EAST WEST UNIVERSITY

# Project Report

## Project title: Restaurant Management Database System

Course: CSE301

Title: Database Systems

Section: 02

Semester: Spring 2021

Submitted To:

Md Mostofa Kamal Rasel, PhD

Assistant Professor

Department of

Computer Science and Engineering

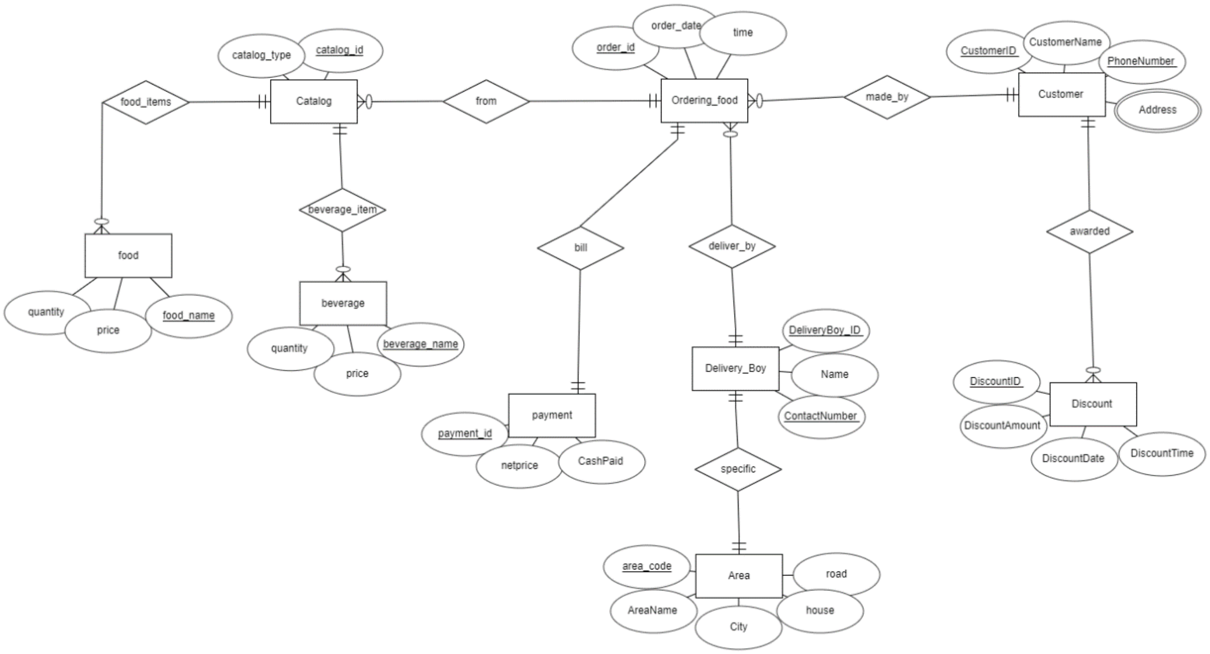
Submitted by:

| | |
|---|---|
| 2018-1-60-075 | Md. Moniruzzaman Shanto |
| 2017-2-68-001 | Shimanto Krishna Chakroborty |
| 2018-1-60-026 | Maisha Jannat |

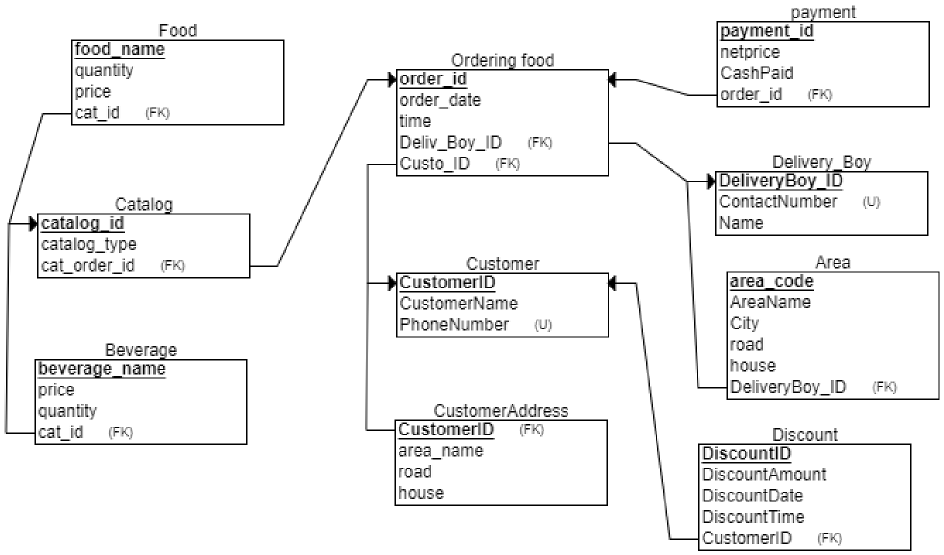
Date of Submission: 25/05/2021

**Problem Statement:** The restaurant maintains the catalog for the list of food and beverage items that it provides. Apart from providing food facility at their own premises, the restaurant takes orders online through their site. Orders on the phone are also entertained. To deliver the orders, we have delivery boys. Each delivery boy is assigned to the specific area code. The delivery boy cannot deliver outside the area which is not assigned to the delivery boy (for every delivery boy there can be a single area assigned to that delivery boy). The customer record is maintained so that premium customer can be awarded discounts.

**ER Diagram:**



**Relational Schema:**



Food
**food_name**
quantity
price
cat_id (FK)

Catalog
**catalog_id**
catalog_type
cat_order_id (FK)

Beverage
**beverage_name**
price
quantity
cat_id (FK)

Ordering food
**order_id**
order_date
time
Deliv_Boy_ID (FK)
Custo_ID (FK)

Customer
**CustomerID**
CustomerName
PhoneNumber (U)

CustomerAddress
**CustomerID** (FK)
area_name
road
house

payment
**payment_id**
netprice
CashPaid
order_id (FK)

Delivery_Boy
**DeliveryBoy_ID**
ContactNumber (U)
Name

Area
**area_code**
AreaName
City
road
house
DeliveryBoy_ID (FK)

Discount
**DiscountID**
DiscountAmount
DiscountDate
DiscountTime
CustomerID (FK)

❖ **Entity sets in the ERD diagram:**

1. Ordering_food
2. Delivery_Boy
3. Area
4. Customer
5. Discount
6. payment
7. Catalog
8. Food
9. Beverage

❖ **Relationships:**

1. One to many relations between Ordering_food and Catalog entity.
2. many to one relation between Ordering_food and Customer entity.
3. One to one relation between Ordering_food and payment entity.
4. many to one relation between Ordering_food and Delivery_Boy entity.
5. many to one relation between food and Catalog entity.
6. many to one relation between beverage and Catalog entity.
7. One to one relation between Delivery_Boy and Area entity.
8. One to many relations between Customer and Discount entity.

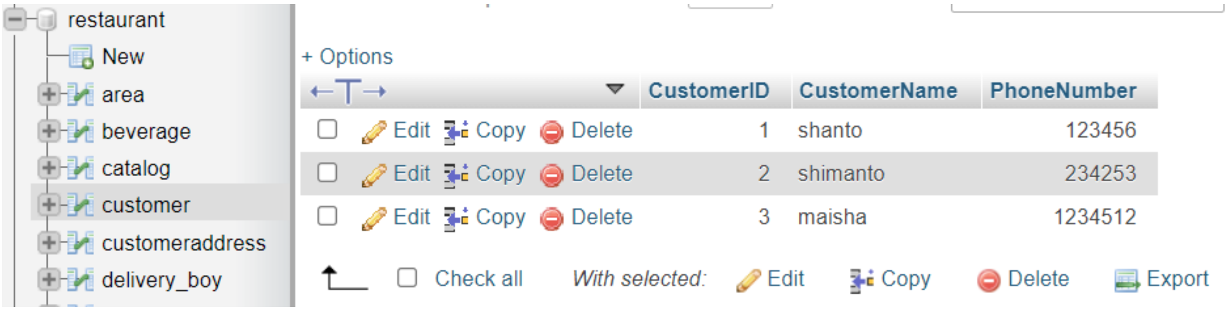
❖ **Tasks performed into the database:**

**1. INSERT query:**

INSERT INTO `customer`(`CustomerID`, `CustomerName`, `PhoneNumber`) VALUES (0001,'shanto',123456);

INSERT INTO `customer`(`CustomerID`, `CustomerName`, `PhoneNumber`) VALUES (0002,'shimanto',234253);

INSERT INTO `customer`(`CustomerID`, `CustomerName`, `PhoneNumber`) VALUES (0003,'maisha',1234512);

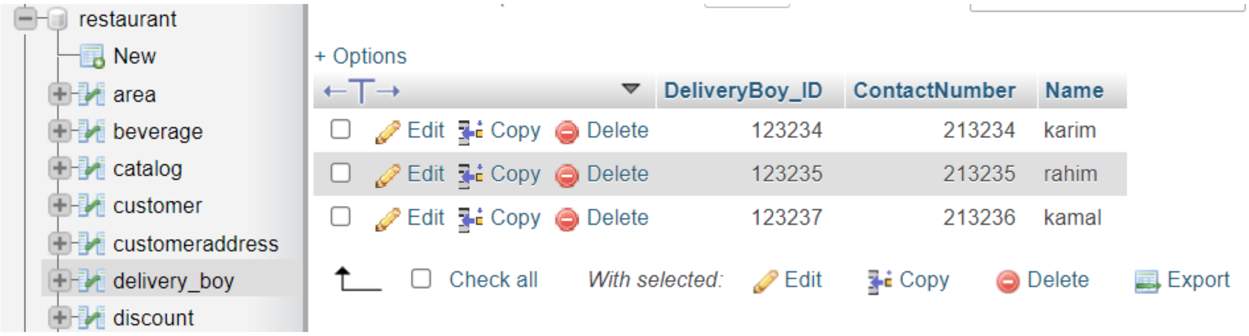


| CustomerID | CustomerName | PhoneNumber |
|---|---|---|
| 1 | shanto | 123456 |
| 2 | shimanto | 234253 |
| 3 | maisha | 1234512 |

INSERT INTO `delivery_boy`(`DeliveryBoy_ID`, `ContactNumber`, `Name`) VALUES (123234,0213234,'karim');

INSERT INTO `delivery_boy`(`DeliveryBoy_ID`, `ContactNumber`, `Name`) VALUES (123235,0213235,'rahim');

INSERT INTO `delivery_boy`(`DeliveryBoy_ID`, `ContactNumber`, `Name`) VALUES (123237,0213236,'kamal');

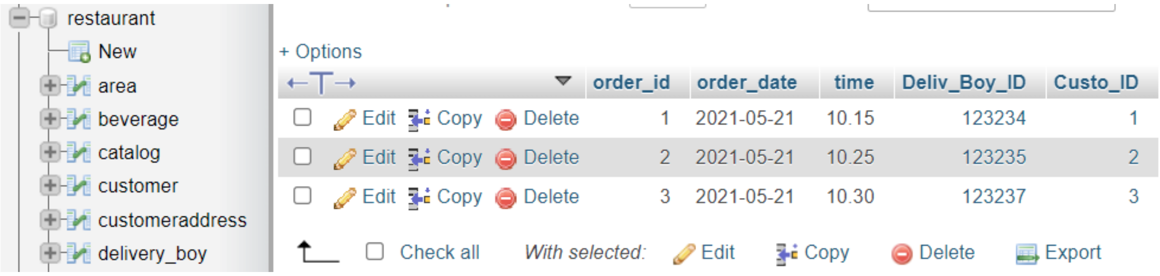


| DeliveryBoy_ID | ContactNumber | Name |
|---|---|---|
| 123234 | 213234 | karim |
| 123235 | 213235 | rahim |
| 123237 | 213236 | kamal |

INSERT INTO `ordering_food`(`order_id`, `order_date`, `time`, `Deliv_Boy_ID`, `Custo_ID`) VALUES (001,CURRENT_DATE,10.15,123234,0001);

INSERT INTO `ordering_food`(`order_id`, `order_date`, `time`, `Deliv_Boy_ID`, `Custo_ID`) VALUES (002,CURRENT_DATE,10.25,123235,0002);

INSERT INTO `ordering_food`(`order_id`, `order_date`, `time`, `Deliv_Boy_ID`, `Custo_ID`) VALUES (003,CURRENT_DATE,10.30,123237,0003);

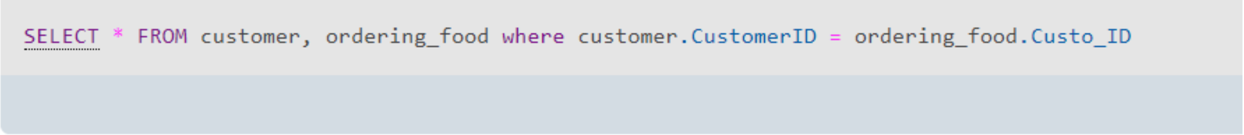


| order_id | order_date | time | Deliv_Boy_ID | Custo_ID |
|---|---|---|---|---|
| 1 | 2021-05-21 | 10.15 | 123234 | 1 |
| 2 | 2021-05-21 | 10.25 | 123235 | 2 |
| 3 | 2021-05-21 | 10.30 | 123237 | 3 |

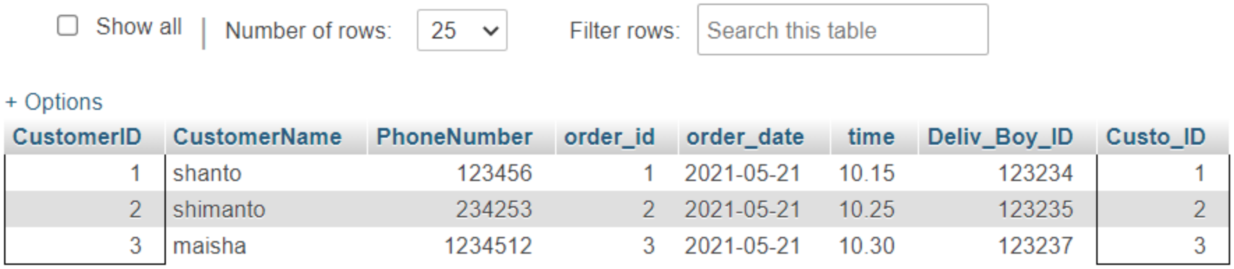
**Insert Query description:** Here we used DDL command INSERT to add new rows into the customer, delivery_boy and ordering_food table to perform tasks. If we wanted to insert any data for every tuple into a table We have to insert in a specific way, the way we have created the table. We also have to remember the primary key and unique key because it cannot be null. Here, CustomerID, DeliveryBoy_ID, order_id these are primary key and we must have to insert value for these.

### 2. Join operation between customer and ordering_food table:

SELECT * FROM customer, ordering_food where customer.CustomerID = ordering_food.Custo_ID



```
SELECT * FROM customer, ordering_food where customer.CustomerID = ordering_food.Custo_ID
```



☐ Show all | Number of rows: 25 Filter rows: Search this table

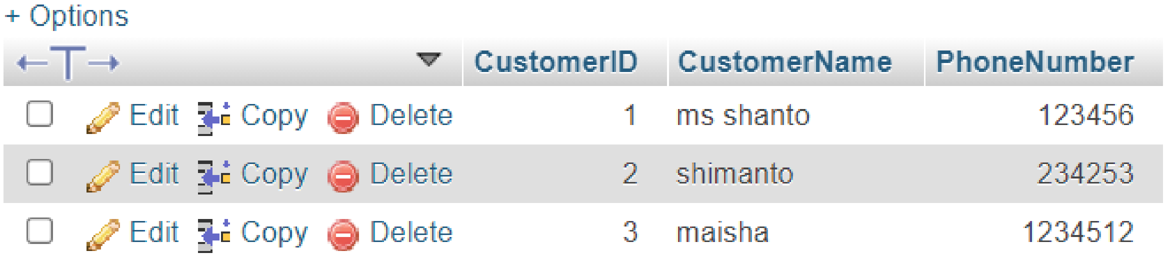
+ Options

| CustomerID | CustomerName | PhoneNumber | order_id | order_date | time | Deliv_Boy_ID | Custo_ID |
|---|---|---|---|---|---|---|---|
| 1 | shanto | 123456 | 1 | 2021-05-21 | 10.15 | 123234 | 1 |
| 2 | shimanto | 234253 | 2 | 2021-05-21 | 10.25 | 123235 | 2 |
| 3 | maisha | 1234512 | 3 | 2021-05-21 | 10.30 | 123237 | 3 |

**Description:** Ordering_food table has foreign key named Custo_ID which is same as the primary key named CustomerID in the Customer table. Here we have performed simple join operation between 'customer' and 'ordering_food' table. We selected all the columns of 'customer' and 'ordering_food' table and merged them into one table by using SELECT Query. We used it to retrieve or access data from the 'customer' and 'ordering_food' tables. We also used 'where' clause to specify which columns will be used to merge both tables. We used Custo_ID and CustomerID to specify and it will help to select all the rows if both becomes same.

### 3. UPDATE query:

UPDATE `customer` SET `CustomerName`='ms shanto' WHERE CustomerName = 'shanto';



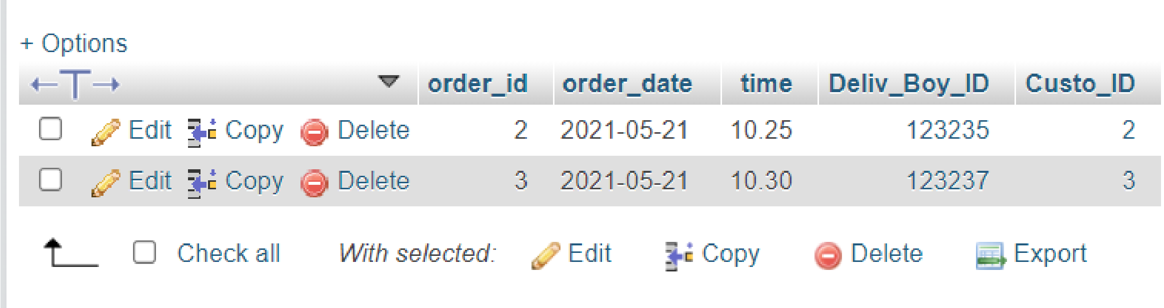
+ Options

| ←T→ | CustomerID | CustomerName | PhoneNumber |
|---|---|---|---|
| ☐ Edit Copy Delete | 1 | ms shanto | 123456 |
| ☐ Edit Copy Delete | 2 | shimanto | 234253 |
| ☐ Edit Copy Delete | 3 | maisha | 1234512 |

**Description:** Here we used UPDATE query to modify the record presented into the 'customer' table. UPDATE is a DML command which helps to change the existing data into any table. Here, CustomerName attribute is used after WHERE to make a condition and if the condition satisfy CustomerName will be changed.

### 4. DELETE query:

DELETE FROM `ordering_food` WHERE order_id =1;



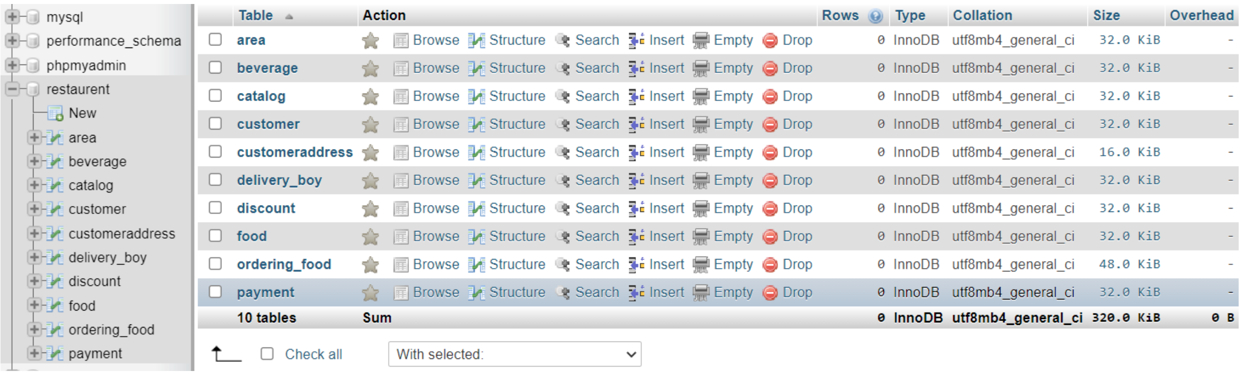
| | order_id | order_date | time | Deliv_Boy_ID | Custo_ID |
|---|---|---|---|---|---|
| □ Edit Copy Delete | 2 | 2021-05-21 | 10.25 | 123235 | 2 |
| □ Edit Copy Delete | 3 | 2021-05-21 | 10.30 | 123237 | 3 |

**Description:** Here, we used DELETE command to remove rows from ‘ordering_food’ table. We used to delete so that we can get the deleted data in future if it is needed. By using Delete command we can remove any rows from table but columns name remains. Here, after WHERE we are using a condition by using primary key order_id to specify which row has to delete.

### ❖ SQL code:

```
CREATE TABLE Delivery_Boy
(
  DeliveryBoy_ID INT NOT NULL,
  ContactNumber INT NOT NULL,
  Name VARCHAR(32) NOT NULL,
  PRIMARY KEY (DeliveryBoy_ID),
  UNIQUE (ContactNumber)
);

CREATE TABLE Area
(
  AreaName VARCHAR(32) NOT NULL,
  City VARCHAR(32) NOT NULL,
  road INT NOT NULL,
  area_code INT NOT NULL,
  house INT NOT NULL,
  DeliveryBoy_ID INT NOT NULL,
  PRIMARY KEY (area_code),
  FOREIGN KEY (DeliveryBoy_ID) REFERENCES Delivery_Boy(DeliveryBoy_ID)
);

CREATE TABLE Customer
(
  CustomerID INT NOT NULL,
  CustomerName VARCHAR(32) NOT NULL,
  PhoneNumber INT NOT NULL,
  PRIMARY KEY (CustomerID),
  UNIQUE (PhoneNumber)
);

CREATE TABLE Discount
(
  DiscountID INT NOT NULL,
  DiscountAmount NUMERIC(6,2) NOT NULL,
  DiscountDate DATE NOT NULL,
  DiscountTime NUMERIC(4,2) NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (DiscountID),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE CustomerAddress
(
  area_name VARCHAR(32) NOT NULL,
  road INT NOT NULL,
  house INT NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (CustomerID),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE Ordering_food
(
  order_id INT NOT NULL,
  order_date DATE NOT NULL,
  time NUMERIC(4,2) NOT NULL,
  Deliv_Boy_ID INT NOT NULL,
  Custo_ID INT NOT NULL,
  PRIMARY KEY (order_id),
  FOREIGN KEY (Deliv_Boy_ID) REFERENCES Delivery_Boy(DeliveryBoy_ID),
  FOREIGN KEY (Custo_ID) REFERENCES Customer(CustomerID)
);

CREATE TABLE payment
(
  payment_id INT NOT NULL,
  netprice NUMERIC(8,2) NOT NULL,
  CashPaid NUMERIC(8,2) NOT NULL,
  order_id INT NOT NULL,
  PRIMARY KEY (payment_id),
  FOREIGN KEY (order_id) REFERENCES Ordering_food(order_id)
);

CREATE TABLE Catalog
(
  catalog_id INT NOT NULL,
  catalog_type VARCHAR(32) NOT NULL,
  cat_order_id INT NOT NULL,
  PRIMARY KEY (catalog_id),
  FOREIGN KEY (cat_order_id) REFERENCES Ordering_food(order_id)
);

CREATE TABLE Food
(
  quantity INT NOT NULL,
  price NUMERIC(6,2) NOT NULL,
  food_name VARCHAR(32) NOT NULL,
  cat_id INT NOT NULL,
  PRIMARY KEY (food_name),
  FOREIGN KEY (cat_id) REFERENCES Catalog(catalog_id)
);

CREATE TABLE Beverage
(
  beverage_name VARCHAR(32) NOT NULL,
  price NUMERIC(6,2) NOT NULL,
  quantity INT NOT NULL,
  cat_id INT NOT NULL,
  PRIMARY KEY (beverage_name),
  FOREIGN KEY (cat_id) REFERENCES Catalog(catalog_id)
);
```

## ❖ All the tables:



- mysql
- performance_schema
- phpmyadmin
- restaurent
  - New
  - area
  - beverage
  - catalog
  - customer
  - customeraddress
  - delivery_boy
  - discount
  - food
  - ordering_food
  - payment

| | Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|---|
| □ | area | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | beverage | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | catalog | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | customer | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | customeraddress | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| □ | delivery_boy | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | discount | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | food | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| □ | ordering_food | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| □ | payment | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| | 10 tables | Sum | 0 | InnoDB | utf8mb4_general_ci | 320.0 KiB | 0 B |

□ Check all With selected: