

DELIVERABLE -2 New tables :

1) Each restaurant supplies one to many menu items. Restaurants are limited to offering up to 10 items for this prototype (think meals like a Cook Out tray). Menu items should have an identifying number, name, description, price, etc.

SQL SCRIPT:

```
CREATE TABLE IF NOT EXISTS `campus_eats_fall2020`.`menu_items` (  
  `item_id` INT NOT NULL,  
  `restaurant_id` INT NOT NULL,  
  `name` VARCHAR(45) NOT NULL,  
  `description` VARCHAR(150) NOT NULL,  
  `price` DECIMAL(10,2) NOT NULL,  
  PRIMARY KEY (`item_id`),  
  INDEX `restaurant_id_idx` (`restaurant_id` ASC) VISIBLE,  
  CONSTRAINT `restaurant_id`  
    FOREIGN KEY (`restaurant_id`)  
    REFERENCES `campus_eats_fall2020`.`restaurant` (`restaurant_id`))
```

ENGINE = InnoDB

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'menu_items' selected under the 'campus_eats_fall2020' database. The main editor window shows the SQL script for creating the 'menu_items' table, which includes a primary key on 'item_id', a foreign key on 'restaurant_id' referencing the 'restaurant' table, and various data types for the other columns. The bottom 'Information' tab shows the table's structure, listing the columns and their data types and constraints.

Table: menu_items

Columns:	
item_id	int PK
restaurant_id	int
name	varchar(45)
description	varchar(150)
price	decimal(10,2)

Table 2:

Altered orders table to add timestamp, date and order status.

SQL SCRIPT:

```
ALTER TABLE `campus_eats_fall2020`.`order`  
  
ADD COLUMN `order_status_id` INT NOT NULL AFTER `delivery_charge`,  
  
ADD COLUMN `timestamp` DATETIME NOT NULL AFTER `order_status_id`,  
  
CHANGE COLUMN `delivery_charge` `delivery_charge` FLOAT NOT NULL ,  
  
ADD INDEX `fk_O_order_status_id_idx` (`order_status_id` ASC) VISIBLE;;  
  
ALTER TABLE `campus_eats_fall2020`.`order`  
  
ADD CONSTRAINT `fk_O_order_status_id`  
  
FOREIGN KEY (`order_status_id`)  
  
REFERENCES `campus_eats_fall2020`.`order_status` (`status_id`);
```

The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'location', 'menu_items', 'order', 'order_status', and 'payments'. The 'order' table is selected, showing its columns: 'order_id', 'person_id', 'delivery_id', 'location_id', 'driver_id', 'restaurant_id', 'total_price', 'delivery_charge', 'order_status_id', and 'timestamp'. The 'order_status_id' column is highlighted in green, and its definition is shown as 'order_status_id int'. The main pane displays a query: 'SELECT * FROM campus_eats_fall2020.order;'. The 'Result Grid' shows the first 5 rows of the 'order' table. The 'Message' pane at the bottom indicates that 101 rows were returned.

order_id	person_id	delivery_id	location_id	driver_id	restaurant_id	total_price	delivery_charge	order_status_id	timestamp
1	1	1	1	1	1	15.63	6.63	0	0000-00-00 00:00:00
2	2	2	2	2	2	18.03	9.43	0	0000-00-00 00:00:00
3	3	3	3	3	3	11.91	7.42	0	0000-00-00 00:00:00
4	4	4	4	4	4	19.13	6.26	0	0000-00-00 00:00:00
5	5	5	5	5	5	13.76	6.24	0	0000-00-00 00:00:00

3. order rating table:

SQL SCRIPT:

```
CREATE TABLE IF NOT EXISTS `campus_eats_fall2020`.`order_rating` (  
  
`id` INT NOT NULL,  
  
`order_id` INT NOT NULL,  
  
`food_rating` INT NULL,
```

```

`delivery_rating` INT NULL,

`comments` VARCHAR(200) NULL,

`picture` VARCHAR(100) NULL,

PRIMARY KEY (`id`),

INDEX `order_id_idx` (`order_id` ASC) VISIBLE,

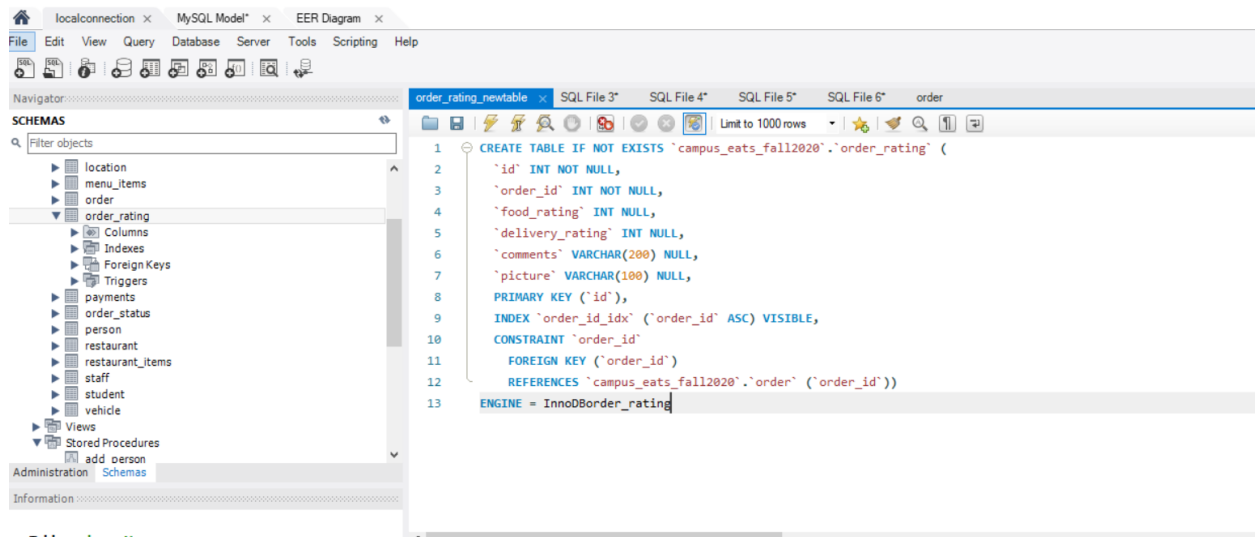
CONSTRAINT `order_id`

FOREIGN KEY (`order_id`)

REFERENCES `campus_eats_fall2020`.`order` (`order_id`))

ENGINE = InnoDB

```



4.payments table:

SQL SCRIPT:

```

CREATE TABLE IF NOT EXISTS `campus_eats_fall2020`.`payments` (

`payment_id` INT NOT NULL AUTO_INCREMENT,

`order_id` INT NOT NULL,

`cust_id` INT NOT NULL,

`amount` FLOAT NOT NULL,

`delivery_charges` FLOAT NOT NULL,

PRIMARY KEY (`payment_id`),

INDEX `_idx` (`order_id` ASC) VISIBLE,

```

```

INDEX `cust_id_idx` (`cust_id` ASC) VISIBLE,

CONSTRAINT ``

FOREIGN KEY (`order_id`)

REFERENCES `campus_eats_fall2020`.`order` (`order_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

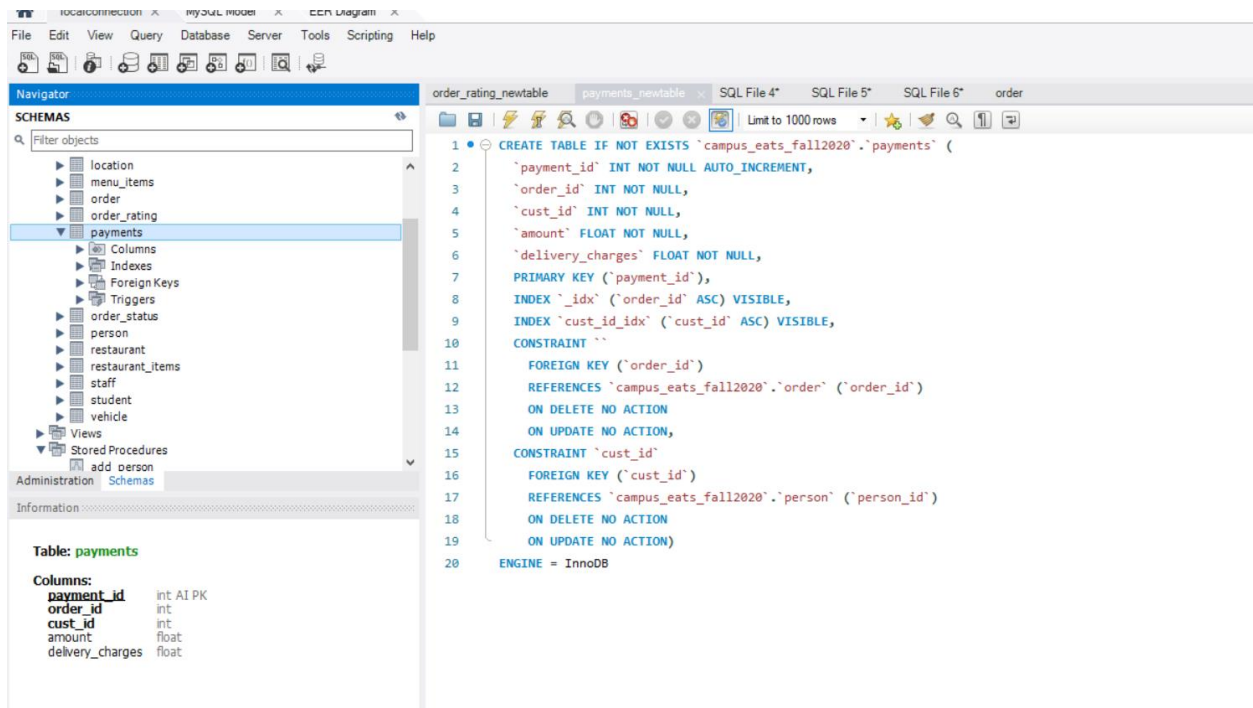
CONSTRAINT `cust_id`

FOREIGN KEY (`cust_id`)

REFERENCES `campus_eats_fall2020`.`person` (`person_id`))

ENGINE = InnoDB

```



5.order status

SQL SCRIPT:

```

CREATE TABLE IF NOT EXISTS `campus_eats_fall2020`.`order_status` (

`status_id` INT NOT NULL,

`status_name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`status_id`))

ENGINE = InnoDB

```

localconnection x MySQL Model* x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- location
- menu_items
- order
- order_rating
- order_status
- payments
- order_status
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- person
- restaurant
- restaurant_items
- staff
- student
- vehicle
- Views
- Stored Procedures

order_rating_newtable payments_newtable menu_items_newtable order_status_newtable x SQL File 6*

```
1 CREATE TABLE IF NOT EXISTS `campus_eats_fall2020`.`order_status` (  
2   `status_id` INT NOT NULL,  
3   `status_name` VARCHAR(50) NOT NULL,  
4   PRIMARY KEY (`status_id`))  
5 ENGINE = InnoDB
```