James Thom!!beard大爷纯正aussie口音

废话超多

TOPIC: Structured, semi-structured and unstructured data

1、结构化数据，即行数据，存储在数据库中，可以用二维表来逻辑表达实现的数据。

比如企业ERP、财务系统；医疗HIS数据库；教育一卡通；政府行政审批；其他核心数据库等。这些应用需要哪些存储方案呢？基本包括高速存储应用需求、数据备份需求、数据共享需求以及数据容灾需求。

2、非结构化数据，即不方便用数据库二维逻辑表来表现的数据。

包括所有格式的办公文档、文本、图片、XML、HTML、各类报表、图像和音频/视频信息等等

3、半结构化数据：就是介于完全结构化数据。如关系型数据库、面向对象数据库中的数据，和完全无结构的数据。如声音、图像文件等之间的数据。HTML文档就属于半结构化数据。它一般是自描述的，数据的结构和内容混在一起，没有明显的区分。

包括邮件、HTML、报表、资源库等等，典型场景如邮件系统、WEB集群、教学资源库、数据挖掘系统、档案系统等等。这些应用对于数据存储、数据备份、数据共享以及数据归档等基本存储需求。

哎呀我靠废话真的好多！！！！！

要申请AWS。 Institution name is RMIT

…回去细看slides闹好了

A1 10%, A2 30%, online test 10%, final 50%

Weekly tests each Friday！！

Data models: relational data model is the dominant model 就是关系模型

奥就是当年学SQLserver的时候得那些东西呗

\*Impedance mismatch: refers to deference between relational model and in-memory record structures

首先，关系数据库在管理复杂数据时显得笨重。其次，被应用软件操作的数据一般是用面向对象的编程语言如C++, Java, Delphi和C#写成，而那些用来转化数据表示和关系数据库元组的代码很冗繁，执行时也有不少耗时。这种应用程序和数据库表示信息的模式之间的不匹配有时也被称为impedance mismatch。

\*with clusters relational databases 数据库群集

\*aggregate-oriented data model聚合数据模型

把一组相互关联的对象视为一个整体单元来操作，这个单元就是聚合。

选用面向聚合的模型决定性因素，在于其非常适合于在集群中运行。

Key-value store: for shopping carts or sth.

Column-family store: can store tabular data. E.g. google bigtable

Graph data

Schemaless database: NoSQL databases are schemaless btw schemaless无模式

Database changes are more flexible, easier to deal with non-uniform data but still need some form 啊误了

MongoDB ?