

# Introduction to Databases

Trishla Shah

[trishla.shah@smu.ca](mailto:trishla.shah@smu.ca)



What is Database?



# Database



- A database is an organized collection of data, stored and accessed electronically.
- Collection of related data; intended for shared use and different uses.
- Represents some aspect of the real world: **miniworld** or **universe of discourse (UoD)**
- Built for a specific purpose

Why should I learn about Database?



# Why should you study databases?

- **Mercenary- make more \$\$\$:**

- Startups need DB talent right away = low employee #
- Massive industry...



- **Intellectual:**

- Science: data poor to data rich
  - No idea how to handle the data!
- Fundamental ideas to/from all of CS:
  - Systems, theory, AI, logic, stats, analysis....

Many great computer systems ideas started in DB.

# Types of Databases

- Traditional Database
- Multimedia Database
- Geographic Information System (GIS)
- Real time Database
- Data Warehouse

# Traditional Database

- It is collection of text and numbers
- It is currently used in:



Banking System



Airline System



Hotel System



Railway System

AND MANY MORE!

# Multimedia Database (MMDB)

- It stores pictures, video clips and sound messages etc.
- Application areas:



Video-on-Demand



News-on-Demand



Digital Libraries



Music Database



# Geographic Information System (GIS)

- Store and analyze maps, weather data, and satellite images
- Applications are:
  - Agriculture Applications
  - Astronomy/Planetary Applications
  - Architecture Applications
- Example:  
**Parking Availability** – Orchestrating a parking available by collecting the percent of spaces occupied versus search time.

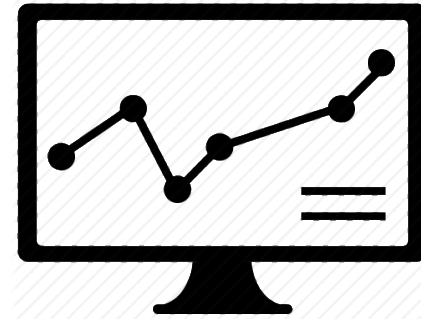
**Other Example:**

[https://www.youtube.com/watch?time\\_continue=29&v=Z9vAp40OHkg](https://www.youtube.com/watch?time_continue=29&v=Z9vAp40OHkg)



# Real-time Dataset

- A Database which uses real time processing to handle workloads whose state is constantly changing.
- For example, stock market, sports Update



# Data warehouses and online analytical processing (OLAP) systems

- Extract and analyze useful business information from very large databases
- Support decision making



Assume that we have one supermarket...



## Where did our data come from ?

- Lots of individual shoppers buying a juice or other items.
- Each transaction stored in database designed to store check out transactions.



## Operational Database

- It supports the day-to-day operations of a company

### Core operational database functionality:

- Gather Data
- Update Data
- Store Data
- Retrieve Data
- Archive Data

Collectively, operational systems are usually referred to as online transaction processing (OLTP)







# Buying juice at superstore:

You place juice on conveyor belt





# Buying juice at superstore:

Cashier swipes barcode over POS scanner



# Buying juice at superstore:

- POS system looks up price of toothpaste
- POS totals cost of transaction + tax
- POS prompts for payment



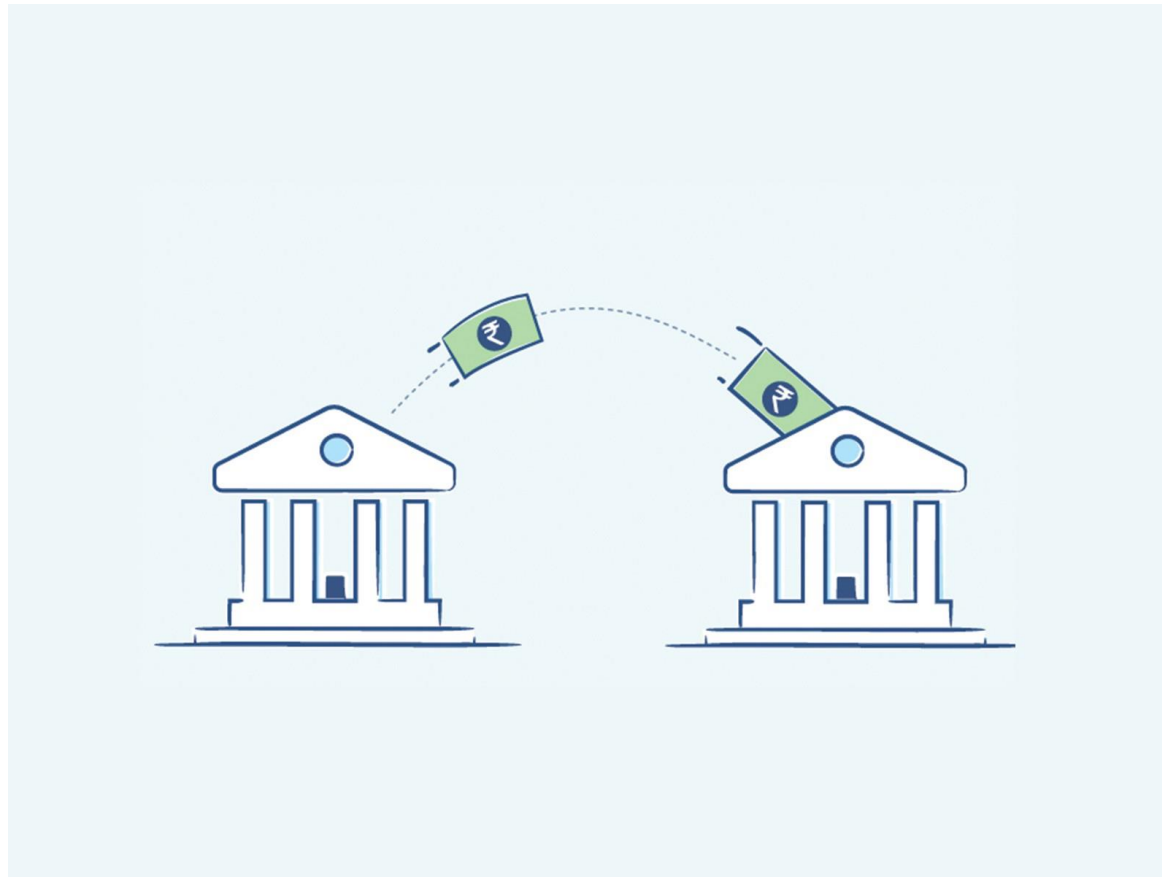
# Buying juice at superstore:

You swipe debit card and enter PIN



# Buying juice at superstore:

POS system transfers cost of juice from  
your bank account Target's account





# Buying juice at superstore:

## POS generates receipt and cashier bags



# Question

- List 3 example of each type of database (you are allowed to use internet!).

# Relational vs Non-relational Database

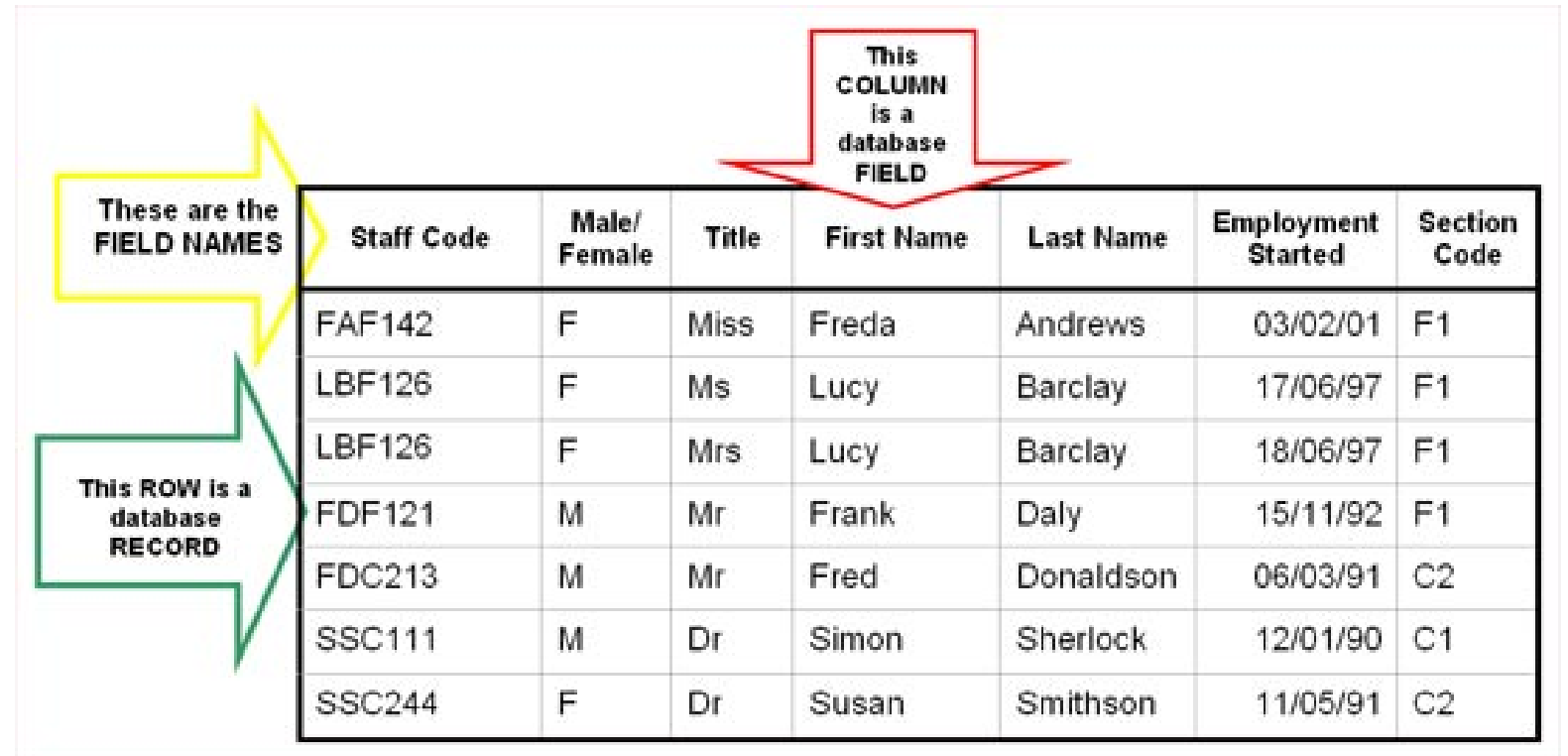


What is a Relational Database?



# Relational Database

- Highly structured table organization with rigidly-defined data formats and record structure.



The diagram shows a table with 7 columns and 7 rows of data. Annotations highlight key concepts:

- A yellow arrow points to the column headers: **These are the FIELD NAMES**
- A red arrow points to the 'First Name' column: **This COLUMN is a database FIELD**
- A green arrow points to the first row of data: **This ROW is a database RECORD**

Staff Code	Male/ Female	Title	First Name	Last Name	Employment Started	Section Code
FAF142	F	Miss	Freda	Andrews	03/02/01	F1
LBF126	F	Ms	Lucy	Barclay	17/06/97	F1
LBF126	F	Mrs	Lucy	Barclay	18/06/97	F1
FDF121	M	Mr	Frank	Daly	15/11/92	F1
FDC213	M	Mr	Fred	Donaldson	06/03/91	C2
SSC111	M	Dr	Simon	Sherlock	12/01/90	C1
SSC244	F	Dr	Susan	Smithson	11/05/91	C2

# TABLES: ROWS AND COLUMNS

- Tables can have hundreds, thousands, sometimes even millions of rows of data. These rows are often called *records*.
- Tables can also have many *columns* of data. Columns are labeled with a descriptive name (say, age for example) and have a specific *data type*. For example For example, a column called age may have a type of INTEGER (denoting the type of data it is meant to hold).

Name	Age	Country
Ned	6	USA
Zenas	10	Canada

# Example of Relational Database

**School Table**

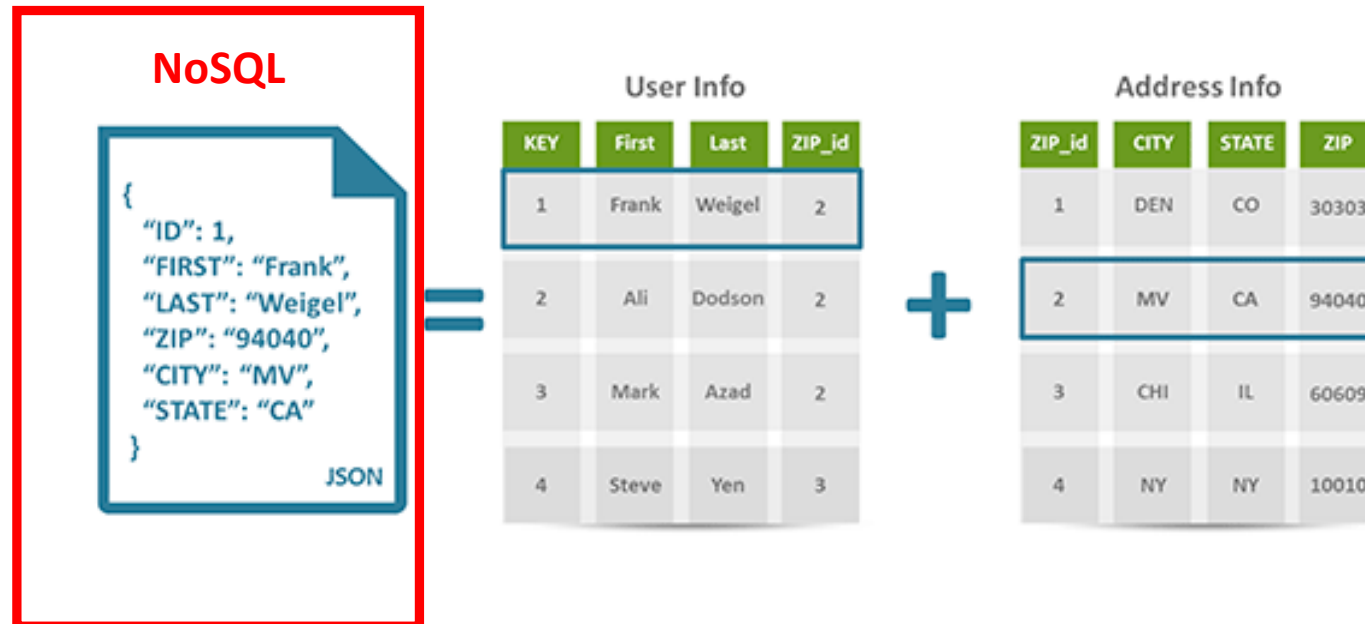
ID	Name
S001	University of Technology
S002	University of Applied Science

**Student Table**

School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

# Non-relational Database

- Un-structured stored in JSON files but the
- In a NoSQL or Non-relational database, there is no fixed schema and no joins



# What is the need of NoSQL Database?



# Non-relational Database

- NoSQL or Non-relational Database is a way to retrieve data or information from Big Data databases. It has seen a steady adoption growth with the rise of Big Data Applications.
- It is much faster and provides more comprehensive analytical information than normal SQL (Relational Database).

# Advantages of Non-relational databases over relational technology

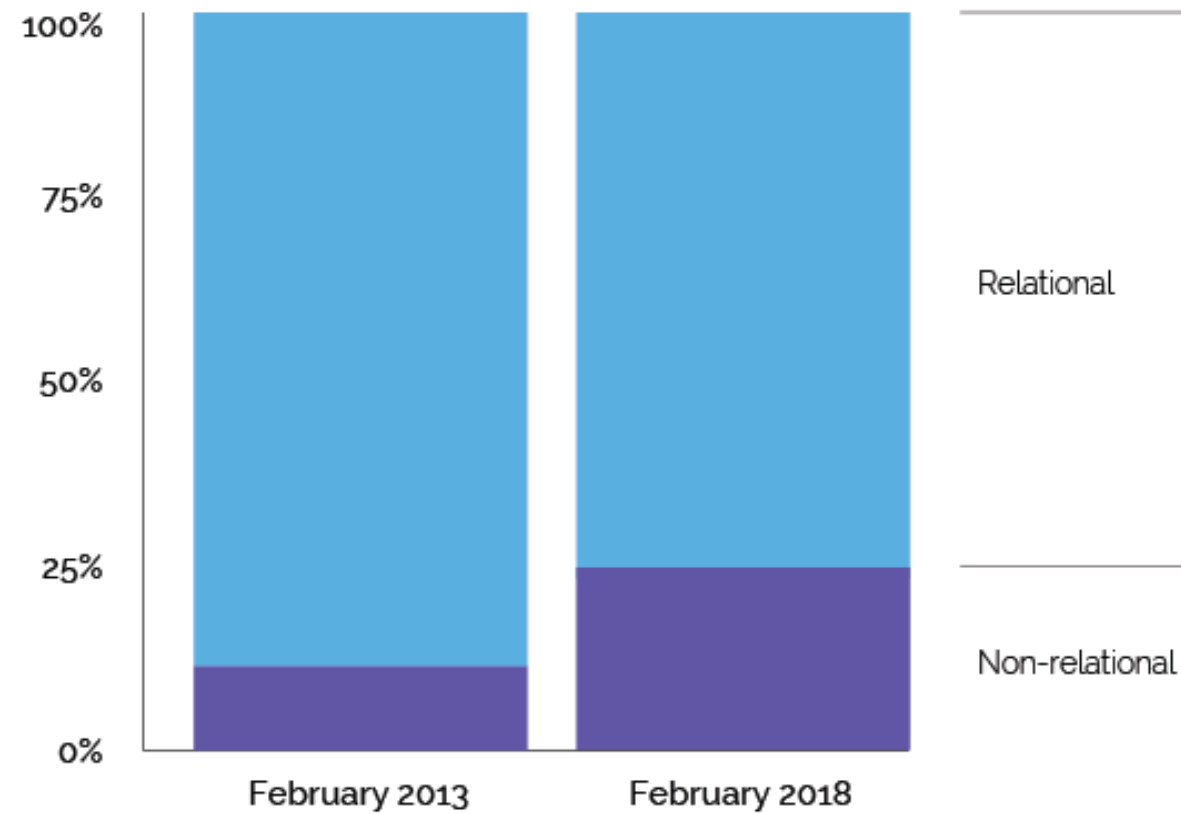
- **Data Models** - Unlike relational models which require predefined schema, NoSQL databases offer flexible schema design that make it much easier to update the database to handle changing application requirements.
- **Data Structure** - Non-relational databases are designed to handle unstructured data that doesn't fit neatly into rows and columns. This matters as most of the data generated today is unstructured.

# Contd....

- Scaling - You can scale your system horizontally by taking advantage of cheap, commodity servers.
- Development Model - NoSQL databases are typically open source which means you don't have to pay any software licensing fees upfront.



## Popularity (percentage) Relational Databases vs. Non-Relational Databases



Source: [https://db-engines.com/en/ranking\\_trend](https://db-engines.com/en/ranking_trend)

# NoSQL



Gaming



Social



IoT



Web



Mobile



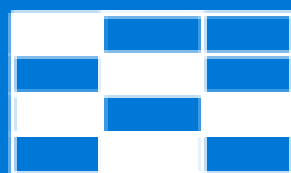
Enterprise



Key/value store



Document  
database



Column family store

# SQL



Web



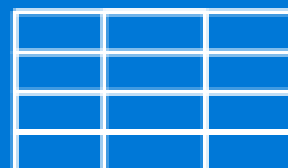
Mobile



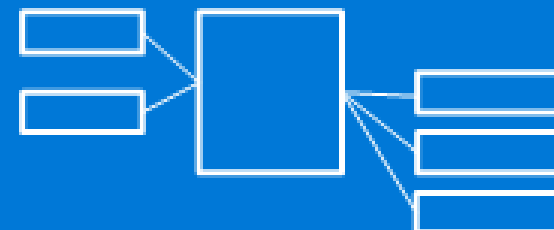
Enterprise



Data mart



Relational table storage



Relationships use joins

# Question

- When should we use SQL (Relational Database)?
- When Should we use NoSQL (Non-relational Database)?

# Database Management System (DBMS)

# What is Database Management System

- Software to create and maintain a database
- The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

# Why DBMS??

- Whether it's kept on the premises or off site, locally managed or handled by a third-party, businesses need a reliable, searchable and adaptable database to handle the constant influx of information.
- But databases don't store, manage and analyze this information on their own. The right database software system — also called a database management system (DBMS) — is critical to maximize performance and minimize IT headaches.

# Top DBMS systems for business professionals

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Microsoft Access
- Teradata
- IBM DB2
- Informix
- SAP ASE
- Amazon's SimpleDB

# File System vs Database Management System



## File System

- A file-based system is a collection of data stored in an orderly manner in a file.
- It is a file packed with data, with no metadata and thus no organizing structure.

## Database Management System (DBMS)

- DBMS is intended for easily organizing, storing and retrieving large amounts of data.

# Advantage of DBMS over FS

- **No redundancy:** Data is normalized in DBMS to minimize the redundancy which helps in keeping data consistent. For Example, student information can be kept at one place in DBMS and accessed by different users.
- **Simplified Data Access:** A user need only name of the relation not exact location to access data, so the process is very simple.
- **Multiple users:** Data can be accessed concurrently by different users at same time in DBMS.

- **Multiple views:** Different views of same data can be created to cater the needs of different users. For Example, faculty salary information can be hidden from student view of data but shown in admin view.
- **Security:** Only authorized users are allowed to access the data in DBMS. Also, data can be encrypted by DBMS which makes it secure.
- **Backup and Recovery mechanism:** DBMS backup and recovery mechanism helps to avoid data loss and data inconsistency in case of catastrophic failures.

# Disadvantage

- Complex software
- Initial cost
- User training

# Actors of a DBS

- Database Administrator (DBA)
- Database Designers
  - Find out requirements
  - Do a conceptual design using a data model
  - Implement design in the chosen DBMS
  - Create views
  - Choose file structures
- End users
- Application programmers

# Demo

- MySQL Database:

[https://www.youtube.com/watch?v=y4yQGR\\_ZLMQ](https://www.youtube.com/watch?v=y4yQGR_ZLMQ)