RELATIONAL DATA MODEL

PREPARED FOR

Project Phase - III

DATA AND APPLICATIONS

PREPARED BY

DAKSH RAWAT - 2018101087

VASU SINGHAL - 2018101074

PRIYANSHU MADAAN - 2018101097

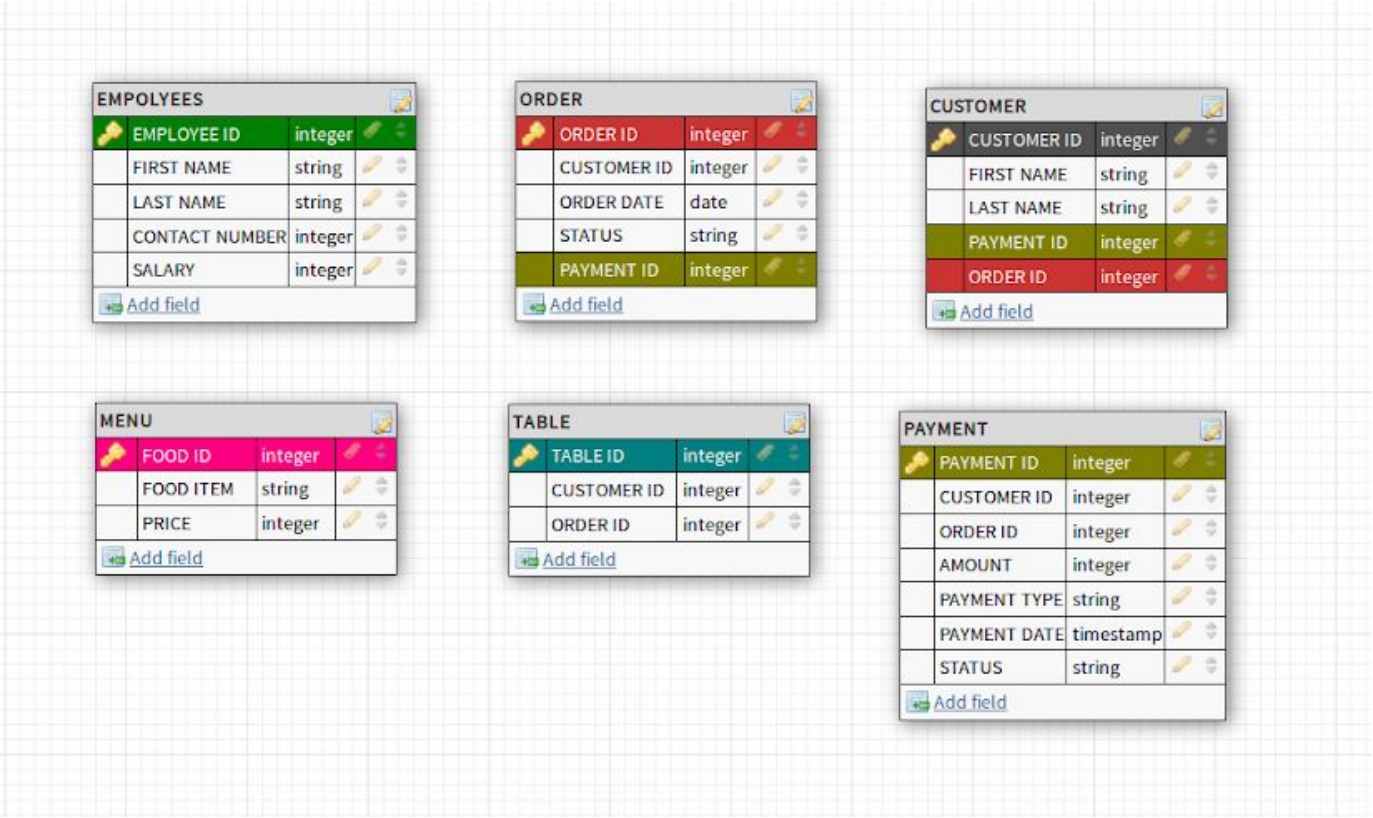
MODIFICATIONS IN THE ER MODEL

- No need for the ‘Menu Id’ attribute for the relation ‘Menu” since there is only one Menu in the restaurant.

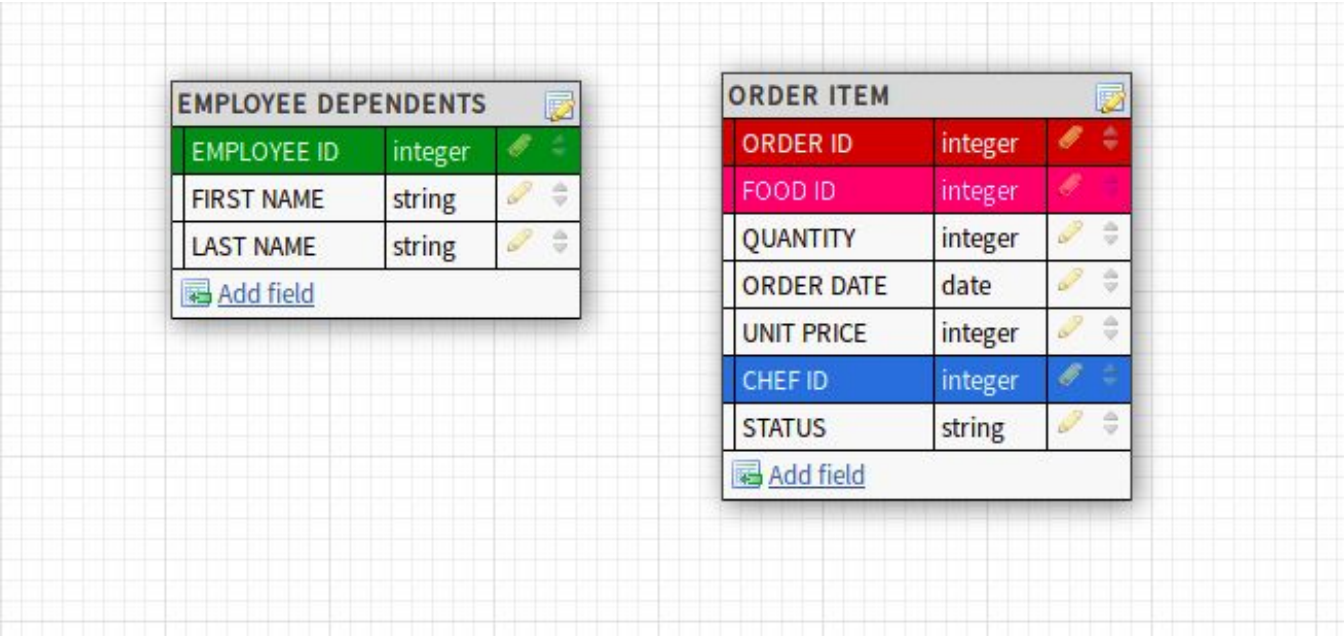
- The ER model we have received assumes food id, food item attributes of menu table to be multi-valued, however, they are not in the following relational data model.
- Removed the ‘Quantity’ attribute of ‘Order’ table.
- Changed the name of the attribute ‘food id’ in order table to ‘food ids’ since it’s a multi-valued attribute and the name was a little confusing in that regard.

ER MODEL TO RELATIONAL MODEL

- STEP 1: Mapping of Regular Entity Types



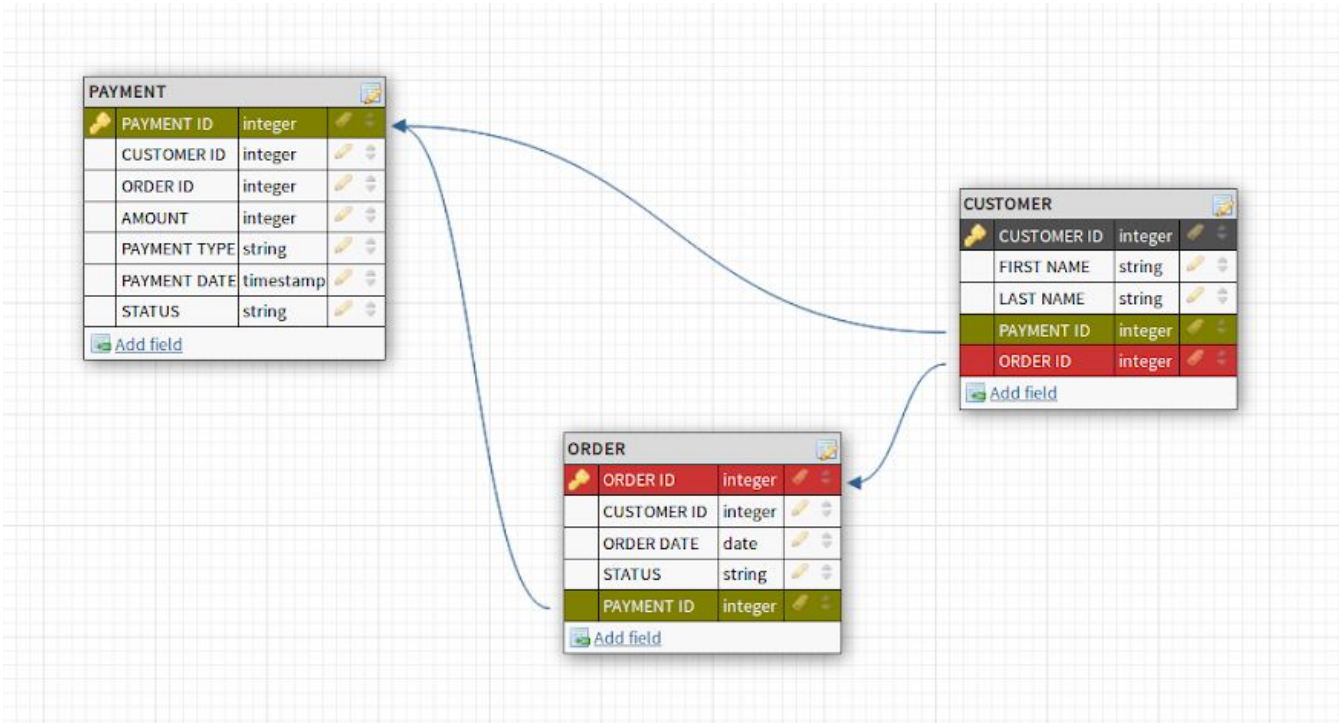
- STEP 2: Mapping of Weak Entity Types



- STEP 3: Mapping of Binary 1:1 Relationship Types
  - We have used FOREIGN KEY APPROACH here.
  - The Relationships are:
    i. Customer **HasPaid** for an order.
      - Included the payment id as a foreign key in customer table.

    ii. Customer **HasOrdered** an order.
      - Included the order id as a foreign key in the customer table.

    iii. **PaymentFor** : exists between an order and a payment
      - Included the payment id as a foreign key in the order table.

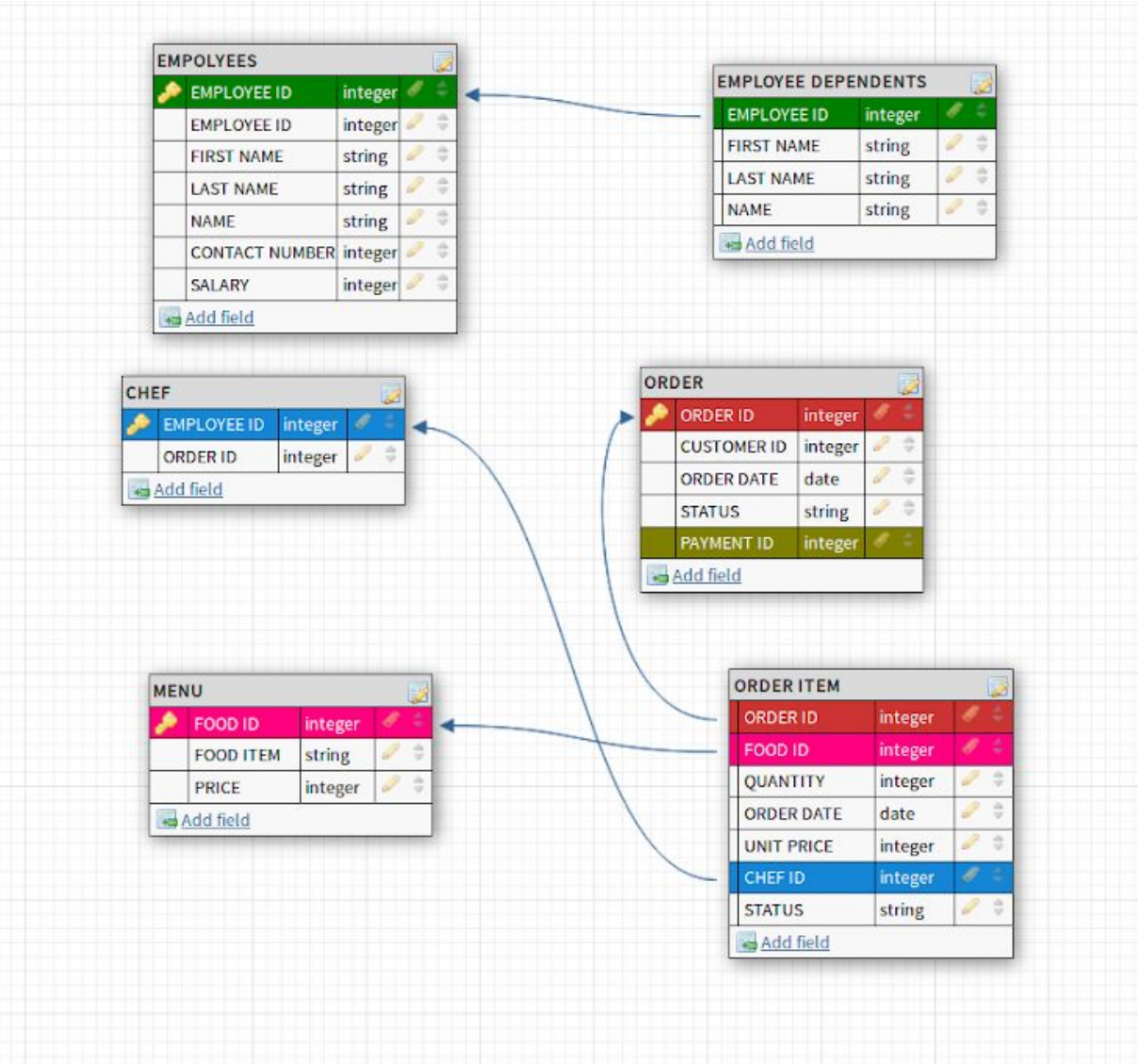


- STEP 4: Mapping of Binary 1:N Relationship Types
    - We have used FOREIGN KEY APPROACH here.
    - The Relationships are:
        i. Chef **IsPreparing** an order item.
            - Included ‘chef id’ as a foreign key in order item table.

        ii. **ItemInOrder**: an identifying relationship that exists between order item and an order.
            - Included ‘order id’ as a foreign key in order item table.

        iii. **ItemInMenu**: exists between on order item and the corresponding food item in the menu.
            - Included ‘food id’ as a foreign key in the order item table.

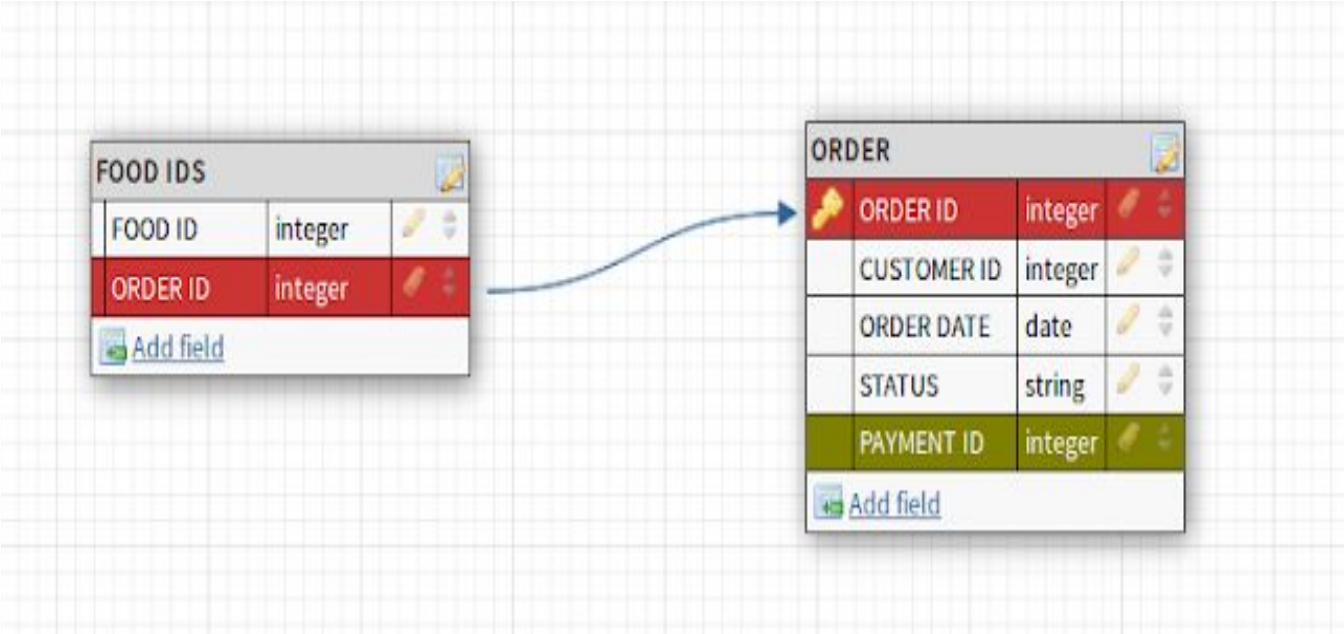
        iv. **IsDependentOf**: an identifying relationship that exists between an employee and his/her dependents.
            - Included ‘employee id’ as a foreign key in the employee dependent table.



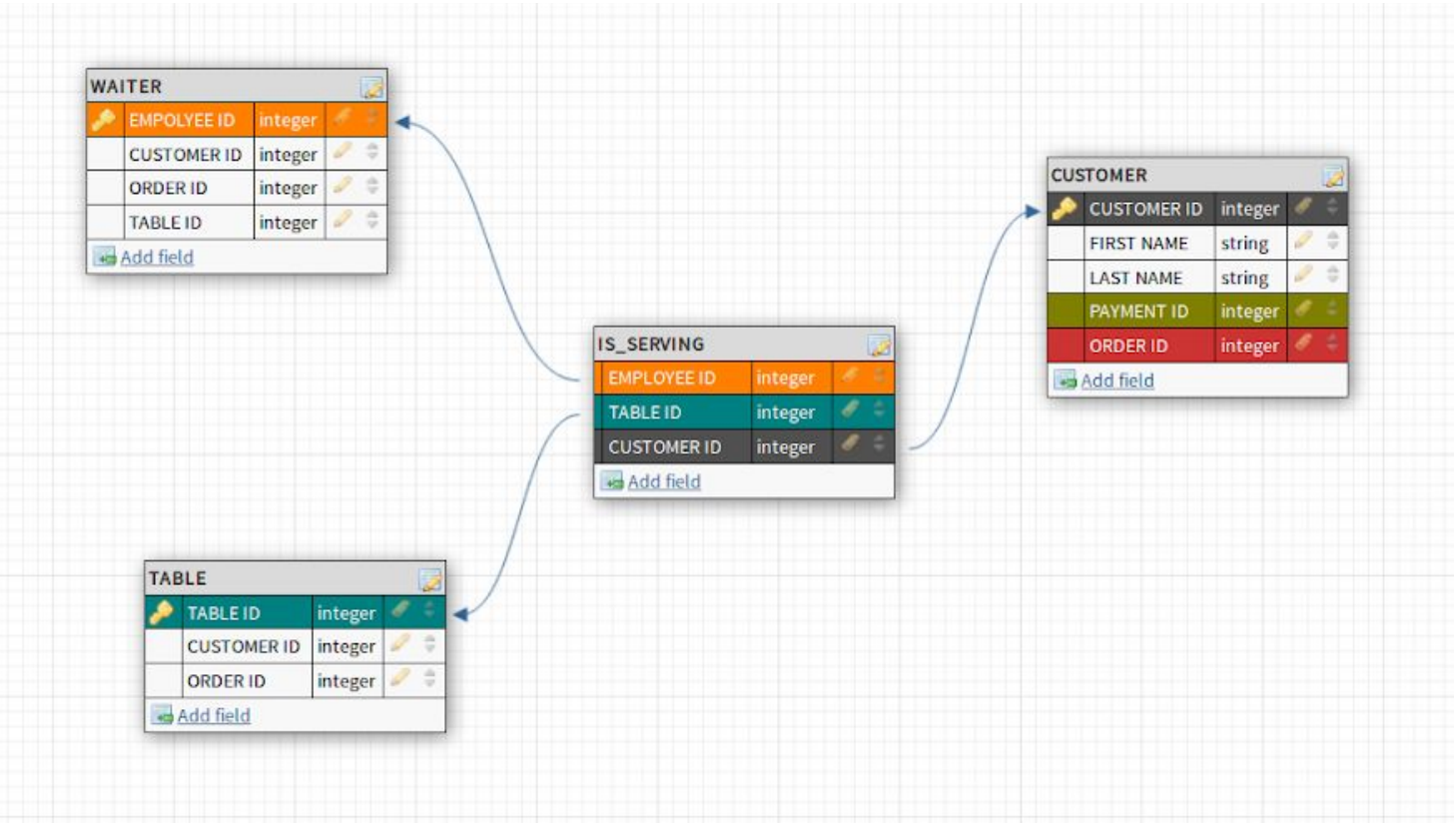
* CHEF is a subclass of EMPLOYEE. Refer step 8.

- STEP 5: Mapping of Binary M:N Relationship Types
    - No such relationships exist.

- STEP 6: Mapping of Multi-valued Attributes
    - FOOD IDS: Created a new relation ‘FOOD IDS’ and included the primary key of relation ‘ORDER’ as foreign key in it.

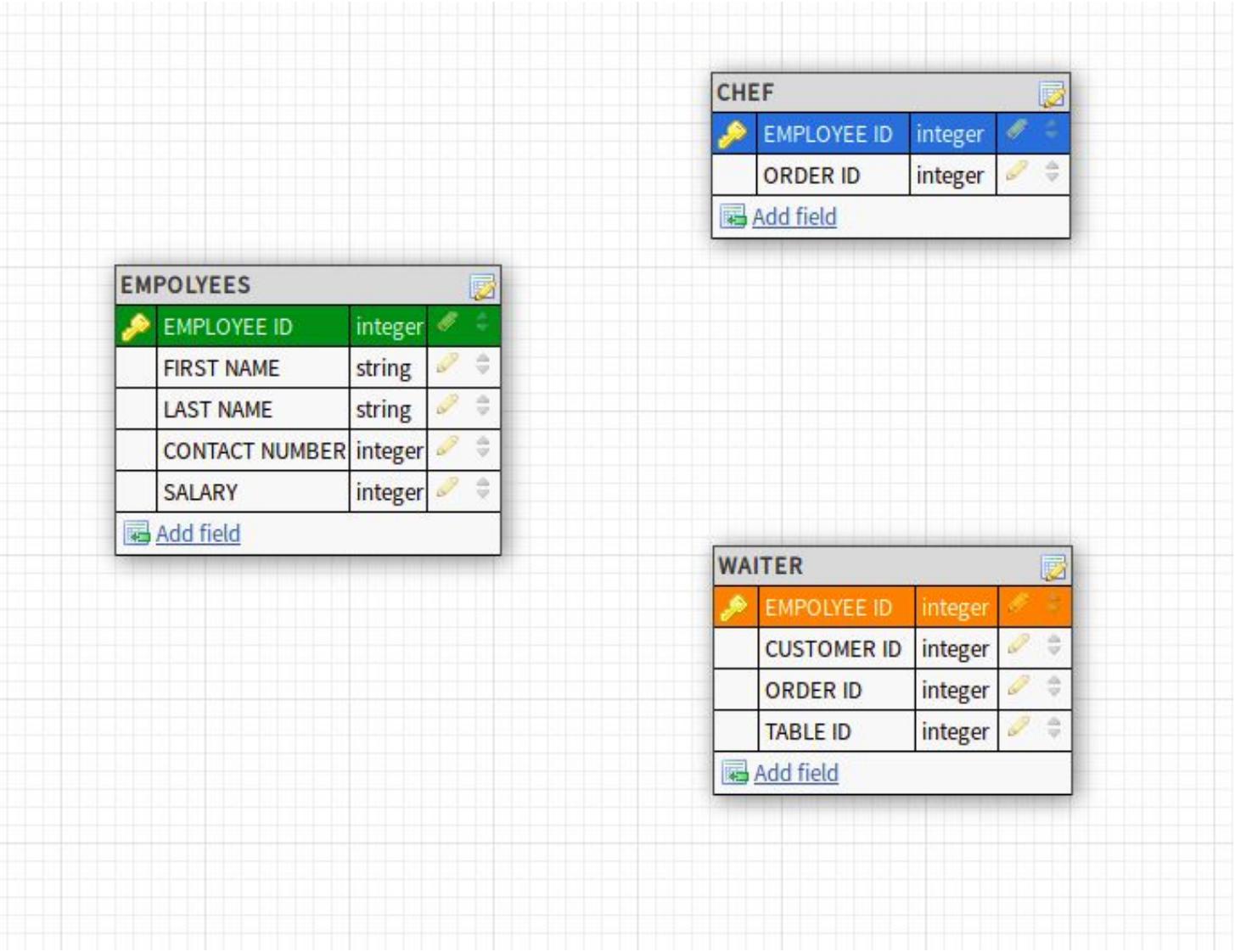


- STEP 7: Mapping of N-ary Relationship Types
    - **IsServing**: Exists between waiter & customer and waiter & table (TERNARY).



* WAITER is a subclass of EMPLOYEE. Refer step 8.

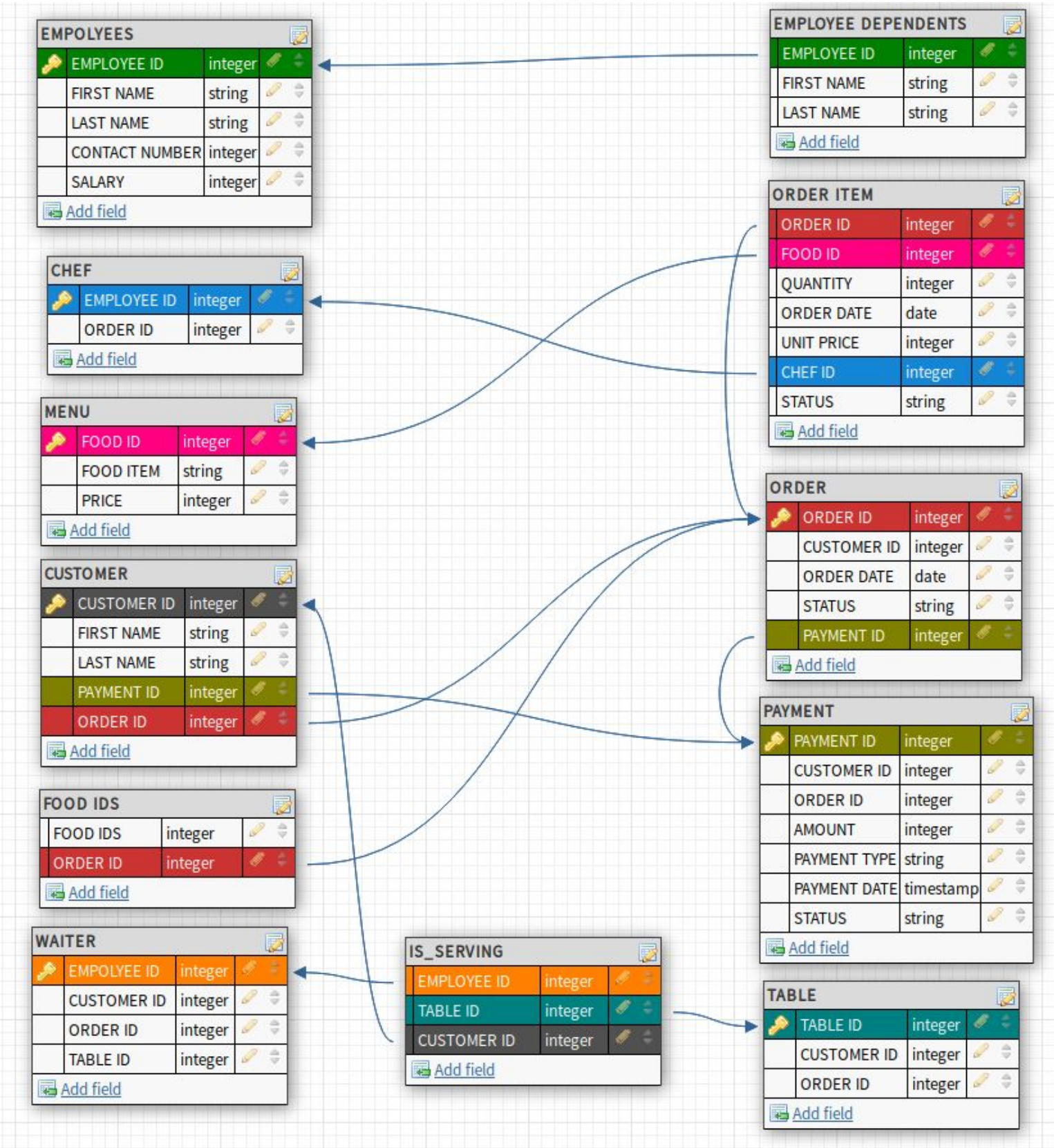
- STEP 8: Mapping Specialization or Generalization
    - There are two Specialization in our model, Chef and Waiter. Both belong to superclass Employee. Therefore a new relation for subclasses Chef and Waiter is made which includes the primary key of its superclass Employee.



- STEP 9: Mapping of Union Types (Categories)

  No Union Types Exist in our Schema

THE RELATIONAL MODEL OBTAINED FROM ER


EMPOLYEES
EMPLOYEE ID integer
FIRST NAME string
LAST NAME string
CONTACT NUMBER integer
SALARY integer
Add field
EMPLOYEE DEPENDENTS
EMPLOYEE ID integer
FIRST NAME string
LAST NAME string
Add field
CHEF
EMPLOYEE ID integer
ORDER ID integer
Add field
ORDER ITEM
ORDER ID integer
FOOD ID integer
QUANTITY integer
ORDER DATE date
UNIT PRICE integer
CHEF ID integer
STATUS string
Add field
MENU
FOOD ID integer
FOOD ITEM string
PRICE integer
Add field
ORDER
ORDER ID integer
CUSTOMER ID integer
ORDER DATE date
STATUS string
PAYMENT ID integer
Add field
CUSTOMER
CUSTOMER ID integer
FIRST NAME string
LAST NAME string
PAYMENT ID integer
ORDER ID integer
Add field
PAYMENT
PAYMENT ID integer
CUSTOMER ID integer
ORDER ID integer
AMOUNT integer
PAYMENT TYPE string
PAYMENT DATE timestamp
STATUS string
Add field
FOOD IDS
FOOD IDS integer
ORDER ID integer
Add field
WAITER
EMPOLYEE ID integer
CUSTOMER ID integer
ORDER ID integer
TABLE ID integer
Add field
IS_SERVING
EMPLOYEE ID integer
TABLE ID integer
CUSTOMER ID integer
Add field
TABLE
TABLE ID integer
CUSTOMER ID integer
ORDER ID integer
Add field

Conversion of Relational Model to 1NF

The relational model is already in 1NF as it does not contain any multivalued attributes.

Conversion of 1NF Relational Model to 2NF

The relational model is already in 2NF (refer to step 6 in converting ER Model to relational model).

Conversion of 2NF Relational Model to 3NF

The relational model is already in 3NF.

EXAMPLE TABLES

- **Employees**

Employee Id	First Name	Last Name	Contact Number	Salary
1	Bob	Geller	9999999999	1324 $
2	Frank	Tribbiani	8888888888	1344 $
3	Matt	Blanc	7777777777	1222 $

- **Chef**

Employee Id	Order Id
1	43
3	43

- **Menu**

Food ID	Food Item	Price
1	Burger	20 $
2	Pizza	25 $

- **Customer**

Customer ID	First Name	Last Name	Payment Id	Order Id
1	Josh	Radner	3456	43

- **Order**

Order Id	Customer Id	Order Date	Status	Payment Id
43	1	23-12-2019	Pending	3456

- **Food Ids**

Food id	Order Id
1	43
2	43

- **Order Item**

Order Id	Food Id	Quantity	Order Date	Unit Price	Chef ID	Status
43	1	3	23-12-2019	20 $	1	Pending
43	2	1	23-12-2019	25 $	3	Pending

- **Table**

Table ID	Customer ID	Order ID
1	1	43
2	-	-

- **Waiter**

Employee ID	Customer ID	Order ID	Table ID
2	1	43	1

- **Employees Dependent**

Employee ID	First Name	Last Name
1	Lisa	Kudrow
3	Mathew	Perry

- **Payment**

Payment ID	Customer ID	Order ID	Amount	Payment Type	Payment Date	Status
3456	1	43	85	Cash	-	Pending

- **Is_Serving**

Employee ID	Table ID	Customer ID
2	1	1