



MONASH University

Information Technology

# FIT3176 Advanced Database Design

Topic 12: Revision and Future Directions

Dr. Minh Le  
Minh.Le@monash.edu

**algorithm** distributed systems **database**  
systems **computation** knowledge ma  
**design** e-business **model** data mining **int**  
distributed systems **database** software  
**computation** knowledge management **an**

\*Adapted from slides developed by Lindsay Smith

# This unit...

1. Advanced Database Design
  - E/R is not complete enough
  - EER, covering superclass and subclass
2. SQL and PL/SQL
  - Trigger, Procedures/Functions, Packages
3. XML and XML DB
  - XML and XML Schema
  - XML in Oracle (XPath)
4. JSON and JSON DB
  - JSON Documents and Schema
  - JSON in Oracle

## 5. **Week 12-Revision and Future Directions**

### Where can we go from here?

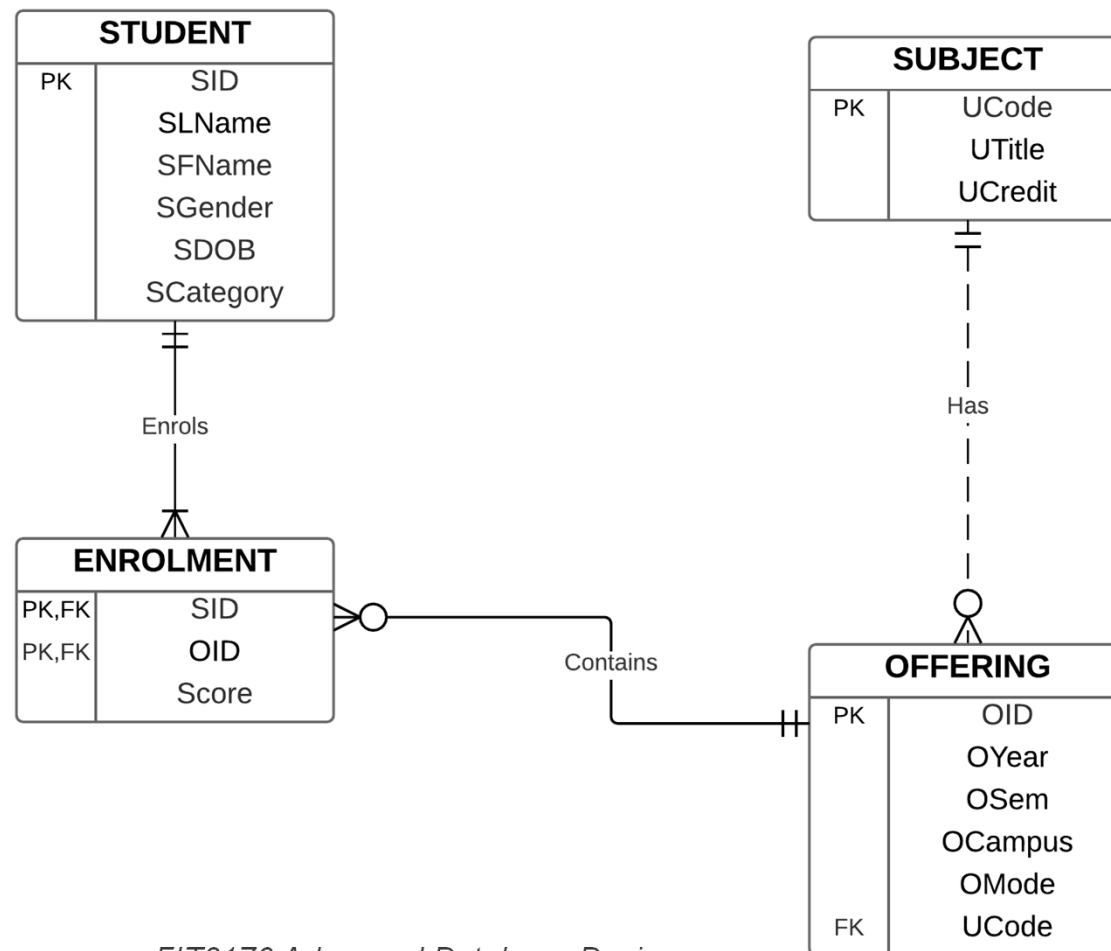
Two directions:

1. Data warehousing and analysis
2. Big Data (NoSQL) (Revisit)

# Data Warehousing and Business Intelligence

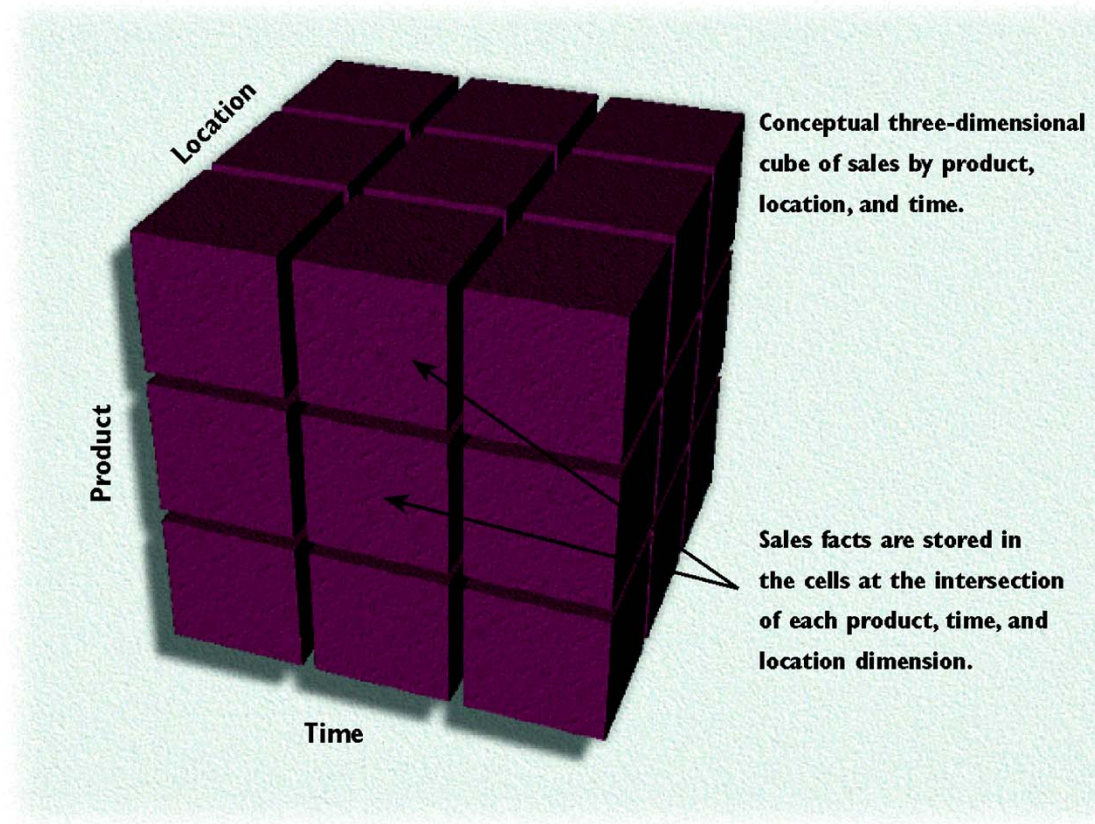
# Operational Database

## The Enrolment Database



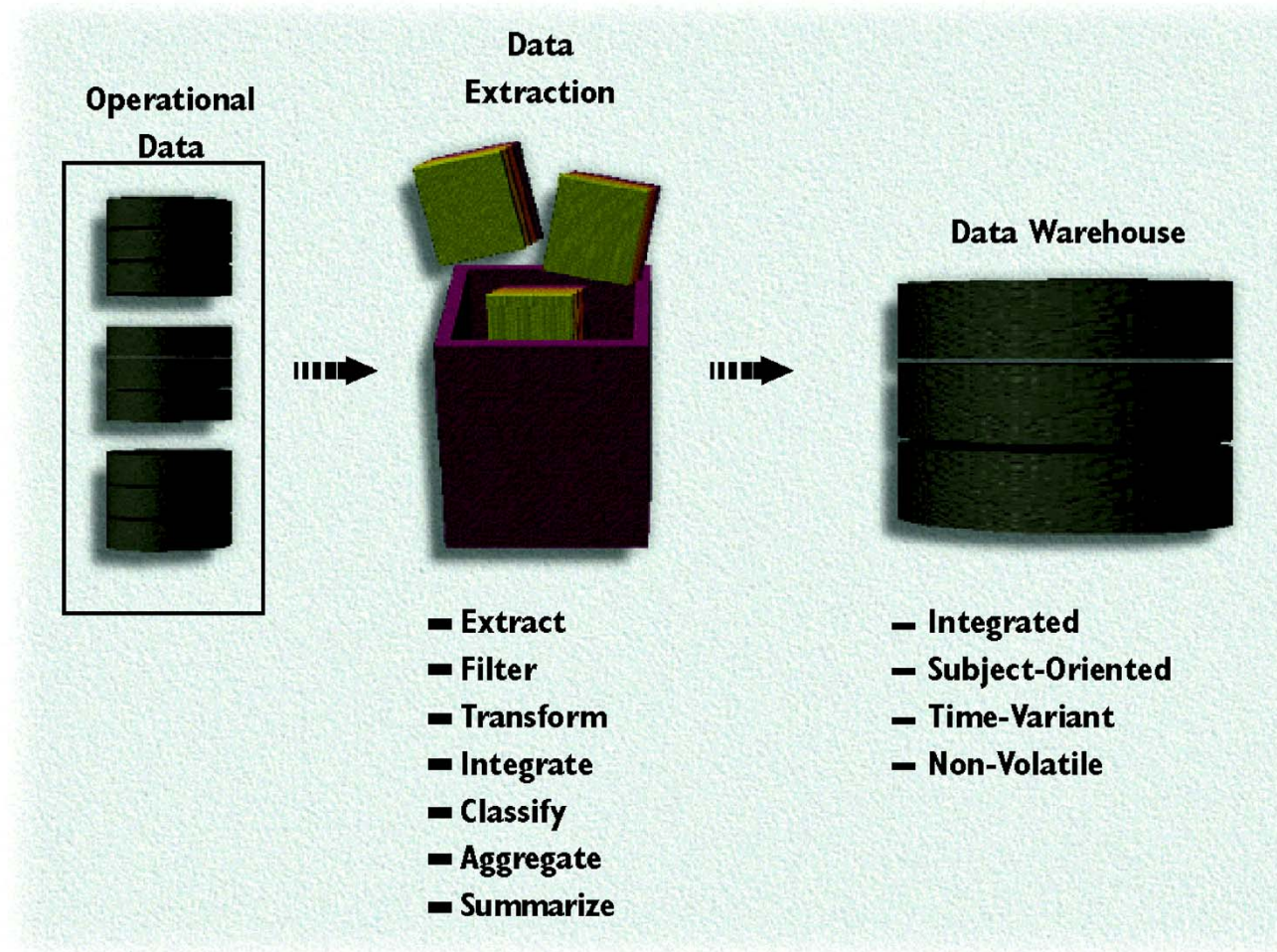
# Data Warehouse

## Multidimensional data view



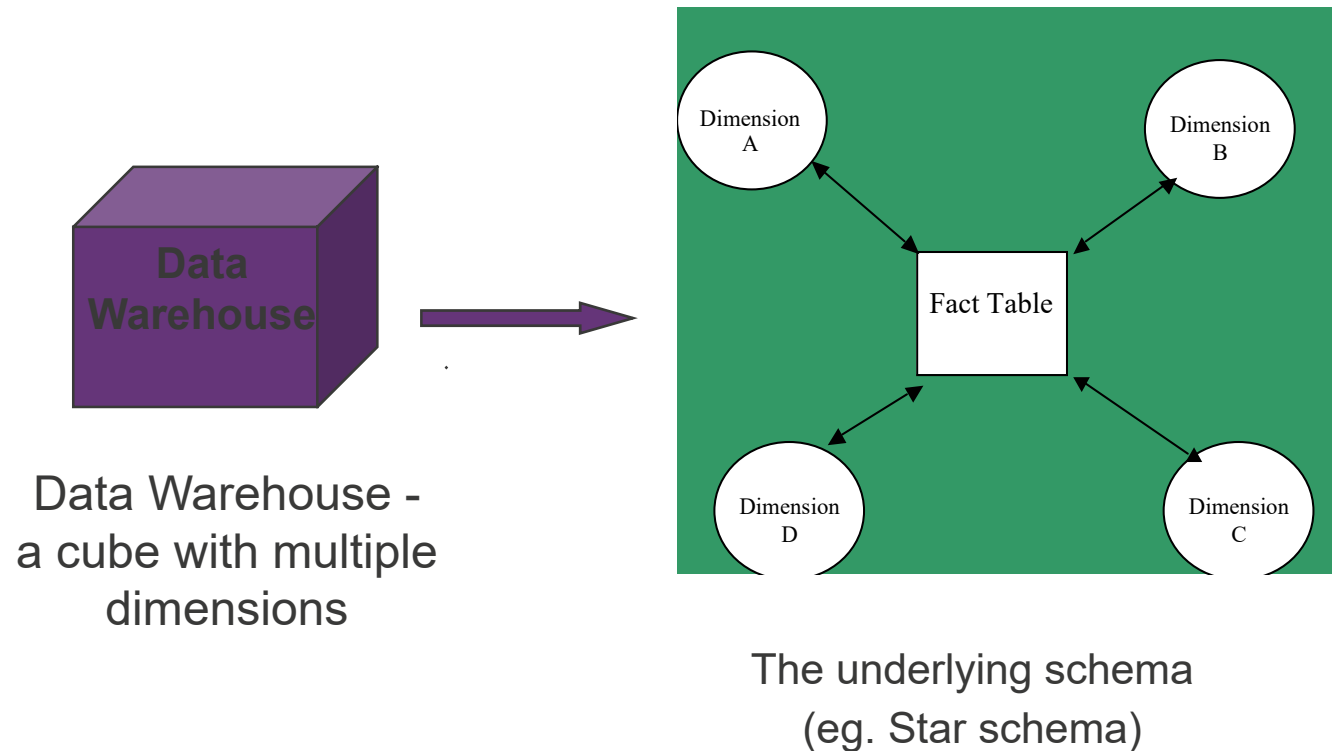


# From Operational to Data Warehousing



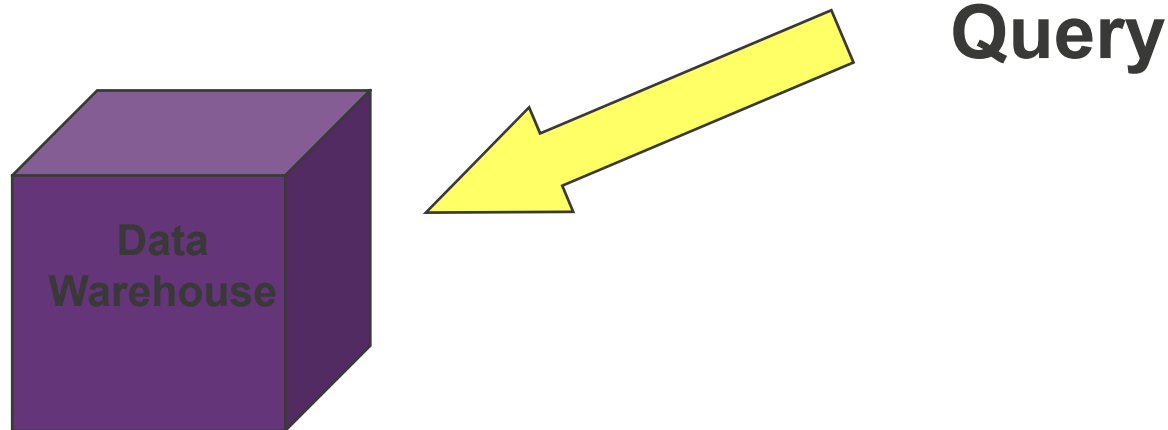
# Data Warehouse Design

The need for **a new data modelling technique** to support multi-dimensional views of a collection of data.



# Data Warehouse Queries

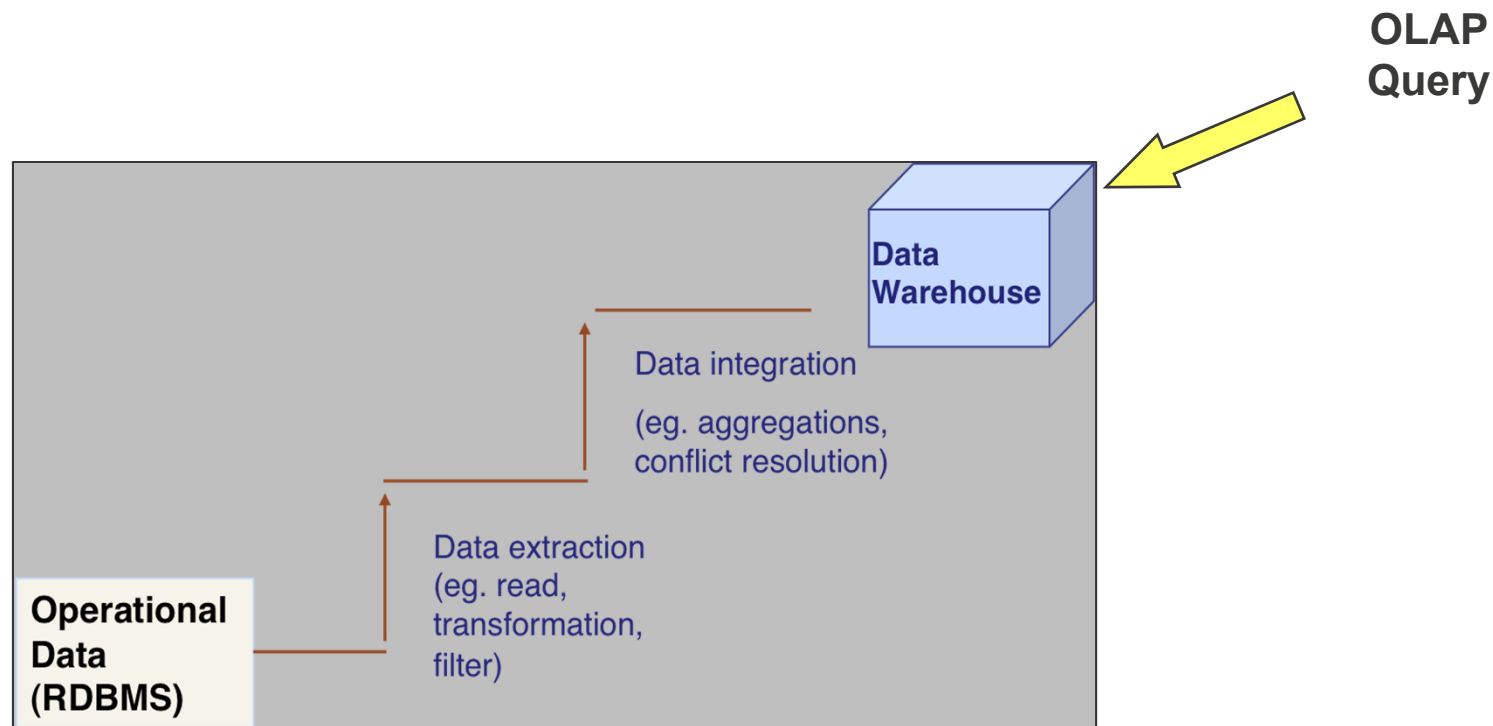
The nature of the new Data Warehouse storage structure requires a tool that supports the **retrieval of large number of records** from very large data sets and summarizes them “on the fly”  $\Rightarrow$  OLAP (**On-Line Analytical Processing**) tool.





# Data Warehousing and BI

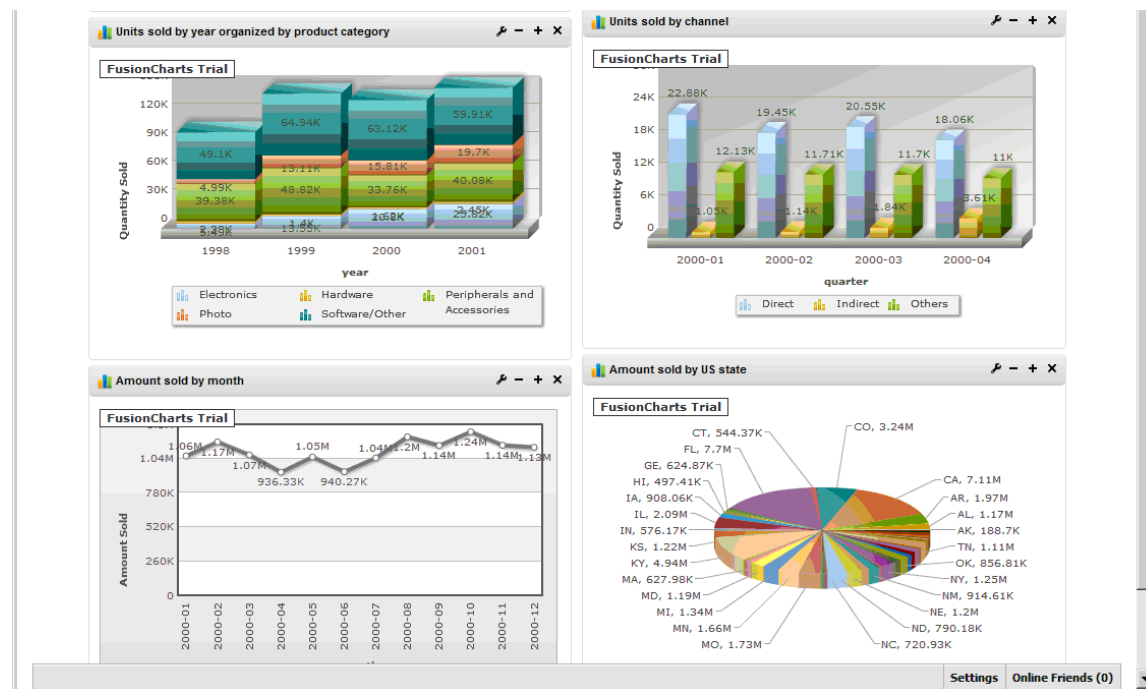
The need for an effective decision support system has motivated the emergence of the new data storage facility called **Data Warehouse**.



# Business Intelligence

A new emerging area that focuses on different techniques and tools to get useful analysis or knowledge from data/information/document collection which will be useful for managerial decision making.

Data Warehousing plays an important role in business intelligence.



# Big Data and NoSQL (Revisit)

# NoSQL, the new DBMS?



- Definition: “Next generation databases mostly addressing some of the points: being **non-relational**, **distributed**, **open-source**, and **horizontal scalable**. The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: **schema-free**, **easy replication support**, **simple API**, **eventually consistent**/BASE (not ACID), a **huge data amount**, and more.”

<http://www.nosql-database.org/>

# NoSQL, the new DBMS?

- NoSQL = **no-SQL** or **Not Only SQL**. Non-relational data storage systems.
- NoSQL movement was probably inspired by **Google's Big Table** or even **Amazon's S3**.
- Explosion of social media sites (Facebook, Twitter) with large data needs
- NoSQL feature: shared-nothing **horizontal scaling** – replicating and partitioning data over many servers.
- However, NoSQL does not provide ACID properties, but **BASE** (Basically Available, Soft state, Eventually consistent !!!)
- The idea is to give up ACID to achieve much higher performance and scalability.



# NoSQL Data Models (1)

1. **Key-value store**
2. **Document-oriented database**
3. **Column data store**

[https://www.youtube.com/watch?v=ql\\_g07C\\_Q5I](https://www.youtube.com/watch?v=ql_g07C_Q5I)

# NoSQL Data Models (2)

## 1. Key-value store

- Each item stored consists of a **key and value pair** (the value may be a numeric, a document, an image, etc)
- Key-value systems treat the data as a single opaque collection which **may have different fields** for every record

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

- **Redis** (*RE*mote *D*irectory *S*erver)
- Oracle NoSQL Database

<http://www.tutorialspoint.com/redis/>

# NoSQL Data Models (3)

## 1. Key-value store

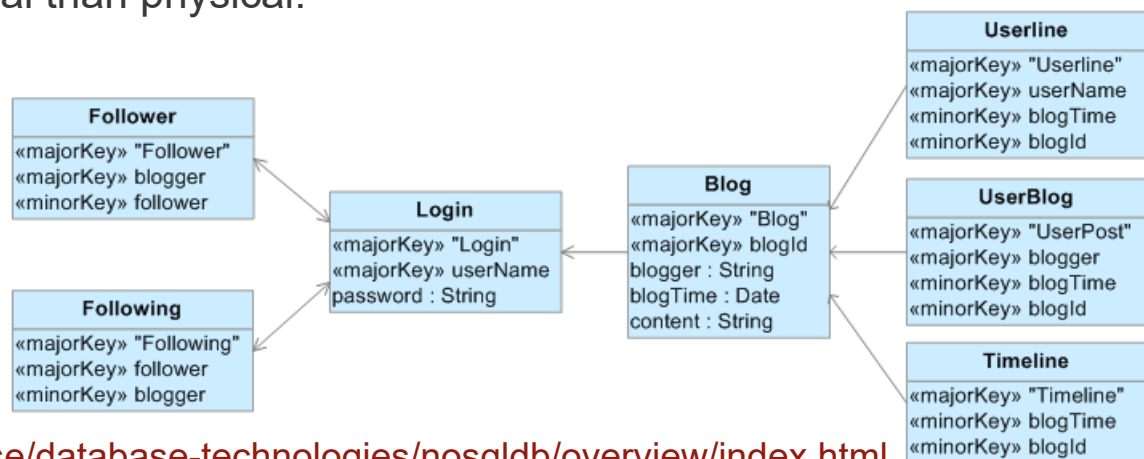
### – Oracle NoSQL Database

Keys consist of a list of Java Strings, grouping into two parts: **major** and **minor** components. A key must have at least one major component.

**Values** are simply stored as **arrays of bytes opaquely**.

**Kvitter Sample Application**: is a Twitter-like microblog sample application. A user is uniquely identified by his/her user name. A user signs in using his/her password. A blog is uniquely identified by the blog Id. A blog is created by a user. A user can follow other users.

Key-value NoSQL databases are inherently **schema-less**. As a result, the schema is more logical than physical.



# NoSQL Data Models (4)

## 1. Key-value store

### – Oracle NoSQL Database

Kvitter is a Java enterprise web application, built using JavaServer Faces (JSF) 2.0 and Oracle NoSQL Database.

The screenshot shows the Kvitter web application interface. At the top, the word "kvitter" is displayed in blue. To its right, navigation links include "Welcome Jason", "Sign out", "Userline", "Posts", "Timeline", and "Help". Below the header is a text input field with the placeholder "Jason, what's happening?" and a "Post" button. The main content area displays a list of posts, each with a user name, text, and timestamp. The posts are: Jason (Hello, world!, Jul 14 at 16:11 PM), John (Wow, finally, NoSql database from Oracle!, Jul 14 at 16:10 PM), April (Kvitter is a sample JSF application on Oracle NoSql Database., Jul 14 at 16:10 PM), Joe (JSF has made a long way. JSF 2.0 is such a joy to use., Jul 14 at 16:08 PM), and June (To SQL or not to SQL ..., Jul 14 at 16:06 PM). On the right side, there are two sections: "Followers" listing June, Jerry, and April, and "Following" listing June, John, Joe, Jerry, and April, with a "More..." link below.

**kvitter** Welcome Jason | Sign out | Userline | Posts | Timeline | Help

Jason, what's happening?

**Jason**  
Hello, world!  
Jul 14 at 16:11 PM

**John**  
Wow, finally, NoSql database from Oracle!  
Jul 14 at 16:10 PM

**April**  
Kvitter is a sample JSF application on Oracle NoSql Database.  
Jul 14 at 16:10 PM

**Joe**  
JSF has made a long way. JSF 2.0 is such a joy to use.  
Jul 14 at 16:08 PM

**June**  
To SQL or not to SQL ...  
Jul 14 at 16:06 PM

More...

**Followers**  
June  
Jerry  
April

**Following**  
June  
John  
Joe  
Jerry  
April  
More...

# NoSQL Data Models (5)

## 2. Document-oriented database

- Document-oriented databases are inherently a subclass of the key-value store.
- In a key-value store, the data is considered to be inherently opaque (not transparent) to the database, whereas a document-oriented system relies on internal structure in the document order to extract metadata that the database engine uses for further optimization.
- XML databases are a specific subclass of document-oriented database
- Document databases get their type information from the data itself, normally store all related information together, and allow every instance of data to be different from any other.



# NoSQL Data Models (6)

## 2. Document-oriented database

- Each item is stored as a document (normally JSON document, but could be XML)
- Practically any "document" containing metadata can be managed in this fashion, and common examples include XML, YAML, JSON, and BSON.

- MongoDB
- CouchDB

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user: 'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user: 'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

<http://www.mongodb.com/presentations/webinar-relational-databases-mongodb-what-you-need-know-0>

# NoSQL Data Models (7)

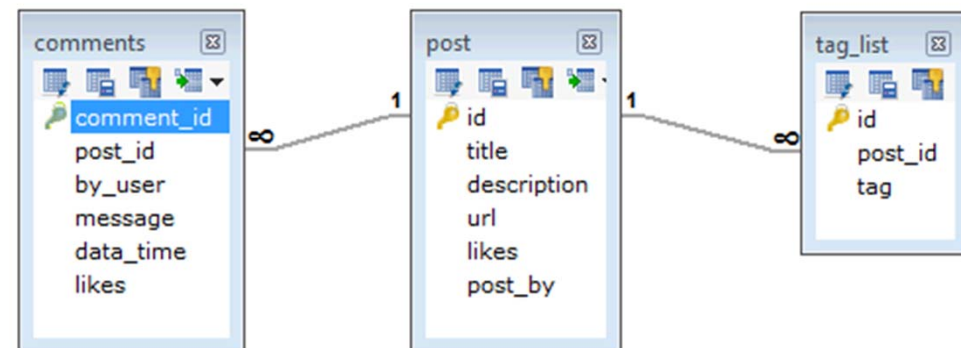
## 2. Document-oriented database

### – MongoDB

A blog website case study:

- Every post has the unique title, description and url.
- Every post can have one or more tags.
- Every post has the name of its publisher and total number of likes.
- Every post have comments given by users along with their name, message, data-time and likes.
- On each post there can be zero or more comments.

An RDBMS Model



<http://www.tutorialspoint.com/mongodb/>

# NoSQL Data Models (8)

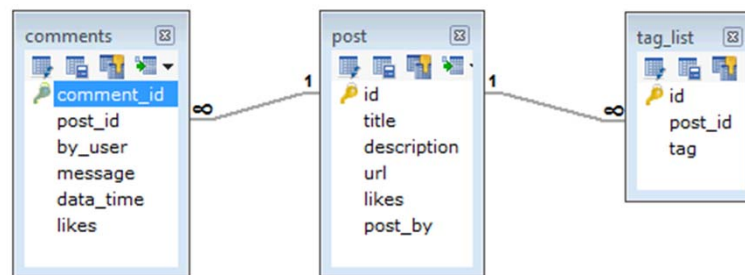
## 2. Document-oriented database

### – MongoDB

A blog website case study:

- Every post has the unique title, description and url.
- Every post can have one or more tags.
- Every post has the name of its publisher and total number of likes.
- Every post has comments given by users along with their name, message, data-time and likes.
- On each post there can be zero or more comments.

### An RDBMS Model



### A MongoDB Model

```
{
  _id: POST_ID
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```

# NoSQL Data Models (9)

## 3. Column data store

- Key points to a set of multiple column values arranged by column family
- Each row may have different columns

ColumnFamily			
Row key	Name	...	Name
	Value		Value
	...		...

- **Cassandra**
- Hbase
- Vertica

```
UserProfile = {  
  Cassandra = {emailAddress:"cassandra@apache.org", age:20},  
  TerryCho = {emailAddress:"terry.cho@apache.org", gender:"male"},  
  Cath = {emailAddress:"cath@apache.org", age:20, gender:"female",  
    address:"Seoul"},  
}
```

# NoSQL Data Models (10)

## 3. Column data store

- A **super column** is where the key is mapped to a value that is column families

```
UserList={
  Cath:{
    username:{firstname:"Cath",lastname:"Yoon"}
    address:{city:"Seoul",postcode:"1234"}
  }
  Terry:{
    username:{firstname:"Terry",lastname:"Cho"}
    account:{bank:"hana",accounted:"1234"}
  }
}
```

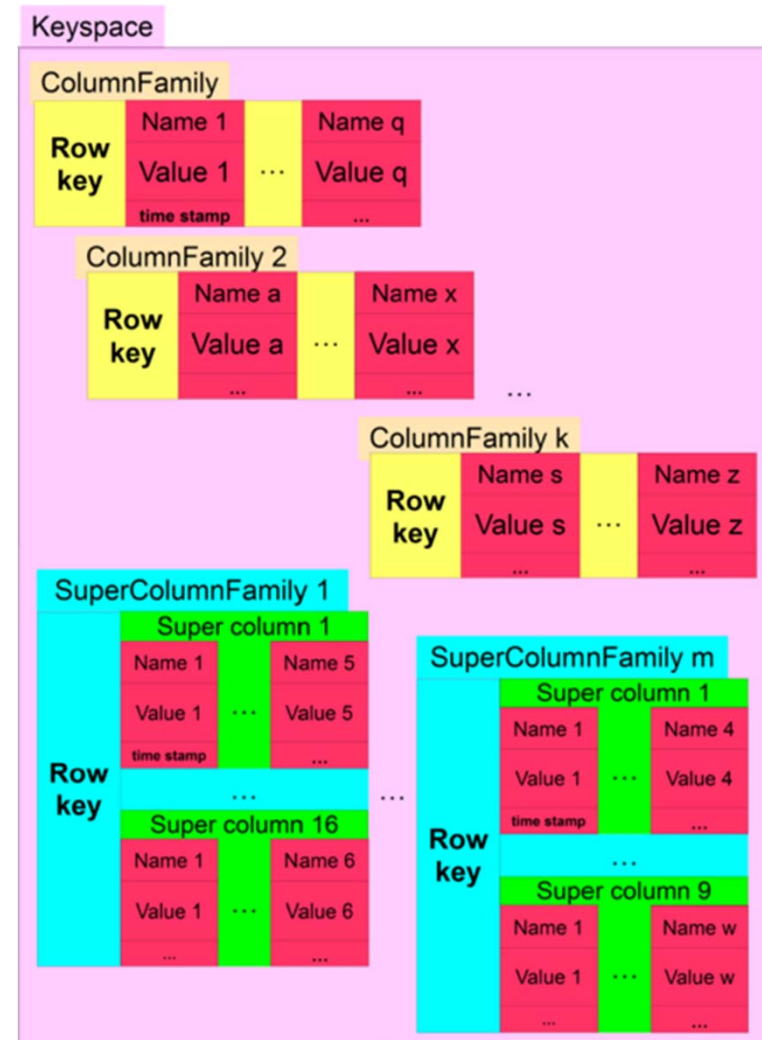
SuperColumnFamily			
Row key	Super column name		
	Name		Name
	Value	...	Value
	...		...
	...		
	Super column name		
	Name		Name
	Value	...	Value
	...		...



# NoSQL Data Models (11)

## 3. Column data store

- A **keyspace** (or **key space**) holds together all column families of a design.
- A keyspace may contain column families or super columns. Each super column contains one or more column family, each column family contains at least one column.
- Generally, there is one keyspace per application.

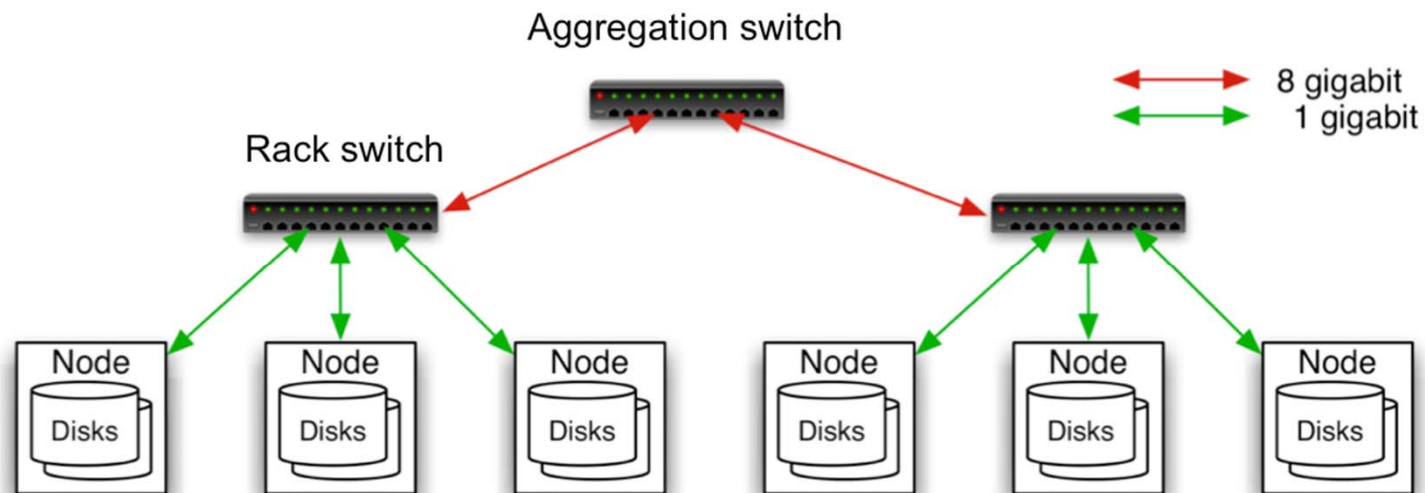


# NoSQL, and Big Data?

- **What is Big Data Technology?**
  - MapReduce programming model for large scale parallel processing of data.

# Parallel Computers

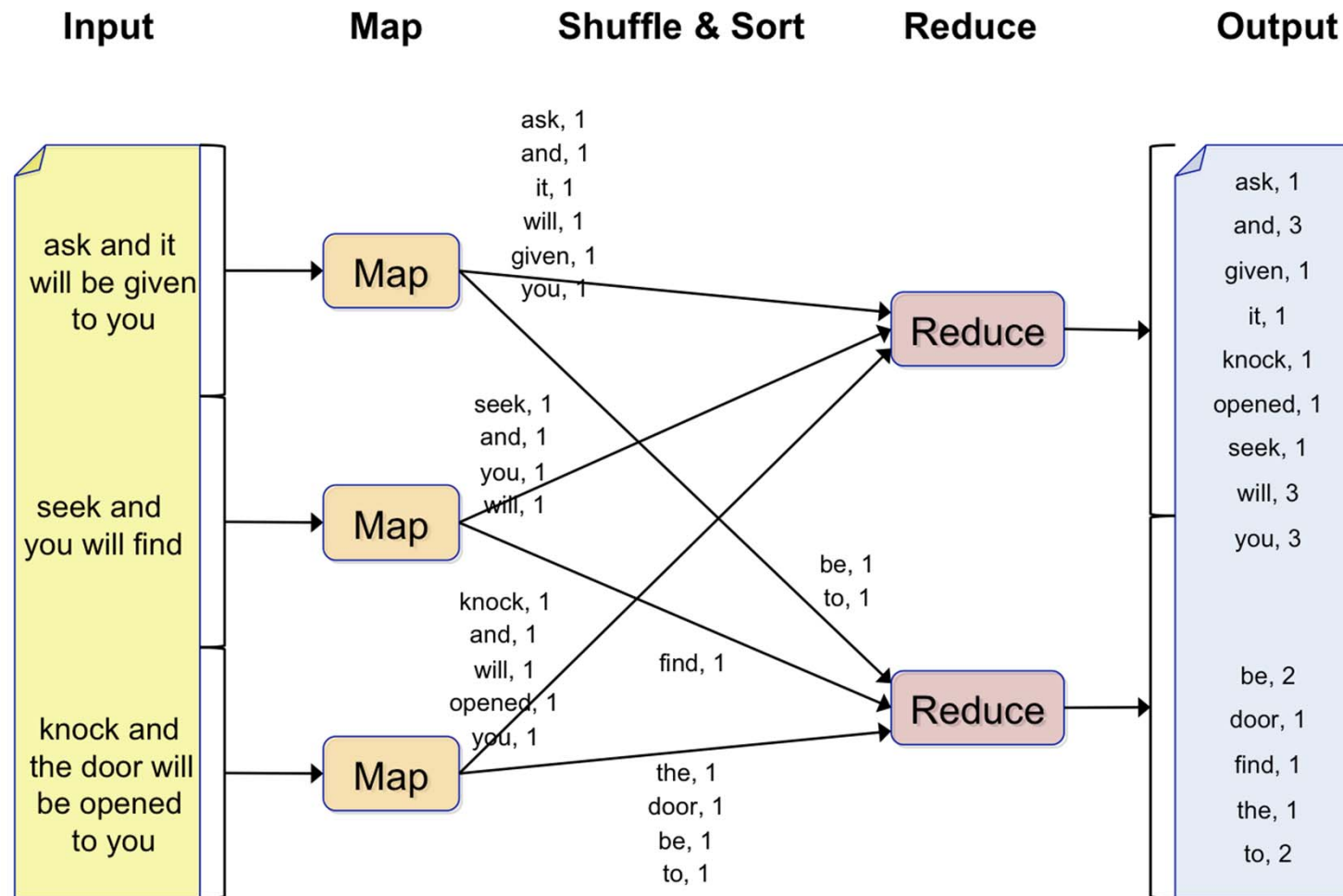
- Constructing high performance parallel computers using a large number of (low-end) **commodity processors**.
- Commodity machines (cheap, but unreliable).
- Commodity network.
- Scalable (1000's of machines, 10,000's of disks)



# MapReduce

- MapReduce is a programming model for large scale parallel processing of Data. The model consist of two functions Map and Reduce.
- Mapper is a function that performs filtering and Reducer clubs the data provided by Mapper.
- Designed for scalability and fault-tolerance.
- Pioneered by Google
  - Processes 20 Petabytes of data per day
- Popularized by open-source Hadoop project
  - Used at Yahoo!, Facebook, Amazon, ...

# MapReduce - Word Count Example





# Exam Discussion

# Requirements to pass FIT3176

- Unit weighting
  - Non exam assessment = 40% (Asst 1: 25% + Asst 2: 15%)
  - Examination = 60%
- Requirement to pass
  - 40% or more in the unit's examination, and
  - 40% or more in the unit's total non-examination assessment, and
  - an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is:


- equal to or greater than 50%, then a mark of 49-N will be recorded for the unit.
- less than 50% then the actual mark for the unit will be recorded.

# 2018 Exam

- There are 6 questions in the exam
- The exam will cover
  - Advanced Entity Relationship Modelling
  - Oracle PL/SQL programming: Triggers, Procedures, Functions, Packages
  - XML: document structure, schema, XML tree representation, XPath, XQuery
  - JSON: document structure, schema, JSON and XML
- Topics will not cover
  - JSON in Oracle database
  - JSON Path expressions

# 2018 Exam (Cont.)

- Exam details
  - Exam Duration: 2 hours
  - Exam date: Thursday, 07 JUNE 2018 (Please always recheck your Exam Timetable to confirm this date and time.)
  - Session Time: AM
  - Location: TBA
- Time is an issue to be careful of
  - Use reading time productively
  - 100 marks in 2 hrs (120 mins)
    - Allowing some 'free' time – 1 min per mark



**THANK YOU  
and  
GOOD LUCK!**