكلفة من ترقية خادم واحد، خاصة في البيئات السحابية.

- قواعد البيانات الموجهة نحو الوثائق (Document-Oriented Databases)

- مقدمة إلى مخازن المستندات

- أ. التعريف: تقوم قواعد البيانات الموجهة نحو المستندات بتخزين المعلومات الموجهة نحو المستندات واستردادها وإدارتها، عادةً بتنسيقات JSON أو BSON أو XML.
- ب. الخصائص الرئيسية: تخزين بيانات هرمية وخالية من المخططات ومستندات قائمة بذاتها.
- ج. متاجر المستندات الشهيرة: MongoDB ، CouchDB ، RavenDB .

MongoDB كمثال رائد

- أ. نظرة عامة: تعد MongoDB واحدة من أكثر قواعد البيانات الموجهة نحو المستندات استخدامًا.
- ب. الميزات: الأداء العالي والتوافر العالي وقابلية التوسع السهلة.
- ج. MongoDB : المجموعات والمستندات وتنسيق BSON.
- د. حالات الاستخدام: أنظمة إدارة المحتوى، والتحليلات في الوقت الفعلي، وتطبيقات إنترنت الأشياء، والمزيد.



- ما هي الطرق المستخدمة لجمع وتحليل متطلبات البيانات؟

تحديد وتشخيص متطلبات البيانات (Identifying Data Requirements)

جمع وتحليل متطلبات البيانات

• مقدمة لجمع متطلبات البيانات

- التعريف: عملية جمع وتحديد احتياجات البيانات لمشروع أو نظام.
- الأهمية: التأكد من أن البيانات التي تم جمعها تلبي احتياجات العمل وتدعم عمليات صنع القرار.

• طرق جمع متطلبات البيانات

- المقابلات

- إجراء مقابلات مع أصحاب المصلحة لفهم احتياجاتهم وتوقعاتهم من البيانات.
- قم بإعداد الأسئلة مسبقاً لتغطية جميع جوانب متطلبات البيانات.
- أمثلة على الأسئلة:

- ما نوع البيانات التي تحتاج إلى جمعها؟

- كم مرة ستحتاج للوصول إلى هذه البيانات؟

- ما هي نقاط البيانات الهامة لاتخاذ القرار الخاص بك؟

- المسوحات والاستبيانات (Surveys and Questionnaires)

- توزيع الاستطلاعات على جمهور أوسع لجمع متطلبات البيانات.
- استخدم الأسئلة المفتوحة والمغلقة للحصول على تعليقات شاملة.

مثال:

plaintext

1. Which data points are essential for your daily operations? (e.g., Customer ID, Order Date)
2. Rate the importance of data accuracy on a scale of 1 to 5.

- ورش العمل وجلسات العصف الذهني (Workshops and Brainstorming Sessions)

- تنظيم ورش عمل لتحديد متطلبات البيانات بشكل تعاوني.
- استخدم تقنيات العصف الذهني للكشف عن جميع احتياجات البيانات المحتملة.

مثال:

- إنشاء خريطة ذهنية تحتوي على فئات بيانات مختلفة (مثل بيانات العملاء وبيانات المبيعات وبيانات المخزون).

— ملاحظة

- مراقبة العمليات التجارية لفهم كيفية استخدام البيانات ومكان وجود فجوات في البيانات.
- تدوين ملاحظات حول نقاط إدخال البيانات وتدفقات البيانات واستخدام البيانات.

تحليل متطلبات البيانات

• تصنيف متطلبات البيانات

— تصنيف البيانات إلى فئات مثل البيانات التشغيلية والبيانات التحليلية والبيانات الرئيسية وبيانات المعاملات.

مثال:

- البيانات التشغيلية: معاملات المبيعات اليومية.

- البيانات التحليلية: تقارير المبيعات الشهرية.

• تحديد أولويات متطلبات البيانات

— تحديد أولوية كل متطلبات البيانات بناءً على تأثيرها على العمليات التجارية.

— استخدم تقنيات مثل MoSCoW (يجب أن يكون ، يمكن أن يكون ، لن يكون) لتحديد الأولويات.

مثال:

```
plaintext
Must have: Customer contact details.
Should have: Customer purchase history.
Could have: Customer feedback.
```

• التحقق من صحة متطلبات البيانات (Validate Data Requirements)

— التحقق من متطلبات البيانات المجمعة مع أصحاب المصلحة لضمان الدقة والاكتمال.

— استخدم تقنيات التحقق مثل الإرشادات والمراجعات.

• التعاون مع أصحاب المصلحة من أجل التشخيص الدقيق (Collaborating with Stakeholders for Accurate Diagnostics)

— تحديد أصحاب المصلحة الرئيسيين

- التعريف: أصحاب المصلحة هم أفراد أو مجموعات مهتمة بنتائج بيانات المشروع.

- أمثلة: محللو الأعمال، ومستخدمو البيانات، وموظفو تكنولوجيا المعلومات، والإدارة، والشركاء الخارجيون.

— إشراك أصحاب المصلحة في عملية متطلبات البيانات

○ اجتماعات منتظمة

— جدول اجتماعات منتظمة مع أصحاب المصلحة لمناقشة احتياجات البيانات والتقدم المحرز.

— استخدم جداول أعمال الاجتماعات للحفاظ على المناقشات مركزة ومنتجة.

مثال:

```
plaintext
Meeting Agenda:
1. Review of current data requirements.
2. Discussion of any new data needs.
3. Validation of existing data requirements.
```

○ آليات التغذية الراجعة (Feedback Mechanisms)

– تنفيذ حلقات التعليقات لجمع مدخلات أصحاب المصلحة بشكل مستمر.

– استخدام أدوات مثل نماذج التعليقات وصناديق الاقتراحات ومنصات التعليقات الرقمية.

مثال:

– نموذج الملاحظات عبر الإنترنت حيث يمكن لأصحاب المصلحة تقديم متطلبات البيانات أو مخاوفهم.

– أدوات التعاون

○ استخدام أدوات التعاون مثل برامج إدارة المشاريع (مثل Trello و Asana) ومنصات الاتصال (مثل Slack

و Microsoft Teams) لتسهيل تفاعل أصحاب المصلحة.

مثال:

– إنشاء لوحة Trello مشتركة لتتبع متطلبات البيانات وتعليقات أصحاب المصلحة.

– تقنيات التعاون الفعال

○ جلسات تطوير التطبيقات المشتركة (JAD).

➤ تنظيم جلسات JAD حيث يقوم أصحاب المصلحة بشكل تعاوني بتطوير متطلبات البيانات.

➤ استخدام الجلسات الميسرة لضمان سماع جميع الأصوات.

مثال:

– جدول أعمال جلسة JAD: العصف الذهني لمتطلبات البيانات، وتحديد الأولويات، والتحقق من صحتها.

– تطوير حالة الاستخدام

○ تطوير حالات الاستخدام لتوضيح كيفية استخدام البيانات في سيناريوهات العالم الحقيقي.

○ مشاركة حالات الاستخدام مع أصحاب المصلحة للتحقق من صحة متطلبات البيانات وتحسينها.

مثال:

plaintext
Use Case: Generating Monthly Sales Reports
- Actor: Sales Manager
- Description: The Sales Manager generates monthly sales reports to analyze sales performance and make strategic decisions.
- Data Requirements: Sales data, product details, customer demographics.

توثيق احتياجات البيانات والمواصفات

● إنشاء مستندات متطلبات البيانات

– مواصفات متطلبات البيانات (DRS)

✓ تحديد غرض ونطاق وسياق متطلبات البيانات.

✓ قائمة متطلبات البيانات التفصيلية، بما في ذلك مصادر البيانات وأنواع البيانات ومعايير جودة البيانات.

مثال:

plaintext
Data Requirement Specification (DRS)
Purpose: To define the data requirements for the customer relationship management (CRM) system.
Scope: Customer data, transaction data, interaction history.
Data Requirements:
- Customer Data: Name, contact details, demographics.
- Transaction Data: Purchase history, order details, payment information.
- Interaction History: Customer service interactions, feedback.

- قاموس البيانات (Data Dictionary)

- ✓ إنشاء قاموس بيانات لتقديم تعريفات وأوصاف تفصيلية لعناصر البيانات.
- ✓ تضمين سمات مثل اسم الحقل ونوع البيانات والوصف والقيود.

مثال:

plaintext
Data Dictionary
Field Name: customer_id
Data Type: INT
Description: Unique identifier for each customer.
Constraints: Primary key, auto-increment.
Field Name: email
Data Type: VARCHAR(100)
Description: Customer's email address.
Constraints: Unique, not null.

• ضمان الوضوح والاتساق (Clarity and Consistency) في التوثيق

- قوالب موحدة (Standardized Templates)

- ✓ استخدام قوالب موحدة لتوثيق متطلبات البيانات لضمان الاتساق.
- مثال: قم بإنشاء قالب يحتوي على أقسام لوصف البيانات ومصدرها وتنسيقها واستخدامها.
- ✓ التحكم في الإصدار
- تنفيذ التحكم في الإصدار لتتبع التغييرات والتحديثات على مستندات متطلبات البيانات.
- مثال: استخدم أرقام الإصدارات واحتفظ بسجل التغيير لكل مستند.

• عملية المراجعة والموافقة (Review and Approval Process)

- مراجعة أصحاب المصلحة

- ✓ مشاركة متطلبات البيانات الموثقة مع أصحاب المصلحة للمراجعة والتعليقات.
- ✓ دمج تعليقات أصحاب المصلحة ووضع اللمسات الأخيرة على الوثائق.
- مثال: أرسل وثيقة DRS إلى أصحاب المصلحة عبر البريد الإلكتروني للمراجعة والموافقة.

- سير عمل الموافقة (Approval Workflow)

- ✓ إنشاء سير عمل الموافقة للموافقة رسميًا على متطلبات البيانات.
- مثال: قم بإنشاء سلسلة اعتماد حيث تتم مراجعة المستند من قبل محلل أعمال، يليه فريق تكنولوجيا المعلومات، وأخيراً تتم الموافقة عليه من قبل مدير المشروع.

• أدوات وتقنيات التشخيص (Diagnostic Tools and Techniques)

✓ أدوات التشخيص المستندة إلى SQL

- الغرض: تساعد أدوات تشخيص SQL في مراقبة أداء قواعد بيانات SQL وتحليله وتحسينه.
- الأهمية: ضمان كفاءة عمليات قاعدة البيانات، وتقليل وقت التوقف عن العمل، وتعزيز إمكانية الوصول إلى البيانات.

✓ أدوات تشخيص SQL الشائعة

- ستوديو إدارة خادم (SQL Server Management Studio (SSMS))

- الميزات: تنفيذ الاستعلام، ومراقبة الأداء، وإدارة قاعدة البيانات.
- مثال: استخدام SSMS لتحديد الاستعلامات بطيئة التشغيل.

```
plaintext
SELECT TOP 10
    qs.sql_handle,
    qs.execution_count,
    qs.total_worker_time AS CPU_Time,
    qs.total_elapsed_time AS Elapsed_Time,
    qs.total_logical_reads AS Logical_Reads,
    SUBSTRING(qt.text, (qs.statement_start_offset/2) + 1,
        ((CASE statement_end_offset
            WHEN -1 THEN DATALENGTH(qt.text)
            ELSE qs.statement_end_offset END
            - qs.statement_start_offset)/2) + 1) AS QueryText
FROM
    sys.dm_exec_query_stats qs
```

```
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt
ORDER BY
qs.total_worker_time DESC;
```

-أوراكل(Oracle Enterprise Manager)

- الميزات: مراقبة شاملة وتشخيص وضبط لقواعد بيانات Oracle.
- مثال: استخدام تقارير مستودع عبء العمل التلقائي (AWR) لتحليل أداء قاعدة البيانات.

- منصة MySQL (MySQL Workbench)

- الميزات: تصميم قاعدة البيانات، وتطوير SQL، وضبط الأداء.
- مثال: استخدام MySQL Workbench لتحديد الاستعلامات البطيئة وتحسينها

```
plaintext
EXPLAIN SELECT * FROM orders WHERE order_date = '2022-01-01';
```

• سياسات إدارة البيانات

خصوصية البيانات والامتثال (Data Privacy and Compliance)

مقدمة لخصوصية البيانات والامتثال

- التعريف: تضمن خصوصية البيانات التعامل مع البيانات الشخصية وفقاً للمتطلبات القانونية والتنظيمية.
- يتضمن الامتثال: الالتزام بالقوانين واللوائح التي تحكم حماية البيانات.
- الأهمية: حماية خصوصية الأفراد، وبناء الثقة، وتجنب العقوبات القانونية.

اللائحة العامة لحماية البيانات (General Data Protection Regulation (GDPR))

- نظرة عامة: اللائحة العامة لحماية البيانات (GDPR) هي لائحة في الاتحاد الأوروبي (EU) تحمي البيانات الشخصية والخصوصية للأفراد.
- المتطلبات الرئيسية:
 - الموافقة (Consent): الحصول على موافقة صريحة من الأفراد قبل معالجة بياناتهم.
 - حق الوصول (Right to Access): السماح للأفراد بالوصول إلى بياناتهم ومعرفة كيفية استخدامها.
 - الحق في المحو (Right to Erasure): السماح للأفراد بطلب حذف بياناتهم.
 - إمكانية نقل البيانات (Data Portability): تمكين الأفراد من نقل بياناتهم بين مقدمي الخدمة.
 - إشعار خرق البيانات (Data Breach Notification): قم بإخطار السلطات والأفراد المتأثرين بانتهاكات البيانات في غضون 72 ساعة.

مثال:

```
plaintext
- Ensure explicit consent forms are used when collecting personal data.
- Implement mechanisms for individuals to request access to or deletion of their data.
```

قانون قابلية نقل التأمين الصحي والمساءلة (Health Insurance Portability and Accountability Act) ((HIPAA)

- نظرة عامة: HIPAA هو قانون أمريكي يحمي السجلات الطبية للأفراد والمعلومات الصحية الشخصية الأخرى.
- المتطلبات الرئيسية:
 - قاعدة الخصوصية (Privacy Rule): تحمي خصوصية المعلومات الصحية للأفراد.
 - القاعدة الأمنية (Security Rule): تضع معايير تأمين المعلومات الصحية الإلكترونية للأفراد.
 - قاعدة الإخطار بالانتهاك (Breach Notification Rule): تتطلب الإخطار بانتهاكات المعلومات الصحية غير الآمنة.

مثال:

plaintext

- Implement access controls to limit who can view and use health information.
- Use encryption to protect health information stored electronically.

• أفضل ممارسات الامتثال (Compliance Best Practices)

- ✓ مراجعة سياسات الخصوصية وتحديثها بانتظام.
- ✓ إجراء عمليات تدقيق منتظمة لضمان الامتثال للوائح ذات الصلة.
- ✓ تدريب الموظفين على ممارسات خصوصية البيانات وأمنها.
- مثال: تنفيذ تقييمات تأثير حماية البيانات (DPIAs) لتحديد مخاطر الخصوصية والتخفيف منها.

• استراتيجيات النسخ الاحتياطي والاسترداد

✓ أهمية النسخ الاحتياطي والاسترداد

- يضمن توفر البيانات وسلامتها في حالة تعطل الأجهزة أو تلف البيانات أو غيرها من الكوارث.
- حاسم لاستمرارية الأعمال والتعافي من الكوارث.

✓ أنواع النسخ الاحتياطية

- النسخ الاحتياطي الكامل: نسخة كاملة من قاعدة البيانات بأكملها.
- مثال: عمل نسخة احتياطية كاملة لقاعدة بيانات SQL.

sql

```
BACKUP DATABASE myDatabase TO DISK = 'C:\backups\myDatabase.bak';
```

- النسخ الاحتياطي التزايد (Incremental Backup): ينسخ فقط البيانات التي تغيرت منذ آخر نسخة احتياطية.

مثال: استخدام rsync للنسخ الاحتياطي المتزايد في قاعدة بيانات NoSQL.

```
shell
rsync -av --progress /data/ /backup/
```

– النسخ الاحتياطي التفاضلي (Differential Backup): نسخ جميع التغييرات التي تم إجراؤها منذ آخر نسخة احتياطية كاملة.

مثال: استخدام النسخ الاحتياطية التفاضلية لـ MySQL.

```
shell
mysqldump --single-transaction --quick --lock-tables=false --all-databases -
-flush-logs > backup.sql
```

✓ استراتيجيات النسخ الاحتياطي

- الجد والأب والابن (Grandfather-Father-Son (GFS)): نظام النسخ الاحتياطي التناوبي الذي يتضمن نسخًا احتياطيًا يوميًا وأسبوعيًا وشهريًا.
- قاعدة النسخ الاحتياطي (Backup Rule): احتفظ بثلاث نسخ من البيانات (الأساسية ونسختين احتياطيتين) على وسيطتين مختلفتين، بنسخة واحدة خارج الموقع.
- الحماية المستمرة للبيانات (Continuous Data Protection (CDP)): النسخ الاحتياطي في الوقت الحقيقي الذي يلتقط كل تغيير في البيانات.

مثال:

```
plaintext
- Perform daily incremental backups, weekly differential backups, and
monthly full backups.
- Store backups on local disk, external storage, and cloud storage.
```

• استراتيجيات التعافي (Recovery Strategies)

- استرداد نقطة زمنية (Point-in-Time Recovery): استعادة قاعدة البيانات إلى نقطة زمنية محددة.
- خطة التعافي من الكوارث (Disaster Recovery Plan): عملية موثقة لاستعادة البيانات واستئناف العمليات بعد وقوع الكارثة.
- الاختبار والتحقق من الصحة (Testing and Validation): اختبار إجراءات النسخ الاحتياطي والاسترداد بانتظام للتأكد من أنها تعمل كما هو متوقع.

مثال: استعادة قاعدة بيانات MongoDB من نسخة احتياطية.

```
shell
mongorestore --drop --db myDatabase /backup/myDatabase
```

إدارة دورة حياة البيانات وسياسات الاحتفاظ بها (Data Lifecycle Management and Retention Policies)

مقدمة لإدارة دورة حياة البيانات ((Data Lifecycle Management (DLM))

- التعريف: DLM هي عملية إدارة البيانات بدءًا من الإنشاء وحتى الحذف، مما يضمن جودة البيانات وأمنها وامتثالها طوال دورة حياتها.
- الأهمية: يضمن إدارة البيانات بكفاءة، ويقلل تكاليف التخزين، ويحافظ على الامتثال للوائح.

مراحل دورة حياة البيانات

- الإنشاء والاكتمال: يتم إنشاء البيانات أو الحصول عليها من مصادر مختلفة.
 - التخزين والصيانة: يتم تخزين البيانات وصيانتها، مما يضمن إمكانية الوصول إليها وسلامتها.
 - الاستخدام والتحليل: يتم استخدام البيانات وتحليلها لاتخاذ القرار.
 - الأرشفة: يتم نقل البيانات التي يتم الوصول إليها بشكل غير متكرر إلى التخزين طويل المدى.
 - الحذف: يتم حذف البيانات بشكل آمن عند عدم الحاجة إليها.
- مثال:

plaintext

- Archive data that has not been accessed in the past year.
- Securely delete data that is no longer required for business or compliance purposes.

• سياسات الاحتفاظ (Retention Policies)

- ✓ التعريف: تحدد سياسات الاستبقاء مدة الاحتفاظ بالبيانات ومتى يجب حذفها.
- ✓ الأهمية: ضمان الامتثال للمتطلبات القانونية والتنظيمية، وتقليل تكاليف التخزين، وتقليل المخاطر.

• عوامل في الاعتبار:

- المتطلبات القانونية والتنظيمية (مثل اللائحة العامة لحماية البيانات (GDPR) وقانون HIPAA).
- احتياجات العمل وأنماط استخدام البيانات.
- حساسية البيانات وتصنيفها.

أمثلة:

- يجب الاحتفاظ بالسجلات المالية لمدة سبع سنوات.
- يجب الاحتفاظ بسجلات الموظف طوال مدة التوظيف بالإضافة إلى خمس سنوات.
- يجب الاحتفاظ ببيانات العميل طوال مدة العلاقة مع العميل بالإضافة إلى عامين.

تنفيذ سياسات الاحتفاظ

- تطوير جداول الاستبقاء (Develop Retention Schedules): تحديد فترات الاحتفاظ لأنواع مختلفة من البيانات.
- أتمتة أرشفة البيانات وحذفها (Automate Data Archiving and Deletion): استخدام أدوات الأتمتة لفرض سياسات الاحتفاظ.

- مراجعة السياسات وتحديثها بانتظام (Regularly Review and Update Policies): تأكد من أن السياسات تظل متوافقة مع المتطلبات القانونية واحتياجات العمل.
 - توثيق السياسات وإبلاغها (Document and Communicate Policies): تأكد من أن جميع أصحاب المصلحة على دراية بسياسات الاستبقاء ويلتزمون بها.
- مثال: تنفيذ الاحتفاظ الآلي بالبيانات في SQL Server.

```
sql
CREATE EVENT NOTIFICATION DataRetentionPolicy
ON SERVER
FOR DROP_TABLE
TO SERVICE 'DataRetentionService', 'current database';

-- Example script to delete old records
DELETE FROM Orders WHERE OrderDate < DATEADD(YEAR, -7, GETDATE());
```

اهداف التمرين العملي: يهدف التمرين الى :

1. فهم تصميم قاعدة البيانات العلائقية
2. التعرف على الكيانات والعلاقات: تحديد الكيانات (مثل الكتب، العملاء، الاستعارات) والعلاقات بينها (مثل علاقة "استعار" بين العميل والكتاب).
3. تطبيق التطبيع (Normalization): تصميم الجداول بطريقة تقلل التكرار وتحسن كفاءة قاعدة البيانات.
4. استخدام المفاتيح (Keys): تحديد المفاتيح الأساسية (Primary Keys) والمفاتيح الخارجية (Foreign Keys) لإنشاء العلاقات بين الجداول.
5. إنشاء الجداول وإدراج البيانات.
6. إنشاء الجداول: كتابة أوامر CREATE TABLE لإنشاء الجداول مع تحديد الأعمدة وأنواع البيانات.
7. إدراج البيانات: استخدام أوامر INSERT INTO لإضافة بيانات أولية إلى الجداول.
8. إنشاء استعلامات البيانات.
9. تحديث وحذف البيانات.
10. إدارة العلاقات بين الجداول.
11. تحسين مهارات SQL

التسهيلات التدريبية اللازمة لتنفيذ التمرين العملي :

المواد الأولية	العدد والادوات	الاجهزة والآلات
• قرطاسية	<ul style="list-style-type: none"> • أدوات ادارة المشروع • أدوات التحرير والتوثيق مثل Microsoft Office • منصات التواصل الاجتماعي • أدوات لتحليل أنماط التعاون ومقاييس الاتصال وتقدم المشروع مثل : Google Analytics Microsoft Power point 	<p>أجهزة الحاسوب، الشاشات، انترنت</p> <p>عالي السرعة، سماعات الرأس، كاميرات الويب، الواح تفاعلية، طابعات، ماسحات ضوئية</p>

الشكل / الصورة	خطوات العمل ومعايير ادائها (الرقمية & الوصفية)
	<p>تحديد المتطلبات:</p> <ul style="list-style-type: none"> • حدد الغرض من قاعدة البيانات. • حدد الكيانات (Entities) التي ستكون موجودة في قاعدة البيانات (مثل العملاء، المنتجات، الطلبات، إلخ). • حدد العلاقات بين هذه الكيانات.
	<p>تصميم الجداول:</p> <ul style="list-style-type: none"> • لكل كيان، قم بإنشاء جدول. • حدد الأعمدة (Attributes) لكل جدول. • حدد المفتاح الأساسي (Primary Key) لكل جدول.
	<p>استعلام البيانات:</p> <p>استخدم عبارة SELECT لاسترجاع البيانات من الجداول.</p>
	<p>تحديث البيانات:</p> <p>استخدم عبارة UPDATE لتعديل البيانات الموجودة.</p>
	<p>حذف البيانات:</p> <p>استخدم عبارة DELETE لحذف البيانات.</p>
	<p>إدارة العلاقات:</p> <p>تأكد من أن العلاقات بين الجداول محفوظة باستخدام المفاتيح الخارجية (Foreign Keys).</p>
	<p>تحسين الأداء:</p> <p>استخدم الفهارس (Indexes) لتحسين أداء الاستعلامات.</p>
	<p>إدارة الأخطاء:</p> <p>استخدم عبارات TRY...CATCH أو BEGIN TRANSACTION لإدارة الأخطاء والحفاظ على سلامة البيانات.</p>
	<p>التوثيق:</p>

	قم بتوثيق تصميم قاعدة البيانات والاستعلامات لسهولة الصيانة المستقبلية
تقييم جودة المنتج: وهي مواصفات جودة ودقة ناتج الأداء المطلوب سواء كان منتج أو خدمة	
	1. وضح الأهداف والتخطيط
	2. تفاعل المشاركين والمشاركة الفعالة
	3. جودة المحتوى والأدوات المستخدمة
	4. تأثير التمرين على المشاركين
	5. التقييم والتغذية الراجعة
تقييم الاتجاهات والسلامة المهنية: وهي الاتجاهات المتعلقة بالمحافظة على الأجهزة والأدوات وترتيبها وتنظيفها وتخزينها والسلامة المهنية المتعلقة بالأشخاص والآلات والمعدات والمواد أثناء تأدية التمرين العملي	
	الوعي والالتزام بقواعد السلامة
	السلوكيات والممارسات اليومية
	الثقافة العامة للسلامة المهنية
	تقييم بيئة العمل
	الالتزام باللوائح والقوانين
	التدريب والتطوير