

CMPE 131

Software Engineering

September 14, 2017

Database Introduction

Ruby on Rails ORM

Presented By
Melvin Ch'ng



Agenda

- Database Management System (DBMS)
- SQL vs NoSQL
- Relational Database Introduction
- Data Modeling Tool
- Rails Object Relational Mapping (ORM)
- Ruby on Rails & Installation
- Q&A

What is a Database Management System (DBMS)?

- A database management system (DBMS) is system software for creating and managing databases.
 - A big program that someone wrote that accesses and updates the files for you
- The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

Source: <http://searchsqlserver.techtarget.com/definition/database-management-system>

Why we need a DBMS?

- Suppose we are building a system to store the information about the following
 - Students
 - Courses
 - Professors
- Can we do it without a database?
 - Yes!
 - students.txt
 - courses.txt
 - professors.txt
 - What if you program crashes?

What does DBMS provides?

- Data abstraction and independence
- Data security
- A locking mechanism for concurrent access
- An efficient handler to balance the needs of multiple applications using the same data
- The ability to swiftly recover from crashes and errors
- Robust data integrity capabilities
- Logging and auditing of activity
- Simple access using a standard application programming interface (API)
- Uniform administration procedures for data

Source: <http://searchsqlserver.techtarget.com/definition/database-management-system>

SQL and NoSQL

- SQL
 - “Structured Query Language”
 - Use SQL commands to create, delete, modify, and query database structures
 - Use a conceptual data modeling technique
 - ER Modelling
- NoSQL
 - “Not only SQL”
 - A mechanism to store and retrieve data that uses a model other than the tabular relations used in relational databases.
 - Big data and real-time web applications

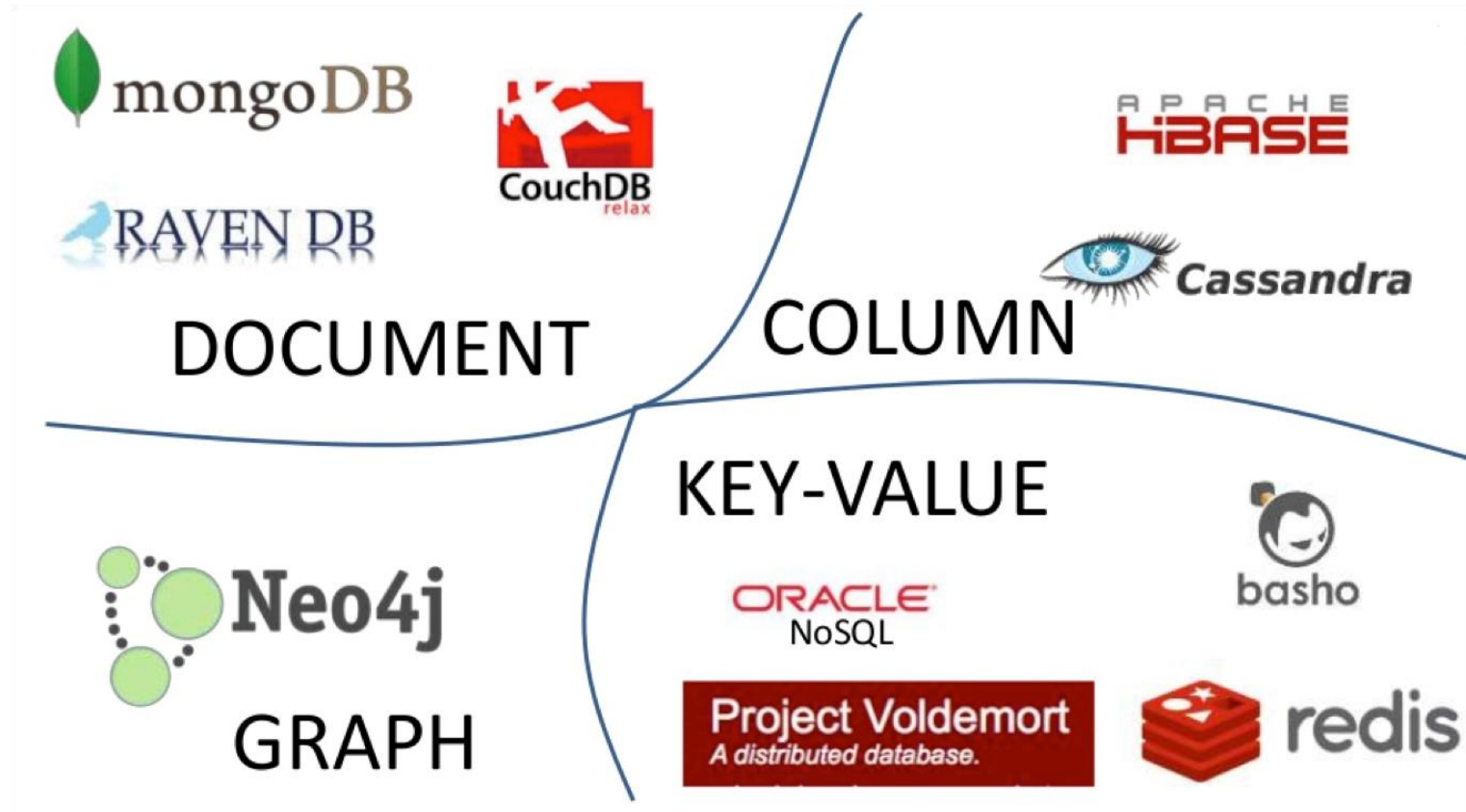
Source: Prof R.Mak, Database System Workshop

SQL vs NoSQL

	SQL	NoSQL
Types	SQL database	Key-value, graph, wide-column, document
Development history	Early 1970s	Late 2000s
Data model	Relational	Various
Schemas	Fixed	Dynamic
Scaling	Vertical	Horizontal
Development model	Mix of closed- and open-source	Open-source
Transaction support	Yes	At certain levels (e.g., document vs. database) depending on the application
Data manipulation	Via SQL language	Via object-oriented API
Consistency	Can be strong	Depends on the database product

Source: Prof R.Mak, Database System Workshop

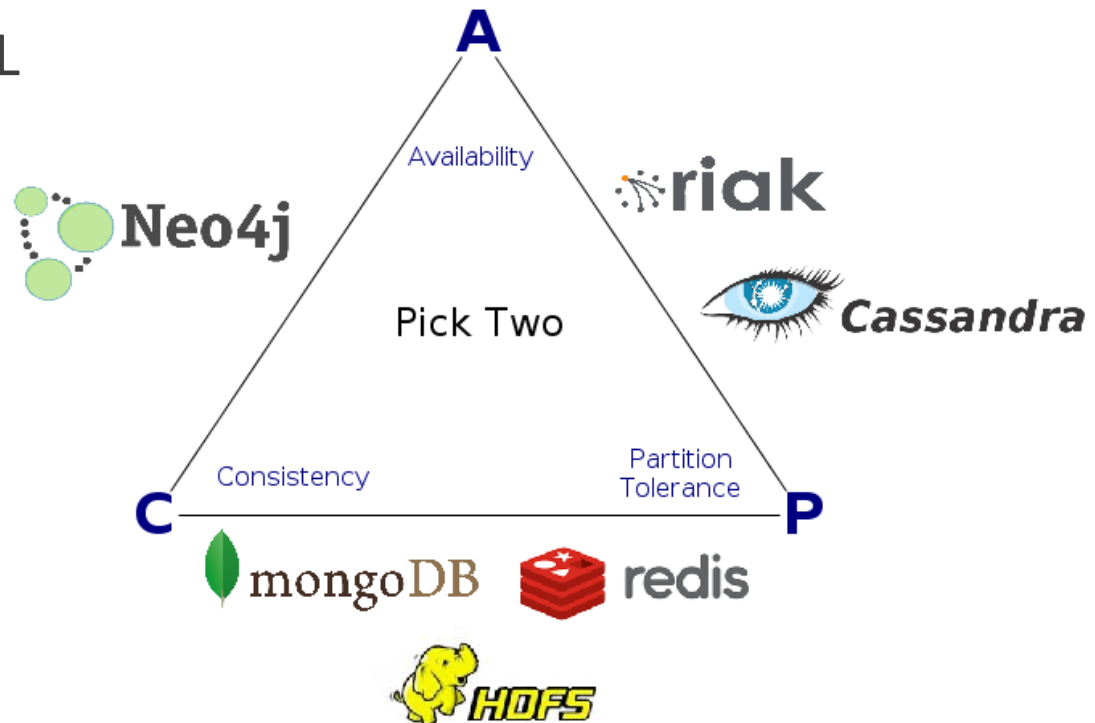
Types of NoSQL Database



Source: Prof R.Mak, Database System Workshop

CAP Theorem

- Consistency, availability, partition tolerance:
You can only choose two out of three.
- Different NoSQL databases emphasize different pairs of CAP.

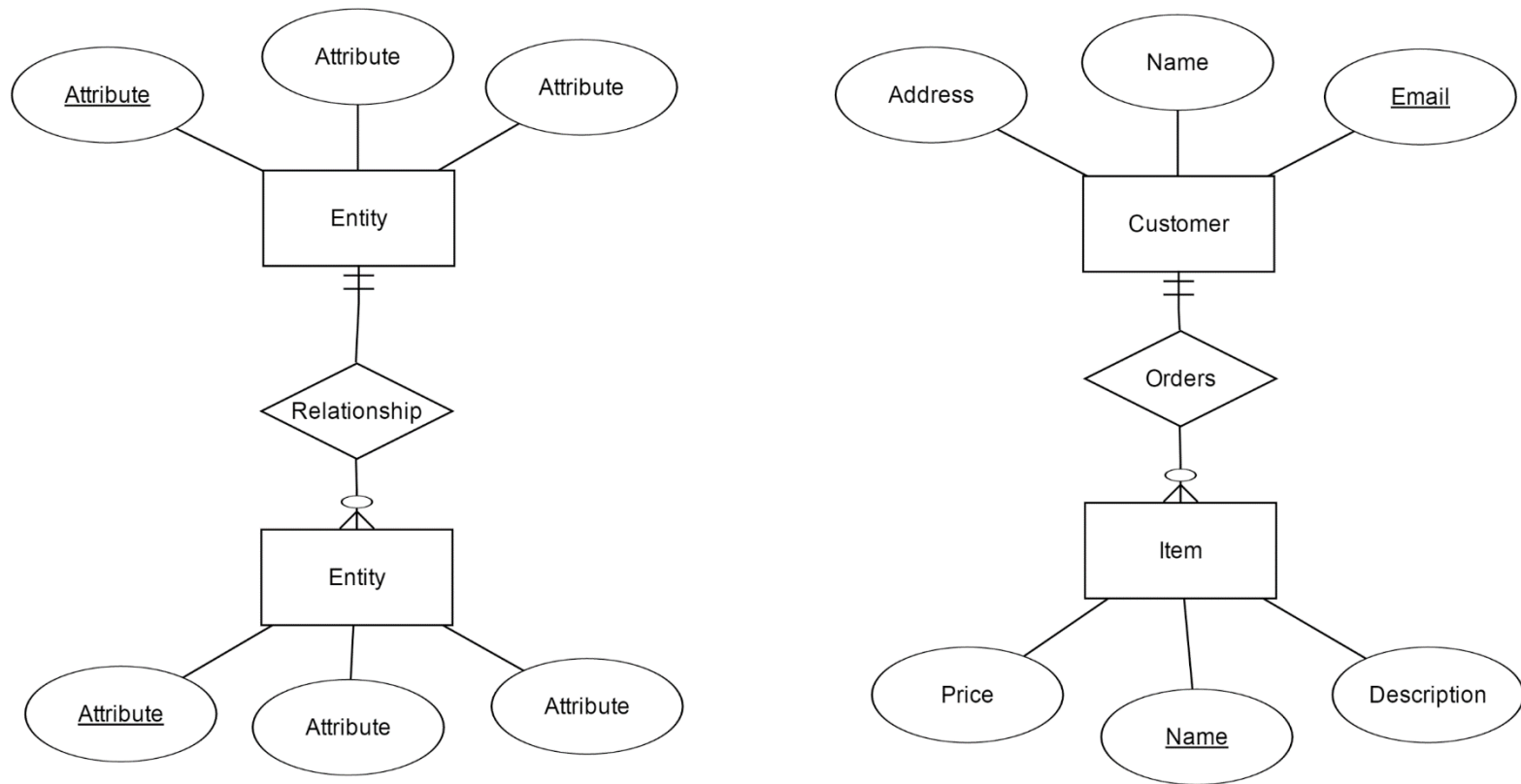


Source: Prof R.Mak, Database System Workshop

Relational Database Introduction

- Entity Relationship & Relation Schema Diagram
- Keys
 - Primary Key (PK) – unique and underline
 - Foreign Key (FK) – use for cross-reference among tables
- SQL Operators
 - Union, Intersection, Difference, Join, ...
- Normalization
 - 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, DKNF
- ERDPlus, Data modeling tool to create diagrams
 - <https://erdplus.com/>

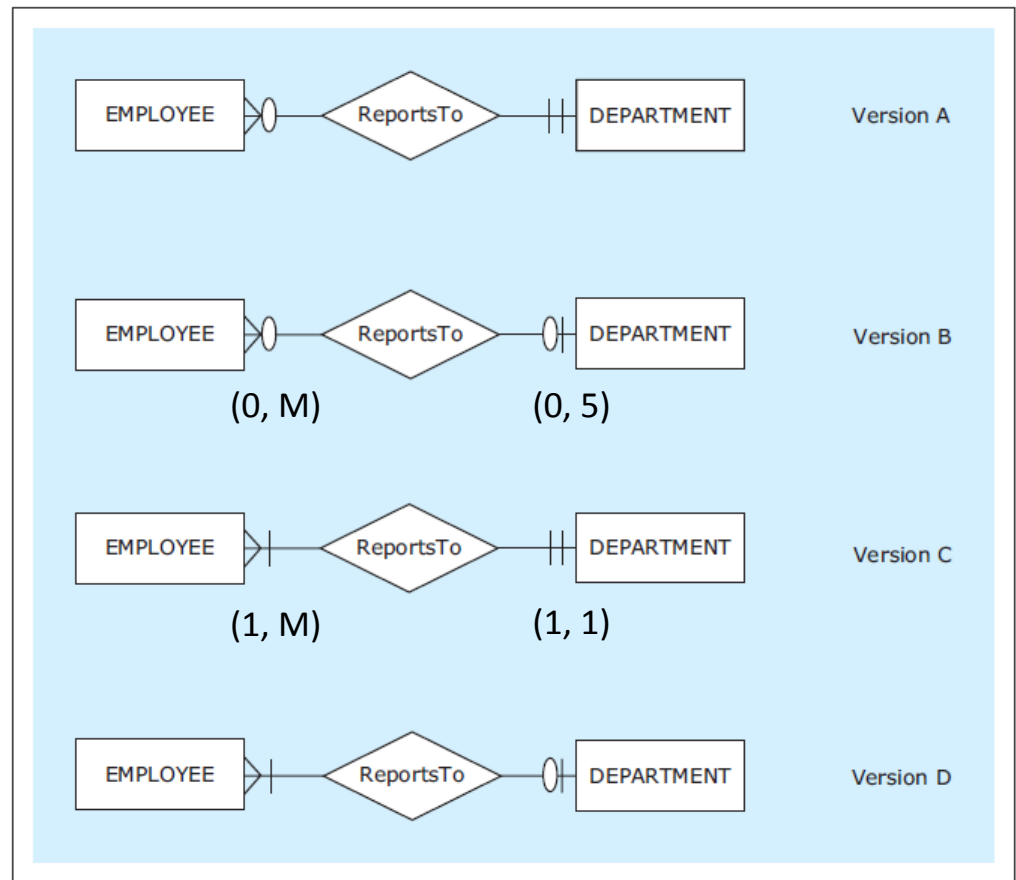
Entity Relationship (ER) Diagram



Source: <http://melvinchng.github.io/rails/BeforeWeBegin.html#14-relational-database-basics>

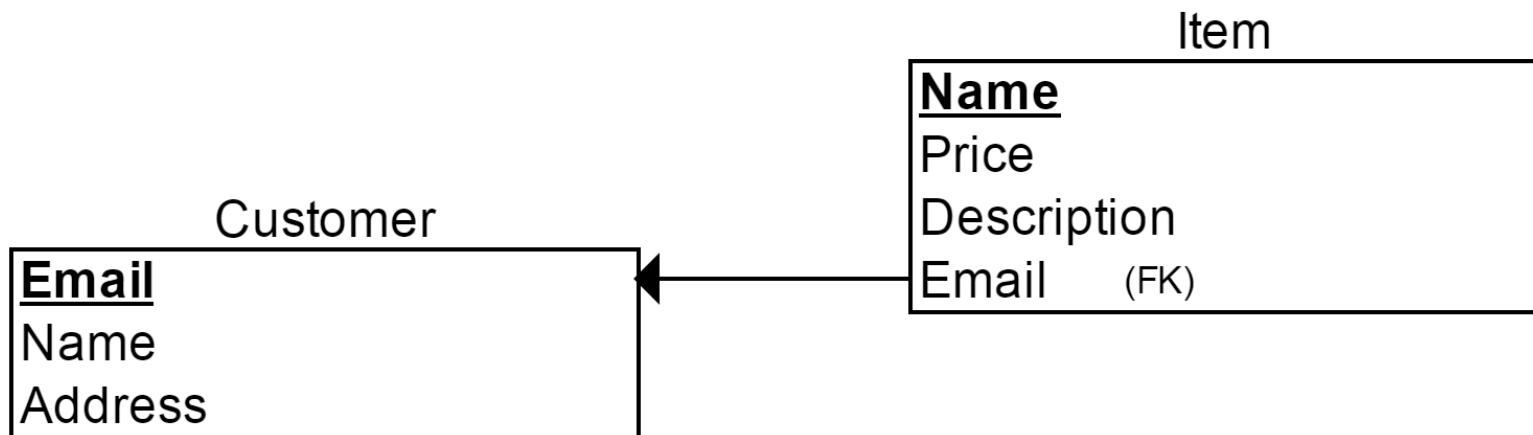
Relationship Cardinality

- Relationship Cardinality
 - One to One
 - One to Many
 - Many to One
 - Many to Many
- Read from left to right or right to left



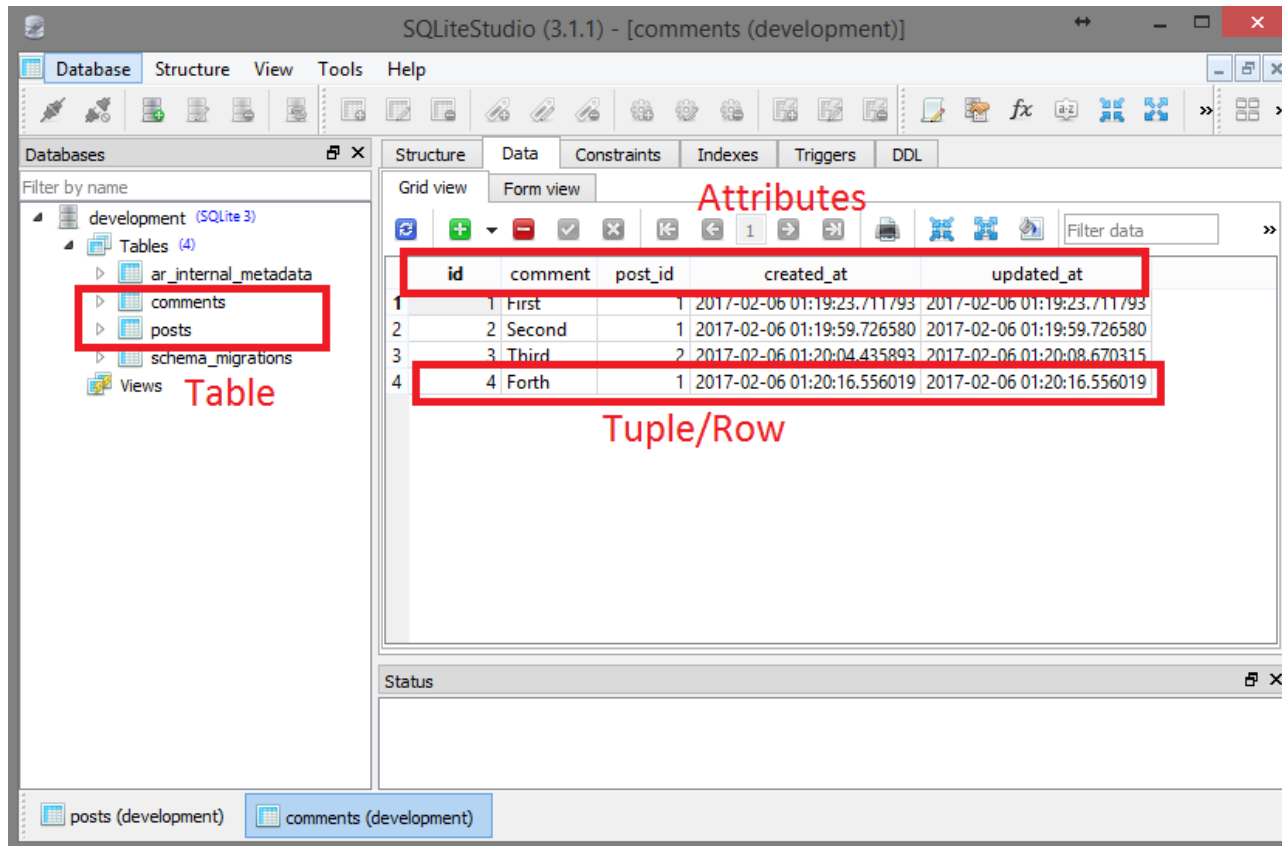
Source: Prof R.Mak, Database System Workshop

Relational Schema Diagram



Source: <http://melvinchng.github.io/rails/BeforeWeBegin.html#14-relational-database-basics>

Database Manager: SQLiteStudio



Source: <http://melvinchng.github.io/rails/BeforeWeBegin.html#14-relational-database-basics>

Normalization

- Improve the design of database tables.
 - Eliminate update anomalies.
- Three normal forms.
 - First normal form (1NF)
 - Second normal form (2NF)
 - Third normal form (3NF)
- From lower to higher, each normal form has increasingly stricter conditions.
 - Even higher normal forms mostly of theoretical value.
 - Boyce-Codd (BCNF), 4NF, 5NF, domain key (DKNF).

Source: Prof R.Mak, Database System Workshop

Normalization, *cont'd*

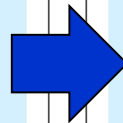
- 1NF
 - Each row is unique.
 - All values in a column must be from the same predefined domain.
 - No column in any row contains multiple values.

Source: Prof R.Mak, Database System Workshop

First Normal Form (1NF)

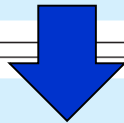
VET CLINIC CLIENT

<u>ClientID</u>	ClientName	PetNo	PetName	PetType
111	Lisa	1	Tofu	Dog
222	Lydia	1	Fluffy	Dog
		2	JoJo	Bird
		3	Ziggy	Snake
333	Jane	1	Fluffy	Cat
		2	Cleo	Cat



VET CLINIC CLIENT

<u>ClientID</u>	ClientName	<u>PetNo</u>	PetName	PetType
111	Lisa	1	Tofu	Dog
222	Lydia	1	Fluffy	Dog
222	Lydia	2	JoJo	Bird
222	Lydia	3	Ziggy	Snake
333	Jane	1	Fluffy	Cat
333	Jane	2	Cleo	Cat



VET CLINIC CLIENT

<u>ClientID</u>	ClientName
111	Lisa
222	Lydia
333	Jane

PET

<u>ClientID</u>	<u>PetNo</u>	PetName	PetType
111	1	Tofu	Dog
222	1	Fluffy	Dog
222	2	JoJo	Bird
222	3	Ziggy	Snake
333	1	Fluffy	Cat
333	2	Cleo	Cat

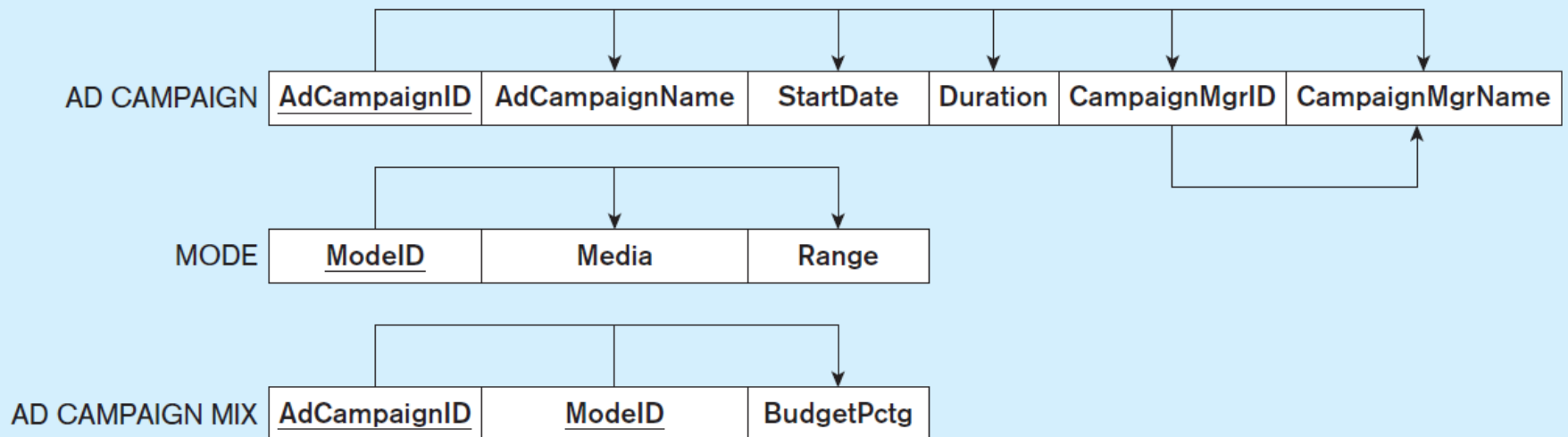
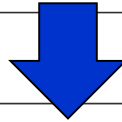
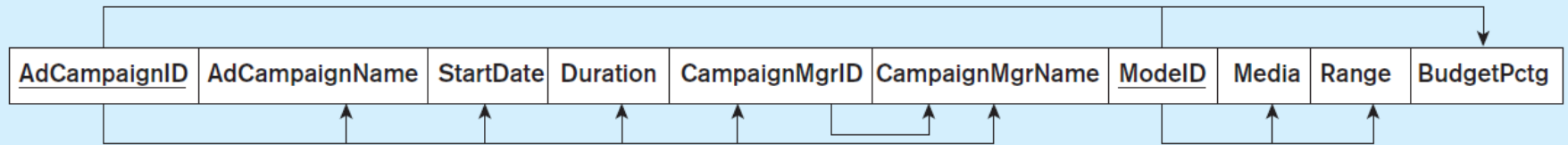
Source: Prof R.Mak, Database System Workshop

Normalization, *cont'd*

- 1NF
 - Each row is unique.
 - All values in a column must be from the same predefined domain.
 - No column in any row contains multiple values.
- 2NF
 - It is in 1NF.
 - It does not contain partial functional dependencies.

Source: Prof R.Mak, Database System Workshop

Second Normal Form (2NF)



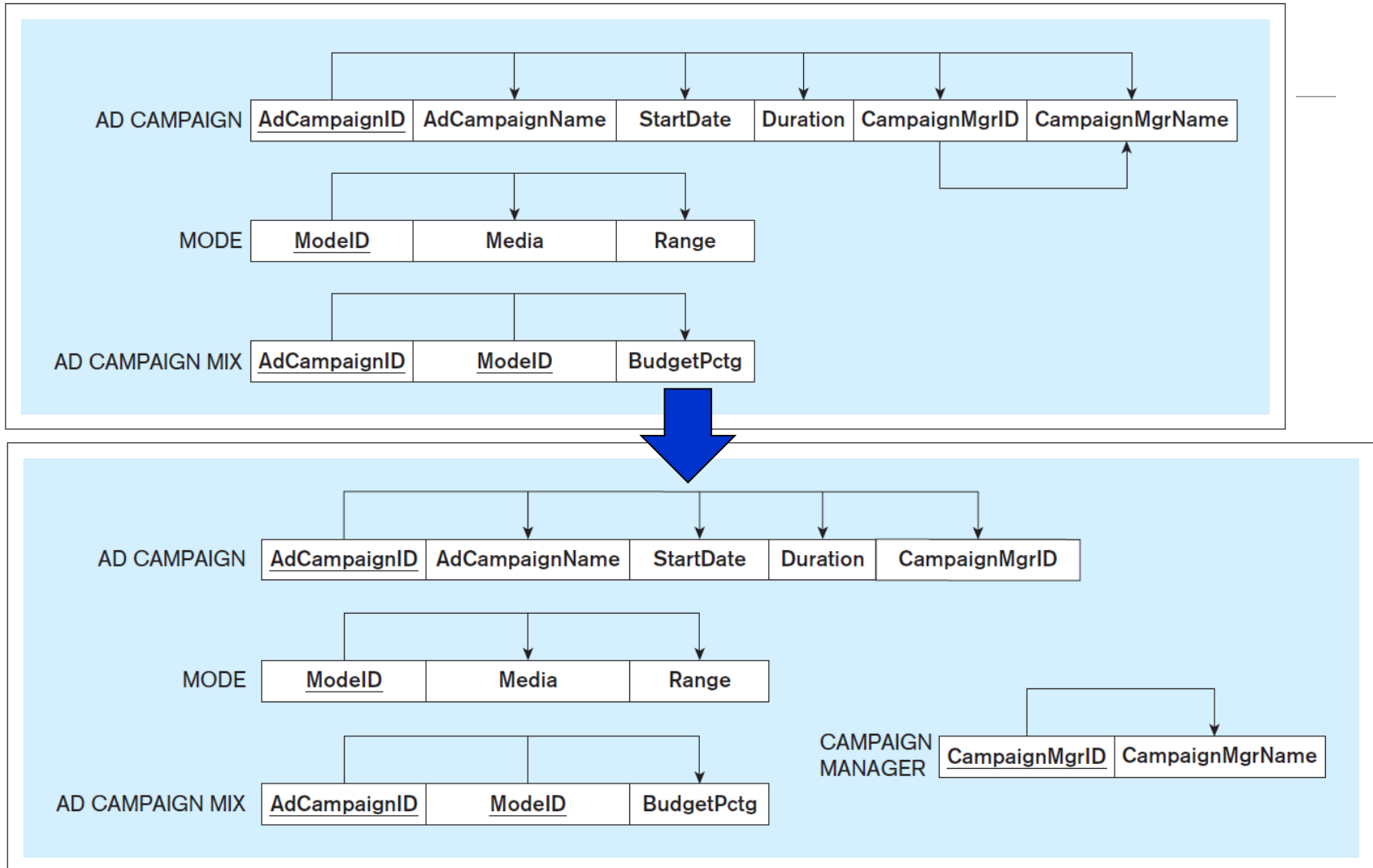
Source: Prof R.Mak, Database System Workshop

Normalization, *cont'd*

- 1NF
 - Each row is unique.
 - All values in a column must be from the same predefined domain.
 - No column in any row contains multiple values.
- 2NF
 - It is in 1NF.
 - It does not contain partial functional dependencies.
- 3NF
 - It is in 2NF.
 - It does not contain transitive functional dependencies.

Source: Prof R.Mak, Database System Workshop

Third Normal Form (3NF)



Normalization vs. Denormalization

- Normalization spreads data out over more tables.
- The result is slower performance.
- Sometimes it make sense to denormalize in order to improve performance.

Source: Prof R.Mak, Database System Workshop

Data Modeling Tool

- [FREE] ERDPlus
 - <http://erdplus.com>
 - Convert ER Diagrams into Relational Schemas automatically
 - Export to standard SQL
 - MySQL, MSSQL, PgSQL, IBM DB2, etc

Source: Prof R.Mak, Database System Workshop

Rails Object Relational Mapper (ORM)

- Simple English:
 - It means you don't have to manually call the database yourself. ORM handles it for you.
 - Rails use ActiveRecord, it is very powerful!
- Follows strong conventions
- Does not require a lot of low-level access to DB
- Make sure that you have the associations setup correctly in `Model`.
 - http://guides.rubyonrails.org/association_basics.html

ORM Example

Id	first_name	last_name	age	gender
1	Julie	Lau	21	F
2	Thomas	Lin	24	M
3	Paul	Tan	26	M

- SQL

- `SELECT * FROM users;`

- Rails ORM

- `User.all`

ORM Example, *cont'd*

- SQL

- `SELECT *`
`FROM users`
`ORDER BY first_name DESC;`

- Rails ORM

- `User.order("first_name DESC")`

ORM Example, *cont'd*

- SQL

- ```
SELECT *
 FROM users
 ORDER BY first_name DESC
 LIMIT 5;
```

- Rails ORM

- ```
User.order("first_name DESC").last(5)
```

ORM Example, *cont'd*

- SQL

- ```
SELECT *
FROM users
WHERE gender = 'm'
ORDER BY first_name DESC
LIMIT 5;
```

- Rails ORM

- ```
User.where("gender = 'm'")  
  .order("first_name DESC")  
  .last(5)
```

ORM Example, *cont'd*

id	first_name	...	id	title	...	user_id
1	Julie	...	1	Hello World	...	1
2	Thomas	...	2	Hello Hello	...	2
3	Paul	...	3	Hello Again	...	2

- SQL

- `SELECT users.*
FROM users
INNER JOIN posts
ON posts.user_id = users.id AND ...`

- ORM

- `User.joins(:posts)`

Ruby on Rails & Installation

- To avoid problems, make sure everyone in the team is
 - Using the same Ruby version, if possible
 - Pick either <Ruby 2.4 or >=Ruby 2.4.1
- Installation guides and videos available
 - Windows, Mac, and Linux
 - <http://melvinchng.github.io/rails/RubyOnRailsInstallation.html>
- Read the documentations and Google is your friend!
- If your RoR does not work, talk to me after the class

Ruby on Rails & Installation

