أكاديمية بيان
Bayan Academy

هندسة البيانات

أكـاديـمـيـة بـيان
Bayan Academy


بسم الله الرحمن الرحيم

أكاديمية بيان
Bayan Academy

في عالم تقوده البيانات وتُعاد فيه صياغة مفاهيم المعرفة والإنتاج، تبرز هندسة البيانات كأحد الأعمدة الاستراتيجية لمسارات التحول الرقمي، حيث تُبنى من خلالها البنى التحتية التي تمكّن المؤسسات من استثمار بياناتها بفعالية واتخاذ قرارات دقيقة.

وانطلاقًا من رسالتها في تمكين المجتمع المهني العربي، حرصت أكاديمية بيان على إعداد هذا الكتيب كمرجع موثوق ومبسط يوازن بين التأصيل العلمي والتطبيق العملي، ويسد فجوة المحتوى العربي في مجال هندسة البيانات.

يغطي هذا الإصدار محاور أساسية تشمل دورة حياة البيانات، تصميم بنية البيانات، تقنيات التخزين، وأدوار مهندس البيانات، مع التركيز على تقديم المعرفة بلغة واضحة تراعي واقع الممارسة في بيئات العمل.

ويأتي هذا الكتيب ضمن جهود الأكاديمية لدعم منظومة البيانات الوطنية وتأهيل كفاءات قادرة على إدارة البيانات وفق أفضل الممارسات العالمية، بما يرفع كفاءة الأداء المؤسسي ويعزز المحتوى العربي المتخصص.

نأمل أن يُسهم هذا الإصدار في إثراء المعرفة الرقمية، ويكون لبنة جديدة في بناء مجتمع بيانات وطني يتميز بالكفاءة والاحتراف، وقادر على مواكبة تحديات المستقبل.

أكاديمية بيان

مع انطلاقة التحول الرقمي، لم تعد البيانات مجرد أرقام أو جداول، بل غدت أحد أهم الأصول الاستراتيجية التي تُبنى عليها القرارات وتُعزز بها الكفاءة التشغيلية. غير أن هذه القيمة لا تتحقق إلا حين تُجمع البيانات وتُنظم وتُعالج بجودة تضمن قابليتها للتحليل والاستخدام الفعّال، وهنا يبرز دور هندسة البيانات كالبنية التحتية الأساسية لتمكين التحليلات المتقدمة والنماذج الذكية.

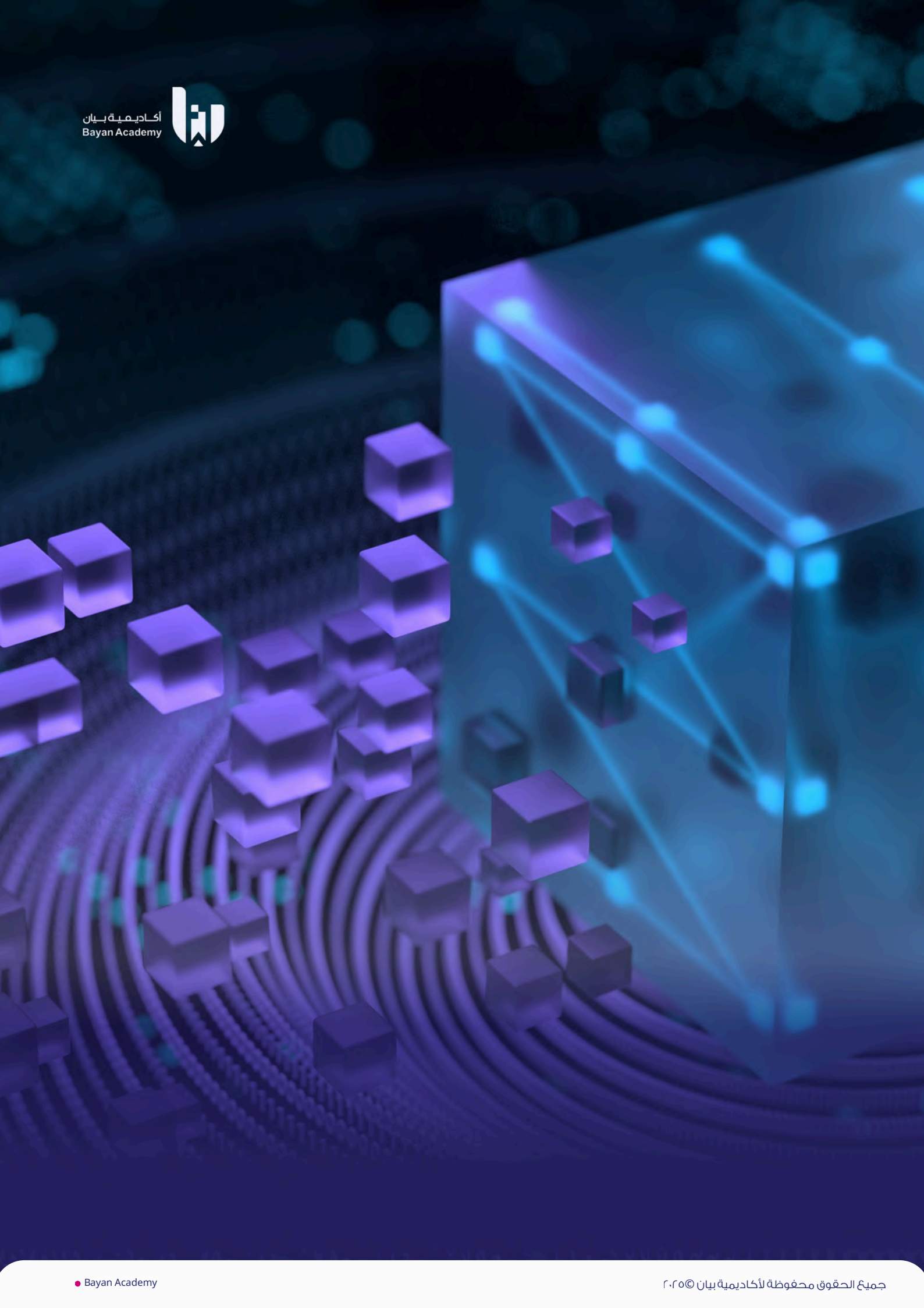
وانطلاقًا من خبرة عملية وتجارب ميدانية متراكمة، اجتهدتُ في إعداد هذا الكتيب ليكون خلاصة معرفية وتطبيقية أضعها بين يدي القارئ. حاولت فيه أن أقدّم دورة حياة البيانات بشكل متسلسل وواضح، بدءًا من جمعها وتنظيفها، مرورًا بآليات تنظيمها ومعالجتها، وصولاً إلى تخزينها وتوفيرها للفرق التحليلية. كما ركزت على الأدوات والمهارات الجوهرية مثل SQL، أنظمة إدارة قواعد البيانات، وتقنيات ETL، لتكون دليلاً عمليًا مبسطًا يسهل الرجوع إليه.

سيجد القارئ بإذن الله في هذه الصفحات ما يعزز فهمه لهندسة البيانات ويمكّنه من تطبيقها بكفاءة في بيئات العمل المختلفة. وأتطلع أن يشعر بعد الانتهاء من قراءة هذا الكتيب أنه قد وجد ما يبحث عنه فعلاً، وأنه أصبح أقرب إلى إتقان مجال يزداد تأثيره وأهميته يومًا بعد يوم.

م. نوف أحمد الجارالله

مستشار هندسة وتحليل البيانات

المحتويات



أكاديمية بيان
Bayan Academy

01
حياة البيانات

(Data lifecycle)

الفصل الأول

حياة البيانات

تُعد هندسة البيانات عنصرًا أساسيًا في بنية الأنظمة الحديثة المعتمدة على البيانات، سنستعرض في هذا الفصل تعريف هندسة البيانات، كما سيتم التطرق إلى المفاهيم الأساسية لهندسة البيانات، مكوناتها الرئيسة، والجوانب التقنية المرتبطة بها، بالإضافة إلى استعراض دور مهندس البيانات، أبرز مهاراته، ومسؤولياته.

1.1 ما هي هندسة البيانات؟

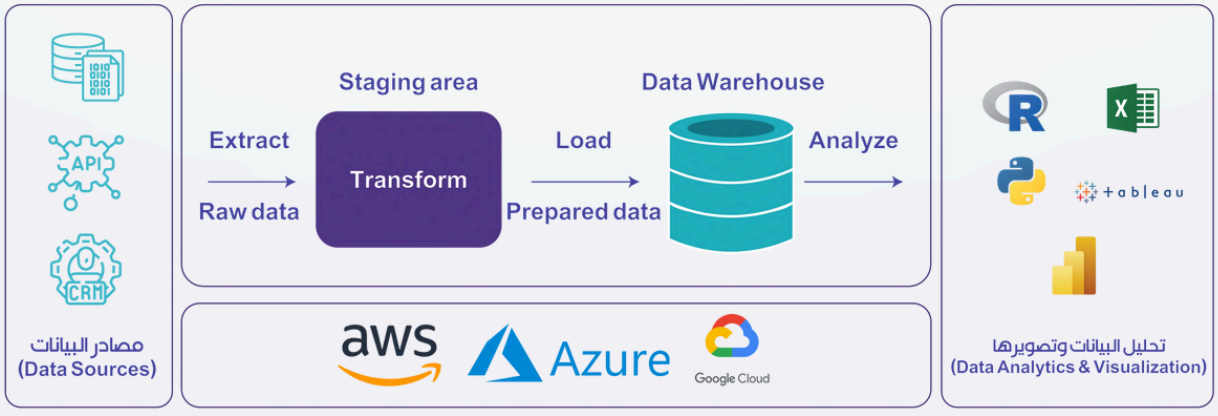
هندسة البيانات هي المجال الذي يركز على تصميم وبناء وصيانة الأنظمة التي تجمع وتخزن وتعالج كميات هائلة من البيانات. الهدف من هندسة البيانات هو ضمان أن تكون البيانات متاحة وموثوقة ومحسّنة للتحليل، مما يمكّن علماء البيانات والمحللين من استخراج رؤى ذات مغزى تدعم اتخاذ القرارات.

1.2 الجوانب الأساسية لهندسة البيانات

1.2.1 تطوير خطوط معالجة سير عمل البيانات(Data Pipeline)

خطوط المعالجة هي مسارات عمل أساسية تقوم باستخراج البيانات من مصادر متعددة، وتحويلها إلى صيغة قابلة للاستخدام، ثم تحميلها في أنظمة التخزين مثل مستودع البيانات(Data Warehouse) أو بحيرة البيانات (Data Lake). يُعرف هذا الإجراء ب(ETL: Extract, Transform, Load) كما هو موضح بالشكل(1).
غالباً ما تكون خطوط المعالجة مؤتمتة لتعمل حسب جداول زمنية أو استجابة لمؤثرات، مما يضمن حداثة البيانات وتوافرها.

الشكل (1) : خطوط معالجة سير عمل البيانات (Data Pipeline)



1.2.2حلول تخزين البيانات

يقوم مهندسو البيانات بتحديد الحلول الأكثر ملاءمة للتخزين بناءً على نوع وحجم البيانات واحتياجات العمل.
يعملون مع خيارات تخزين متعددة تشمل:

قواعد البيانات العلائقية (Relational Database) مثل
(MySQL و PostgreSQL)

بحيرات البيانات (Data Lake) لتخزين البيانات الخام مثل
(Amazon S3 و Google Cloud Storage)

مستودعات البيانات (Data Warehouse) مثل
(Snowflake و Amazon Redshift)

يهدف المهندسون إلى تحقيق التوازن بين تكلفة التخزين وسرعة الوصول للبيانات.

1.2.3 تحويل وتنظيف البيانات:

نادرًا ما تكون البيانات الخام جاهزة للتحليل، لذلك يقوم مهندسو البيانات بتنظيف البيانات والتحقق من جودتها وترتيبها للتأكد من أنها مناسبة للاستخدام.
يتضمن ذلك معالجة القيم المفقودة وتوحيد التنسيقات وإزالة البيانات المكررة.

تكامل البيانات

يشمل تكامل البيانات جمع البيانات من مصادر متعددة لإنشاء مجموعة بيانات موحدة. قد يتضمن هذا الربط بين واجهات برمجة التطبيقات (API)، قواعد البيانات، مصادر البث الحي، وغيرها. يستخدم مهندسو البيانات أدوات مثل(Apache Kafka و Talend) لإدارة تكامل البيانات سواء في الوقت الفعلي أو على دفعات.

تنظيم سير العمل للبيانات

تتطلب عمليات البيانات المعقدة تنسيقاً لإدارة التبعيات وضمان تنفيذها بسلاسة. تُستخدم أدوات مثل Airflow (Apache و Prefect) لأتمتة وإدارة سير العمل.

أمن البيانات وإدارتها

يعمل مهندسو البيانات على حماية البيانات الحساسة من خلال تطبيق التشفير، وإعداد قيود الوصول، وضمان الامتثال للمتطلبات التنظيمية.
تشمل إدارة البيانات وضع معايير جودة البيانات، وتحديد مالكي البيانات، وضمان إمكانية تتبع البيانات لأغراض التدقيق.

البيانات الضخمة والحوسبة الموزعة

غالباً ما تتطلب هندسة البيانات التعامل مع البيانات الضخمة، حيث تكون الطرق التقليدية للتخزين والمعالجة غير كافية. يستخدم المهندسون أطر الحوسبة الموزعة مثل(Apache Spark و Hadoop) لمعالجة مجموعات البيانات الكبيرة بشكل متوازٍ، مما يضمن السرعة والكفاءة.

الحوسبة السحابية والبنية التحتية السحابية

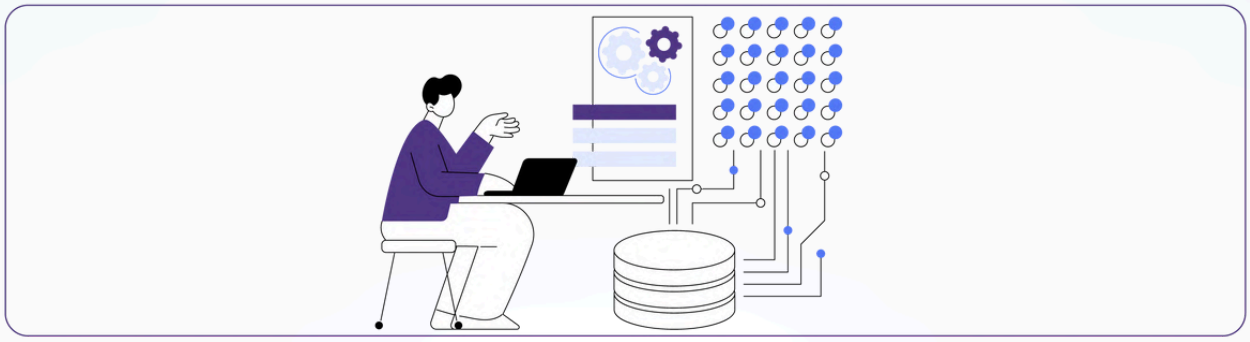
مع انتشار المنصات السحابية مثل(AWS و Google Cloud و Azure) يقوم مهندسو البيانات بتصميم البنية التحتية السحابية التي تشمل توفير الموارد وإعداد تخزين البيانات القابل للتوسع.

1.3 من هو مهندس البيانات؟

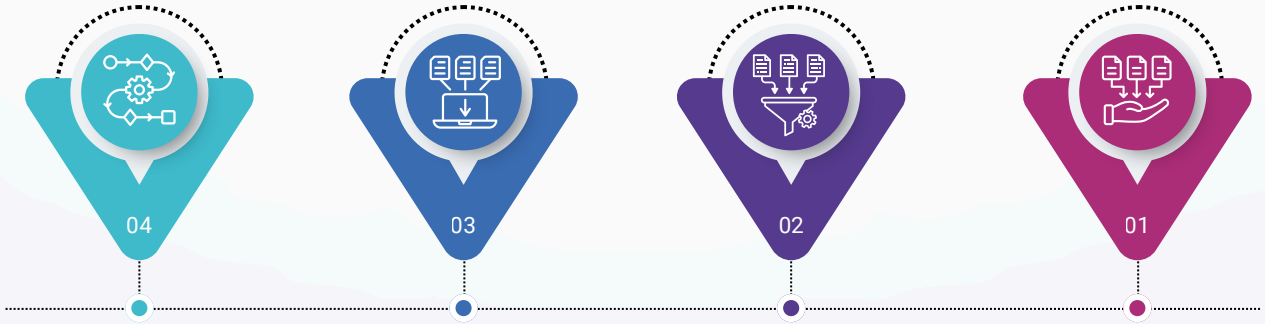
يُعد مهندس البيانات (Data Engineer) المسؤول عن بناء الأنظمة التي تجمع البيانات من مصادرها المختلفة، ومعالجتها وتنظيمها وتخزينها بطريقة تضمن جاهزيتها للاستخدام في التحليل واتخاذ القرارات.

يعمل مهندس البيانات على تطوير بنية تحتية متكاملة لبيانات المؤسسة تشمل:

مهندس البيانات (Data Engineer)



1.4 أبرز أدوار مهندس البيانات



جمع البيانات:

يأخذ البيانات من مصادر متنوعة (مثل قواعد البيانات، أو الإنترنت، أو أجهزة الاستشعار) وينقلها إلى النظام الرئيسي (Main system) للشركة.

تنظيم البيانات:

يقوم بترتيب البيانات وتنظيفها، بحيث تكون واضحة وخالية من الأخطاء. هذا يتضمن إزالة البيانات المكررة أو إصلاح البيانات غير الكاملة.

تخزين البيانات:

يضع البيانات في أماكن مخصصة مثل قواعد البيانات أو بحيرات البيانات، ويحدد أفضل طريقة لتخزينها بحيث تكون آمنة وسهلة الوصول.

تسهيل الوصول للبيانات:

ينشئ مسارات أو خطوط لنقل البيانات بشكل آلي، حتى يتم تحديثها وتكون متاحة في أي وقت يحتاجه الفريق.

لماذا هذا الدور مهم؟ بدون مهندس البيانات، ستكون البيانات غير منظمة أو غير دقيقة، مما يجعل من الصعب على الشركات استخدامها لاتخاذ قرارات صحيحة.

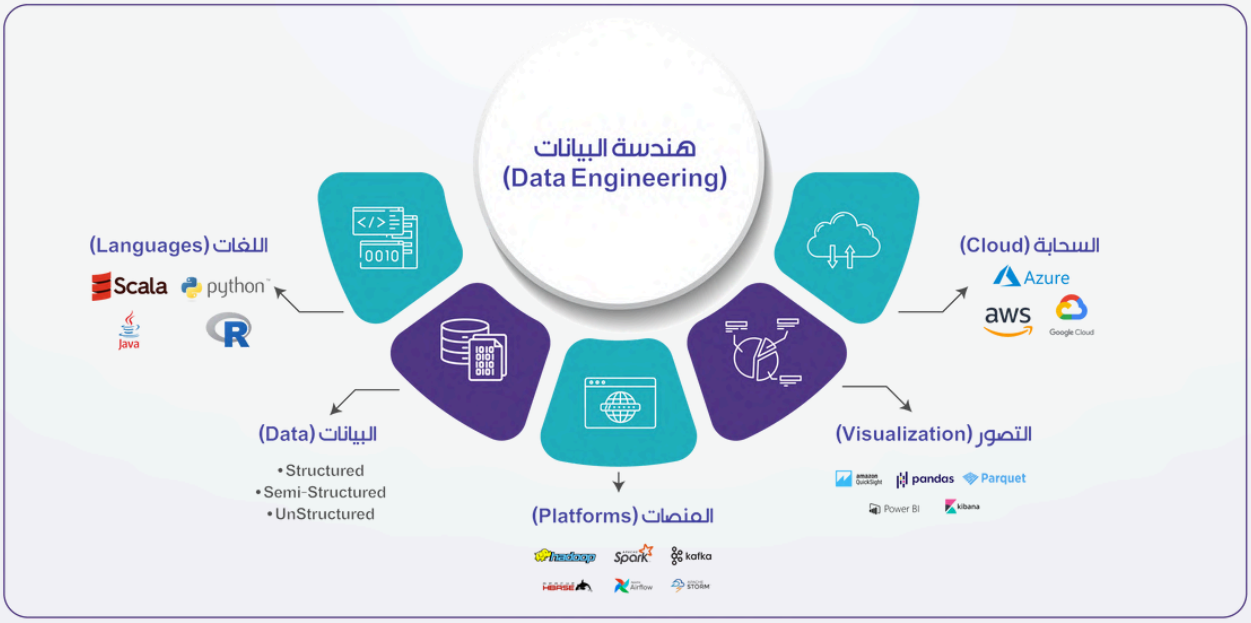
1.5 المهارات الأساسية والانشطة لمهندسي البيانات

تعد هندسة البيانات مجالاً متنوعًا يتطلب مزيجًا من المهارات الفنية والمهارات الشخصية، بالإضافة إلى مجموعة من الأنشطة التي تضمن جمع البيانات ومعالجتها وتوفيرها بكفاءة للتحليل. وفيما يلي المهارات الأساسية والأنشطة المرتبطة بهندسة البيانات.

1.6 المهارات الاساسية لمهندسي البيانات

يوضّح الشكل(2) المجالات الرئيسة التي تشكّل عمل مهندس البيانات:

الشكل (2): المجـالات الرئـيسة لعمـل مهندس البيانات



مهارات البرمجة

إتقان لغات البرمجة مثل(Python Javag و Scala) لكتابة سكربتات معالجة البيانات وبناء سير عمل البيانات.
المعرفة بلغة (SQL) لاستعلام وإدارة قواعد البيانات العلائقية (Relational Data).

إدارة قواعد البيانات ومخازن البيانات

معرفة بمفاهيم وأدوات مخازن البيانات مثل (Googleg Amazon Redshift BigQuery).
الخبرة في أنظمة إدارة قواعد البيانات مثل(MySQL و PostgreSQL و MongoDB).
عمليات (ETL):الاستخراج، التحويل، التحميل.
فهم لمفاهيم (ETL) والخبرة باستخدام أدوات (ETL) مثل (Talend و Apache NiFi أو Informatica) لتكامل البيانات.القدرة على تصميم وتنفيذ سير عمل البيانات التي تعمل تلقائيًا.

تقنيات البيانات الضخمة

معرفة بأطر البيانات الضخمة مثل(Apache Hadoop و Apache Spark Apache Kafka) لمعالجة كميات كبيرة من البيانات. فهم الحوسبة الموزعة وأنظمة التخزين.

تصميم البيانات

المهارات في تصميم نماذج البيانات التي تمثل البيانات بشكل فعال وتدعم احتياجات الأعمال.
معرفة بالتطبيع وعدم التطبيع (Normalization - Denormalization)وكذلك(Star Schema - Snowflake Schema).

المنصات السحابية

الخبرة في خدمات السحابة مثل (AWS Google Cloudg و Azure Platform) لتخزين البيانات ومعالجتها.
فهم الهياكل المعمارية للبيانات (Data architectures) المستندة إلى السحابة والحوسبة بدون خادم.

جودة البيانات وإدارتها

القدرة على تنفيذ تدابير جودة البيانات، والمراقبة، وعمليات التحقق.
المعرفة بأطر حوكمة البيانات والامتثال للمعايير.

أمان البيانات

معرفة بممارسات أمان البيانات، بما في ذلك التشفير، وضوابط الوصول، والامتثال للوائح الخصوصية.

التعاون والتواصل

مهارات تواصل قوية للتعاون مع علماء البيانات والمحللين وغيرهم من أصحاب المصلحة لفهم متطلبات البيانات.
القدرة على توثيق تدفقات العمل والعمليات وسلسلة البيانات.

مهارات حل المشكلات والتحليل

مهارات تحليلية قوية لاستكشاف مشكلات البيانات وتحسين تدفقات البيانات.
القدرة على التفكير النقدي والإبداعي لحل التحديات المعقدة المتعلقة بالبيانات.

1.7 الأنشطة الرئيسية في هندسة البيانات

01

جمع البيانات:

جمع البيانات من مصادر متعددة، بما في ذلك واجهات برمجة التطبيقات، وقواعد البيانات، وتغذيات البيانات الخارجية.
إعداد عمليات تلقائية لجمع البيانات بشكل منتظم.

02

تحويل البيانات:

تنظيف وتحويل وتنسيق البيانات الخام لضمان جودتها، بما في ذلك تطبيع البيانات وتوحيدها.
تنفيذ عمليات تعزيز البيانات لتحسين جودة البيانات.

03

تخزين وإدارة البيانات:

تصميم وإدارة حلول تخزين البيانات (بحيرات البيانات، مخازن البيانات) لتلبية احتياجات المؤسسة.
ضمان تخزين البيانات بشكل آمن وفعال.

04

بناء سير عمل البيانات:

تطوير وصيانة سير عمل البيانات التي تسهل حركة البيانات من أنظمة المصادر إلى أدوات التخزين والتحليل كما هو موضح بالشكل(3).
أتمتة تنفيذ تدفقات بيانات باستخدام أدوات التنسيق مثل (Apache Airflow).

05

المراقبة والصيانة:

مراقبة خطوط أنابيب البيانات وتدفقات العمل لأدائها وموثوقيتها.
إجراء الصيانة الدورية لضمان جودة البيانات وأداء النظام.

06

التعاون:

العمل بشكل وثيق مع علماء البيانات والمحللين لفهم احتياجاتهم من البيانات وتوفير المجموعات المطلوبة.
المشاركة في فرق متعددة التخصصات لمواءمة استراتيجيات البيانات مع الأهداف التجارية.

08

التوثيق:

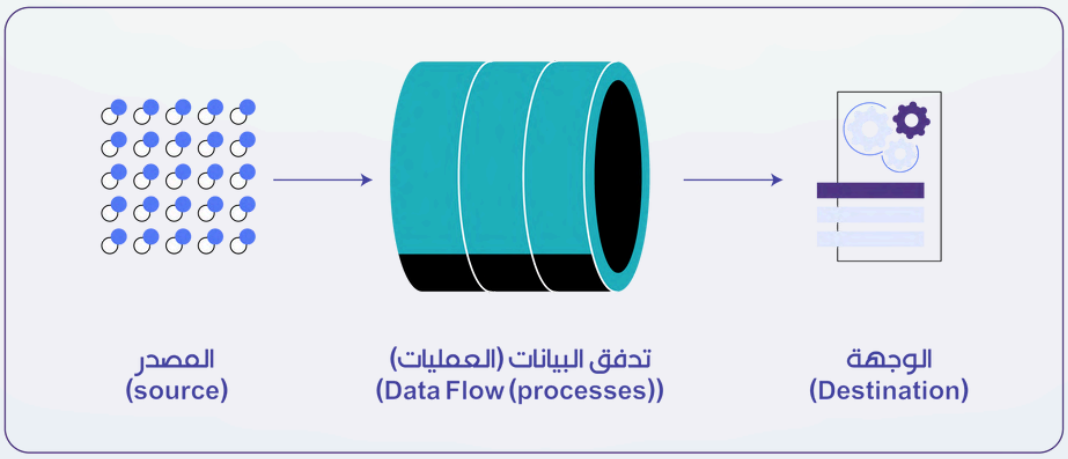
إنشاء وثائق شاملة لهياكل البيانات، وتدفقات العمل، ونماذج البيانات لضمان الشفافية ومشاركة المعرفة.
الحفاظ على سلسلة بيانات واضحة لتتبع تدفق البيانات عبر الأنظمة.

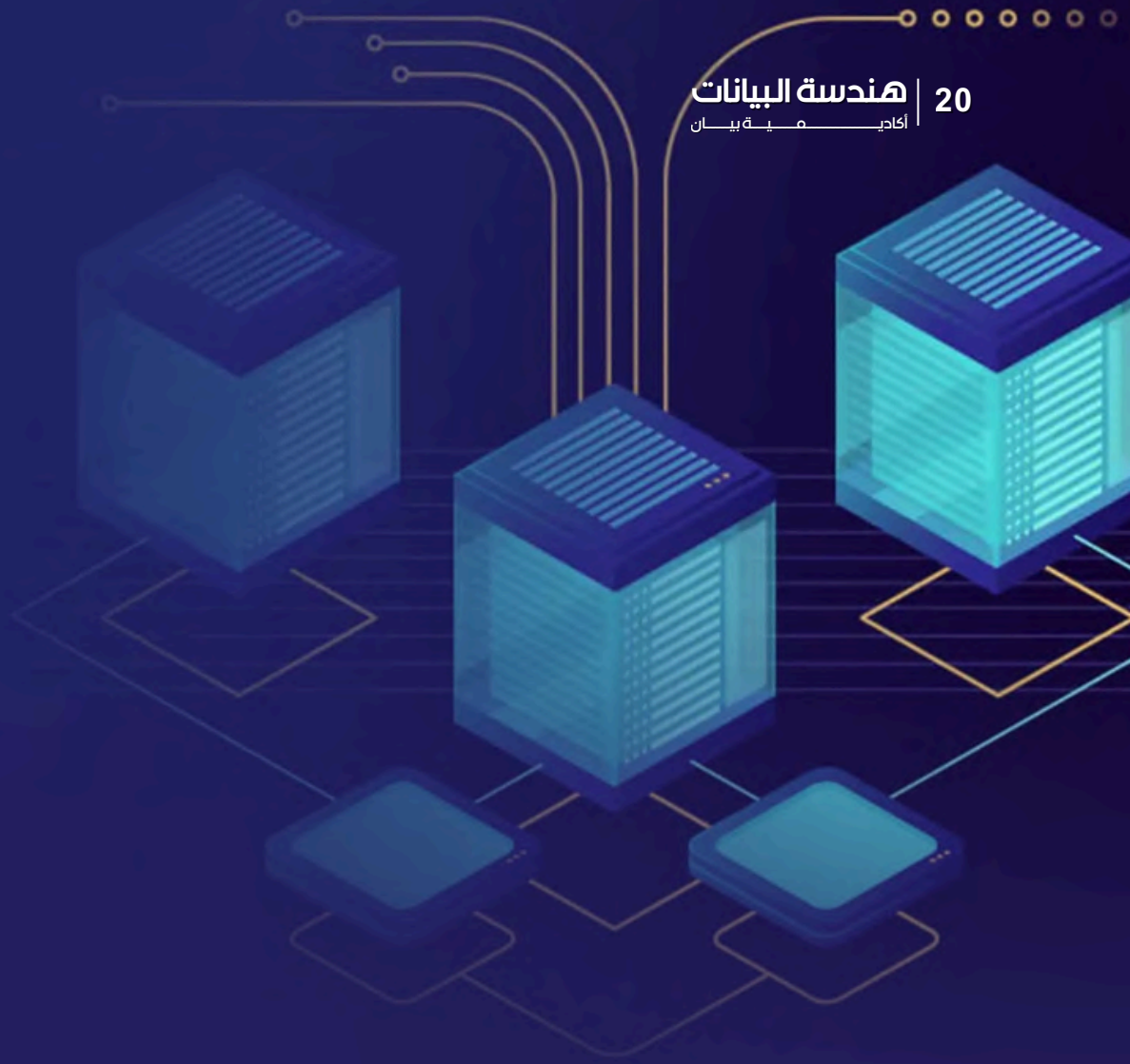
09

تحسين الأداء:

تحليل وتحسين تدفقات معالجة البيانات من أجل الكفاءة والسرعة.
تنفيذ أفضل الممارسات لتخزين البيانات والوصول إليها لتعزيز الأداء.

الشكل (3) : سير عمل البيانات (Data Pipeline)

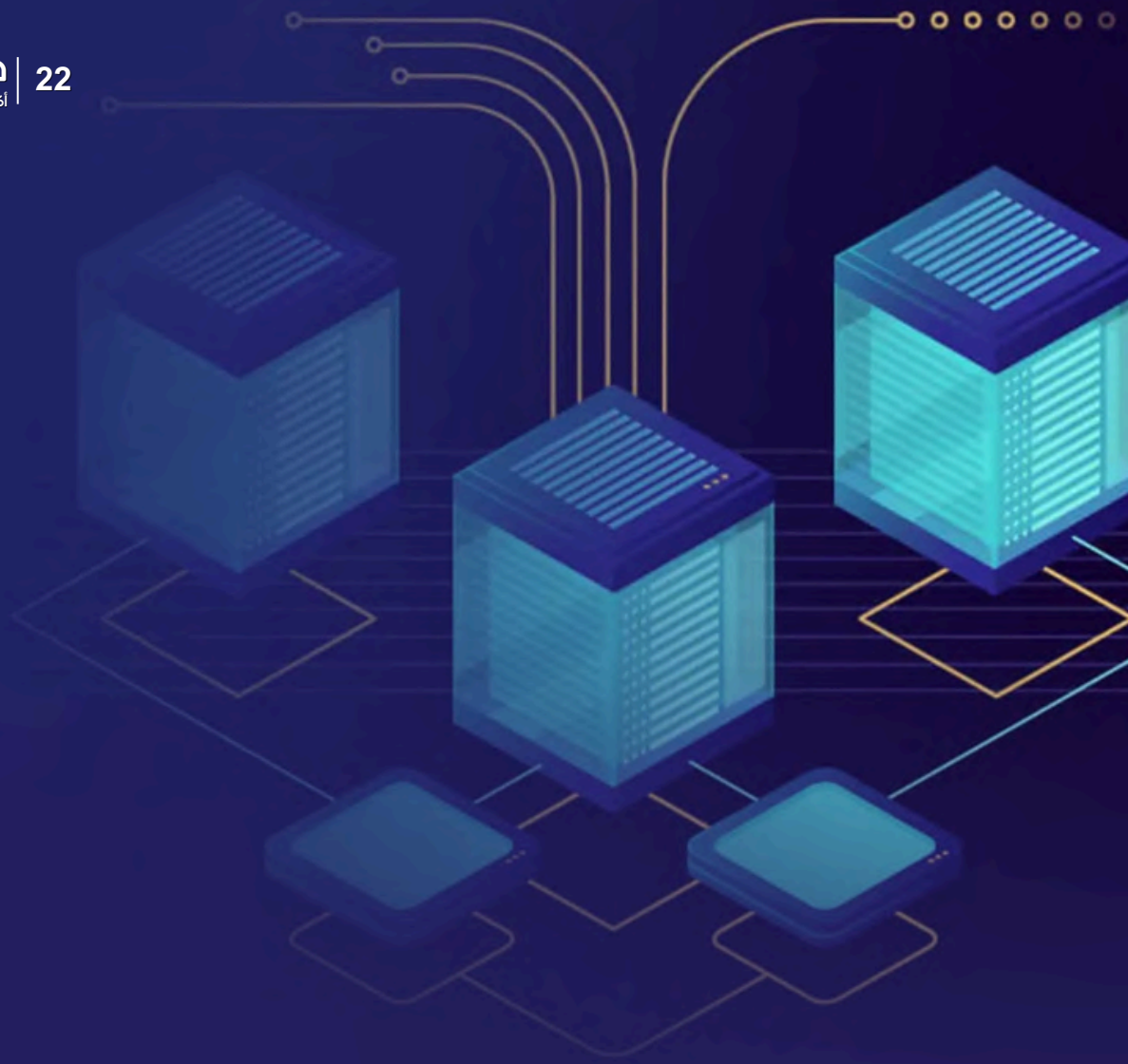




في ختام الفصل الأول

حياة البيانات

تسهم هندسة البيانات في تمكين المؤسسات من استثمار بياناتها بفعالية، من خلال بناء أنظمة موثوقة لجمع البيانات وتحويلها وتحسين جودتها. وبالاستفادة من المهارات التقنية والأدوات الحديثة، كما يضمن مهندسو البيانات توفير بيانات دقيقة تدعم التحليل واتخاذ القرار في بيئات عمل ديناميكية.



02
دورة حياة هندسة البيانات

Data Engineering Lifecycle

الفصل الثاني

دورة حياة هندسة البيانات

تُمثل دورة حياة هندسة البيانات عنصر أساسي في بناء منظومات بيانات حديثة، حيث تنظّم العمليات التي يتم من خلالها تحويل البيانات الخام إلى مخرجات قابلة للاستخدام والتحليل. كما تركّز على تطوير البنية التحتية والأنظمة التي تُمكّن من إدارة البيانات ومعالجتها بكفاءة واستدامة. سنستعرض في هذا الفصل مجموعة من المحاور الأساسية. الفرق بين دورة حياة البيانات ودورة حياة هندسة البيانات، والمراحل الأساسية لدورة حياة هندسة البيانات، كما سنتعرف على العوامل الممتدة عبر جميع المراحل، مثل الأمان، جودة البيانات، الأتمتة، وهندسة البرمجيات، باعتبارها ركائز داعمة لضمان كفاءة واستمرارية النظام.

2.1 ما هي دورة حياة هندسة البيانات؟

دورة حياة هندسة البيانات هي مجموعة من المراحل التي تهدف إلى تحويل البيانات الخام إلى منتج نهائي مفيد يمكن للمحللين وعلماء البيانات ومهندسي تعلم الآلة الاستفادة منها. تركز دورة حياة هندسة البيانات على بناء الأنظمة والبنية التحتية التي تدعم إدارة البيانات وتحليلها بشكل فعال. في هذه الدورة، يتم التركيز على المفاهيم الأساسية لكل مرحلة مع تخصيص التفاصيل التقنية للمراحل التالية.

2.2 ما الفرق بين دورة حياة البيانات و دورة حياة هندسة البيانات؟

دورة حياة هندسة البيانات:
ركز على المراحل التقنية التي يتحكم بها مهندسو البيانات لبناء أنظمة إدارة البيانات.

دورة حياة البيانات:
تشمل جميع مراحل التعامل مع البيانات، من إنشائها إلى الأرشفة أو الحذف، وتتضمن الأدوار المختلفة مثل الحوكمة والتحليل وإدارة البيانات.

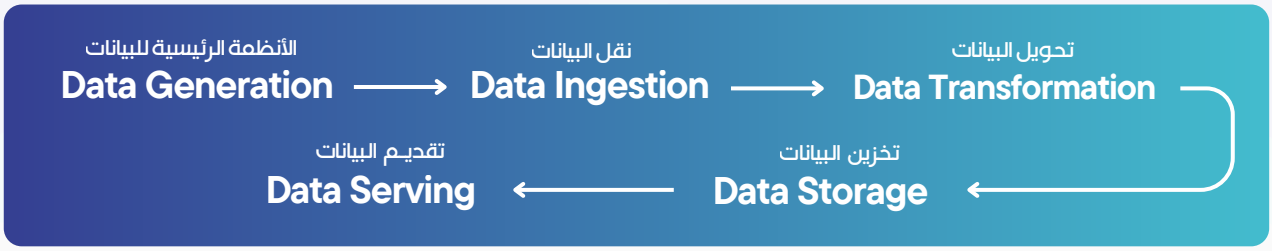
حياة البيانات

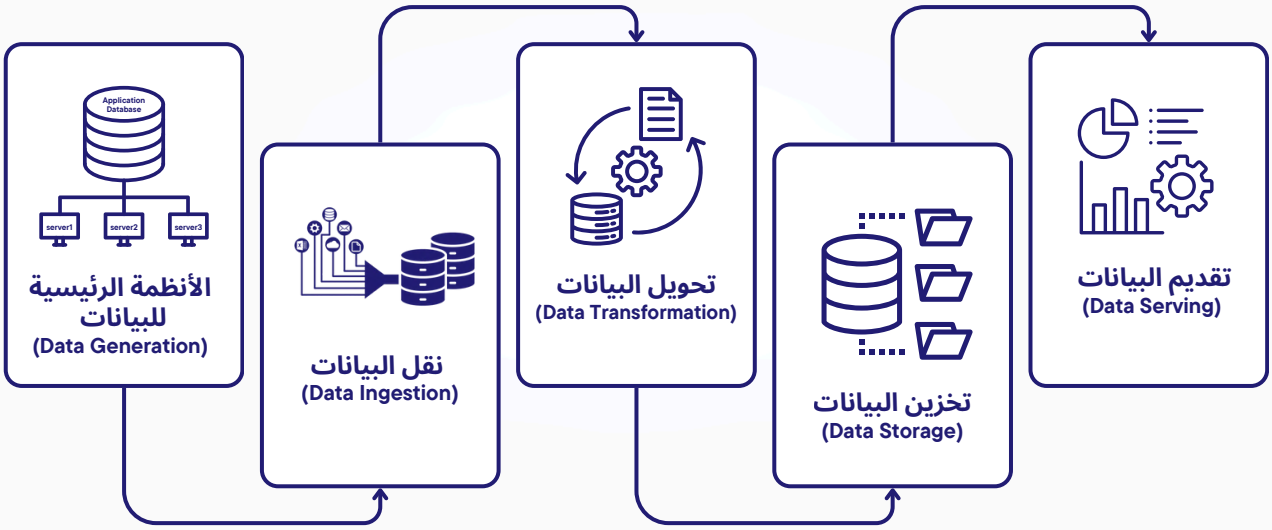
دورة حيـــــاة هندسة البيانات

Data Lifecycle

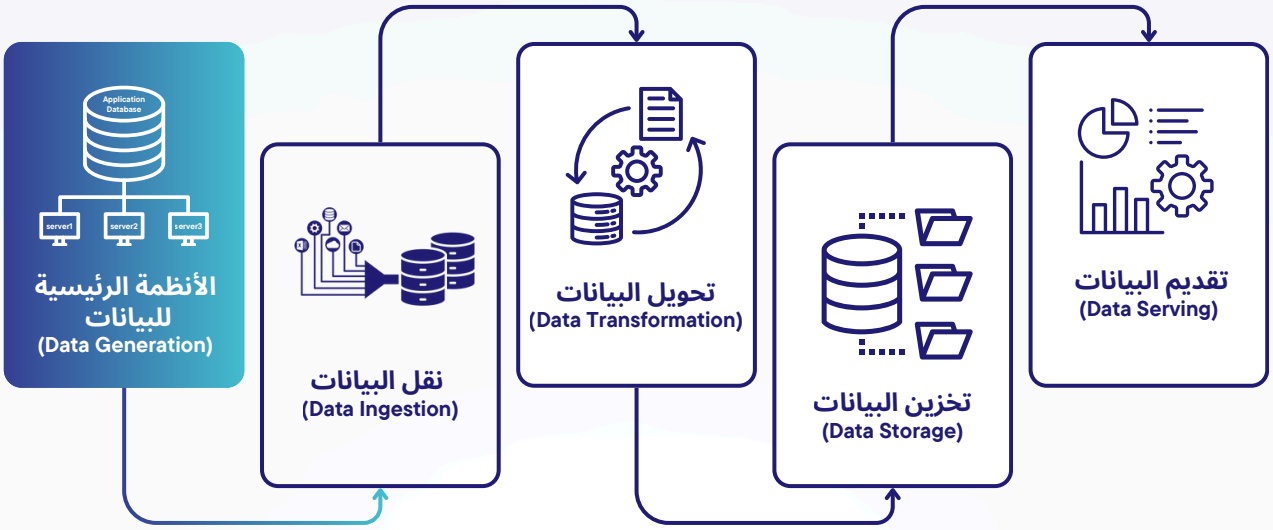
Data Engineering Lifecycle

2.3 ما هي مراحل دورة حياة هندسة البيانات؟





2.3.1 المرحلة الأولى : الأنظمة الرئيسية للبيانات (Data Generation)



التعريف:

الأنظمة الرئيسية للبيانات هي المرحلة التي تبدأ فيها البيانات، حيث يتم إنتاجها بواسطة الأنظمة المصدر التي تعتبر الأساس لأي نظام هندسة بيانات.

الوصف:

الأنظمة المصدر تُنتج البيانات بتنوع واسع من الأنواع والأحجام والسرعات. تشمل أمثلة الأنظمة المصدر قواعد البيانات التشغيلية، أجهزة إنترنت الأشياء (IoT)، وأنظمة الرسائل. قد تكون البيانات الناتجة إما بيانات منظمة (structured data) أو بيانات غير منظمة(unstructured data)، وأي تغييرات في الأنظمة المصدر قد تؤثر على بقية دورة حياة البيانات.

الأهداف:

1. جمع البيانات من المصادر بشكل متسق ودقيق.

2. فهم آلية إنتاج البيانات لضمان تكاملها وجودتها.

3. تحديد العوامل التي تؤثر على تكرار الإنتاج وسرعته.

أمثلة:

01 **إنترنت الأشياء (IoT):**

أجهزة استشعار وأجهزة ذكية ترسل بياناتها إلى أنظمة مركزية، مثل تلك المستخدمة في الزراعة، النقل، والرعاية الصحية.

02 **أنظمة بيع المنتجات:**

مصدر بيانات كبير يشمل المنتجات، المبيعات، والمدفوعات، حيث يتم إنتاج بيانات متنوعة مرتبطة بالمنتجات مثل الكميات المباعة والأسعار.

الاعتبارات الهندسية:

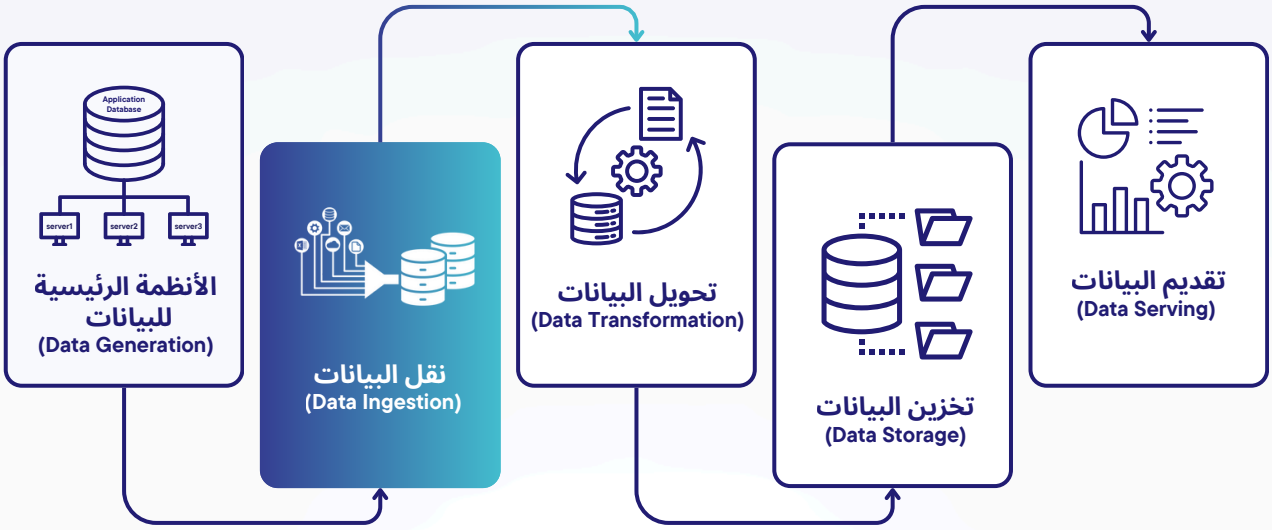
تصنيف البيانات:
هل تأتي من تطبيقات أم من أجهزة (IoT)؟

معدل الإنتاج:
كم حدث يتم إنتاجه في الثانية؟

جودة البيانات:
ما مدى تكرار الأخطاء والتكرارات؟

إدارة التغيير:
كيفية التعامل مع تغييرات الهيكل أو البيانات باستخدام تقنيات مثل (Change Data Capture)

2.3.2 المرحلة الثانية: نقل البيانات (Data Ingestion)



التعريف:

نقل البيانات هو عملية نقل البيانات من الأنظمة المصدر إلى الأنظمة المعالجة أو التخزين.

الوصف:

يتم نقل البيانات إما دفعة واحدة (Batch) أو يتم نقلها في الوقت الحالي (Streaming). تشمل التحديات في هذه المرحلة جودة البيانات، موثوقية الأنظمة المصدر، وحجم البيانات. تتم عملية النقل عندما تصبح البيانات جاهزة للتحويل أو التخزين.

الأهداف:

1.نقل البيانات من المصدر إلى وجهتها بكفاءة ودقة.

2. تقليل زمن التأخير بين الإنتاج والاستهلاك.

3. الحفاظ على جودة البيانات للاستخدام الفوري.

أمثلة:

01

النقل بالدفعات (Batch Ingestion):

معالجة كميات كبيرة من البيانات دفعة واحدة، مثل معالجة بيانات يوم كامل من المعاملات.

02

النقل بالتدفق (Streaming Ingestion):

إرسال البيانات فور إنتاجها أو بالقرب من الوقت الحقيقي، كما في أنظمة تتبع الزوار على الإنترنت، حيث يتم جمع البيانات بشكل لحظي عبر تدفق بيانات مستمر لمعالجتها أو تخزينها في الوقت الفعلي.

الاعتبارات الهندسية:

اختيار التقنية المناسبة:

هل يناسب تدفق البيانات(Batch Ingestion) أو دفعات(Batch Ingestion)؟

الموثوقية:

هل الأنظمة المصدر موثوقة؟

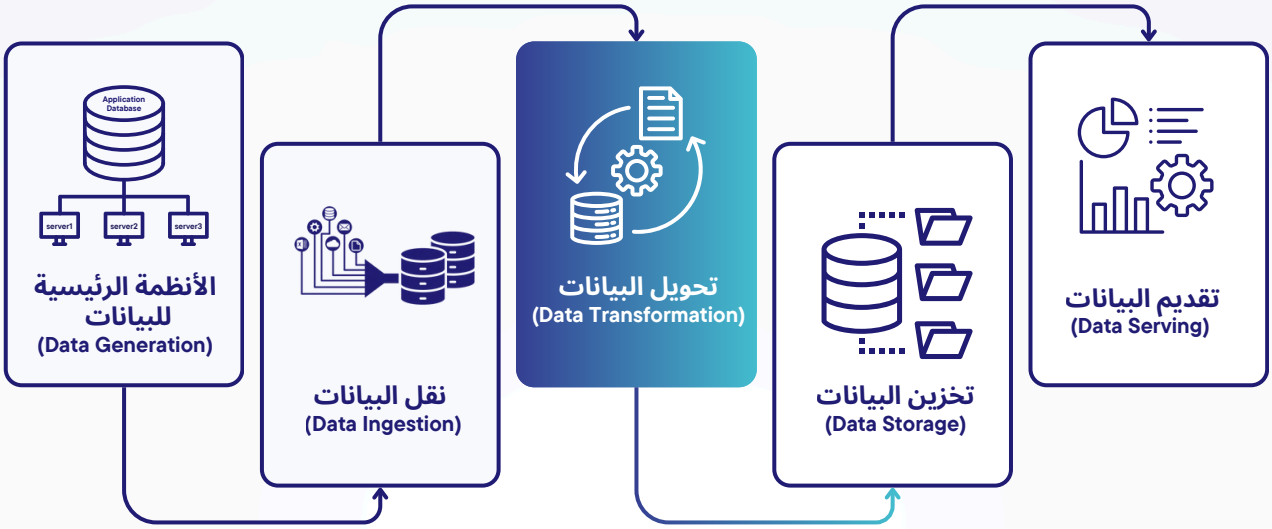
وجهة البيانات:

هل الأنظمة اللاحقة جاهزة لاستقبال البيانات؟

جودة البيانات:

هل البيانات في حالة جيدة للاستخدام الفوري؟

2.3.3 المرحلة الثالثة: التحويل (Data Transformation)



التعريف:

التحويل هو المرحلة التي يتم فيها معالجة البيانات الخام لتكون جاهزة للاستخدام التحليلي أو التعليمي.

الوصف:

تشمل التحويلات تنظيف البيانات، تطبيق قواعد الأعمال (Business Rules)، وإنشاء الميزات لتحسين التحليلات. الهدف هو تحويل البيانات إلى أشكال مهيأة للاستخدام.

الأهداف:

1. تحسين جودة البيانات وجعلها أكثر قيمة للمستخدمين.

2. تجهيز البيانات لدعم النماذج الذكية أو التحليلات.

3. ضمان توافق البيانات مع متطلبات التقارير أو التطبيقات.

أمثلة:

01 **تنظيف البيانات:**
إزالة القيم الناقصة أو المتكررة.

02 **تطبيق قواعد الأعمال (Business Rules):**
مثل تحويل العملات أو حساب الضرائب.

03 **تحسين الميزات:**
استخراج ميزات جديدة لنماذج الذكاء الاصطناعي، مثل استخدام بيانات المشتريات السابقة لتحسين التوصيات للعملاء.

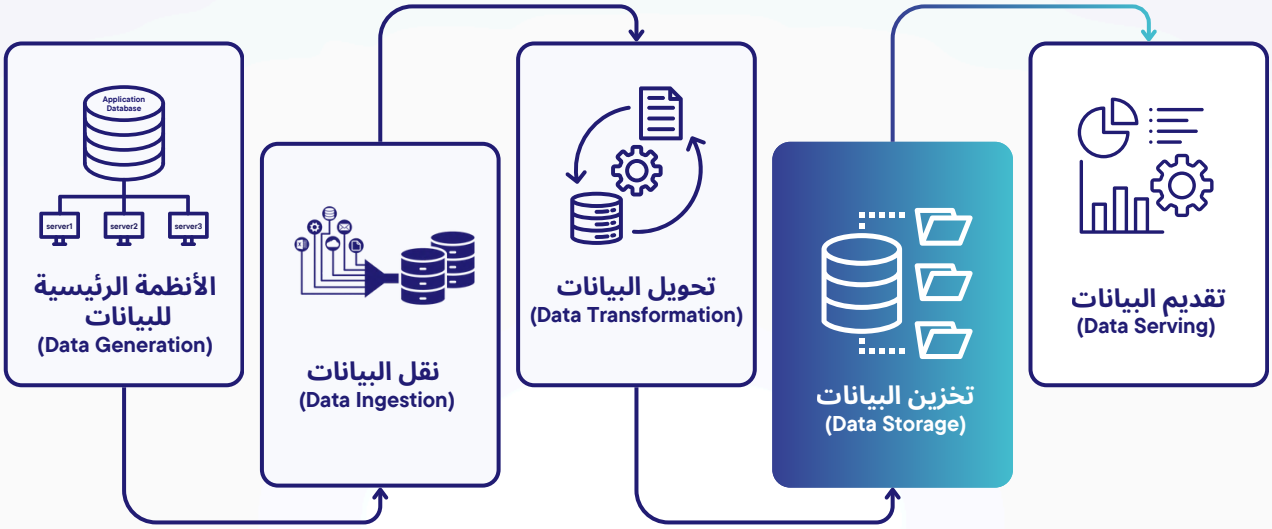
الاعتبارات الهندسية:

اختيار الأدوات المناسبة:
مثل (Spark أو SQL).

إدارة التحويلات:
هل يتم تسجيل جميع التحويلات لتتبع الأخطاء؟

الأداء:
هل التحويلات تقلل من زمن المعالجة دون التأثير على الجودة؟

2.3.4 المرحلة الرابعة: التخزين (Data Storage)



التعريف:

التخزين هو العملية التي يتم فيها حفظ البيانات في أنظمة تخزين مُخصصة لدعم المعالجة المستقبلية.

الوصف:

البيانات المخزنة يمكن أن تكون منظمة (Structured) أو غير منظمة (Unstructured). يتنوع التخزين بين بحيرات البيانات (Data Lakes)، مستودعات البيانات (Data Warehouses)، والتخزين الكائني (Object Storage). يتم التخزين في عدة مراحل لدعم عمليات النقل، التحويل، والتقديم.

الأهداف:

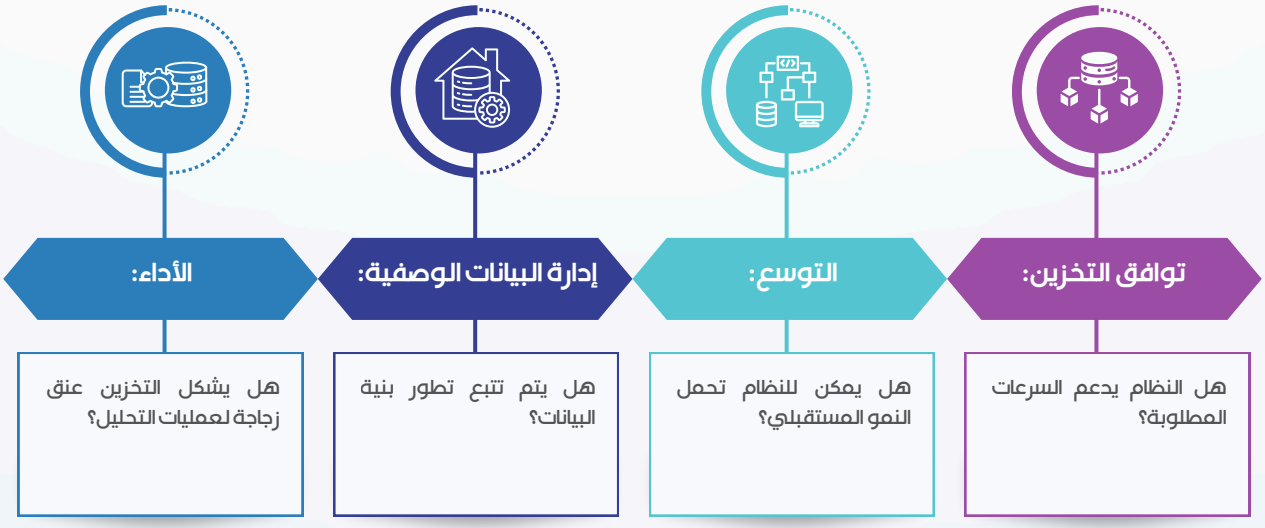
1. توفير نظام تخزين آمن وقابل للتوسع.
2. دعم عمليات القراءة والكتابة بسرعات مناسبة.
3. إدارة البيانات الوصفية لتتبع تطورها وتحليلها.

أمثلة:

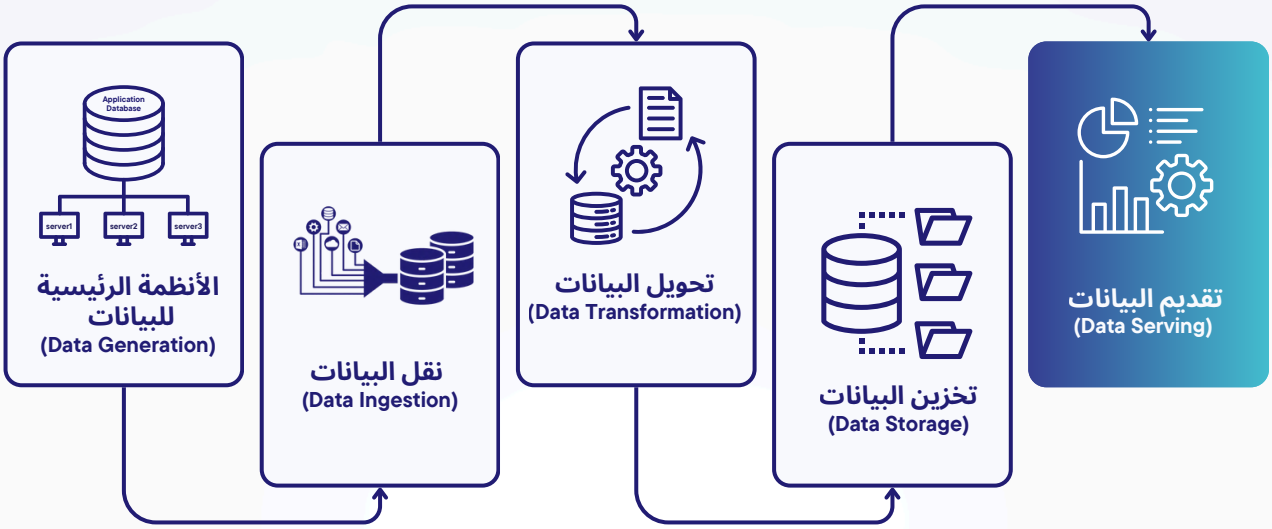
01 **بحيرات البيانات (Data Lakes):**
مناسبة لتخزين البيانات الخام غير المنظمة.

02 **مستودعات البيانات (Data Warehouses):**
تُستخدم لتحليل البيانات المنظمة، مثل تخزين بيانات المستخدمين التي تم تحليلها من تطبيقات التجارة الإلكترونية في مستودع بيانات لتحليل سلوك العملاء.

الاعتبارات الهندسية:



2.3.5 المرحلة الخامسة: تقديم البيانات (Data Serving)



التعريف:

تقديم البيانات هو المرحلة التي يتم فيها عرض البيانات للمستخدمين النهائيين أو الأنظمة الأخرى.

الوصف:

البيانات تُقدم بأشكال متعددة، مثل تقارير الأعمال، واجهات برمجية، أو أنظمة الذكاء الاصطناعي. تشمل هذه المرحلة تقديم البيانات للتحليل، التنبؤ، أو دمجها مع أنظمة تشغيلية.

الأهداف:

1. توفير وصول سريع وسهل للبيانات.

2. تلبية متطلبات المستخدمين النهائيين أو الأنظمة الأخرى.

3. دعم التحليلات الفورية أو المدمجة في الأنظمة التشغيلية.

أمثلة:

01

التحليل التشغيلي:

مثل مراقبة العمليات الجارية مثل المخزون.

02

الذكاء الاصطناعي:

تدريب النماذج أو تقديم التوصيات الفورية للعملاء بناءً على بيانات المعاملات السابقة.

03

التغذية العكسية:

إعادة البيانات المعالجة إلى الأنظمة المصدر، مثل تحديث قاعدة بيانات المخزون بناءً على تحليل الطلب.

الاعتبارات الهندسية:

الأداء:

هل الأنظمة قادرة على التعامل مع كميات كبيرة من الاستفسارات؟

التوسع:

هل يمكن للنظام دعم عدد متزايد من المستخدمين؟

الأمان:

هل يتم حماية البيانات من الوصول غير المصرح به؟

2.4 هل هناك عوامل مشتركة عبر جميع المراحل؟

نعم، هناك عدة عوامل تمتد عبر جميع المراحل لدعم دورة حياة هندسة البيانات بكفاءة:

01 الأمان:
حماية البيانات في جميع المراحل.

02 إدارة البيانات:
التتبع والتحكم بجودة البيانات.

03 الأتمتة:
تسهيل العمليات لتقليل الأخطاء البشرية.

04 هندسة البرمجيات:
تصميم أنظمة قابلة للصيانة والتوسع.

في ختام الفصل الثاني

دورة حياة هندسة البيانات

تعد دورة حياة هندسة البيانات إطارًا أساسيًا لبناء أنظمة فعالة لإدارة البيانات وتحويلها إلى أصول قابلة للاستخدام في التحليل واتخاذ القرار. ومن خلال التركيز على المراحل التقنية المحورية، إلى جانب العوامل الداعمة مثل الأمان، والأتمتة، وإدارة البيانات، تضمن هذه الدورة توفير بنية تحتية متينة ومرنة تلبي احتياجات المؤسسات الحالية وتواكب متطلبات المستقبل في عالم البيانات.

03
تصميم بنية البيانات

Data Architecture design

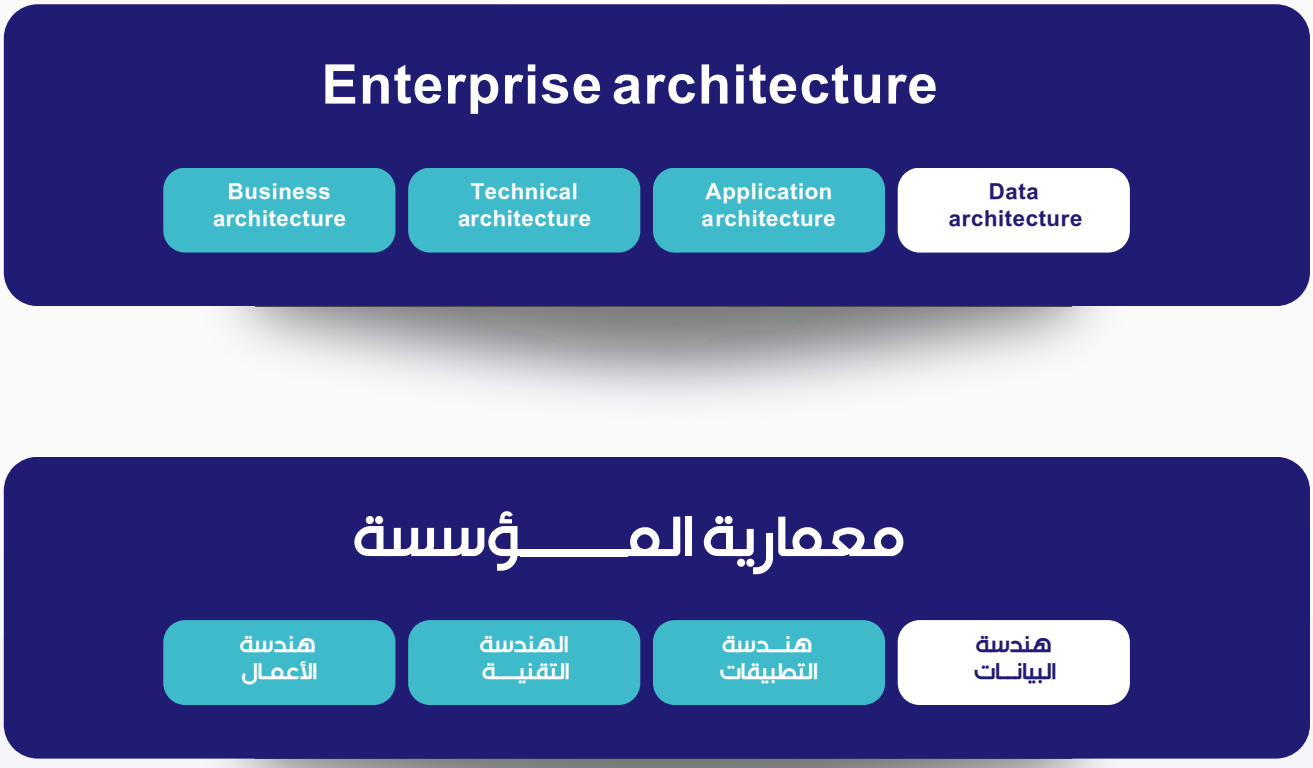
الفصل الثالث

تصميم بنية البيانات

يعتبر تصميم بنية بيانات فعّالة عنصرًا أساسيًا في بناء أنظمة معلوماتية مرنة وقابلة للتكامل. فبنية البيانات لا تقتصر على الجانب التقني فقط، بل تمثل إطارًا استراتيجيًا لإدارة دورة حياة البيانات بما يضمن قابلية التوسع، الأمان، والفعالية من حيث التكلفة. سنستعرض في هذا الفصل محورين رئيسين لفهم وتطبيق تصميم بنية البيانات، المبادئ الأساسية للبنية الجيدة، وتطور معماريات البيانات.

3.1 تصميم بنية بيانات فعّالة الأساس لنظام متكامل ومرن

بنية البيانات الجيدة هي المحرك الأساسي لنجاح الأنظمة الحديثة. فهي ليست مجرد إطار تقني، بل استراتيجية شاملة لدورة حياة البيانات، تركز على المرونة، التوسع، والأمان، مع تحقيق توازن مثالي بين الكفاءة والتكلفة. إليك المبادئ الأساسية (GU1) التي يجب مراعاتها عند تصميم بنية بيانات فعّالة.



3.2 ما هي بنية البيانات؟

Data architecture		هندسة البيانات	
Operational	Technical	التشغيل	التقنية
What	*How*	ماذا	كيف

بنية البيانات هي الإطار الفني والتنظيمي لتخزين ومعالجة وتبادل البيانات داخل الأنظمة. تختلف المعايير المثلى حسب احتياجات الشركات، لكن السمات الأساسية للبنية الفعالة تشمل:

01 المرونة:
قدرة النظام على التكيف مع تغييرات الأعمال أو التكنولوجيا.

02 التوسع:
دعم النمو المستقبلي دون التأثير على الأداء.

03 الأمان:
حماية البيانات من التهديدات.

04 إدارة التكلفة (FinOps):
تحقيق الكفاءة المالية.

05 البساطة:
تسهيل الصيانة والتطوير عبر تقليل التعقيد.

3.3 المبادئ الأساسية لبنية البيانات الجيدة

اختيار المكونات المشتركة بحكمة

- الممارسة: استخدم أدوات قياسية تعزز التعاون بين الفرق مع الحفاظ على خصوصية احتياجات كل فريق.
- التحدي: تجنب الإفراط في التعميم الذي قد يعيق الابتكار.

التخطيط للفشل

- الهدف: توقع الأعطال وتصميم حلول للتعافي السريع مثل استراتيجيات (RTO - Recovery Time Objective) و (RPO - Recovery Point Objective) لضمان استمرارية الأعمال.
- المثال: الأنظمة الموزعة تضمن استمرار العمليات حتى في حالة فشل أحد المكونات، مثل تقنيات التعافي السريع (Fault Tolerance) المستخدمة في الأنظمة السحابية.

تصميم للتوسع

- المفهوم: أنظمة تدعم التوسع الأفقي (إضافة موارد) والمرونة في إدارة الأحمال الموسمية.
- المثال: الأنظمة السحابية بدون خوادم، مثل (AWS Lambda Azureg Functions) تتكيف تلقائيًا مع حجم الاستخدام.

التصميم المستمر

- النظرة المستقبلية: البنية ليست ثابتة، ويجب أن تتطور مع تغييرات الأعمال والتكنولوجيا.
- التطبيق: احتفظ بتصميم مرن يسهل تعديله عبر الزمن لضمان التكيف مع التقنيات الجديدة.

بناء أنظمة مترابطة بشكل ضعيف

- الفائدة: مكونات مستقلة تسهل التطوير والنشر دون تأثير على الأجزاء الأخرى.
- المثال: تقسيم الأنظمة إلى خدمات مصغرة تعتمد على واجهات (API) مثل النهج الذي يتبعه معظم تطبيقات الخدمات السحابية الحديثة.

اتخاذ قرارات قابلة للتراجع

- المفهوم: اتخذ قرارات تقنية يمكن تعديلها بسهولة مستقبلاً.
- الاستراتيجية: استخدام أدوات وتقنيات ذات استبدال مرن لتجنب قفل النظام في تقنيات قديمة.

إعطاء الأولوية للأمان

- النموذج: اعتماد نموذج "الأمان بدون ثقة" (Zero Trust) وتوزيع المسؤولية الأمنية بين الفرق.
- التطبيق: تصميم حماية متعددة الطبقات لكل مكون من النظام، مثل استخدام تشفير البيانات، والتحقق الثنائي (2FA)، ورصد الأنشطة المشبوهة.

تبني (FinOps)

- التكامل: تعزيز التعاون بين الفرق الهندسية والمالية لتحسين إدارة التكاليف السحابية.
- التحدي: مراقبة الموارد ديناميكيًا لتحقيق أعلى كفاءة في استخدام الخدمات السحابية، مثل استخدام أدوات مثل (AWS Azureg Cost Explorer Cost Management).

3.4 تطور معماريات البيانات

مستودع البيانات (Data Warehouse):

- تخزن البيانات المنظمة وتدعم الاستعلامات السريعة باستخدام أنظمة معالجة موازية.
- أمثلة: (Amazon Redshift، Google BigQuery، Snowflake).

بحيرة البيانات (Data Lake):

- تخزن البيانات بجميع أنواعها دون هيكلة مسبقة، مع تحديات في الإدارة والتكاليف.
- يمكن استخدامها لتحليل البيانات غير المهيكلة مثل النصوص والصور والفيديوهات.

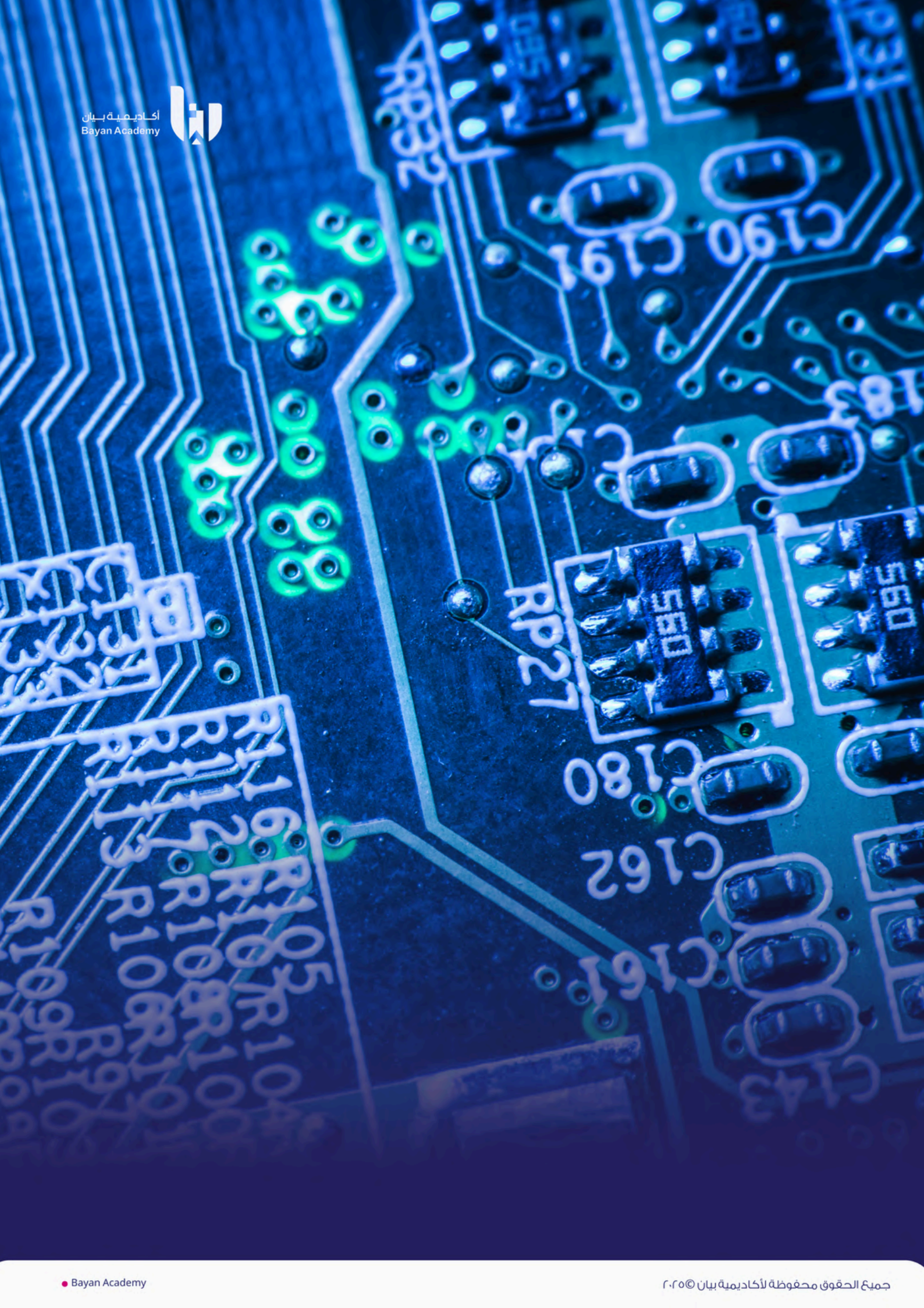
مستودع بحيرة البيانات (Data Lakehouse):

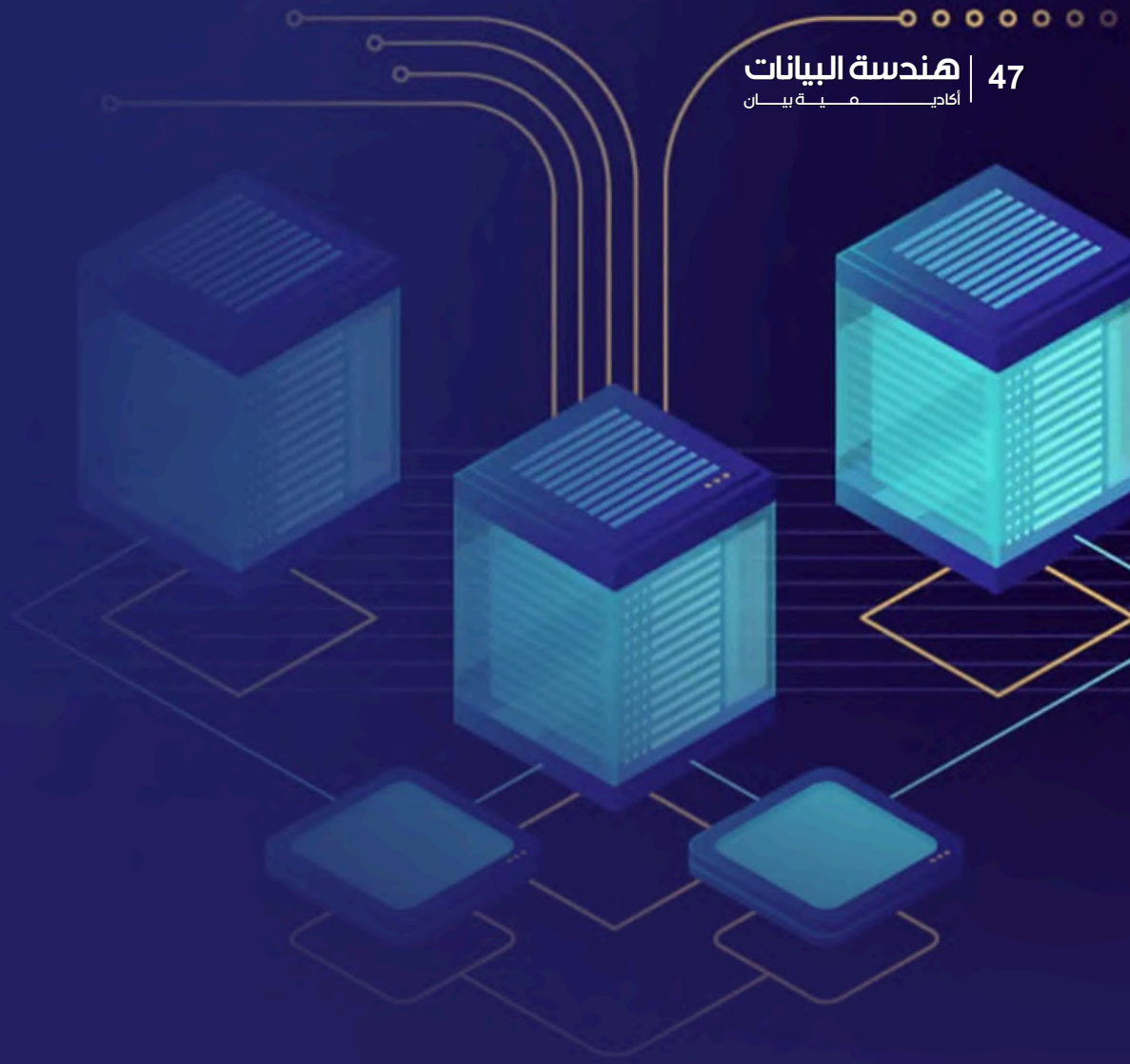
- تجمع بين إدارة البيانات المنظمة وغير المنظمة، مع دعم التحليلات المتقدمة.
- أمثلة: (Delta Lakeg Databricks Lakehouse).

في ختام الفصل الثالث

تصميم بنية البيانات

تصميم بنية بيانات فعالة يتطلب منظورًا شاملاً يوازن بين التقنية، المرونة، وإدارة التكاليف. فهو ليس مجرد تنفيذ خطوات فنية، بل يمثل حجر الأساس لبنية تحتية قادرة على التطور والتكيف مع المستقبل، وتحقيق أقصى استفادة من البيانات. ومع تسارع الابتكارات التكنولوجية، ستظل هذه الأنظمة تتطور لتتيح للشركات استخدام البيانات بذكاء واستراتيجية. أما التحديات المستقبلية، فقد تتمثل في مواكبة التقنيات السحابية المتقدمة أو تحقيق التوازن بين التكلفة والأداء. ولهذا، من الضروري أن تواصل المؤسسات تحسين بنية بياناتها باستمرار لضمان جاهزيتها لما هو قادم.





04
إنتاج البيانات

Data production

الفصل الرابع

إنتاج البيانـــات

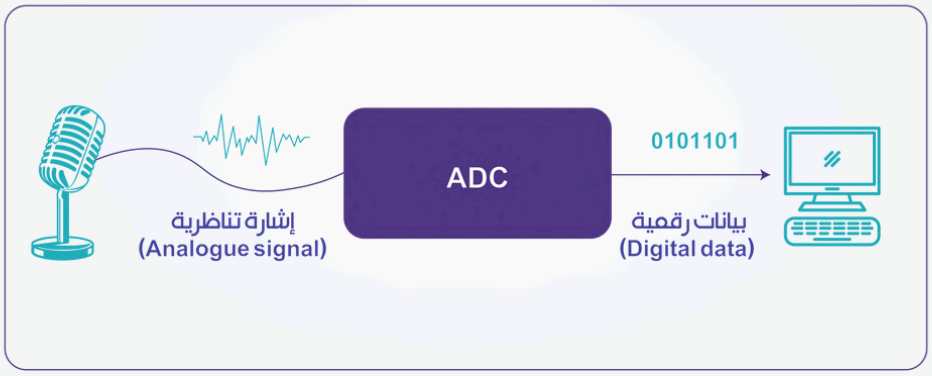
يمثل إنتاج البيانات من الأنظمة المصدرية نقطة الانطلاق الأساسية في هندسة البيانات، حيث يُبنى عليها فهم شامل لطبيعة البيانات ومصادرها. يتناول هذا الفصل مفاهيم محورية تشمل تصنيف البيانات، أنواع الأنظمة المصدرية، المبادئ الأساسية لإدارة البيانات مثل (ACID وCRUD) ، إضافة إلى الجوانب العملية للتكامل، الأمان، والتعامل مع قواعد البيانات العلائقية (Relational Database) وغير العلائقي(GU1ة(Non-Relational Database). تهدف هذه المحاور إلى تمكين بيئة بيانات موثوقة وآمنة تدعم التحليل والتشغيل بكفاءة.

4.1 مقدمة عن انتاج البيانات

تعد عملية انتاج البيانات من الأنظمة المصدرية من أساسيات هندسة البيانات. تُعتبر أنظمة المصدر (Source Systems) هي النقطة الأولى التي يتم فيها جمع البيانات، حيث يجب على المهندسين فهم مصادر البيانات وخصائصها لكي يتمكنوا من التعامل معها بفعالية. تلعب هذه الأنظمة دورًا حاسمًا في تحديد كيفية استخدام البيانات لاحقًا في العمليات التحليلية أو المعاملات.

4.2 مصادر البيانات : البيانات التناظرية (Analog Data) مقابل البيانات الرقمية (Digital Data)

الشكل (4) : مصـــادر البيـــانات



البيانات التي تُنشأ تأتي بشكلين رئيسيين كما هو موضح بالشكل (4):

البيانات التناظريـــة (ANALOG DATA)

هي بيانات تُنتج من الظواهر الطبيعية في العالم الحقيقي، مثل الصوت البشري أو الكتابة اليدوية، وتُعبر عن معلومات عبر إشارات مستمرة يمكن أن تأخذ قيمًا غير محدودة ضمن نطاق معين. لا يمكن معالجتها مباشرة بالحواسيب الرقمية، بل تحتاج إلى تحويلها إلى بيانات رقمية أولاً.

مثال: المكالمات الصوتية عبر الهاتف، ملاحظات مكتوبة يدويًا.

البيانات الرقميـــة (DIGITAL DATA)

هي بيانات تُعبّر عنها بأرقام ثنائية (0و1)، ويمكن إنشاؤها مباشرة بواسطة الأنظمة الرقمية أو تحويلها من بيانات تناظرية. تتميز بأنها متقطعة، وسهلة التخزين والمعالجة باستخدام الحواسيب.

مثال: المعاملات المالية عبر الإنترنت، بيانات الاستشعار من أجهزة إنترنت الأشياء (IoT).

4.3 أنواع أنظمة المصدر الأساسية (Types of Source Systems)

تتنوع الأنظمة التي تولد البيانات بشكل كبير، وأهمها:

الملفات (Files):

التعريف:
تُستخدم كوسيلة لتبادل البيانات، وتشمل تنسيقات مثل(Excel، CSV، JSON، XML).

أنواع الملفات:
قد تكون هذه الملفات مُنظَّمة (Structured)، شبه مُنظَّمة (Semi-Structured)، أو غير مُنظَّمة (Unstructured) كما هو موضح بالشكل(5).

مثال:
ملف (CSV) يحتوي على بيانات مبيعات.

الشكل (5) : أنـــواع الملفـات

واجهات برمجة التطبيقات (APIs - Application Programming Interfaces):

التعريف:
تُستخدم لتبادل البيانات بين الأنظمة، وغالبًا تتطلب صيانة مخصصة.

مثال:
واجهة (API) لتبادل البيانات بين تطبيق التجارة الإلكترونية ومزود خدمات الدفع.

تتنوع الأنظمة التي تولد البيانات بشكل كبير، وأهمها:

قواعد بيانات المعاملات (OLTP - Online Transaction Processing):

التعريف:

تُستخدم لإجراء العمليات الفورية والمعاملات، وتدعم التفاعل مع العديد من المستخدمين في وقت واحد.

مثال:

قاعدة بيانات تُخزّن طلبات العملاء في متجر إلكتروني.

أنظمة التحليل عبر الإنترنت(OLAP - Online Analytical Processing):

التعريف:

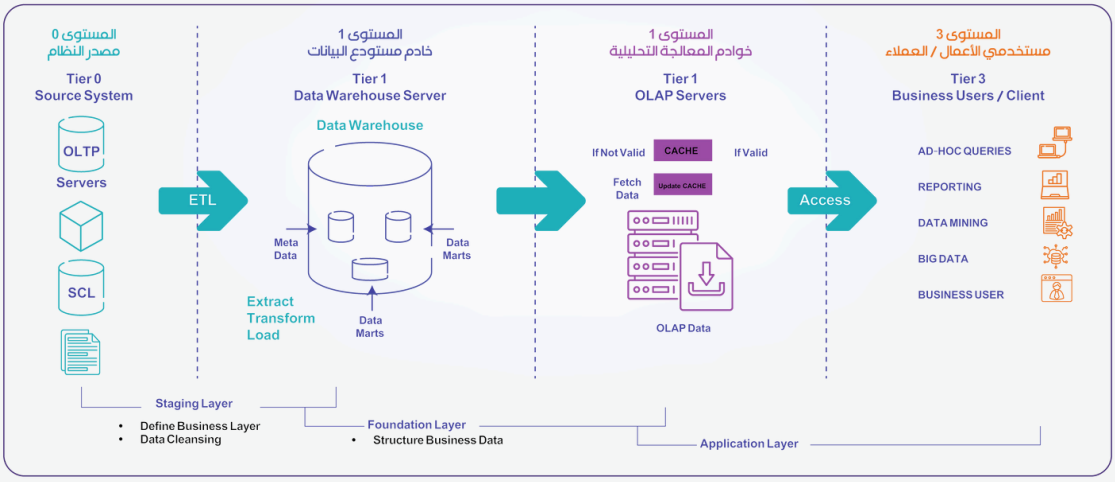
مخصصة لتحليل كميات كبيرة من البيانات.

مثال:

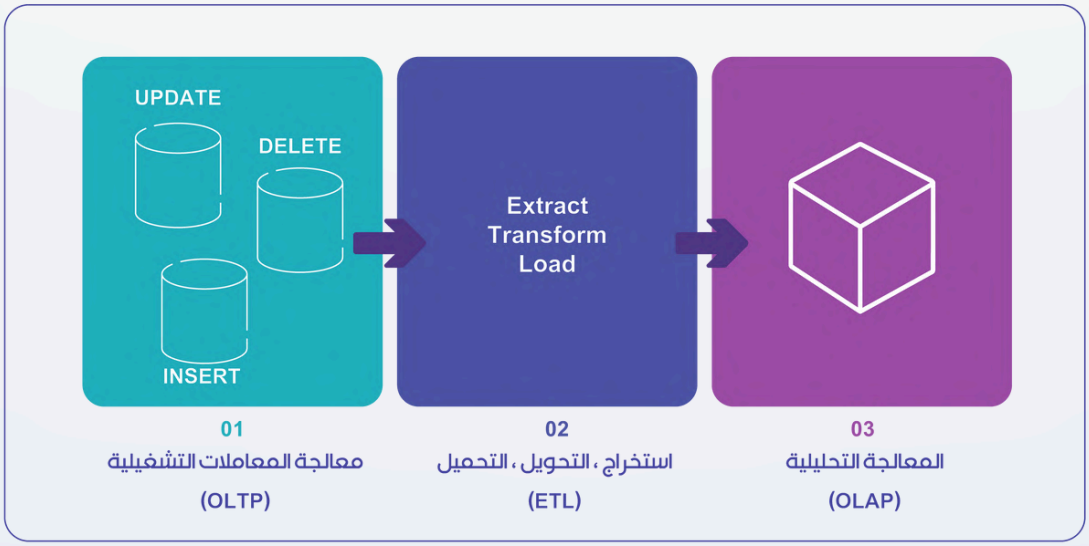
النظام الذي يستخدم لتحليل بيانات المبيعات التاريخية لمتاجر البيع بالتجزئة.

الشكل (6) : عملية تحليل البيانات عبر الانترنت (OLAP - Online Analytical Processing)

تحليل البيانات بفعالية، مما يتيح اتخاذ قرارات في الوقت الفعلي والحصول على رؤى استراتيجية طويلة المدى.
analyze data effectively, enabling real-time decision-making and long-term strategic insights.



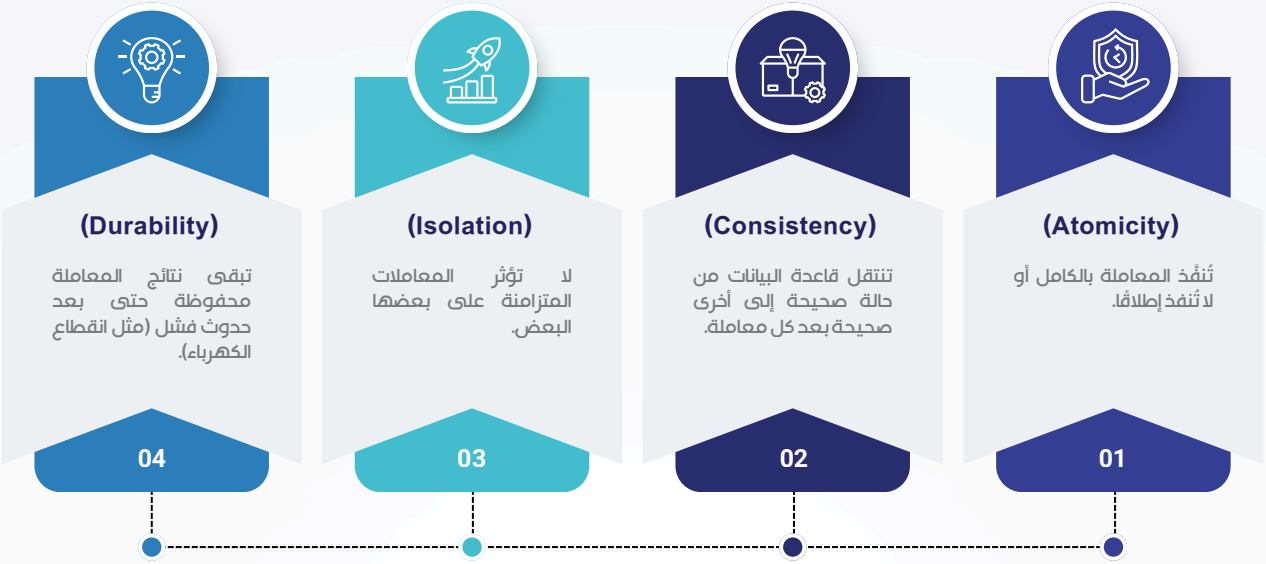
الشكل (7) : المراحل الأساسية لمعالجة البيانات عبر الإنترنت



4.4 مفاهيم رئيسية في إدارة البيانات (Key Concepts in Data Management):

4.4.1 خصائص ACID (Atomicity, Consistency, Isolation, Durability)

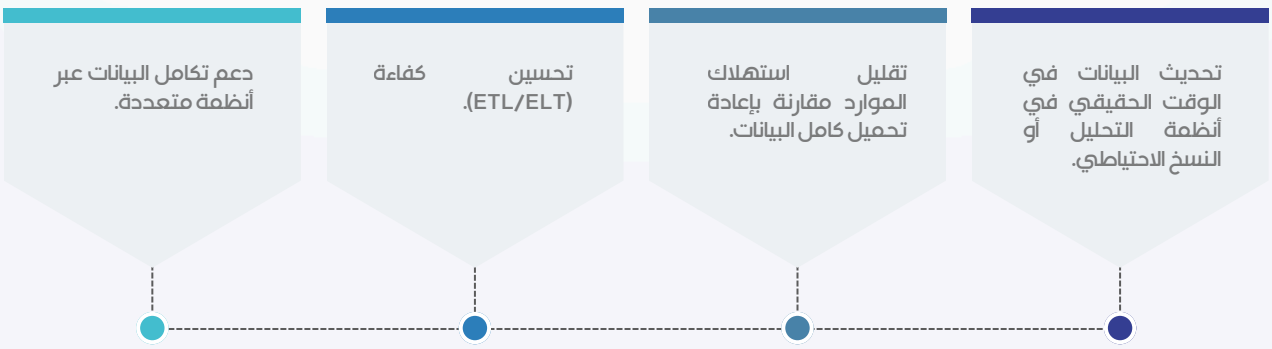
هي مجموعة من المبادئ تضمن تنفيذ المعاملات في قواعد البيانات بشكل موثوق، وتشمل:



تضمن خصائص (ACID) سلامة البيانات ومنع الفساد في أنظمة قواعد البيانات، خاصة في التطبيقات الحساسة مثل المعاملات المصرفية أو أنظمة الحجز.

4.4.2 استخراج التغييرات في البيانات(CDC - Change Data Capture)

استخراج التغييرات في البيانات (CDC - Change Data Capture) هي تقنية تُستخدم لاكتشاف وتسجيل التغييرات (إضافة، تعديل، حذف) التي تحدث في مصادر البيانات، مثل قواعد البيانات التشغيلية، بهدف تمكين التحديثات المستمرة للأنظمة التحليلية أو نُظم التخزين الأخرى.



مثال: عند تحديث عنوان أحد العملاء في قاعدة البيانات، تقوم تقنية (CDC) بتحديد هذا التغيير وإرساله تلقائيًا إلى مستودع البيانات التحليلي ليبقى محدّثًا دون الحاجة لإعادة تحميل كل جدول العملاء.

4.4.3 السجلات (Logs)

هي ملفات أو تدفقات بيانات تسجل الأحداث التي تحدث في الأنظمة أو التطبيقات، وتحتوي عادة على معلومات مثل: من قام بالعملية، وماذا حدث، ومتى حدث ذلك. تُستخدم لمراقبة النظام، تتبع الأخطاء، وتحليل الأمان.

مثال:عند محاولة مستخدم تسجيل الدخول إلى تطبيق ما، يتم تسجيل الحدث في ملف السجلات متضمناً اسم المستخدم، نوع الحدث (محاولة دخول)، والتاريخ والوقت.

4.5 ممارسات عملية (Practical Considerations)

فهم وثائق الأنظمة المصدرية (Source System Documentation)

الاطلاع على وثائق الأنظمة المصدرية ضروري لفهم هيكل البيانات، القيود، أنواع الحقول، وسجلات الأعمال، ما يضمن استيعاب البيانات بشكل دقيق ويسهم في تقليل الأخطاء أثناء التكامل أو التحليل.

السجلات كأداة لضمان التناسق (Logs for Consistency)

تُعد سجلات النظام أداة مهمة لمراقبة تدفق البيانات وتحديد حالات الفشل أو الانحرافات، مما يساعد على ضمان توافر البيانات وتناسقها عبر مراحل النقل والمعالجة.

التعامل مع الأنظمة الهجينة (Hybrid Systems)

عند الجمع بين الأنظمة التشغيلية (OLTP) والتحليلية (OLAP)، يصبح من الضروري فهم كيفية تفاعل البيانات في الزمن الحقيقي أو شبه الحقيقي، والتعامل مع تحديات مثل تأخر التحديث أو تكرار البيانات، لضمان تكامل شامل بين العمليات والتحليلات.

4.6 عمليات (CRUD Operations) الأساسيات والأهمية

(CRUD) هو اختصار يشير إلى أربع عمليات أساسية تُستخدم في إدارة البيانات ضمن التطبيقات التي تعتمد على قواعد البيانات. هذه العمليات تشكّل الأساس لأي نظام يتعامل مع البيانات، سواء كانت قاعدة بيانات علائقية (RDBMS) أو غير علائقية (NoSQL) أو حتى خدمات واجهات برمجة التطبيقات (APIs).

العمليات الأربع في (CRUD):

إنشـــــــاء (Create):

- تمثل عملية إدخال بيانات جديدة إلى النظام.
- مثال: إضافة مستخدم جديد إلى قاعدة بيانات تحتوي على بيانات العملاء.

تحديـــــث (Update):

- تُستخدم لتعديل بيانات موجودة بالفعل داخل النظام.
- مثال: تحديث عنوان البريد الإلكتروني للمستخدم.

قـــــــراءة (Read):

- هي عملية استرجاع البيانات الموجودة من قاعدة البيانات أو النظام.
- تُستخدم هذه العملية لاستعلام أو عرض بيانات معينة.
- مثال: استرجاع بيانات المستخدم بناءً على معرفه الفريد (ID).

حـــــــذف (Delete):

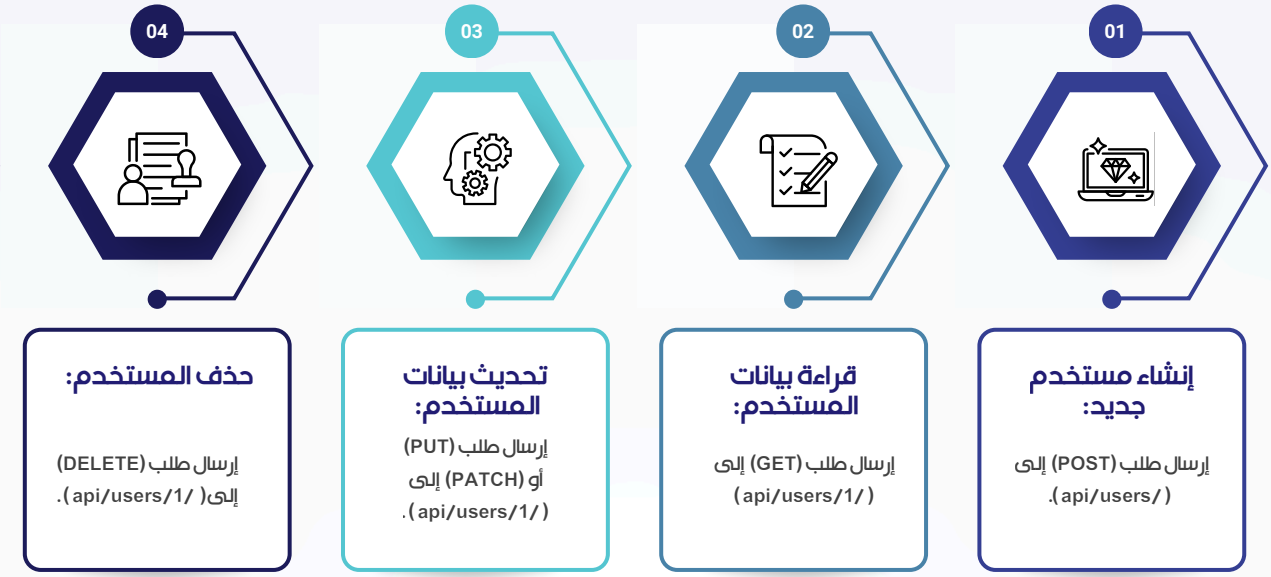
- تعني إزالة بيانات غير مرغوب فيها أو غير مطلوبة من النظام.
- مثال: حذف حساب مستخدم بناءً على معرفه.

أهمية (CRUD):

1. **إدارة البيانات:** تُعتبر العمليات الأربعة الأساس في أي تطبيق يعمل مع البيانات، مما يجعلها ضرورية لتخزين البيانات واسترجاعها وتعديلها بشكل فعّال.
2. **تفاعل المستخدم:** معظم التطبيقات تعتمد على هذه العمليات لتلبية احتياجات المستخدمين، مثل إنشاء حساب جديد، تحديث البيانات الشخصية، أو حذف عنصر.
3. **التكامل مع الأنظمة الأخرى:** تُستخدم (CRUD) بشكل شائع مع واجهات برمجة التطبيقات (APIs) التي تسمح للأنظمة بالتواصل ومشاركة البيانات.

4.6.1 التكامل مع واجهات برمجة التطبيقات (APIs)

في التطبيقات الحديثة، تُستخدم عمليات (CRUD) بشكل شائع ضمن واجهات(RESTful APIs) لتوفير وظائف البيانات الأساسية عبر الإنترنت. كل عملية من (CRUD) ترتبط بطريقة (HTTP Method) محددة وتُطبّق على مسارات (Endpoints) معينة.



هذه العلاقة بين(CRUD وREST) تجعل من الممكن إنشاء أنظمة مرنة وقابلة للتكامل مع واجهات خارجية، مثل تطبيقات الجوال أو أنظمة شركاء الأعمال.

4.7 نمط الإدراج فقط (Insert-Only Pattern)

نمط تصميم يُستخدم في أنظمة البيانات حيث يتم **إدراج سجل جديد** لكل تغيير بدلاً من تعديل السجلات القديمة.

الآلية:

1. تُسجّل كل عملية تعديل أو تحديث كسطر جديد في الجدول.
2. يُستخدم الطابع الزمني (Timestamp) أو معرف الإصدار (Version ID) لتتبع تسلسل التغييرات.

الفوائد:

1. **التدقيق (Auditability):** يمكن الرجوع إلى النسخ السابقة من البيانات بسهولة.
2. **عدم فقدان البيانات:** لا يتم حذف أو تعديل السجلات الأصلية.
3. **المرونة في التحليلات الزمنية:** يمكن تحليل البيانات كما كانت في أي وقت سابق.

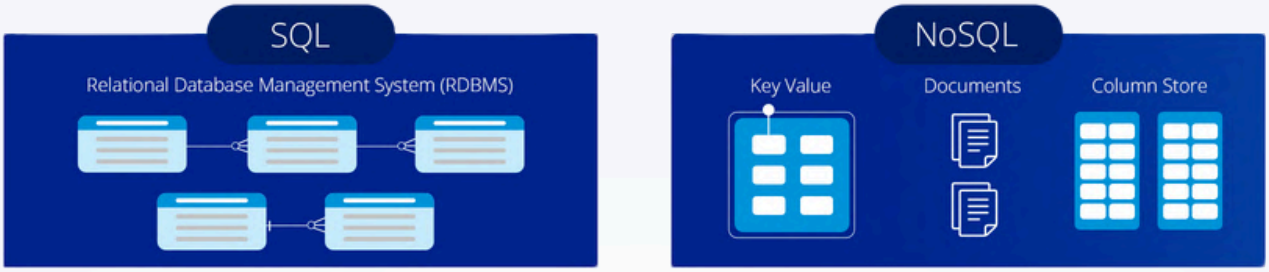
التحديات:

1. **نمو حجم البيانات بسرعة:** بسبب تراكم السجلات مع كل تغيير.
2. **تعقيد الاستعلامات:** تحتاج إلى تحديد أحدث سجل لكل كيان.

أمثلة الاستخدام:

1. أنظمة تتبع العمليات المالية.
2. نظم ذكاء الأعمال التي تحتاج إلى تاريخ البيانات الكامل (Data Lineage).

4.8 ملخص شامل عن قواعد البيانات العلائقية (RDBMS - Relational Databases) وغير العلائقية (NoSQL Databases)



قواعد البيانات غير العلائقية (NOSQL DATABASES)

لا تعتمد على الجداول؛ تعتمد على نماذج مرنة مثل المستندات أو المفاتيح والقيم. صُممت للتعامل مع البيانات الضخمة والمتغيرة بسرعة.

المزايا:

- مرونة في تصميم نماذج البيانات (Schema-less).
- أداء عالي مع البيانات الكبيرة أو عند وجود أحجام قراءة/كتابة كبيرة.
- قابلة للتوسع الأفقي بسهولة.
- مناسبة للتطبيقات الحديثة مثل تطبيقات الجوال , إنترنت الاشياء (IoT)، وتحليلات الوقت الحقيقي.

التحديات:

- دعم محدود للاستعلامات المعقدة أو العلاقات العميقة بين الكيانات.
- خصائص (ACID) غالبًا غير مفعّلة بالكامل (خصوصًا التناسق الفوري).
- تحتاج إلى معرفة خاصة بكل نوع نظرًا لتنوعها الكبير.

الأنواع مع أمثلة:

- مخازن المفاتيح والقيم (Key-Value Stores): (Redis، DynamoDB)
- مخازن الوثائق (Document Stores): (MongoDB، Couchbase)
- قواعد البيانات ذات الأعمدة الواسعة (Wide Column Stores): (Cassandra، HBase)
- قواعد البيانات البيانية (Graph Databases): (Neo4j، Amazon Neptune)

استخدم (NOSQL)

إذا كانت البيانات ضخمة، متغيرة البنية، أو إذا كنت تحتاج إلى أداء وسرعة عالية في القراءة والكتابة.

قواعد البيانات العلائقية (RELATIONAL DATABASES - RDBMS)

تعتمد على نموذج الجداول (صفوف وأعمدة). تُستخدم لغة (SQL) للاستعلام وتنفيذ العمليات.

المزايا:

- تدعم خصائص (ACID) (الذرية، التناسق، العزل، المتانة).
- مناسبة للبيانات المنظمة التي تحتوي على علاقات واضحة.
- قوية في تنفيذ الاستعلامات المعقدة والانضمامات (Joins).
- ناضجة وتتمتع بدعم مجتمعي واسع وأدوات متقدمة.

التحديات:

- أقل مرونة في التعامل مع البيانات غير المنظمة أو المتغيرة الهيكل.
- صعوبة في التوسع الأفقي (Horizontal Scaling).
- قد لا تكون مثالية للبيانات الضخمة أو ذات السرعة العالية في التغيير.

الأنواع مع أمثلة:

- (MySQL)
- (PostgreSQL)
- (Oracle Database)
- (Microsoft SQL Server)

استخدم (RDBMS)

إذا كانت بياناتك منظمة، والعلاقات معقدة، وتحتاج إلى ضمان قوي للتكامل والدقة.

4.9 إدارة البيانات في الأنظمة المصدرية (Data Management in Source Systems)

يجب على المهندسين التأكد من أن البيانات في الأنظمة المصدرية محمية أثناء التخزين والنقل. من الضروري التأكد من:

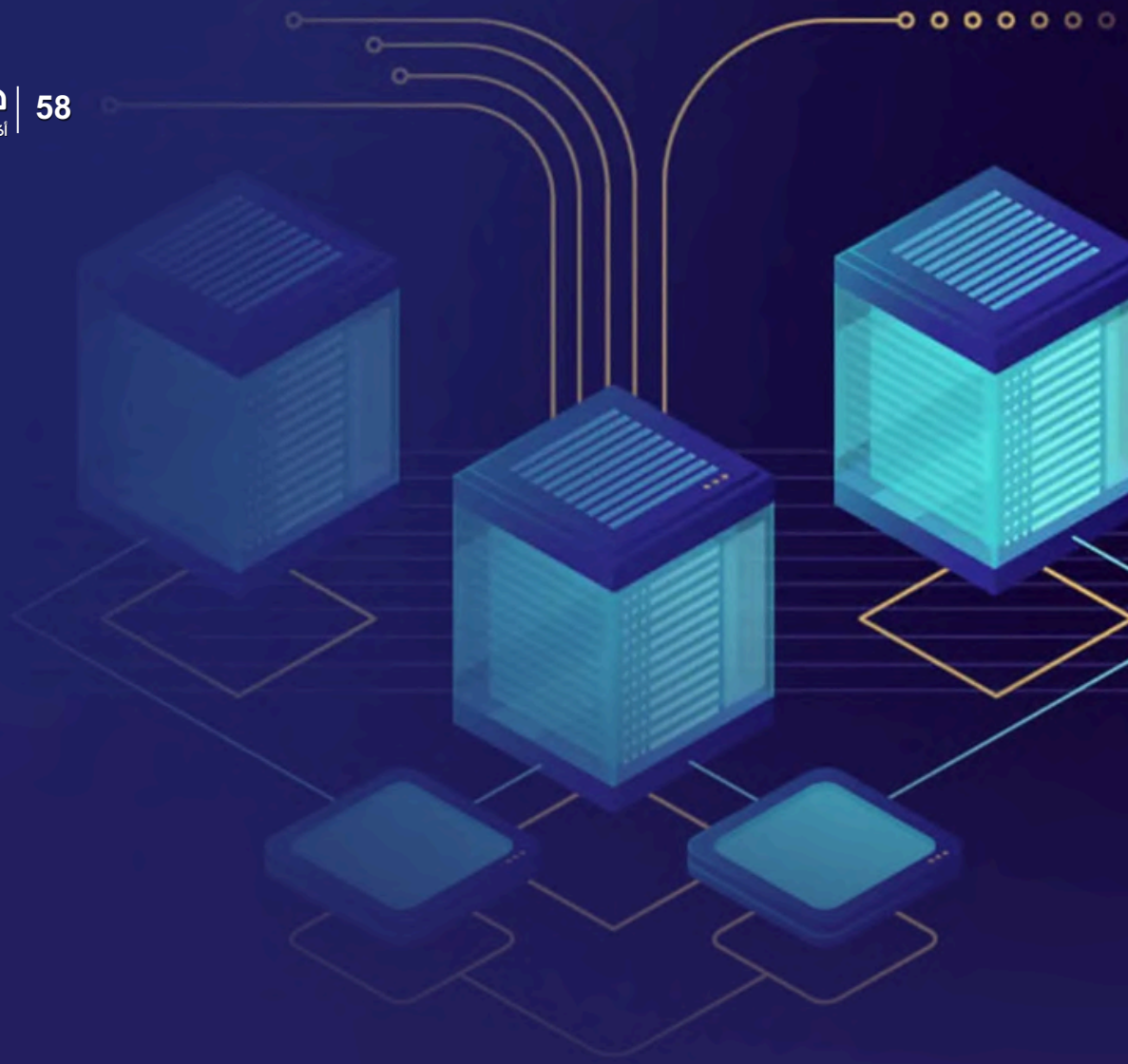
التشفير (Encryption):
تأكيد أن البيانات مشفرة بشكل مناسب.

التحقق من الوصول (Access Control):
التأكد من أن الوصول يتم عبر شبكات آمنة.

إدارة المفاتيح (Key Management):
الحفاظ على سرية المفاتيح وكلمات المرور.

تواجه مهندسي البيانات تحديات كبيرة

في التعامل مع البيانات في الأنظمة المصدرية، حيث غالبًا ما يكون لديهم تحكم ضئيل في طريقة إدارة البيانات. من الضروري التعاون مع فرق دعم الأنظمة لضمان تحقيق أفضل ممارسات لإدارة البيانات.



في ختام الفصل الرابع

إنتاج البيانات

تعرفنا أن عملية إنتاج البيانات من الأنظمة المصدرية يعتبر عنصر أساسي في هندسة البيانات، حيث تشكل نقطة البداية لدورة حياة البيانات. ويتطلب التعامل مع هذه البيانات فهمًا عميقًا لمصادرها وأنواعها المختلفة. كما تمثل مفاهيم مثل (CRUDgACID) والتكامل بين الأنظمة عناصر أساسية في إدارة البيانات، بينما يظل تأمين البيانات وضمان سلامتها من العوامل الحاسمة لضمان فعالية الأنظمة المصدرية.

05
تخزين البيانات

Data storage

الفصل الخامس

تخزين البيانات

التخزين هو عنصر أساسي في أي نظام إدارة بيانات، سواء كان ذلك في المؤسسات الصغيرة أو الشركات الكبرى. اختيار استراتيجية التخزين المناسبة يساعد في تحسين الأداء، تقليل التكاليف، وضمان استمرارية الأعمال. في هذا الفصل، سنستعرض استراتيجيات التخزين المختلفة، متى يتم استخدامها، وما مزايا وعيوب كل منها.

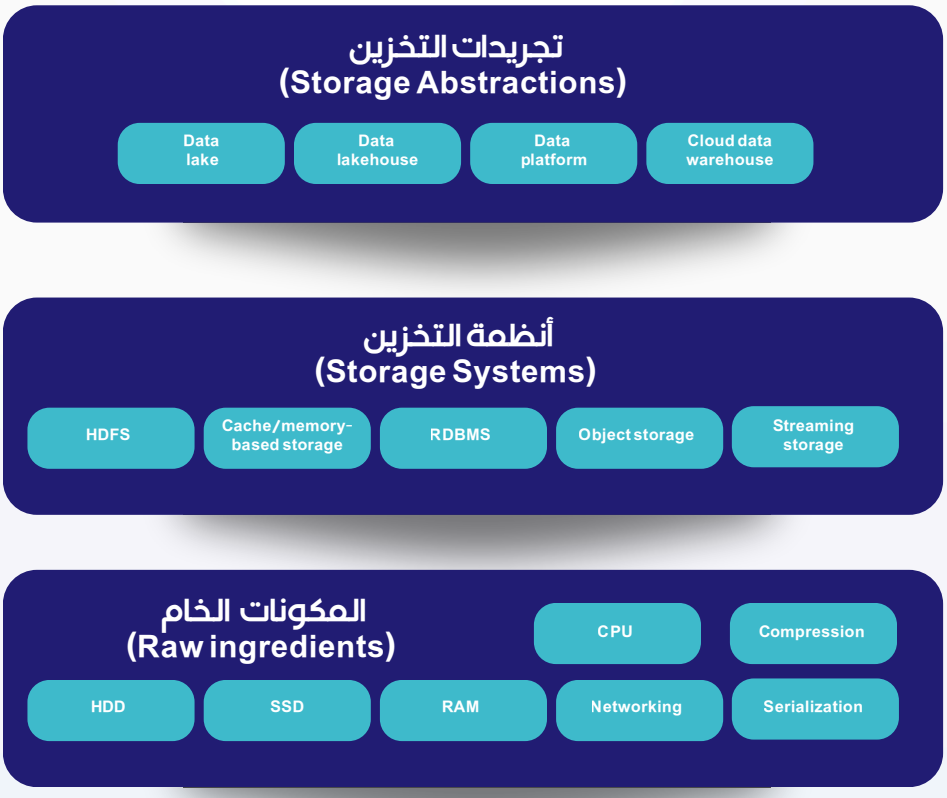
5.1 أهمية التخزين في هندسة البيانات

- التخزين هو الركيزة الأساسية لدورة حياة هندسة البيانات، ويؤثر بشكل مباشر على عمليات الإدخال، التحويل، والتقدير.
- البيانات تنتقل عبر مراحل متعددة، مما يتطلب تخزينها عدة مرات قبل أن تستهلكها الأنظمة المختلفة.
- التخزين يجب أن يكون فعالاً وآمنًا وقابلاً للتطوير لضمان الاستخدام الأمثل للبيانات.

5.2 الفرق بين التخزين في الأنظمة المصدر والتخزين في هندسة البيانات

01 **الأنظمة المصدر:**
غالبًا غير مُدارة أو متحكم فيها من قبل مهندسي البيانات.

02 **التخزين في هندسة البيانات:**
يتم التحكم به مباشرة لدعم دورة الحياة الكاملة للبيانات، من الإدخال إلى التحليل والذكاء الاصطناعي.



5.3 مكونات أنظمة التخزين الأساسية

محركات الأقراص المغناطيسية(HDDs):

- تعتمد على أسطوانات دوارة مغناطيسية لتخزين البيانات.
- تكلفتها منخفضة (3 سنت لكل جيجابايت).
- مناسبة للتخزين الضخم لكنها بطيئة مقارنة بـ SSDs.
- ابتكارات حديثة مثل HAMR وSMR والأقراص المملوءة بالهيليوم زادت من سعتها إلى 20 تيرابايت.

محركات أقراص الحالة الصلبة(SSDs):

- أسرع من HDDs لكنها أغلى تكلفة.
- مثالية للوصول السريع للبيانات الحرجة مثل البيانات الساخنة.

الذاكرة العشوائية (RAM):

- تُستخدم في التخزين المؤقت والمعالجة السريعة.
- غير دائمة، مما يعني الحاجة إلى تدابير حماية من فقدان البيانات عند انقطاع الكهرباء.

الشبكات ووحدات المعالجة المركزية (CPUs):

- الشبكات أصبحت جزءًا أساسيًا من التخزين، خاصة في الأنظمة السحابية والموزعة.
- وحدات المعالجة المركزية تلعب دورًا في تسريع استرجاع البيانات وضغطها وتحليلها.

الشكل (8) : هيكلية هرمية للذاكرة المؤقتة (الكاش) توضح أنواع التخزين المختلفةمع تقديرات تقريبية للأسعار ومستويات الأداء الخاصة بكل نوع

Storage type	Data fetch latency[a]	Bandwidth	Price
CPU cache	1 nanosecond	1 TB/s	N/A
RAM	0.1 microseconds	100 GB/s	$10/GB
SSD	0.1 milliseconds	4 GB/s	$0.20/GB
HDD	4 milliseconds	300 MB/s	$0.03/GB
Object storage	100 milliseconds	10 GB/s	$0.02/GB per month
Archival storage	12 hours	Same as object storage once data is available	$0.004/GB per month

[a] A microsecond is 1,000 nanoseconds, and a millisecond is 1,000 microseconds.

5.4 استراتيجيات التخزين: المفهوم، الأنواع، والاستخدامات

الأنواع	المفهوم	المزايا	العيوب	الاستخدامات المثلى
التخزين التقليدي	تخزين البيانات داخليًا على خوادم محلية داخل المنشأة.	• تحكم كامل • سرعة وصول عالية • أمان محلي	• تكاليف مرتفعة • صعوبة التوسع • إدارة معقدة	• المؤسسات ذات المتطلبات الأمنية العالية مثل: البنوك • الشركات: التي تحتاج إلى أداء عالٍ واستجابة فورية
التخزين السحابي	تخزين البيانات لدى مزودي خدمات سحابية (مثل AWS، Azure، Google Cloud).	• مرونة عالية • تكلفة أقل مقدمًا • الوصول من أي مكان • نسخ احتياطي وتعافي من الكوارث	• يعتمد على الإنترنت • مخاوف الأمان والخصوصية • تكاليف اشتراك مستمرة	• الشركات الناشئة • المؤسسات ذات الفرق الموزعة جغرافيا • التطبيقات التي تتطلب مشاركة بيانات سريعة وآمنة
التخزين الهجين	دمج بين التخزين المحلي والسحابي لتخزين البيانات الحساسة محليًا والباقي سحابيًا.	• يجمع بين الأمان والمرونة • تحسين الأداء • خفض التكاليف	• إدارة معقدة • تكاليف إضافية للتكامل	• الشركات التي تتعامل مع بيانات حساسة وتحتاج في الوقت ذاته إلى تخزين بيانات ضخمة بأمان ومرونة.
التخزين الموزع	توزيع البيانات عبر خوادم متعددة حول العالم.	• أداء عالمي محسن • توافر عال • تعزيز الأمان	• إدارة معقدة (تكرار وتشفير) • تكاليف تشغيلية مرتفعة	• الشركات العالمية • التطبيقات التي تتطلب استجابة من عدة مواقع • خدمات تتطلب تكرار بيانات مثل شبكات (CDN)
تخزين الكائنات	تخزين البيانات ككائنات ببيانات وصفية (Metadata).	• قابلية توسع كبيرة • إدارة سهلة • تكلفة منخفضة	• أداء أبطأ لبعض التطبيقات • لا يدعم التعديل الجزئي	• أرشفة ونسخ احتياطي • تخزين وسائط كبيرة (صور، فيديوهات) • تطبيقات الذكاء الاصطناعي التي تتعامل مع بيانات ضخمة

5.5 استراتيجيات تحسين التخزين

التخزين متعدد المستويات (Tiered Storage):
استخدام مزيج من وحدات التخزين مثل الأقراص الصلبة (HDD)، الأقراص الحالة الصلبة (SSD)، وذاكرة الوصول العشوائي (RAM)، بهدف تحقيق توازن بين الأداء والتكلفة.

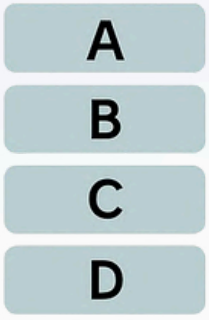
التخزين الموزع (Distributed Storage):
توزيع البيانات عبر عدة خوادم أو مراكز بيانات بهدف تحسين الأداء وزيادة التوافر (Availability) والاعتمادية.

تقنيات RAID (Redundant Array of Independent Disks):
تُستخدم لزيادة الأداء والمتانة من خلال التوازي في قراءة البيانات وكتابتها عبر أقراص متعددة، مع توفير الحماية من فقدان البيانات في بعض مستويات (RAID).

التخزين السحابي (Cloud Storage):
يوفر مرونة وتوسعًا كبيرًا في التخزين، مع إمكانية الوصول من أي مكان، لكنه يتطلب مراعاة دقيقة للتكلفة، الأمان، ومتطلبات الخصوصية.

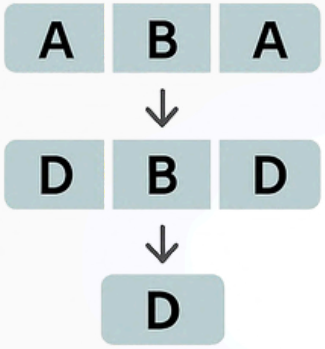
5.6 تخزين البيانات وأنظمته

5.6.1 أساسيات تخزين البيانات



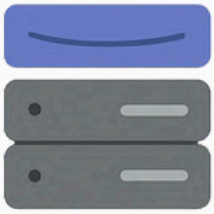
التسلسل (Ordering)

يؤثر على أداء الاستعلامات، واستهلاك المعالج، وزمن الاستجابة.



الضغط (Compression)

يقلل حجم البيانات ويزيد سرعة المسح عبر الأقراص.



التخزين المؤقت (Caching)

يخزن البيانات المتكررة في طبقة سريعة للوصول.

التخزين المؤقت (Caching) | الضغط (Compression) | التسلسل (Ordering)

5.6.2 أنظمة تخزين البيانات

التخزين الملفي (File Storage):
نظام تخزين يعتمد على تنظيم البيانات على شكل ملفات ومجلدات، ويُستخدم في أنظمة التشغيل والتطبيقات التقليدية.

تخزين الكتل (Block Storage):
يقسم البيانات إلى كتل منفصلة تُخزن بشكل مستقل، ويُستخدم في وحدات التخزين ذات الأداء العالي مثل(SSD) والأقراص المغناطيسية. مناسب لقواعد البيانات وأنظمة المعاملات.

RAID (مجموعة مكررة من الأقراص المستقلة):
نظام يدمج عدة أقراص لتوفير أداء أعلى وحماية ضد فقدان البيانات. تختلف مستويات(RAID) في التوازن بين الأداء والتكرار والتكلفة.

شبكة تخزين المناطق (SAN - Storage Area Network):
شبكة تخزين عالية الأداء تُستخدم لتوفير تخزين الكتل لأجهزة الخوادم في مراكز البيانات. تُدار بشكل مركزي وتتيح مشاركة فعالة للموارد.

التخزين الافتراضي للكتل (Cloud Block Storage):
يدعم الأداء العالي، مثل Amazon EBS وسرعة النقل كمقياس IOPS ويستخدم وحدات للأداء.

التخزين الموزع (Distributed Storage):
يُوفر التوسع الأفقي والموثوقية من خلال توزيع البيانات على عدة خوادم، مع إمكانيات معالجة متوازية واستعادة من الأعطال.

5.6.3 مقارنة بين أنظمة التخزين

الأنظمة	المزايا	العيوب
التخزين الملفي	مناسب للملفات الصغيرة وسهل الاستخدام	أقل كفاءة في الأداء مقارنة بالكتل
تخزين الكتل	أداء سريع وتحكم دقيق في التخزين	يتطلب إدارة معقدة
مجموعة مكررة من الأقراص المستقلة (RAID)	يحسن الأداء والموثوقية	يعتمد على نوع RAID المستخدم
شبكة تخزين المناطق (SAN)	تخزين سريع وعالي الأداء عبر الشبكة	تكلفة مرتفعة وتعقيد في الإدارة
التخزين السحابي	مرونة وتوسع حسب الحاجة	قد يكون مكلفًا على المدى الطويل

5.7 استراتيجيات التخزين والمعالجة في تحسين الأداء وتقليل التكاليف

قواعد البيانات العمودية (Columnar Databases)

تعتمد قواعد البيانات العمودية على تخزين البيانات حسب الأعمدة بدلاً من الصفوف، مما يوفر أداءً أعلى في عمليات التحليل والاستعلامات. يسمح هذا النمط من التخزين بقراءة بيانات محددة بسرعة أكبر، مما يجعله مثاليًا لعمليات التحليلات والاستعلامات التي تتعامل مع كميات كبيرة من البيانات. **يوضح الشكل (9)** مقارنة بين طريقة تخزين البيانات في قاعدة البيانات العمودية وطريقة تخزينها في قاعدة البيانات الصفية (Row-oriented database).

الشكل (9) : مقارنة بين تخزين البيانات في قاعدة البيانات العمودية و تخزينها في قاعدة البيانات الصفية

أهم الميزات:

- الكفاءة في معالجة البيانات الكبيرة.
- تحسين ضغط البيانات.
- سهولة المسح والتصفية.

التقسيم والتجميع:

01 التقسيم (Partitioning):

يتم تقسيم الجداول إلى أقسام أصغر بناءً على حقول محددة.

02 التجميع (Clustering):

يتم تجميع الصفوف المتشابهة معًا داخل القسم نفسه.

مثال تطبيقي:

Snowflake هو نظام تخزين بيانات سحابي يعتمد على تقسيم دقيق لتحسين أداء الاستعلامات.

5.8 التجريدات في هندسة البيانات(Abstractions in Data Engineering)

التجريدات تحدد كيفية تخزين البيانات ومعالجتها بناءً على الغرض منها كما هو موضح بالشكل (10)، وتنقسم إلى نوعين رئيسيين:

الشكل (10) : التجريدات في هنـدسة البيـانات

5.8.1 نموذج متطور:

أنظمة تجمع بين ميزات بحيرات ومستودعات البيانات مثل Lakehouse Architecture.

الهيكل في البيانات (Schema Management)

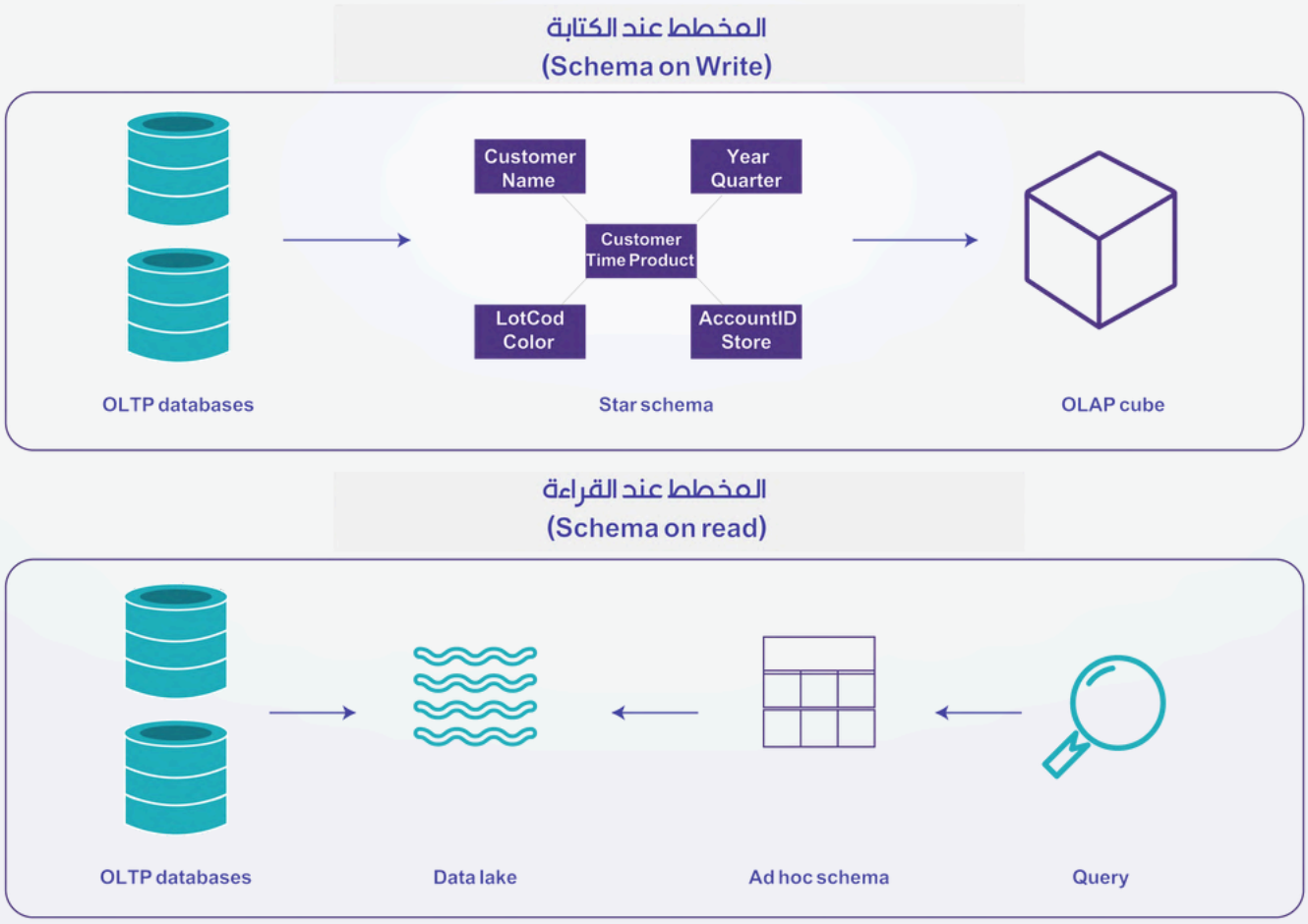
الهيكل عند الكتابة (SCHEMA ON WRITE):

يتم تحديد بنية البيانات عند إدخالها.

الهيكل عند القراءة (SCHEMA ON READ):

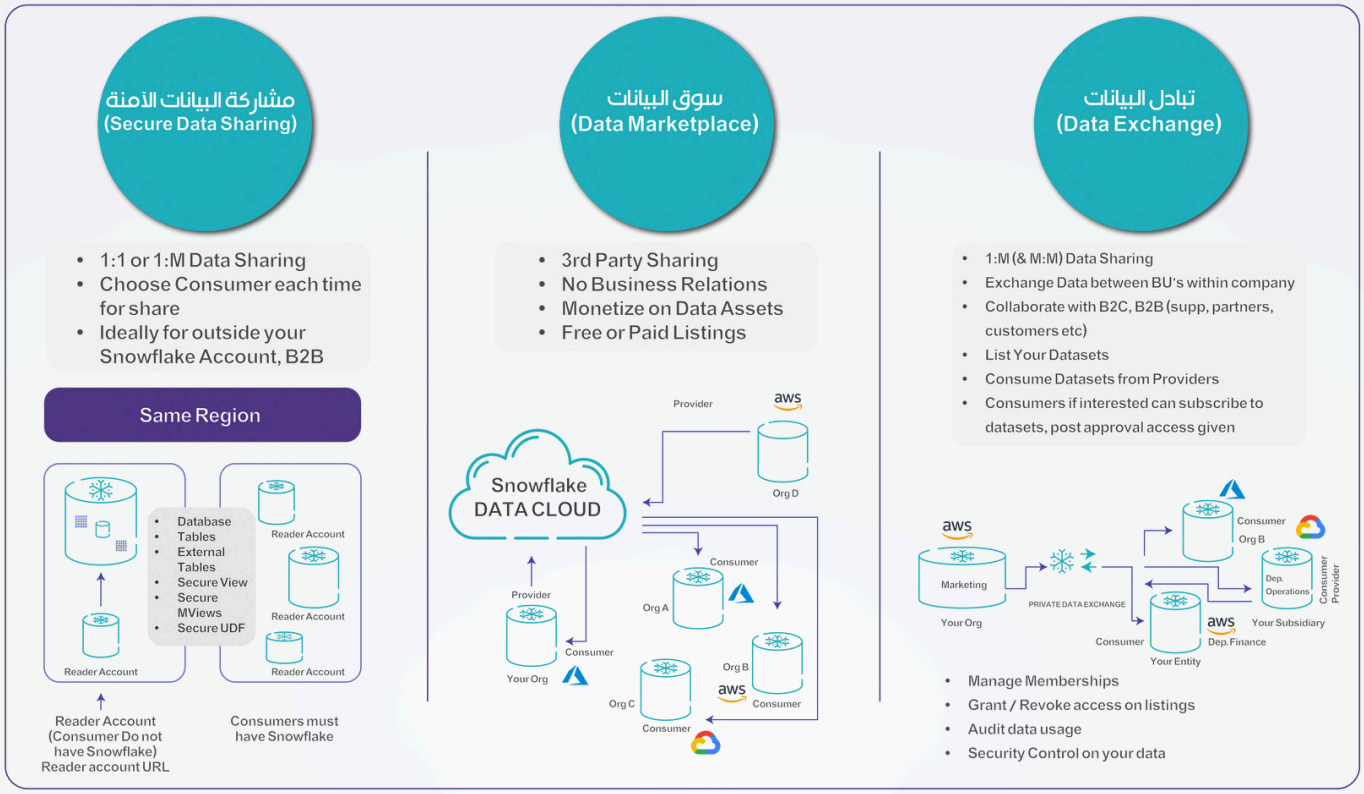
تُخزن البيانات كما هي، ويتم تحديد البنية عند الاستعلام عنها.

الشكل (11) : الهيكـــل في البيـــانات



2.8.5 مشاركة البيانات (Data Sharing) :

الشكل (12) : يوضح طريقة مشاركة البيانات



تقنيات حديثة في مشاركة البيانات مثل Data Sharing in Snowflake و Google BigQuery Data Exchange.

فصل الحوسبة عن التخزين (Compute & Storage Separation)

توفر مرونة أكبر وتخفيضًا للتكاليف عند فصل التخزين عن الحوسبة.

أمثلة على أنظمة تعتمد هذه التقنية:

snowflake Google Big Query

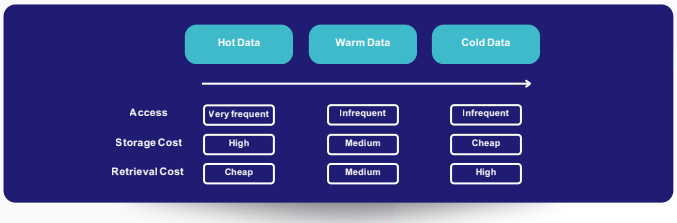
5.9 التخزين وإدارة البيانات في هندسة البيانات

أنواع البيانات حسب الوصول إليها:

البيانات الساخنة (HOT DATA):
يتم الوصول إليها بشكل متكرر جدًا.

البيانات الدافئة (WARM DATA):
يتم الوصول إليها بشكل شبه منتظم.

البيانات الباردة (COLD DATA):
يتم الوصول إليها نادرًا.



5.9.1 الركائز الاستراتيجية لحوكمة وتخزين البيانات

الامتثال واللوائح:
بعض القوانين مثل HIPAA و PCI تتطلب الاحتفاظ بالبيانات لفترة معينة.

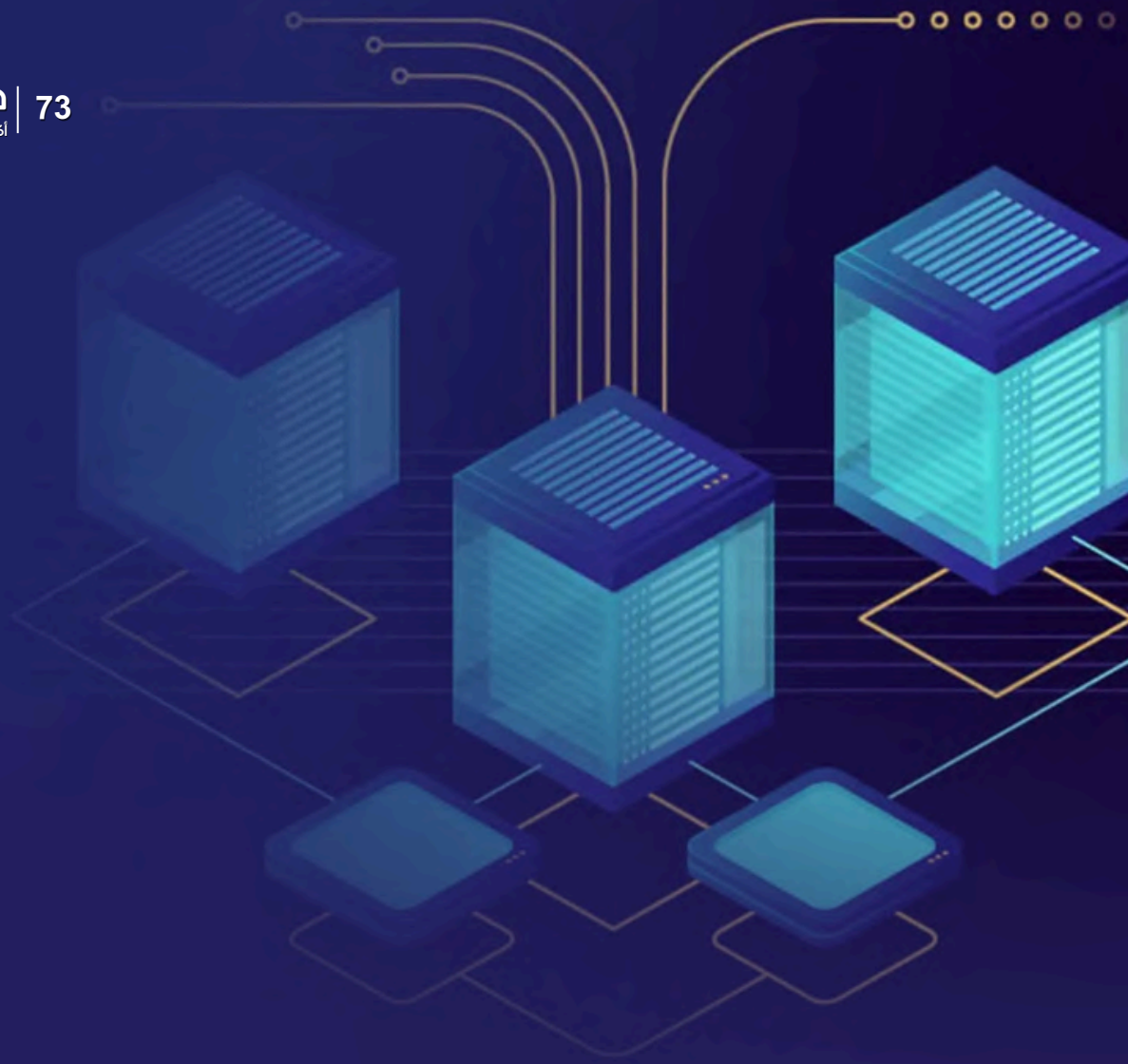
إدارة البيانات وتنظيمها:
فهرسة البيانات وإدارة البيانات الوصفية تجعل من السهل العثور على البيانات المناسبة بسرعة.

تكلفة تخزين البيانات:
التخزين هو استثمار، ويجب موازنة تكلفة الاحتفاظ بالبيانات مع فائدتها الفعلية.

خصوصية البيانات والامتثال للوائح:
يجب أن يكون النظام قادرًا على حذف البيانات عند الطلب أو إخفاء بيانات حساسة.

الأمان وإدارة الوصول:
الأمان يجب أن يتبع مبدأ أقل الامتيازات، بحيث لا يُمنح الوصول إلى البيانات إلا عند الضرورة.

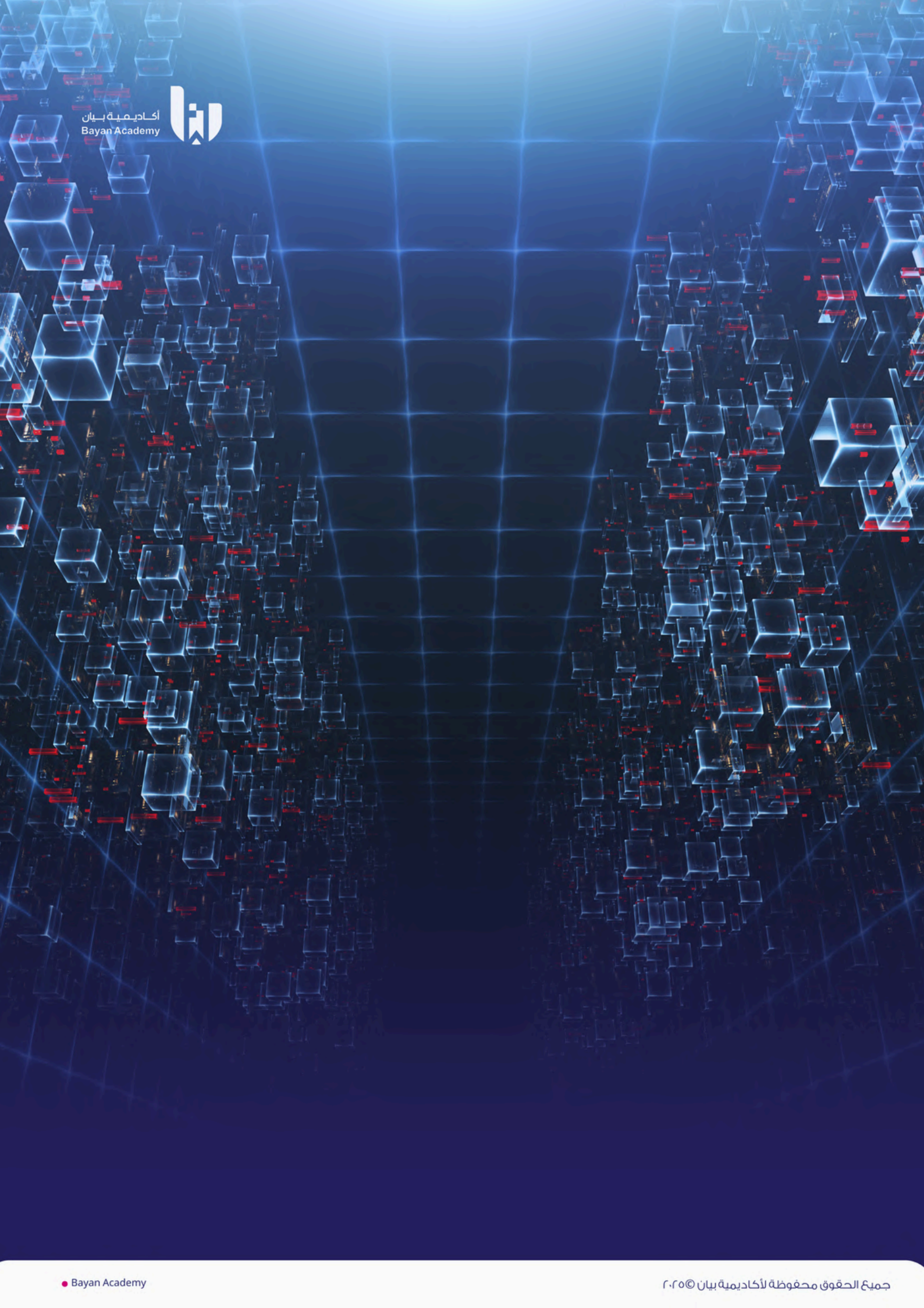
DataOps ودوره في التخزين:
DataOps يدمج بين إدارة البيانات ومراقبة أدائها لضمان الكفاءة والجودة.

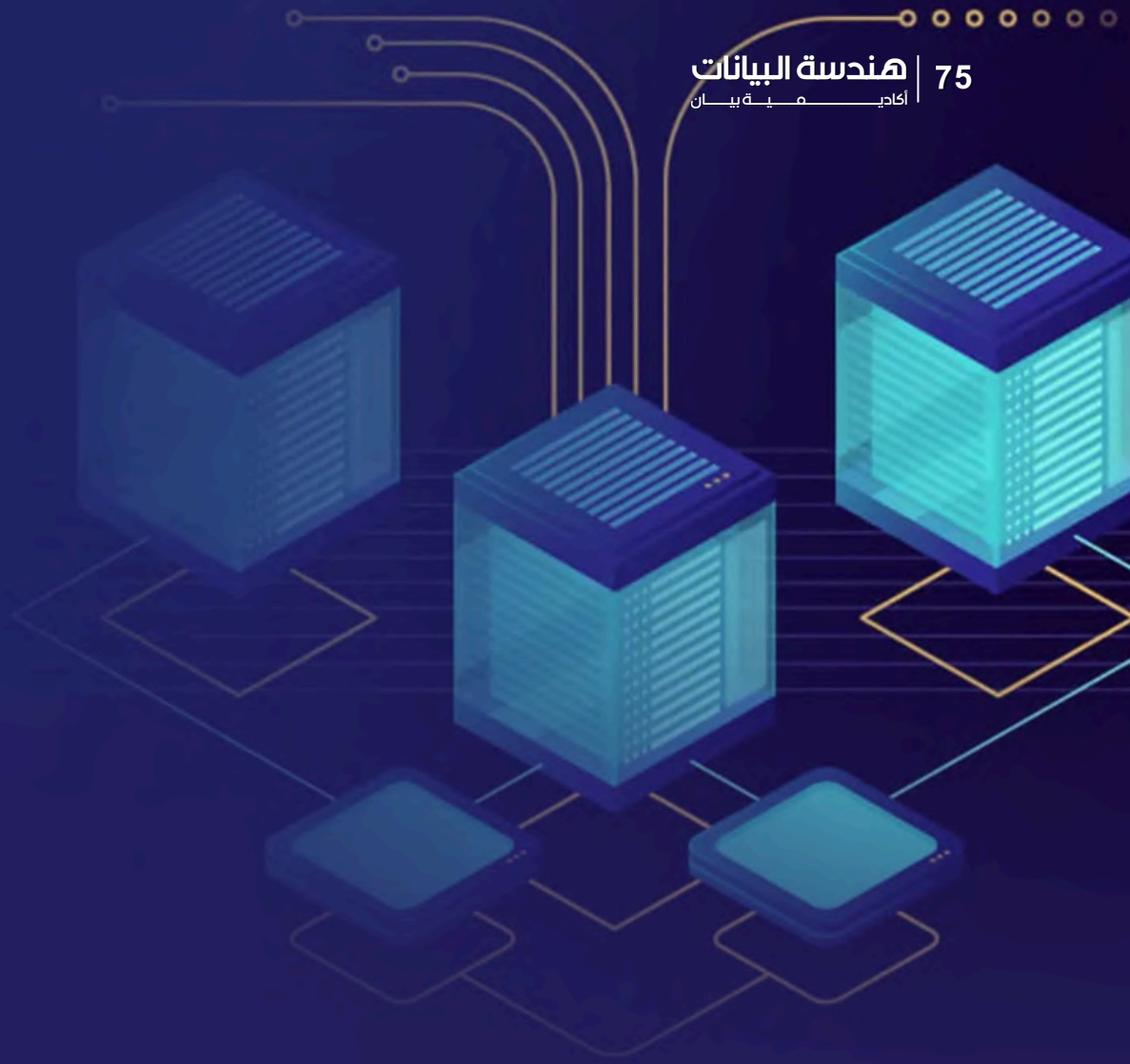


في ختام الفصل الخامس

تخزين البيانات

اختيار استراتيجية التخزين المناسبة يعتمد على احتياجات الشركة، الميزانية، ومستوى الأمان المطلوب. المؤسسات الكبيرة غالبًا ما تتبنى حلولاً هجينة تجمع بين أكثر من استراتيجية لتحقيق أفضل توازن بين الأداء والتكلفة.إذا كنت تفكر في تنفيذ استراتيجية تخزين جديدة، فمن المهم تحليل متطلباتك الحالية والمستقبلية لضمان اختيار الحل الأمثل.





06
استيعاب البيانات

Data ingestion

الفصل السادس

استيعاب البيانات

تعلمنا عن الأنظمة المصدرية المختلفة التي ستواجهها كمهندس بيانات، وعن طرق تخزين البيانات. الآن، دعونا نركز على الأنماط والخيارات المتعلقة باستيعاب البيانات من هذه الأنظمة. في هذا الفصل، سنناقش استيعاب البيانات، أهم الاعتبارات الهندسية في هذه المرحلة، الأنماط الرئيسية مثل الاستيعاب الدفعي (Batch) والتدفق المستمر (Streaming)، التقنيات المستخدمة، وكيفية تأثير العوامل الأساسية في عملية الاستيعاب.

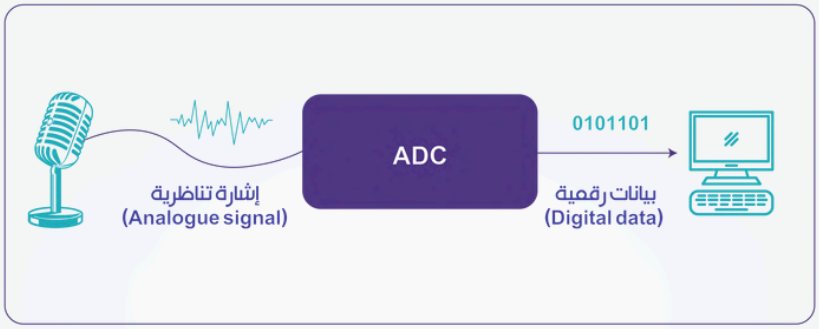
6.1 ما هو استيعاب البيانات؟

هو عملية نقل البيانات من مصادرها الأصلية إلى أنظمة التخزين أو التحليل، مثل مستودعات البيانات أو بحيرات البيانات. في سياق هندسة البيانات، يُعد استيعاب البيانات خطوة أساسية في بداية سير عمل البيانات، حيث تُجمع البيانات الخام تمهيدًا لتحليلها أو تحويلها لاحقًا.

مثال:

عند استخدامك لتطبيق بنكي على الهاتف المحمول، يتم تحديث رصيد حسابك تلقائيًا بعد كل معاملة مالية. هذه البيانات تنتقل من أنظمة البنك المصدرية إلى قاعدة بيانات مركزية، حيث تُخزن وتُعالج ليتم عرضها لك عبر التطبيق.

الشكل (13):



6.2 الفرق بين استيعاب البيانات (Data Ingestion) وتكامل البيانات (Data Integration)

من الضروري التمييز بين مفهومي استيعاب البيانات (Data Ingestion) وتكامل البيانات (Data Integration)، رغم ارتباطهما في سياق إدارة البيانات.

استيعاب البيانات (DATA INGESTION):

عملية نقل البيانات من مصادرها (مثل قواعد البيانات، واجهات برمجة التطبيقات، أو ملفات (CSV) إلى وجهات التخزين مثل مستودعات البيانات.

تكامل البيانات (DATA INTEGRATION):

عملية دمج البيانات من مصادر متعددة في نموذج موحد، بهدف الحصول على رؤية شاملة وموحدة يمكن تحليلها.

مثال تطبيقي:

في شركة تجزئة تمتلك فروعًا متعددة، يتم أولاً استيعاب بيانات المبيعات من كل فرع إلى مخزن بيانات مركزي. لاحقًا، تُدمج هذه البيانات مع بيانات المخزون وسلاسل التوريد لتوليد تقارير تحليلية متكاملة عن أداء الأعمال.

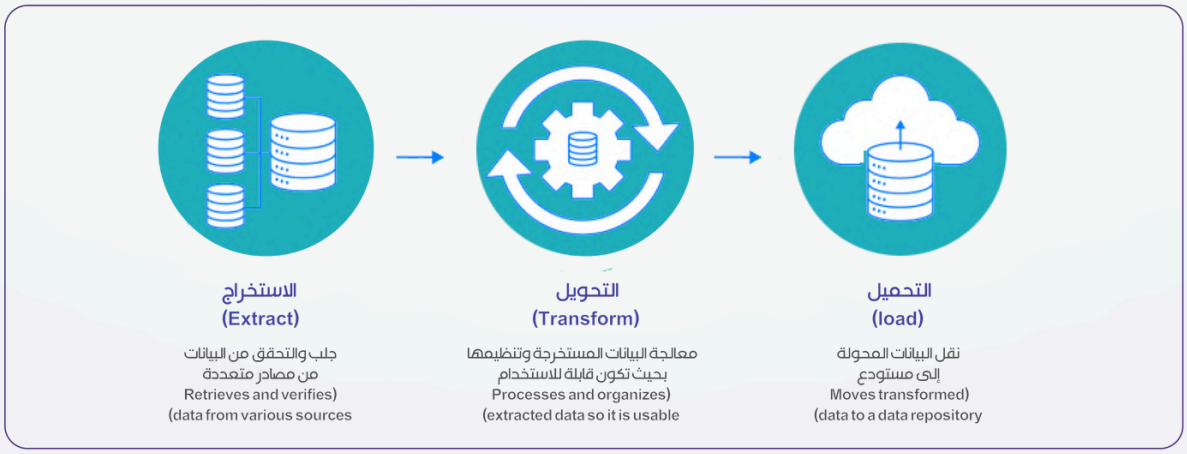
6.3 تعريف سير عمل البيانات (Data Pipeline)

هو سلسلة من الخطوات والعمليات التي تنتقل من خلالها البيانات، بدءًا من مصادرها وحتى أنظمة التخزين أو التحليل. تبدأ هذه السلسلة غالبًا بعملية الاستيعاب، تليها خطوات مثل التحويل، التنظيف، ثم التحميل النهائي كما هو موضح بالشكل(14).

من أشهر أنماط سير العمل

(ETL) (استخراج - تحويل - تحميل)	(ELT)(استخراج - تحميل - تحويل)	(REVERSE ETL)(العكس)
تُستخدم البيانات بعد معالجتها وتنظيفها.	تُخزن البيانات أولاً ثم تُحول في مرحلة لاحقة.	تُرسل البيانات من مستودعات التحليل إلى أنظمة التشغيل مثل أدوات التسويق أو أنظمة إدارة العملاء.

الشكل (14) : عملية (الاستخراج , التحويل والتحميل)



مثال تطبيقي:

في منصات مثل (Netflix) ، يتم استيعاب بيانات المستخدمين (مثل مدة المشاهدة، تفضيلات المحتوى) في الوقت الحقيقي. تُحلل هذه البيانات لاحقًا لتحديث خوارزميات التوصية، مما يُحسّن تجربة المستخدم بشكل مستمر.

6.4 المتطلبات الهندسية الرئيسية لمرحلة الاستيعاب

عند تصميم أو بناء نظام استيعاب بيانات، يجب طرح بعض الأسئلة الأساسية مثل:

01
ما هو الاستخدام المتوقع للبيانات المستوعبة؟

02
هل يمكن إعادة استخدام هذه البيانات لتجنب استيعاب نسخ متعددة منها؟

03
أين سيتم تخزين البيانات؟

04
كم مرة يجب تحديث البيانات من المصدر؟

مثال تطبيقي:

شركة لوجستية تحتاج إلى استيعاب بيانات التوصيل في الوقت الحقيقي لتتبع الشحنات. عند تصميم النظام، يجب أن تضمن أن البيانات دقيقة ومحدثة لتوفير تحديثات لحظية للعملاء.

6.5 الفرق بين البيانات المحدودة وغير المحدودة

الشكل (15): الفرق بين البيانات المحدودة و غير المحدودة

الفرق بين البيانات المحدودة وغير المحدودة

البيانات غير المحدودة (UNBOUNDED DATA)

هي بيانات تُنتج بشكل مستمر وبدون نقطة نهاية واضحة. تُستخدم غالبًا في أنظمة البث والمعالجة الفورية، حيث تتطلب تقنيات معالجة في الوقت الحقيقي (stream processing).

أمثلة:
- تغريدات على منصة X (تويتر سابقًا)
- سجلات المعاملات البنكية الفورية
- بيانات أجهزة إنترنت الأشياء (IoT)

البيانات المحدودة (BOUNDED DATA)

هي بيانات تُجمع خلال فترة زمنية محددة أو من مجموعة مغلقة، ويمكن تحميلها وتحليلها دفعة واحدة. تُستخدم غالبًا في عمليات التحليل الدفعي (batch processing).

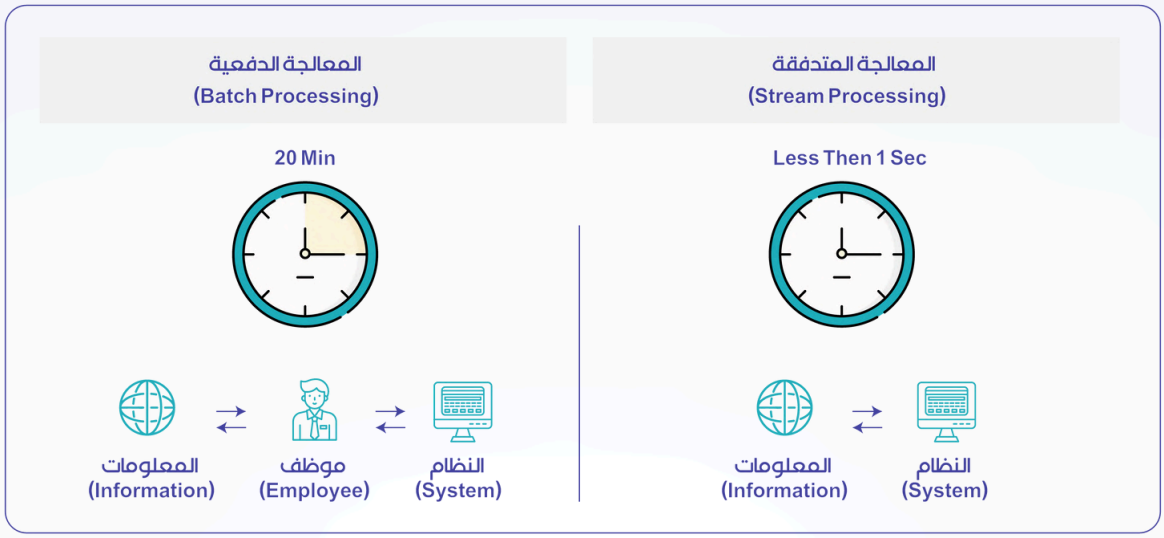
أمثلة:
- نتائج امتحانات الطلاب بعد نهاية الفصل الدراسي
- تقارير المبيعات اليومية أو الأسبوعية
- ملفات(CSV) المحفوظة من نظام معين

الخاصية	البيانات المحدودة	البيانات غير المحدودة
طريقة المعالجة	Batch	Streaming
نقطة النهاية	معروفة	غير معروفة (مستمرة)
حجم البيانات	يمكن احتواؤه	ينمو باستمرار
أمثلة	تقارير، نتائج، ملفات	أحداث زمنية، سجلات مباشرة

هذا التصنيف مهم جدًا عند اختيار بنية المعالجة المناسبة مثل (Spark Apache Kafka Streaming) للبيانات غير المحدودة، و(Apache Hive أو Snowflake) للبيانات المحدودة.

6.6 تدفق البيانات - تكرارية الاستيعاب (Frequency of Ingestion)

تشير تكرارية الاستيعاب إلى عدد المرات التي يتم فيها إدخال البيانات من الأنظمة المصدرية إلى أنظمة التخزين أو التحليل. يُحدد هذا النمط بناءً على طبيعة النظام والاحتياجات التحليلية، وتنقسم أنماط الاستيعاب الشائعة إلى:



01 الاستيعاب الدفعي (BATCH INGESTION)

يتم فيه جمع البيانات ومعالجتها على دفعات دورية (مثل كل ساعة، يوم، أو أسبوع). مناسب للأنظمة التي لا تتطلب تحديثات فورية.

مثال:
تقارير المبيعات اليومية في المتاجر، التي تُجمع في نهاية كل يوم.

02 الاستيعاب عبر دفعات صغيرة (MICRO-BATCH)

تقنية وسيطة بين الاستيعاب الدفعي واللحظي، حيث تُقسّم البيانات إلى دفعات صغيرة (مثلاً كل دقيقة أو أقل)، وتُعالج شبه لحظيًا.

مثال:
تحليلات زوار موقع إلكتروني يتم تحديثها كل 30 ثانية لتحديد سلوك التصفح.

03 الاستيعاب في الوقت الحقيقي (REAL-TIME INGESTION)

يتم فيه التقاط البيانات ومعالجتها لحظة حدوثها تقريبًا (زمن تأخير منخفض جدًا). يُستخدم في الأنظمة التي تتطلب قرارات فورية.

مثال:
أنظمة اكتشاف الاحتيال في البنوك التي تراقب كل معاملة مالية مباشرة لاتخاذ إجراء فوري في حال الاشتباه.

النوع	زمن التأخير (Latency)	الاستخدام النموذجي	التعقيد التقني
Batch	عالي (دقائق – ساعات)	التقارير الدورية	منخفض
Micro-Batch	متوسط (ثوانٍ – دقائق)	تحليلات شبه لحظية	متوسط
Real-Time	منخفض جدًا (ميلي ثانية – ثوانٍ)	أنظمة حساسة للوقت كالاحتيال	مرتفع

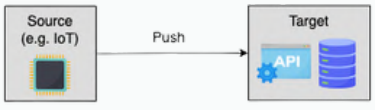
6.7 أنماط جلب البيانات(Data Access Patterns): الدفع مقابل السحب مقابل الاستطلاع

في أنظمة استيعاب البيانات، تُستخدم عدة أنماط لجلب البيانات من المصدر إلى الوجهة، ويُحدد الاختيار بناءً على طبيعة النظام، زمن التأخير المطلوب، وحجم البيانات.

01 الدفع (PUSH)

في هذا النمط، يقوم المصدر بإرسال البيانات تلقائيًا إلى الوجهة عند حدوث حدث معين. يُستخدم عادةً في التطبيقات ذات الزمن الحقيقي أو التي تتطلب استجابة سريعة.

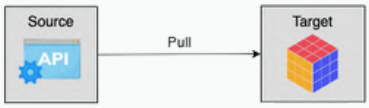
مثال:
أنظمة الإشعارات الفورية، حيث يُرسل الخادم إشعارًا تلقائيًا إلى الهاتف عند وصول رسالة أو حدث جديد.



02 السحب (PULL)

في هذا النمط، تقوم الوجهة بطلب البيانات من المصدر عند الحاجة. هذا النمط مناسب للأنظمة التي لا تحتاج إلى تحديثات مستمرة أو في الوقت الحقيقي.

مثال:
لوحة تحكم تحليلية تُحدث البيانات فقط عند دخول المستخدم إليها.



03 الاستطلاع (POLLING)

هو شكل خاص من السحب، حيث تقوم الوجهة بإجراء فحص دوري للمصدر (مثل كل دقيقة أو كل ساعة) للتحقق من توفر بيانات جديدة. يُستخدم عند غياب آلية إشعارات مباشرة من المصدر.

مثال:
تطبيق الطقس الذي يجلب تحديثات الحالة الجوية من الخادم كل ساعة تلقائيًا.

النمط	من يُبادر بالنقل؟	زمن التأخير	الاستهلاك من الموارد	الاستخدام النموذجي
(Push) الدفع	المصدر	منخفض جدًا	منخفض	إشعارات، بث مباشر، أحداث
(Pull) السحب	الوجهة	حسب الطلب	منخفض	تقارير عند الطلب
(Polling) الاستطلاع	الوجهة (بشكل دوري)	متوسط إلى مرتفع	متوسط إلى مرتفع	تطبيقات غير لحظية، طقس، تحديثات دورية

6.8 أنظمة الرسائل وتدفق الأحداث

أنظمة الرسائل وتدفق الأحداث (Message Brokers & Event Streaming)

في الأنظمة الحديثة التي تتطلب تدفقًا سريعًا ومستمرًا للبيانات، تُستخدم أنظمة الرسائل وأنظمة بث الأحداث كوسيط بين مُنتِجي البيانات (مثل التطبيقات أو أجهزة الاستشعار) ومستهلكيها (مثل قواعد البيانات أو محركات التحليل).

ما هي أنظمة الرسائل؟

هي برمجيات وسيطة تستقبل الرسائل من مصدر ما وتعيد توجيهها إلى جهة مستهدفة بشكل آمن وفعال، سواء كانت الرسائل فورية أو مجدولة أو متكررة.

ما هو تدفق الأحداث؟

هو نمط من أنماط استيعاب البيانات يعتمد على معالجة البيانات فور توليدها، وهي مهمة جدًا في التطبيقات الحساسة للزمن مثل تحليل السلوك اللحظي أو المراقبة.

النظام	الاستخدام الأساسي	نقاط القوة
Apache Kafka	معالجة تدفقات ضخمة من البيانات الزمنية	سرعة وأداء عالي، دعم للتخزين المؤقت
RabbitMQ	إرسال رسائل بين الأنظمة	خفيف، مرن، يدعم أنماط متعددة للرسائل
Google Pub/Sub	خدمات سحابية لبث الرسائل	تكامل ممتاز مع (Google Cloud)

مثال تطبيقي:

في السيارات ذاتية القيادة، تُنتج المستشعرات كميات كبيرة من البيانات (مثل المسافة، السرعة، الكاميرات)، وتُرسل هذه البيانات في الوقت الحقيقي عبر (Kafka) إلى وحدات المعالجة لاتخاذ قرارات القيادة الفورية، مثل الكبح أو تغيير المسار.

استخدامات أخرى:

المعاملات البنكية:

مراقبة المعاملات في الوقت الحقيقي للكشف عن الاحتيال.

منصات الفيديو:

تحليل ما يشاهده المستخدم لحظيًا لتوصية محتوى مشابه.

أنظمة التجارة الإلكترونية:

تتبع سلوك الزبائن أثناء تصفحهم لتحديث العروض.

6.9 مشاركة البيانات والتعاون مع أصحاب المصلحة

في بيئة هندسة البيانات، لا تقتصر المهام على الجانب التقني فحسب، بل تشمل أيضًا التفاعل والتنسيق مع مختلف الفرق التي تُنتج وتستهلك البيانات. نجاح سير عمل البيانات (Data Pipeline) يعتمد بشكل كبير على جودة هذا التعاون.

من هم أصحاب المصلحة (Stakeholders)؟

الفئة الدور في دورة البيانات المطورون يبنون التطبيقات التي تُنتج البيانات علماء البيانات يحللون البيانات لتوليد النماذج والتوصيات المحللون يقدمون تقارير تعتمد على البيانات فرق الأعمال يستخدمون النتائج لاتخاذ قرارات استراتيجية

أهمية التعاون:

فهم بنية البيانات:
يساعد المطورون في شرح مصدر البيانات وتنسيقاتها.

ضمان الجودة:
يتعاون المهندسون مع المحللين لضمان نظافة البيانات وتناسقها.

تحديد الاحتياجات التحليلية:
يتبادل علماء البيانات مع المهندسين متطلبات النماذج التنبؤية.

توجيه التطوير:
الفرق غير التقنية مثل التسويق أو المبيعات تُوجه جهود البيانات لتحقيق أهداف محددة.

مثال تطبيقي:

في شركة تجارة إلكترونية، يجتمع فريق البيانات مع فريق التسويق لفهم احتياجات العروض الترويجية. يقوم مهندسو البيانات بتوفير بيانات سلوك المستخدم (مثل: المنتجات التي شوهدت أو أضيفت للسلة) بطريقة سهلة الاستخدام، ليقوم فريق التسويق بتخصيص العروض بناءً على هذه الأنماط.

أفضل الممارسات:

- اجتماعات دورية بين الفرق
- توثيق واضح لتدفقات البيانات ومصادرها
- واجهات برمجية (APIs) أو لوحات بيانات تفاعلية للمشاركة

6.10 الأمان وإدارة البيانات

من الضروري ضمان حماية البيانات أثناء نقلها وتخزينها، خاصةً في القطاعات الحساسة مثل الصحة والمالية.

مثال تطبيقي:
في المستشفيات، يتم تشفير بيانات المرضى عند نقلها بين الأنظمة لضمان الامتثال لمعايير الخصوصية مثل (HIPAA).

أفضل الممارسات:

- التشفير (Encryption): تأمين البيانات أثناء النقل والتخزين (مثل TLS، HTTPS، وAES).
- الشبكات الخاصة (VPNs): لمنع الوصول غير المصرح به.
- إدارة تغييرات المخطط (Schema Evolution): التخطيط لكيفية التعامل مع تغييرات بنية البيانات بمرونة دون كسر النظام.

6.11 عمليات البيانات (DataOps)

يشير مفهوم(DataOps) إلى المنهجيات والأدوات التي تضمن تشغيل خطوط البيانات بشكل موثوق وقابل للتوسع، تمامًا مثل(DevOps)في تطوير البرمجيات.

مثال تطبيقي:
في البورصات المالية، فإن أي انقطاع في تدفق بيانات الأسهم قد يؤدي إلى خسائر بملايين الدولارات في ثوانٍ.

الأهمية:

- تقليل زمن الأعطال
- تسريع إطلاق الميزات التحليلية
- تحسين جودة البيانات التشغيلية

6.12 اختبارات جودة البيانات (Data Quality Testing)

تعد مراقبة جودة البيانات أمرًا أساسيًا لضمان دقة التحليلات والقرارات المستندة إلى البيانات.

مثال تطبيقي:

في تطبيقات الملاحة مثل (Google Maps)، تتم مراجعة بيانات المواقع باستمرار لضمان تحديد دقيق للمسارات.

تشمل الاختبارات:

التحقق من القيم المفقودة أو الشاذة

التأكد من التوافق مع المخطط

اختبار التكرار والدقة والاتساق

6.13 التنسيق والتنظيم (Orchestration)

التنسيق يعني إدارة ترتيب وجدولة وتنفيذ خطوات خط أنابيب البيانات بشكل منظم وآمن.

مثال تطبيقي:

في أنظمة الحجز الفندقي، يتم تنسيق بيانات الحجز من منصات متعددة لتجنب الحجز المزدوج والتضارب.

أدوات شائعة:

Apache Airflow

Dagster

Prefect

في ختام الفصل السادس

استيعاب البيانات

تُعد مرحلة استيعاب البيانات الأساس الذي يُبنى عليه نجاح أي بنية تحتية لتحليل البيانات. من خلال استيعاب البيانات بشكل آمن، مرن، ومخطط جيدًا، تضمن المؤسسات: سهولة التوسع, سلامة البيانات و جودة التحليلات النهائية. إن فهم تكرار الاستيعاب، أنماط الجلب، معالجة البيانات غير المحدودة، وضمان الأمان والجودة والتنسيق هو ما يمكّن المهندس من بناء أنظمة قابلة للصمود، ومستدامة، ومؤتمتة.

07

الاستعلامات ، النمذجة و تدفق البيانات

Queries, Modeling, and Transformation

الفصل السابع

الاستعلام النمذجة وتدفق البيانات

يركز هذا الفصل على ثلاث مفاهيم رئيسية في هندسة البيانات: الاستعلامات، نمذجة البيانات، وتدفق البيانات. يبدأ بتوضيح دور الاستعلامات في استخراج البيانات وتحليلها، ثم ينتقل إلى شرح نمذجة البيانات وأهميتها في تنظيم المعلومات بما يعكس منطق العمل. كما يستعرض التحديات المتعلقة بنمذجة البيانات المتدفقة، ويعرض أبرز الممارسات التي تساهم في تحسين الأداء وضمان جودة البيانات في الأنظمة الحديثة.

7.1 الاستعلامات في هندسة البيانات

الاستعلامات (Data Queries)هي عملية أساسية في هندسة البيانات وعلم البيانات، حيث تُستخدم لاسترجاع (قراءة) البيانات من قواعد البيانات وتحليلها. ومن خلالها، يمكن للمستخدم تحديد بيانات معينة من قاعدة بيانات، بناءً على شروط محددة.

ما هو الاستعلام؟

الاستعلام هي عملية إرسال طلب إلى قاعدة البيانات لاسترجاع أو تعديل البيانات المخزنة. وتُعد لغة الاستعلام المهيكلة (Structured Query Language) SQLهي اللغة الأكثر استخدامًا لتنفيذ الاستعلامات, وتحتوي على انواع الاوامر كالتالي:

الفئة	الوصف	أمثلة
DDL (Data Definition Language)	تُستخدم لتعريف هيكل الجداول والعلاقات	CREATE TABLE, ALTER TABLE, DROP TABLE
DML (Data Manipulation Language)	تُستخدم لإضافة وتعديل وحذف البيانات	SELECT, INSERT, UPDATE, DELETE
DCL (Data Control Language)	للتحكم في صلاحيات المستخدمين	GRANT, REVOKE
TCL (Transaction Control Language)	لإدارة العمليات داخل المعاملات	COMMIT, ROLLBACK

7.1.1 لغة البيانات التعريفية (DDL - Data Definition Language)

7.1.2 لغة التلاعب بالبيانات (DML - Data Manipulation Language)

تُستخدم لإضافة وتعديل وحذف البيانات داخل الكائنات في قاعدة البيانات. تعد الأوامر التالية من الأوامر الأساسية في (DML):

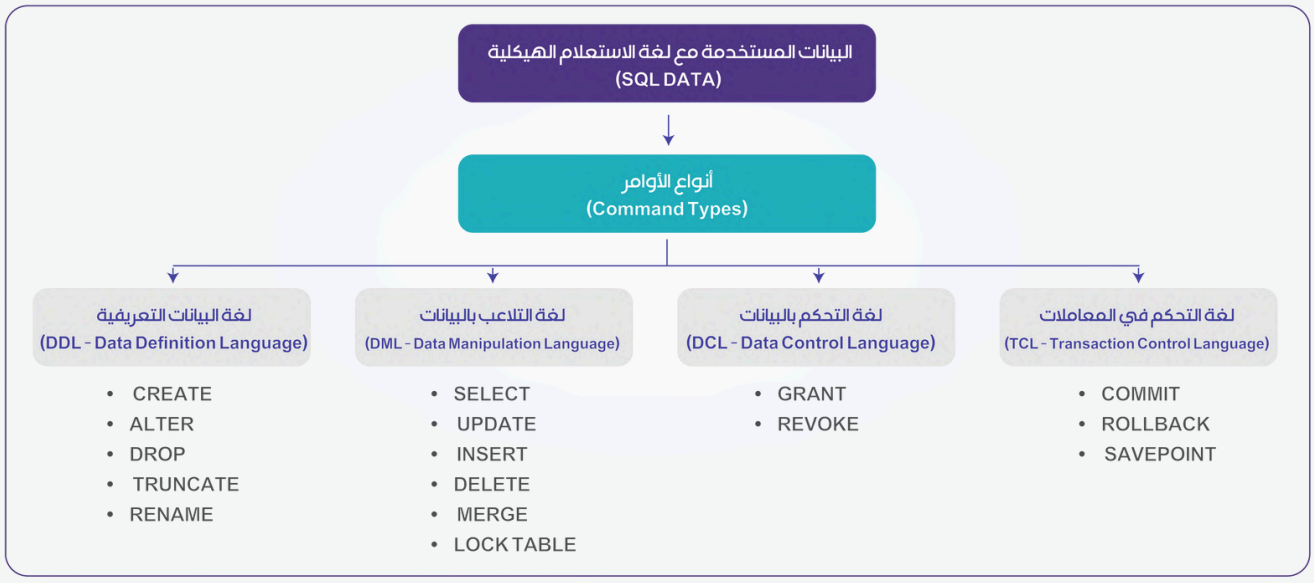
7.1.3 لغة التحكم بالبيانات (DCL - Data Control Language)

تُستخدم لإدارة الوصول إلى قاعدة البيانات. الأوامر المستخدمة في (DCL)تشمل:

7.1.4 لغة التحكم في المعاملات (TCL - Transaction Control Language)

تُستخدم للتحكم في تفاصيل المعاملات في قاعدة البيانات كما هو موضح بالشكل(16). تشمل الأوامر المستخدمة في TCL:

الشكل (16): لغـة التحكـم في المعـاملات



7.1.5 كتابة الاستعلامات الأساسية:

1 **SELECT**
للحصول على البيانات.

2 **WHERE**
لتحديد شرط معين.

3 **GROUP BY**
لتجميع البيانات.

4 **HAVING**
لتحديد شرط على النتائج المُجمّعة.

5 **ORDER BY**
لترتيب النتائج.

7.1.6 أنواع الربط (JOIN):

01 **INNER JOIN**
يجلب البيانات المشتركة بين جدولين.

02 **LEFT JOIN**
يجلب كل البيانات من الجدول الأيسر وما يُطابقه من الجدول الأيمن.

03 **RIGHT JOIN**
يجلب كل البيانات من الجدول الأيمن وما يُطابقه من الجدول الأيسر.

04 **FULL JOIN**
يجلب جميع البيانات سواء تطابقت أم لا.

7.1.7 تحسين أداء الاستعلامات:

7.1.8 فهم خطة التنفيذ (Execution Plan):

لماذا فهم خطة التنفيذ؟ لتحديد سبب بطء الاستعلامات وتحسين الأداء.

01

أداة مساعدة:

في (PostgreSQL) مثلاً:
(sql)
(EXPLAIN ANALYZE)
SELECT * FROM sales ؛
(WHERE amount > 500

02

النتيجة:

توضح إن كان يتم استخدام
(Seq Scan):
مسح تسلسلي بطيء أو
(Index Scan): أداء أعلى.

03

نصيحة:

استعمل أدوات مثل
(Explain Visualizer)
لفهم الخطة بصريًا.

7.2 ما هو نموذج البيانات(Data Model)؟

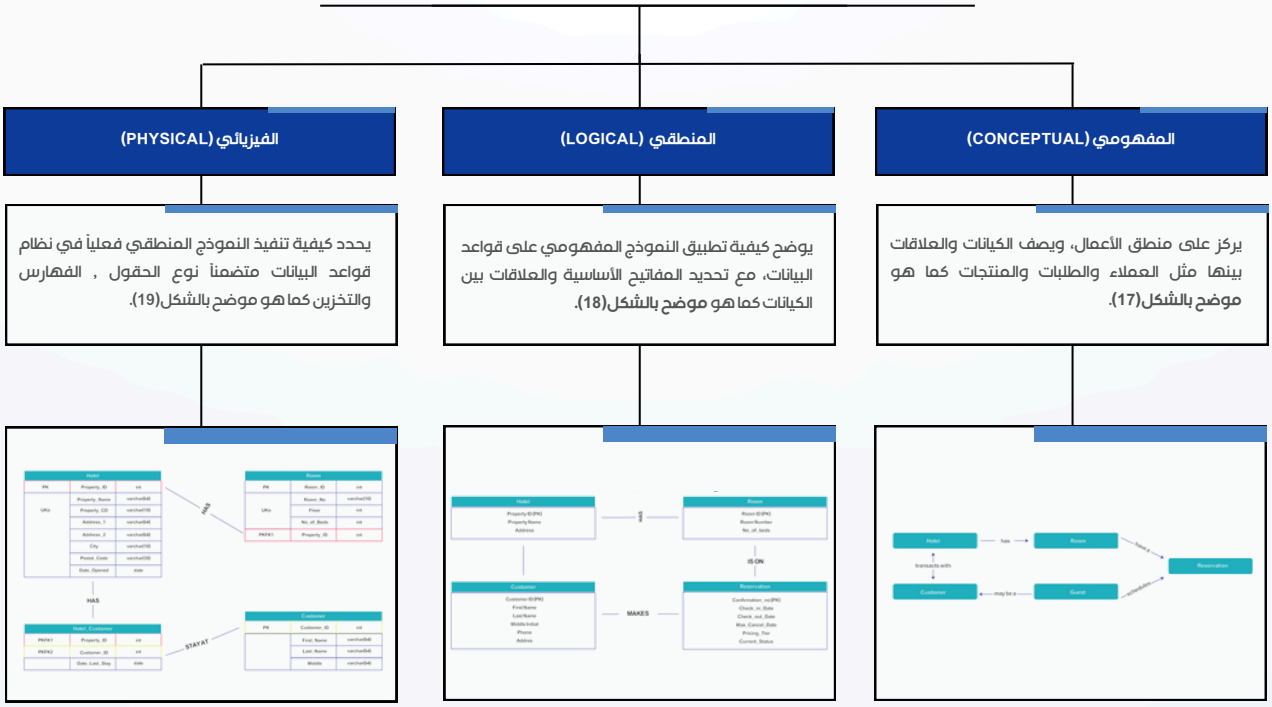
هي عملية تصميم تمثل منطق الأعمال وقواعدها بطريقة منظمة في طبقة البيانات. قد يظن البعض أنها مملة أو خاصة فقط بالمؤسسات الكبيرة، لكنها في الحقيقة ممارسة أساسية لضمان أن تكون البيانات مرتبة، متماسكة، وقابلة للاستخدام لاتخاذ قرارات فعالة.

قد يظن البعض أن النمذجة مخصصة للمؤسسات الكبيرة فقط أو أنها عملية مملة، لكن في الواقع هي من الممارسات الأساسية التي تضمن جودة البيانات وتنظيمها.

7.2.1 أهمية نمذجة البيانات في المؤسسات

النمذجة تُترجم التعريفات التجارية المختلفة إلى نموذج بيانات موحد وواضح. على سبيل المثال، قد تعني كلمة "عميل" أشياء مختلفة في أقسام متعددة داخل نفس المؤسسة، لذا من الضروري تعريفها بدقة لضمان تناسق التحليل والتقارير عبر جميع الأقسام.

أنواع نمذجة البيانات



شكل (17) علاقة الكيانات

شكل (18) نموذج البيانات المنطقي

شكل (19) نموذج البيانات الفيزيائية

7.2.2 أهمية إشراك أصحاب المصلحة في عملية النمذجة

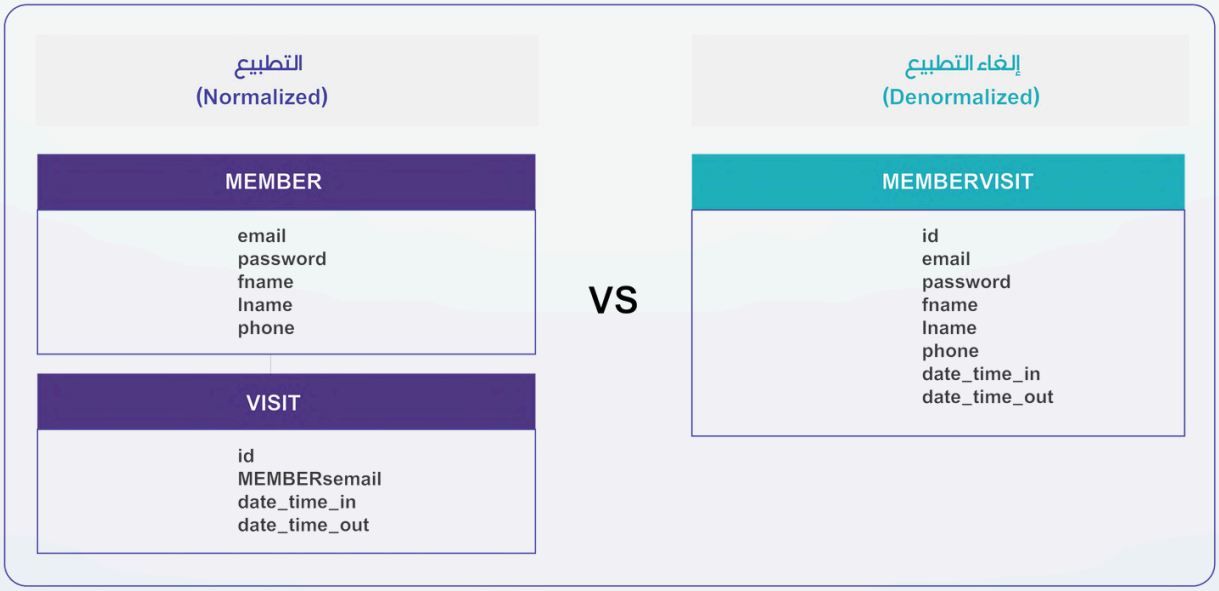
نجاح نمذجة البيانات يتطلب تعاونًا فعالاً مع أصحاب المصلحة منذ بداية المشروع. يجب على مهندسي البيانات فهم أهداف وتعريفات الأعمال لضمان تقديم نموذج بيانات عالي الجودة يدعم رؤى قابلة للتنفيذ.عملية النمذجة تفاعلية وتحتاج إلى مراجعات وتحديثات مستمرة لضمان توافق النموذج مع احتياجات العمل.

جزء مهم من نمذجة البيانات هو تنظيم البيانات بشكل منطقي داخل قواعد البيانات. وهنا يأتي مفهوم التطبيع، وهو تقنية تُستخدم لترتيب البيانات داخل الجداول لتقليل التكرار وتحسين تكامل العلاقات.

بعبارة أخرى، التطبيع هو أداة رئيسية ضمن نمذجة البيانات المنطقية تساعد في بناء نموذج بيانات سليم، يسهل صيانته ويعزز جودة البيانات.

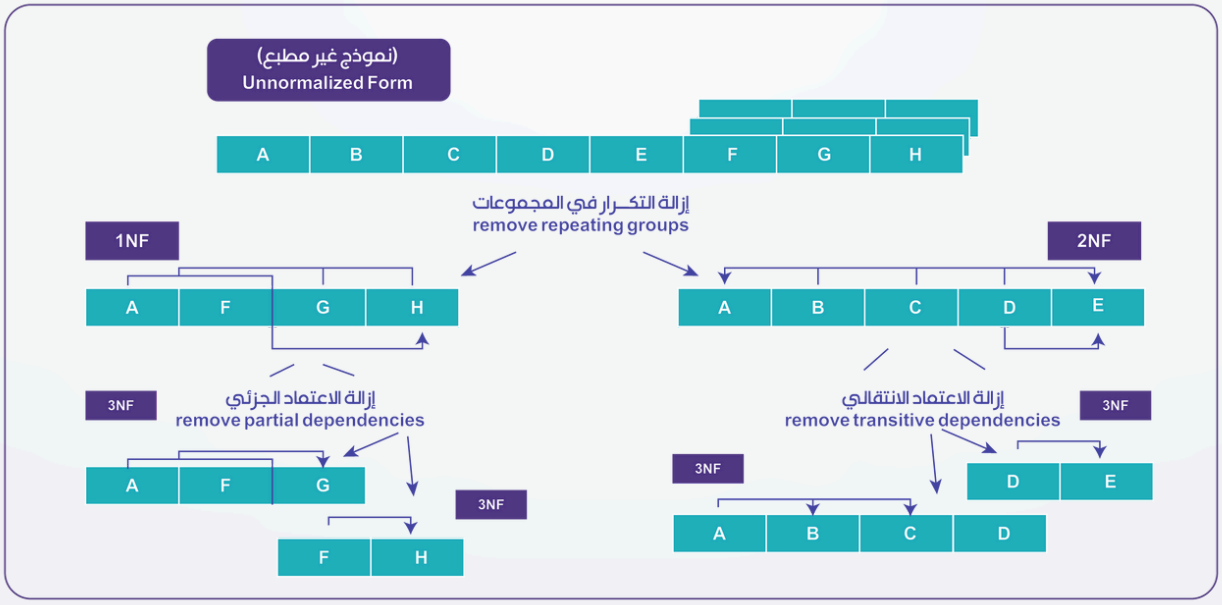
التطبيع هو عملية تنظيم البيانات في قواعد البيانات لمنع التكرار وتحسين تكامل العلاقات بين الجداول. قدمها إدغار كود في السبعينيات بهدف تقليل الأخطاء وزيادة كفاءة الصيانة **يوضح الشكل (20)** مقارنة بين التطبيع وإلغاء التطبيع.

الشكل (20): مقارنة بين التطبيع وإلغاء التطبيع



- أشكال التطبيع (Normalization):

الشكل (21) : أشكال التطبيع



7.2.3 الاعتماديات الانتقالية (Transitive Dependencies) وعلاقتها بالتطبيع:

- **ما هي الاعتماديات الانتقالية؟**

هي حالة تحدث عندما يعتمد حقل غير مفتاح على حقل غير مفتاح آخر، عبر علاقة غير مباشرة مع المفتاح الأساسي.

مثال:
العمود (A) هو المفتاح الأساسي.

العمود (B)يعتمد على(A).

العمود (C)يعتمد على (B)وليس على A مباشرة.

في هذه الحالة، نقول إن (C) تعتمد اعتمادًا انتقاليًا على (A) عبر (B).

- **علاقة الاعتماديات الانتقالية بالتطبيع:**

في (3NF)، الهدف هو إزالة الاعتماديات الانتقالية. أي يجب أن يعتمد كل حقل غير مفتاح على المفتاح الأساسي مباشرة فقط، وليس عبر حقل غير مفتاح آخر.

مثال عملي
في جدول("ORDERS"):

(OrderID) هو المفتاح الأساسي.(CustomerID) مرتبط بـ (OrderID).

(CustomerName) و (CustomerAddress) مرتبطان بـ (CustomerID) لكن مخزنين في جدول ("Orders").

هنا (CustomerName) و (CustomerAddress) تعتمد اعتمادًا انتقاليًا على (OrderID) عبر (CustomerID).

كيفية الإصلاح:

نقل (CustomerName) و (CustomerAddress) إلى جدول منفصل يسمى ("Customers")يحتوي على (CustomerID) كمفتاح أساسي، وترك جدول ("Orders") يحتوي فقط على (OrderID) و (CustomerID).

- **إزالة التطبيع (De-normalization):**

إزالة التطبيع تعني دمج جداول متعددة في جدول واحد بهدف تقليل عمليات الانضمام (Joins) وتحسين سرعة استعلامات القراءة، خصوصاً في بيئات التحليل مثل مستودعات البيانات.

استخدامات إزالة التطبيع	عيوب إزالة التطبيع
• تسريع استعلامات القراءة المعقدة. • تقليل تعقيد الاستعلامات الناتج عن الانضمامات المتعددة. • قبول زيادة التكرار لتحقيق أداء أعلى.	• زيادة استهلاك مساحة التخزين. • تعقيد تحديث البيانات بسبب التكرار. • احتمالية تناقض البيانات إذا لم تُدار التحديثات بدقة.

- **التطبيع الإضافي (6NF)**

هناك أشكال تطبيعية إضافية مثل (5NF، 4NF، و6NF)، تُستخدم في حالات متخصصة تهدف إلى تقليل التكرار أكثر وتحسين الأداء، لكنها نادرًا ما تُستخدم في النماذج التقليدية.

- **تقنيات نمذجة بيانات البيئات التحليلية (Batch Analytical Data)**

في بيئات بحيرات البيانات (Data Lakes) ومستودعات البيانات (Data Warehouses)، تُحول البيانات الخام إلى نماذج منظمة (صفوف وأعمدة) باستخدام عدة أساليب رئيسية:

Kimball	Inmon	Data Vault
يعتمد على نموذج النجمة (Star Schema) لتسهيل التحليل.	يعتمد على بناء مستودع بيانات موحد ومطبع (3NF).	يدمج بين الأسلوبين لمرونة أكبر.

7.2.4 مقاربة (Inmon) في نمذجة البيانات

موضوعي (Subject-Oriented):
يركز مستودع البيانات على مجالات معينة مثل المبيعات أو التسويق.

مُدمج (Integrated):
يتم دمج البيانات من مصادر متعددة.

غير متغير (Non-volatile):
تبقى البيانات ثابتة بعد تخزينها.

زمني (Time-variant):
يمكن الاستعلام عن البيانات عبر فترات زمنية مختلفة.

يتم نقل البيانات من النظام المصدر إلى مستودع البيانات بتنسيق(3NF) ، ثم تحويلها إلى مارتات بيانات حسب الحاجة للتحليل.

نموذج النجمة (Star Schema)

- **(Inmon) مقابل (Kimball)**

الخاصية	Inmon	Kimball
الهيكلية	مستودع بيانات موحد ومطبع (3NF)	نموذج النجمة وتخزين تصاعدي
التكرار	أقل	أكثر
الأداء	تكامل أكبر، أداء متوسط	أداء أسرع لاستعلامات التحليل
الاستخدام	مؤسسات تحتاج دقة عالية	مؤسسات تحتاج سرعة تحليل

7.2.5 الأبعاد البطيئة التغير (Slowly Changing Dimensions - SCD)

الأبعاد البطيئة التغير (Slowly Changing Dimensions - SCD) هناك ثلاثة أنواع رئيسية من الأبعاد البطيئة التغير:

النوع 1 (TYPE 1):

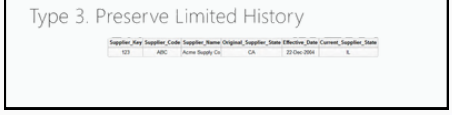
حيث يتم استبدال البيانات القديمة بالبيانات الجديدة، مما يعني أن التاريخ لا يُحفظ.

النوع 2 (TYPE 2):

حيث يتم إنشاء سجل جديد مع كل تغيير، مما يسمح بتتبع التاريخ.

النوع 3 (TYPE 3):

يتم تعديل السجل الحالي بإضافة حقول جديدة بدلاً من إنشاء سجل جديد.



7.3 تحديات وأفضل الممارسات في نمذجة البيانات المتدفقة وتحويلاتها (Streaming Data Modeling and Transformations):

في ظل توسع استخدام البيانات المتدفقة (Streaming Data) في المؤسسات الحديثة، خصوصًا في البيئات التي تتطلب تحديثات لحظية مثل تحليلات الوقت الحقيقي (Real-time Analytics) أو أنظمة المراقبة، تواجه فرق البيانات تحديات خاصة تختلف عن تلك التي نواجهها في النمذجة التقليدية للبيانات الدفعة (Batch Data).

التحديات الرئيسية في نمذجة البيانات المتدفقة:

01 التعامل مع تدفق البيانات المستمر والغير متوقع
- البيانات تأتي بشكل مستمر وغير منتظم، مما يصعب حفظها وتنظيمها بشكل تقليدي.
- اختلاف حجم البيانات وتنوعها يتطلب نماذج مرنة وقابلة للتوسع.

02 ضمان جودة وتكامل البيانات في الوقت الحقيقي
- صعوبة تطبيق قواعد التطبيع والتحقق من التكامل كما في النماذج التقليدية، بسبب السرعة العالية والتحديث المستمر.
- التعامل مع البيانات الناقصة أو غير الصحيحة أثناء تدفقها.

03 التعامل مع حالات البيانات البطيئة التغير (SCD) في الزمن الحقيقي
- تحديث أبعاد بطيئة التغير أثناء تدفق البيانات يتطلب استراتيجيات مختلفة عن التحديث الدوري في البيئات التقليدية.

04 إدارة النسخ المتكررة (DE-NORMALIZATION) بشكل محسوب
- قد يكون من الضروري تخزين بيانات متكررة لتحسين سرعة القراءة، لكن هذا يزيد من تعقيد التحديثات وصيانة الاتساق.

05 تنفيذ قواعد جودة بيانات مرنة
- تضمين فحص وتصفية البيانات أثناء تدفقها باستخدام أدوات تدفق بيانات ذكية (مثل Apache Kafka، Apache Flink).

06 تطبيق المعالجة الجزئية (MICRO-BATCHING) عند الحاجة:
- تجميع البيانات الصغيرة في دفعات قصيرة لتسهيل المعالجة والتحديث مع الحفاظ على سرعة الاستجابة.

07 التكامل مع أنظمة البيانات الدفعة بفعالية:
- استخدام منصات(ETL/ELT) قادرة على التعامل مع كلا النوعين من البيانات، وضمان تزامن البيانات بين المستودع وبيانات التدفق.

08 المراقبة المستمرة وأتمتة التحديثات:
- مراقبة جودة البيانات وأداء النماذج بشكل لحظي، مع أتمتة عمليات الإصلاح والتحديث لتقليل الأخطاء البشرية.

09 التكامل بين البيانات المتدفقة والبيانات الدفعة
- دمج البيانات الحية مع البيانات المخزنة يحتاج تنسيقاً دقيقاً لضمان التناسق.

أفضل الممارسات في نمذجة البيانات المتدفقة

استخدام نماذج مرنة وقابلة للتوسع:
- الاعتماد على نماذج مبسطة تقلل من التعقيد في التمثيل مثل نماذج نجمة مبسطة أو مخازن بيانات مخصصة للبيانات المتدفقة.

تبني استراتيجيات التطبيع والدمج المتوازنة:
- اعتماد مزيج بين التطبيع لتقليل التكرار حيثما أمكن، وإزالة التطبيع (De-normalization) لتسريع الوصول حيث يكون ذلك ضرورياً.

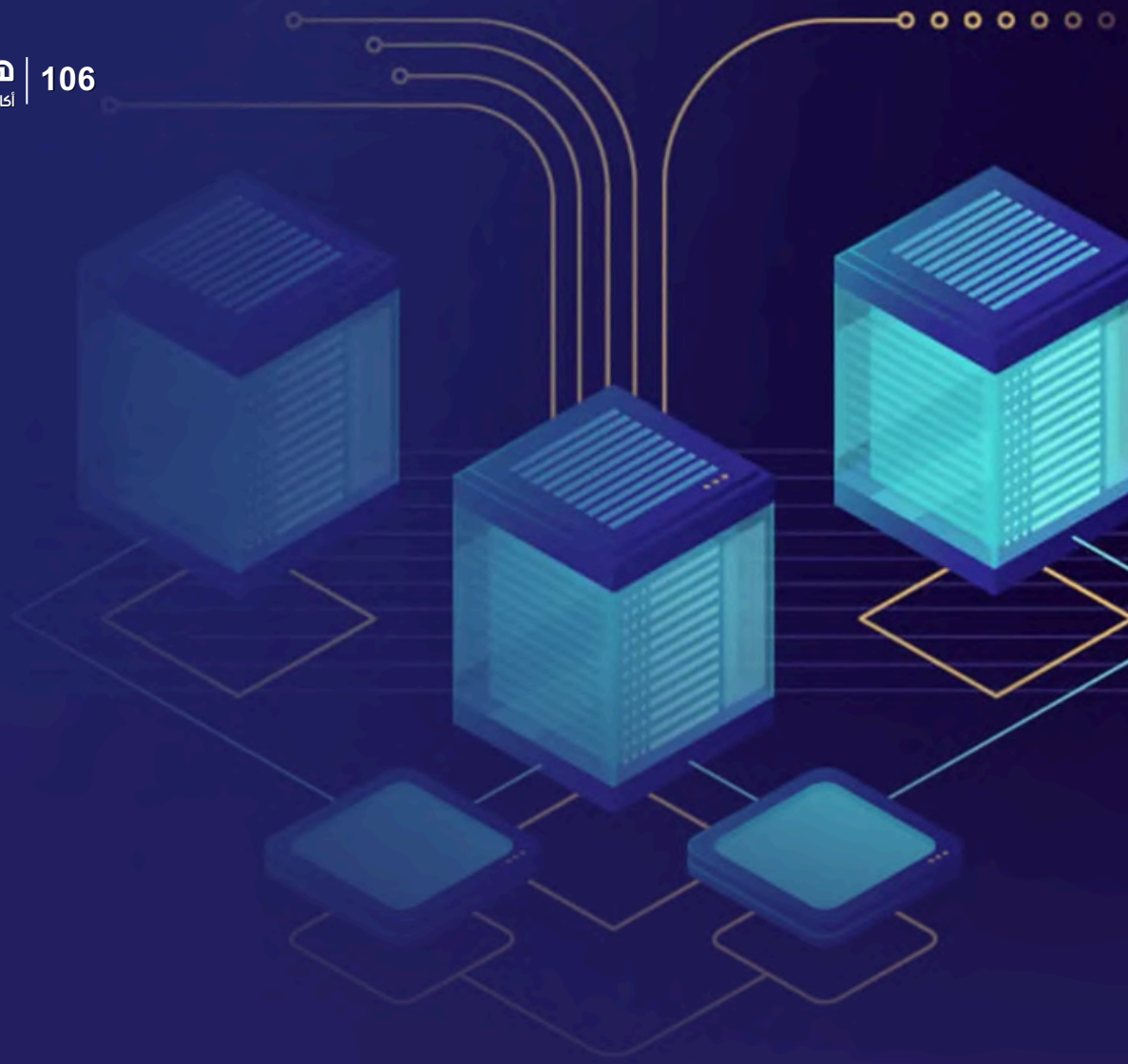
إدارة الأبعاد بطيئة التغير بآليات مخصصة:
- تطبيق تقنيات تحديث ذكية مثل نوع 2 (SCD Type 2) مع آليات مخصصة لتسجيل التغييرات في الوقت الحقيقي.

نمذجة البيانات المتدفقة تتطلب توازنًا دقيقًا بين جودة البيانات، سرعة الوصول إليها، ومرونة النموذج لتلبية احتياجات التحليل في الوقت الحقيقي. الالتزام بأفضل الممارسات وتقنيات التحديث المتقدمة يساعد المؤسسات على استخراج القيمة الفعلية من بياناتها المتدفقة، مع الحفاظ على الاتساق والتناسق عبر كامل نظام البيانات.

في ختام الفصل السابع

الاستعلام النمذجة وتدفق البيانات

تُعد الاستعلامات ونماذج البيانات من أساسيات هندسة البيانات، حيث تتيح الاستعلامات استرجاع البيانات وتحليلها بكفاءة، بينما تساعد نماذج البيانات على تنظيمها وتمثيلها بما يعكس واقع الأعمال ويدعم اتخاذ القرار. تختلف النماذج وتقنيات المعالجة حسب طبيعة البيانات سواء كانت دافعية أو متدفقة، ويؤثر اختيار الاستراتيجية المناسبة - مثل(Kimball أو Inmon أو Data Vault)- في فعالية التحليل وأداء النظام. ومن خلال تطبيق أفضل الممارسات مثل التطبيع، وفهم الأبعاد البطيئة التغير، وتحسين الاستعلامات، يمكن بناء منظومة بيانات مرنة، دقيقة، وذات كفاءة عالية تدعم تطور الأعمال وتفاعلها مع التغيرات المستقبلية.



08

اختيار التقنيات عبر دورة حياة هندسة البيانات

(Selecting Technologies Across the Data Engineering Lifecycle)

الفصل الثامن

اختيار التقنيات عبر دورة حياة هندسة البيانات

في هذا الفصل الأخير من كتيب هندسة البيانات، نستكشف كيف تختار التقنيات المناسبة عبر دورة حياة هندسة البيانات، مع التركيز على التحول إلى الحوسبة السحابية، التحديات المالية المرتبطة بها، وتقنيات الإدارة الحديثة مثل(FinOps).

فهم هذه الجوانب يمكن أن يحدث فارقًا استراتيجيًا في نجاح مشاريع البيانات، ويساعد المؤسسات على تحقيق المرونة، الاستدامة، والابتكار في بيئات متغيرة بسرعة.

8.1 اختيار التقنيات عبر دورة حياة هندسة البيانات

لماذا نحتاج للتمييز بين التقنيات الثابتة والمتغيرة؟

إن فهم الفرق بين هذين النوعين من التقنيات يساعد الشركات على اتخاذ قرارات استراتيجية تحافظ على استقرار الأنظمة التقنية وتجنب استثمار الأموال في تقنيات قد تصبح قديمة بسرعة.

- **تعريف التقنيات وأنواعها مع أمثلة :**

النوع	التعريف	أمثلة على التقنيات	المميزات	التحديات
التقنيات المتغيرة (Transitory Technologies)	تقنيات تتطور بسرعة وتظهر وتختفي في فترة قصيرة. تتطلب متابعة مستمرة.	أدوات تدفق البيانات مثل: Apache Kafka، Apache NiFi)، وأطر تحليل البيانات الجديدة	مرونة عالية، دعم الابتكار، استجابة سريعة للسوق	تكلفة التحديثات المستمرة، خطر التقادم السريع
التقنيات الثابتة (Immutable Technologies)	تقنيات مستقرة ومتطورة ببطء، توفر استدامة على المدى الطويل.	قواعد بيانات مثل(PostgreSQL)، أنظمة تخزين مثل (Amazon S3)، لغات مثل(SQL Bashg)	استقرار، تكاليف تشغيل منخفضة على المدى الطويل	أقل مرونة في التكيف مع تغييرات السوق السريعة

الخلاصة:

- التقنيات الثابتة توفر أمانًا واستدامة للاستثمار على المدى الطويل.
- التقنيات المتغيرة تتطلب متابعة مستمرة وتحديثات متكررة.
- من الأفضل مراجعة الأدوات التقنية كل عامين لضمان التوازن بين الاستدامة والمرونة.

8.2 اهمية العمل على الانظمة السحابية

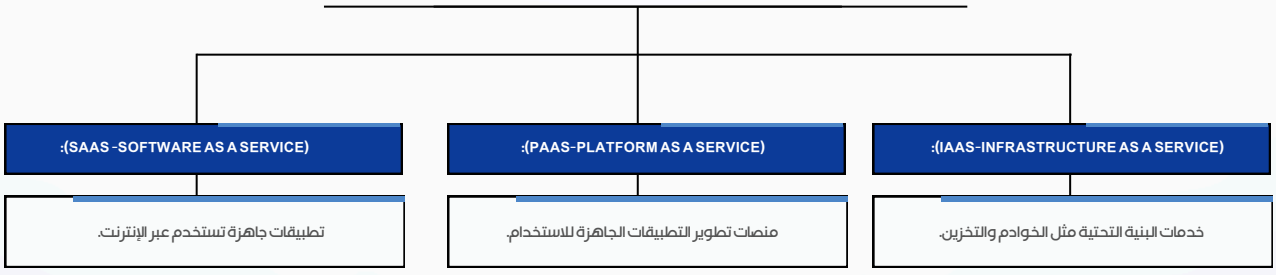
لماذا يجب على الشركات التفكير في الانتقال إلى السحابة؟

توفر السحابة مزايا مثل التوسع السريع، تقليل التكاليف الرأسمالية، وتسهيل الوصول إلى التقنيات الحديثة، لكنها تتطلب فهمًا عميقًا للنماذج المالية والاستراتيجية لضمان الاستفادة القصوى.

- **نماذج الانتقال إلى السحابة (Cloud Adoption Models):**

النموذج	التعريف	المزايا	العيوب
الخوادم المحلية (On-premises)	الاعتماد على مراكز بيانات داخلية خاصة بالشركة.	تحكم كامل في البيئة، أمان البيانات	تكلفة عالية، صعوبة التوسع
السحابة الخاصة (Private Cloud)	بيئة سحابية مخصصة للشركة فقط، سواء مستضافة داخليًا أو خارجيًا.	أمان وخصوصية أفضل من السحابة العامة، مرونة	تكلفة متوسطة، تعقيد الإدارة
السحابة العامة (Public Cloud)	خدمات سحابية من مزودين مثل (AWS، Azure، Google Cloud).	مرونة عالية، دفع حسب الاستخدام، توسع سريع	مخاطر أمنية محتملة، التكاليف قد تكون غير متوقعة

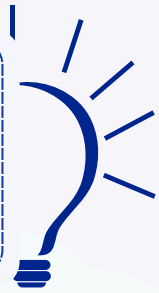
شرح مبسط للخدمات السحابية:



الخلاصة:

- الانتقال إلى السحابة يعزز مرونة الأعمال ويخفض التكاليف الرأسمالية.
- يتطلب فهم نماذج التسعير ومراقبة دقيقة للتكاليف عبر أدوات مثل (FinOps).

8.3 السحابة الهجينة والمتعددة



ماذا تعني السحابة الهجينة والمتعددة؟

توفر هذه البيئات مرونة أكبر في اختيار الخدمات المناسبة لكل نوع من البيانات والعمليات، مما يساعد الشركات على تحقيق توازن بين الأمان، الأداء، والتكلفة.

- **نماذج الانتقال إلى السحابة (Cloud Adoption Models):**

النوع	التعريف	المزايا	التحديات
السحابة الهجينة (Hybrid Cloud)	دمج السحابة الخاصة مع السحابة العامة لتوزيع البيانات والعمليات	تخزين البيانات الحساسة محليًا، واستغلال مزايا السحابة العامة	تعقيد الإدارة، تكامل الأمان
السحابة المتعددة (Multicloud)	استخدام خدمات من مزودين مختلفين مثل (AWS وAzureg)	تقليل الاعتماد على مزود واحد، تعزيز الاستقرار	زيادة تعقيد الإدارة، تحديات التكامل

- **سيناريو عملي:**

شركة تستخدم السحابة الهجينة لتخزين البيانات المالية الحساسة في السحابة الخاصة، بينما تستفيد من السحابة العامة لتحليل البيانات الكبيرة غير الحساسة.

الخلاصة:

- السحابة الهجينة توفر توازنًا بين الأمان والمرونة.
- السحابة المتعددة تقلل المخاطر المرتبطة بمزود خدمة واحد.

8.4 التحديات المرتبطة بالانتقال إلى السحابة

التحديات المالية (Financial Challenges)

الانتقال إلى السحابة قد يؤدي إلى تكاليف غير متوقعة بسبب تعقيد نماذج التسعير، والتكاليف الخفية مثل رسوم نقل البيانات (Data egress fees).

نصائح للتعامل مع التحديات المالية:

- مراجعة استخدام الموارد بشكل دوري (شهرًا على الأقل).
- تحديد ميزانيات واضحة للموارد السحابية.
- الاستفادة من أدوات مراقبة التكاليف.

الخلاصة:

- التحديات المالية للسحابة يمكن تقليلها عبر التخطيط والمراقبة الدقيقة.
- ضرورة الحذر من التكاليف الخفية التي قد تؤثر على الميزانية.

8.5 تقنيات(FinOps)

ما هو(FinOps)؟

(FinOps)هو نهج إداري يجمع بين الفرق التقنية والمالية بهدف تحسين كفاءة الإنفاق على الخدمات السحابية، ويضمن التعاون بين الأطراف المختلفة لإدارة التكاليف بفعالية.

- **إدارة التكاليف في السحابة (Cloud Cost Management)**

يعمل(FinOps) على وضع سياسات وتطبيق أدوات تساعد في تتبع، تحليل، وتحسين تكاليف السحابة.

- **أدوات (FinOps) شائعة:**

(CloudHealth) (CloudCheckr) (AWS Cost Explorer) (Google Cloud Billing)

- **الخطوات الأساسية لـ(FinOps):**

التخطيط:
تحديد الميزانيات والأهداف المالية

المراقبة:
متابعة استهلاك الموارد بشكل دوري

التحليل:
تقييم الأنماط والاتجاهات في الإنفاق

التكيف:
اتخاذ إجراءات لتحسين الكفاءة وتخفيض التكاليف

الخلاصة:

- (FinOps) يوفر إطارًا عمليًا لإدارة التكاليف السحابية بكفاءة.
- التعاون بين الفرق التقنية والمالية ضروري لنجاحه.

8.6 السحابة بدون خوادم (Serverless Cloud Technologies)

ما هي السحابة بدون خوادم؟

تقنيات تتيح تشغيل التطبيقات والخدمات بدون الحاجة لإدارة البنية التحتية للخوادم، حيث يدير مزود الخدمة السحابي كل الموارد المطلوبة.

مزايا السحابة بدون خوادم:

- **التوسع التلقائي:** الموارد تتوسع تلقائيًا حسب الطلب.
- **الدفع حسب الاستخدام:** تدفع فقط مقابل الموارد المستخدمة فعليًا.
- **تسريع التطوير:** تقليل الحاجة لإدارة البنية التحتية يسرّع إطلاق الخدمات.

عيوب السحابة بدون خوادم:

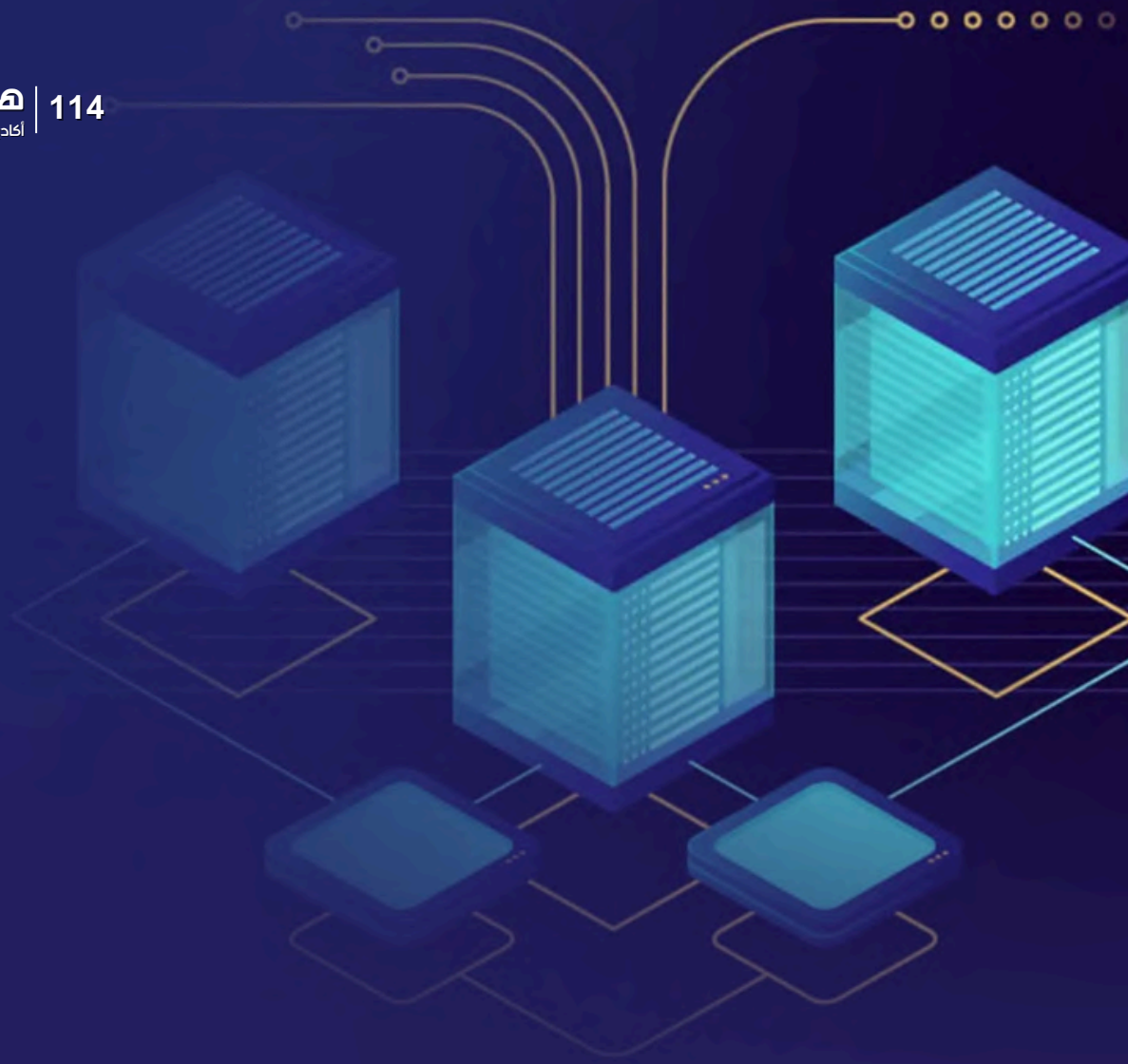
- قد تواجه تحديات في ضمان الأداء خلال فترات الحمل العالي.
- قيود على مدة تنفيذ الوظائف.
- صعوبة في مراقبة الأداء وتعقيد التتبع.

- **أمثلة على تقنيات (Serverless):**

(AWS Lambda) | (Azure Functions) | (Google Cloud Functions)

الخلاصة:

- السحابة بدون خوادم توفر مرونة كبيرة، لكنها تتطلب إدارة ذكية لتفادي مشاكل الأداء.



في ختام الفصل الثامن

اختيار التقنيات عبر دورة حياة هندسة البيانات

اختيار الأدوات والتقنيات السحابية ليس مجرد قرار تكتيكي، بل هو قرار استراتيجي يؤثر في قدرة الشركة على التوسع والابتكار.من المهم تحقيق توازن بين اعتماد التقنيات الثابتة التي توفر استدامة واستقرار، والتقنيات المتغيرة التي تمنح المرونة والابتكار.

ابدأ بتقييم بيئتك التقنية الحالية، واعتمد استراتيجيات مناسبة تضمن استدامة استثماراتك مع الحفاظ على المرونة اللازمة لمواكبة التغيرات المستقبلية.

ختاماً

تمثل هندسة البيانات اليوم أكثر من مجرد تخصص تقني؛ إنها دعامة استراتيجية لتمكين المؤسسات من استثمار أصولها الرقمية وتحقيق أقصى فاعلية في بيئة عمل تتسم بالتغير السريع والتعقيد المتزايد.

وقد سعينا من خلال هذا الكتيب إلى تقديم رؤية متكاملة ومبسطة لمفاهيم هندسة البيانات، مدعومة بأمثلة تطبيقية وأساليب حديثة، بهدف بناء معرفة عملية راسخة لدى القارئ، تُمكّنه من التعامل بوعي وكفاءة مع مختلف مراحل دورة البيانات، والتحديات التقنية المرتبطة بها.

ومع التوسع المستمر في البيانات الضخمة، والحوسبة السحابية، والتقنيات المتقدمة، تزداد الحاجة إلى تعلّم مستمر وتطوير دائم للمهارات. ونأمل أن يكون هذا الإصدار نقطة انطلاق لكل ممارس أو مهتم، نحو اكتساب أدوات التمكين في عالم تصنع فيه القيمة عبر إدارة البيانات وتحليلها بذكاء واحتراف.

ووفاءً بالتزامها نحو تمكين الكفاءات العربية، تُجدد أكاديمية بيان حرصها على تقديم محتوى معرفي متخصص باللغة العربية، يواكب التحولات التقنية، ويسهم في إعداد جيل قادر على قيادة مشاريع البيانات بثقة، والمساهمة في تحقيق أهداف التحول الرقمي الوطني.

نثق أن ما يحمله هذا الكتيب من معرفة سيكون خطوة فارقة في مسيرتك المهنية، ونسعد بأن نكون شركاء في رحلتك نحو التميز في عالم البيانات.

أكاديمية بيان

المصادر

01 data engineering fundamentals book

02 https://www.oreilly.com/library/view/fundamentals-of-data/9781098108297/

03 https://www.finops.org

04 https://aws.amazon.com/professional-services/CAF/

05 https://cloud.google.com/architecture/framework

06 https://learn.microsoft.com/en-us/cloud-adoption-framework/

07 https://docs.aws.amazon.com/lambda/

08 https://cloud.google.com/functions

09 https://learn.microsoft.com/en-us/azure/azure-functions/

10 https://www.vmware.com/products/cloudhealth.html

11 https://www.cloudcheckr.com

12 https://www.db-book.com/

13 https://www.pearson.com/us/higher-education/program/Elmasri-Fundamentals-of-Database-Systems-7th-Edition/PGM310557.html

14 https://www.tutorialspoint.com/dbms/dbms_normalization.htm

15 https://docs.oracle.com/cd/B19306_01/server.102/b14220/normalization.htm

16 https://www.geeksforgeeks.org/transitive-dependency-in-dbms/

17 https://kimballgroup.com/data-warehouse-business-intelligence-resources/books/

18 https://www.inmoncif.com/data-warehouse-concepts/

19 https://docs.microsoft.com/en-us/azure/data-factory/tutorial-slowly-changing-dimension

20 https://kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/slowly-changing-dimensions/

21 https://www.confluent.io/blog/streaming-architecture-best-practices/

22 https://kafka.apache.org/documentation/

23 https://cloud.google.com/architecture/real-time-data-processing

24	data engineering fundamentals book
25	https://www.oreilly.com/library/view/fundamentals-of-data/9781098108297/
26	https://www.oreilly.com/library/view/streaming-systems/9781491983867/
27	https://learn.microsoft.com/en-us/training/paths/design-build-data-engineering/
28	https://docs.confluent.io/platform/current/kafka/introduction.html
29	https://martinfowler.com/articles/microservices.html
30	https://d1.awsstatic.com/whitepapers/aws-disaster-recovery.pdf
31	https://cloud.google.com/blog/products/identity/foundations-of-zero-trust
32	https://www.finops.org/about/what-is-finops/
33	https://aws.amazon.com/aws-cost-management/
34	https://azure.microsoft.com/en-us/services/cost-management/
35	https://databricks.com/product/data-lakehouse
36	https://www.snowflake.com/blog/the-snowflake-data-cloud-data-lakehouse/
37	https://cloud.google.com/bigquery/docs/architecture
38	https://docs.aws.amazon.com/redshift/latest/mgmt/welcome.html
39	https://cloud.google.com/bigquery/docs
40	https://docs.snowflake.com/en/
41	https://www.db-book.com/
42	https://www.pearson.com/us/higher-education/program/Elmasri-Fundamentals-of-Database-Systems-7th-Edition/PGM310557.html
43	https://www.tutorialspoint.com/dbms/dbms_normalization.htm
44	https://docs.oracle.com/cd/B19306_01/server.102/b14220/normalization.htm
45	https://www.geeksforgeeks.org/transitive-dependency-in-dbms/
46	https://kimballgroup.com/data-warehouse-business-intelligence-resources/books/

47 https://www.inmoncif.com/data-warehouse-concepts/

48 https://docs.microsoft.com/en-us/azure/data-factory/tutorial-slowly-changing-dimension

49 https://kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/slowly-changing-dimensions/

50 https://www.confluent.io/blog/streaming-architecture-best-practices/

51 https://kafka.apache.org/documentation/

52 https://cloud.google.com/architecture/real-time-data-processing

53 https://atlan.com/data-ingestion-101/

54 https://atlan.com/know/data-ingestion-vs-etl/

55 https://airbyte.com/data-engineering-resources/data-ingestion-vs-data-integration

56 https://matillion.com/blog/an-introduction-to-data-ingestion

57 https://medium.com/@nripapathak/data-ingestion-patterns-95624ab1e360

58 https://medium.com/@raj.busint/batch-vs-micro-batch-vs-streaming-when-to-use-what-and-why-it-matters-f6063456c389

59 https://upsolver.com/blog/batch-stream-a-cheat-sheet

60 https://quix.io/blog/apache-kafka-vs-rabbitmq-comparison

61 https://www.simplilearn.com/kafka-vs-rabbitmq-article

62 https://www.cloudamqp.com/blog/when-to-use-rabbitmq-or-apache-kafka.html

63 https://en.wikipedia.org/wiki/RabbitMQ

64 https://en.wikipedia.org/wiki/Apache_Kafka

65 https://medium.com/@sheikh.hamza.arshad/choosing-the-right-messaging-system-kafka-redis-rabbitmq-activemq-and-nats-compared-fa2dd385976f

67 https://gcore.com/learning/nats-rabbitmq-nsq-kafka-comparison

68 https://en.wikipedia.org/wiki/Extract%2C_transform%2C_load

69 https://www.amazon.com/Fundamentals-Data-Engineering-Robust-Systems/dp/1098108302

70 https://waruithemystery.hashnode.dev/unleashing-the-power-of-data-storage

71 https://medium.com/towards-data-engineering/data-engineering-lifecycle-d1e7ee81632e

72 https://www.linkedin.com/pulse/comprehensive-guide-data-engineering-part-two-lifecycle-imade-cpuze

73 https://www.itamg.com/data-storage/hdd-vs-ssd/

74 https://www.geeksforgeeks.org/10-advantages-and-disadvantages-of-cloud-storage/

75 https://objectfirst.com/guides/data-storage/tiered-storage-best-practices/

76 https://www.linkedin.com/pulse/data-partitioning-clustering-performance-optimization-lata-ujmae

77 https://www.montecarlodata.com/blog-data-warehouse-vs-data-lake-vs-data-lakehouse-definitions-similarities-and-differences/

78 https://www.upsolver.com/blog/manage-your-data-schema-on-read-vs-schema-on-write

79 https://medium.com/art-of-data-engineering/schema-on-read-vs-schema-on-write-29ad2db3b50e

80 https://dataintensive.net/

81 https://martinfowler.com/articles/microservices.html

82 https://d1.awsstatic.com/whitepapers/aws-disaster-recovery.pdf

83 https://cloud.google.com/blog/products/identity/foundations-of-zero-trust

84 https://www.finops.org/about/what-is-finops/

85 https://aws.amazon.com/aws-cost-management/

86 https://azure.microsoft.com/en-us/services/cost-management/

89 https://databricks.com/product/data-lakehouse

90 https://www.snowflake.com/blog/the-snowflake-data-cloud-data-lakehouse/
https://cloud.google.com/bigquery/docs/architecture

91 https://docs.aws.amazon.com/redshift/latest/mgmt/welcome.html

92 https://docs.snowflake.com/en/

93 https://cloud.google.com/bigquery/docs

94 https://medium.com/towards-data-engineering/data-engineering-lifecycle-d1e7ee81632e

95 https://lumenalta.com/insights/6-stages-of-the-data-engineering-lifecycle%3A-from-concept-to-execution

96	https://medium.com/@talha002/aws-data-engineering-data-engineering-life-cycle-9ae7b94fc03e
97	https://www.getcensus.com/blog/how-understanding-the-data-engineering-lifecycle-helps-us-all-work-better-with-data-engineers
98	Google Cloud - Data Engineering on Google Cloud Platform Specialization
99	Microsoft Learn – Azure Data Engineer Learning Path
100	AWS Certified Data Analytics – Specialty Guide
76	Designing Data-Intensive Applications by Martin Kleppmann (O'Reilly)
77	Apache Kafka Documentation
78	Apache Airflow Documentation
79	Snowflake Documentation
80	Talend Documentation
81	IBM – What is a Data Engineer?
82	Data Engineering Cookbook by Andreas Kretz

academy.btech.sa bayan-academy +966534298259

Bayan_academia Training@bfuture.edu.sa