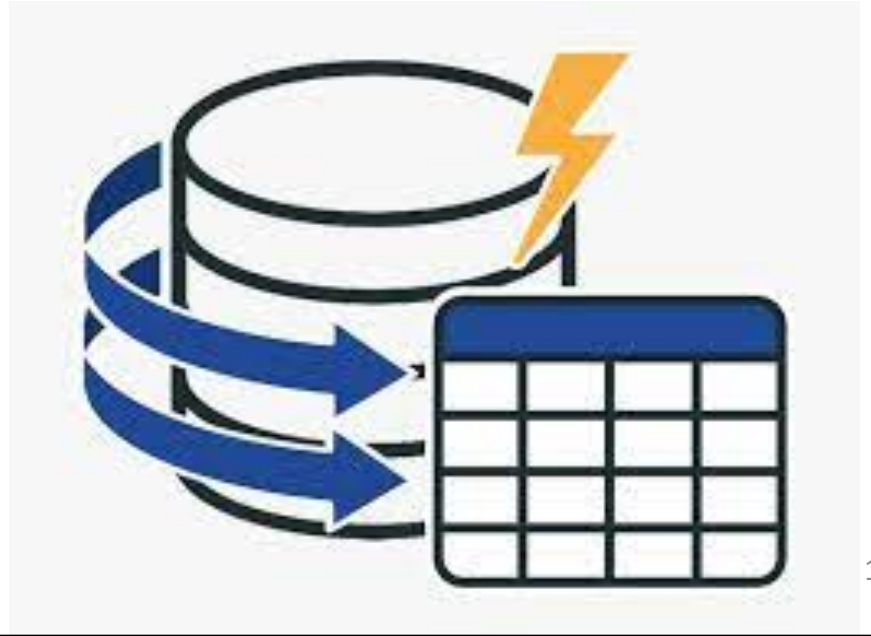
Topic 05:
Data Model Diagram

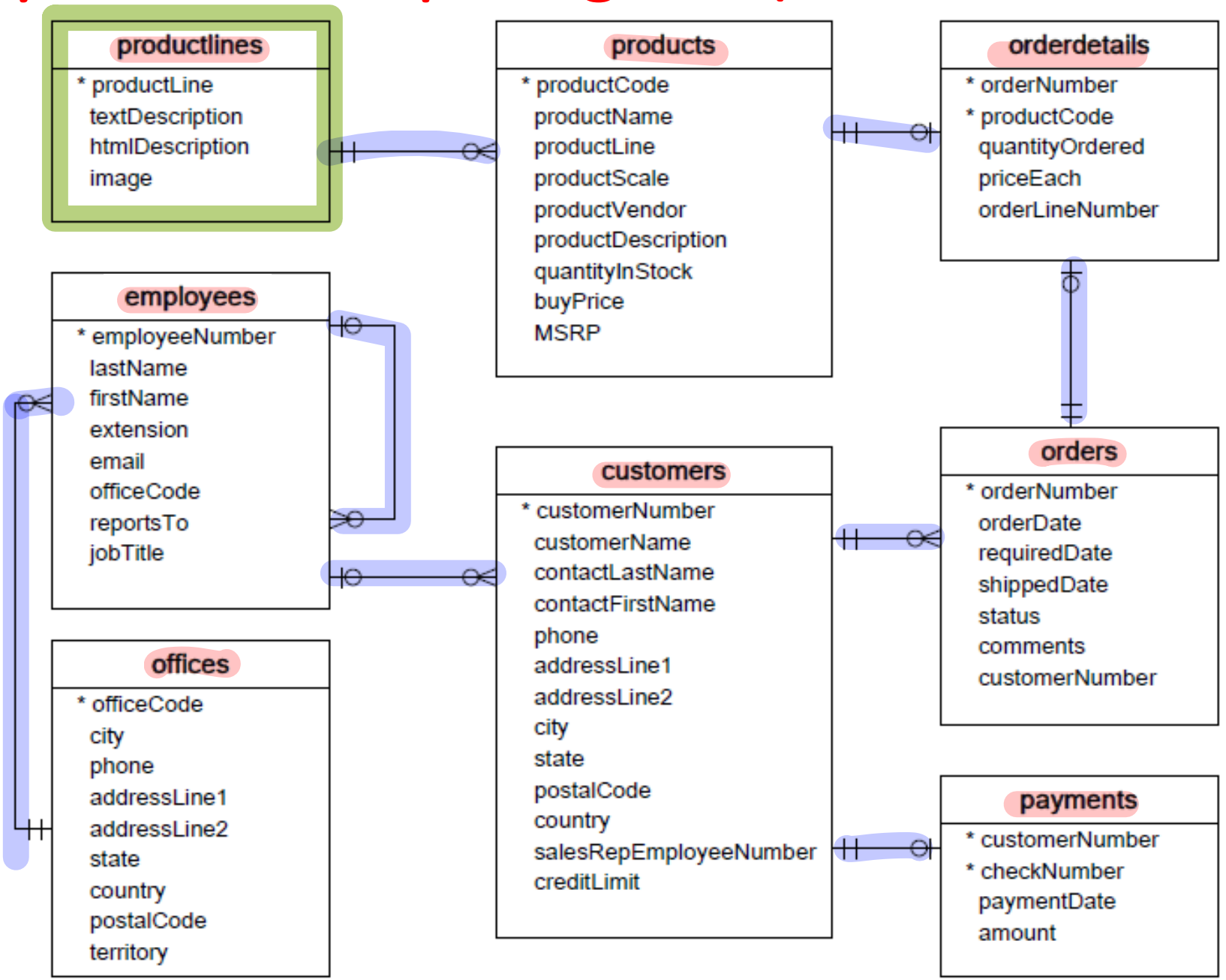
BDM3302: Data Management

Data Models

- There are three main popular data models which are
  - Entity Relationship Diagrams (E-R Diagrams)
  - Unified Modeling Language (UML)
  - Data Dictionary (will be used in DBM Tool Lab Class)

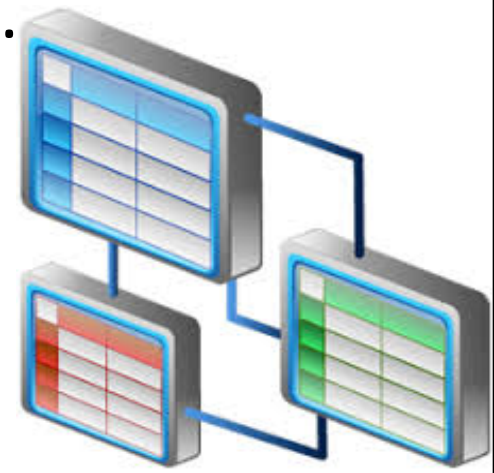


Entity Relationship Diagram (Classic Models)

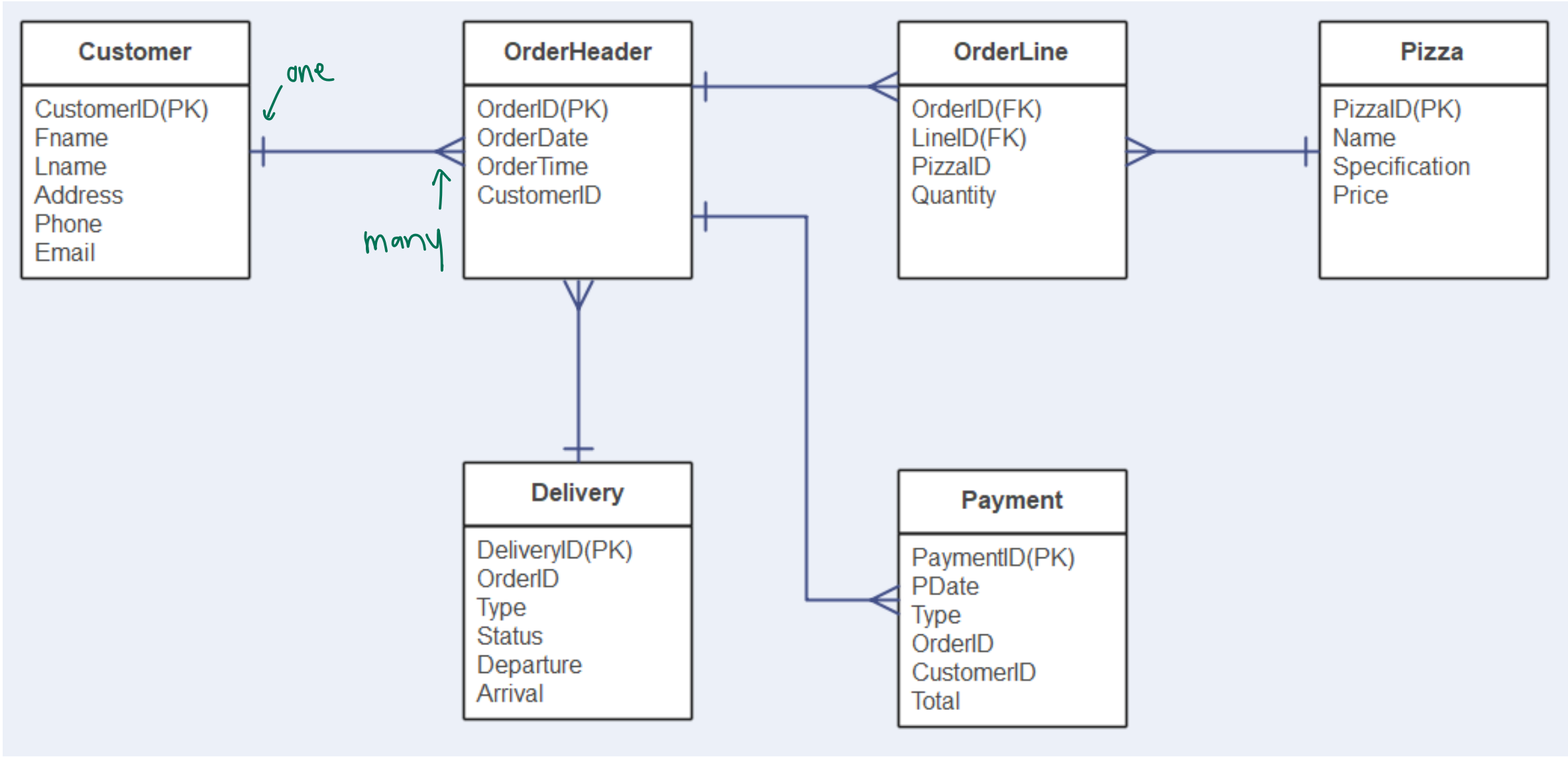


Entity Relationship Diagrams (E-R Diagrams)

- Entity-Relationship modeling is a default technique for modeling and the design of relational (traditional) databases. In this notation architect identifies:
- **Entities** representing objects (or tables in relational database),
- **Attributes** of entities including **data type,**
- **Relationships** between entities/objects (or foreign keys in a database).
- E-R Diagrams work well if you want to design a relational (classic) database, Excel databases or CSV files (tabular data). They work well for visualization of database schemas and communication of top-level view of data.



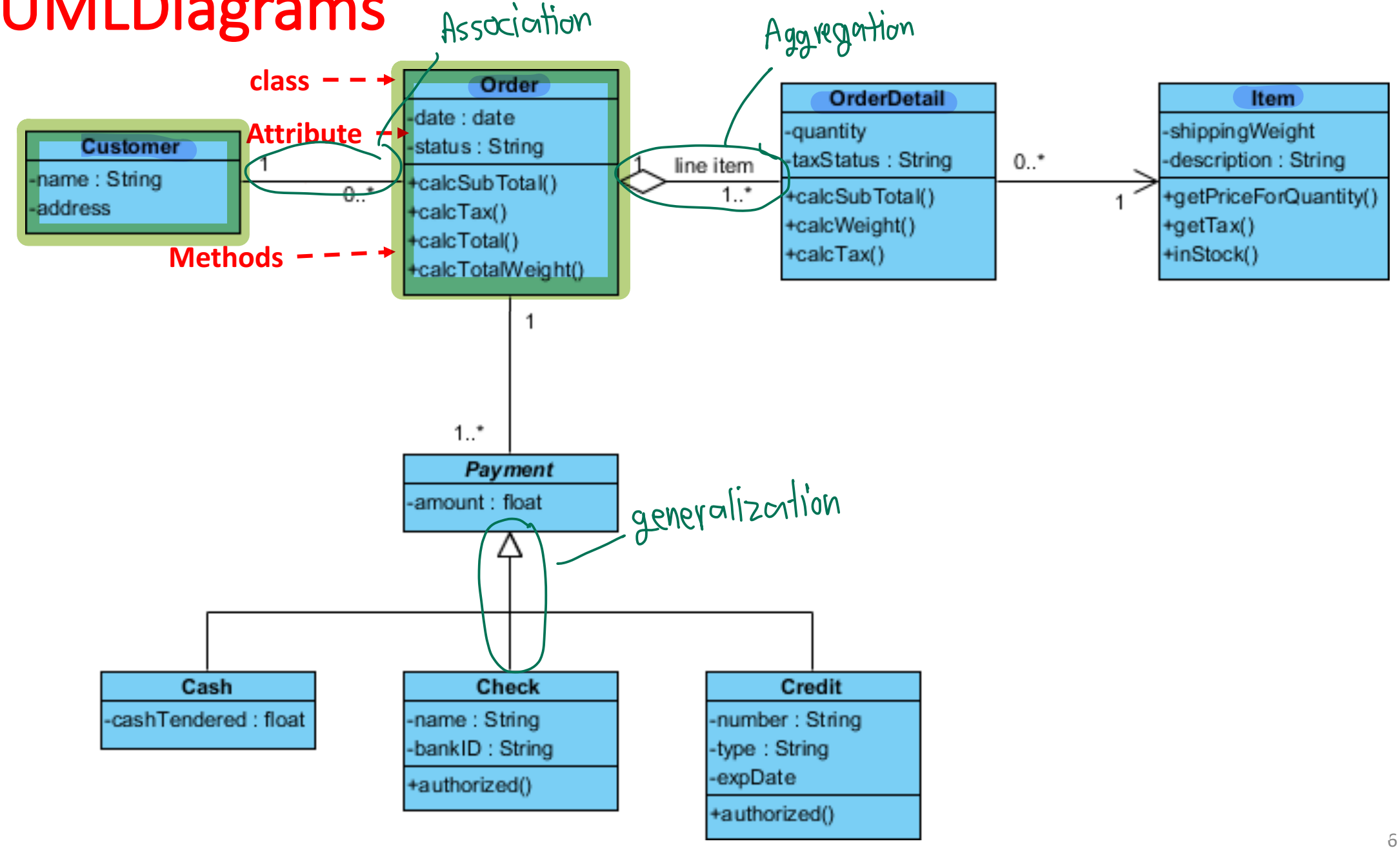
Entity Relationship Diagrams (E-R Diagrams)



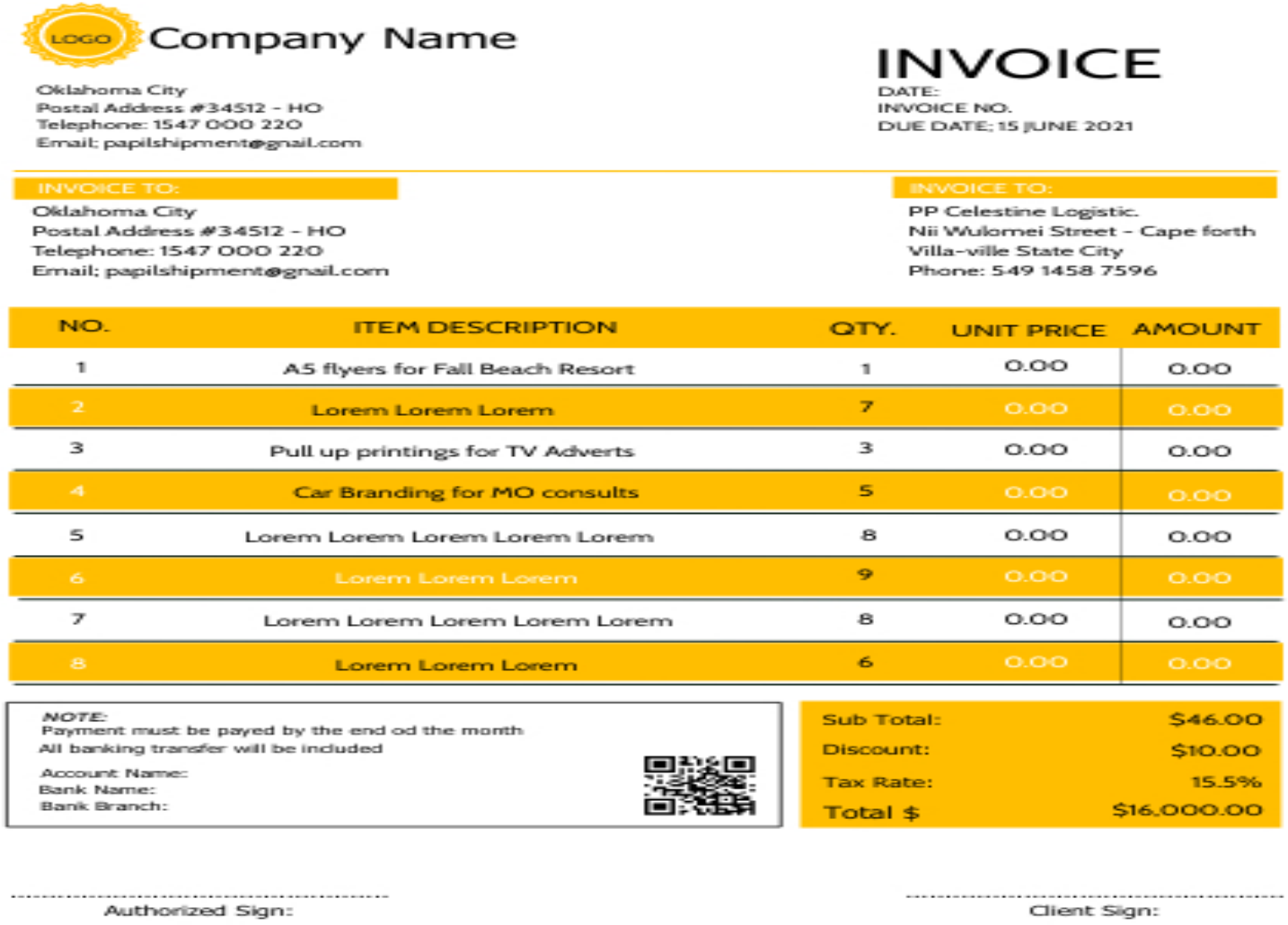
UML Class Diagrams

- **UML** (Unified Modeling Language) is a standardized family of notations for modeling and design of information systems. It was derived from various existing notations to provide a standard for *software engineering or software developers*.
- Class diagrams are equivalent of ERDs in relational world and are mostly used to design classes in object-oriented programming languages (such as Java or C#).
- In class diagrams architects define:
- **Classes** (equivalent of entity in relational world),
- **Attributes** of a class (same as in an ERD) including **data type**,
- **Methods** associated to specific class, representing its behavior (in relational world those would be stored procedures),
- **Relationships** grouped into two categories:
  - **Relationships between objects** (instances of Classes) differentiated into Dependency, Association, Aggregation and Composition (equivalent to relationships in an ERD),
  - **Relationships between classes** of two kinds Generalization/Inheritance and Realization/Implementation (this has no equivalent in relational world).

UMLDiagrams



Business Document Sample (Invoice)



LOGO **Company Name**

Oklahoma City
Postal Address #34512 - HO
Telephone: 1547 000 220
Email: papilshipment@gnail.com

INVOICE

DATE:
INVOICE NO.
DUE DATE: 15 JUNE 2021

INVOICE TO:
Oklahoma City
Postal Address #34512 - HO
Telephone: 1547 000 220
Email: papilshipment@gnail.com

INVOICE TO:
PP Celestine Logistic.
Nii Wulomei Street - Cape forth
Villa-ville State City
Phone: 549 1458 7596

NO.	ITEM DESCRIPTION	QTY.	UNIT PRICE	AMOUNT
1	A5 flyers for Fall Beach Resort	1	0.00	0.00
2	Lorem Lorem Lorem	7	0.00	0.00
3	Pull up printings for TV Adverts	3	0.00	0.00
4	Car Branding for MO consults	5	0.00	0.00
5	Lorem Lorem Lorem Lorem Lorem	8	0.00	0.00
6	Lorem Lorem Lorem	9	0.00	0.00
7	Lorem Lorem Lorem Lorem Lorem	8	0.00	0.00
8	Lorem Lorem Lorem	6	0.00	0.00

NOTE:
Payment must be payed by the end od the month
All banking transfer will be included

Account Name:
Bank Name:
Bank Branch:

Sub Total:	$46.00
Discount:	$10.00
Tax Rate:	15.5%
Total $	$16,000.00

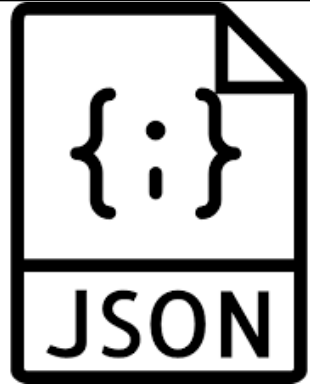
Authorized Sign:

Client Sign:

TERM & CONDITION APPLYS
Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem Lorem

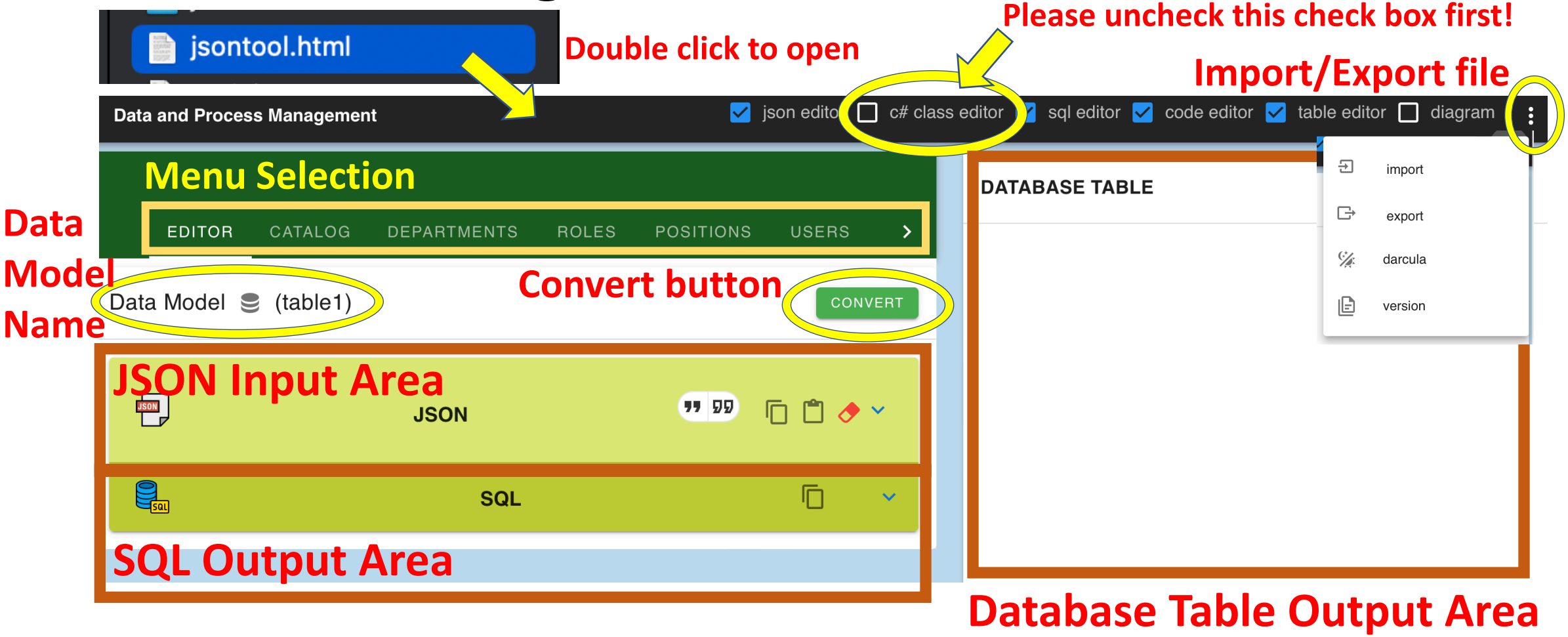
Thank You Customer for Chosing Us

Data Modeling Tool



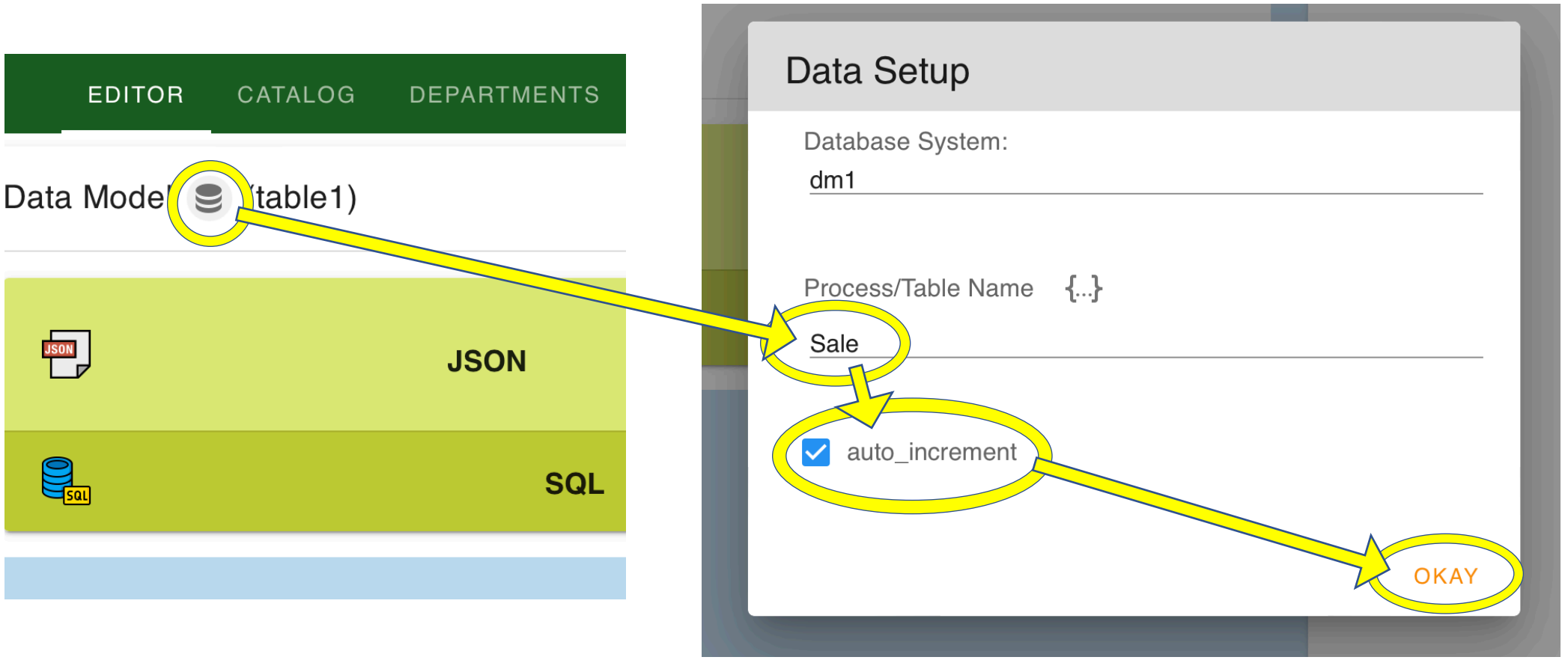
- DBM Tool v.1.2.2 by Dr. Anan Osothslip (Download from LMS and unzip file).
- It is the tool that introduces how data can be constructed and converted into the SQL and database tables.
- The input can be JSON only that process the output into SQL and database tables.
- This tool also provides catalog, departments, roles, positions, users, processes, and assets in order to identify what kind of these data to support your business **(will use in Data Accessibility lab class later).**
- The Internet connection is also required for this tool.

Data Modeling Tool Overview



Data Modeling Tool (Lab for beginners)

- First, let’s try to create Data Setup from “table1” to ”Sale”.



Data Modeling Tool (Lab for beginners)

- Second, try to input sample JSON how customer orders food.

Data Modeling Tool (Lab for beginners)

- JSON structure explains …
  - Stands for JavaScript Object Notation
  - It is a text format for storing and transporting data
  - It is a “self-describing” and easy to understand
- The example looks like this:

```
{
    “name”: “John”,
    “age”: 30,
    “car”: “Toyota”
}
```

“Mr. John with age of 30 years old has got a Toyota car.”

It identifies with 3 properties:

1. name = John (text)
2. age = 30 (integer)
3. car = Toyota (text)

*Each property has a value

Data Modeling Tool (Lab for beginners)

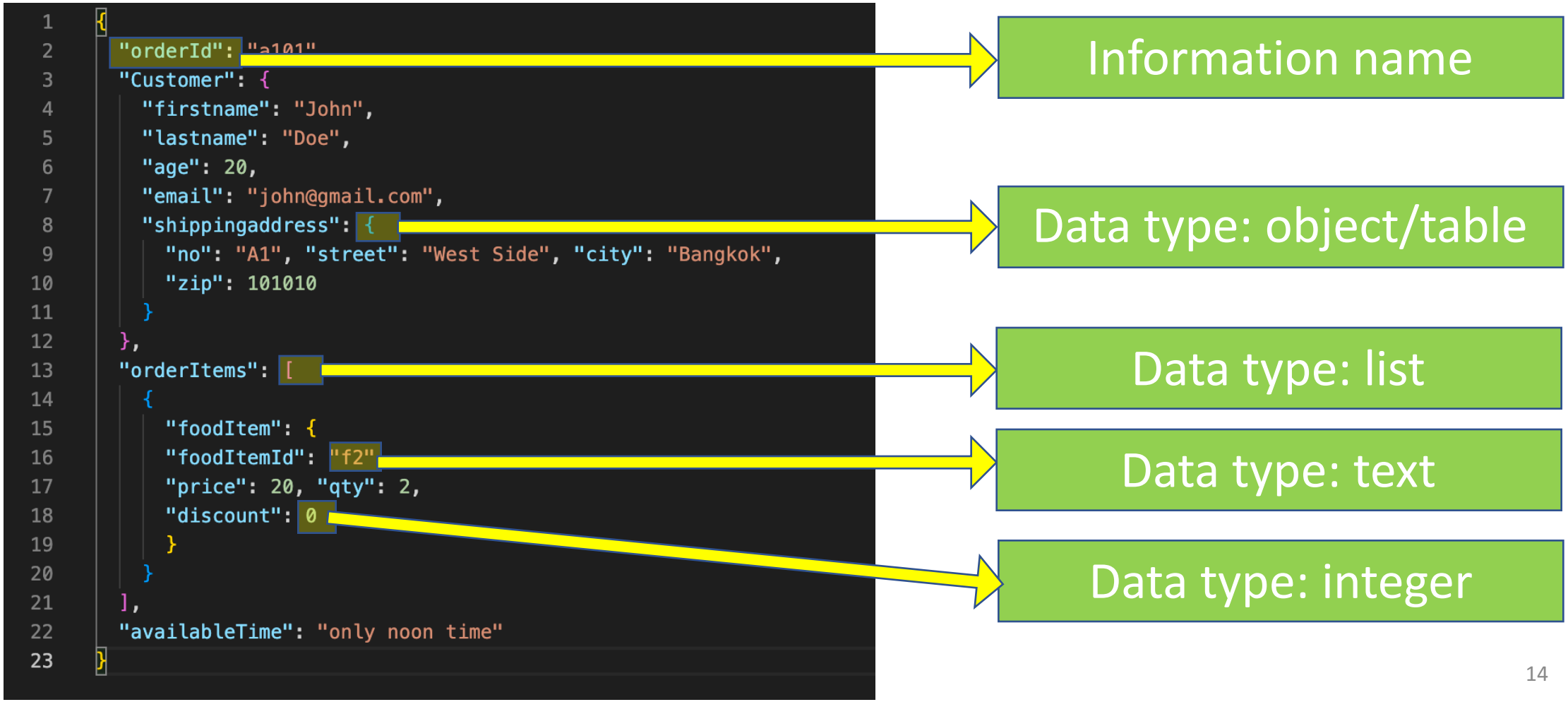
- JSON structure explains …

```json
{
  "orderId": "a101",
  "Customer": {
    "firstname": "John",
    "lastname": "Doe",
    "age": 20,
    "email": "john@gmail.com",
    "shippingaddress": {
      "no": "A1", "street": "West Side", "city": "Bangkok",
      "zip": 101010
    }
  },
  "orderItems": [
    {
      "foodItem": {
      "foodItemId": "f2",
      "price": 20, "qty": 2,
      "discount": 0
      }
    }
  ],
  "availableTime": "only noon time"
}
```

"Mr. John Doe would like to order two f2 foods for price $20 each and deliver to his home only noon time."

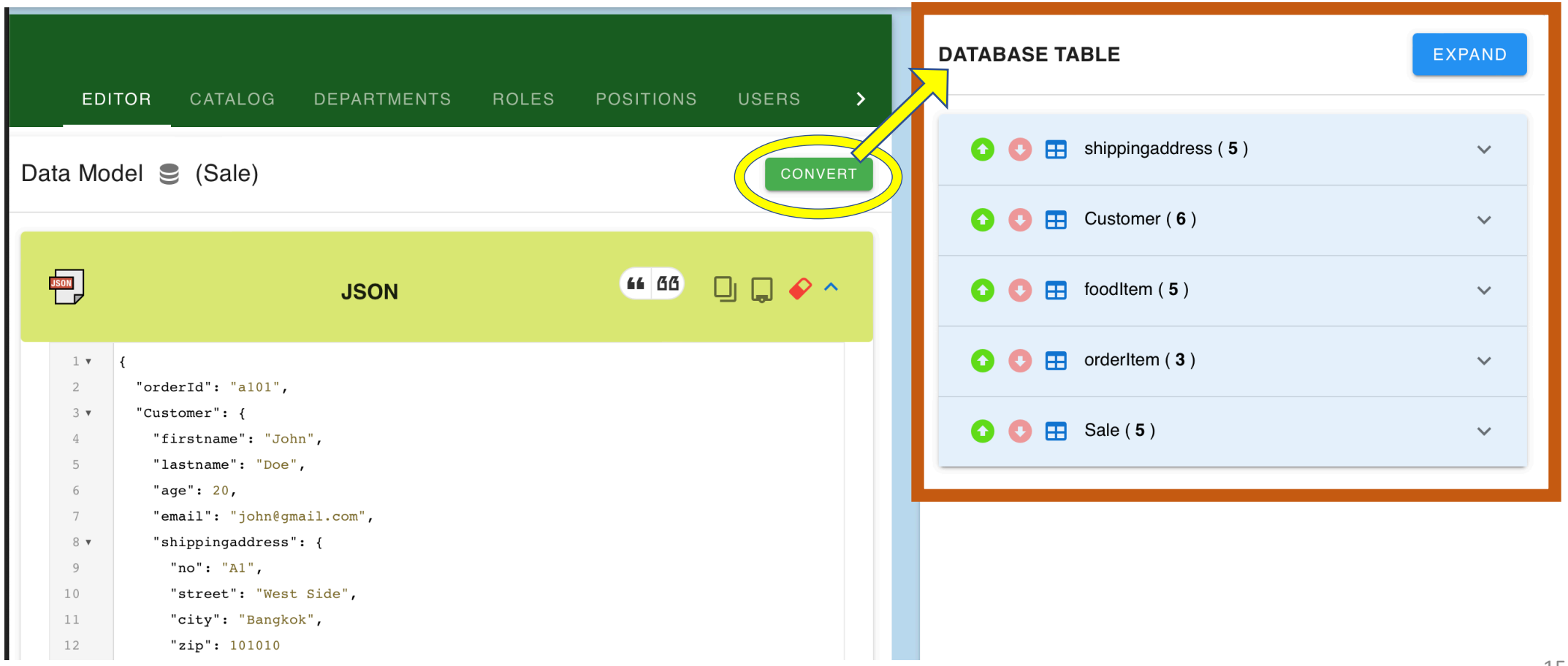
Data Modeling Tool (Lab for beginners)

- JSON structure explains …



Data Modeling Tool (Lab for beginners)

- Third, try to click convert button and see the result.



Data Modeling Tool (Lab for beginners)

- Somehow, you will see SQL appear when JSON is converted. You may bring SQL to create **.sql** file and try to import into **“mysql”** also (Empty).

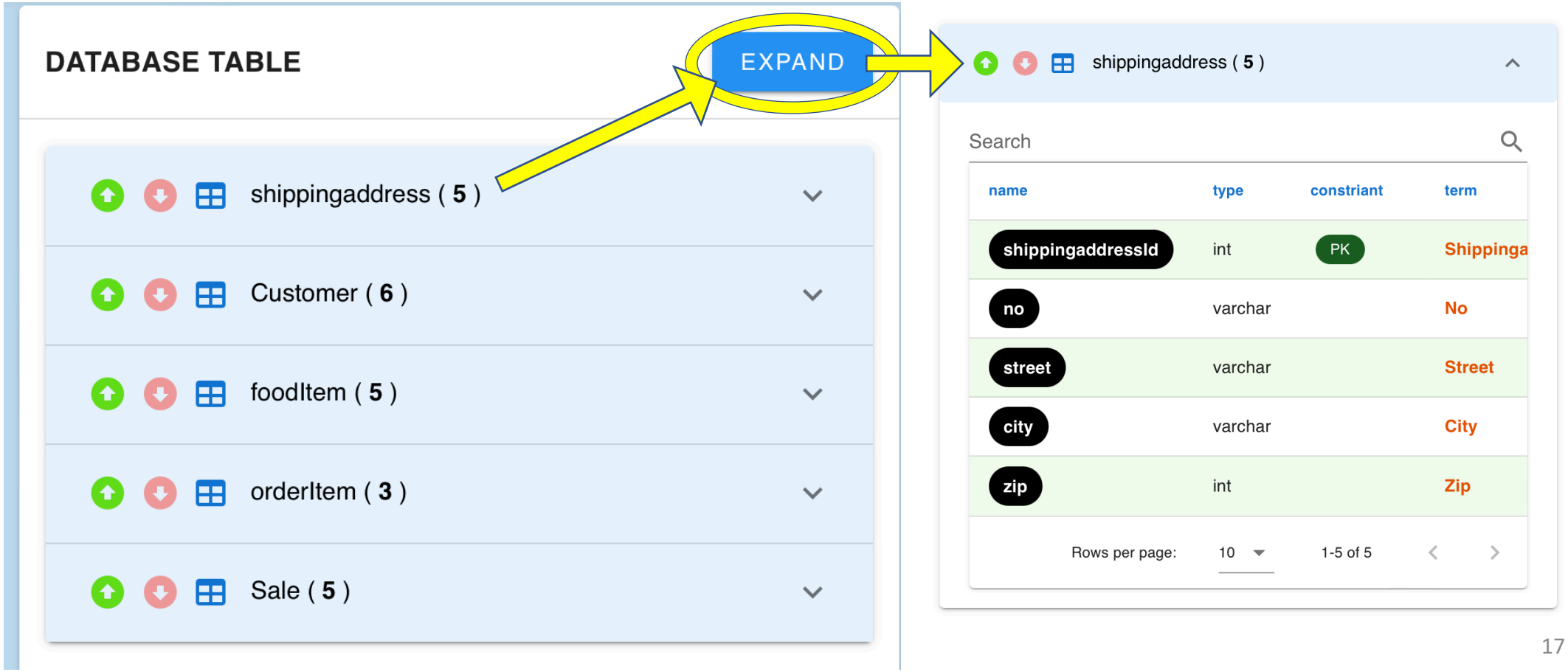
SQL

```sql
drop database IF EXISTS dm1;
CREATE DATABASE dm1 /*!40100 DEFAULT CHARACTER SET utf8mb4 */;
USE dm1;
create table `Shippingaddress` (
 shippingaddressId int(11) not null auto_increment,
 no longtext  default null,
 street longtext  default null,
 city longtext  default null,
 zip int(11) not null,
 primary key (shippingaddressId)
) engine=InnoDB default charset=utf8mb4;


create table `Customer` (
 CustomerId int(11) not null auto_increment,
 firstname longtext  default null,
 lastname longtext  default null,
 age int(11) not null,
 email longtext  default null,
 shippingaddressId int(11) not null,
 primary key (CustomerId),
 KEY IX_Customer_shippingaddressId (shippingaddressId),
```

```
mysql> use dm1;
Database changed
mysql> show tables;
+----------------+
| Tables_in_dm1  |
+----------------+
| Customer       |
| Sale           |
| Shippingaddress |
+----------------+
3 rows in set (0.00 sec)

mysql>
```

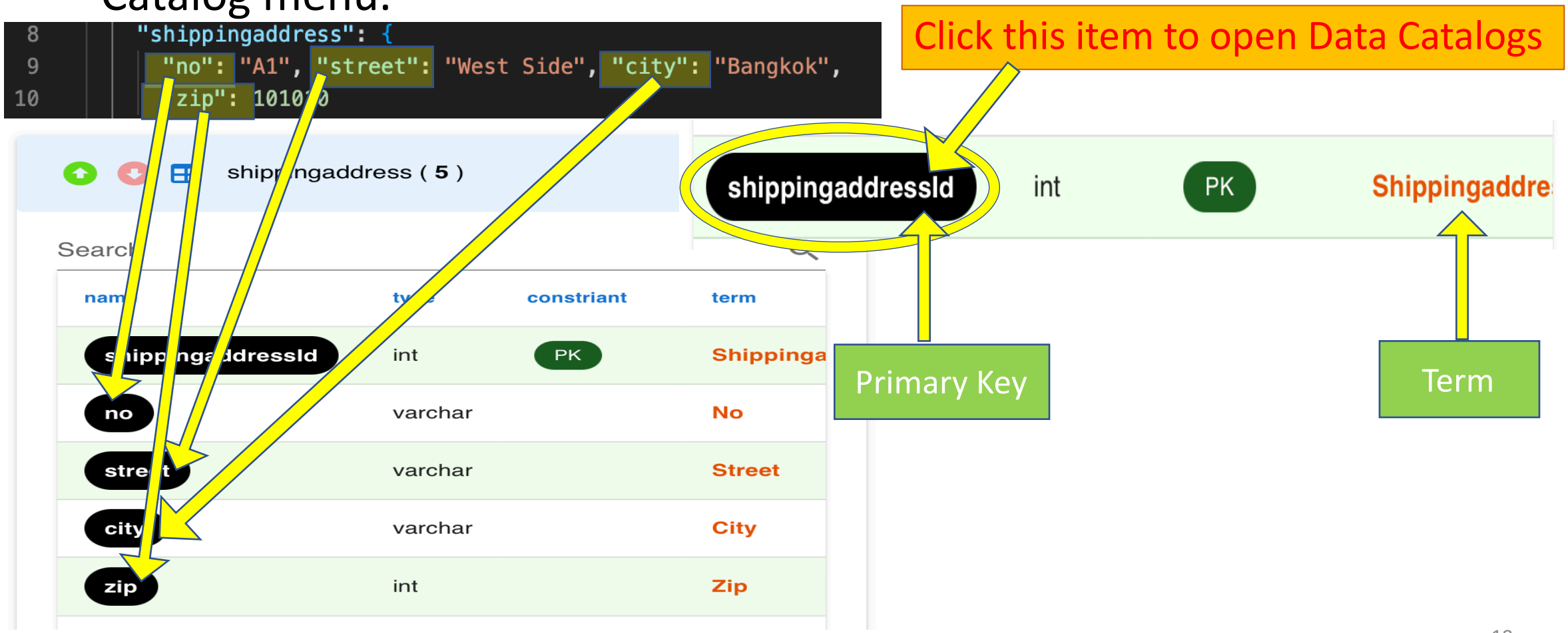
Data Modeling Tool (Lab for beginners)

- Four, let's try to click Expand button. You will see each table displays data dictionary for each field. (e.g., shippingaddress)



Data Modeling Tool (Lab for beginners)

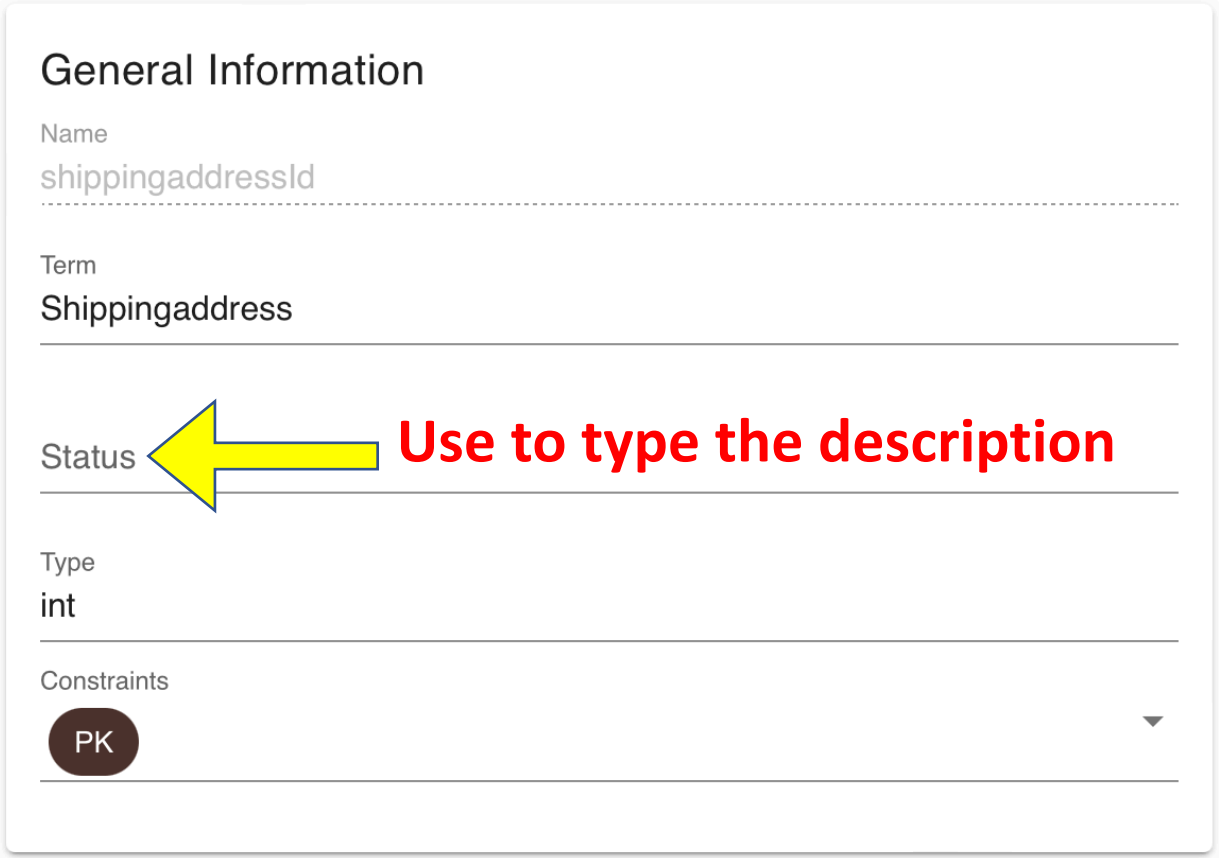
- Five, try to click name item to open Data Catalogs which displays in Catalog menu.



Data Modeling Tool (Lab for beginners)

- Display Data Catalogs for **“shippingaddressId”.**

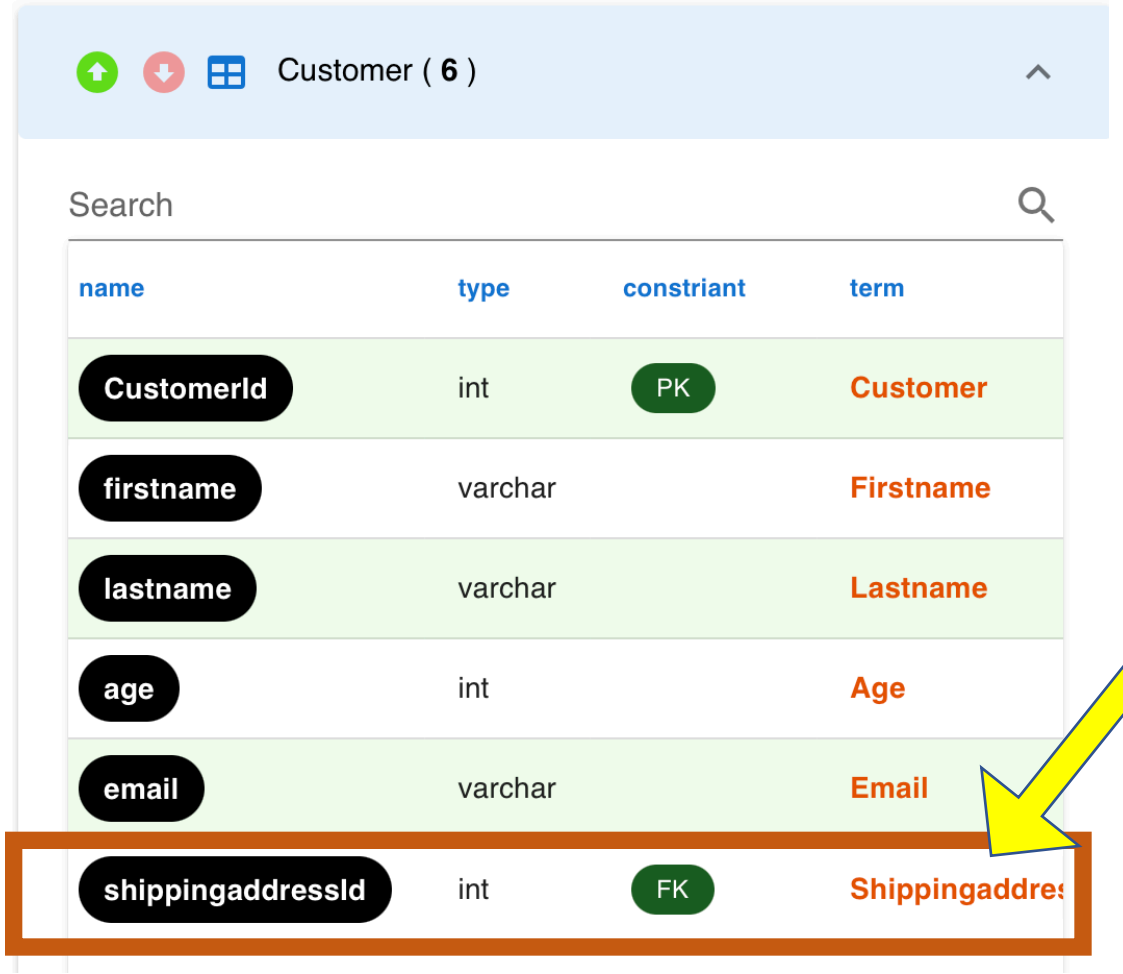
Data Element Catalogs**> shippingaddress**



Try to click other name items in each table to open Data Catalogs and see the different result …

Data Modeling Tool (Lab for beginners)

- Six, let's see inside the Customer table (one-to-one relationship).

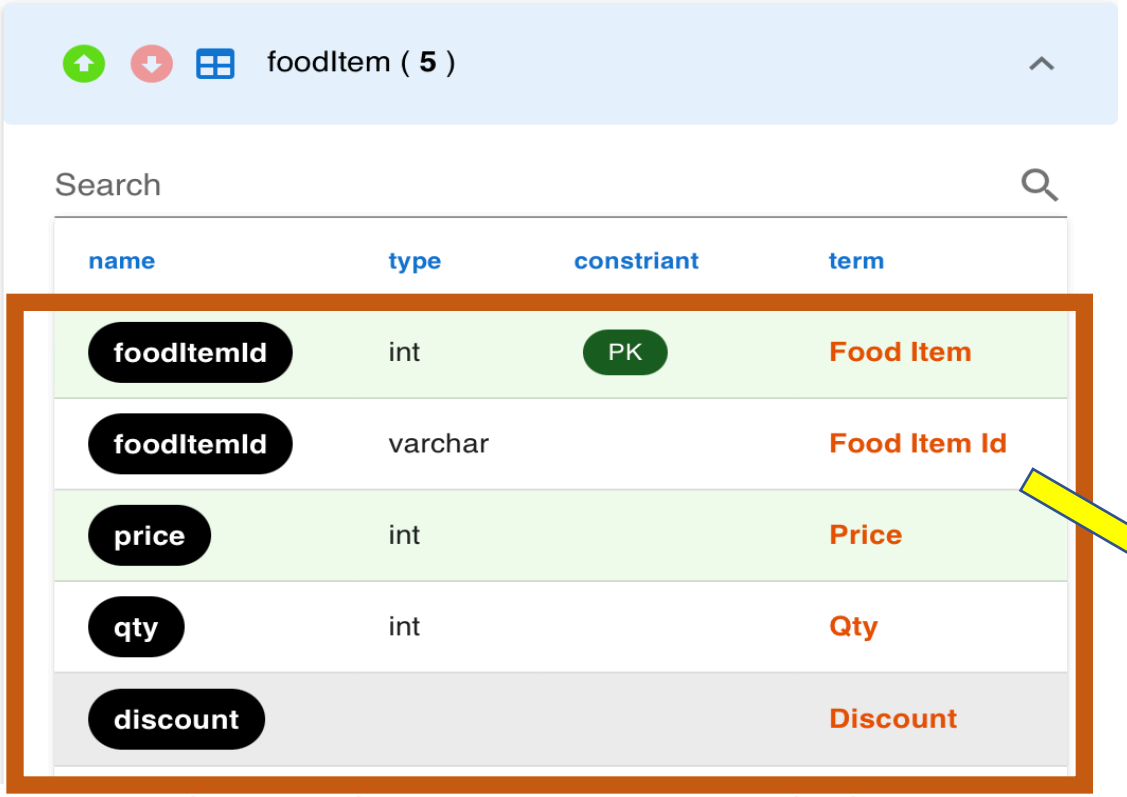



```
2   "orderId": "a101",
3   "Customer": {
4     "firstname": "John",
5     "lastname": "Doe",
6     "age": 20,
7     "email": "john@gmail.com",
8     "shippingaddress": {
9       "no": "A1", "street": "West Side", "city": "Bangkok",
10      "zip": 101010
11    }
12  },
```

Why does "shippingaddressId" appear as a foreign key inside Customer table?

Data Modeling Tool (Lab for beginners)

- Seven, let's see inside the foodItem table (subset of orderItems).

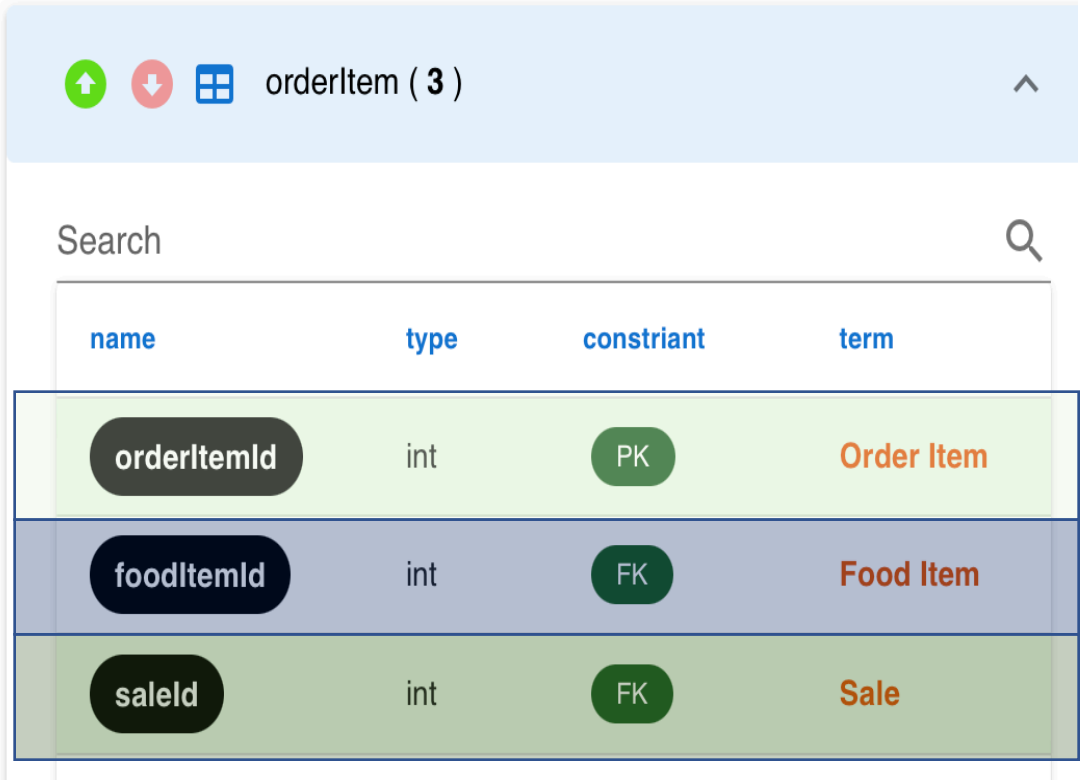


See the colors inside each layer which corresponds to JSON

```
{
  "orderId": "a101",
  "Customer": {
    "firstname": "John",
    "lastname": "Doe",
    "age": 20,
    "email": "john@gmail.com",
    "shippingaddress": {
      "no": "A1", "street": "West Side", "city": "Bangkok",
      "zip": 101010
    }
  },
  "orderItems": [
    {
      "foodItem": {
      "foodItemId": "f2",
      "price": 20, "qty": 2,
      "discount": 0
      }
    }
  ],
  "availableTime": "only noon time"
}
```

Data Modeling Tool (Lab for beginners)

- Eight, let's see inside the orderItems table (many-to-many relationship).

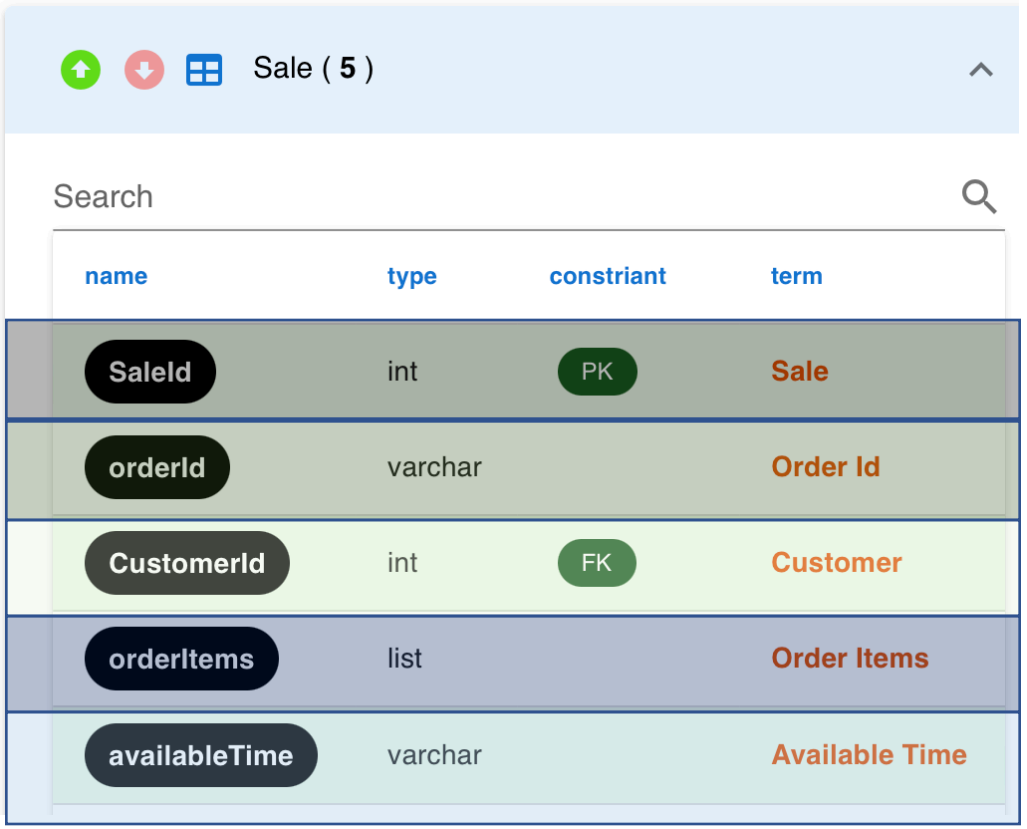


See the colors inside each layer which corresponds to JSON

```
{
  "orderId": "a101",
  "Customer": {
    "firstname": "John",
    "lastname": "Doe",
    "age": 20,
    "email": "john@gmail.com",
    "shippingaddress": {
      "no": "A1", "street": "West Side", "city": "Bangkok",
      "zip": 101010
    }
  },
  "orderItems": [
    {
      "foodItem": {
      "foodItemId": "f2",
      "price": 20, "qty": 2,
      "discount": 0
      }
    }
  ],
  "availableTime": "only noon time"
}
```

Data Modeling Tool (Lab for beginners)

- Nine, let's see inside the Sale table (one-to-many relationship).

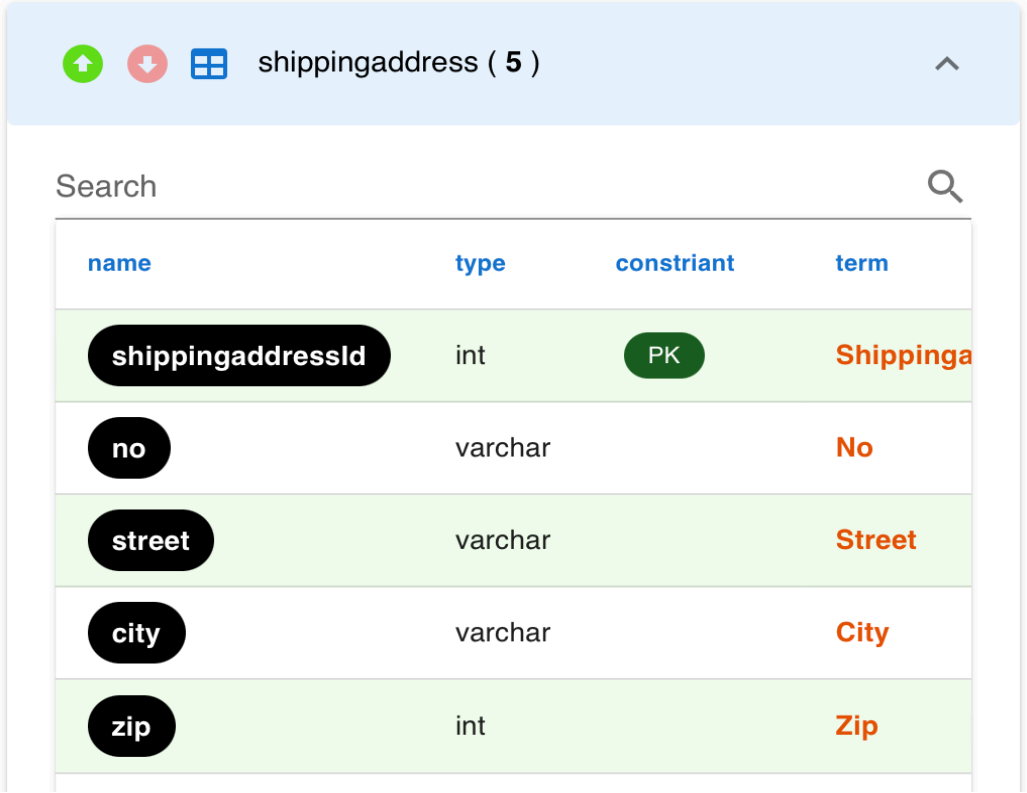


See the colors inside each layer which corresponds to JSON

```
{
  "orderId": "a101",
  "Customer": {
    "firstname": "John",
    "lastname": "Doe",
    "age": 20,
    "email": "john@gmail.com",
    "shippingaddress": {
      "no": "A1", "street": "West Side", "city": "Bangkok",
      "zip": 101010
    }
  },
  "orderItems": [
    {
      "foodItem": {
      "foodItemId": "f2",
      "price": 20, "qty": 2,
      "discount": 0
      }
    }
  ],
  "availableTime": "only noon time"
}
```

Data Modeling Tool (Lab for beginners)

- Moreover, you can see the SQL view with an empty record.



SQL

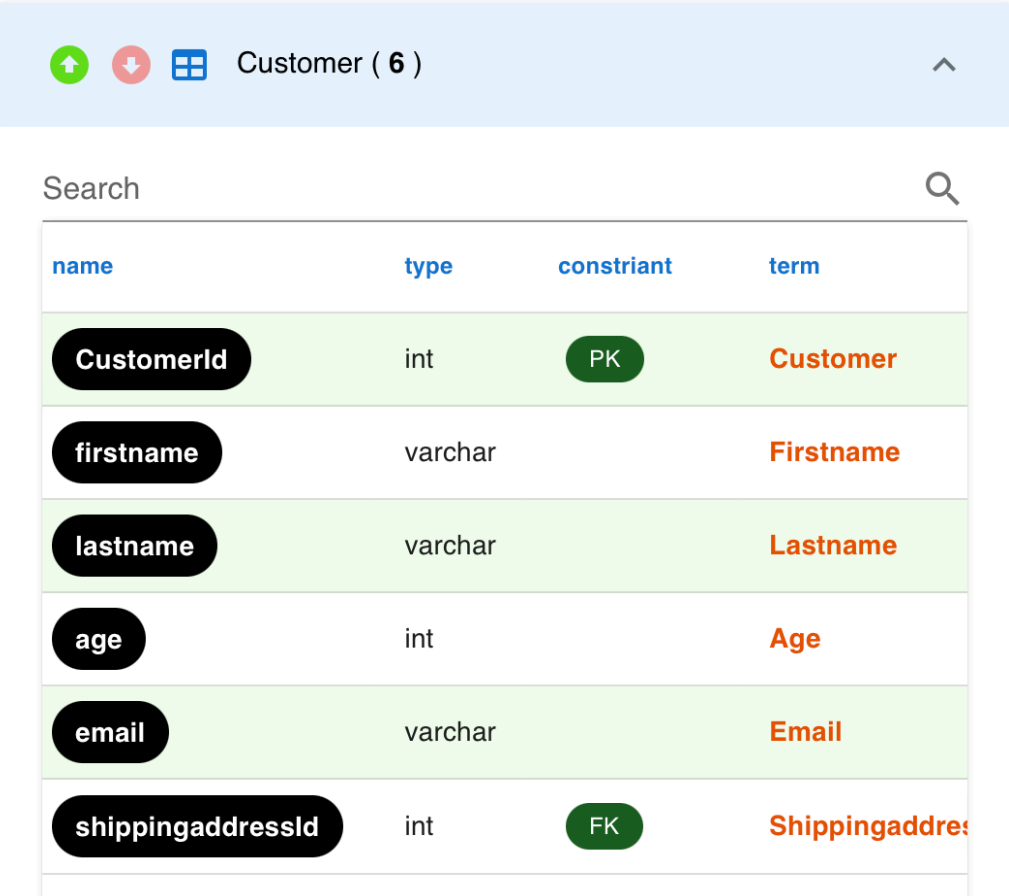
```
create table `Shippingaddress` (
 shippingaddressId int(11) not null auto_increment,
 no longtext  default null,
 street longtext  default null,
 city longtext  default null,
 zip int(11) not null,
 primary key (shippingaddressId)
) engine=InnoDB default charset=utf8mb4;
```

```
mysql> describe shippingaddress;
+-------------------+----------+------+-----+---------+----------------+
| Field             | Type     | Null | Key | Default | Extra          |
+-------------------+----------+------+-----+---------+----------------+
| shippingaddressId | int      | NO   | PRI | NULL    | auto_increment |
| no                | longtext | YES  |     | NULL    |                |
| street            | longtext | YES  |     | NULL    |                |
| city              | longtext | YES  |     | NULL    |                |
| zip               | int      | NO   |     | NULL    |                |
+-------------------+----------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql>
```

How is JSON different from SQL?

Data Modeling Tool (Lab for beginners)

- Moreover, you can see the SQL view with an empty record.



How is JSON different from SQL?

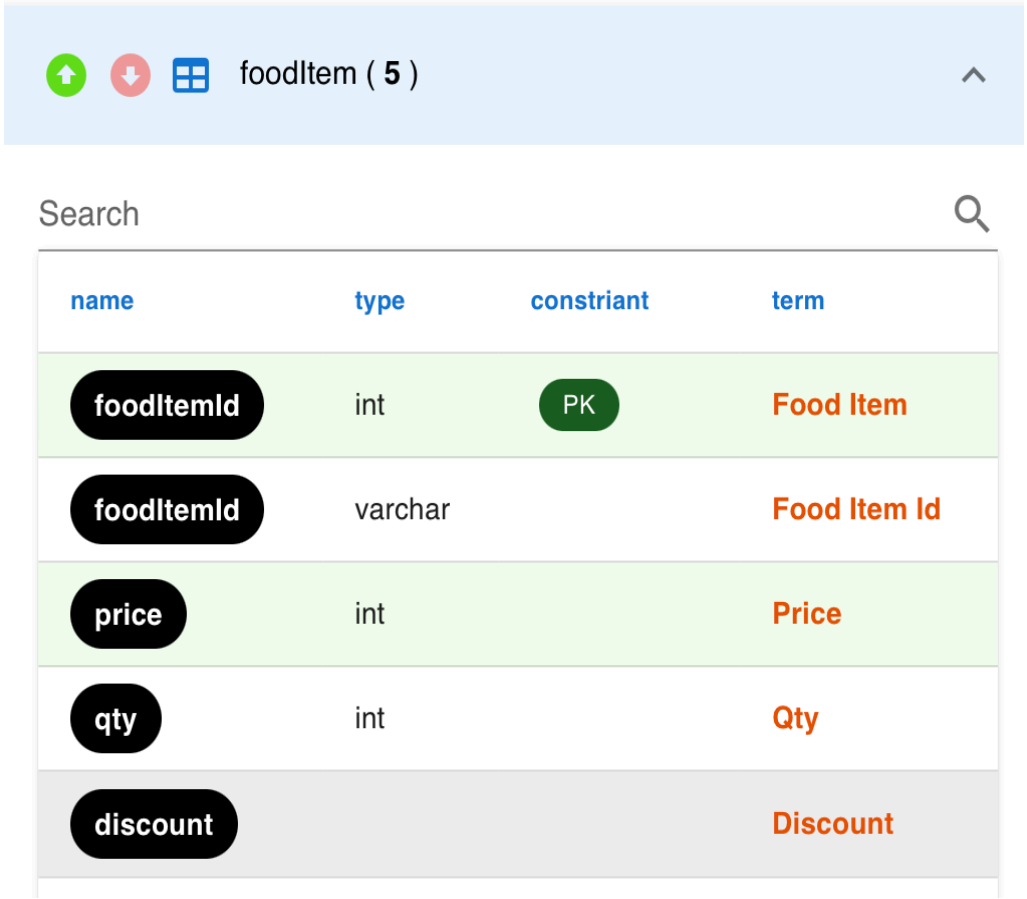
SQL

```
12  create table `Customer` (
13   CustomerId int(11) not null auto_increment,
14   firstname longtext  default null,
15   lastname longtext  default null,
16   age int(11) not null,
17   email longtext  default null,
18   shippingaddressId int(11) not null,
19   primary key (CustomerId),
20   KEY IX_Customer_shippingaddressId (shippingaddressId),
21   CONSTRAINT FK_Customer_Shippingaddress_shippingaddressId FOREIGN KEY
     (shippingaddressId) REFERENCES `Shippingaddress` (shippingaddressId) ON DELETE CASCADE
22  ) engine=InnoDB default charset=utf8mb4;
```

```
mysql> describe Customer;
+-------------------+----------+------+-----+---------+----------------+
| Field             | Type     | Null | Key | Default | Extra          |
+-------------------+----------+------+-----+---------+----------------+
| CustomerId        | int      | NO   | PRI | NULL    | auto_increment |
| firstname         | longtext | YES  |     | NULL    |                |
| lastname          | longtext | YES  |     | NULL    |                |
| age               | int      | NO   |     | NULL    |                |
| email             | longtext | YES  |     | NULL    |                |
| shippingaddressId | int      | NO   | MUL | NULL    |                |
+-------------------+----------+------+-----+---------+----------------+
6 rows in set (0.01 sec)

mysql>
```

Data Modeling Tool (Lab for beginners)

- Moreover, you can see the SQL view with an empty record.



How is JSON different from SQL?

SQL

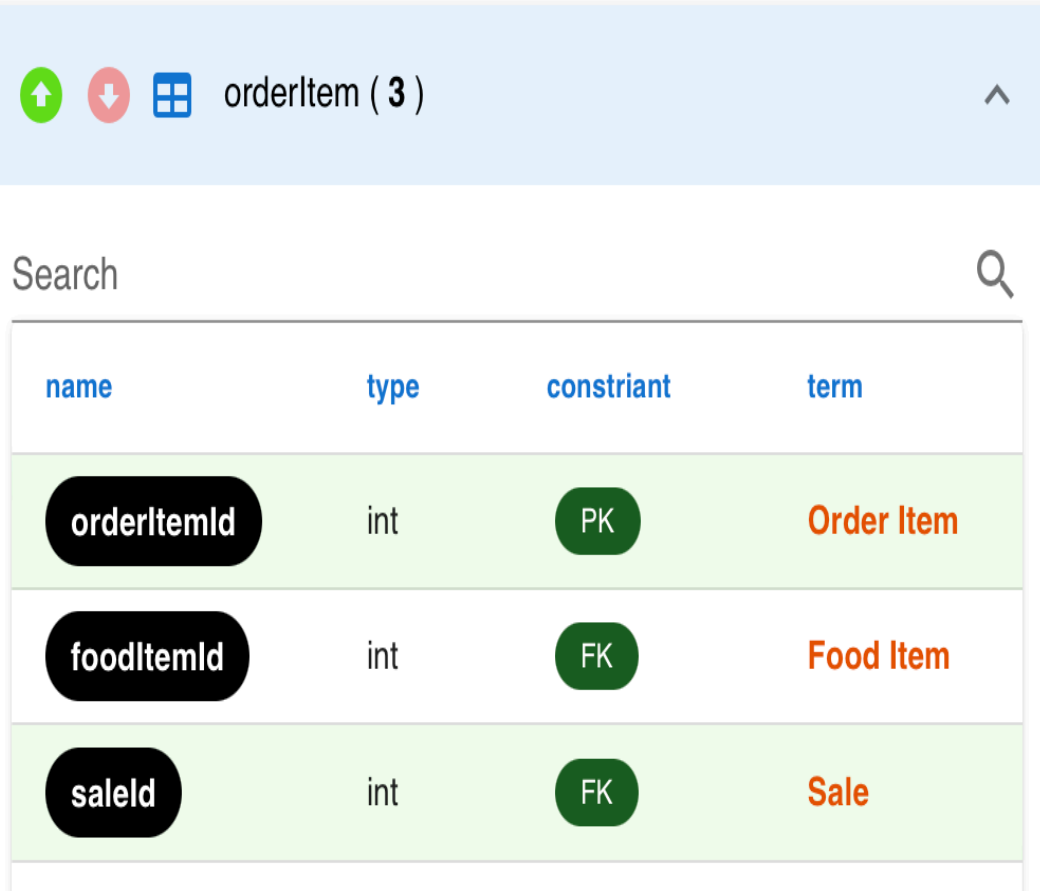
```
25  create table `FoodItem` (
26   foodItemId int(11) not null auto_increment,
27   foodItemId longtext  default null,
28   price int(11) not null,
29   qty int(11) not null,
30   primary key (foodItemId)
31  ) engine=InnoDB default charset=utf8mb4;
```

```
ERROR 1060 (42S21): Duplicate column name 'foodItemId'
ERROR 1824 (HY000): Failed to open the referenced table 'FoodItem'
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

foodItem table cannot be include in mysql (Need fixing a bit) …

Data Modeling Tool (Lab for beginners)

- Moreover, you can see the SQL view with an empty record.



How is JSON different from SQL?

SQL

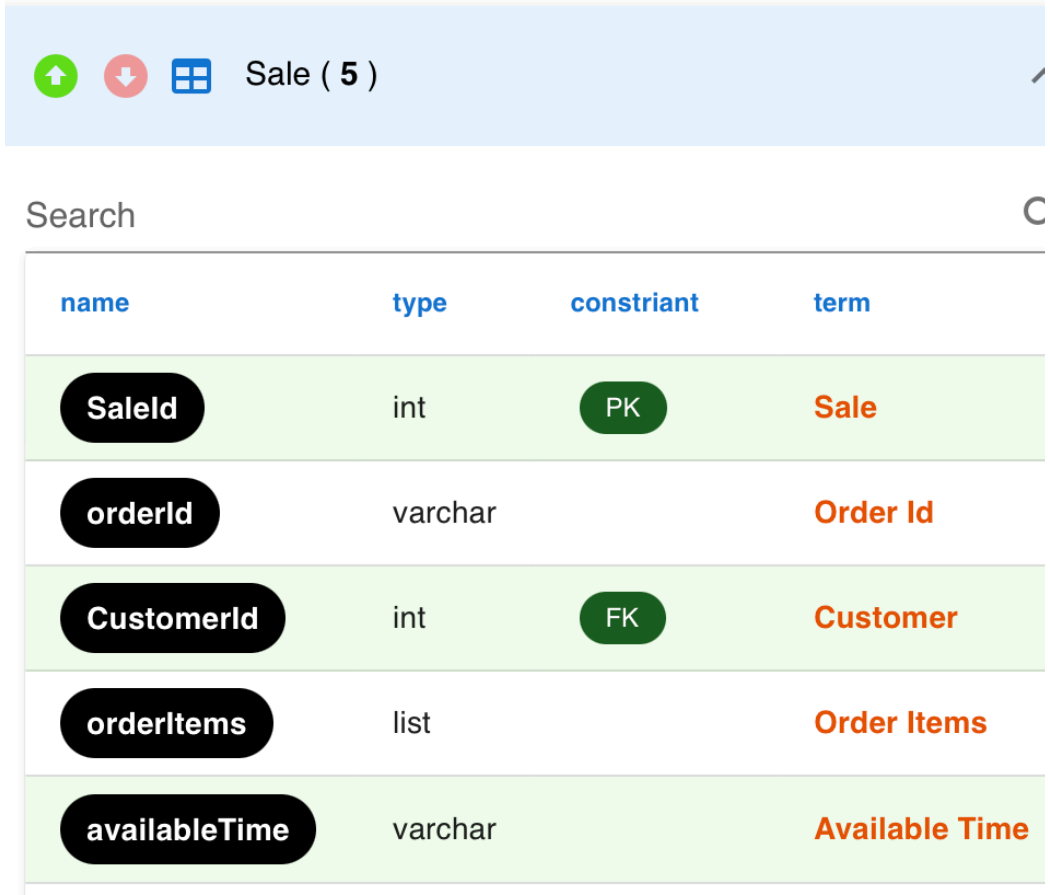
```
34  create table `OrderItem` (
35   orderItemId int(11) not null auto_increment,
36   foodItemId int(11) not null,
37   saleId int(11) not null,
38   primary key (orderItemId),
39   KEY IX_OrderItem_foodItemId (foodItemId),
40   KEY IX_OrderItem_saleId (saleId),
41   CONSTRAINT FK_OrderItem_FoodItem_foodItemId FOREIGN KEY (foodItemId) REFERENCES
     `FoodItem` (foodItemId) ON DELETE CASCADE,
42   CONSTRAINT FK_OrderItem_Sale_saleId FOREIGN KEY (saleId) REFERENCES `Sale` (saleId) ON
     DELETE CASCADE
43  ) engine=InnoDB default charset=utf8mb4;
```

```
ERROR 1060 (42S21): Duplicate column name 'foodItemId'
ERROR 1824 (HY000): Failed to open the referenced table 'FoodItem'
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

orderItem table cannot be include in mysql (Need fixing a bit) …

Data Modeling Tool (Lab for beginners)

- Moreover, you can see the SQL view with an empty record.




```
46  create table `Sale` (
47   SaleId int(11) not null auto_increment,
48   orderId longtext  default null,
49   CustomerId int(11) not null,
50   availableTime longtext  default null,
51   primary key (SaleId),
52   KEY IX_Sale_CustomerId (CustomerId),
53   CONSTRAINT FK_Sale_Customer_CustomerId FOREIGN KEY (CustomerId) REFERENCES `Customer`
     (CustomerId) ON DELETE CASCADE
54  ) engine=InnoDB default charset=utf8mb4;
```

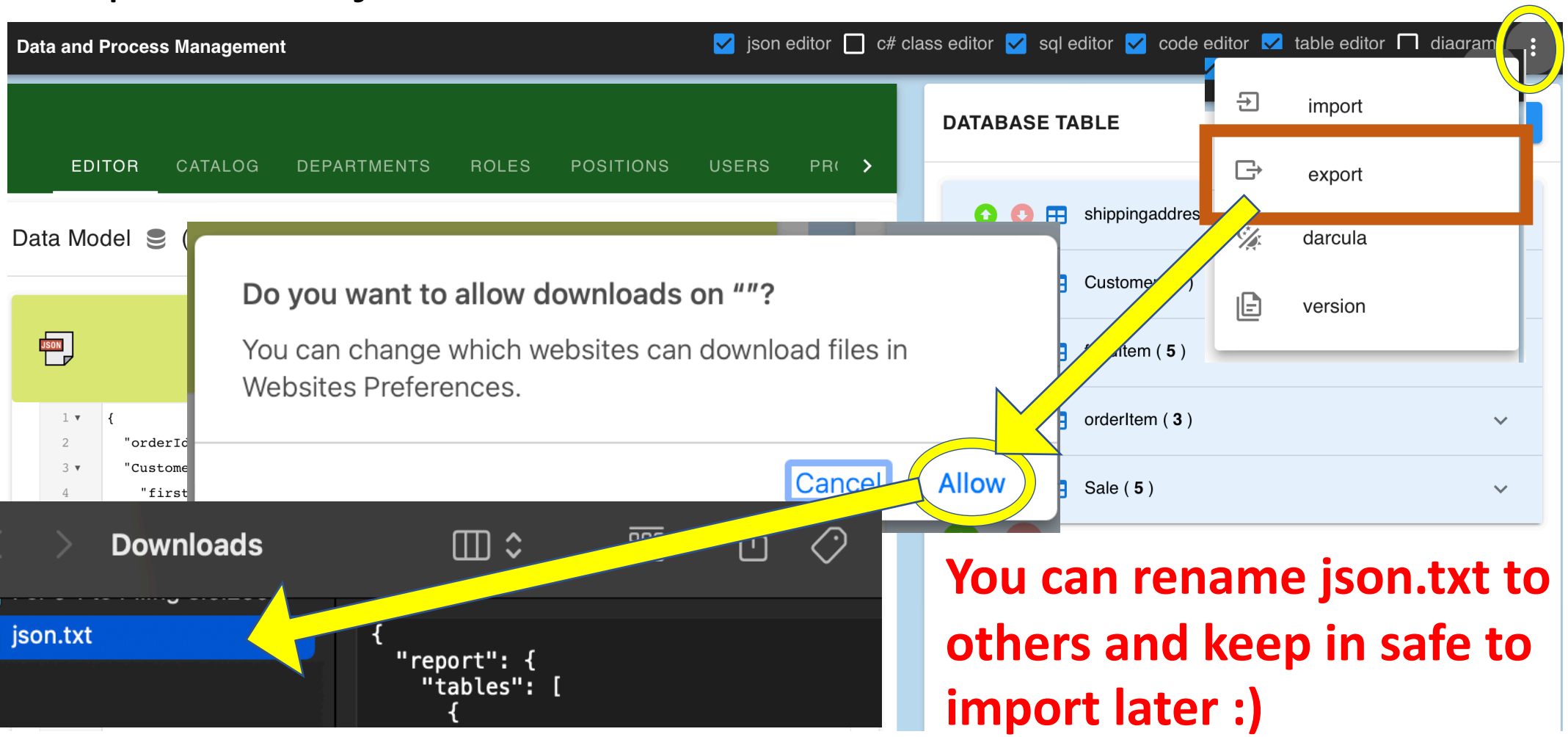
```
mysql> describe Sale;
+---------------+----------+------+-----+---------+----------------+
| Field         | Type     | Null | Key | Default | Extra          |
+---------------+----------+------+-----+---------+----------------+
| SaleId        | int      | NO   | PRI | NULL    | auto_increment |
| orderId       | longtext | YES  |     | NULL    |                |
| CustomerId    | int      | NO   | MUL | NULL    |                |
| availableTime | longtext | YES  |     | NULL    |                |
+---------------+----------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql>
```

How is JSON different from SQL?

Data Modeling Tool (Lab for beginners)

- Export file to **json.txt**

Click this circle button and choose export

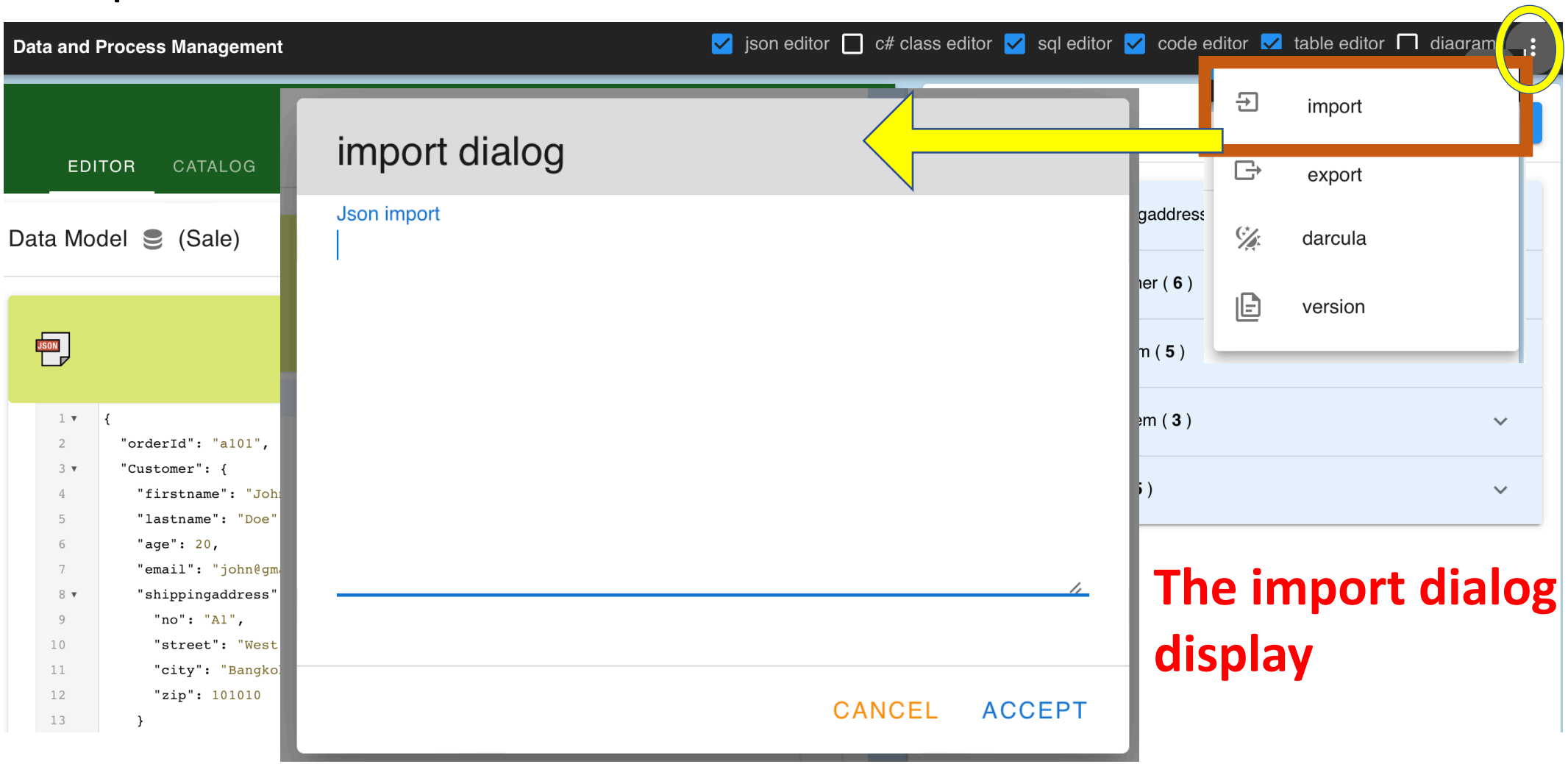


You can rename json.txt to others and keep in safe to import later :)

Data Modeling Tool (Lab for beginners)

- Import file into DBM Tool.

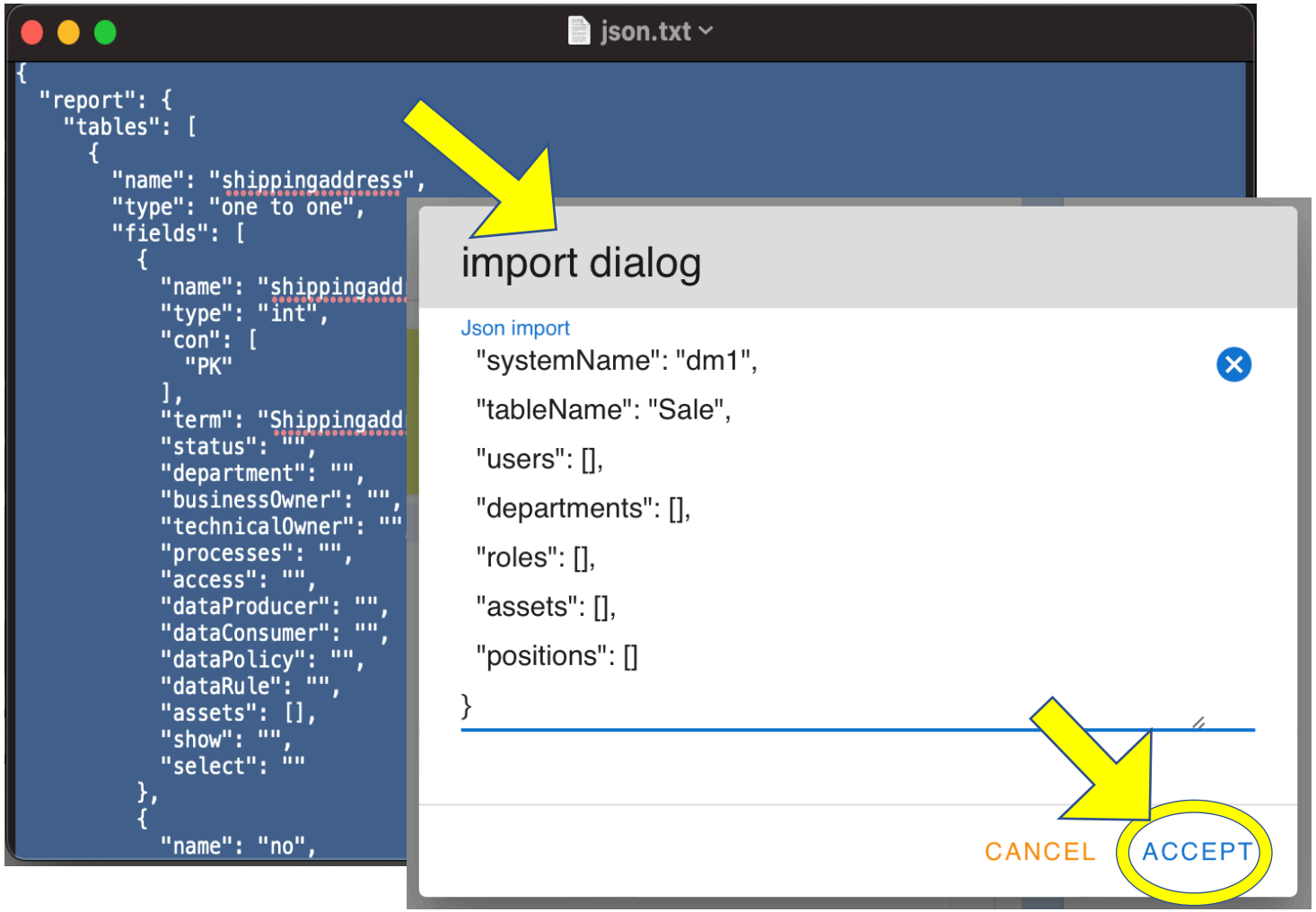
Click this circle button and choose import



The import dialog will display

Data Modeling Tool (Lab for beginners)

- Import file into DBM Tool.



Step for importing file:

1. Open **json.txt** into **Notepad** or **TextEdit** App
2. Use hotkey **“Ctrl + A”** to select all texts
3. Use hotkey **“Ctrl + C”** to copy
4. Click in the **text area** of **import dialog**
5. Use hotkey **“Ctrl + V”** to paste and click ACCEPT

Data Modeling Tool (Lab for beginners)

- Let's try this (In class).
  - Interpret this business communication phrase into JSON. Then use DBM Tool to convert and see the different result:

"Ms. Evangeline McDowell would like to rent a Honda Civic, plate 2AB1820, year 2016 with dark silver color for 2,000 baht a day to drive in Phuket for 3 days. (Hint: rentId is "R1301")

She is 24 years old and lives in 1135 Ladphrao 81 Wangthonglang Wangthonglang Bangkok 10310, and phone number is 0954220896"