Relational Databases Introduction

Big Data Systems

Dr. Rubi Boim

Discussion

Assume you are hired to develop a system that manage an e-commerce store

- Keep tracks of the users

- Keep tracks of the items

- Keep tracks of the purchases

How would you do it?
Wix is not an option :)

Easily…

- Create a program that saves the data into text files

  - `/store/users.txt`

  - `/store/items.txt`

  - `/store/purchases.txt`

- Update the files according to the application logic

  - If a new user register, add her to `users.txt`

  - If user purchase an item, update `purchases.txt` with the basket

  - …

3

Stuff to consider

- What happens if a user updates her name?

- What happens if a user updates her credit card?

- What happen if we expend to different countries?

(more) Stuff to consider

- There is a need from the management to know:

  - What is the average order amount?

  - How many users bought items worth more than $1k?

  - Which are the most popular items (in the last week)?

  - Who are the users who haven't purchased anything in the last 3 months, but spent over $100 before?

  - …

- Is it still that easy?
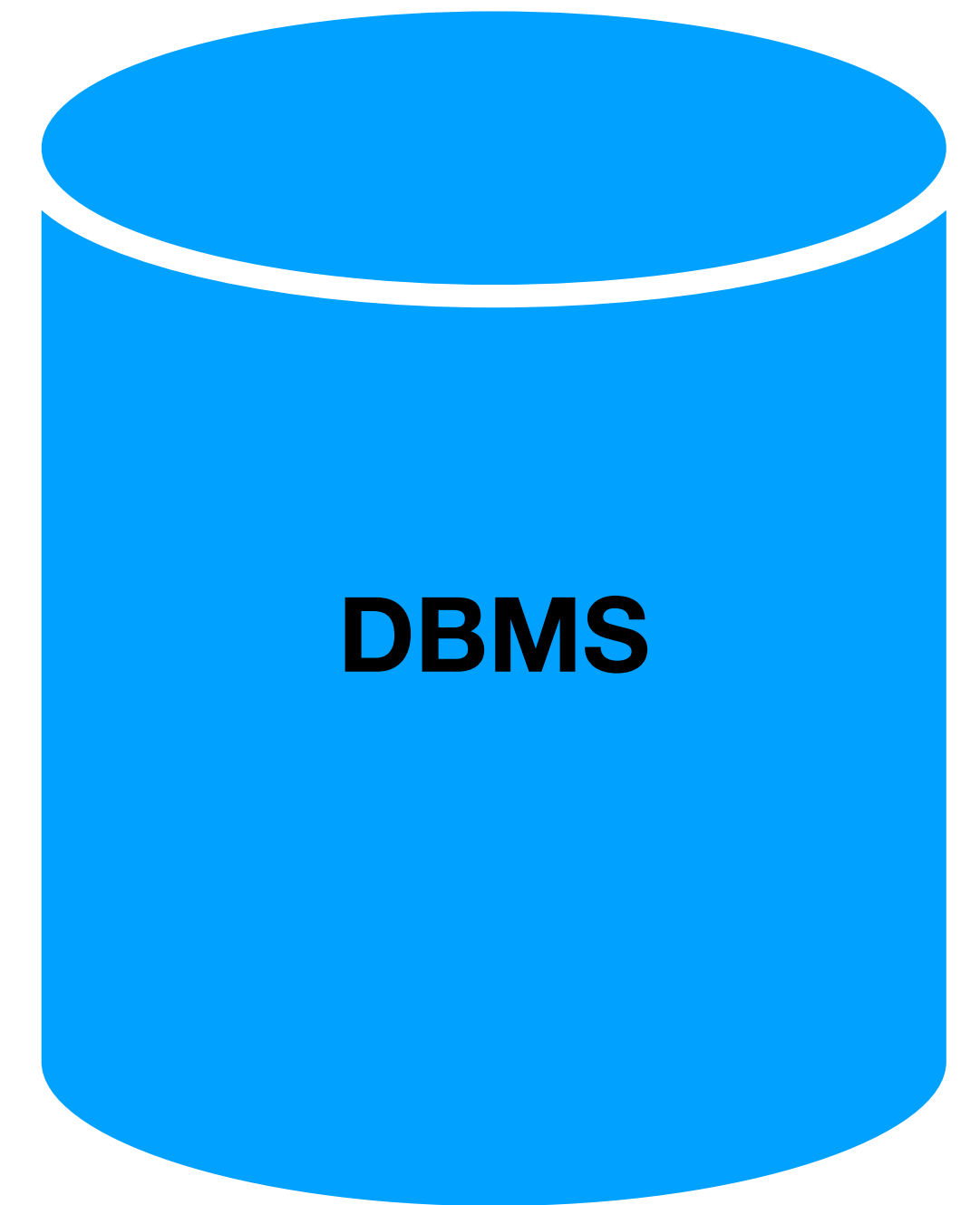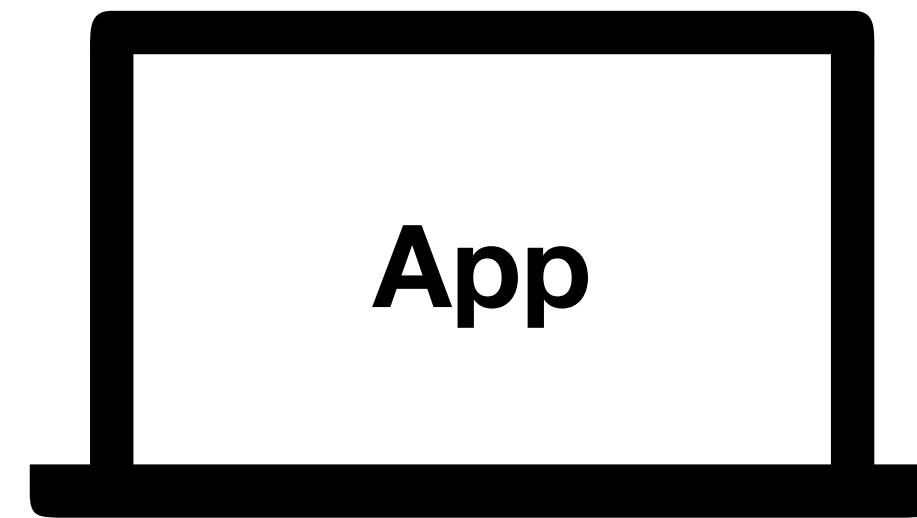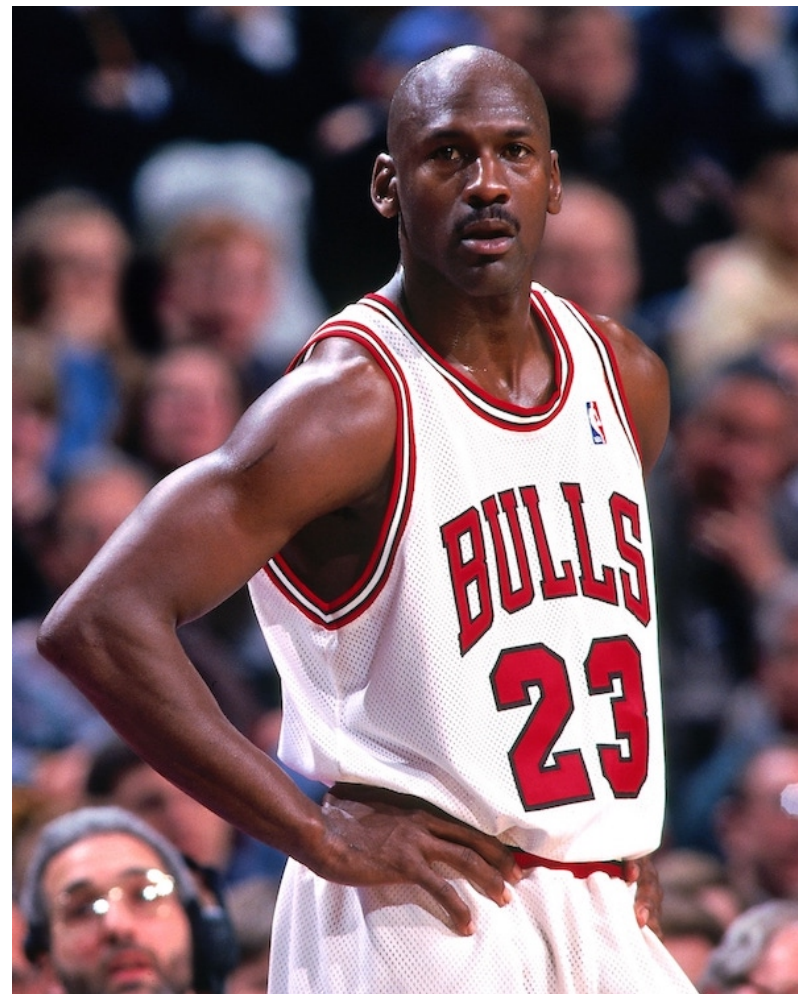
(more more) Stuff to consider

- How to backup the data?

- How to handle concurrency?

- What happens if the system crash in the middle of a purchase operation(s)?
  (the credit card is charged but the data was not added to the purchases file)

DBMS
database management system

DBMS

- A <u>software</u> that capture and analyze data by interactions with other applications
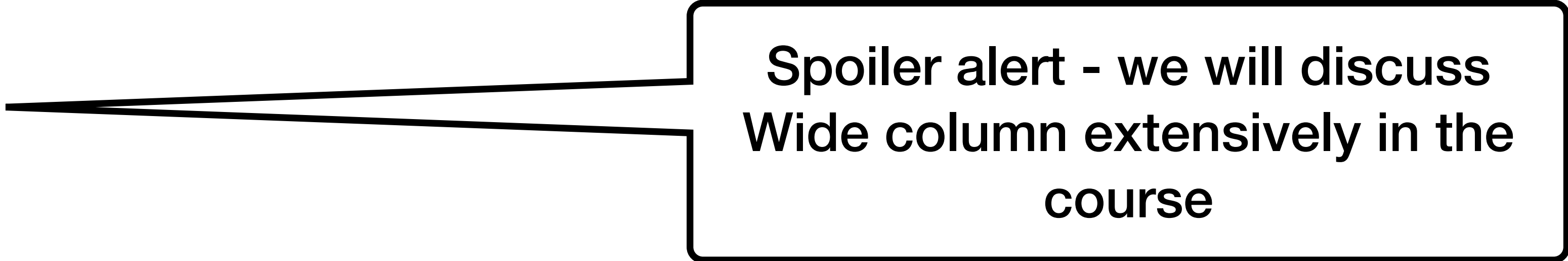
Modern DBMS supports

- Different data types

- User defined queries

- Transactions

- Query engine / optimization

- Storage management
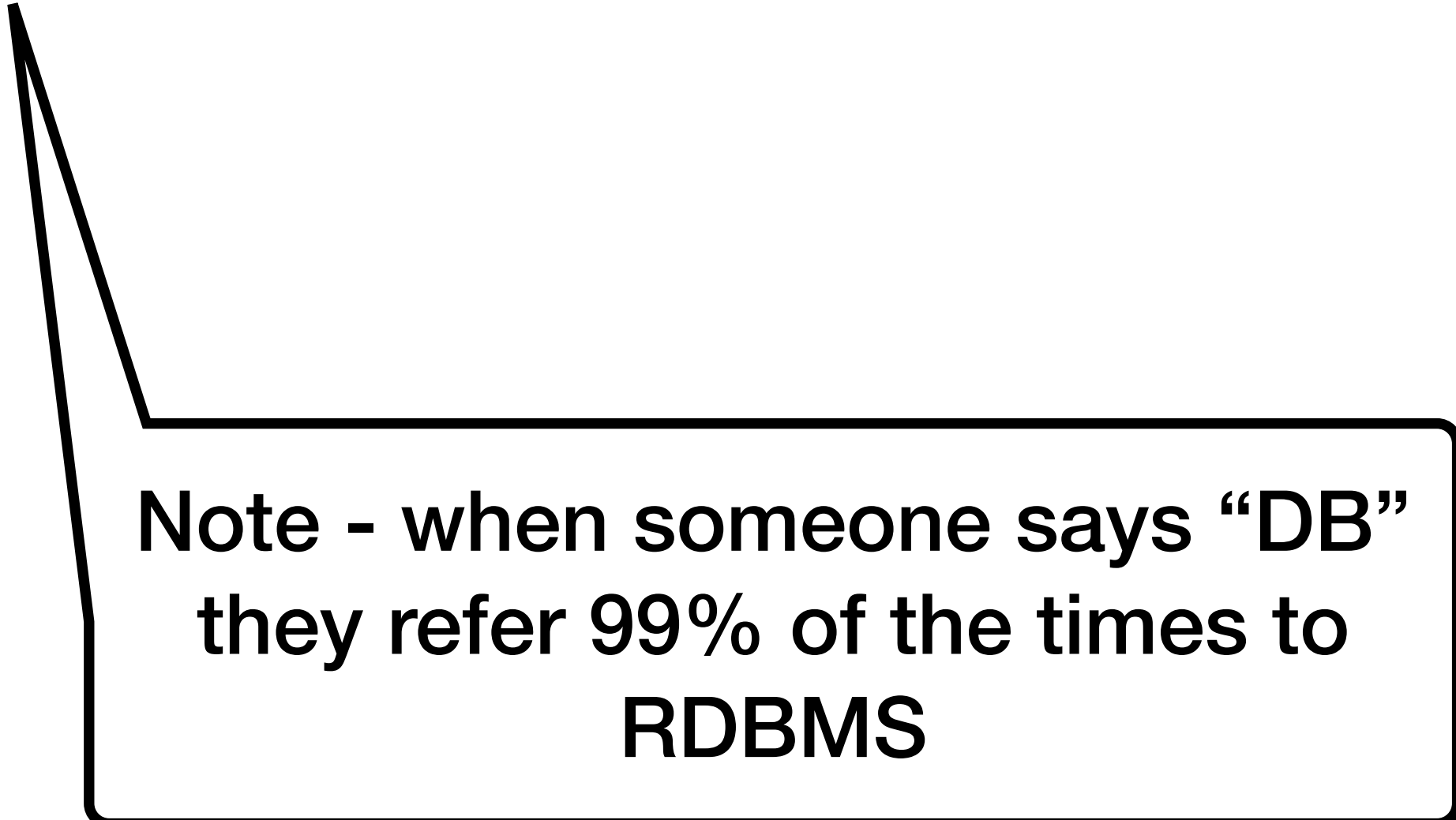
- Access management

- …

Database types

Databases are classified based on their data model

- **Table (Relational)**

- Key Value

- Graph

- Document

- Wide column

- …

Spoiler alert - we will discuss Wide column extensively in the course

Relational DBMS

Note - when someone says "DB" they refer 99% of the times to RDBMS

Relational model

- Data is stored in tables of columns and rows

- A <u>unique key</u> identify each row
  a table without a primary key - anti pattern

- The table is <u>unordered</u> (no first / last)

Table / relation

Columns / attributes

users

user_id	name	city	brithdate
101	Rubi Boim	Tel Aviv	<null>
102	Tova Milo	Tel Aviv	<null>
103	Lebron James	Los Angeles	30/12/1984
104	Michael Jordan	Chicago	17/02/1963

Rows / tuples

Data types (sample)

Only atomic types - no sets / lists / maps…

- Characters: `char, varchar, text…`

- Numbers: `bit, int, bigint, float…`

- Time: `date, datetime, timestamp …`

✳ Each DB (MySQL, SQLServer…) has a slightly different implementation

Table schema

users

user_id	name	city	brithdate
101	Rubi Boim	Tel Aviv	<null>
102	Tova Milo	Tel Aviv	<null>
103	Lebron James	Los Angeles	30/12/1984
104	Michael Jordan	Chicago	17/02/1963

`users(user_id, name, city, birthdate)`

items

item_id	title	company	price
2003	iPad	Apple	$499
2004	iPhone	Apple	$899
2005	55' LED TV	Samsung	$1549
2006	USB charger	Logitech	$29

Discussion:
can the title or company
act as the key?

`items(item_id, title, company, price)`

SQL as API

users

user_id	name	city	brithdate
101	Rubi Boim	Tel Aviv	<null>
102	Tova Milo	Tel Aviv	<null>
103	Lebron James	Los Angeles	30/12/1984
104	Michael Jordan	Chicago	17/02/1963

```
SELECT user_id, name
FROM users
WHERE city = "Tel Aviv"
ORDER BY name
```
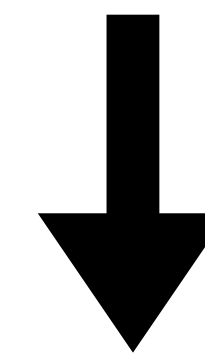
user_id	name
101	Rubi Boim
102	Tova Milo

Data integrity in RDBMS

- Referential integrity support
  primary and foreign keys

- ACID transactions support
  Atomicity, **C**onsistency, **I**solation, **D**urability

- **One of the best features of RDBMS**
  compared to NoSQL

RDBMS is a swiss pocket knife

RDBMS

You can implement almost anything with it.
But sometimes it is better to use a dedicated tool