Brunch Restaurant E-R Diagram

Team #31: Matthew Duran & Meghan Shah | Spring 2021

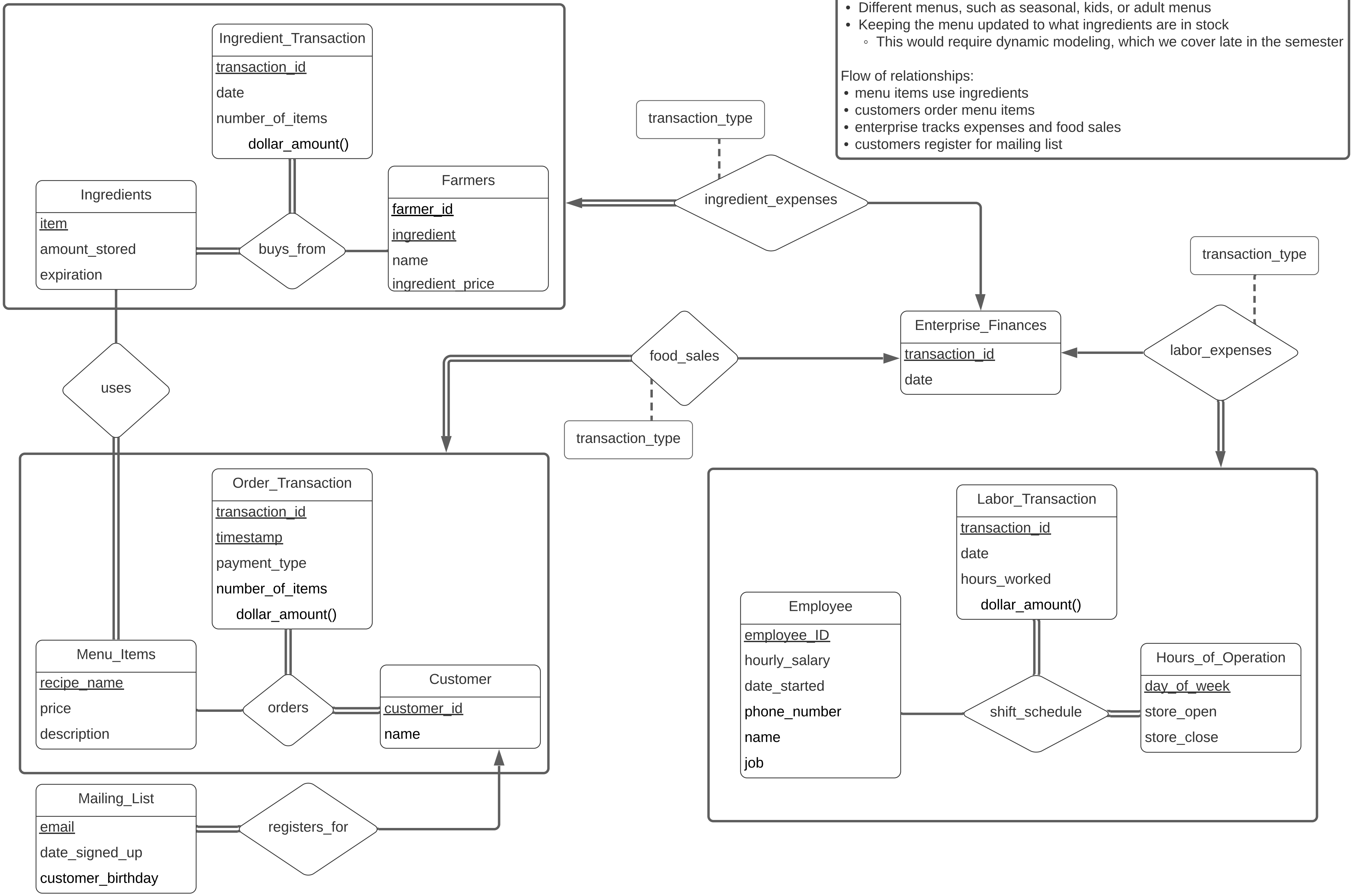
E-R Diagram Notes

Challenges:

- How to keep track of customers (assume we get a unique name for each) and full customer bills
- Tying each transaction to a unique identifier, and employee to a shift
  - this has been solved through the use of ternary relationships
- Cash versus credit transactions
- Different menus, such as seasonal, kids, or adult menus
- Keeping the menu updated to what ingredients are in stock
  - This would require dynamic modeling, which we cover late in the semester

Flow of relationships:

- menu items use ingredients
- customers order menu items
- enterprise tracks expenses and food sales
- customers register for mailing list



Brunch Restaurant Entity SQL Schema

Team #31: Matthew Duran & Meghan Shah | Spring 2021

```
Create table employee (
 employee_id varchar (10),
 hourly_salary numeric (5,2) not null,
 date_started date not null,
 phone_number varchar (13),
 name varchar (30) not null,
 job varchar (15) not null,
 Primary key (employee_id),
 Check (hourly_salary > 14) );
```

```
Create table hours_of_operation (
 day_of_week varchar (9),
 store_open time not null,
 store_close time not null,
 Primary key (day_of_week) );
```

```
Create table enterprise_finances (
 transaction_id varchar (15),
 date date,
 Primary key (transaction_id));
```

```
Create table menu_items (
 recipe_name varchar (50),
 price numeric (5,2) not null,
 description varchar (200) not null,
 Primary key (recipe_name),
 Check (price > -1) );
```

```
Create table customer (
 customer_id varchar (10),
 name varchar (40),
 Primary key (customer_id) );
```

```
Create table mailing_list (
 email varchar (25)
 date_signed_up date not null,
 customer_birthday date,
 Primary key (email) );
```

```
Create table ingredients (
 item varchar (50),
 amount_stored int not null,
 expiration_date date,
 Primary key (item) );
```

```
Create table farmers (
 farmer_id varchar (10)
 name varchar (50) not null,
 ingredient varchar (50),
 ingredient_price numeric (7,2) not null,
 Primary key (farmer_id, ingredient),
 Check (ingredient_price > -1) );
```

```
Create table ingredient_transaction (
 transaction_id varchar (15),
 date date not null,
 number_of_items int not null,
 Primary Key (transaction_id),
 Check (number_of_items > 0) );
```

```
Create table order_transaction (
 transaction_id varchar (15),
 timestamp timestamp not null,
 payment_type varchar (10) not null,
 number_of_items int not null,
 Primary Key (transaction_id,
timestamp),
 Check (number_of_items > 0) );
```

```
Create table labor_transaction (
 transaction_id varchar (15),
 date date not null,
 hours_worked numeric (5,2) not null,
 Primary Key (transaction_id),
 Check (hours_worked > -1) ) ;
```

Brunch Restaurant Relationship SQL Schema

Team #31: Matthew Duran & Meghan Shah | Spring 2021

```
Create table orders (
 customer_id varchar (10),
 transaction_id varchar (15),
 timestamp timestamp,
 recipe_name varchar (30),
 Primary key (transaction_id, timestamp, customer_id,
recipe_name),
 Foreign Key (transaction_id, timestamp) references
order_transaction,
 Foreign key (customer_id) references customer,
 Foreign key (recipe_name) references menu_items);
```

```
Create table uses (
 recipe_name varchar (50),
 item varchar (50),
 Primary key (recipe_name, item),
 Foreign Key (recipe_name) references menu_items,
 Foreign Key (item) references ingredients );
```

```
Create table registers_for (
 email varchar (25),
 customer_id varchar (10),
 Primary key (email, customer_id),
 Foreign key (email) references mailing _list,
 Foreign key (customer_id) references customer );
```

```
Create table shift_schedule (
 transaction_id varchar (15),
 day_of_week varchar (9),
 employee_id varchar (10),
 Primary Key (transaction_id, day_of_week, employee_id),
 Foreign Key (day_of_week) references hours_of_operation,
 Foreign Key (employee_id) references employee,
 Foreign Key (transaction_id) references labor_transaction );
```

```
Create table buys_from (
 item varchar (50),
 transaction_id varchar (15),
 farmer_id varchar (10),
 ingredient varchar (50),
 Primary key (item, transaction_id, farmer_id, ingredient),
 Foreign Key (farmer_id, ingredient) references farmers,
 Foreign Key (item) references ingredients,
 Foreign Key (transaction_id) references
ingredient_transaction);
```

```
Create table labor_expenses (
 transaction_type varchar (20) not null,
 transaction_id varchar (15),
 day_of_week varchar (9),
 employee_id varchar (10),
 Primary Key (transaction_id, day_of_week, employee_id),
 Foreign Key (day_of_week) references hours_of_operation,
 Foreign Key (employee_id) references employee,
 Foreign Key (transaction_id) references enterprise_finances,
 Foreign Key (transaction_id) references labor_transaction,
 Check (transaction_type = 'labor') );
```

```
Create table ingredient_expenses (
 transaction_type varchar (20) not null,
 transaction_id varchar (15),
 item varchar (50),
 farmer_id varchar (10),
 ingredient varchar (50),
 Primary key (transaction_id, item, farmer_id, ingredient),
 Foreign key (transaction_id) references ingredient_transaction,
 Foreign key (transaction_id) references enterprise_finances,
 Foreign Key (farmer_id, ingredient) references farmers,
 Foreign Key (item) references ingredients,
 Check (transaction_type = 'food_expenses') );
```

```
Create table food_sales (
 transaction_type varchar (20) not null,
 transaction_id varchar (15),
 timestamp timestamp,
 customer_id varchar (10),
 recipe_name varchar (50),
 Primary key (transaction_id, timestamp, customer_id,
recipe_name),
 Foreign Key (transaction_id, timestamp) references
order_transaction,
 Foreign Key (transaction_id) references enterprise_finances,
 Foreign key (customer_id) references customer,
 Foreign key (recipe_name) references menu_items,
 Check (transaction_type = 'revenue') );
```