

# Database System

## Normalization & Final Project Introduction

# Agenda

- Normalization
  - First Normal Form
  - Second Normal Form
  - Third Normal Form
- Final Project Introduction
  - RESTful API & Swagger
  - Postman
  - JWT
  - About Project

# Normalization

# **What is Normalization?**

# **What is Normalization?**

**Divides larger table to smaller table and  
links them using relationship**

# Why Normalization?

# Why Normalization?

**Eliminate redundant, avoid inconsistency and store data logically.**

# First Normal Form

- primary key
- no multivalued attribute



# First Normal Form

- primary key
- no multivalued attribute

ssn	name	item	price	amount	total	date
123456789	Raymond	MacBook Pro, iPad	59900, 10500	1, 1	70400	11/26
987654321	Paul	MacBook Air, iPad Pro	30900, 24900	1, 1	55800	11/11

<u>ssn</u>	name	<u>item</u>	price	amount	total	<u>date</u>
123456789	Raymond	MacBook Pro	59900	1	59900	11/26
123456789	Raymond	iPad	10500	1	10500	11/26
987654321	Paul	MacBook Air	30900	1	30900	11/11
987654321	Paul	iPad Pro	24900	1	24900	11/11

# Second Normal Form

- satisfy First Normal Form
- no partial dependency

# Second Normal Form

- satisfy First Normal Form
- no partial dependency

<u>ssn</u>	name	<u>item</u>	price	amount	total	<u>date</u>
123456789	Raymond	MacBook Pro	59900	1	59900	11/26
123456789	Raymond	iPad	10500	1	10500	11/26
987654321	Paul	MacBook Air	30900	1	30900	11/11
987654321	Paul	iPad Pro	24900	1	24900	11/11

<u>ssn</u>	name
123456789	Raymond
987654321	Paul

<u>item</u>	price
MacBook Pro	59900
iPad	10500
MacBook Air	30900
iPad Pro	24900

<u>ssn</u>	<u>item</u>	amount	total	<u>date</u>
123456789	MacBook Pro	1	59900	11/26
123456789	iPad	1	10500	11/26
987654321	MacBook Air	1	30900	11/11
987654321	iPad Pro	1	24900	11/11

# Third Normal Form

- satisfy Second Normal Form
- no transitive dependency

# Third Normal Form

- satisfy Second Normal Form
- no transitive dependency

<u>ssn</u>	<u>item</u>	amount	total	<u>date</u>
123456789	MacBook Pro	1	59900	11/26
123456789	iPad	1	10500	11/26
987654321	MacBook Air	1	30900	11/11
987654321	iPad Pro	1	24900	11/11

<u>ssn</u>	<u>item</u>	amount	<u>date</u>
123456789	MacBook Pro	1	11/26
123456789	iPad	1	11/26
987654321	MacBook Air	1	11/11
987654321	iPad Pro	1	11/11

# Third Normal Form

- satisfy Second Normal Form
- no transitive dependency

<u>id</u>	customer_id	item_id	amount	date
1	1	1	1	11/26
2	1	2	1	11/26
3	2	3	1	11/11
4	2	4	1	11/11

<u>id</u>	ssn	name
1	123456789	Raymond
2	987654321	Paul

<u>id</u>	item	price
1	MacBook Pro	59900
2	iPad	10500
3	MacBook Air	30900
4	iPad Pro	24900

**Must Normalization?**

# **Must Normalization?**

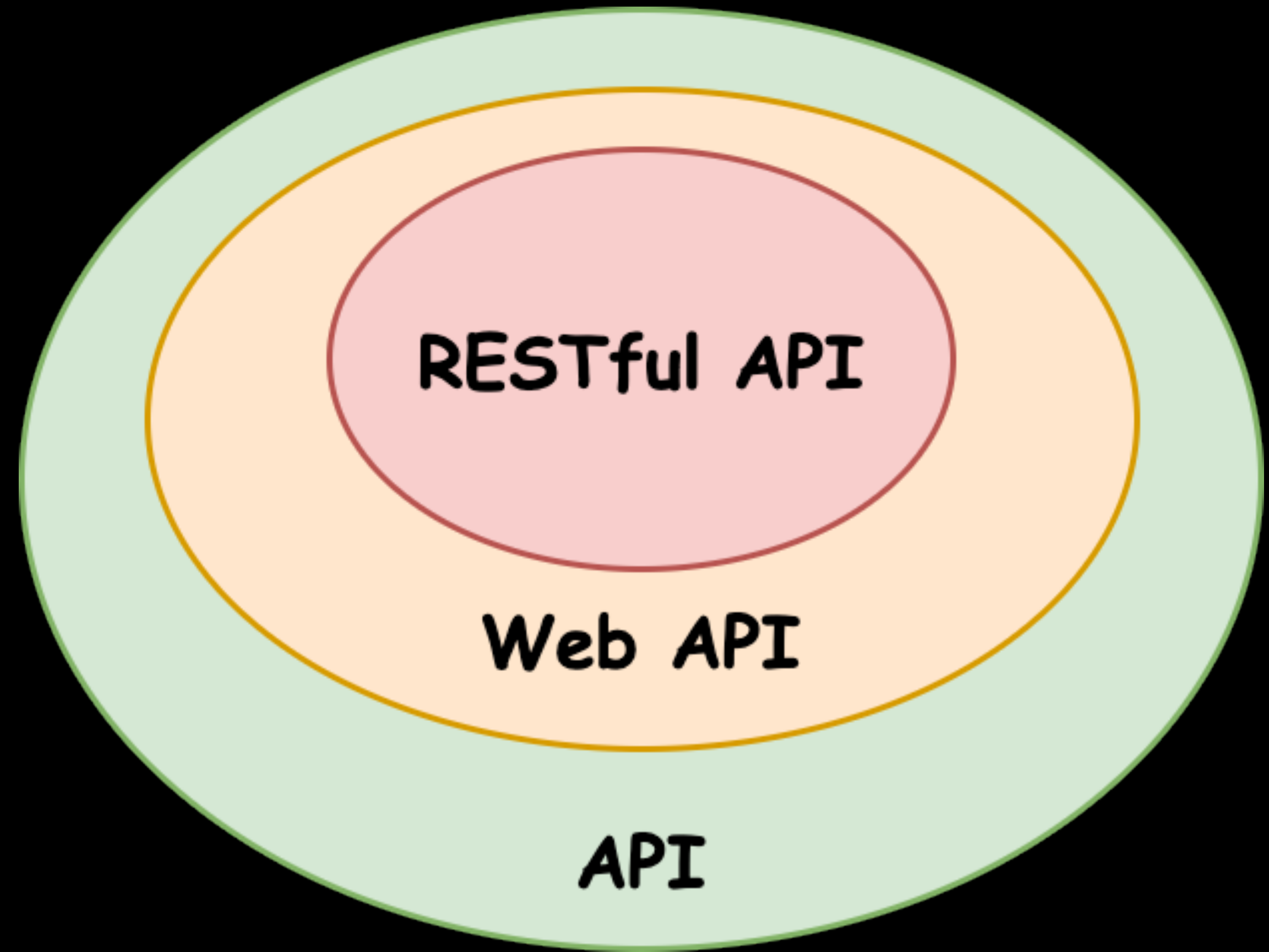
**It depends. (JSON, XML support and rarely modified column)**



# Final Project Introduction

# RESTful API

Popular Software Architectural Style



# RESTful API

- REST = REpresentational State Transfer
- noun (resource)
  - `/ {resource} / ...`
- verb (HTTP method)
  - POST (create)
  - GET (read)
  - PUT (update/replace), PATCH (update/modify)
  - DELETE (delete)

# Swagger

## RESTful API Document



Swagger™

Supported by SMARTBEAR

# Postman

## Web API Testing Tool



# JWT



- JWT = JSON Web Token
- structure: xxxx.yyyy.zzzz
- header: Base64 encoded to form the first part (xxxx)
- payload: Base64 encoded to form the second part (yyyy)
- signature: encode xxx.yyyy by the declared algorithm in header with self defined key to form the third part (zzzz)

# About Project

- Database: SQLite
- API Implementation: RESTful API (ref: openapi.yml)
- Authorization & Authentication: JWT
- System Optimization: Correctness and Availability
- Front-End: optional

**Thanks for Your Attention.**