IT-234 – database concepts

UNIT 10 – NON-RELATIONAL DATABASE MODELS

overview

Tables are very useful to store data.

Humans understand tables; they just make sense.

But is a table always the best way to store data?

That would depend on the data.

overview

In this unit, you will explore some data that does not lend itself to the relational model.

You will learn about other databases that organize the data in a different way.

The relational model is great for transactional data.

overview

Customers, orders, suppliers, and venders are all related to one another and fit well in the relational model.

However, you can use the database in other ways.

overview

- While the transaction database is working, accepting sales and creating orders, you can use an online analytical processing database to store the data for reporting.

overview

After completing this unit, you should be able to:

- Explain when to use the relational database model.
- Describe alternative database systems.

relational databases

They work with structured data.

Relationships in the system have constraints, which promotes a high level of data integrity.

There are limitless indexing capabilities, which results in faster query response times.

They are excellent at keeping data transactions secure.

relational databases

They provide the ability to write complex SQL queries for data analysis and reporting.

Their models can ensure and enforce business rules at the data layer adding a level of data integrity not found in a non-relational database.

They use SQL (structured query language) for shaping and manipulating data, which is very powerful.

relational databases

They are table and row oriented.

SQL databases are best fit for heavy duty transactional type applications.

SQL database examples: MySQL, Oracle, PostgreSQL, IBM DB2, Microsoft Access, and Microsoft SQL Server.

relational databases

Pros

- Works with structured data
- Supports strict atomicity, consistency, isolation, durability (ACID) transactional consistency
- Supports joins
- Built-in data integrity
- Large eco-system

relational databases

Pros

Relationships via constraints	
Limitless indexing	
Strong SQL	
OLTP and OLAP	
Most off-the-shelf applications run on RDBMS	

relational databases

Cons

- Does not scale out horizontally (concurrency and data size) – only vertically, unless use sharding
- Data is normalized, meaning lots of joins, affecting speed
- Difficulty in working with semi-structured data
- Schema-on-write
- Cost

Non-relational databases

The non-relational database, or NoSQL database, stores data.

However, unlike the relational database, there are no tables, rows, primary keys or foreign keys.

Instead, the non-relational database uses a storage model optimized for specific requirements of the type of data being stored.

Non-relational databases

Some of the more popular NoSQL databases are MongoDB, Apache Cassandra, Redis, Couchbase and Apache HBase.

There are four popular non-relational types: document data store, column-oriented database, key-value store, and graph database.

Often combinations of these types are used for a single application.

Key-value stores

This is the least complicated of the NoSQL databases and, as the name would indicate, the key-value store is simply a collection of key-value pairs contained within an object.

Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

MAC table

Key	Value
10.94.214.172	3c:22:fb:86:c1:b1
10.94.214.173	00:0a:95:9d:68:16
10.94.214.174	3c:1b:fb:45:c4:b1

An Example of Key-value database

A document data store manages a set of named string fields and object data values in an entity referred to as a "document" typically stored in the form of JSON documents, which can be encoded in a variety of ways, including XML, YAML, JSON, BSON or as plain text.

The fields within documents are exposed, allowing an application to query and filter data using field values.

Document data stores

The document stores do not require all documents to maintain identical data structures, which provides a great deal of flexibility.

It’s easy to see then how this flexibility can be leveraged as an organization’s requirements change.

Document data stores

Document Data stores

- Document stores are a bit more complex than key-value stores.
- They don't assume a particular document structure specified with a schema.
- The document store is designed to store everyday documents as is, and they allow for complicated querying.
- MongoDB and CouchDB are both examples of document stores.

Document data stores

XML Document Example

```xml
<artist>
  <artistname>Iron Maiden</<artistname>
  <albums>
    <album>
      <albumname  lbumname>
      <datereleased>2015</datereleased>
      <genre>Hard Rock</genre>
    </album>
    <album>
      <albumname>Killers</albumname>
      <datereleased>1981</datereleased>
      <genre>Hard Rock</genre>
    </album>
    <album>
      <albumname>Powerslave</albumname>
      <datereleased>1984</datereleased>
      <genre>Hard Rock</genre>
    </album>
    <album>
      <albumname>Somewhere in Time</albumname>
      <datereleased>1986</datereleased>
      <genre>Hard Rock</genre>
    </album>
  </albums>
</artist>
```

JSON Document Example

```json
{
    '_id' : 1,
    'artistName' : { 'Iron Maiden' },
    'albums' : [
        {
            'albumname' : 'The Book of Souls',
            'datereleased' : 2015,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Killers',
            'datereleased' : 1981,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Powerslave',
            'datereleased' : 1984,
            'genre' : 'Hard Rock'
        }, {
            'albumname' : 'Somewhere in Time',
            'datereleased' : 1986,
            'genre' : 'Hard Rock'
        }
    ]
}
```

Columnar data stores

A columnar data store organizes data into columns, which is conceptually similar to the relational database.

The true advantage of a column-family database is in its denormalized approach to structuring sparse data, which comes from its column-oriented approach to storing data.

Columnar data stores

Columnar databases are typically used to store large amounts of data for analysis, such as in Business Intelligence systems.

By storing all the records for one field together, columnar databases can query and perform analysis on that similar data far quicker than row-based databases.

Columnar data stores

As an example, if you want to know the mean order total for all of your customers, a columnar database would only need to look at the order total column to pull the data, and can quickly calculate the mean.

Performing the same operation in a row-based database might require scanning millions or billions of rows to gather all the values.

Example - Relational Database Representation

ID Number	Name	Address	Telephone
111	Bob Smith	21 Main Street	555-1212
112	Jane Doe	14 First Avenue	444-2222

Example – Equivalent Columnar Data Store Representation

ID Number	111, 112
Name	Bob Smith, Jane Doe
Address	22 Main Street, 14 First Avenue
Telephone	555-1212, 444-2222

Columnar data stores

Most complex non-relational database type

It's designed to efficiently store relations between entities.

When data is greatly interconnected, such as purchasing and manufacturing systems or referencing catalogs, graph databases are a good solution.

Graph databases

Graph databases

- The possibilities for graph NoSQL databases are infinite, and with the data we collect becoming increasingly interconnected, graph databases are going to continue to gain in popularity, including the still-dominant relational database.

Graph databases

Instead of the Structure Query Language (SQL) used by relational databases, the NoSQL database uses Object-Relational-Mapping (ORM).

The concept of ORM is the ability to write queries using your preferred programming language.

Some of the more popular ORMs are Java, Javascript, .NET and PHP.

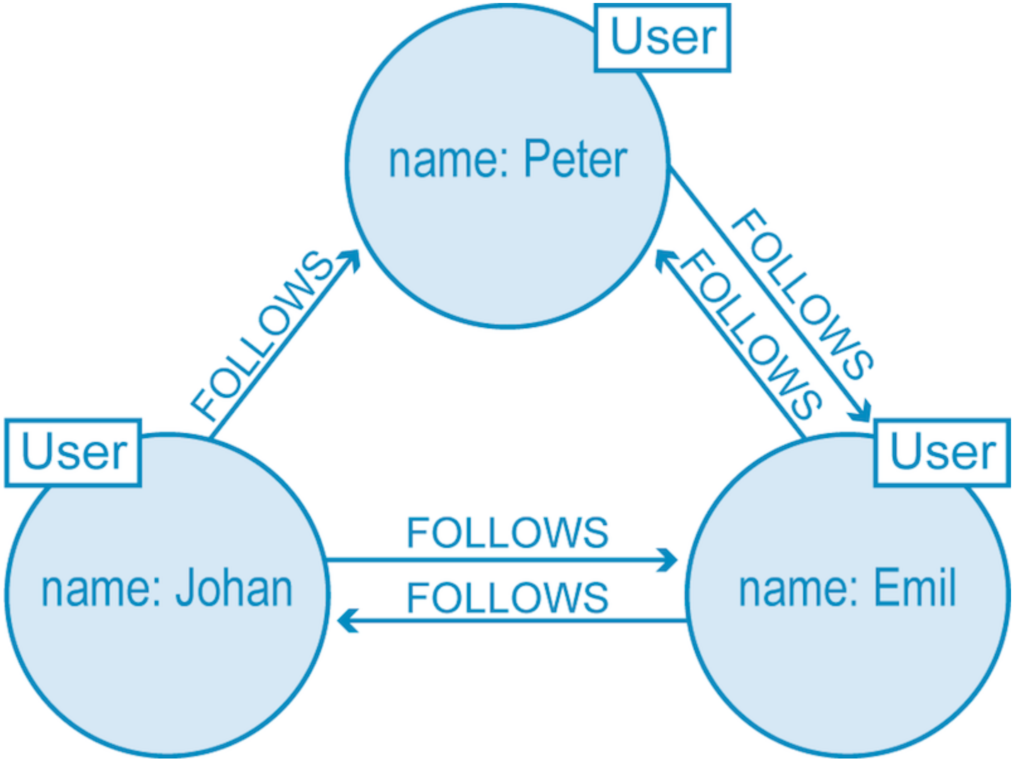
Graph databases

In the illustration on the next slide, we have a small slice of Twitter users represented in a graph database.

Each node (labeled **User**) belongs to a single person and is connected with relationships describing how each user is connected.

Graph databases

- As seen below, Peter and Emil follow each other, as do Emil and Johan, but although Johan follows Peter, Peter hasn't (yet) reciprocated.



NON-relational databases

Pros

- Capable of handling big data.
- As it is schema-less and table free, it offers a high level of flexibility with data models.
- It is a low-cost database and the open-source NoSQL databases provide very affordable solutions to small enterprises.

NON-relational databases

Pros

- Easier and low-cost scalability.
  - You don’t need to increase the hardware for scaling.
  - You just need to add more servers to the pool as NoSQL is schema-free and built on distributed systems.
- Detailed database modeling is not required here.
  - Hence it saves time and effort.

NON-relational databases

Cons

- The benefits of NoSQL come at the cost of relaxing ACID properties.
  - NoSQL offers only eventual consistency.
- Relatively less community support.
- Lacks standardization, unlike SQL, which in turn creates some issues during migration.
- Inter-operability is also a concern in the case of NoSQL databases.

Big data

There are some things that are so big that they have implications for everyone, whether we want it or not.

Big Data is one of those things and is completely transforming the way we do business and is impacting most other parts of our lives.

Big data - implications

Big data - Activity Data

Simple activities like listening to music or reading a book are now generating data

Digital music players and eBooks collect data on our activities

Your smart phone collects data on how you use it, and your web browser collects information on what you are searching for

Big data - Activity Data

Your credit card company collects data on where you shop, and your shop collects data on what you buy

It is hard to imagine any activity that does not generate data.

Big data - CONVERSATION Data

Our conversations are now digitally recorded

It all started with emails but nowadays most of our conversations leave a digital trail

Just think of all the conversations we have on social media sites like Facebook or Twitter

Even many of our phone conversations are now digitally recorded.

Big data - Photo and Video Image Data

Just think about all the pictures we take on our smart phones or digital cameras

We upload and share 100s of thousands of them on social media sites every second

The increasing amounts of CCTV cameras take video images and we up-load hundreds of hours of video images to YouTube and other sites every minute

Big data - Sensor Data

We are increasingly surrounded by sensors that collect and share data

Take your smart phone

- It contains a global positioning sensor to track exactly where you are every second of the day
- It includes an accelerometer to track the speed and direction at which you are travelling

We now have sensors in many devices and products

Big data - The Internet of Things Data

We now have a multitude of devices that are able to collect and process data

- Smart watches
- Smart fridges
- Smart alarms
- Smart TVs

The Internet of Things, or Internet of Everything connects these devices

- Example - The traffic sensors on the road send data to your alarm clock which will wake you up earlier than planned because the blocked road means you have to leave earlier to make your 9 AM meeting…

Big data – The Four v's

With the datafication comes big data, which is often described using the four V's:

- Volume
- Velocity
- Variety
- Veracity

Big data – volume

Refers to the vast amounts of data generated every second

We are not talking Terabytes (10^{12}) but Zettabytes (10^{21}), Yottabytes (10^{24}), or Brontobytes (10^{27})

1 ZB = 1,000,000,000,000 TB	1 YB = 1,000,000,000,000,000 TB	1 BB = 1,000,000,000,000,000,000 TB

Big data – volume

If we take all the data generated in the world between the beginning of time and 2000, the same amount of data will soon be generated every minute

New big data tools use distributed systems so that we can store and analyze data across databases that are dotted around anywhere in the world

Big data – volume

- The amount of data matters
- With Big Data, you'll have to process high volumes of low-density, unstructured data
- This can be data of unknown value, such as Twitter data feeds, clickstreams on a webpage or a mobile app, or sensor-enabled equipment
- For some organizations, this might be tens of terabytes of data
- For others, it may be hundreds of petabytes

Big data – velocity

Refers to the speed at which new data is generated and the speed at which data moves around

Velocity is the fast rate at which data is received and (perhaps) acted on

Normally, the highest velocity of data streams directly into memory versus being written to secondary storage (e.g., hard drive, solid-state disk, etc.)

Big data – velocity

Some internet-enabled smart products operate in real time or near real time and will require real-time evaluation and action

Just think of social media messages going viral in seconds

Technology allows us now to analyze the data while it is being generated (sometimes referred to as in-memory analytics), without ever putting it into databases

Big data – Variety

Refers to the many types of data that are available

Traditional data types were structured and fit neatly in a relational database

With the rise of Big Data, data comes in new unstructured data types

Unstructured and semi-structured data types, such as text, audio, and video require additional preprocessing to derive meaning and support metadata

Big data – Variety

In the past we only focused on structured data that neatly fitted into tables or relational databases, such as financial data

In fact, 80% of the world's data is unstructured (text, images, video, voice, etc.)

With Big Data technology we can now analyze and bring together data of different types such as messages, social media conversations, photos, sensor data, video or voice recordings

Big data – Veracity

Refers to the messiness or trustworthiness of the data

The accuracy of an analysis depends on the quality of the captured data

Data has intrinsic value.

But it's of no use until that value is discovered.

Equally important: How truthful is your data — and how much can you rely on it?

Big data – Veracity

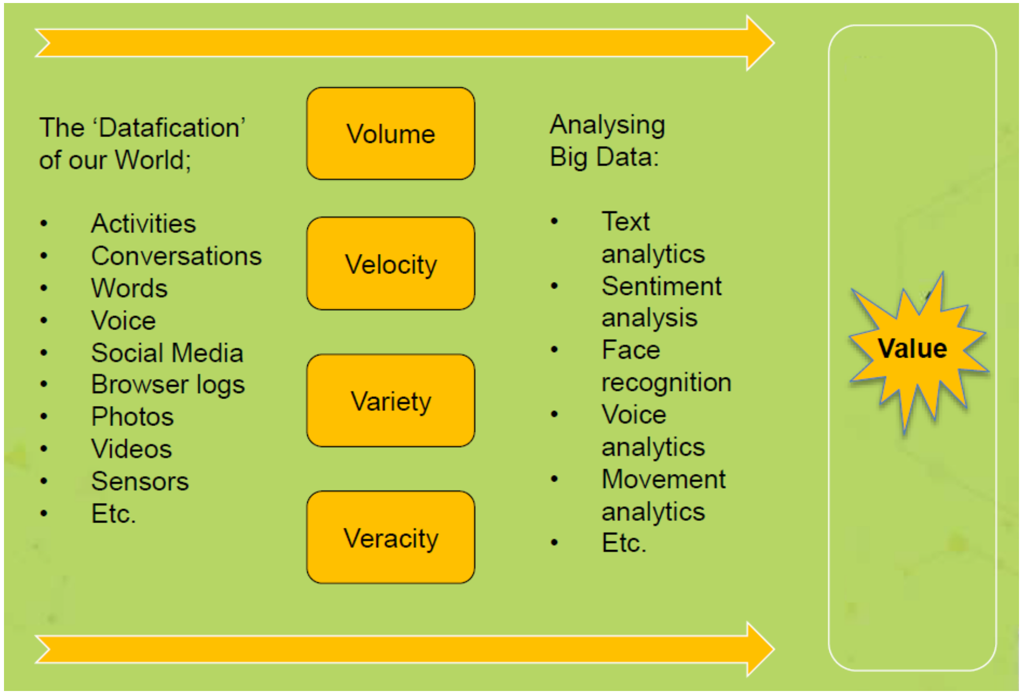
- With many forms of Big Data, quality and accuracy are less controllable
  - Just think of Twitter posts with hash tags, abbreviations, typos and colloquial speech as well as the reliability and accuracy of content
  - Technology now allows us to work with this type of data

Turning Big Data into Value

The datafication of our world gives us unprecedented amounts of data in terms of Volume, Velocity, Variety and Veracity

Technology such as cloud computing and distributed systems together with the latest software and analysis approaches allow us to leverage all types of data to gain insights and add value

Turning Big Data into Value



Big data use case
Better understand and target customers

- To better understand and target customers, companies expand their traditional data sets with social media data, browser, text analytics or sensor data to get a more complete picture of their customers
- The big objective, in many cases, is to create predictive models
- Using Big Data:
  - Telecom companies can now better predict customer churn
  - Retailers can predict what products will sell
  - Car insurance companies understand how well their customers actually drive

Big data use case Understand and Optimize Business Processes

Big Data is also increasingly used to optimize business processes

Retailers are able to optimize their stock based on predictive models generated from social media data, web search trends and weather forecasts

Another example is supply chain or delivery route optimization using data from geographic positioning and radio frequency identification sensors

Big data use case - Improving Health

The computing power of big data analytics enables us to find new cures and better understand and predict disease patterns

We can use all the data from smart watches and wearable devices to better understand links between lifestyles and diseases

Big data analytics also allow us to monitor and predict epidemics and disease outbreaks, simply by listening to what people are saying

"Feeling rubbish today in bed with a cold"

Searches for on the Internet, i.e. "cures for flu"

Big data use case Improving Security and Law Enforcement

- Security services use big data analytics to foil terrorist plots and detect cyber attacks
- Police forces use Big Data tools to catch criminals and even predict criminal activity
- Credit card companies use Big Data analytics to detect fraudulent transactions

Big data use case Improving Sports Performance

Most elite sports have now embraced Big Data analytics

Many use video analytics to track the performance of every player in a football or baseball game

Sensor technology is built into sports equipment such as basket balls or golf clubs

Many elite sports teams track athletes outside of the sporting environment

- Using smart technology to track nutrition and sleep, as well as social media conversations to monitor emotional well-being

Big data use case Improving and Optimizing Cities and Countries

- Big data is used to improve many aspects of our cities and countries
  - Example - It allows cities to optimize traffic flows based on real time traffic information as well as social media and weather data
- A number of cities are currently are using Big Data analytics with the aim of turning themselves into Smart Cities, where the transport infrastructure and utility processes are all joined up
  - Where a bus would wait for a delayed train and where traffic signals predict traffic volumes and operate to minimize jams.

Big data – The Big Picture

- We are in the midst of a transformation to a Big Data economy
- Any business or organization that doesn't seriously consider the implications of Big Data runs the risk of becoming irrelevant and/or being left behind
  - Maintain competitive advantage