IS3331 Database Management

Group Project -- Database Design Report

Tutorial Session: L01

Group 3

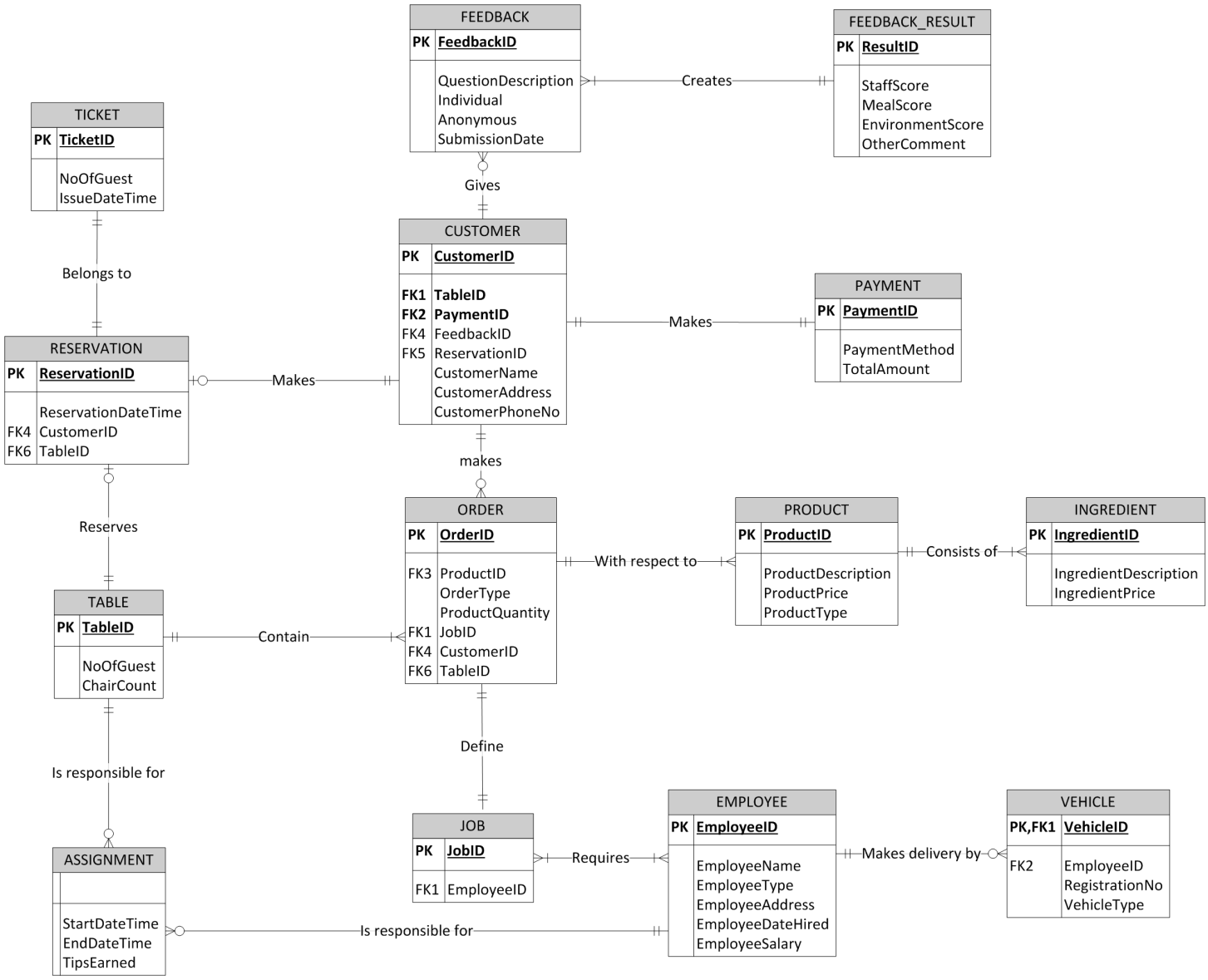
Student Name: Chow Sai Yat Morris 54796760

Table of Content

Introduction:

Cheesecake factory is a company based in United States and have expanded its branches to Hong Kong in early April this year, which is located in Tsim Sha Tsui, ground floor of Harbour City. It has been well known for its large servings and signature cheesecakes. We have developed an ER diagram mainly focuses on its ordering processes, employee's' job and table assignments as well as the customer feedback system. Furthermore, we have also added a ticketing function for customers' convenience.

E-R Diagram:



Relationships:

- The restaurant provides table reservation service by distributing tickets to the waiting customers. The identifier of reservation is Reservation ID, whereas the identifier of ticket is Ticket ID. We identify the following additional attributes for reservation: ReservationDateTime, Customer ID and Table ID. Another attributes for ticket are NoOfGuest and IssueDateTime. Each ticket must belong to one reservation.

- Customers can book table in advance using the reservation service. The identifier of table is Table ID and its attributes are NoOfGuest and Chair Count. Each reservation must reserve one table, but not all tables are being reserved.

- The restaurant serves a number of customers. The customer entity is developed to store the record of customers in the database. The identifier of customer is Customer ID, and another attributes are Table ID, Payment ID, Reservation ID, Feedback ID, Customer Name, Customer Address, and Customer PhoneNo. A reservation can be made by a customer, but a customer do not necessarily have to make a reservation in order to dine at the restaurant. Moreover, employees are being assigned to serve certain tables and the entity "Assignment" specify their starting and ending datetime of work as well as the tips earned.

- Regarding to the orders placed by customers, the order entity is developed. The identifier of order is Order ID, and its attributes are Product ID, Table ID, Job ID, Customer ID, Order Type, and Product Quantity. Each order must be placed by a customer, but a customer can place any number of orders.

- The identifier of product is Product ID and its attributes are Product Description, ProductType and Product Price. Each product belongs to one order, but an order can consist of more than one product. For instance, a customer placed an order in the Cheesecake Factory, and he ordered more than one products, which are pasta carbonara and strawberry cheesecake.

- After order is being taken, ingredients are essential for the preparation of meal, so the there is the ingredient entity with Ingredient ID as the identifier, Ingredient Price and Ingredient Description as the attributes.

- Employees are assigned to do the jobs. For each job, at least one or many employees are required, and each employee is required to do at least one or more than one jobs. The identifier of job is Job ID, whereas the identifier of employee is Employee ID and OrderType. We identify the Employee ID as the additional attribute for job. On the other hand, attributes for entity employee are Employee Name, Employee Type, Employee Address, Employee Date Hired, and Employee Salary.

- While the associative entity ASSIGNMENT links the TABLE and EMPLOYEE entities. It is to indicate that employees are being assigned to serve no or many tables.

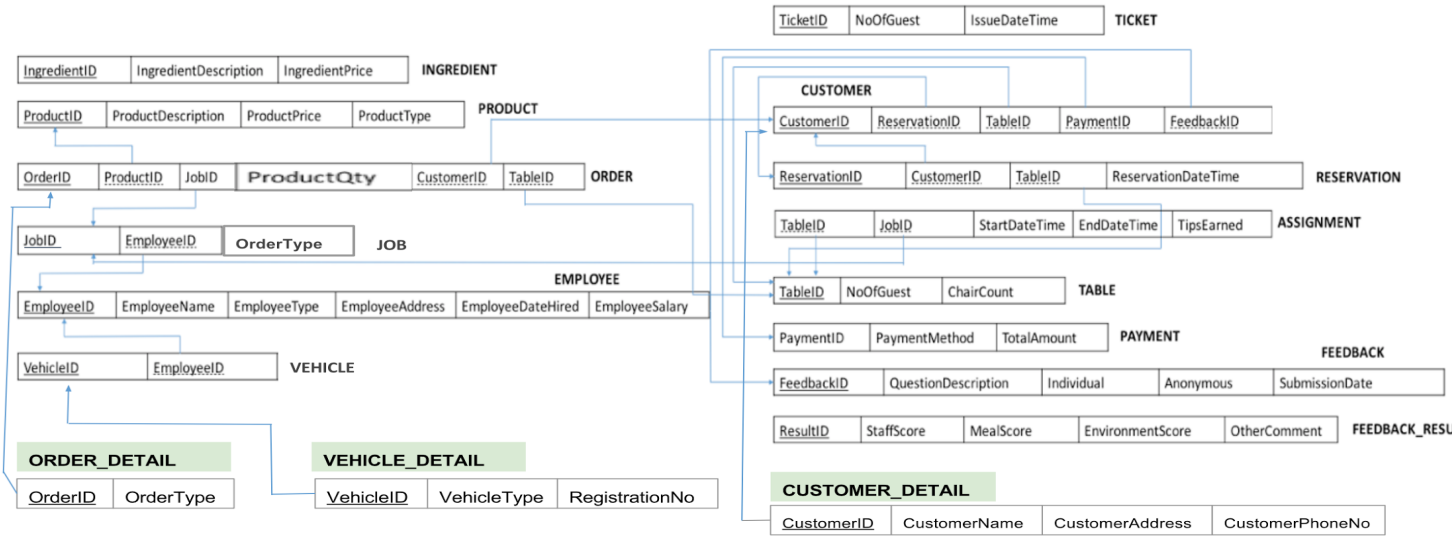
- Part of the employees who are the deliveryman will be assigned to vehicles for meal delivery. In order to create a database to store the vehicles information,,the entity “Vehicle” exists to facilitate the usage of company vehicles, with Vehicle ID as the identifier and Employee ID, Vehicle Registration No, and Vehicle Type as the attributes.

- When customers finished their meals, they have to pay the bills. For the payment entity, the identifier is Payment ID, and its attributes are Payment Method and Total Amount.

- Feedback forms are distributed to customers for the better understanding of customers’ needs and service improvements. The feedback entity is developed to store the information of each collected feedback form, with Feedback ID as the identifier and Question Description, Individual respondent, Anonymous respondent, and Submission Date as the attributes. Each feedback form is submitted by one customer, but a customer can give any number of feedbacks to the restaurant.

- The feedback result entity is created to store the details of the customer feedbacks. Its identifier is Reuslt ID and its attributes are Staff Score, Meal Score, Environment Score, and Other Comment. Each feedback creates one and only one feedback result while each feedback result is created by at least one or many feedbacks.

Schema:



SQL Table Design:

1.Ticket:

```
create table Ticket (
TicketID varchar (4) ,
NoOfGuest int not null,
IssueDateTime datetime not null,
constraint Ticket_PK primary key (TicketID)
);
```

2. Reservation:

```
create table Reservation (
ReservationID varchar (4) ,
TableID varchar (4) not null,
ReservationDateTime datetime not null,
CustomerID varchar (6) not null,
constraint Reservation_PK primary key (ReservationID),
constraint Reservation_FK4 foreign key (TableID) references Table(TableID),
constraint Reservation_FK6 foreign key (CustomerID) references Customer(CustomerID));
```

3. Dining_Table:

```
create table Dining_Table (
TableID varchar (4) ,
NoOfGuest int not null,
ChairCount int not null,
constraint Dining_Table_PK primary key (TableID));
```

4. Assignment:

```
create table Assignment (
TableID varchar not null (4),
StartDateTime datetime not null,
EndDateTime datetime not null,
TipsEarned int not null
Constraint Dining_Table);
```

5. Customer:

```
create table Customer (
CustomerID varchar (6) ,
TableID varchar (4) not null ,
PaymentID varchar (6) not null,
ReservationID varchar (4) not null,
FeedbackID varchar (4) not null,
CustomerAddress char (100) not null,
CustomerName char (100) not null,
CustomerPhoneNo char  (100) not null,
constraint Customer_PK primary key (CustomerID),
constraint Customer_FK1 foreign key (TableID) references Dining_Table (TableID),
constraint Customer_FK2 foreign key (PaymentID) references Payment (PaymentID),
constraint Customer_FK4 foreign key (FeedbackID) references Feedback (FeedbackID),
constraint Customer_FK5 foreign key (ReservationID) references Reservation (ReservationID));
```

6. Orders:

```
create table Orders (
OrderID varchar (4) ,
ProductID varchar (4)not null,
TableID varchar (4)not null,
JobID varchar (5) not null,
CustomerID varchar (6) not null,
OrderType char  (100)not null
        Check (OrderType IN ('Take-away' , 'Dine-in')),
ProductQuantity int not null,
constraint Order_PK primary key (OrderID),
constraint Order_FK1 foreign key (JobID) references Job (JobID),
constraint Order_FK3 foreign key (ProductID) references Product(ProductID),
constraint Order_FK4 foreign key (CustomerID) references Customer(CustomerID),
constraint Order_FK6 foreign key (TableID) references Dining_Table (TableID));
```

7. Product:

```
create table Product (
ProductID varchar (4),
ProductDescription char (100) not null,
ProductPrice int not null,
ProductType char (100)not null,
constraint Product_PK primary key (ProductID));
```

8. Ingredient:

```
create table Ingredient (
IngredientID varchar (10),
IngredientPrice int not null,
IngredientDescription char (100) not null,
constraint Ingredient_PK primary key (IngredientID));
```

9. Job:

```
create table Job (
JobID varchar (5),
EmployeeID varchar (6)not null,
constraint Job_PK primary key (JobID));
```

10. Employee:

```
create table Employee (
EmployeeID varchar(6),
EmployeeName char (100) not null,
EmployeeType char (1)not null,
EmployeeAddress char (100) not null,
EmployeeDateHired datetime not null,
EmployeeSalary int not null,
constraint Employee_PK primary key (EmployeeID));
```

11. Vehicle:

```
create table Vehicle (
VehicleID varchar(3),
EmployeeID varchar (6) not null,
VehicleType char (100)not null,
RegistrationNo char (100)not null,
constraint Vehicle_PK primary key (VehicleID),
constraint Vehicle_FK2 foreign key (EmployeeID) references Employee (EmployeeID));
```

12. Payment:

```
create table Payment (
PaymentID varchar(6),
PaymentMethod char (100) not null,
TotalAmount int not null,
constraint Payment_PK primary key (PaymentID));
```

13. Feedback:

```
create table Feedback (
FeedbackID varchar(4),
SubmissionDate datetime not null,
Individual varchar (20)not null,
Anonymous varchar (20)not null,
QuestionDescription char (100)not null,
constraint Feedback_PK primary key (FeedbackID));
```

13. Feedback_result:

```
create table Feedback_Result (
ResultID varchar(8),
StaffScore int  not null,
MealScore int  not null,
EnvironmentScore int  not null,
OtherComment char (100)not null,
constraint Feedback_Result_PK primary key (ResultID));
```

Insert Sample Data:

1.Ticket:

```
insert into Ticket values ('T001' , '5' , '2017-10-10 12:12:12')
insert into Ticket values ('T002' , '2' , '2017-10-11 18:45:00')
insert into Ticket values ('T003' , '3' , '2017-10-12 19:00:00')
insert into Ticket values ('T004' , '1' , '2017-10-13 19:30:00')
insert into Ticket values ('T005' , '7' , '2017-10-14 18:45:00')
```

2. Dining_Table:

```
insert into Dining_Table values ('TB01' , '5' , '6' )
insert into Dining_Table values ('TB02' , '2' , '4' )
insert into Dining_Table values ('TB03' , '3' , '4' )
insert into Dining_Table values ('TB04' , '1' , '2' )
insert into Dining_Table values ('TB05' , '7' , '10')
```

3. Payment:

```
insert into Payment values('P00001' , 'Credit Card' , '1200')
insert into Payment values('P00002' , 'Cash' , '250')
insert into Payment values('P00003' , 'Credit Card','650')
insert into Payment values('P00004' , 'mobile payment' ,'236')
insert into Payment values('P00005' , 'Credit Card' , '1600')
```

4. FeedBack:

```
insert into Feedback values('001', '2017-10-10', 'IndividR00001', 'AnonyR00001', 'Nice')
insert into Feedback values('002', '2017-10-11', 'IndividR00002', 'AnonyR00002', 'Perfect')
insert into Feedback values('003', '2017-10-12', 'IndividR00003', 'AnonyR00003', 'Not bad')
insert into Feedback values('004', '2017-10-13', 'IndividR00004', 'AnonyR00004', 'Bad')
insert into Feedback values('005', '2017-10-14', 'IndividR00005', 'AnonyR00005', 'High quality')
```

5. Feedback_Result:

```
insert into Feedback_Result values('FBR00001', '8' , '8' , '10', 'provide larger food portion')
insert into Feedback_Result values('FBR00002', '10' , '10' , '10', 'It is very good')
insert into Feedback_Result values('FBR00003', '8' , '7' , '8', 'the amount of food should increase')
insert into Feedback_Result values('FBR00004', '9' , '3' , '8', 'The taste is not very good')
insert into Feedback_Result values('FBR00005', '6' , '10' , '7', 'service should be improved')
```

6. Product:

```
insert into Product values('P001','GODIVA® Chocolate Cheesecake', '79', 'Dessert')
insert into Product values('P002','Tiramisu', '79', 'Dessert')
insert into Product values('P003', 'Pasta Carbonara', '156', 'Pasta')
insert into Product values('P004', 'Fettuccini alfredo', '156', 'Pasta')
insert into Product values('P005', 'Caesar Salad', '80', 'Salad')
```

7. Ingredient:

```
insert into Ingredient values('CC01', '10', 'Cream Cheese')
insert into Ingredient values('Cho01', '5', 'Chocolate')
insert into Ingredient values('Co01', '7', 'Coffee')
insert into Ingredient values('Ba01', '5', 'Bacon')
insert into Ingredient values('Pe01', '4', 'peas')
insert into Ingredient values('Che01', '10', 'Cheese')
insert into Ingredient values('Pas01', '8', 'Pasta')
insert into Ingredient values('Par01', '10', 'Parsley')
insert into Ingredient values('S01', '11', 'Salad')
```

8. Employee:

```
insert into Employee values('E00001', 'Adrian Cheung', 'M', 'Tseung Kwan O', '2017-05-01', '25000')
insert into Employee values('E00002', 'John Chan', 'D', 'Kowloon Tong', '2017-07-07','13500')
insert into Employee values('E00003', 'Cristyne Ma', 'C', 'Central', '2017-08-08', '20000')
insert into Employee values('E00004', 'Sandy Chen', 'W', 'Causeway Bay', '2017-10-10', '14000')
insert into Employee values('E00005', 'Vicki Sui', 'W', 'Po Lam', '2017-05-29', '15000')
```

9. Job:

```
insert into Job values('J0001' , 'E00001')
insert into Job values('J0002' , 'E00002')
insert into Job values('J0003' , 'E00003')
insert into Job values('J0004' , 'E00004')
insert into Job values('J0005' , 'E00005')
```

10. Vehicle:

```
insert into Vehicle values('V01','E00001','Car','AC8888')
insert into Vehicle values('V02','E00002','Motorbike','JC1234')
insert into Vehicle values('V03','E00003','Car','CM1234')
insert into Vehicle values('V04','E00004','Bicycle','SC1234')
insert into Vehicle values('V05','E00005','Motorbike','VS1234')
```

11. Reservation:

```
insert into Reservation values ('R001' , 'TB01' , '2017-10-10 12:12:12' , 'C00001')
insert into Reservation values ('R002' , 'TB02' , '2017-10-11 18:45:00' , 'C00002')
insert into Reservation values ('R003' , 'TB03' , '2017-10-12 19:00:00' , 'C00003')
insert into Reservation values ('R004' , 'TB04' , '2017-10-13 19:30:00' , 'C00004')
insert into Reservation values ('R005' , 'TB05' , '2017-10-14 18:50:00' , 'C00005')
```

12. Customer:

```
insert into Customer values ('C00001' , 'TB01' , 'P00001', 'R001', '001', 'Yuen Long', 'Tommy Poon',
'93215678')
insert into Customer values ('C00002' , 'TB02' , 'P00002', 'R002', '002', 'Kwun Tong', 'Jian Ma',
'55443322')
insert into Customer values ('C00003' , 'TB03' , 'P00003', 'R003', '003', 'Wan Chai', 'Mary Lam',
'93435656')
insert into Customer values ('C00004' , 'TB04' , 'P00004', 'R004', '004', 'Tsim Sha Tsui', 'Marco Au',
'56781234')
insert into Customer values ('C00005' , 'TB05' , 'P00005', 'R005', '005', 'Hung Hom', 'David Hui',
'64159763')
```

13. Orders:

```
insert into Orders values ('O001', 'P001', 'TB01', 'J0001', 'C00001', 'Dine-in', '3')
insert into Orders values ('O002', 'P002', 'TB01', 'J0001', 'C00001', 'Take-away', '2')
insert into Orders values ('O003', 'P003', 'TB01', 'J0001', 'C00001', 'Dine-in', '3')
insert into Orders values ('O004', 'P004', 'TB01', 'J0001', 'C00001', 'Dine-in', '1')
insert into Orders values ('O005', 'P005', 'TB01', 'J0001', 'C00001', 'Dine-in', '2')

insert into Orders values ('O006', 'P004', 'TB02', 'J0004', 'C00002', 'Dine-in', '1')
insert into Orders values ('O007', 'P005', 'TB02', 'J0004', 'C00002', 'Dine-in', '1')

insert into Orders values ('O008', 'P001', 'TB03', 'J0005', 'C00003', 'Dine-in', '2')
insert into Orders values ('O009', 'P003', 'TB03', 'J0005', 'C00003', 'Dine-in', '2')
insert into Orders values ('O010', 'P005', 'TB03', 'J0005', 'C00003', 'Take-away', '2')

insert into Orders values ('O011', 'P004', 'TB04', 'J0003', 'C00004', 'Take-away', '1')
insert into Orders values ('O012', 'P005', 'TB04', 'J0004', 'C00004', 'Dine-in', '1')

insert into Orders values ('O013', 'P001', 'TB05', 'J0001', 'C00005', 'Dine-in', '2')
insert into Orders values ('O014', 'P002', 'TB05', 'J0001', 'C00005', 'Dine-in', '2')
insert into Orders values ('O015', 'P003', 'TB05','J0001', 'C00005', 'Dine-in', '4')
insert into Orders values ('O016', 'P004', 'TB05', 'J0001', 'C00005', 'Dine-in', '2')
insert into Orders values ('O017', 'P005', 'TB05', 'J0001', 'C00005', 'Dine-in', '4')
```

14. Assignment:

```
insert into Assignment values ('TB01','2017-10-10 12:12:12', '2017-10-10 14:30:00', '21')
insert into Assignment values ('TB02','2017-10-11 18:45:00', '2017-10-11 19:45:00', '14')
insert into Assignment values ('TB03','2017-10-12 19:00:00', '2017-10-12 21:00:00', '20')
insert into Assignment values ('TB04','2017-10-13 19:35:00', '2017-10-13 20:10:00', '0')
insert into Assignment values ('TB05','2017-10-14 18:45:00', '2017-10-14 21:45:00', '28')
```

Queries and Business Applications:

Business Application 1: Creating invoice

Query: Assemble the information necessary to create an invoice for order number O999.

```
SELECT Customer.CustomerID, CustomerName, Orders.OrderID, OrderType, ProductQuantity,
Product.ProductDescription, ProductPrice, (Orders.ProductQuantity*Product.ProductPrice)
FROM Customer, Orders, Product
WHERE Orders.CustomerID = Customer.CustomerID
AND Orders.ProductID = Product.ProductID
AND Orders.OrderID < 'O999';
```

Result:

C00001	Tommy Poon		O001	Dine-in		
	3	GODIVA® Chocolate Cheesecake			79	237
C00001	Tommy Poon		O002	Take-away		
	2	Tiramisu			79	158
C00001	Tommy Poon		O003	Dine-in		
	3	Pasta Carbonara			156	468
C00001	Tommy Poon		O004	Dine-in		
	1	Fettuccini alfredo			156	156
C00001	Tommy Poon		O005	Dine-in		
	2	Caesar Salad			80	160
C00002	Jian Ma		O006	Dine-in		
	1	Fettuccini alfredo			156	156
C00002	Jian Ma		O007	Dine-in		
	1	Caesar Salad			80	80
C00003	Mary Lam		O008	Dine-in		
	2	GODIVA® Chocolate Cheesecake			79	158
C00003	Mary Lam		O009	Dine-in		
	2	Pasta Carbonara			156	312
C00003	Mary Lam		O010	Take-away		
	2	Caesar Salad			80	160
C00004	Marco Au		O011	Take-away		
	1	Fettuccini alfredo			156	156
C00004	Marco Au		O012	Dine-in		
	1	Caesar Salad			80	80
C00005	David Hui		O013	Dine-in		
	2	GODIVA® Chocolate Cheesecake			79	158
C00005	David Hui		O014	Dine-in		
	2	Tiramisu			79	158
C00005	David Hui		O015	Dine-in		
	4	Pasta Carbonara			156	624

Business application 2: Creating sales report

Query: What is the revenue of each product? What are the products with the highest/lowest revenues?

```
SELECT Product.ProductID, ProductDescription, SUM(Orders.ProductQuantity) as TotalQuantity,
SUM(Product.ProductPrice * Orders.ProductQuantity)as Revenue
FROM Product, Orders
WHERE Orders.ProductID = Product.ProductID
Group by Product.ProductID,ProductDescription
Order by Revenue DESC
```

Result:

ProductID	ProductDescription	TotalQuantity	Revenue
P003	Pasta Carbonara	9	1404
P005	Caesar Salad	10	800
P004	Fettuccini alfredo	5	780
P001	GODIVA® Chocolate Cheesecake	7	553
P002	Tiramisu	4	316

Query: What are the highest, lowest, and average expenses of customer ?

```
SELECT MAX (TotalAmount) as HighestExpenses, MIN (TotalAmount) as LowestExpenses, AVG
(TotalAmount) as AverageExpenses
FROM  Payment;
```

Result:

HighestExpenses	LowestExpenses	AverageExpenses
1600	236	787

Query: Classify the price of products into cheap, moderate and expensive respectively

```
SELECT *, 'Classes of Product'  =
 CASE
        WHEN ProductPrice <50 THEN 'Cheap'
        WHEN ProductPrice Between 50 and 150 THEN ' Moderate'
       Else 'Expensive'
       END
FROM Product;
```

Result:

ProductID	ProductDescription	ProductPrice	ProductType	Classes of Product
P001	GODIVA® Chocolate Cheesecake	79	Dessert	Moderate
P002	Tiramisu	79	Dessert	Moderate
P003	Pasta Carbonara	156	Pasta	Expensive
P004	Fettuccini alfredo	156	Pasta	Expensive
P005	Caesar Salad	80	Salad	Moderate

Business Application 3: Analyzing customer feedbacks to enhance service quality

Query: How customers rate our service and what are the average score?

```
SELECT AVG(StaffScore) as AverageStaffScore, AVG(MealScore) as AverageMealScore,
AVG(EnvironmentScore) as AverageEnvironmentScore
From Feedback_Result;
```

Result:

AverageStaffScore	AverageMealScore	AverageEnviromentScore
5	7	8

Business Application 4: Understanding customer behaviour to adjust operation strategy

Query: List the total number of popular products ordered in different product type in the product table .

```
SELECT ProductType, (Count(ProductID)*Orders.ProductQuantity) as TimesOrdered,
ProductDescription, ProductPrice
FROM Product
Inner join Orders On Product.ProductID = Orders.ProductID
GROUP BY ProductType,  ProductDescription, ProductPrice, ProductQuantity
HAVING (Count(Product.ProductID)*Orders.ProductQuantity) > 0
ORDER BY ProductType;
```

Result:

ProductType	TimeOrdered	ProductDescription	ProductPrice
Dessert	7	GODIVA® Chocolate Cheesecake	79
Dessert	4	Tiramisu	79
Pasta	5	Fettuccini alfredo	156
Pasta	9	Pasta Carbonara	156
Salad	10	Caesar Salad	80

Query: What will a lone diner or a party of 2 diners order?

```
Select ProductID, Count(ProductID) as OrderTimes
From Orders
Inner Join Dining_Table on Orders.TableID = Dining_table.TableID
Where Dining_table.NoOfGuest in ('1','2')
Group by ProductID
```

Result:

ProductID	OrderTimes
P004	2
P005	2

Query: What are the numbers of guests during peak hours?

```
Select StartDateTime, EndDateTime, Dining_Table.NoOfGuest
From Assignment
Inner Join Dining_Table On Assignment.TableID = Dining_Table.TableID
Where StartDateTime > '18:00:00'
And EndDateTime < '22:00:00'
And Dining_Table.NoOfGuest >0
Order by NoOfGuest DESC;
```

Result:

StartDateTime	EndDateTime	NoOfGuest
2017-10-14 18:45:00.000	2017-10-14 21:45:00.000	7
2017-10-12 19:00:00.000	2017-10-12 21:00:00.000	3
2017-10-11 18:45:00.000	2017-10-11 19:45:00.000	2
2017-10-13 19:35:00.000	2017-10-13 20:10:00.000	1

Query: Which payment method is the most commonly used?

```
SELECT PaymentMethod, Count(PaymentID) as TotalAmount
From Payment
Group By PaymentMethod
Order By Count(PaymentID) desc;
```

Result:

PaymentMethod	TotalAmount
Credit Card	3
Mobile payment	1
Cash	1

Business Application 5: Managing employees

Query: List, in alphabetical order, the employee name and the average salary for each waiter and cook having an average salary over $15,000.

```
SELECT EmployeeName,  AVG (EmployeeSalary) as AverageEmployeeSalary

FROM Employee

WHERE EmployeeType IN ('W', 'C')

GROUP BY EmployeeName

HAVING  AVG (EmployeeSalary)>15000

ORDER BY EmployeeName;
```

Result:

EmployeeName	AverageEmployeeSalary
Cristyne Ma	20000

Query: Show the vehicle type for the couriers who earn more than $10,000.

```
SELECT Vehicle.EmployeeID, VehicleID, VehicleType
FROM Vehicle
        JOIN Employee
        ON Employee.EmployeeID = Vehicle.EMployeeID
        WHERE EmployeeSalary > 10000;
```

Result:

EmployeeID	VehicleID	VehicleType
E00001	V01	Car
E00002	V02	Motorbike
E00003	V03	Car
E00004	V04	Bicycle

E00005	V05	Motorbike

Query: How many employees have joined the Cheesecake Factory since 11/11/2016?

```
SELECT COUNT(*) EmployeeDateHired
FROM Employee
WHERE EmployeeDateHired >= '11-NOV-2016';
```

Result:

EmployeeDateHired
5

Appendix

Data Dictionary:

Table	PK/ FK	Attribute Name	Description	Example	Data Type
Ticket	PK	TicketID	Unique Identifier for each ticket issued.	“T001” for the first ticket issued	VarChar
		NoOfGuest	Number of guests per ticket	4 people for ticket number “T001”	Int
		IssueDateTime	The issue time of the ticket	12:12:12 10/10/2017	Time
Reservation	PK	ReservationID	Unique Identifier for each customer reservation.	“R001” for the first reservation made	VarChar
	FK	TableID	Unique Identifier for each table. (equivalent to table number)	“TB01”	VarChar
		ReservationDateTime	Time and date of the reservation.	12:12:12 10/10/2017	TimeDate
	FK	CustomerID	Unique Identifier for each customer.	“C00001” for the first customer	VarChar

Dinging_Table	PK	TableID	Unique Identifier for each table. (equivalent to table number)	“TB01”	VarChar
		NoOfGuest	Number of guest of the table	3 customers sitting at TB01	Int
		ChairCount	Number of seats of the table	4 seats for TB01	Int
Assignment	FK	TableID	Unique Identifier for each table. (equivalent to table number)	‘TB01’	Varchar
		StartDateTime	Admission time of customer	12:12:12 10/10/2017	Time
		EndDateTime	Leaving time of customer	12:12:12 10/10/2017	Time
		TipsEarned	Amount of tips earned by waiter	15	Int
Customer	PK	CustomerID	Unique Identifier for each customer.	“C00001” for the first customer	VarChar

	FK	TableID	Unique Identifier for each table. (equivalent to table number)	“TB01”	VarChar
	FK	PaymentID	Unique Identifier for each payment record.	“P00001” for the payment of the first bill	VarChar
	FK	ReservationID	Unique Identifier for each customer reservation.	“R001” for the first reservation made	VarChar
	FK	FeedbackID	Unique identifier for feedback	“001” for friendly service	VarChar
		CustomerAddress	The address of customers	Kowloon Tong	Char
		CustomerName	The name of customers	Jian Ma	Char
		CustomerPhoneNo	The phone number of customer	98765432	Char
Orders	PK	OrderID	Unique Identifier for each customer order.	“O001” for the first order placed	VarChar

	FK	ProductID	Unique Identifier for each meal.	“MS01” for chocolate milkshake	VarChar
	FK	TableID	Unique Identifier for each table. (equivalent to table number)	“TB01”	VarChar
	FK	JobID	Unique Identifier for the task needed to be assigned to employees.	“J0001” for the first job	VarChar
	FK	CustomerID	Unique Identifier for each customer.	“C00001” for the first customer	VarChar
		OrderType	Type of food order.	Take-away, Dine-in	Char
		ProductQuantity	Quantity of product ordered by the customer in each order	2	Int
Product	PK	ProductID	Unique Identifier for each meal.	“MS01” for chocolate milkshake	VarChar
		ProductDescription	Name of product	“Chocolate Milkshake”	Char
		ProductPrice	Price of product	12.5	int

		ProductType	Type of product	Snacks, Pasta, Dessert	Char
Ingredient	PK	IngredientID	Unique Identifier for each ingredient.	“FY01” for strawberry frozen yogurt	VarChar
		IngredientPrice	Price of ingredient	12.5	int
		IngredientDescription	Name of ingredient	“Strawberry frozen yogurt”	Char
Job	PK	JobID	Unique Identifier for the task needed to be assigned to employees.	“J0001” for the first job	VarChar
	FK	EmployeeID	Unique Identifier for each employee. (equivalent to staff number)	“E00001” for the first employee hired	VarChar
Employee	PK	EmployeeID	Unique Identifier for each employee. (equivalent to staff number)	“E00001” for the first employee hired	VarChar
		EmployeeName	Name of employee	Jian Ma	Char

		EmployeeType	Position of employee	“M” for manager, “D” for deliveryman, “C” for cook, “W” for waiter	Char
		EmployeeAddress	Address of employee	Kowloon Tong	Char
		EmployeeDateHired	Date joined of employee	10/10/2017	Date
		EmployeeSalary	Salary of employee	13500	Int
Vehicle	PK	VehicleID	Unique Identifier for each vehicle used for delivery.	“V01” for the first vehicle bought	VarChar
	FK	EmployeeID	Unique Identifier for each employee. (equivalent to staff number)	“E00001” for the first employee hired	VarChar
		VehicleType	Type of vehicle.	Motorbike, bicycle	Char
		RegistrationNo	License number of the vehicle	VH1234	Char
Payment	PK	PaymentID	Unique Identifier for each payment record.	“P00001” for the payment of the first bill	VarChar

		PaymentMethod	Type of payment.	Cash, credit card, mobile payment	Char
		Total Amount	Amount of the meal paid by customer.	50	Int
Feedback	PK	FeedbackID	Unique identifier for feedback	“001” for friendly service	VarChar
		SubmissionDate	Submission date of the feedback	10/10/2017	Time
		Individual	Feedback submitted by known customers such as our members	”IndividR00001” for the first known customer’s feedback	
		Anonymous	Feedback submitted by unknown customers	”AnonyR00001” for the first anonymous customer’s feedback (such as new customers or customers who do not willing to disclose their identities)	
		QuestionDescription	Detail of the question	”What do you think about the quality of our food?”	Char

Feedback_Result	PK	ResultID	Unique Identifier for each feedback result.	”FBR00001” for the first feedback result	VarChar
		StaffScore	Score graded by customers according to the attitude, behaviour and efficiency of staff	0-10	Int
		MealScore	Score graded by customers according to the taste, design and the worthiness of a particular meal	0-10	Int
		EnvironmentScore	Score evaluated by customers according to the light, declaration and atmosphere of the restaurant	0-10	Int
		OtherComment	Additional comment given by customer	E.g. “provide larger food portion”	Char