

BIG DATA



❖ Agenda

- *Data, Information, Knowledge*
- *Historical of the Database (Big Data)*
- *Data revolution*
- *Definition of Big Data*
- *Why do we need big data?*
- *Big Data Applications*

البيانات Data

- . يشير مفهوم البيانات إلى مجموعة من الحقائق التي تمثل مدخلات النظام، وهي حقائق في حالتها الأولية ليست لها معنى حيث لم يتم معالجتها.
- . تتخذ البيانات أشكال عديدة من العبارات أو الأرقام أو الرموز، وتمثل المواد الخام للمعلومات.

المعلومات Information

- . هي مجموعة من البيانات التي تمت معالجتها وتنظيمها وتنسيقها، وت تكون من عدد من المفاهيم والأفكار.
- . كما يشير مفهوم المعلومات إلى البيانات المعالجة التي تجيب على الأسئلة التي تبدأ بأين ومتى ومن وماذا.
- . عرّف العلماء أيضًا المعلومات على أنها بيانات خضعت للمعالجة والتحليل والتصفيية من أجل الوصول إلى هدف محدد ، حيث تمت معالجتها إما يدوياً أو عبر الحاسب الآلي.
- . أشار بعض العلماء في تعريف المعلومات لغوياً بأنه مصطلح يعود اشتراقه إلى مصطلح العلم الذي يشير في مفهومه إلى مجموعة من المعاني المختلفة التي يتعلمها المتلقى.

المعرفة Knowledge

- . يشير مفهوم المعرفة إلى أنها مجموعة من المعلومات التي يتم تداولها بين شخص وآخر من أجل الاستفادة منها.
- . كما تُعرّف المعرفة على أنها حصيلة الخبرات التي اكتسبها الإنسان، وتتطابق الاستفادة منها تطبيق وممارسة كل ما تم تعلمه.

Historical of the Database (Big Data)

يمكن إرجاع تاريخ البيانات الضخمة إلى تطور تكنولوجيا المعلومات، والقدرات الحسابية. فيما يلي لمحّة موجزة عن المعلم الرئيسي في تاريخ البيانات (الضخمة) :

1. 1950s - The Emergence of Data Management:

- Data management and storage emerged with the development of early computing machines.
- Magnetic tapes and punch cards were used for data storage and retrieval.

الخمسينيات - ظهور إدارة البيانات:

- ظهرت إدارة البيانات وتخزينها مع تطوير آلات الحوسبة المبكرة.
- تم استخدام الأشرطة المغناطيسية وبطاقات اللكم لتخزين البيانات واسترجاعها.

2. 1960s - Hierarchical and Network Databases:

- The first generation of databases emerged, including hierarchical and network databases.

الستينيات - قواعد بيانات التسلسل الهرمية والشبكية:

- ظهر الجيل الأول من قواعد البيانات ، بما في ذلك قواعد بيانات التسلسل الهرمي والشبكات.

3. 1970s - The Relational Database Revolution:

- The most transformative development in the history of databases occurred with the introduction of the relational database model.

- سبعينيات القرن الماضي - ثورة قاعدة البيانات العلائقية:
- حدث التطور الأكثر تحويلياً في تاريخ قواعد البيانات مع إدخال نموذج قاعدة البيانات العلائقية.

4. 1980s - Commercialization and Widespread Adoption:

- The 1980s witnessed the commercialization and widespread adoption of relational database management systems (RDBMS).

- الثمانينات - التسويق والاعتماد على نطاق واسع:
- شهدت الثمانينات التسويق والاعتماد الواسع على نطاق واسع لأنظمة إدارة قاعدة البيانات العلائقية .

4. 1990s - Advancements in Database Technology:

- The 1990s brought advancements in database technology, including support for multimedia data types and the growth of **client-server architectures**.
 - التسعينيات - التطورات في تكنولوجيا قاعدة البيانات:
 - جلبت التسعينيات التطورات في تكنولوجيا قواعد البيانات ، بما في ذلك دعم أنواع بيانات الوسائط المتعددة ونمو بنية خادم العميل.

5. 2000s Emergence of Big Data Technologies:

- The term "big data" started gaining traction in the mid-2000s as organizations faced challenges managing and analyzing large and diverse datasets.
- 2000 - العقد الأول من القرن الحادي والعشرين :
 - ظهور تقنيات البيانات الضخمة، بدأ مصطلح "البيانات الضخمة" في اكتساب المزيد من الاهتمام في منتصف العقد الأول من القرن الحادي والعشرين، حيث واجهت المؤسسات تحديات في إدارة وتحليل مجموعات البيانات الكبيرة والمتنوعة.

6. 2004: Google's MapReduce:

Google introduced the MapReduce programming model, which allowed for the processing of large datasets across distributed computing clusters. This became a foundation for many big data processing frameworks.

قدمت Google نموذج البرمجة MapReduce، الذي سمح بمعالجة مجموعات البيانات الكبيرة عبر مجموعات الحوسبة الموزعة. وأصبح هذا أساساً للعديد من أطر معالجة البيانات الضخمة.

Con..

7. 2005: Hadoop is Created:

Hadoop, an open-source framework for distributed storage and processing of big data, was created by Doug Cutting and Mike Cafarella. Hadoop became a key technology for handling large-scale data processing tasks.

هو إطار عمل مفتوح المصدر للتخزين الموزع ومعالجة البيانات الضخمة، تم إنشاؤه بواسطة

Doug .Cutting and Mike Cafarella

أصبحت Hadoop تقنية أساسية للتعامل مع مهام معالجة البيانات واسعة النطاق.

8. 2008 Introduction of NoSQL Databases:

NoSQL databases, designed to handle unstructured and semi-structured data, gained popularity. MongoDB, Couchbase, and Cassandra were among the early NoSQL databases.

اكتسبت قواعد بيانات NoSQL، المصممة للتعامل مع البيانات غير المنظمة وشبه المنظمة، شعبية كبيرة. كانت MongoDB وCassandra وCouchbase من بين قواعد بيانات NoSQL المبكرة.

9. 2009 Apache Spark:

Apache Spark, an open-source, distributed computing system, was introduced. Spark provided a faster, more flexible alternative to MapReduce for big data processing.

تم تقديم Apache Spark ، وهو نظام حوسبة موزع ومفتوح المصدر. تعتبر أداة Spark بديلاً أسرع وأكثر مرونة لـ MapReduce لمعالجة البيانات الضخمة.

Con..

10. 2010s: Cloud Computing and Big Data Platforms:

Cloud computing platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, provide scalable and on-demand resources for big data processing. Big data platforms like Apache Hadoop, Apache Spark, and Apache Flink became widely adopted.

توفر منصات الحوسبة السحابية، مثل Microsoft Azure وAmazon Web Services (AWS) وGoogle Cloud Platform، موارد قابلة للتطوير وحسب الطلب لمعالجة البيانات الضخمة. حيث أصبحت منصات البيانات الضخمة مثل Apache Flink وApache Spark وApache Hadoop تستخدم على نطاق واسع.

11. 2012 Introduction of Apache Kafka:

Apache Kafka, a distributed streaming platform, was introduced. Kafka became a popular choice for handling real-time data streams.

- ظهر Apache Kafka، هي منصة بث موزعة.
- و تعتبر هذه الاداة خياراً شائعاً للتعامل مع تدفقات البيانات في الوقت الفعلي.

12. 2015 Big Data Analytics Maturity:

Big data analytics developed, with more organizations leveraging advanced analytics, machine learning, and artificial intelligence to derive insights from large datasets.

- بُرِزَت تحليلات البيانات الضخمة، حيث بدأ الكثيرون من المؤسسات باستخدام التحليلات المتقدمة والتعلم الآلي والذكاء الاصطناعي لاستخلاص الرؤى من مجموعات البيانات الكبيرة.

Con..

13. 2020s Continued Evolution:

Big data continues to evolve with technological advancements like Kubernetes, containerization, and edge computing. Organizations focus on integrating big data analytics with AI and machine learning for more sophisticated insights.

تستمر البيانات الضخمة في التطور مع التقدم التكنولوجي مثل Kubernetes، والحاويات، والحوسبة المتطورة و تركز المؤسسات على دمج تحليلات البيانات الضخمة مع الذكاء الاصطناعي والتعلم الآلي للحصول على رؤى أكثر تطوراً.

يشير مصطلح "ثورة البيانات" في سياق البيانات الضخمة إلى التغييرات العميقة والتحويلية الناجمة عن النمو السريع وإمكانية الوصول إلى مجموعات البيانات الكبيرة والمعقدة. ويرتبط ارتباطاً وثيقاً بالمفهوم الأوسع لثورة البيانات، حيث يركز بشكل خاص على تأثير البيانات الضخمة على مختلف الصناعات والتقنيات والممارسات المتعلقة بهذه البيانات. وفيما يلي بعض الجوانب الرئيسية لثورة البيانات في البيانات الضخمة:

Evolution of Technology

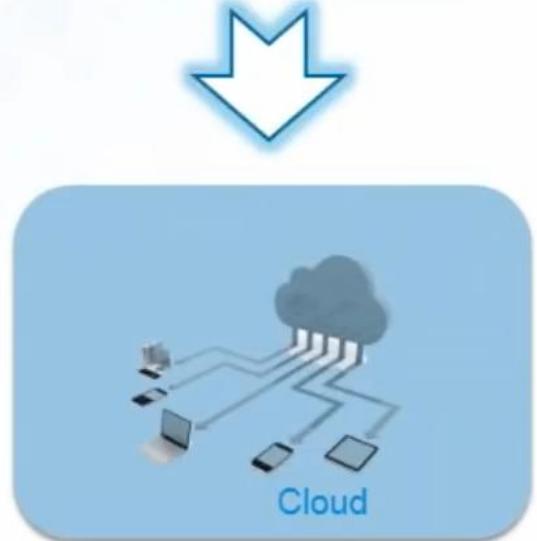
التطور التقني edureka!

1 Evolution of
Technology

2 IOT

3 Social Media

4 Data evolved
to Big Data



1 Evolution of Technology

2 IOT

3 Social Media

4 Data evolved to Big Data



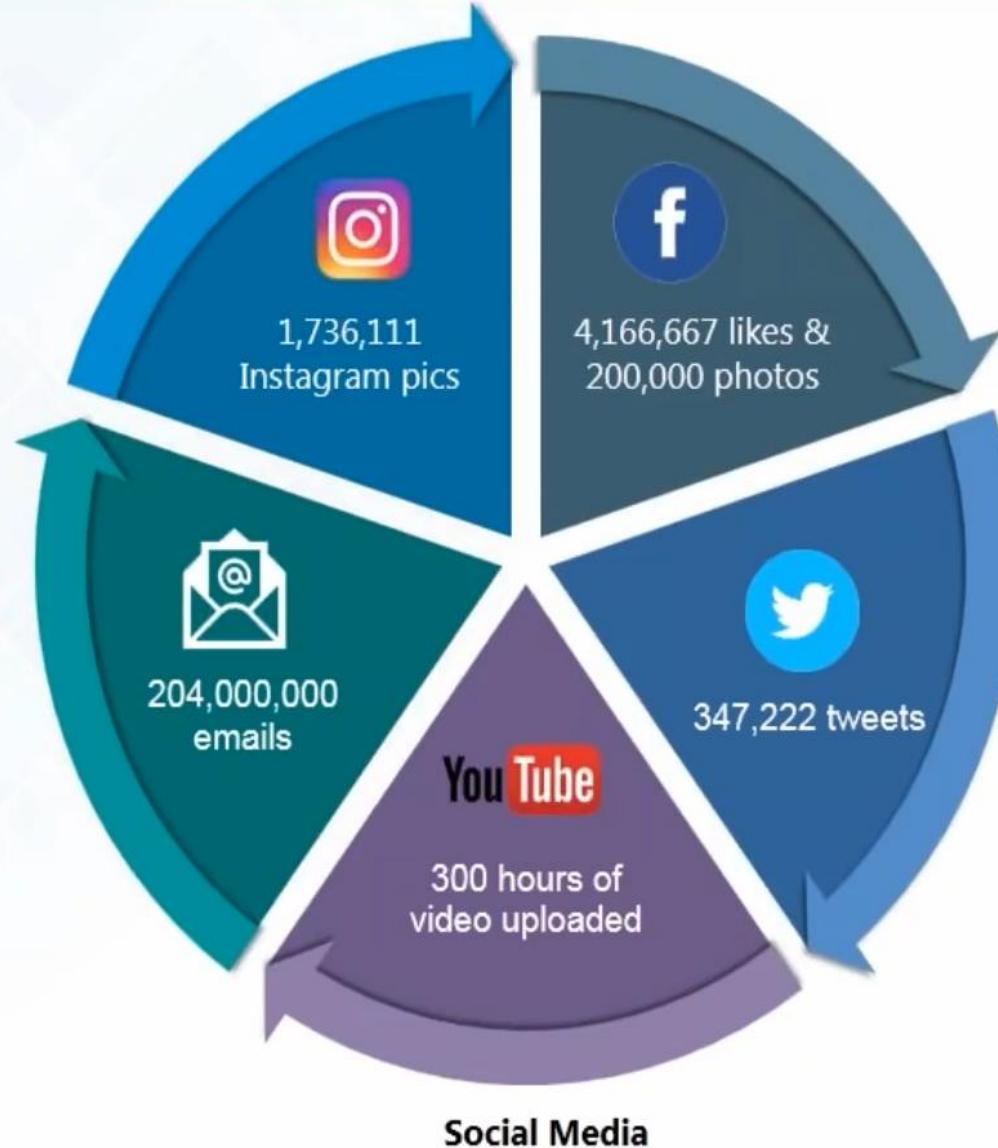
IOT: 50 Billion devices by 2020

1 Evolution of Technology

2 IOT

3 Social Media

4 Data evolved to Big Data



1 Evolution of Technology

2 IOT

3 Social Media

4 Other Factors



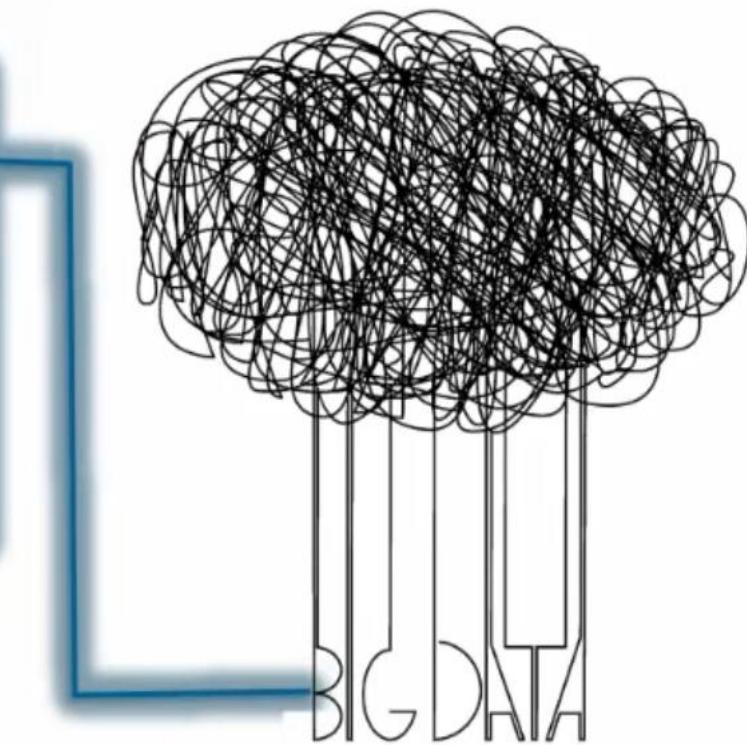
ما هي البيانات الكبيرة (الضخمة)؟

ما هي البيانات الكبيرة (الضخمة)؟
What Is Big Data?

What is Big Data

Big data is the term for collection of data sets so **large and complex** that it becomes difficult to process using on-hand database system tools or traditional data processing applications

تشير البيانات الضخمة إلى
مجموعات البيانات الكبيرة
والمعقدة التي تكون ضخمة
جداً بحيث لا يمكن
معالجتها وإدارتها وتحليلها
بشكل فعال باستخدام
أدوات وأساليب معالجة
البيانات التقليدية .



1

Volume

الحجم : تتضمن البيانات الضخمة عادةً كميات هائلة من المعلومات،

والتي غالباً ما يتم

إنشاؤها بسرعة أو معدل مرتفع. ويمكن أن يشمل ذلك البيانات من وسائل التواصل الاجتماعي

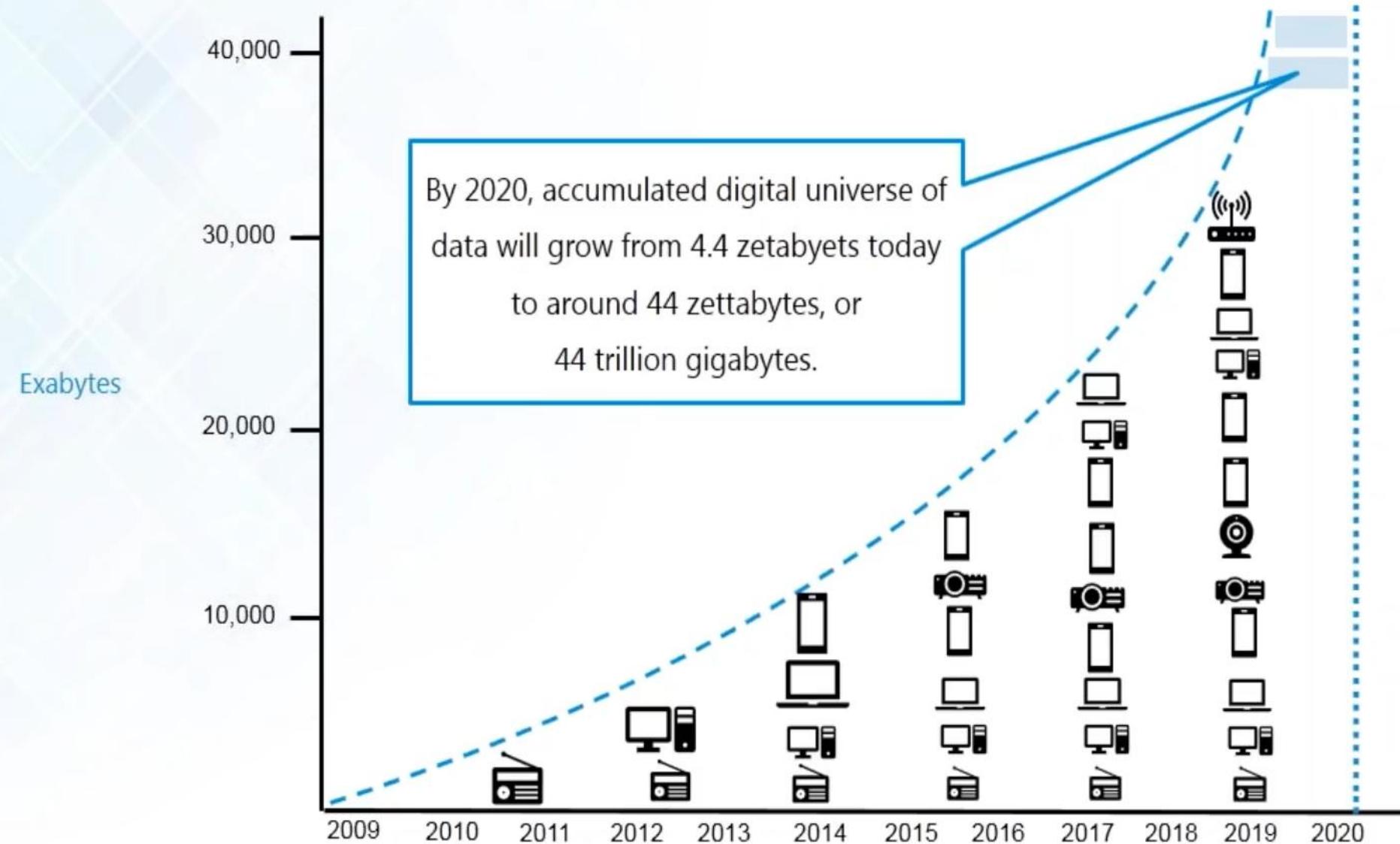
وأجهزة الاستشعار

ومدونات الويب والمزيد.

حجم البيانات كبير جداً

لدرجة أنه يتجاوز سعة

قواعد البيانات وأنظمة

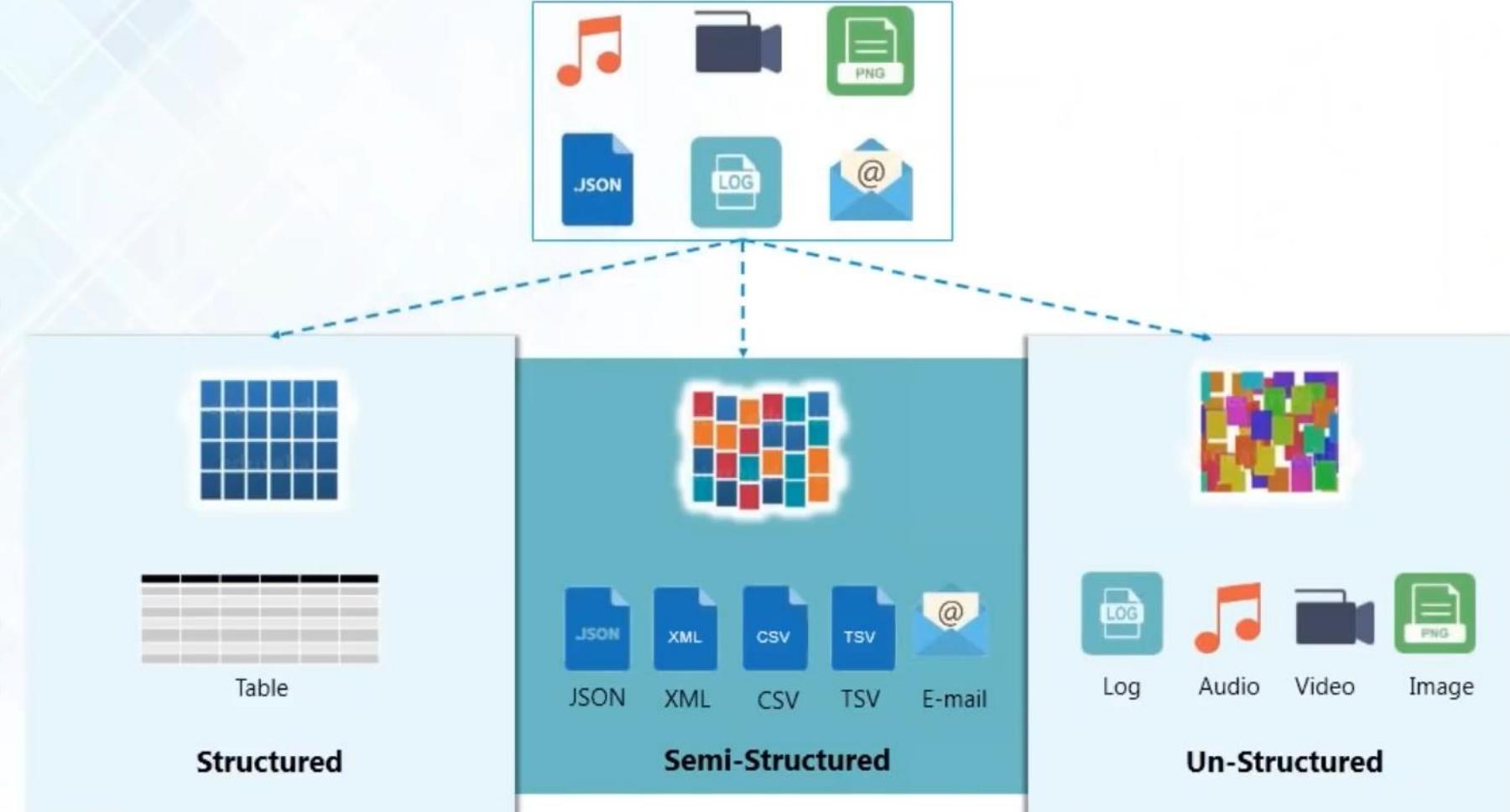


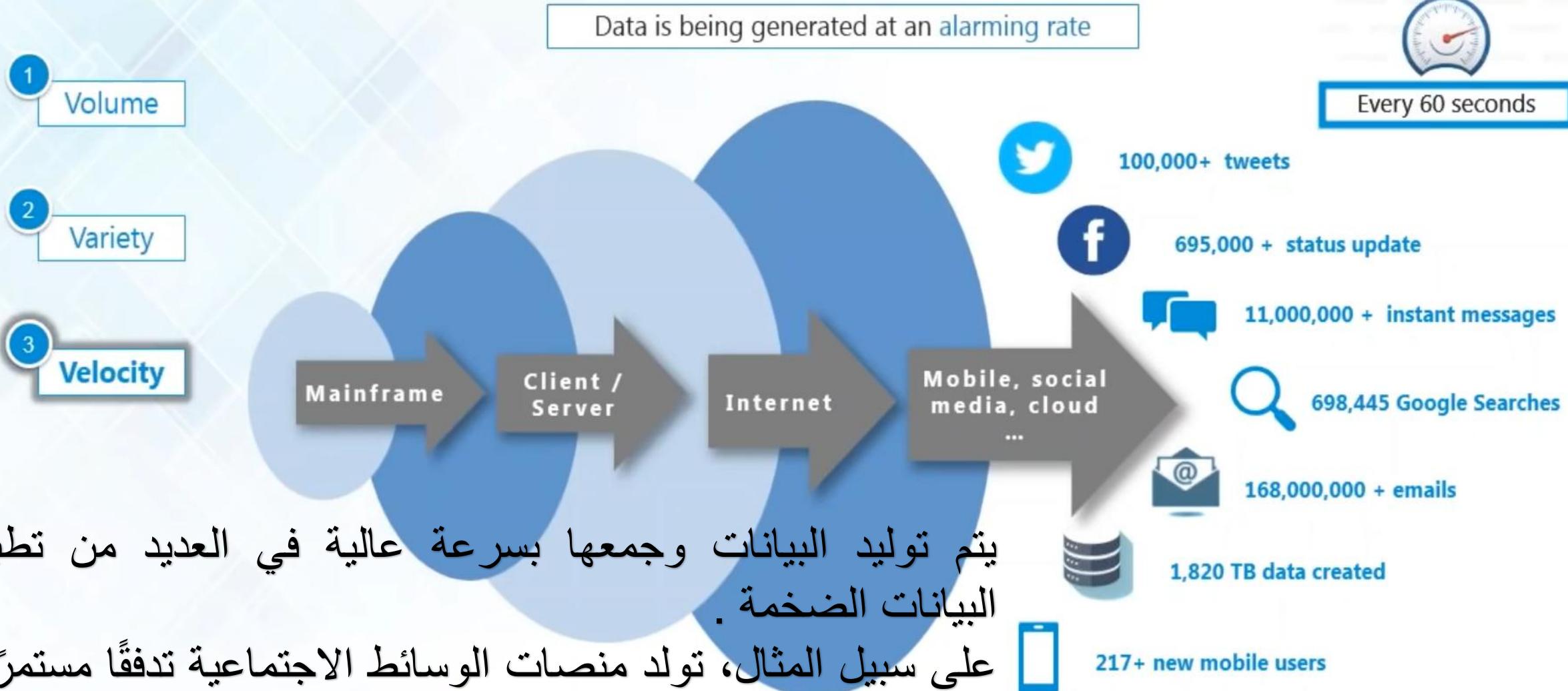
Different kinds of data is being generated from various sources

1 Volume

2 Variety

تشمل البيانات الضخمة مجموعة واسعة من أنواع البيانات، بما في ذلك البيانات المنظمة (مثل البيانات الموجودة في قواعد البيانات)، والبيانات شبه المنظمة (مثل XML أو JSON)، والبيانات غير المنظمة (مثل النصوص والصور ومقاطع الفيديو).





يتم توليد البيانات وجمعها بسرعة عالية في العديد من تطبيقات البيانات الضخمة .

على سبيل المثال، تولد منصات الوسائل الاجتماعية تدفقاً مستمراً من البيانات، وتنتج أجهزة الاستشعار في أجهزة إنترنت الأشياء (Internet of Things) في الوقت الفعلي .

1

Volume

2

Variety

3

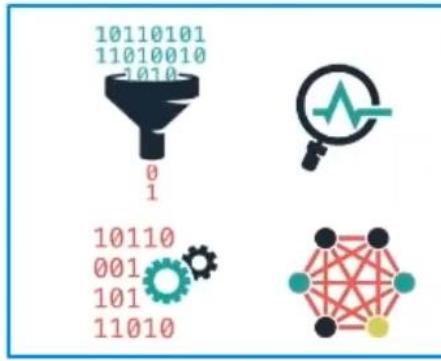
Velocity

4

Value

Mechanism to bring the correct meaning out of the data

1. الهدف النهائي للعمل مع البيانات الضخمة هو استخلاص رؤى قيمة.



2. اتخاذ قرارات تعتمد على البيانات، وخلق قيمة تجارية أو مجتمعية ويعتبر استخراج المعلومات المفيدة من البيانات الضخمة هو الهدف الرئيسي.

يشير إلى جودة البيانات ومصدقتيها. يمكن أن تتضمن البيانات الضخمة بيانات مشوشة أو غير كاملة أو غير دقيقة، وتعد إدارة جودة البيانات مصدر قلق كبير في تطبيقات البيانات الضخمة.

1

Volume

2

Variety

3

Velocity

4

Value

5

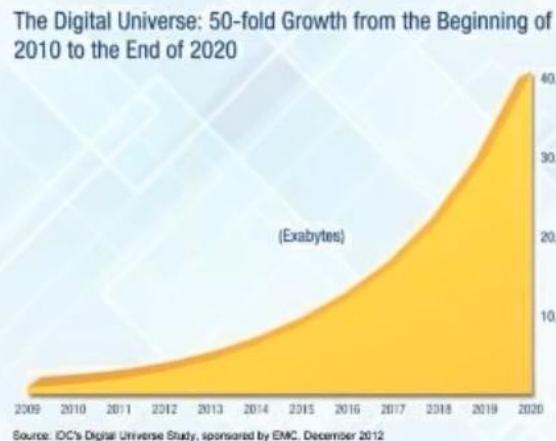
Veracity

	Min	Max	Mean	SD
1	4.3	?	5.84	0.83
2	2.0	4.4	3.05	50000000
3	15000	7.9	1.20	0.43
4	0.1	2.5	?	0.76

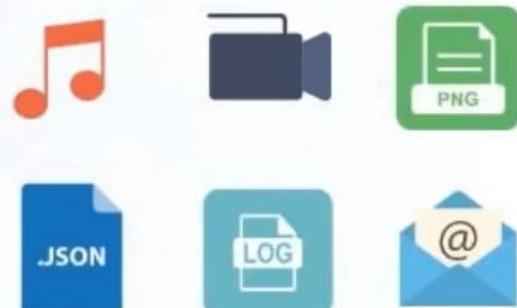
Uncertainty and inconsistencies in the data

5 V's of Big Data

edureka!



Volume



Different kinds of data is being generated from various sources



Data is being generated at an alarming rate



Mechanism to bring the correct meaning out of the data

Value

Min	Max	Mean	SD
4.3	?	5.84	0.83
2.0	4.4	3.05	50000000
15000	7.9	1.20	0.43
0.1	2.5	?	0.76

Uncertainty and inconsistencies in the data

Veracity

....

V's associated with Big Data may grow with time

للاستفادة بشكل فعال من إمكانات البيانات الضخمة، تستخدم المؤسسات **تحليلات البيانات المتقدمة**، والتعلم الآلي، وتقنيات التنقيب عن البيانات. ويساعد ذلك في الكشف عن الأنماط والاتجاهات والرؤى التي يمكن أن تفيد عملية صنع القرار، وتعزيز المنتجات والخدمات، وتحسين الكفاءة التشغيلية، والمزيد. تُستخدم تقنيات مثل قواعد بيانات Hadoop وSpark وNoSQL وبثيرات البيانات بشكل شائع لمعالجة البيانات الضخمة وتخزينها.

لماذا نحتاج البيانات الضخمة؟

Why do we need big data?

ମହା ଦେଶ ପାଇଁ ବିଶ୍ୱାସ କରିବାକୁ ପାଇଁ

ବିଶ୍ୱାସ କରିବାକୁ ପାଇଁ ମହା ଦେଶ

Why do we need big data?

تستخدم العديد من الشركات البيانات الضخمة في أنظمتها لتحسين العمليات وتقديم خدمة أفضل للعملاء وإنشاء حملات تسويقية مخصصة واتخاذ إجراءات أخرى يمكن أن تؤدي في النهاية إلى زيادة الإيرادات والأرباح. تتمتع الشركات التي تستخدمها بشكل فعال بميزة تنافسية محتملة على تلك التي لا تستخدمها لأنها قادرة على اتخاذ قرارات عمل أسرع وأكثر استنارة.

كما ان لديها الكثير من التطبيقات التي يتم استخدامها حاليا من قبل العديد من الشركات والمؤسسات الكبيرة والتي سوف نتحدث عنها في الفقرة التالية .

تطبيقات البيانات الضخمة.

Big Data Applications.

• تطبيقات البيانات الضخمة.

• Big Data Applications.

تنوع تطبيقات البيانات الضخمة وتغطي صناعات مختلفة، مما يزيد من القدرة على معالجة وتحليل كميات كبيرة من البيانات للحصول على رؤى واتخاذ قرارات مستنيرة واستخلاص القيمة. فيما يلي بعض تطبيقات البيانات الضخمة البارزة عبر مختلف القطاعات:

1. **تحليلات الرعاية الصحية Healthcare Analytics** : تحسين رعاية المرضى: تحليل سجلات المرضى والبيانات الطبية لتحسين خطط العلاج، والتنبؤ بتفشي الأمراض، وتحسين تقديم الرعاية الصحية.
2. **الخدمات المالية Financial Services** : كشف الاحتيال: تحديد ومنع الأنشطة الاحتيالية من خلال تحليل الأنماط والاختراقات في المعاملات المالية. إدارة المخاطر: تقييم وإدارة المخاطر المالية من خلال تحليل بيانات السوق والمؤشرات الاقتصادية والاتجاهات التاريخية.
3. **البيع بالتجزئة والتجارة الإلكترونية Retail and E-Commerce** : التسويق المخصص: استخدام بيانات العملاء لتقديم توصيات وعروض برويجية وإعلانات مخصصة.

4. الاتصالات : Telecommunications

- تحسين الشبكة: تحليل بيانات الشبكة لتحسين الأداء والتنبؤ بأعطال المعدات وتحسين جودة الخدمة.
- إدارة تجربة العملاء: استخدام البيانات لتحسين رضا العملاء ومعالجة مشكلات الخدمة وتعزيز تجربة المستخدم.

5. التصنيع والصناعة : Manufacturing and Industry

- الصيانة التنبؤية: استخدام بيانات الاستشعار والتحليلات للتنبؤ بأعطال المعدات وتحسين جداول الصيانة.
- مراقبة الجودة: تحليل بيانات الإنتاج لتحسين جودة المنتج وتقليل العيوب وتعزيز عمليات التصنيع.

6. المدن الذكية : Smart Cities

- التخطيط الحضري: تحليل البيانات من مصادر مختلفة، بما في ذلك أجهزة الاستشعار وأجهزة إنترنت الأشياء، لتحسين تخطيط المدن وإدارة حركة المرور والخدمات العامة.
- السلامة العامة: استخدام البيانات لتعزيز تدابير السلامة العامة، والتنبؤ بمناطق الجريمة الساخنة، وتحسين الاستجابة الحالات الطارئ.

7. الطاقة والمرافق :Energy and Utilities

- إدارة الشبكة: تحليل البيانات من الشبكات الذكية لتحسين توزيع الطاقة، وتقليل الهدر، وتعزيز موثوقية الشبكة.
- الصيانة التنبؤية للبنية التحتية: استخدام البيانات للتنبؤ بأعطال المعدات وتحسين الصيانة في قطاع الطاقة.

8. تحليلات التعليم Education Analytics

- تحسين أداء الطلاب: تحليل البيانات التعليمية لخصيص تجارب التعلم، وتحديد الطلاب المعرضين للخطر، وتحسين أساليب التدريس.

9. الإعلام والترفيه :Media and Entertainment

- توصية المحتوى: استخدام بيانات المستخدم للتوصية بالمحتوى المخصص على منصات البث.
- تحليلات الجمهور: تحليل سلوك المشاهد لفهم تفضيلات الجمهور وتحسين إنشاء المحتوى.

10. الخدمات اللوجستية والنقل :Logistics and Transportation

- تحسين الطريق: تحليل بيانات المرور واللوجستيات لتحسين طرق النقل وتقليل أوقات التسليم.
- إدارة الأسطول: استخدام البيانات لمراقبة وإدارة أساطيل المركبات بكفاءة.

11. الموارد البشرية :Human Resources

- اكتساب المواهب والاحتفاظ بها: تحليل بيانات الموظفين لتحسين عمليات التوظيف وإدارة المواهب واستراتيجيات الاحتفاظ بالموظفيين.

12. زراعة :Agriculture

- الزراعة الدقيقة: استخدام البيانات من أجهزة الاستشعار والأقمار الصناعية لتحسين الممارسات الزراعية، وتحسين منتجات المحاصيل، وتقليل استخدام الموارد.

- ▶ هذه مجرد أمثلة قليلة، وتستمر تطبيقات البيانات الضخمة في التوسع مع تقدم التكنولوجيا .
- ▶ القاسم المشترك بين هذه التطبيقات هو استخدام كميات كبيرة من البيانات للحصول على رؤى قابلة للتنفيذ، وتحسين عمليات صنع القرار، ودفع الابتكار عبر مختلف الصناعات

Big Data Lecture

Relational Data Base
قاعدة البيانات العلائقية

Introduction to Data Base

لقد صار مصطلح قاعدة البيانات من أهم مفردات المجتمعات المتقدمة هذه الأيام، وصارت تطبيقات قواعد البيانات جزءاً هاماً من مكونات أي عملية تجارية أو اقتصادية أو سياسية أو تعليمية، بل مع تطور الويب وانتشار تقنيات الانترنت، صارت قواعد البيانات تمثل حجر الأساس في جميع عمليات تبادل المعلومات، ومكون رئيسي في أي نظام معلومات محاسن، مستفيدة من التطور المناظر في تقنيات الاتصال وشبكات الحاسوب، وقبل أن نتساءل عن ماهية قواعد البيانات، سوف نتأمل عملياً تطبيقات الحالية، ونناقش أهم أنواع تطبيقات قواعد البيانات في الواقع.

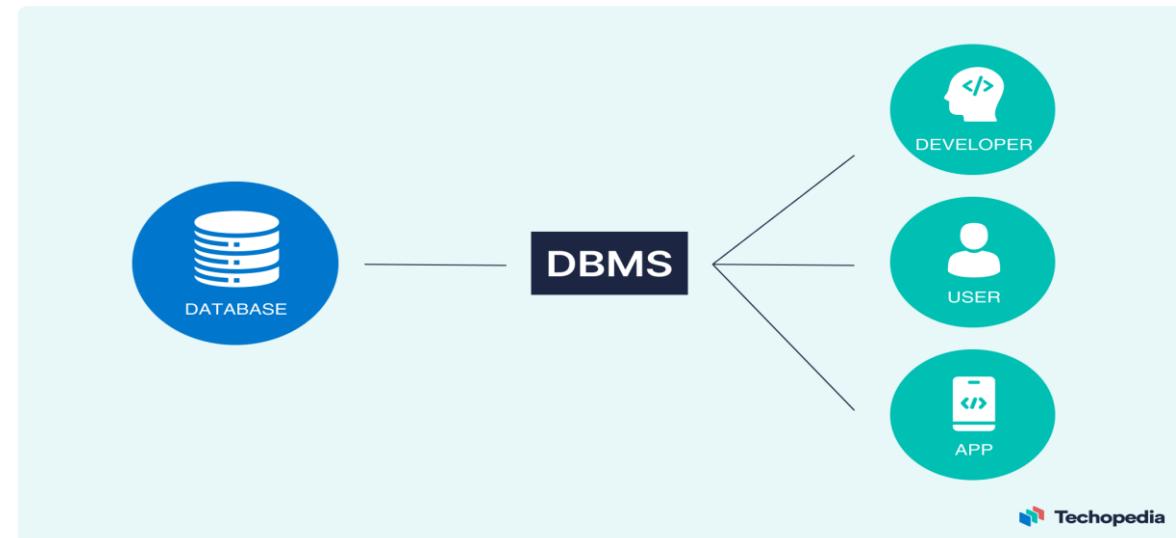
قاعدة البيانات database: هي مجموعة مشتركة من البيانات المترابطة والمتجانسة منطقياً، والتي صممت كي تلبي الاحتياج المستمر للمعلومات لمنظمة معينة.

Database is a Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.

Database Management System

تعريف نظام إدارة قاعدة البيانات

ما هي إدارة نظام قاعدة البيانات DBMS ؟ Database Management Information System) DBMS هي عبارة عن مجموعة البرامج التي تدير وتحكم بعمليه تخزين واسترجاع البيانات، وتتوفر كذلك إمكانية قيام عدد كبير من المستخدمين من الوصول والتعامل مع البيانات، وينظر إليها كذلك على أنها حلقة الوصل بين المستخدمين وقاعدة البيانات، بحيث تقوم باستقبال مطلبات المستخدمين ومن ثم نقلها إلى قاعدة البيانات وتنفيذ البرامج اللازمة لتنفيذ هذه المطلبات ومن ثم تزويد المستخدم بالنتائج المطلوبة.



RDBMS Life Cycle

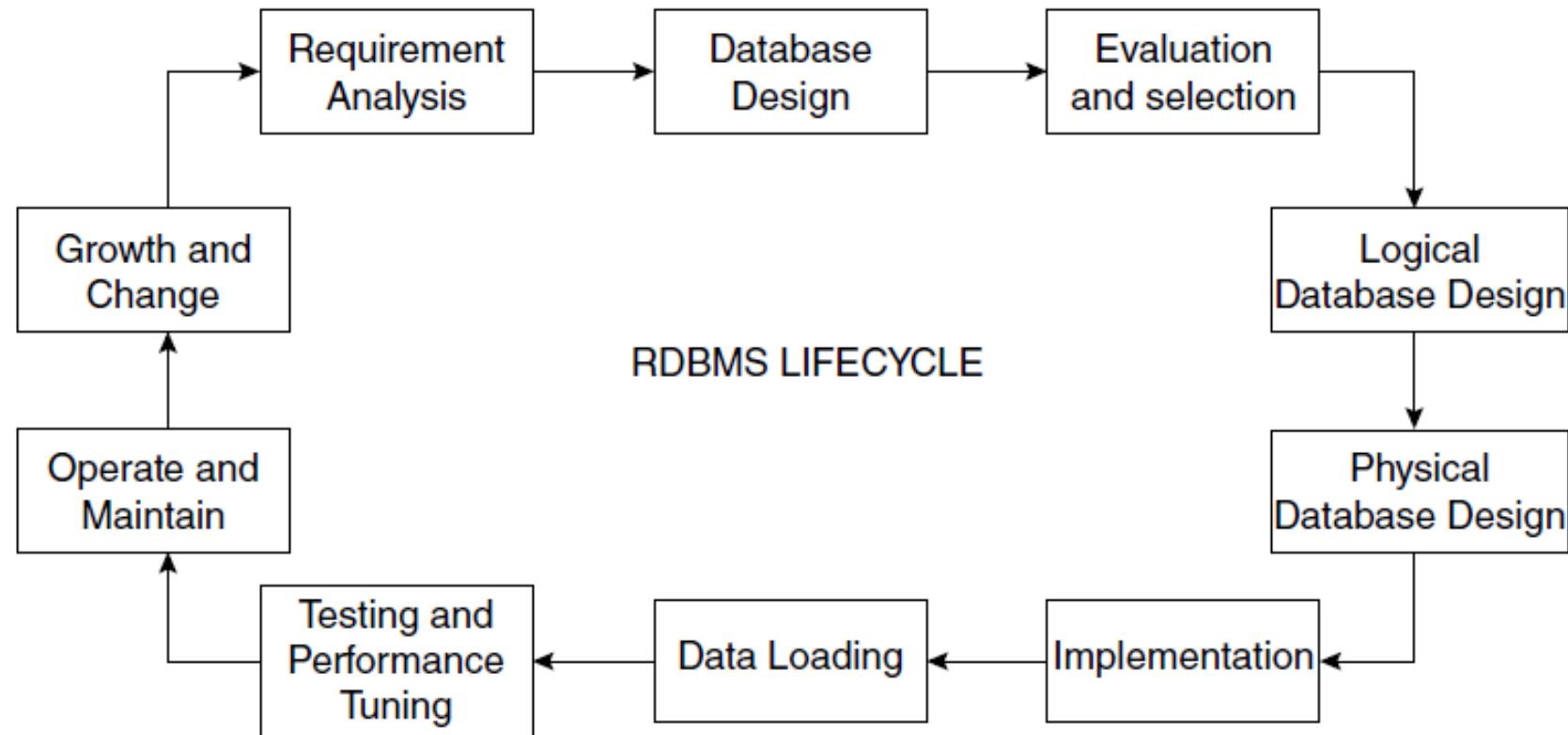


Figure 3.2 RBDMS life cycle.

Components of Database Management System

تشتمل بيئة نظام قاعدة البيانات على مكونات مختلفة تدير قاعدة البيانات وتفاعل معها. تضمن هذه المكونات تخزين البيانات واسترجاعها وإدارتها بكفاءة.

تتضمن المكونات الهامة لبيئة نظام قاعدة البيانات ما يلي:

1. قاعدة البيانات Database: المكون المركزي للبيئة، قاعدة البيانات، حيث يتم تنظيم البيانات وتنظيمها. ويتضمن الجداول والصفوف والأعمدة والعلاقات التي تتبع مخططاً محدداً أو نموذج بيانات.

2. نظام إدارة قواعد البيانات (DBMS): نظام إدارة قواعد البيانات هو البرنامج المسؤول عن إدارة قاعدة البيانات والتحكم فيها. يعمل ك وسيط بين المستخدمين أو التطبيقات وقاعدة البيانات نفسها. يقوم نظام إدارة قواعد البيانات (DBMS) بمهام تخزين البيانات واسترجاعها وأمنها والتحكم في التزامن وإدارة المعاملات. تتضمن أمثلة برامج إدارة قواعد البيانات MySQL و PostgreSQL و Oracle و SQL Server (قواعد بيانات MongoDB).

Components of Database Management System (cont..)

3 . الأجهزة :Hardware

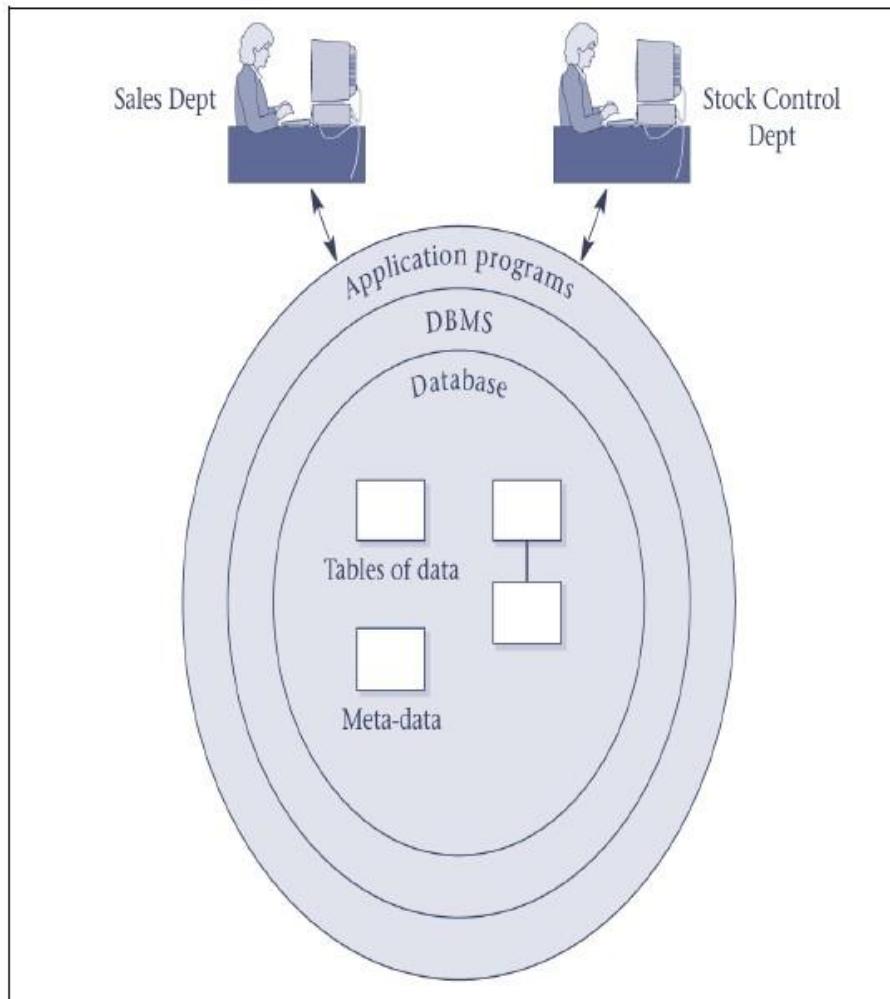
تشتمل مكونات الأجهزة على البنية التحتية المادية التي تدعم نظام قاعدة البيانات. يتضمن ذلك الخوادم وأجهزة التخزين والبنية التحتية للشبكة وموارد الحوسبة. يعد أداء الأجهزة وقابلية تطويرها أمرًا بالغ الأهمية لفاءة نظام قاعدة البيانات.

4 . البرمجيات :Software

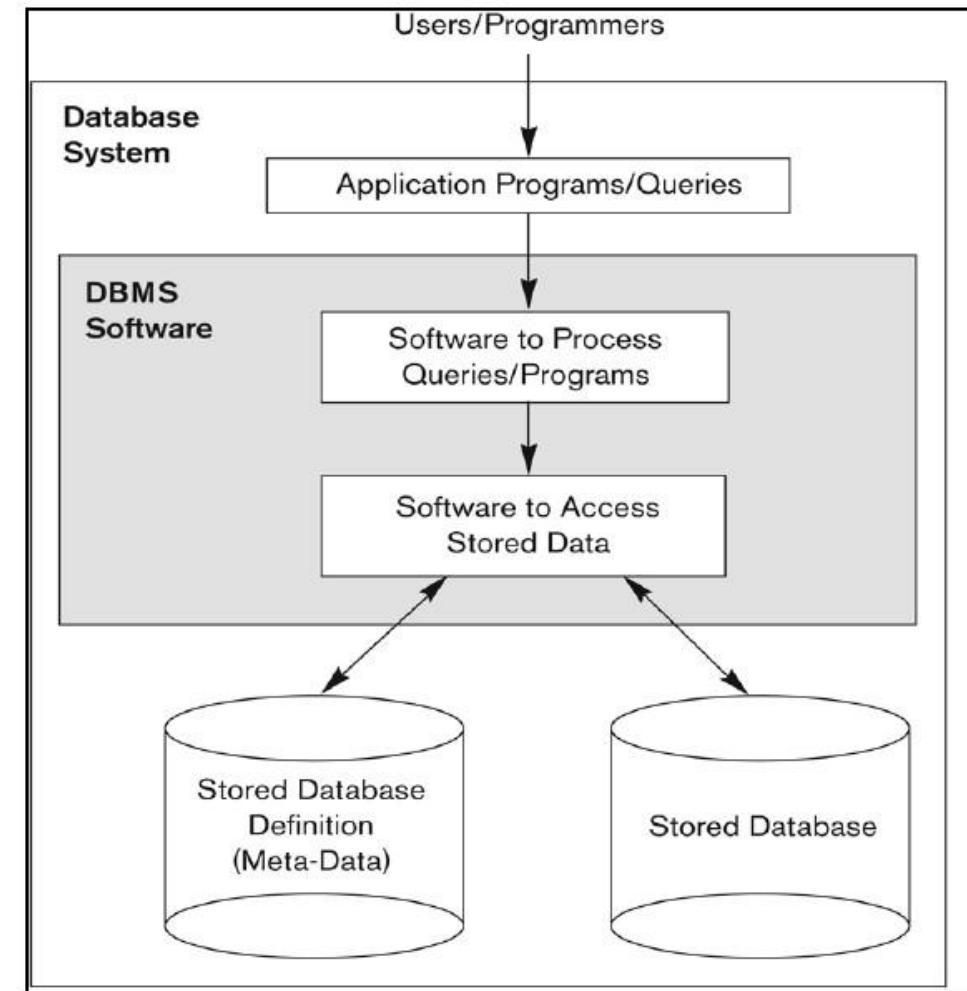
إلى جانب نظام إدارة قواعد البيانات، تعد مكونات البرامج المختلفة جزءاً من بيئة نظام قاعدة البيانات. قد يشمل ذلك نظام التشغيل (OS) الذي يعمل عليه نظام إدارة قواعد البيانات (DBMS) وأي برنامج تطبيقي أو برنامج وسيط يتفاعل مع قاعدة البيانات.

Components of Database Management System (cont..)

- ب -



- ١ -



شكل يوضح طريقتين لعرض نظام قاعدة البيانات (المستخدم، التطبيق البرمجي، الـDBMS ، البيانات)

Components of Database Management System (cont..)

5. المستخدمون Users: يتفاعل المستخدمون مع نظام قاعدة البيانات لأداء مهام مختلفة، مثل الاستعلام عن البيانات، وإدخال سجلات جديدة، وتحديث البيانات الموجودة، وإنشاء التقارير. يمكن تصنيف المستخدمين إلى أدوار مختلفة، وكل منها مستويات مختلفة من الوصول والأذونات.
- مسؤولو قواعد البيانات System administrator (DBA): وهو الشخص المسؤول عن إدارة عمل البيئة العامة التي يعمل بها نظام قاعدة البيانات.
 - مصمم قاعدة البيانات Database designer: الوظيفة الأساسية لمصمم قاعدة البيانات هي تصميم هيكل نظام قاعدة البيانات.
 - مبرمجي و محللي قواعد البيانات Database programmers and analysts
 - المستخدم النهائي Final Database User

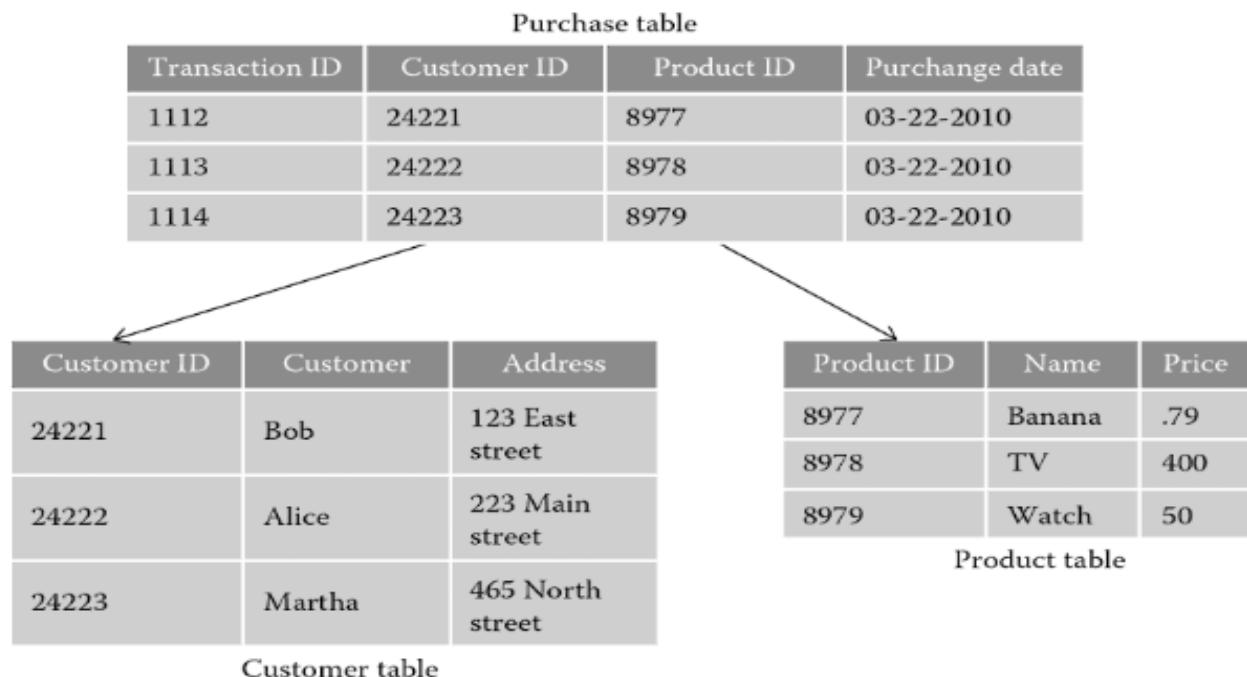
نموذج البيانات Data Model

هو تمثيل مفاهيمي يحدد كيفية هيكلة البيانات وتنظيمها وارتباطها داخل نظام قاعدة البيانات. إنه بمثابة مخطط أو قواعد لتصميم البيانات وتخزينها في قاعدة البيانات. تُستخدم نماذج البيانات للتأكد من تخزين البيانات بكفاءة ودقة وبطريقة تدعم متطلبات المؤسسة أو التطبيق.

هناك عدة أنواع من نماذج البيانات، ولكل منها طرائقها الخاصة في تمثيل البيانات والعلاقات. تشمل أنواع نماذج البيانات الأكثر شيوعاً ما يلي:

1. نموذج البيانات العلاقة Relational Data Model:

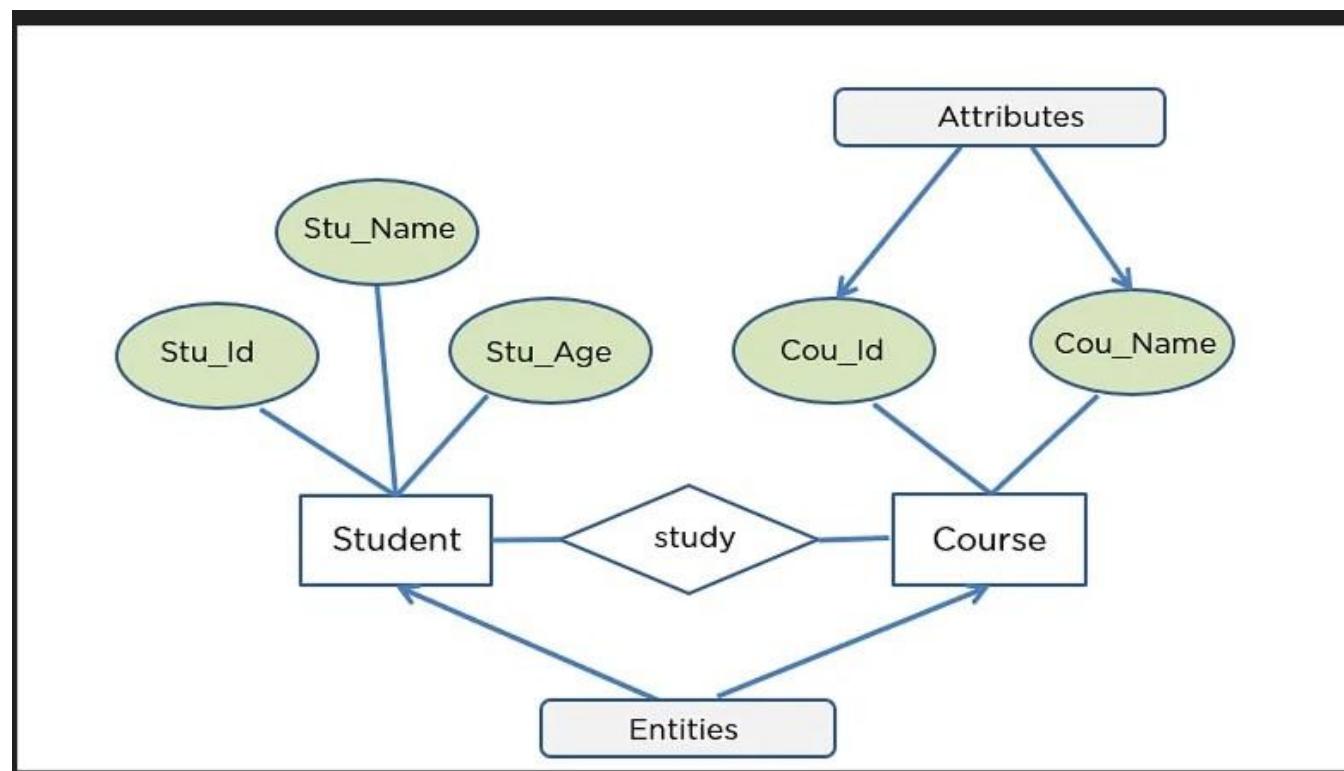
- أحد نماذج البيانات الأكثر استخداماً. وهو يمثل البيانات في شكل جداول، حيث تكون الجداول من صفوف (rows) وأعمدة (columns). يتم تعريف العلاقات بين الجداول باستخدام المفاتيح، مثل المفاتيح الأساسية والمفاتيح الخارجية. تستخدم قواعد البيانات العلاقة SQL (لغة الاستعلام الهيكلية) للاستعلام عن البيانات ومعالجتها.



2. نموذج علاقة الكيان (نموذج ER)

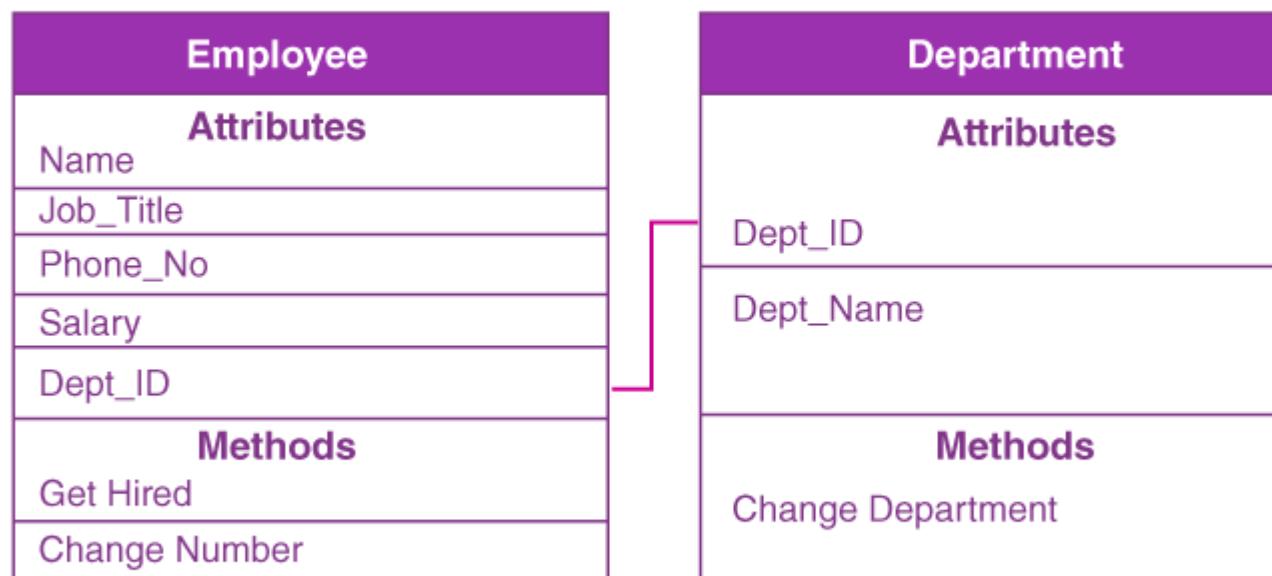
يمثل نموذج ER البيانات باستخدام الكيانات (الكائنات أو المفاهيم) وال العلاقات بينها. تمتلك الكيانات سمات تصف خصائصها. ويتم وصف كيفية ارتباط الكيانات عن طريق تمثيلها بالعلاقات. تُستخدم مخطوطات ER

بشكل شائع لتصور



3. نموذج البيانات كائنية التوجه Object-Oriented Data Model

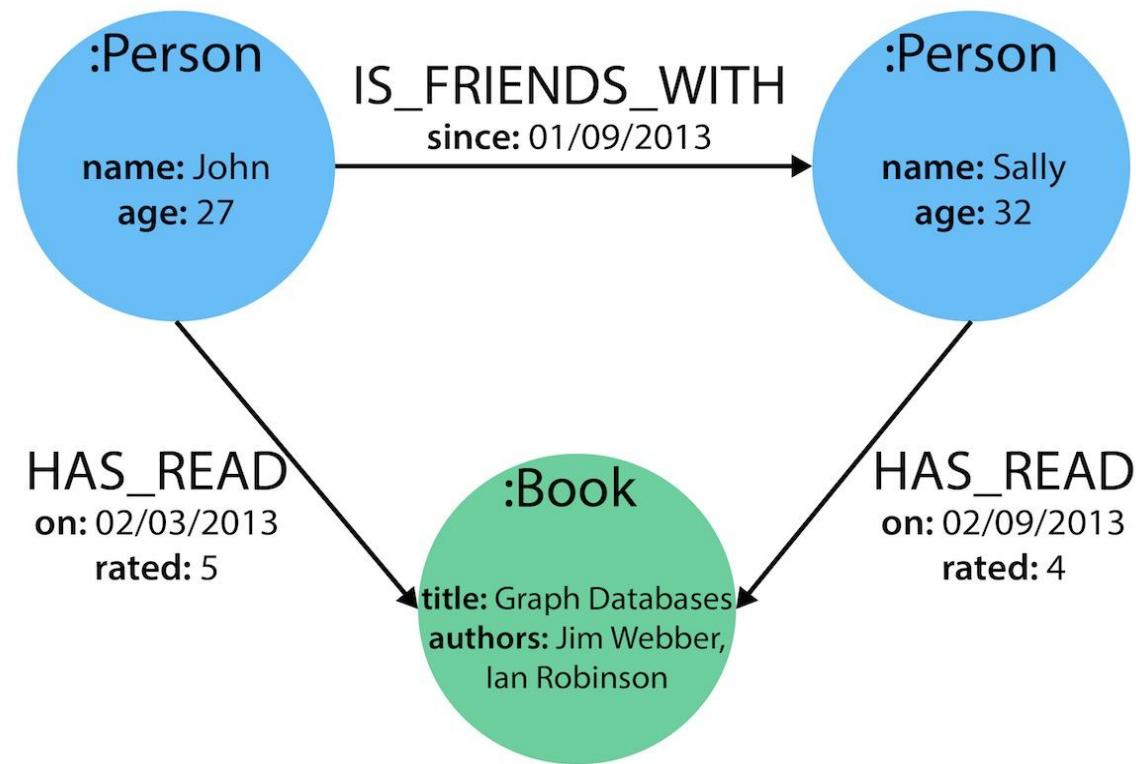
يمثل هذا النموذج البيانات ككائنات، على غرار كيفية تعريف الكائنات في لغات البرمجة كائنية التوجه. تقوم الكائنات بتغليف البيانات والسلوك، مما يجعلها مناسبة لنماذج كيانات العالم الحقيقي المعقدة.



Object Oriented Model

4. نموذج بيانات الرسم البياني :Graph Data Model

يمثل هيكل الرسم البياني البيانات كعقد (بيانات) وحواف (علاقات). يعد هذا النموذج مثالياً للتعامل مع العلاقات المعقدة والبيانات المتصلة.



5. نموذج بيانات السلسلة الزمنية Time-Series Data Model:

يركز هذا النموذج على تخزين البيانات ذات الطابع الزمني والاستعلام عنها بكفاءة، مثل قراءات المستشعر وإدخالات السجل. ويوفر ميزات متخصصة للبيانات المستندة إلى الوقت.

يعتمد اختيار نموذج البيانات على المتطلبات المحددة للمؤسسة أو التطبيق وطبيعة البيانات المخزنة. يحتوي كل نموذج بيانات على نقاط القوة والضعف الخاصة به، ومن المهم تحديد النموذج الذي يتواافق بشكل أفضل مع بنية البيانات واحتياجات التطبيق.

بمجرد تحديد نموذج البيانات، فإنه يكون الأساس لإنشاء نماذج بيانات أكثر تفصيلاً وتقنيّة، مثل نماذج البيانات المنطقية والمادية، التي توفر تفاصيل التنفيذ المحددة اللازمة لتصميم قاعدة البيانات وتطويرها.

تجريد البيانات (Abstraction of Data)

يطلق في علم الحاسوب على عملية تعريف البرامج أو البيانات من خلال إظهار واجهة منسجمة مع معنى وغاية هذا البرنامج ولكن يتم إخفاء التنفيذ الفعلي له . الغرض من هذا هو التخفيف من التعقيدات التي يجب على المبرمج التعامل معها، وبالتالي يكون بإمكانه التركيز على أمور أقل في الوقت الواحد.

يشير تجريد البيانات إلى تبسيط هيكل البيانات المعقدة وتفاعلاتها من خلال تقديم عرض مبسط ورفع المستوى للبيانات . إنه يخفي التفاصيل التقنية الأساسية لتخزين البيانات واسترجاعها ومعالجتها، مما يسهل على المستخدمين والتطبيقات والمطوريين العمل مع قاعدة البيانات . ويعد تجريد البيانات مفهوماً أساسياً في أنظمة إدارة قواعد البيانات (DBMS) وعادةً ما يتم تحقيقه من خلال مستويات مختلفة من التجريد.

ملاحظة : يطلق على اسم تجريد البيانات ببساطة، إنها عملية استخراج البيانات ويمكنك أيضاً القول إن تحديد العناصر المطلوبة يسمى التجريد.

تجريد البيانات (cont..)

وتشمل هذه المستويات:

a. المستوى المادي Physical Level:

هذا هو أدنى مستوى لتجريد البيانات. ويتعامل مع التخزين الفعلي للبيانات على وسائل التخزين، مثل محركات الأقراص الثابتة أو محركات الأقراص ذات الحالة الصلبة. في هذا المستوى، تشمل الاعتبارات تنسيقات تخزين البيانات وتنظيم الملفات والفهرسة وطرق الوصول. يهتم مسؤولو قواعد البيانات ومهندسو الأنظمة في المقام الأول بالمستوى المادي.

تجريد البيانات (cont..)

b. المستوى المنطقي: Logical Level:

يلخص المستوى المنطقي التفاصيل المادية ويركز على بنية البيانات وتنظيمها كما يراها المستخدمون النهائيون ومطورو التطبيقات. وهو يحدد مخطط قاعدة البيانات، بما في ذلك الجداول والعلاقات والقيود ونموذج البيانات المستخدم (على سبيل المثال، العلاقة، الهرمية، أو الموجهة للكائنات). المستوى المنطقي هو حيث يقوم مصممو قواعد البيانات بإنشاء نموذج مفاهيمي يمثل متطلبات بيانات المنظمة دون القلق بشأن كيفية تخزين البيانات فعليًا.

تجريد البيانات (cont..)

c. مستوى العرض: View Level:

يُوفر مستوى العرض طبقة تجريد إضافية تمكن المستخدمين من التفاعل مع قاعدة البيانات باستخدام طرق عرض بيانات مبسطة ومخصصة. طرق العرض هي جداول افتراضية تم إنشاؤها عن طريق تحديد مجموعات فرعية محددة من البيانات، وتطبيق عوامل التصفية، وتحديد الأعمدة محل الاهتمام. يمكن للمستخدمين العمل مع طرق العرض دون فهم المخطط الأساسي المعقد أو بنية قاعدة البيانات. يعد هذا المستوى مفيداً بشكل خاص لأمن البيانات والخصوصية، لأنه يسمح بالوصول المتحكم فيه إلى البيانات.

مثال: تخيل أنك تقوم بتصميم نظام قاعدة بيانات مكتبة يقوم بتخزين معلومات حول الكتب والمؤلفين والمستفيدين واستعارة الكتب. سنستكشف تجريد البيانات على ثلاثة مستويات: المادية والمنطقية والعرضية.

1. المستوى المادي: Physical Level

- الوصف: على المستوى المادي، يتعامل نظام إدارة قاعدة البيانات مع تخزين البيانات والوصول إليها على جهاز تخزين، مثل القرص الصلب.
- إجراءات التجريد: يركز مسؤولو قواعد البيانات على تحسين تخزين البيانات. وهم يقررون كيفية تنظيم ملفات البيانات على القرص، واستخدام هيكل الفهرسة، وتنسيقات التخزين.
- في هذا المستوى، قد يختار المسؤولون تخزين بيانات الكتاب في موقع وتنسيقات ملفات محددة، واستخدام الفهرسة لتسريع عمليات البحث، وتحديد كيفية تخزين البيانات فعليًا على القرص.

2. المستوى المنطقي:

- الوصف: يلخص المستوى المنطقي التفاصيل المادية الأساسية ويحدد مخطط قاعدة البيانات، بما في ذلك الجداول وال العلاقات والقيود.
 - إجراءات التجريد: يقوم مصممو قواعد البيانات بإنشاء نموذج بيانات منطقي بناءً على احتياجات المنظمة. وهي تحدد الجداول والسمات والمفاتيح الأساسية والمفاتيح الخارجية والعلاقات بين الكيانات.
 - مثال: في قاعدة بيانات مكتبتنا، على المستوى المنطقي، يقوم المصممون بإنشاء جداول مثل "الكتب" و "المؤلفون" و "المستفيدون" و "المقرضون". وهي تحدد سمات مثل BookID، العنوان، و AuthorID، BorrowDate، و PatronID، و الاقتراضات.

3. مستوى العرض:

- **الوصف:** يوفر مستوى العرض عرضاً مخصصاً ومبسطاً للبيانات المستخدمين النهائيين والتطبيقات، مما يسمح لهم بالتفاعل مع قاعدة البيانات دون فهم المخطط الكامل.
- **إجراءات التجريد:** يقوم مسؤولو قواعد البيانات أو مطورو التطبيقات بإنشاء طرق عرض بناءً على متطلبات المستخدم. تحدد طرق العرض الأعمدة والصفوف التي تكون مرئية وقد تتضمن عوامل تصفيية وروابط.
- بالنسبة لموظفي المكتبة والمستفيدين الذين يتفاعلون مع كتالوج المكتبة عبر الإنترنت، يمكن إنشاء عرض "BookCatalog". يتضمن هذا العرض المعلومات ذات الصلة فقط، مثل عناوين الكتب وأسماء المؤلفين وحالة التوفير. يمكن للمستخدمين البحث عن الكتب ومعرفة ما إذا كانت متوفرة دون التعامل مع التعقيد الأساسي لمخطط قاعدة البيانات.

Introduction to Transaction Processing

المعاملة: برنامج يتضمن عملية أو أكثر من عمليات الوصول إلى قاعدة البيانات مثل:

عمليات القراءة استرجاع قاعدة البيانات (SQL SELECT)

عمليات الكتابة (التعديل على قاعدة البيانات SQL INSERT، UPDATE، DELETE).

وتعتبر المعاملة: وحدة منطقية لمعالجة قاعدة البيانات.

مثال: تحويل رصيد بنكي بقيمة 100 دولار من حساب جاري إلى حساب توفير في قاعدة بيانات البنك ملحوظة: كل تنفيذ لبرنامج ما ، هو عبارة عن معاملة مميزة distinct transaction بمعاملات مختلفة .

أنظمة معالجة المعاملات: هي أنظمة قواعد بيانات كبيرة متعددة المستخدمين تدعم آلاف المعاملات المتزامنة (عمليات المستخدم) في الدقة. ويوجد لدينا طريقتين للمعالجة وهي كالتالي:

المتزامنة المتداخلة: ويتم التنفيذ المتزامن للعمليات في وحدة معالجة مركزية واحدة.

المعالجة المتوازية: يتم تنفيذ العمليات بشكل متزامن في وحدات المعالجة المركزية المتعددة (الشكل 21.1).

Introduction to Transaction Processing (cont.)

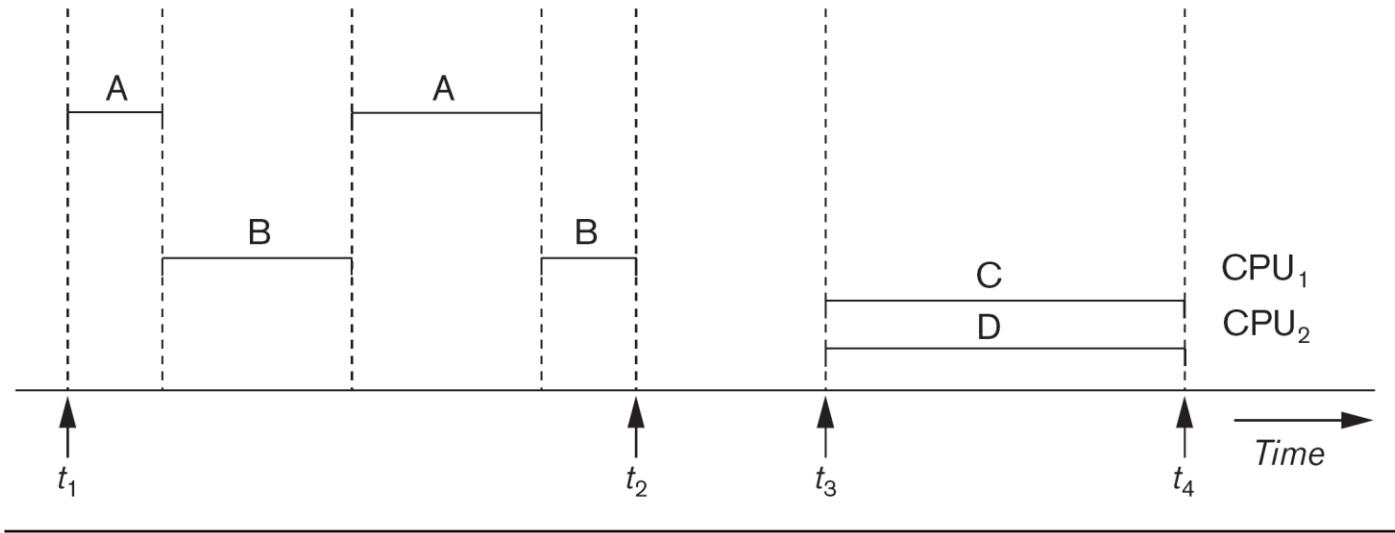


Figure 21.1
Interleaved processing versus parallel processing of concurrent transactions.

مفاهيم المعاملات والنظام Transaction and System Concepts

المعاملة هي وحدة عمل ذرية إما أن تكتمل بالكامل أو لا تتم على الإطلاق. تمر المعاملة عبر عدة حالات (الشكل 21.4، مشابه لحالات العملية في أنظمة التشغيل). حالات المعاملة:

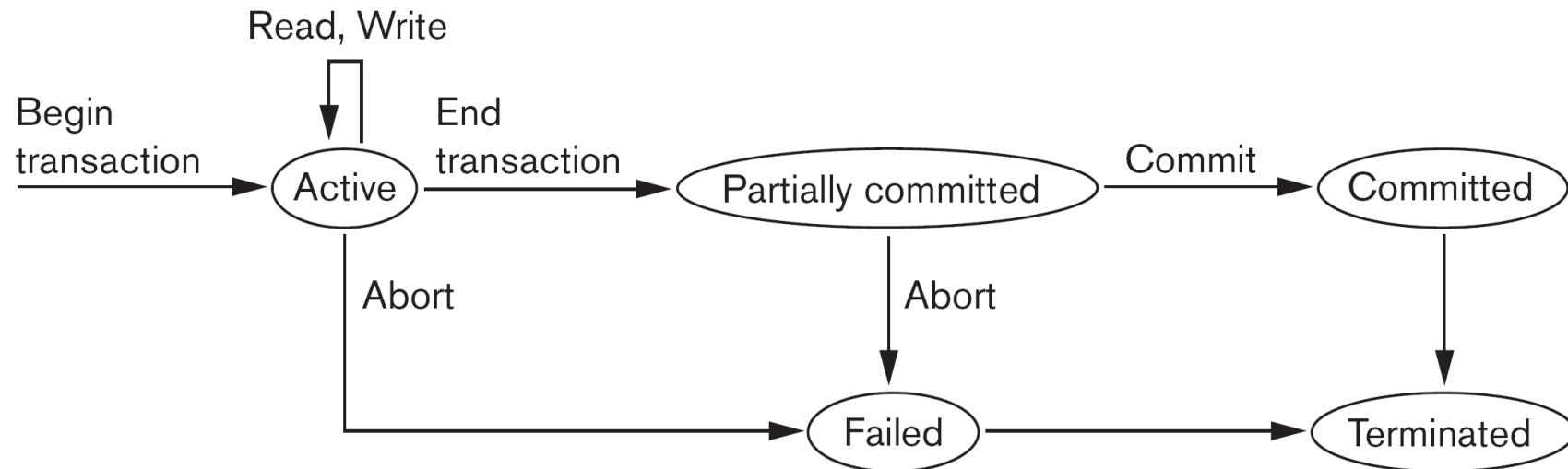
الحالة النشطة Active state (تنفيذ عمليات القراءة والكتابة)
حالة الالتزام جزئيا Partially committed state (منتهية ولكن في انتظار فحوصات النظام لتحديد النجاح أو الفشل)

حالة الالتزام Committed state (نجحت المعاملة)
الحالة الفاشلة Failed state (فشل المعاملة، يجب التراجع عنها)
حالة الإنهاe Terminated State (نظام إجازات المعاملة)

Transaction States in DBMS

Figure 21.4

State transition diagram illustrating the states for transaction execution.



مفاهيم المعاملات والنظام (تابع) Transaction and System Concepts (cont.)

يحتاج DBMS Recovery Manager إلى نظام لتبني العمليات التالية (في ملف سجل النظام):

- بداية تنفيذ المعاملة begin_transaction
- القراءة أو الكتابة Read or write: عمليات القراءة أو الكتابة على عناصر قاعدة البيانات التي يتم تنفيذها كجزء من المعاملة.
- نهاية العملية end_transaction يحدد نهاية عمليات معاملة القراءة والكتابه.
- قد لا يزال يتعين على النظام التحقق مما إذا كان من الممكن تطبيق التغييرات (عمليات الكتابة) التي أدخلتها المعاملة بشكل دائم على قاعدة البيانات (معاملة تثبيت commit transaction) أو ما إذا كان يجب التراجع عن المعاملة (إلغاء المعاملة abort transaction) لأنها تنتهك التحكم في التزامن أو لبعض الأمور

مفاهيم المعاملات والنظام (تابع) Transaction and System Concepts (cont.)

يقوم مدير الاسترداد Recovery Manager ب تتبع العمليات التالية :

الالتزام المعاملة Commit_transaction: يشير إلى نهاية المعاملة بنجاح؛ يمكن الالتزام بأمان بأي تغييرات (عمليات كتابية) يتم تنفيذها بواسطة المعاملة في قاعدة البيانات ولن يتم التراجع عنها.

التراجع Abort_transaction: انتهت معاملة الإشارات دون جدوى؛ يجب التراجع عن أي تغييرات أو تأثيرات قد تكون المعاملة قد طبقتها على قاعدة البيانات.

مفاهيم المعاملات والنظام (تابع) Transaction and System Concepts (cont.)

ملف سجل النظام The System Log File

وهو ملف إلهاقي فقط لتنبيه كافة العمليات لجميع المعاملات بالترتيب الذي حدثت به.

هذه المعلومات مطلوبة أثناء التعافي من حالات الفشل. ويتم الاحتفاظ بالسجل على القرص - ولا يتأثر باستثناء القرص أو الفشل الذريع.

كما هو الحال مع ملفات القرص الأخرى، يتم الاحتفاظ بمخزن مؤقت للذاكرة الرئيسية للاحتفاظ بالسجلات التي يتم إلهاقها حتى يتم إلهاق المخزن المؤقت بالكامل بنهائية ملف السجل على القرص يتم نسخ سجل احتياطياً بشكل دوري إلى مخزن أرشيفي (شريط) للحماية من حالات الفشل الكارثية.

Desirable Properties of Transactions

الخصائص المرغوبة للمعاملات

خصائص المعاملة المرغوبة (خصائص ACID) يجب أن تمتلك المعاملة عدة خصائص، تسمى غالباً خصائص ACID، ليتم تنفيذها في نظام إدارة قواعد البيانات DBMS

1. الذريّة Atomicity

2. تناسق Consistency

3. عزل Isolation

4. متانة Durability

يتم استخدامها لحفظ الاتساق في قاعدة البيانات قبل وبعد المعاملة.

Desirable Properties of Transactions (con...)

الخصائص المرغوبة للمعاملات

الذرية (عدم التجزئة) Atomicity

ينص على أن جميع عمليات المعاملة تتم مرة واحدة، وإذا لم يكن الأمر كذلك، يتم إحباط المعاملات.

لا يوجد منتصف الطريق، أي أن المعاملة لا يمكن أن تحدث جزئيا. يتم التعامل مع كل معاملة كوحدة واحدة ويتطلب تشغيلها حتى الانتهاء أو لا يتم تنفيذها على الإطلاق.

تتضمن الذرية العمليتين التاليتين:

إحباط Rollback : إذا تم إحباط المعاملة، فلن تكون جميع التغييرات التي تم إجراؤها مرئية.

الالتزام commit : إذا تم تنفيذ المعاملة، فستكون جميع التغييرات التي تم إجراؤها مرئية.

Desirable Properties of Transactions (con...)

الخصائص المرغوبة للمعاملات

تناسق Consistency

يتم الحفاظ على قيود التكامل بحيث تكون قاعدة البيانات متسقة قبل المعاملة وبعدها.

سيؤدي تنفيذ المعاملة إلى ترك قاعدة البيانات إما في حالتها المستقرة السابقة أو في حالتها المستقرة الجديدة.

تنص الخاصية المتسقة لقاعدة البيانات على أن كل معاملة ترى مثيلاً ثابتاً لقاعدة البيانات.

يتم استخدام المعاملة لتحويل قاعدة البيانات من حالة متسقة إلى حالة متسقة أخرى.

Desirable Properties of Transactions (con...)

الخصائص المرغوبة للمعاملات

عزل Isolation

يوضح أن البيانات المستخدمة في وقت تنفيذ المعاملة لا يمكن استخدامها في المعاملة الثانية حتى اكتمال المعاملة الأولى.

بشكل منفصل، إذا تم تنفيذ المعاملة T_1 واستخدام عنصر البيانات X ، فلا يمكن الوصول إلى عنصر البيانات هذا بواسطة أي معاملة أخرى T_2 حتى تنتهي المعاملة T_1 .

قام النظام الفرعي للتحكم في التزامن لنظام إدارة قواعد البيانات (DBMS) بفرض خاصية العزل.

Desirable Properties of Transactions (con...)

الخصائص المرغوبة للمعاملات

متانة Durability

يتم استخدام خاصية المتانة للإشارة إلى أداء الحالة المتسقة لقاعدة البيانات. تنص على أن المعاملة أجرت التغييرات الدائمة.

ولا يمكن فقدانها بسبب التشغيل الخاطئ لمعاملة خاطئة أو بسبب فشل النظام. عند اكتمال المعاملة، تصل قاعدة البيانات إلى حالة تعرف بالحالة المتسقة. ولا يمكن فقدان هذه الحالة المتسقة، حتى في حالة فشل النظام.

يتحمل نظام الاسترداد الفرعي لنظام إدارة قواعد البيانات مسؤولية خاصية المتانة.

Beyond Relational Data Base (**Not Only SQL**)

Relational Database Disadvantages

عيوب قاعدة البيانات العلاقة

على الرغم من وجود فوائد أكثر لاستخدام قواعد البيانات العلاقة، إلا أن لها بعض القيود. دعونا نرى القيود أو العيوب في استخدام قاعدة البيانات العلاقة.

- عدم قابلية التوسيع Lack of Scalability

أثناء استخدام قاعدة البيانات العلاقة عبر خوادم متعددة، يتغير هيكلها ويصبح من الصعب التعامل معها، خاصة عندما تكون كمية البيانات كبيرة. ونتيجة لذلك، فإن البيانات غير قابلة للتوسيع على خوادم التخزين الفعلية المختلفة. في النهاية، يتأثر أدائها، أي عدم توفر البيانات ووقت التحميل وما إلى ذلك. ومع زيادة حجم قاعدة البيانات أو توزيعها بشكل أكبر مع عدد أكبر من الخوادم، سيكون لذلك آثار سلبية مثل مشكلات زمن الوصول والتوفير التي تؤثر على الأداء العام.

Relational Database Disadvantages

عيوب قاعدة البيانات العلائقية

- مشكلة الصيانة Maintenance Problem

تصبح صيانة قاعدة البيانات العلائقية صعبة بمرور الوقت بسبب الزيادة في البيانات. يتبعن على المطورين والمبرمجين قضاء الكثير من الوقت في صيانة قاعدة البيانات.

- التكلفة Cost

يعد إعداد وصيانة نظام قاعدة البيانات العلائقية مكلفاً. يمكن أن تكون التكلفة الأولية للبرنامج وحده باهظة الثمن بالنسبة للشركات الصغيرة، ولكنها تزداد سوءاً عندما تأخذ في الاعتبار تعين فني محترف يجب أن يكون لديه أيضاً خبرة في هذا النوع المحدد من البرامج.

Relational Database Disadvantages

عيوب قاعدة البيانات العلائقية

- التعقيد في الهيكل (تأثير أداء العلاقات) Complexity in Structure

يمكن لقواعد البيانات العلائقية تخزين البيانات فقط في شكل جدول مما يجعل من الصعب تمثيل العلاقات المعقدة بين الكائنات. هذه مشكلة لأن العديد من التطبيقات تتطلب أكثر من جدول واحد لتخزين كافة البيانات الضرورية التي يتطلبها منطق التطبيق الخاص بها.

- انخفاض الأداء مع مرور الوقت Decrease in performance over time

يمكن أن تصبح قاعدة البيانات العلائقية أبطأ، ليس فقط بسبب اعتمادها على جداول متعددة. عندما يكون هناك عدد كبير من الجداول والبيانات في النظام، فإن ذلك يؤدي إلى زيادة التعقيد. يمكن أن يؤدي ذلك إلى بطء أوقات الاستجابة للاستعلامات أو حتى الفشل الكامل لها، اعتماداً على عدد الأشخاص الذين قاموا بتسجيل الدخول إلى الخادم في وقت

Relational Database Disadvantages

عيوب قاعدة البيانات العلائقية

• التعقيد في البيئات الموزعة Complexity in Distributed Environments

قواعد البيانات العلائقية التقليدية ليست دائمًا مناسبة تمامًا للبيئات الموزعة أو السحابية، حيث قد تنتشر البيانات عبر مواقع أو عقد متعددة.

يمكن أن تكون إدارة الاتساق consistency وتحمل الأقسام والتوفير في بيئة موزعة أمرًا صعبًا، مما يؤدي إلى ظهور قواعد بيانات بديلة مصممة لمثل هذه البيئات.

Relational Database Disadvantages

عيوب قاعدة البيانات العلائقية

- صعوبة التعامل مع البيانات الضخمة Difficulty in Handling Big Data: قد تواجه قواعد البيانات العلائقية تحديات في التعامل بكفاءة مع مجموعات البيانات الكبيرة جدًا، خاصة عند مقارنتها بقواعد بيانات NoSQL الموزعة والمصممة خصيصًا للبيانات الضخمة. يمكن أن تحدث اختلافات في الأداء، وقد يلزم إعادة النظر في التصميم المعماري من أجل قابلية التوسيع والسرعة.
- أدت الزيادة الهائلة في حجم البيانات، والتي تفاص بالتيرابايت والبيتا بايت، إلى تحدي نظام RDBMS للتعامل مع مثل هذا الحجم الهائل من البيانات.
- لمعالجة هذه المشكلة، قام نظام RDBMS بزيادة عدد المعالجات وإضافة المزيد من وحدات الذاكرة، مما أدى إلى زيادة التكلفة.
- كما يقرب من 80% من البيانات التي تم جلبها كانت شبه منظمة وغير منتظمة، وهو الأمر الذي لم يتمكن RDBMS من

Relational Database Disadvantages

عيوب قاعدة البيانات العلاقة

Table 1.1 Differences In The Attributes Of Big Data And RDBMS.

Attributes	Rdbms	Big Data
Data Volume	Gigabytes to Terabytes	Petabytes to Zettabytes
Organization	Centralized	Distributed
Data Type	Structured	Structured, Unstructured And Semi-structured
Hardware Type	High-end Model	Commodity Hardware
Updates	Read/Write Many Times	Write Once, Read Many Times
Schema	Static	Dynamic

Not Only SQL Databases

قواعد بيانات NoSQL

هي نوع من أنواع أنظمة إدارة قواعد البيانات التي توفر آلية لتخزين mechanism for storing data واسترجاع retrieving وإدارة البيانات managing data المصممة بطرق أخرى غير نموذج قاعدة البيانات العلائقية التقليدية. يشير المصطلح "ليس فقط SQL إلى" NoSQL ، مع التركيز على أن قواعد البيانات هذه قد تدعم نماذج البيانات بما يتجاوز الهياكل الجدولية المستخدمة في قواعد البيانات العلائقية.

NoSQL هي تقنية تمثل فئة من المنتجات التي لا تتبع مبادئ RDBMS وغالباً ما ترتبط بتخزين واسترجاع كميات هائلة من البيانات. يجدون تطبيقاتهم في البيانات الضخمة وتطبيقات الويب الأخرى في الوقت الفعلي. تعد قابلية التوسيع الأفقي والمخطط المرن والموثوقية والتسامح مع الأخطاء بعضًا من ميزات قواعد بيانات NoSQL ويتم تنظيم قواعد بيانات NoSQL بإحدى الطرق التالية:

NoSQL Databases

قواعد بیانات NoSQL

Why NoSQL?



CAP Theorem

نظرية CAP

تعتبر نظرية CAP، المعروفة أيضًا باسم نظرية بروير، مبدأً أساسياً في الحوسبة الموزعة التي تنص على المقاييس والقيود الخاصة بتصميم ونشر أنظمة موزعة موثوقة وعالية الأداء. تؤكد النظرية أنه من المستحيل على نظام تخزين البيانات الموزع أن يلبي في الوقت نفسه الجوانب الثلاثة الحاسمة المتمثلة في الاتساق والتوافر وتحمل القسم. بعبارات أبسط، تسلط نظرية CAP الضوء على مقايضة متأصلة في النظام، حيث يمكن تحقيق اثنين فقط من هذه السمات الأساسية الثلاثة بشكل فعال في أي وقت محدد.

CAP is the acronym for **consistency, availability, and partition tolerance**, formulated by Eric Brewer.

CAP Theorem

نظرية CAP

الاتساق : Consistency

يشير الاتساق إلى فكرة أن جميع العقد داخل النظام الموزع تعرض نفس البيانات والقيم بالضبط في أي حالة معينة. وهذا يعني أنه عند حدوث معاملة بيانات، تكون جميع العقد داخل النظام على علم بالتغيير، وبالتالي، فإن طلب الوصول إلى هذه البيانات سيؤدي إلى نفس النتيجة عبر النظام، مما يضمن سلامة البيانات.

CAP Theorem

نظرية CAP

التوفر : Availability

هو مقياس لمدى تكرار استجابة النظام الموزع لطلبات المستخدم بنجاح. يمكن للنظام المتوفّر للغاية أن يضمن أنه سيعترف بسرعة وبدقّة بكل طلب مستخدم من خلال أي من العقد الخاصة به، بغض النظر عن حجم الطلبات أو الحالة الحالية للعقد الفردي.

تسامح القسم Partition Tolerance

يتعلّق تسامح القسم بقدرة النظام على تحمل انقطاع الاتصال أو الأعطال داخل الشبكة. يمكن للنظام

التسامح مع الأقسام أن يستمر في الأداء بأفضل حالاته حتى في حالة حدوث انقطاع كامل في الاتصال بين

CAP Theorem

نظريّة CAP

في ضوء نظرية CAP، يصبح من الواضح أنه عند تصميم الأنظمة الموزعة، من المهم تحقيق التوازن بين هذه السمات الثلاث، حيث أن تحقيقها جمیعاً في وقت واحد أمر بعيد المنال. من الناحية العملية، يدور هذا بشكل عام حول اختيار حل وسط يتضمن التضحية بسمة واحدة لصالح السمتين الآخرين بناءً على أولويات النظام وحالات الاستخدام ومتطلبات العمل.

وفي سياق تطوير الواجهة الخلفية، تلعب نظرية CAP دوراً حاسماً في تشكيل التصميم المعماري وعملية صنع القرار لمختلف أنظمة البيانات الموزعة. تساعد هذه النظرية المهندسين على تقييم إيجابيات وسلبيات

CAP Theorem

نظرية CAP

فمن بين قواعد البيانات الموزعة المختلفة، تعطي الأنظمة الشائعة مثل Apache و Amazon DynamoDB

الأولوية للجمع بين التوفر والتسامح مع التقسيم AP، بينما تركز قواعد البيانات الأخرى مثل Cassandra

على الاتساق والتسامح مع التقسيم CP . أو أنظمة RDBMS مثل PostgreSQL على Google Cloud Spanner

كما سيتم توضيجه في الشكل التالي :

CAP Theorem

نظرية CAP

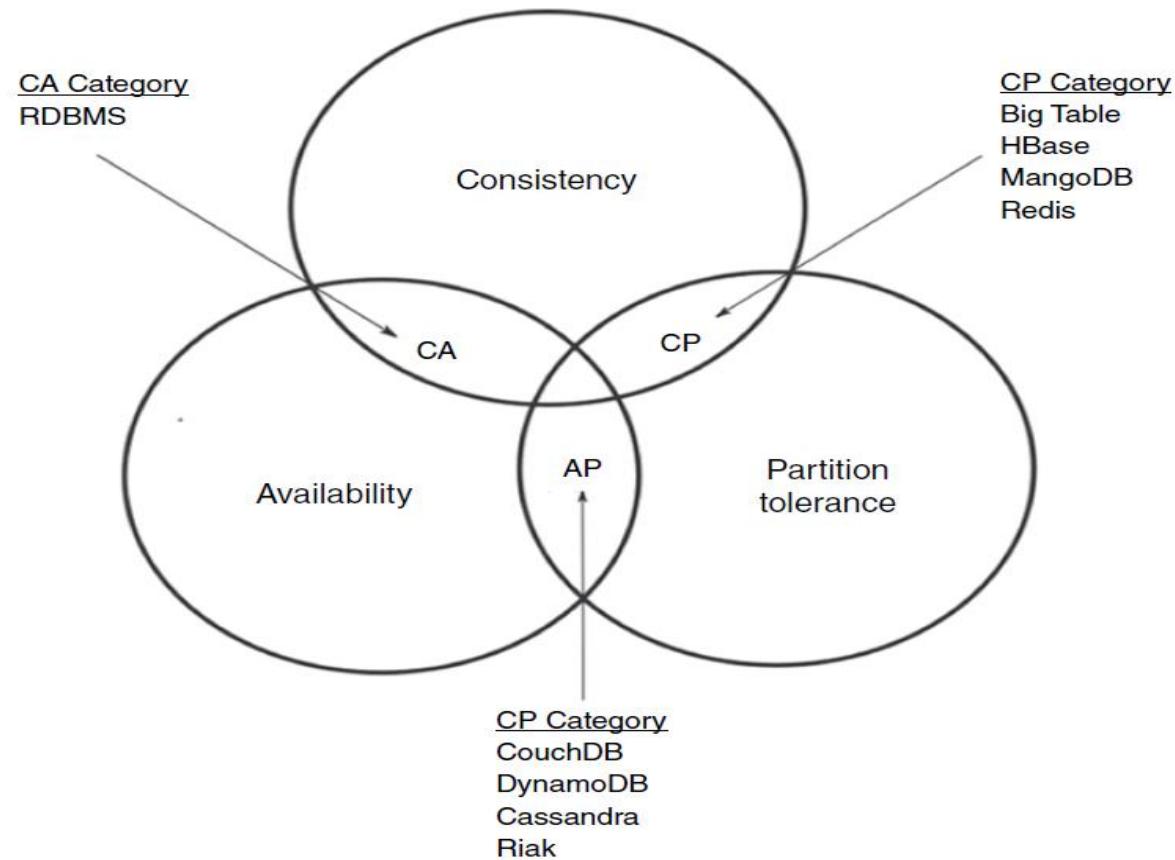


Figure 3.1 Properties of a system following CAP theorem.

BASE properties خصائص BASE

هو اختصار لمجموعة من الخصائص المتعلقة بتصميم قاعدة البيانات بناءً على نظرية CAP . وهذه

الخصائص هي كالتالي:

. 1. متحدة بشكل أساسي Basically Available

. 2. حالة ناعمة Soft State

. 3. ومتسقة في النهاية Eventually Consistent

حيث تحمل قواعد بيانات NoSQL خصائص BASE او تعمل وفق هذه الخصائص.

BASE properties خصائص BASE

متاح بشكل أساسي (BA)

تؤكد هذه الخاصية على أن النظام يظل جاهزاً للعمل وسريع الاستجابة، مما يوفر التوفير الأساسي على الرغم من حالات الفشل. ويعني ذلك أن النظام قد لا يضمن دائمًا الاتساق ولكنه يركز على البقاء متاحاً لعمليات القراءة والكتابة، حتى في حالة وجود أخطاء أو أقسام الشبكة. ويقال أن قاعدة البيانات متاحة بشكل أساسي إذا كان النظام متاحاً دائماً على الرغم من فشل الشبكة.

BASE properties خصائص BASE

الحالة الناعمة (S):Soft State (S)

نماذج الاتساق الصارمة الموجودة في قواعد بيانات ACID، والتي تحافظ على حالة جامدة وثابتة، تسمح قواعد البيانات BASE بمزيد من المرونة في حالة البيانات. تعرف الحالة الناعمة بأن حالة النظام قد تكون عابرة ويمكن أن تتتطور بناءً على عوامل مختلفة.

الحالة الناعمة تعني أن عقد قاعدة البيانات قد تكون غير متسقة عند تنفيذ عملية القراءة. على سبيل المثال، إذا قام مستخدم بتحديث سجل في العقدة A قبل تحديث العقدة B، التي تحتوي على نسخة من البيانات في العقدة A، وإذا طلب المستخدم

BASE properties خصائص BASE

متسقة في نهاية المطاف **(EC)**

في نهاية المطاف، يشير الثبات إلى أنه، مع مرور فترة طويلة بما فيه الكفاية دون تحديثات أو تعارضات، فإن جميع النسخ المتماثلة في النظام الموزع سوف تتقرب إلى نفس الحالة المتسقة. بمعنى آخر، في حين أن النظام قد لا يضمن الاتساق الفوري، فإنه يضمن أنه مع مرور الوقت، سيتم التوفيق بين جميع النسخ المتماثلة وتحقيق حالة متسقة. يسمح هذا النهج بقابلية التوسيع والتوافر بشكل أفضل في الأنظمة الموزعة.

BASE properties خصائص BASE

الحالة التي تتبع الحالة الناعمة هي الاتساق النهائي. يقال إن قاعدة البيانات قد حفظت الاتساق بمجرد تحدث التغييرات في البيانات على جميع العقد. ينص التناسق النهائي على أن عملية القراءة التي يقوم بها المستخدم متبوعة مباشرة بعملية الكتابة قد تؤدي إلى إرجاع بيانات غير متناسقة. على سبيل المثال، إذا قام مستخدم بتحديث سجل في العقدة A، وطلب مستخدم آخر قراءة نفس السجل من العقدة B قبل تحديث السجل، فستكون البيانات الناتجة غير متناسقة؛ ومع ذلك، بعد تحقيق الاتساق في نهاية المطاف، يحصل المستخدم على القيمة الصحيحة.

Schemeless Databases

قواعد بيانات بلا مخططات

قواعد البيانات غير المخططة هي تلك التي لا تتطلب أي مخطط صارم لتخزين البيانات . يمكنهم تخزين البيانات بأي تنسيق، سواء كانت منظمة أو غير منتظمة . عندما يتعين تخزين البيانات في RDBMS، يجب تصميم المخطط أولاً .

المخطط عبارة عن بنية محددة مسبقاً لقاعدة بيانات توفر تفاصيل حول الجداول والأعمدة الموجودة في الجداول وأنواع البيانات التي يمكن أن يحتوي عليها كل عمود. قبل أن يتم تخزين البيانات في قاعدة البيانات هذه، يجب تحديد المخطط لها . باستخدام قاعدة بيانات المخطط، يجب معرفة نوع البيانات التي يجب تخزينها مسبقاً، بينما مع قاعدة بيانات بدون مخطط، يكون من السهل تخزين أي نوع من البيانات دون معرفة مسبقة بالبيانات؛ كما أنه يسمح بتخزين البيانات حيث يحتوي كل سجل على حقول محددة مختلفة . سيؤدي تخزين هذا النوع من البيانات في قاعدة بيانات المخطط إلى جعل الجدول في حالة فوضى

Schemeless Databases (con..) قواعد بيانات بلا مخططات

قاعدة بيانات NoSQL هي قاعدة بيانات بدون مخطط، حيث يكون تخزين البيانات أسهل بكثير مقارنة بقاعدة البيانات التقليدية. يسمح نوع القيمة الرئيسية لقاعدة بيانات NoSQL للمستخدم بتخزين أي بيانات تحت المفتاح. لا تضع قاعدة البيانات الموجهة نحو المستندات أي قيود على البنية الداخلية للمستند المراد تخزينه. تسمح قاعدة بيانات مخزن الأعمدة للمستخدم بتخزين أي بيانات ضمن عمود. تتيح قاعدة بيانات الرسم البياني للمستخدم إضافة حواف وخصائص إليها دون أي قيود.

NoSQL Databases

قواعد بيانات NoSQL

قاعدة بيانات NoSQL هي قاعدة بيانات غير علائقية مصممة لتخزين واسترجاع البيانات شبه المنظمة وغير المنظمة. وقد تم تصميمها للتغلب على مشكلات قابلية التوسيع في البيانات الكبيرة والأداء، والتي لم يتم تصميم قواعد البيانات التقليدية لمعالجتها. يتم استخدامه على وجه التحديد عندما تحتاج المؤسسات إلى الوصول إلى حجم كبير من البيانات غير المنظمة ومعالجتها وتحليلها. على عكس أنظمة قواعد البيانات التقليدية، التي تنظم البيانات في الجداول، تنظم قاعدة بيانات NoSQL البيانات في أزواج من المفاتيح/القيمة أو الصفوف. كما يوحي الاسم، لا تدعم قاعدة بيانات NoSQL SQL فحسب، بل تدعم أيضًا لغات الاستعلام الأخرى، وهي HQL للاستعلام عن البيانات المنظمة، و XQuery للاستعلام عن ملفات XML، و SPARQL للاستعلام عن بيانات RDF، وما إلى ذلك.

أشهر تطبيقات قاعدة بيانات NoSQL هي Google Bigtable و SimpleDB و Cassandra.

NoSQL vs RDMS

RDBMS	NoSQL
Structured data with a rigid schema. بيانات منظمة مع مخطط صارم.	Structured, Unstructured, and Semi-structured data with a flexible schema. البيانات المنظمة وغير المنظمة وشبه المنظمة مع مخطط مرن.
Extract, Transform, Load (ETL) required. مطلوب استخراج وتحويل وتحميل .	ETL is not required. غير مطلوب.
Storage in rows and columns. التخزين في الصفوف والأعمدة.	Data are stored in a Key/Value pairs database, Columnar database, Document database, and Graph Database. يتم تخزين البيانات في قاعدة بيانات أزواج المفاتيح/القيمة، وقاعدة البيانات العمودية، وقاعدة بيانات المستندات، وقاعدة بيانات الرسم البياني.
RDBMS is based on ACID transactions. ACID stands for Atomic, Consistent, Isolated and Durable. يعتمد RDBMS على معاملات ACID. يرمز ACID إلى ذري ومتسلق ومعزول ومتين.	NoSQL is based on BASE transactions. BASE stands for Basically available, Soft state, Eventual consistency. يعتمد NoSQL على المعاملات الأساسية. يرمز BASE إلى الحالة الناعمة المتاحة بشكل أساسى والاتساق النهائي.
RDBMS Scale up when the data load increases, i.e., expensive servers are brought to handle the additional load. يتم توسيع نطاق نظام RDBMS عند زيادة تحميل البيانات، أي أنه يتم إحضار خوادم باهظة الثمن للتعامل مع الحمل الإضافي.	NoSQL is highly scalable at a low cost. NoSQL scales out to meet the extra load. i.e., low-cost commodity servers are distributed across the cluster. قابلة للتطوير بشكل كبير وبتكلفة منخفضة. يتسع NoSQL لتلبية الحمل الإضافي. أي أن خوادم السلع منخفضة التكلفة يتم توزيعها عبر المجموعة.

NoSQL vs RDMS

RDBMS	NoSQL
SQL server, Oracle, and MySQL are some of the examples. خادم SQL، Oracle، و MySQL هي بعض الأمثلة.	MongoDB, HBase, Cassandra are some of the examples. تعد Cassandra و HBase و MongoDB بعض الأمثلة.
Structured Query Language is used to query the data stored in the data warehouse. يتم استخدام لغة الاستعلام المنظمة للاستعلام عن البيانات المخزنة في مستودع البيانات.	Hive Query Language (HQL) is used to query the data stored in HDFS. يتم استخدام لغة استعلام الخلية HQL للاستعلام عن البيانات المخزنة في HDFS.
Matured and stable. Matured indicates that it has been in existence for a number of years. ناضجة ومستقرة. يشير النضج إلى أنه موجود منذ عدة سنوات.	Flexible Incubation. Incubation indicates that it is in existence from the recent past. حداثة مرنة. وتشير الحداثة إلى أنها موجودة منذ الماضي القريب.
RDBMS vs. NoSQL databases	

Features of NoSQL Databases

قواعد بيانات NoSQL مميزات

- **— : Schema less**

قاعدة بيانات NoSQL هي قاعدة بيانات بدون مخطط حيث يكون تخزين البيانات أسهل بكثير مقارنة بقاعدة البيانات التقليدية. نظرًا لأن قواعد بيانات SQL تحتوي على مخطط صارم، يجب القيام بالكثير من العمل مقدماً قبل تخزين البيانات في قاعدة البيانات، بينما في قاعدة بيانات NoSQL، التي لا تحتوي على مخطط، يمكن تخزين البيانات دون معرفة سابقة بالمخطط.

Features of NoSQL Databases

مميزات قواعد بيانات NoSQL

- قابلية التوسيع الأفقي : **Horizontal scalability**

على عكس قواعد بيانات SQL، التي تتمتع بقابلية التوسيع الرأسي، تتمتع قواعد بيانات NoSQL بقابلية التوسيع الأفقي. لديهم القدرة على النمو ديناميكياً مع المتطلبات المتغيرة بسرعة. يتم تنفيذ قابلية التوسيع الأفقي من خلال التجزئة والنسخ المتماثل حيث تتم مشاركة ملفات قاعدة البيانات عبر خوادم متعددة، ويتم نسخ الملفات لجعل النظام متسامحاً مع الأخطاء في حالة الصيانة المخطط لها أو الأحداث أو انقطاع الخدمة. يدعم NoSQL المشاركة اليدوية والتلقائية. كما تدعم قواعد بيانات NoSQL النسخ التلقائي عبر مواقع جغرافية متعددة لمقاومة حالات الفشل الإقليمية. يتم تخزينها دون معرفة سابقة بالمخطط.

Features of NoSQL Databases

مميزات قواعد بيانات NoSQL

• الحوسبة الموزعة - Distributed computing

تسمح الحوسبة الموزعة بتخزين البيانات في أكثر من جهاز واحد، مما يزيد من الموثوقية. يتم تقسيم مجموعة بيانات كبيرة واحدة وتخزينها عبر عقد متعددة، ويمكن معالجة البيانات بالتوازي.

• تكلفة أقل -Lower cost

تستخدم قواعد بيانات SQL خوادم موثوقة وعالية الأداء نظرًا لأنها قابلة للتطوير رأسياً، بينما يمكن لقواعد بيانات NoSQL العمل على أجهزة سلعية منخفضة التكلفة ومنخفضة الأداء نظرًا لأنها قابلة للتطوير أفقياً. إنها تسمح بإضافة خوادم رخيصة لتلبية الطلب المتزايد على أجهزة التخزين والمعالجة.

Features of NoSQL Databases

مميزات قواعد بيانات NoSQL

• غير علائقية –Non-Relational

تم تصميم قواعد البيانات العلائقية للتعرف على كيفية ارتباط الجداول المخزنة بعضها البعض. على سبيل المثال، في تجارة التجزئة عبر الإنترنت، يرتبط صف منتج واحد بالعديد من صفوف العملاء. وبالمثل، يمكن أن يرتبط كل صف عميل بصفوف منتجات متعددة. يتم التخلص من مفهوم العلاقة هذا في قاعدة بيانات غير علائقية. هنا، كل منتج يتضمن العميل معه. وهذا يعني أن العميل يتم تكراره مع كل صف منتج يستخدمه. سيتطلب القيام بذلك مساحة إضافية ولكنه يتميز بسهولة التخزين والاسترجاع.

Features of NoSQL Databases

قواعد بيانات NoSQL مميزات

- تتعامل مع حجم كبير من البيانات - Handles large a volume of data

قواعد البيانات العلائقية قادرة على التعامل مع الجداول التي تحتوي على ملايين السجلات، والتي كانت تعتبر بيانات ضخمة في الماضي. لكن اليوم، في العالم الرقمي، أصبحت هذه البيانات ضئيلة، وقد نمت الجداول إلى مليارات وتريليونات من الصفوف. أيضًا، كان نظام RDBMS مقتصرًا على التعامل فقط مع البيانات التي تتناسب مع بنية الجدول. قواعد بيانات NoSQL قادرة على التعامل مع هذا النمو الهائل في البيانات بشكل أكثر كفاءة. إنهم قادرون على التعامل مع حجم هائل من البيانات المنظمة وغير المنظمة وشبه المنظمة.

NoSQL Databases Type

قواعد بيانات NoSQL

1. أزواج القيمة الرئيسية

Key-value Pairs

2. قواعد البيانات الموجهة نحو المستندات

Document-oriented Databases

3. قواعد بيانات الرسوم البيانية

Graph Databases

4. قواعد البيانات الموجهة نحو الأعمدة

Column-oriented Databases

Key-Value Store Databases

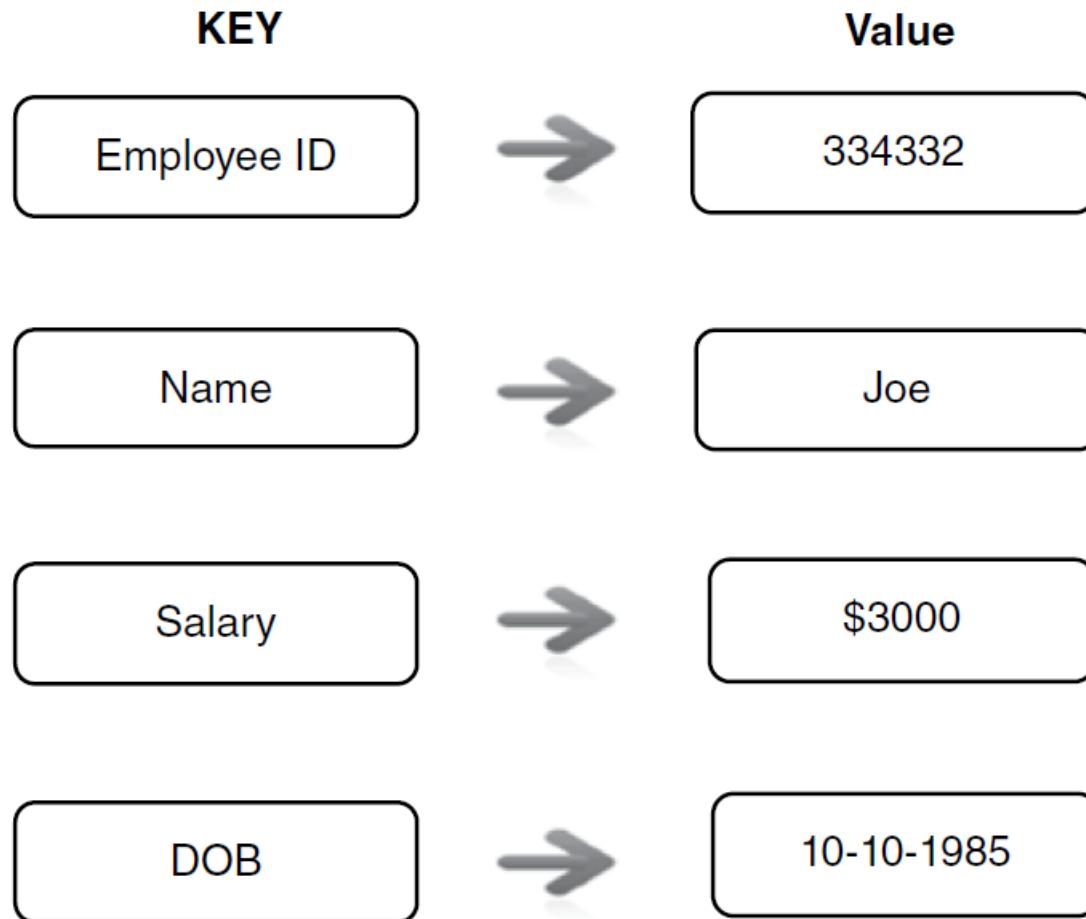
قاعدة بيانات مخزن القيمة

تعد قاعدة بيانات مخزن القيمة الرئيسية أبسط قاعدة بيانات وأكثرها كفاءة والتي يمكن تنفيذها بسهولة. يسمح للمستخدم بتخزين البيانات في أزواج قيمة المفتاح دون أي مخطط. يتم عادةً تقسيم البيانات إلى قسمين: المفتاح والقيمة. المفتاح عبارة عن سلسلة، والقيمة هي البيانات الفعلية؛ ومن ثم زوج القيمة الرئيسية المرجعية. يشبه تنفيذ قاعدة بيانات القيمة الأساسية جداول التجزئة. يتم استرجاع البيانات باستخدام المفتاح كالفهرس. لا توجد مفاتيح بديلة أو مفاتيح خارجية، كما في حالة RDBMS، وهي أسرع بكثير من RDBMS. يتضمن التطبيق العملي لمخازن البيانات ذات القيمة الأساسية عربة تسوق عبر الإنترنت وتخزين معلومات الجلسة في الألعاب عبر الإنترنت، في حالة الألعاب متعددة اللاعبين. يوضح الشكل 3.4 قاعدة بيانات القيمة الرئيسية حيث يكون معرف الموظف والاسم والراتب وتاريخ الميلاد هو المفتاح، والبيانات المقابلة له هي القيمة. Amazon DynamoDB هي قاعدة بيانات NoSQL تم إصدارها بواسطة Amazon. قواعد البيانات ذات القيمة الرئيسية الأخرى هي Riak، Redis، و Berkeley DB، و Memcached، و Hamster DB. يتم إنشاء كل قاعدة بيانات للتعامل مع التحديات الجديدة، ويتم استخدام كل منها لحل التحديات المختلفة.

Key-Value Store Databases

قاعدة بيانات مخزن القيمة

Figure 3.4 A key-value store database.



Column-Store Database

قاعدة البيانات الموجهة نحو الأعمدة

تقوم قاعدة البيانات الموجهة نحو الأعمدة ب تخزين البيانات كأعمدة بدلاً من تخزينها كصفوف. للحصول على فهم أفضل، تتم هنا مقارنة قاعدة بيانات الأعمدة بقاعدة البيانات الموجهة نحو الصف وشرح كيف يؤدي مجرد اختلاف في التخطيط الفعلي لنفس البيانات إلى تحسين الأداء.

تتمثل طريقة عمل قاعدة بيانات مخزن الأعمدة في حفظ البيانات في أقسام من الأعمدة بدلاً من أقسام من الصفوف. يعتمد اختيار كيفية تخزين البيانات، سواء كانت موجهة نحو الصنوف أو موجهة نحو الأعمدة، على احتياجات استرجاع البيانات. تقوم بيانات OLTP (online transaction processing) معالجة المعاملات عبر الإنترنت باسترداد عدد أقل من الصفوف والمزيد من الأعمدة، وبالتالي فإن قاعدة البيانات الموجهة نحو الصنوف مناسبة. يقوم OLAP المعالجة التحليلية عبر الإنترنت باسترداد عدد أقل من الأعمدة والمزيد من الصفوف، وبالتالي فإن قاعدة البيانات الموجهة نحو الأعمدة مناسبة.

Column-Store Database

قاعدة البيانات الموجهة نحو الأعمدة

Employee_Id	Name	Salary	City	Pin_code
3623	Tony	\$6000	Huntsville	35801
3636	Sam	\$5000	Anchorage	99501
3967	Williams	\$3000	Phoenix	85001
3987	Andrews	\$2000	Little Rock	72201

Row store database

Employee_Id	Name	Salary	City	Pincode
3623	Tony	\$6000	Huntsville	35801
3636	Sam	\$5000	Anchorage	99501

Column store database

Document-Oriented Database

قاعدة البيانات الموجهة للمستندات

قاعدة البيانات الموجهة للمستندات قابلة للتطوير أفقياً. عندما يزيد حمل البيانات، تتم إضافة المزيد من الأجهزة السلعية لتوزيع حمل البيانات. تم تصميم قاعدة البيانات هذه من خلال اعتماد مفهوم الوثيقة. تقوم المستندات بتغليف البيانات بتنسيق XML أو JSON أو YAML أو بتنسيق ثلائى MS Word أو PDF في قاعدة البيانات الموجهة نحو المستندات، سيتم التعامل مع المستند بأكمله كوحدة واحدة كاملة.

يمكن الاستعلام عن البيانات المخزنة في قاعدة البيانات الموجهة للمستندات باستخدام أي مفتاح بدلاً من الاستعلام باستخدام المفتاح الأساسي فقط. على غرار RDBMS، تسمح قواعد البيانات الموجهة للمستندات بإنشاء البيانات وقراءتها وتحديثها وحذفها. من أمثلة قواعد البيانات الموجهة للمستندات Microsoft Document DB وCouchDB وMongoDB.

Document-Oriented Database

قاعدة البيانات الموجهة للمستندات

KEY-VALUE DATABASE

```
{  
    "FirstName": "Sam",  
    "LastName": "Andrews",  
    "Age": 28,  
    "Address":  
    {  
        "StreetAddress": "3 Fifth Avenue",  
        "City": "New York",  
        "State": "NY"  
        "PostalCode": "10118-3299"  
    }  
}
```

1298

KEY



VALUE

DOCUMENT DATABASE

```
{  
    "ID": "1298",  
    "FirstName": "Sam",  
    "LastName": "Andrews",  
    "Age": 28,  
    "Address":  
    {  
        "StreetAddress": "3 Fifth Avenue",  
        "City": "New York",  
        "State": "NY",  
        "PostalCode": "10118-3299"  
    }  
}
```

Graph-Oriented Database

قاعدة بيانات موجهة للرسم البياني

قاعدة بيانات موجهة للرسم البياني: تقوم قاعدة البيانات الموجهة نحو الرسم البياني ب تخزين الكيانات، المعروفة أيضًا باسم العقد وال العلاقات بينها. كل عقدة لها خصائص، وال العلاقات بين العقد تعرف بالحواف. العلاقات لها خصائص وأهمية اتجاهية.

يتم استخدام خصائص العلاقات للاستعلام عن قاعدة بيانات الرسم البياني. قد تمثل هذه الخصائص المسافة بين العقد، أو على سبيل المثال، قد تمثل العلاقة بين شركة وموظفي الشخص، وهي عدد سنوات خبرة الموظف، ودور الموظف، وما إلى ذلك.

Graph-Oriented Database

قاعدة بيانات موجهة للرسم البياني

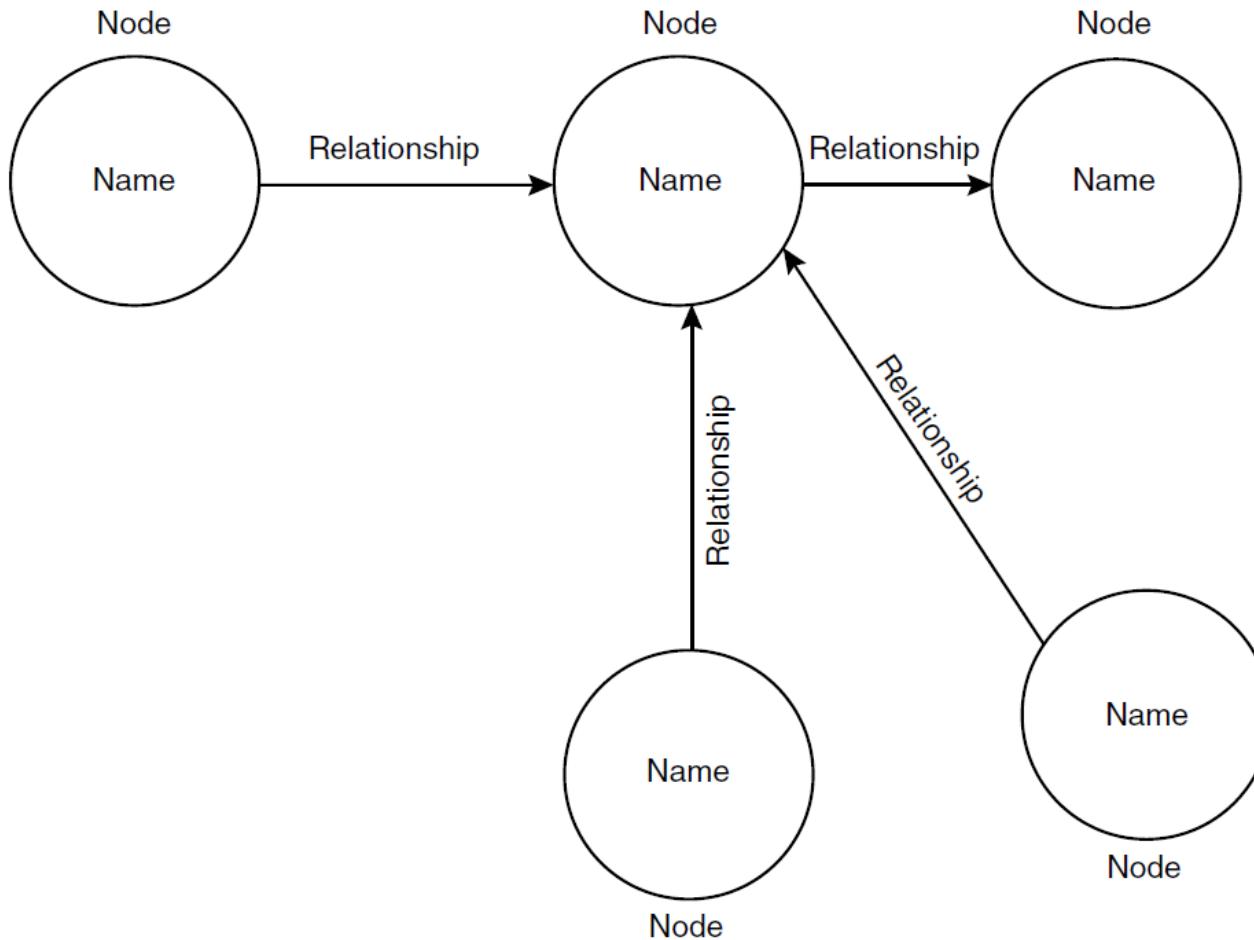


Figure 3.5 General representation of graph database.

NoSQL Operations

عمليات الـ NoSQL

Create database

Drop Database

Create collection

Drop collection

Insert Document

Update Document

Delete Document

Query Document

Can we migrating from RDBMS to NoSQL??



Key-Value Store Databases

قاعدة بيانات مخزن القيمة

1. **Amazon DynamoDB**
2. **Microsoft Azure Table Storage**

1. **Apache Cassandra**
2. **MongoDB,**
3. **CouchDB,**
4. **Microsoft Document DB.**

Different types of graph databases are

Neo4J,

InfiniteGraph,

HyperGraphDB, AllegroGraph, GraphBase, and orientDB. Figure 3.5 represents a graph database.