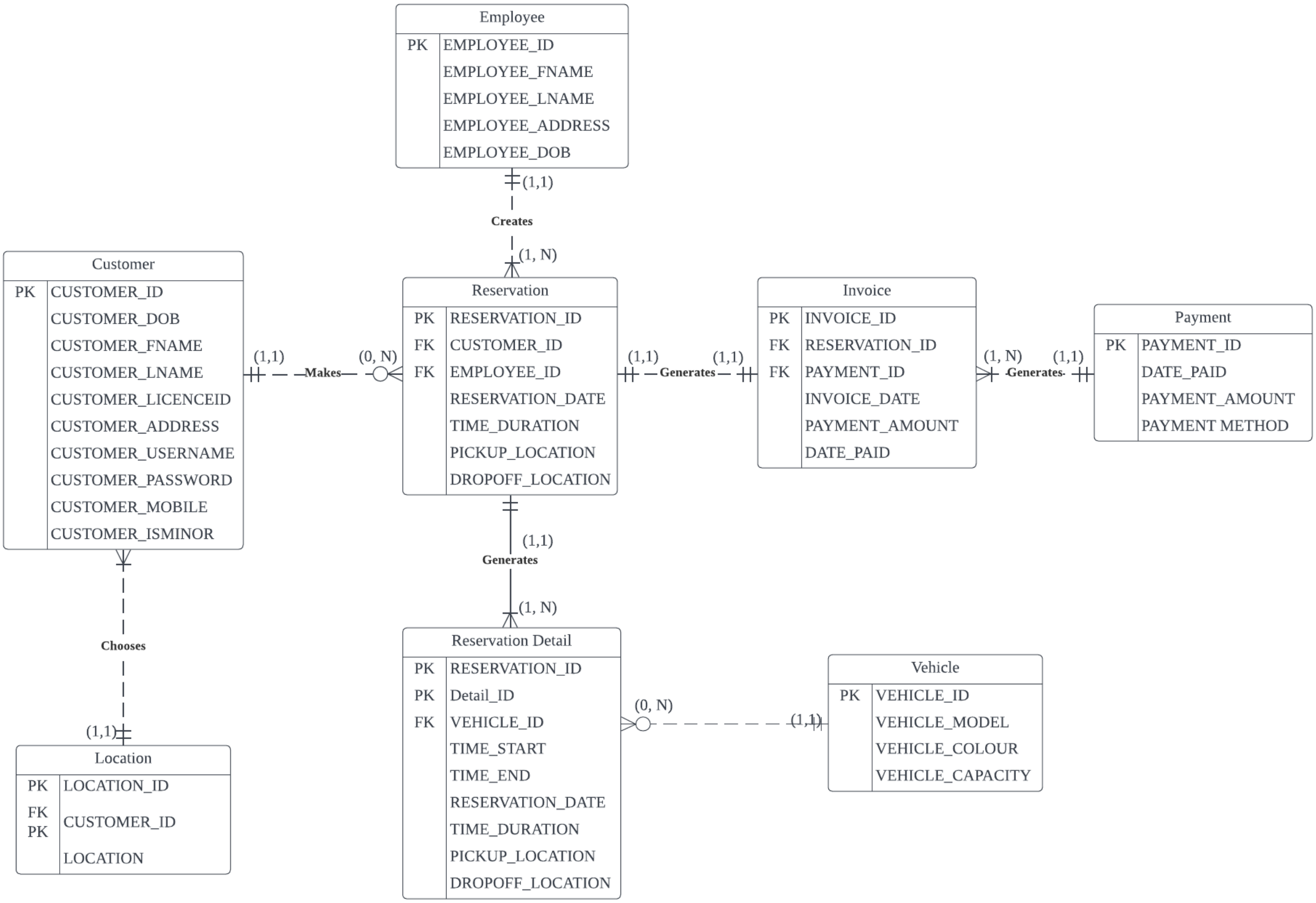
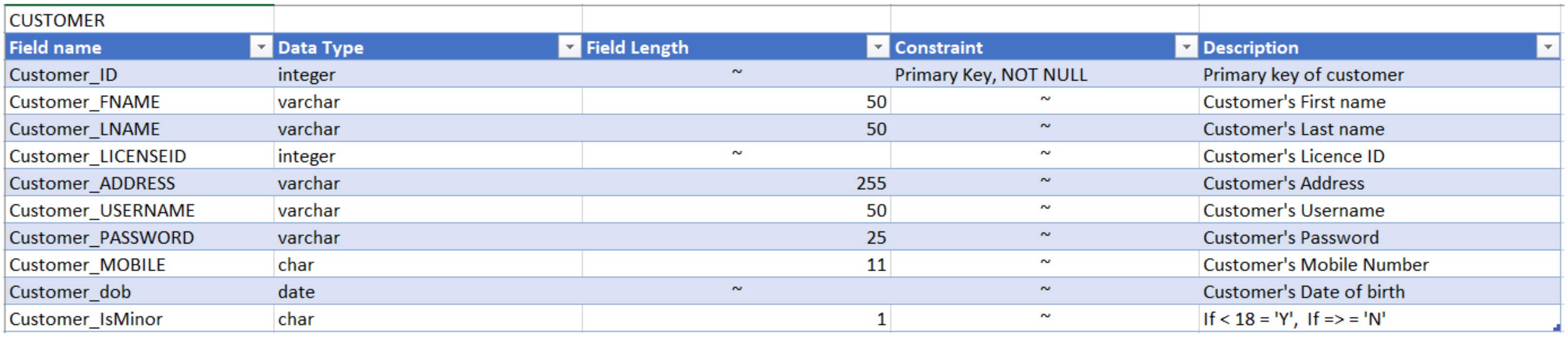
ERD DIAGRAM:


Employee
PK EMPLOYEE_ID
EMPLOYEE_FNAME
EMPLOYEE_LNAME
EMPLOYEE_ADDRESS
EMPLOYEE_DOB
(1,1)
Creates
(1, N)
Customer
PK CUSTOMER_ID
CUSTOMER_DOB
CUSTOMER_FNAME
CUSTOMER_LNAME
CUSTOMER_LICENCEID
CUSTOMER_ADDRESS
CUSTOMER_USERNAME
CUSTOMER_PASSWORD
CUSTOMER_MOBILE
CUSTOMER_ISMINOR
(1,1)
Makes
(0, N)
Reservation
PK RESERVATION_ID
FK CUSTOMER_ID
FK EMPLOYEE_ID
RESERVATION_DATE
TIME_DURATION
PICKUP_LOCATION
DROPOFF_LOCATION
(1,1)
Generates
(1,1)
Invoice
PK INVOICE_ID
FK RESERVATION_ID
FK PAYMENT_ID
INVOICE_DATE
PAYMENT_AMOUNT
DATE_PAID
(1, N)
Generates
(1,1)
Payment
PK PAYMENT_ID
DATE_PAID
PAYMENT_AMOUNT
PAYMENT METHOD
(1,1)
Generates
(1, N)
Chooses
(1,1)
Location
PK LOCATION_ID
FK
PK
CUSTOMER_ID
LOCATION
Reservation Detail
PK RESERVATION_ID
PK Detail_ID
FK VEHICLE_ID
TIME_START
TIME_END
RESERVATION_DATE
TIME_DURATION
PICKUP_LOCATION
DROPOFF_LOCATION
(0, N)
(1,1)
Vehicle
PK VEHICLE_ID
VEHICLE_MODEL
VEHICLE_COLOUR
VEHICLE_CAPACITY

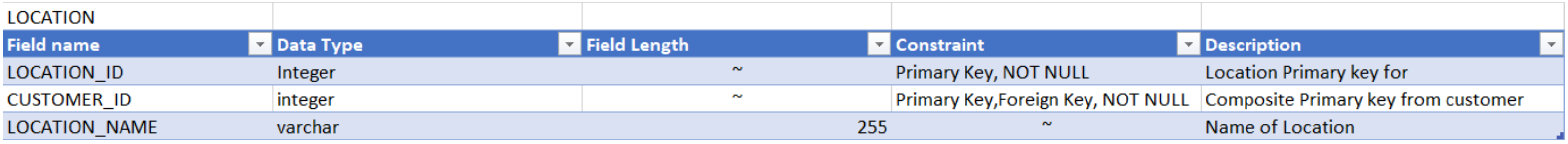
CUSTOMER:



CUSTOMER

Field name	Data Type	Field Length	Constraint	Description
Customer_ID	integer	~	Primary Key, NOT NULL	Primary key of customer
Customer_FNAME	varchar	50	~	Customer's First name
Customer_LNAME	varchar	50	~	Customer's Last name
Customer_LICENSEID	integer	~	~	Customer's Licence ID
Customer_ADDRESS	varchar	255	~	Customer's Address
Customer_USERNAME	varchar	50	~	Customer's Username
Customer_PASSWORD	varchar	25	~	Customer's Password
Customer_MOBILE	char	11	~	Customer's Mobile Number
Customer_dob	date	~	~	Customer's Date of birth
Customer_IsMinor	char	1	~	If < 18 = 'Y', If => = 'N'

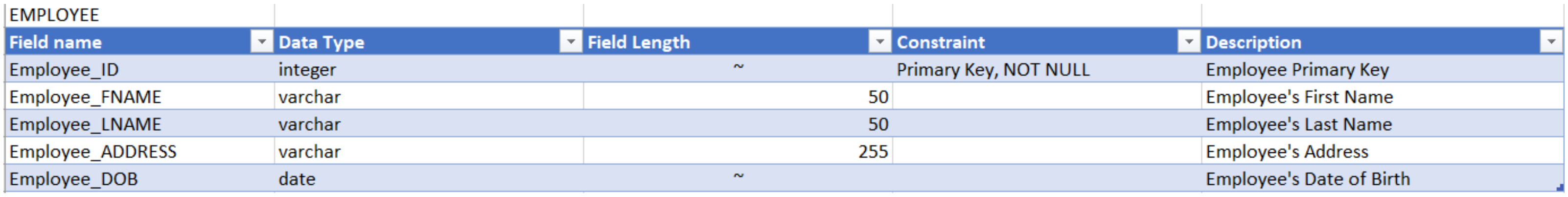
LOCATION:



LOCATION

Field name	Data Type	Field Length	Constraint	Description
LOCATION_ID	Integer	~	Primary Key, NOT NULL	Location Primary key for
CUSTOMER_ID	integer	~	Primary Key,Foreign Key, NOT NULL	Composite Primary key from customer
LOCATION_NAME	varchar	255	~	Name of Location

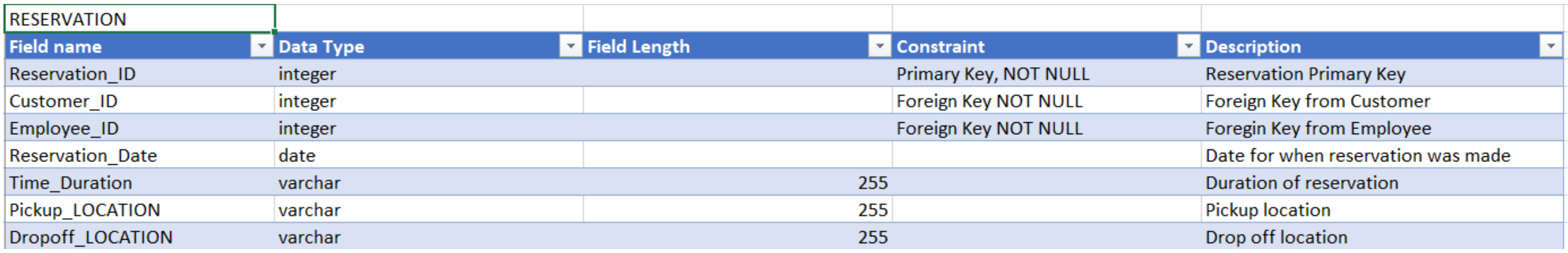
EMPLOYEE:



EMPLOYEE

Field name	Data Type	Field Length	Constraint	Description
Employee_ID	integer	~	Primary Key, NOT NULL	Employee Primary Key
Employee_FNAME	varchar	50		Employee's First Name
Employee_LNAME	varchar	50		Employee's Last Name
Employee_ADDRESS	varchar	255		Employee's Address
Employee_DOB	date	~		Employee's Date of Birth

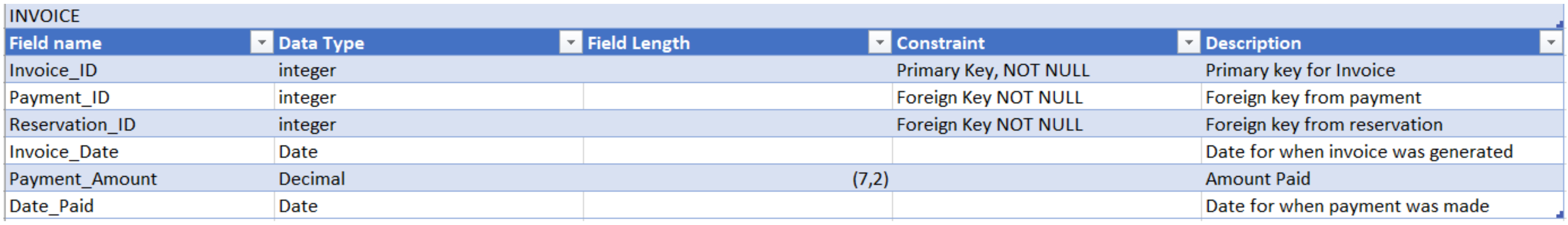
RESERVATION:



RESERVATION

Field name	Data Type	Field Length	Constraint	Description
Reservation_ID	integer		Primary Key, NOT NULL	Reservation Primary Key
Customer_ID	integer		Foreign Key NOT NULL	Foreign Key from Customer
Employee_ID	integer		Foreign Key NOT NULL	Foregin Key from Employee
Reservation_Date	date			Date for when reservation was made
Time_Duration	varchar	255		Duration of reservation
Pickup_LOCATION	varchar	255		Pickup location
Dropoff_LOCATION	varchar	255		Drop off location

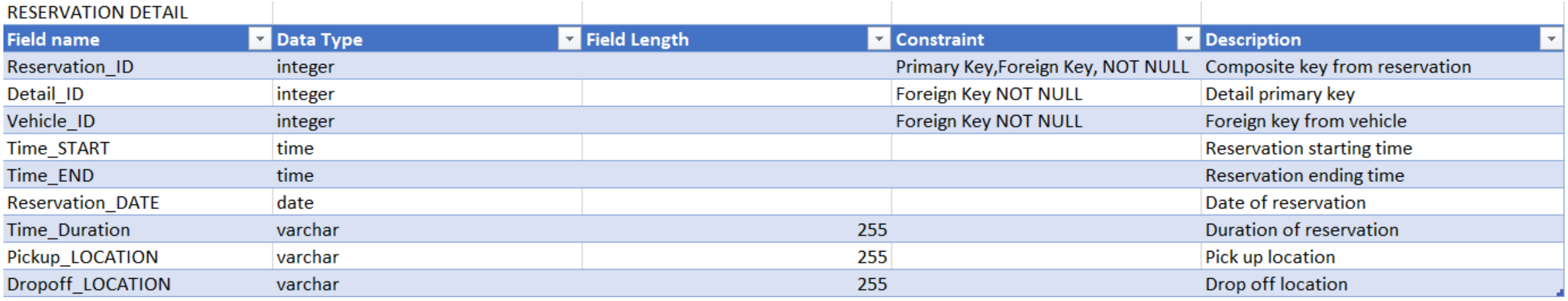
INVOICE:



INVOICE

Field name	Data Type	Field Length	Constraint	Description
Invoice_ID	integer		Primary Key, NOT NULL	Primary key for Invoice
Payment_ID	integer		Foreign Key NOT NULL	Foreign key from payment
Reservation_ID	integer		Foreign Key NOT NULL	Foreign key from reservation
Invoice_Date	Date			Date for when invoice was generated
Payment_Amount	Decimal	(7,2)		Amount Paid
Date_Paid	Date			Date for when payment was made

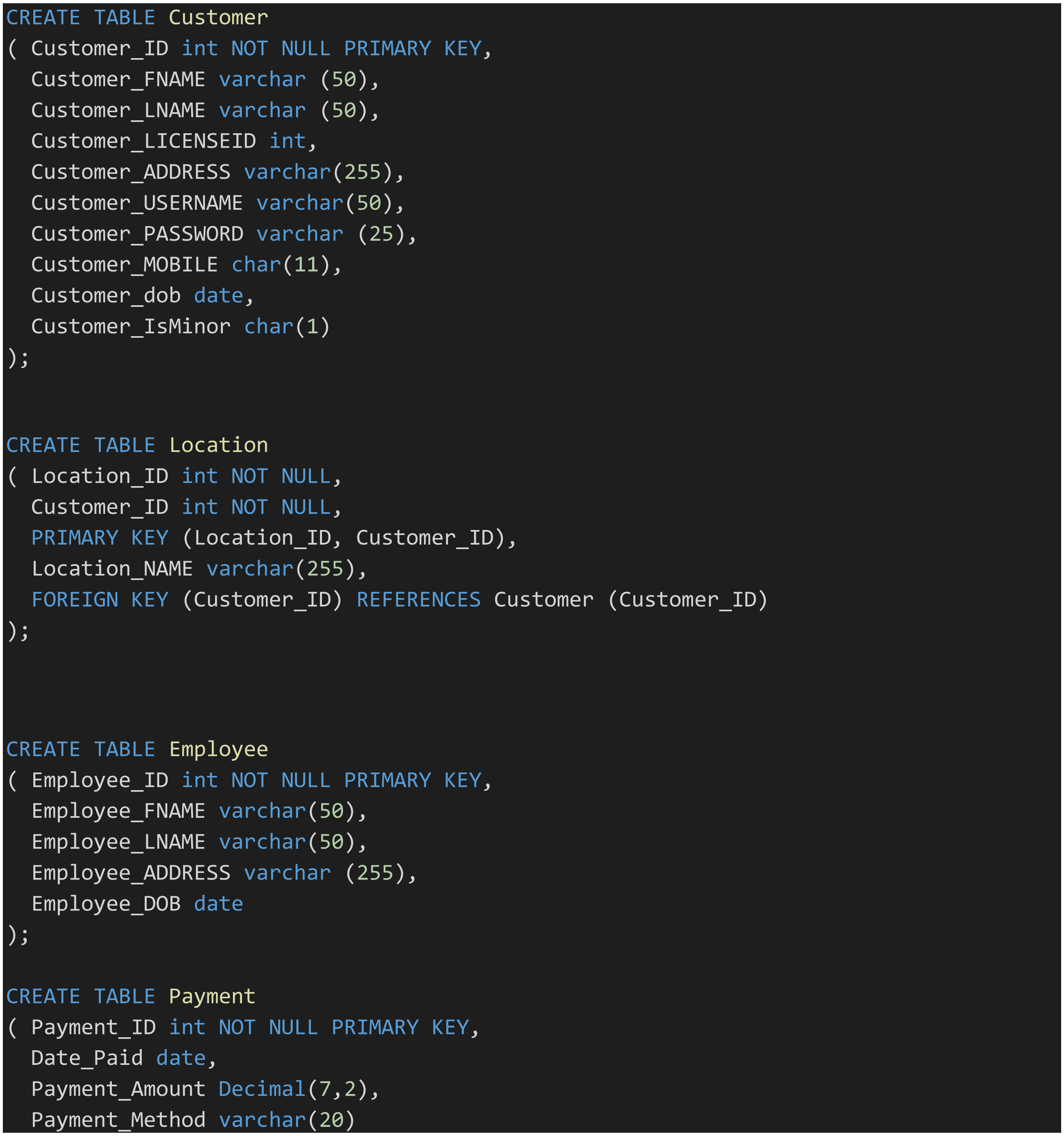
RESERVATION DETAIL:



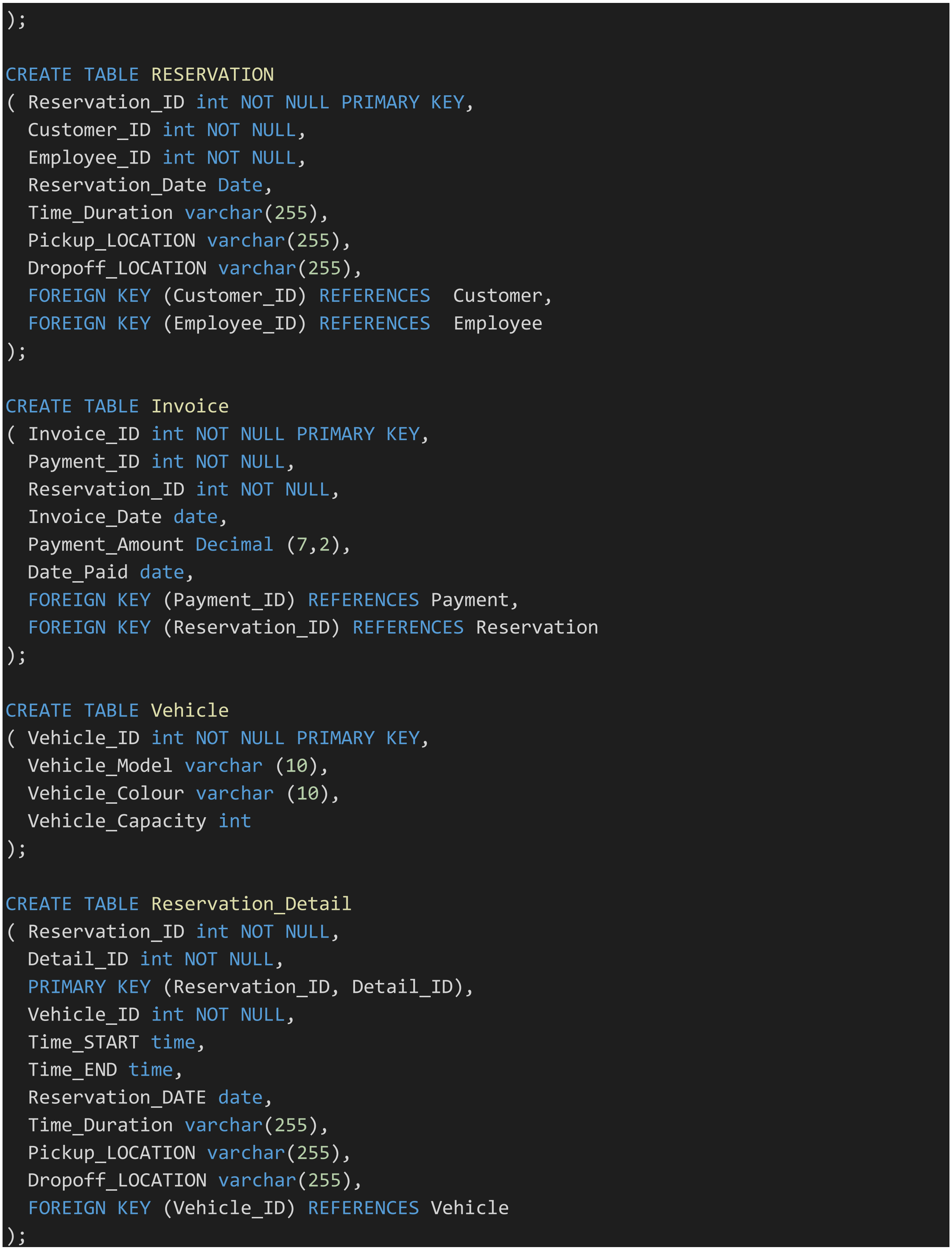
RESERVATION DETAIL

Field name	Data Type	Field Length	Constraint	Description
Reservation_ID	integer		Primary Key,Foreign Key, NOT NULL	Composite key from reservation
Detail_ID	integer		Foreign Key NOT NULL	Detail primary key
Vehicle_ID	integer		Foreign Key NOT NULL	Foreign key from vehicle
Time_START	time			Reservation starting time
Time_END	time			Reservation ending time
Reservation_DATE	date			Date of reservation
Time_Duration	varchar	255		Duration of reservation
Pickup_LOCATION	varchar	255		Pick up location
Dropoff_LOCATION	varchar	255		Drop off location

TABLE CREATION CODE:

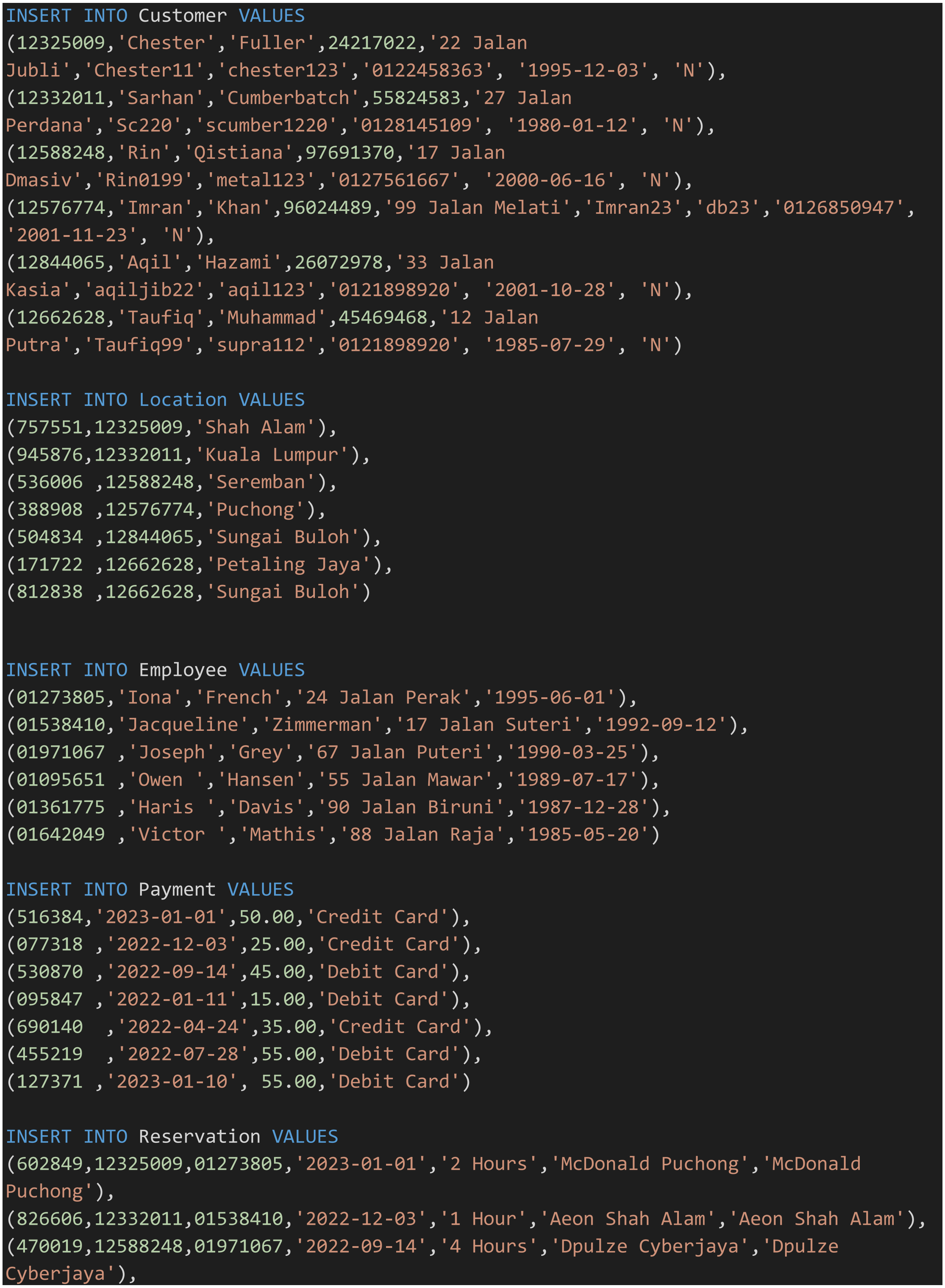


```
CREATE TABLE Customer
( Customer_ID int NOT NULL PRIMARY KEY,
  Customer_FNAME varchar (50),
  Customer_LNAME varchar (50),
  Customer_LICENSEID int,
  Customer_ADDRESS varchar(255),
  Customer_USERNAME varchar(50),
  Customer_PASSWORD varchar (25),
  Customer_MOBILE char(11),
  Customer_dob date,
  Customer_IsMinor char(1)
);


CREATE TABLE Location
( Location_ID int NOT NULL,
  Customer_ID int NOT NULL,
  PRIMARY KEY (Location_ID, Customer_ID),
  Location_NAME varchar(255),
  FOREIGN KEY (Customer_ID) REFERENCES Customer (Customer_ID)
);



CREATE TABLE Employee
( Employee_ID int NOT NULL PRIMARY KEY,
  Employee_FNAME varchar(50),
  Employee_LNAME varchar(50),
  Employee_ADDRESS varchar (255),
  Employee_DOB date
);

CREATE TABLE Payment
( Payment_ID int NOT NULL PRIMARY KEY,
  Date_Paid date,
  Payment_Amount Decimal(7,2),
  Payment_Method varchar(20)
```

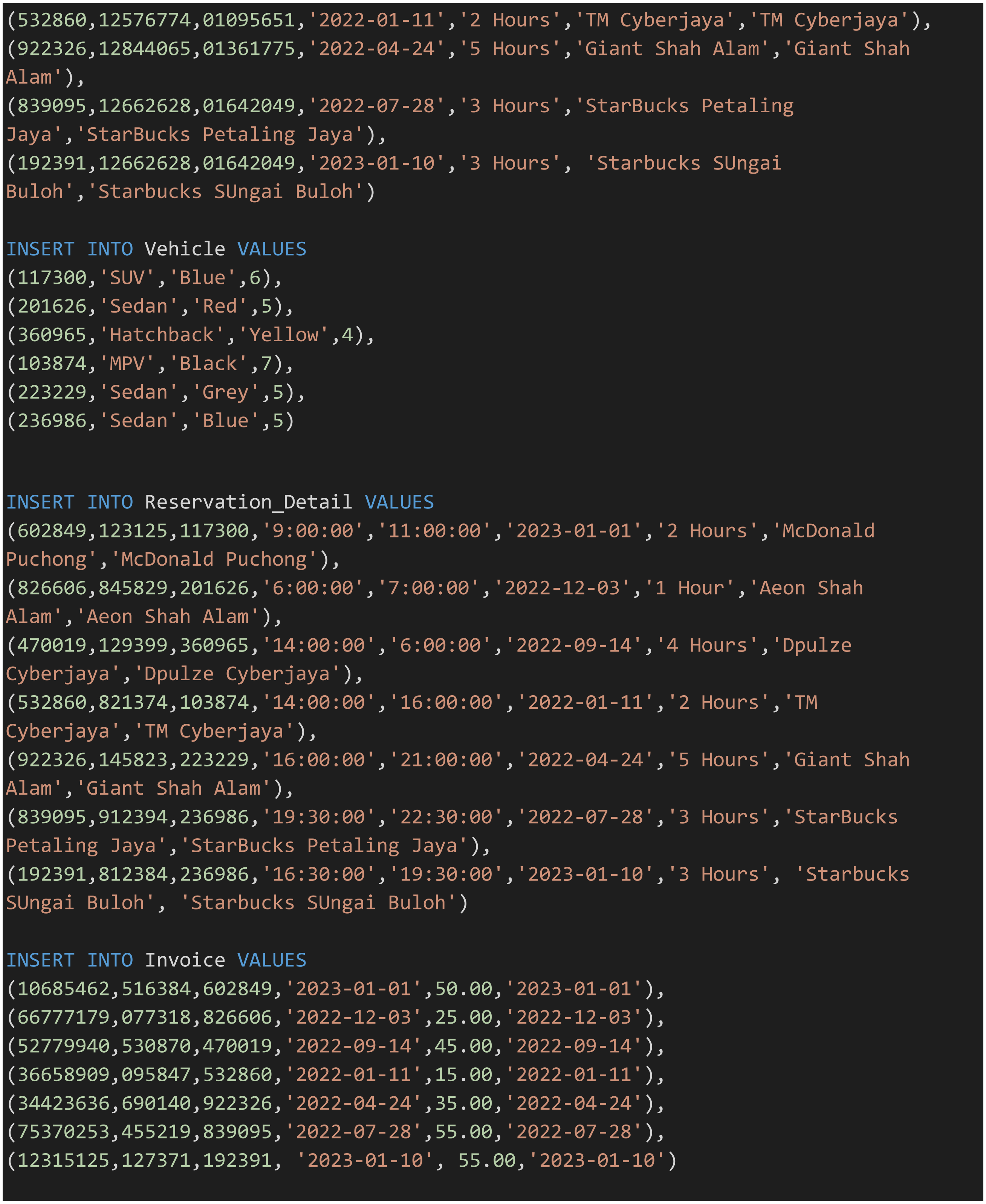


```sql
);

CREATE TABLE RESERVATION
( Reservation_ID int NOT NULL PRIMARY KEY,
  Customer_ID int NOT NULL,
  Employee_ID int NOT NULL,
  Reservation_Date Date,
  Time_Duration varchar(255),
  Pickup_LOCATION varchar(255),
  Dropoff_LOCATION varchar(255),
  FOREIGN KEY (Customer_ID) REFERENCES  Customer,
  FOREIGN KEY (Employee_ID) REFERENCES  Employee
);

CREATE TABLE Invoice
( Invoice_ID int NOT NULL PRIMARY KEY,
  Payment_ID int NOT NULL,
  Reservation_ID int NOT NULL,
  Invoice_Date date,
  Payment_Amount Decimal (7,2),
  Date_Paid date,
  FOREIGN KEY (Payment_ID) REFERENCES Payment,
  FOREIGN KEY (Reservation_ID) REFERENCES Reservation
);

CREATE TABLE Vehicle
( Vehicle_ID int NOT NULL PRIMARY KEY,
  Vehicle_Model varchar (10),
  Vehicle_Colour varchar (10),
  Vehicle_Capacity int
);

CREATE TABLE Reservation_Detail
( Reservation_ID int NOT NULL,
  Detail_ID int NOT NULL,
  PRIMARY KEY (Reservation_ID, Detail_ID),
  Vehicle_ID int NOT NULL,
  Time_START time,
  Time_END time,
  Reservation_DATE date,
  Time_Duration varchar(255),
  Pickup_LOCATION varchar(255),
  Dropoff_LOCATION varchar(255),
  FOREIGN KEY (Vehicle_ID) REFERENCES Vehicle
);
```

DATA INSERTION CODE:



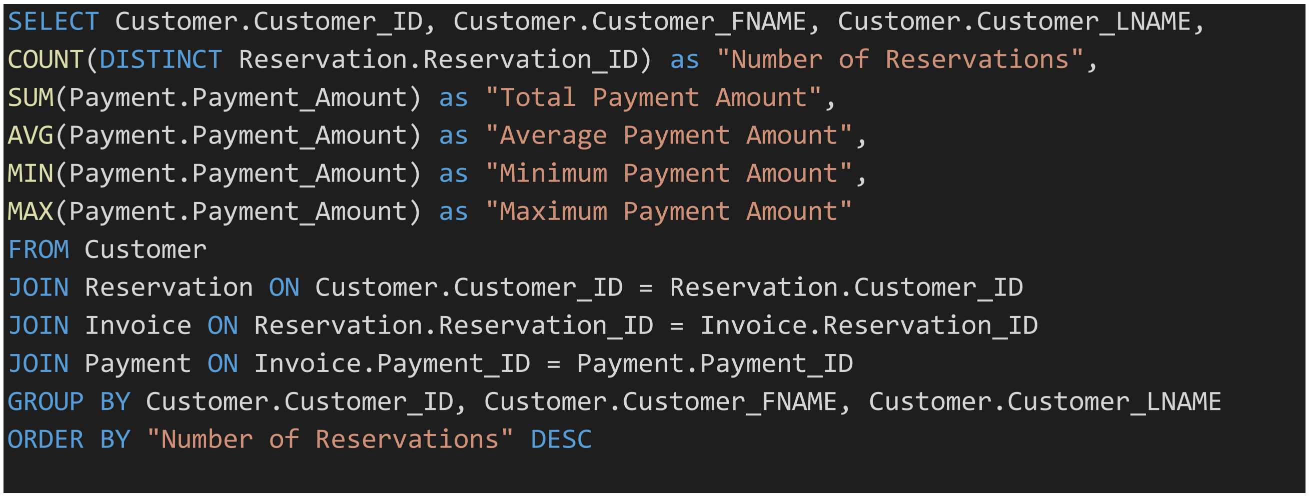
```
INSERT INTO Customer VALUES
(12325009,'Chester','Fuller',24217022,'22 Jalan
Jubli','Chester11','chester123','0122458363', '1995-12-03', 'N'),
(12332011,'Sarhan','Cumberbatch',55824583,'27 Jalan
Perdana','Sc220','scumber1220','0128145109', '1980-01-12', 'N'),
(12588248,'Rin','Qistiana',97691370,'17 Jalan
Dmasiv','Rin0199','metal123','0127561667', '2000-06-16', 'N'),
(12576774,'Imran','Khan',96024489,'99 Jalan Melati','Imran23','db23','0126850947',
'2001-11-23', 'N'),
(12844065,'Aqil','Hazami',26072978,'33 Jalan
Kasia','aqiljib22','aqil123','0121898920', '2001-10-28', 'N'),
(12662628,'Taufiq','Muhammad',45469468,'12 Jalan
Putra','Taufiq99','supra112','0121898920', '1985-07-29', 'N')

INSERT INTO Location VALUES
(757551,12325009,'Shah Alam'),
(945876,12332011,'Kuala Lumpur'),
(536006 ,12588248,'Seremban'),
(388908 ,12576774,'Puchong'),
(504834 ,12844065,'Sungai Buloh'),
(171722 ,12662628,'Petaling Jaya'),
(812838 ,12662628,'Sungai Buloh')


INSERT INTO Employee VALUES
(01273805,'Iona','French','24 Jalan Perak','1995-06-01'),
(01538410,'Jacqueline','Zimmerman','17 Jalan Suteri','1992-09-12'),
(01971067 ,'Joseph','Grey','67 Jalan Puteri','1990-03-25'),
(01095651 ,'Owen ','Hansen','55 Jalan Mawar','1989-07-17'),
(01361775 ,'Haris ','Davis','90 Jalan Biruni','1987-12-28'),
(01642049 ,'Victor ','Mathis','88 Jalan Raja','1985-05-20')

INSERT INTO Payment VALUES
(516384,'2023-01-01',50.00,'Credit Card'),
(077318 ,'2022-12-03',25.00,'Credit Card'),
(530870 ,'2022-09-14',45.00,'Debit Card'),
(095847 ,'2022-01-11',15.00,'Debit Card'),
(690140  ,'2022-04-24',35.00,'Credit Card'),
(455219  ,'2022-07-28',55.00,'Debit Card'),
(127371 ,'2023-01-10', 55.00,'Debit Card')

INSERT INTO Reservation VALUES
(602849,12325009,01273805,'2023-01-01','2 Hours','McDonald Puchong','McDonald
Puchong'),
(826606,12332011,01538410,'2022-12-03','1 Hour','Aeon Shah Alam','Aeon Shah Alam'),
(470019,12588248,01971067,'2022-09-14','4 Hours','Dpulze Cyberjaya','Dpulze
Cyberjaya'),
```



```
(532860,12576774,01095651,'2022-01-11','2 Hours','TM Cyberjaya','TM Cyberjaya'),
(922326,12844065,01361775,'2022-04-24','5 Hours','Giant Shah Alam','Giant Shah
Alam'),
(839095,12662628,01642049,'2022-07-28','3 Hours','StarBucks Petaling
Jaya','StarBucks Petaling Jaya'),
(192391,12662628,01642049,'2023-01-10','3 Hours', 'Starbucks SUngai
Buloh','Starbucks SUngai Buloh')

INSERT INTO Vehicle VALUES
(117300,'SUV','Blue',6),
(201626,'Sedan','Red',5),
(360965,'Hatchback','Yellow',4),
(103874,'MPV','Black',7),
(223229,'Sedan','Grey',5),
(236986,'Sedan','Blue',5)


INSERT INTO Reservation_Detail VALUES
(602849,123125,117300,'9:00:00','11:00:00','2023-01-01','2 Hours','McDonald
Puchong','McDonald Puchong'),
(826606,845829,201626,'6:00:00','7:00:00','2022-12-03','1 Hour','Aeon Shah
Alam','Aeon Shah Alam'),
(470019,129399,360965,'14:00:00','6:00:00','2022-09-14','4 Hours','Dpulze
Cyberjaya','Dpulze Cyberjaya'),
(532860,821374,103874,'14:00:00','16:00:00','2022-01-11','2 Hours','TM
Cyberjaya','TM Cyberjaya'),
(922326,145823,223229,'16:00:00','21:00:00','2022-04-24','5 Hours','Giant Shah
Alam','Giant Shah Alam'),
(839095,912394,236986,'19:30:00','22:30:00','2022-07-28','3 Hours','StarBucks
Petaling Jaya','StarBucks Petaling Jaya'),
(192391,812384,236986,'16:30:00','19:30:00','2023-01-10','3 Hours', 'Starbucks
SUngai Buloh', 'Starbucks SUngai Buloh')

INSERT INTO Invoice VALUES
(10685462,516384,602849,'2023-01-01',50.00,'2023-01-01'),
(66777179,077318,826606,'2022-12-03',25.00,'2022-12-03'),
(52779940,530870,470019,'2022-09-14',45.00,'2022-09-14'),
(36658909,095847,532860,'2022-01-11',15.00,'2022-01-11'),
(34423636,690140,922326,'2022-04-24',35.00,'2022-04-24'),
(75370253,455219,839095,'2022-07-28',55.00,'2022-07-28'),
(12315125,127371,192391, '2023-01-10', 55.00,'2023-01-10')
```

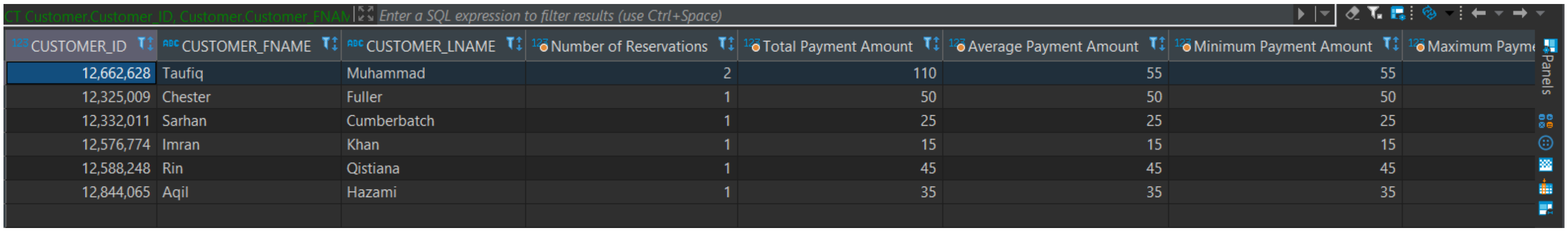
DATA MANIPULATION:

Aggregate Function:



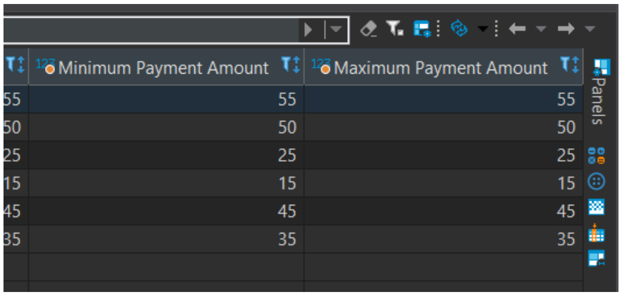
```
SELECT Customer.Customer_ID, Customer.Customer_FNAME, Customer.Customer_LNAME,
COUNT(DISTINCT Reservation.Reservation_ID) as "Number of Reservations",
SUM(Payment.Payment_Amount) as "Total Payment Amount",
AVG(Payment.Payment_Amount) as "Average Payment Amount",
MIN(Payment.Payment_Amount) as "Minimum Payment Amount",
MAX(Payment.Payment_Amount) as "Maximum Payment Amount"
FROM Customer
JOIN Reservation ON Customer.Customer_ID = Reservation.Customer_ID
JOIN Invoice ON Reservation.Reservation_ID = Invoice.Reservation_ID
JOIN Payment ON Invoice.Payment_ID = Payment.Payment_ID
GROUP BY Customer.Customer_ID, Customer.Customer_FNAME, Customer.Customer_LNAME
ORDER BY "Number of Reservations" DESC
```

Output:



CT Customer.Customer_ID, Customer.Customer_FNAM | Enter a SQL expression to filter results (use Ctrl+Space)

CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	Number of Reservations	Total Payment Amount	Average Payment Amount	Minimum Payment Amount	Maximum Payme
12,662,628	Taufiq	Muhammad	2	110	55	55	
12,325,009	Chester	Fuller	1	50	50	50	
12,332,011	Sarhan	Cumberbatch	1	25	25	25	
12,576,774	Imran	Khan	1	15	15	15	
12,588,248	Rin	Qistiana	1	45	45	45	
12,844,065	Aqil	Hazami	1	35	35	35	

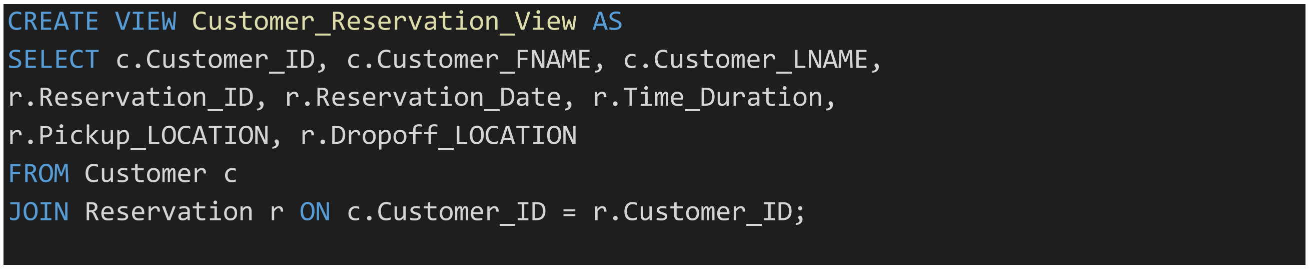


	Minimum Payment Amount	Maximum Payment Amount
55	55	55
50	50	50
25	25	25
15	15	15
45	45	45
35	35	35

Purpose:

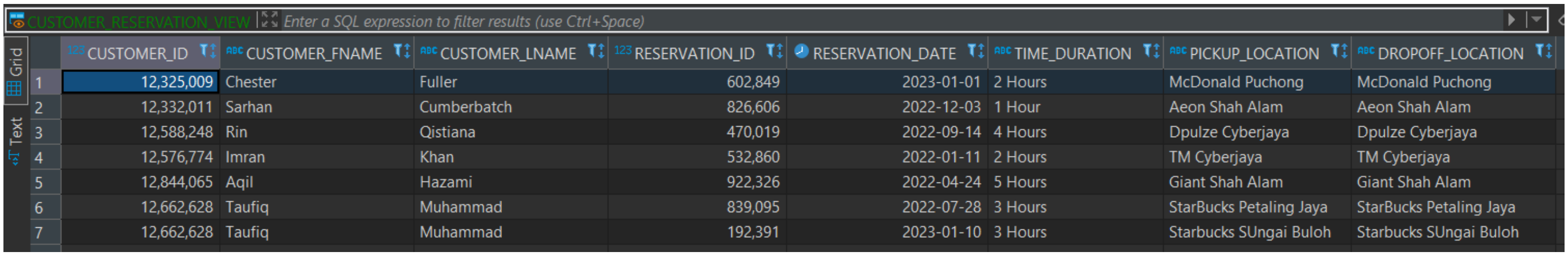
This query retrieves customer reservation statistics, including the number of reservations, total payment amount, average payment amount, minimum and maximum payment amount, and orders the result by number of reservations in descending order.

View:



```
CREATE VIEW Customer_Reservation_View AS
SELECT c.Customer_ID, c.Customer_FNAME, c.Customer_LNAME,
r.Reservation_ID, r.Reservation_Date, r.Time_Duration,
r.Pickup_LOCATION, r.Dropoff_LOCATION
FROM Customer c
JOIN Reservation r ON c.Customer_ID = r.Customer_ID;
```

Output:



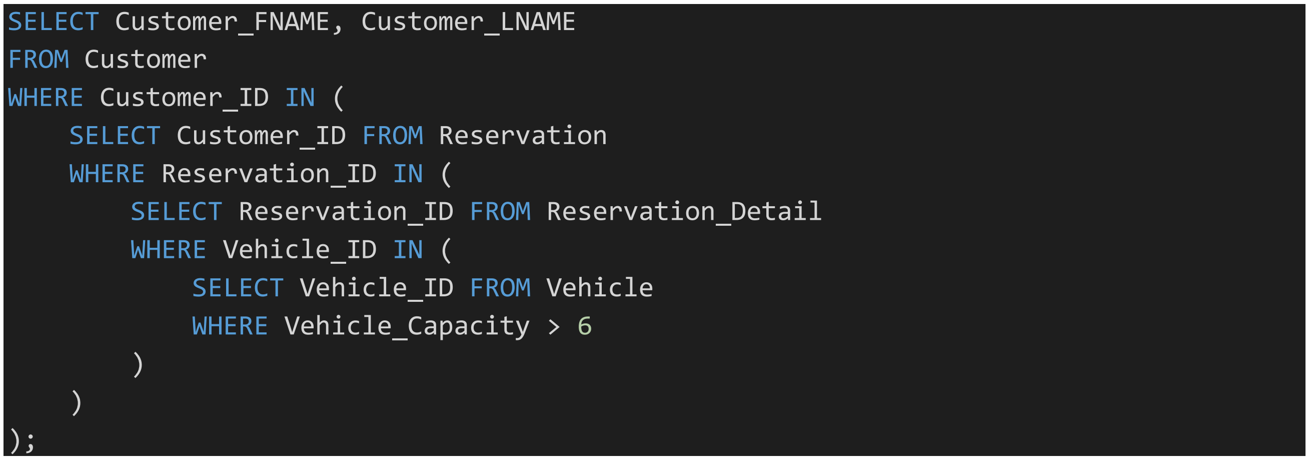
CUSTOMER_RESERVATION_VIEW
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	RESERVATION_ID	RESERVATION_DATE	TIME_DURATION	PICKUP_LOCATION	DROPOFF_LOCATION
1	12,325,009	Chester	Fuller	602,849	2023-01-01	2 Hours	McDonald Puchong	McDonald Puchong
2	12,332,011	Sarhan	Cumberbatch	826,606	2022-12-03	1 Hour	Aeon Shah Alam	Aeon Shah Alam
3	12,588,248	Rin	Qistiana	470,019	2022-09-14	4 Hours	Dpulze Cyberjaya	Dpulze Cyberjaya
4	12,576,774	Imran	Khan	532,860	2022-01-11	2 Hours	TM Cyberjaya	TM Cyberjaya
5	12,844,065	Aqil	Hazami	922,326	2022-04-24	5 Hours	Giant Shah Alam	Giant Shah Alam
6	12,662,628	Taufiq	Muhammad	839,095	2022-07-28	3 Hours	StarBucks Petaling Jaya	StarBucks Petaling Jaya
7	12,662,628	Taufiq	Muhammad	192,391	2023-01-10	3 Hours	Starbucks SUngai Buloh	Starbucks SUngai Buloh

Purpose:

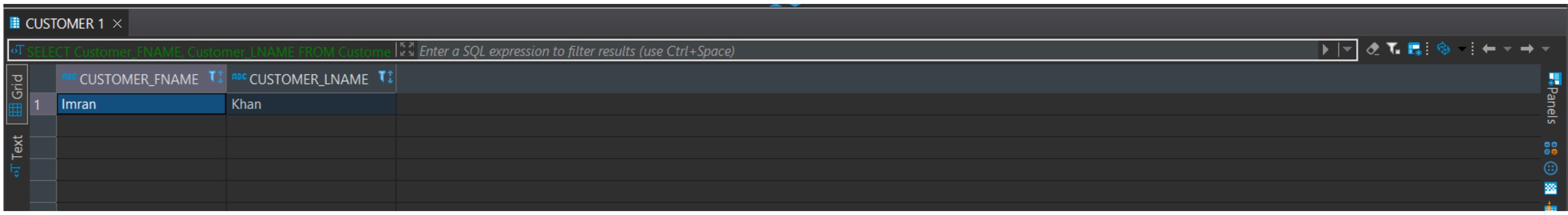
This function creates a view that displays the customer ID, first name, last name, reservation ID, reservation date, time duration, pickup location, and drop off location for all customers who have made a reservation. The view is created by joining the Customer table with the Reservation table on the customer ID.

Nested Query:



```
SELECT Customer_FNAME, Customer_LNAME
FROM Customer
WHERE Customer_ID IN (
    SELECT Customer_ID FROM Reservation
    WHERE Reservation_ID IN (
        SELECT Reservation_ID FROM Reservation_Detail
        WHERE Vehicle_ID IN (
            SELECT Vehicle_ID FROM Vehicle
            WHERE Vehicle_Capacity > 6
        )
    )
);
```

Output:

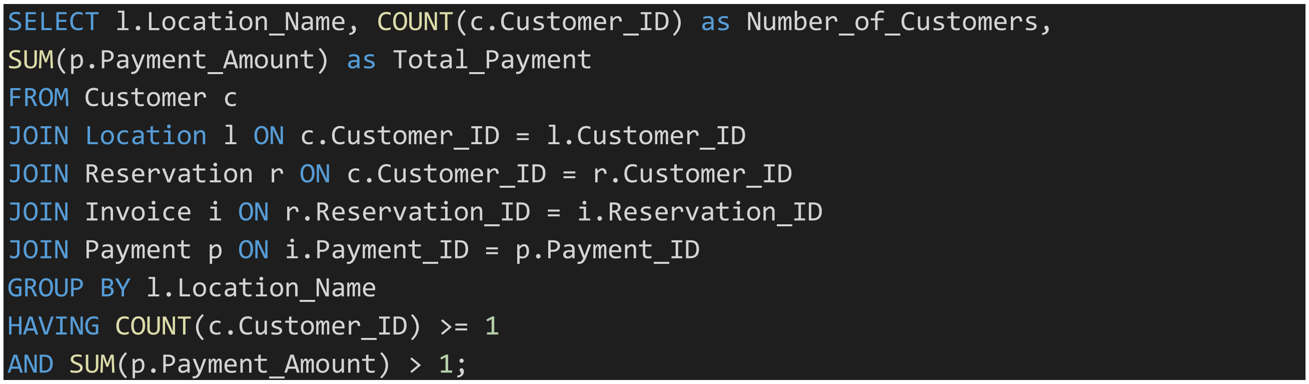


CUSTOMER 1
SELECT Customer_FNAME, Customer_LNAME FROM Custome
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Panels

	CUSTOMER_FNAME	CUSTOMER_LNAME
1	Imran	Khan

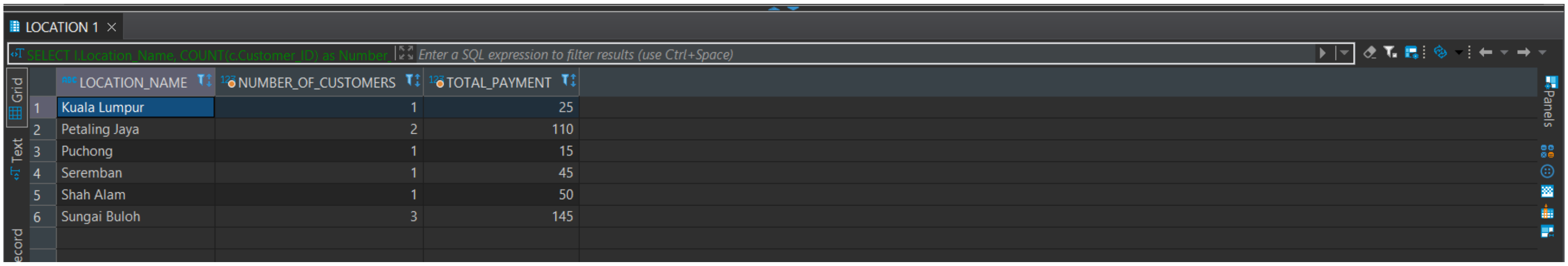
Purpose:
This query selects the first name and last name of customers who have made a reservation for a vehicle that has a capacity greater than 6. The query does this by first selecting the customer ID of customers who have made a reservation by using a subquery that joins the Reservation table with the Reservation Detail table. Then, it uses another subquery to select the vehicle ID of the vehicles that have a capacity greater than 6 by joining the Vehicle table with the Reservation Detail table.

Group by & Having:



```
SELECT l.Location_Name, COUNT(c.Customer_ID) as Number_of_Customers,
SUM(p.Payment_Amount) as Total_Payment
FROM Customer c
JOIN Location l ON c.Customer_ID = l.Customer_ID
JOIN Reservation r ON c.Customer_ID = r.Customer_ID
JOIN Invoice i ON r.Reservation_ID = i.Reservation_ID
JOIN Payment p ON i.Payment_ID = p.Payment_ID
GROUP BY l.Location_Name
HAVING COUNT(c.Customer_ID) >= 1
AND SUM(p.Payment_Amount) > 1;
```

Output:



LOCATION 1
SELECT l.Location_Name, COUNT(c.Customer_ID) as Number_
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
ecord
Panels

	LOCATION_NAME	NUMBER_OF_CUSTOMERS	TOTAL_PAYMENT
1	Kuala Lumpur	1	25
2	Petaling Jaya	2	110
3	Puchong	1	15
4	Seremban	1	45
5	Shah Alam	1	50
6	Sungai Buloh	3	145

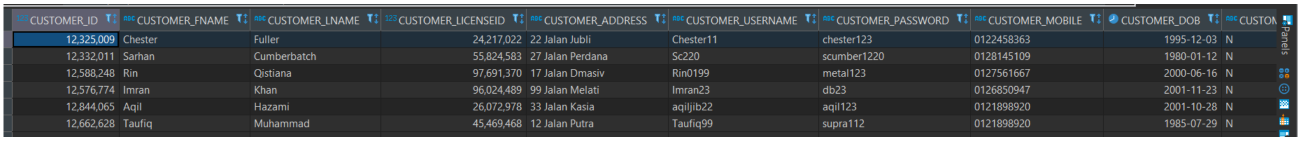
Purpose:
The purpose for this query is to view the total number of reservations made based on the location name. It also outputs the total payment for each location.

Triggers:

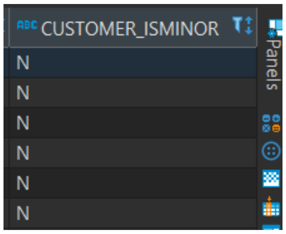
```
CREATE TRIGGER trg_customer12223
AFTER INSERT ON Customer
REFERENCING NEW AS new_row
FOR EACH ROW MODE DB2SQL
BEGIN
    IF new_row.CUSTOMER_DOB < '2000-10-10' THEN
        UPDATE Customer SET CUSTOMER_ISMINOR ='N' where CUSTOMER_ID =
new_row.CUSTOMER_ID;
    ELSE
        UPDATE Customer SET CUSTOMER_ISMINOR ='Y' where CUSTOMER_ID =
new_row.CUSTOMER_ID;
    END IF;
END
```

Output:

Before:



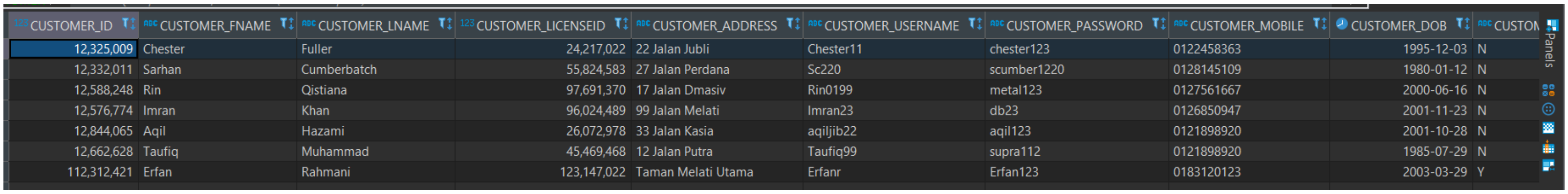
CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_LICENSEID	CUSTOMER_ADDRESS	CUSTOMER_USERNAME	CUSTOMER_PASSWORD	CUSTOMER_MOBILE	CUSTOMER_DOB	CUSTON
12,325,009	Chester	Fuller	24,217,022	22 Jalan Jubli	Chester11	chester123	0122458363	1995-12-03	N
12,332,011	Sarhan	Cumberbatch	55,824,583	27 Jalan Perdana	Sc220	scumber1220	0128145109	1980-01-12	N
12,588,248	Rin	Qistiana	97,691,370	17 Jalan Dmasiv	Rin0199	metal123	0127561667	2000-06-16	N
12,576,774	Imran	Khan	96,024,489	99 Jalan Melati	Imran23	db23	0126850947	2001-11-23	N
12,844,065	Aqil	Hazami	26,072,978	33 Jalan Kasia	aqiljib22	aqil123	0121898920	2001-10-28	N
12,662,628	Taufiq	Muhammad	45,469,468	12 Jalan Putra	Taufiq99	supra112	0121898920	1985-07-29	N



CUSTOMER_ISMINOR
N
N
N
N
N
N

After stored procedure is used to add a customer:

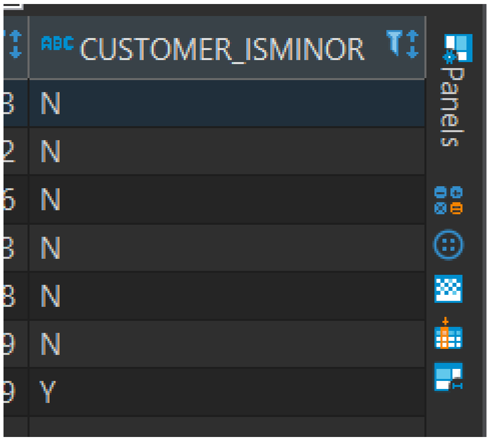
```
CALL insert_customer(112312421,'Erfan','Rahmani',123147022,'Taman Melati
Utama','Erfanr','Erfan123','0183120123', '2003-03-29', 'Y');
```



CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_LICENSEID	CUSTOMER_ADDRESS	CUSTOMER_USERNAME	CUSTOMER_PASSWORD	CUSTOMER_MOBILE	CUSTOMER_DOB	CUSTON
12,325,009	Chester	Fuller	24,217,022	22 Jalan Jubli	Chester11	chester123	0122458363	1995-12-03	N
12,332,011	Sarhan	Cumberbatch	55,824,583	27 Jalan Perdana	Sc220	scumber1220	0128145109	1980-01-12	N
12,588,248	Rin	Qistiana	97,691,370	17 Jalan Dmasiv	Rin0199	metal123	0127561667	2000-06-16	N
12,576,774	Imran	Khan	96,024,489	99 Jalan Melati	Imran23	db23	0126850947	2001-11-23	N
12,844,065	Aqil	Hazami	26,072,978	33 Jalan Kasia	aqiljib22	aqil123	0121898920	2001-10-28	N
12,662,628	Taufiq	Muhammad	45,469,468	12 Jalan Putra	Taufiq99	supra112	0121898920	1985-07-29	N
112,312,421	Erfan	Rahmani	123,147,022	Taman Melati Utama	Erfanr	Erfan123	0183120123	2003-03-29	Y

Note: The table is too long so included another screenshot for the part that were cutout.

Purpose:



	CUSTOMER_ISMINOR
3	N
2	N
6	N
3	N
8	N
9	N
9	Y

The purpose for this trigger is to automatically determine whether a customer is underage or not.

Stored Procedure:

```
CREATE PROCEDURE insert_customer
(
  IN p_Customer_ID INTEGER,
  IN p_Customer_FNAME VARCHAR(50),
  IN p_Customer_LNAME VARCHAR(50),
  IN p_Customer_LICENSEID INTEGER,
  IN p_Customer_ADDRESS VARCHAR(255),
  IN p_Customer_USERNAME VARCHAR(50),
  IN p_Customer_PASSWORD VARCHAR (25),
  IN p_Customer_MOBILE CHAR(11),
  IN p_Customer_dob DATE,
  IN p_Customer_IsMinor CHAR(1)
)
BEGIN
  INSERT INTO Customer (Customer_ID, Customer_FNAME, Customer_LNAME,
Customer_LICENSEID,
  Customer_ADDRESS, Customer_USERNAME, Customer_PASSWORD, Customer_MOBILE,
Customer_dob, Customer_IsMinor)
  VALUES (p_Customer_ID, p_Customer_FNAME, p_Customer_LNAME, p_Customer_LICENSEID,
 p_Customer_ADDRESS, p_Customer_USERNAME, p_Customer_PASSWORD, p_Customer_MOBILE,
p_Customer_dob, p_Customer_IsMinor);
END
```

To use the stored procedure:

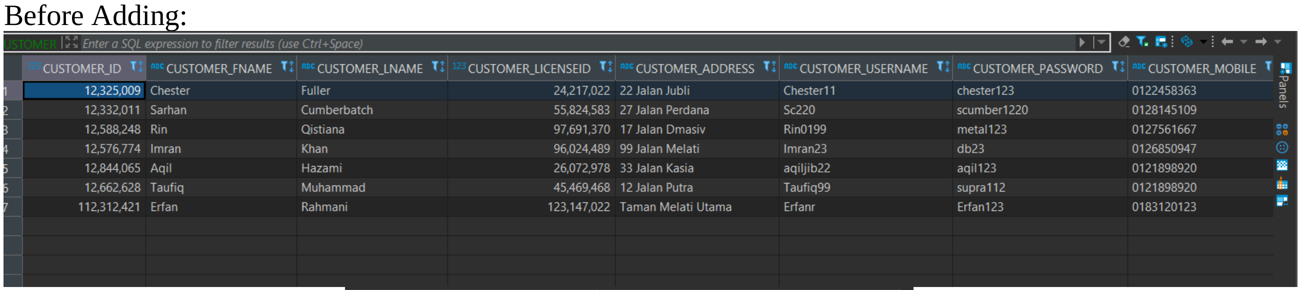
```
CALL insert_customer(11212421,'shubo','faisal',123147022,'12
Ampang','erafaniscool','iloveerfan','821381239', '2001-12-08', 'Y');
```

Purpose:

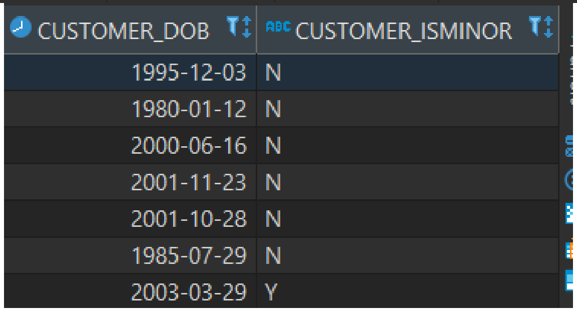
Add customers to the database without extra lines of code.

Outputs:

Before Adding:

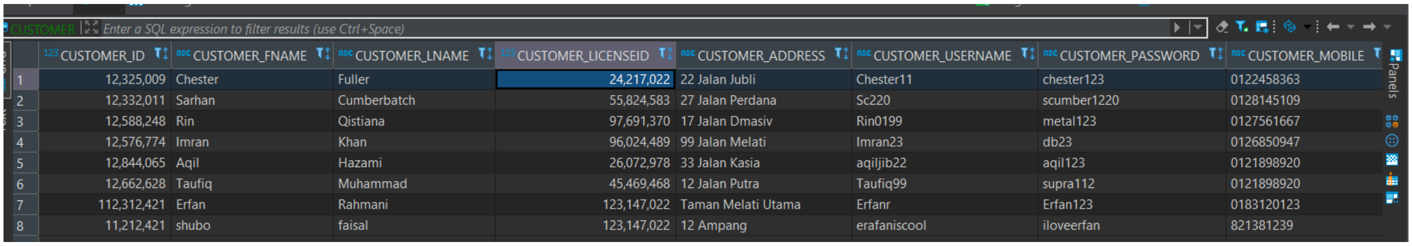


	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_LICENSEID	CUSTOMER_ADDRESS	CUSTOMER_USERNAME	CUSTOMER_PASSWORD	CUSTOMER_MOBILE
1	12,325,009	Chester	Fuller	24,217,022	22 Jalan Jubli	Chester11	chester123	0122458363
2	12,332,011	Sarhan	Cumberbatch	55,824,583	27 Jalan Perdana	Sc220	scumber1220	0128145109
3	12,588,248	Rin	Qistiana	97,691,370	17 Jalan Dmasiv	Rin0199	metal123	0127561667
4	12,576,774	Imran	Khan	96,024,489	99 Jalan Melati	Imran23	db23	0126850947
5	12,844,065	Aqil	Hazami	26,072,978	33 Jalan Kasia	aqiljib22	aqil123	0121898920
6	12,662,628	Taufiq	Muhammad	45,469,468	12 Jalan Putra	Taufiq99	supra112	0121898920
7	112,312,421	Erfan	Rahmani	123,147,022	Taman Melati Utama	Erfanr	Erfan123	0183120123

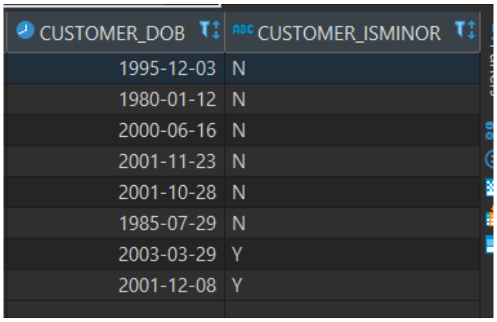


CUSTOMER_DOB	CUSTOMER_ISMINOR
1995-12-03	N
1980-01-12	N
2000-06-16	N
2001-11-23	N
2001-10-28	N
1985-07-29	N
2003-03-29	Y

After adding:



	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_LICENSEID	CUSTOMER_ADDRESS	CUSTOMER_USERNAME	CUSTOMER_PASSWORD	CUSTOMER_MOBILE
1	12,325,009	Chester	Fuller	24,217,022	22 Jalan Jubli	Chester11	chester123	0122458363
2	12,332,011	Sarhan	Cumberbatch	55,824,583	27 Jalan Perdana	Sc220	scumber1220	0128145109
3	12,588,248	Rin	Qistiana	97,691,370	17 Jalan Dmasiv	Rin0199	metal123	0127561667
4	12,576,774	Imran	Khan	96,024,489	99 Jalan Melati	Imran23	db23	0126850947
5	12,844,065	Aqil	Hazami	26,072,978	33 Jalan Kasia	aqiljib22	aqil123	0121898920
6	12,662,628	Taufiq	Muhammad	45,469,468	12 Jalan Putra	Taufiq99	supra112	0121898920
7	112,312,421	Erfan	Rahmani	123,147,022	Taman Melati Utama	Erfanr	Erfan123	0183120123
8	11,212,421	shubo	faisal	123,147,022	12 Ampang	erafaniscool	iloveerfan	821381239



CUSTOMER_DOB	CUSTOMER_ISMINOR
1995-12-03	N
1980-01-12	N
2000-06-16	N
2001-11-23	N
2001-10-28	N
1985-07-29	N
2003-03-29	Y
2001-12-08	Y

Code used:

```
CALL insert_customer(112312421,'shubo','faisal',123147022,'12
Ampang','erafaniscool','iloveerfan','821381239', '2001-12-08', 'Y');
```

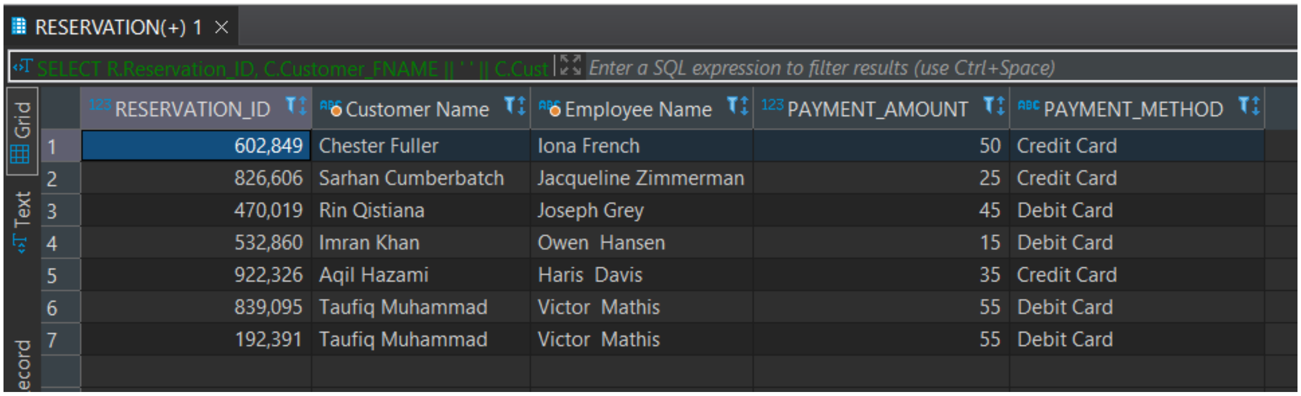
Note: the last part of the call (Customer_Isminor) is automatically adjusted no matter what is put in the call function. This is due to the trigger.

Four queries not included in lecture:

Concatenating strings:

```
SELECT R.Reservation_ID, C.Customer_FNAME || ' ' || C.Customer_LNAME AS "Customer
Name", E.Employee_FNAME || ' ' || E.Employee_LNAME AS "Employee Name",
P.Payment_Amount, P.Payment_Method
FROM Customer C
JOIN Reservation R ON C.Customer_ID = R.Customer_ID
JOIN Employee E ON R.Employee_ID = E.Employee_ID
JOIN Invoice I ON R.Reservation_ID = I.Reservation_ID
JOIN Payment P ON I.Payment_ID = P.Payment_ID;
```

Output:


RESERVATION(+) 1
SELECT R.Reservation_ID, C.Customer_FNAME || ' ' || C.Cust
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
ecord

	RESERVATION_ID	Customer Name	Employee Name	PAYMENT_AMOUNT	PAYMENT_METHOD
1	602,849	Chester Fuller	Iona French	50	Credit Card
2	826,606	Sarhan Cumberbatch	Jacqueline Zimmerman	25	Credit Card
3	470,019	Rin Qistiana	Joseph Grey	45	Debit Card
4	532,860	Imran Khan	Owen Hansen	15	Debit Card
5	922,326	Aqil Hazami	Haris Davis	35	Credit Card
6	839,095	Taufiq Muhammad	Victor Mathis	55	Debit Card
7	192,391	Taufiq Muhammad	Victor Mathis	55	Debit Card

Purpose:

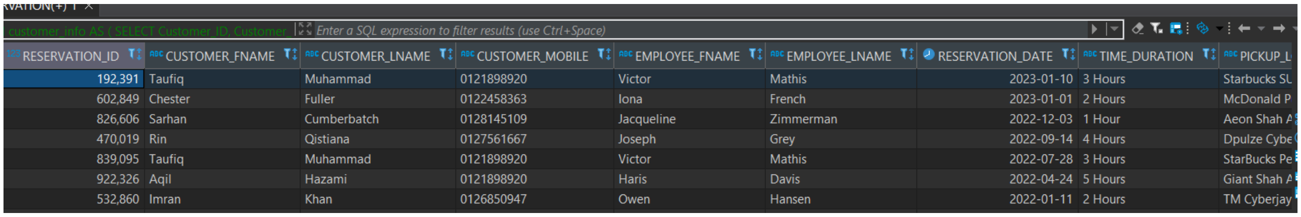
This query concatenates both the customer and employee’s first and last name. The employees in the table are in charge of the reservation the customers have made. The payment amount and payment methods are also displayed.

This query allows us to see a table with the customer’s full names. This is extremely useful because first names are not very uncommon.

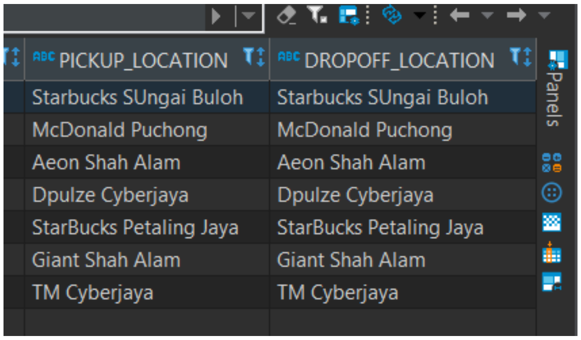
USE OF CTE (COMMON TABLE EXPRESSION):

```
WITH customer_info AS (
SELECT Customer_ID, Customer_FNAME, Customer_LNAME, Customer_MOBILE
FROM Customer
), employee_info AS (
SELECT Employee_ID, Employee_FNAME, Employee_LNAME
FROM Employee
)
SELECT Reservation_ID, ci.Customer_FNAME, ci.Customer_LNAME, ci.Customer_MOBILE,
ei.Employee_FNAME, ei.Employee_LNAME, Reservation_Date, Time_Duration,
Pickup_LOCATION, Dropoff_LOCATION
FROM Reservation
JOIN customer_info ci ON Reservation.Customer_ID = ci.Customer_ID
JOIN employee_info ei ON Reservation.Employee_ID = ei.Employee_ID
ORDER BY Reservation_Date DESC;
```

Output:


RVATION(+) 1
customer_info AS (SELECT Customer_ID, Customer_
Enter a SQL expression to filter results (use Ctrl+Space)

RESERVATION_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_MOBILE	EMPLOYEE_FNAME	EMPLOYEE_LNAME	RESERVATION_DATE	TIME_DURATION	PICKUP_L
192,391	Taufiq	Muhammad	0121898920	Victor	Mathis	2023-01-10	3 Hours	Starbucks SU
602,849	Chester	Fuller	0122458363	Iona	French	2023-01-01	2 Hours	McDonald P
826,606	Sarhan	Cumberbatch	0128145109	Jacqueline	Zimmerman	2022-12-03	1 Hour	Aeon Shah A
470,019	Rin	Qistiana	0127561667	Joseph	Grey	2022-09-14	4 Hours	Dpulze Cybe
839,095	Taufiq	Muhammad	0121898920	Victor	Mathis	2022-07-28	3 Hours	StarBucks Pe
922,326	Aqil	Hazami	0121898920	Haris	Davis	2022-04-24	5 Hours	Giant Shah A
532,860	Imran	Khan	0126850947	Owen	Hansen	2022-01-11	2 Hours	TM Cyberjay

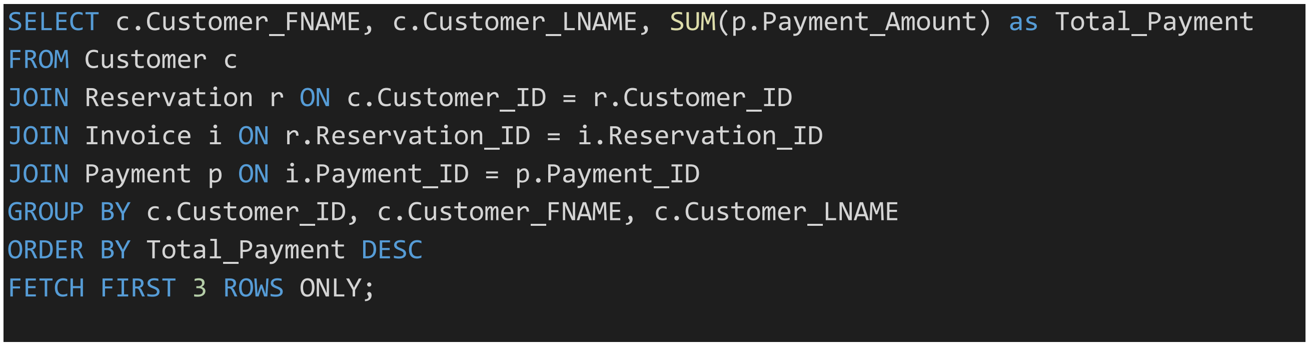


PICKUP_LOCATION	DROPOFF_LOCATION
Starbucks SUngai Buloh	Starbucks SUngai Buloh
McDonald Puchong	McDonald Puchong
Aeon Shah Alam	Aeon Shah Alam
Dpulze Cyberjaya	Dpulze Cyberjaya
StarBucks Petaling Jaya	StarBucks Petaling Jaya
Giant Shah Alam	Giant Shah Alam
TM Cyberjaya	TM Cyberjaya

Purpose:

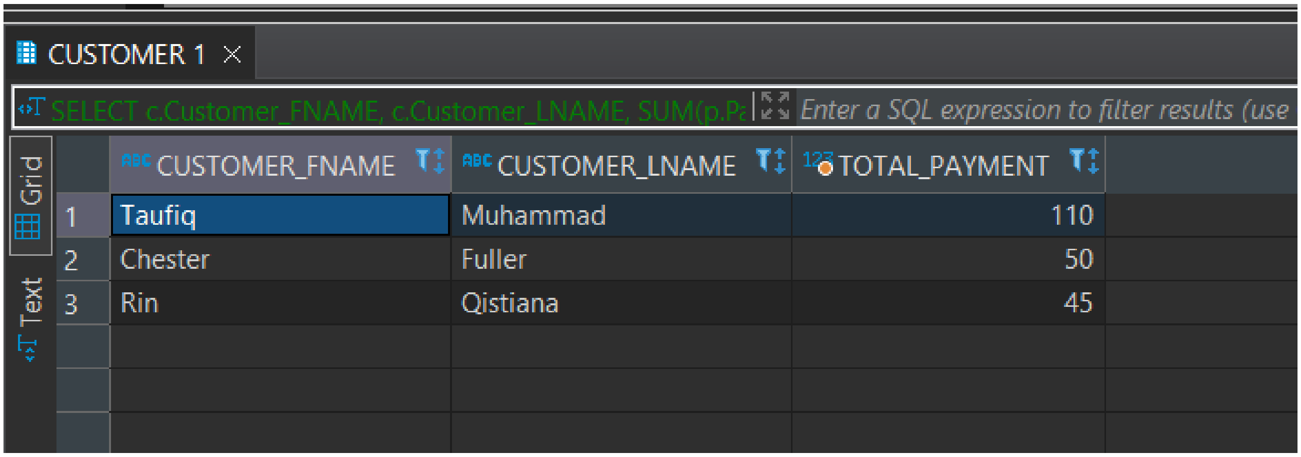
This query retrieves reservation details by joining customer and employee information using CTEs, and orders the result by reservation date in descending order.

Display top N records:



```
SELECT c.Customer_FNAME, c.Customer_LNAME, SUM(p.Payment_Amount) as Total_Payment
FROM Customer c
JOIN Reservation r ON c.Customer_ID = r.Customer_ID
JOIN Invoice i ON r.Reservation_ID = i.Reservation_ID
JOIN Payment p ON i.Payment_ID = p.Payment_ID
GROUP BY c.Customer_ID, c.Customer_FNAME, c.Customer_LNAME
ORDER BY Total_Payment DESC
FETCH FIRST 3 ROWS ONLY;
```

Output:



CUSTOMER 1
SELECT c.Customer_FNAME, c.Customer_LNAME, SUM(p.Pa
Enter a SQL expression to filter results (use
Grid
Text

	CUSTOMER_FNAME	CUSTOMER_LNAME	TOTAL_PAYMENT
1	Taufiq	Muhammad	110
2	Chester	Fuller	50
3	Rin	Qistiana	45

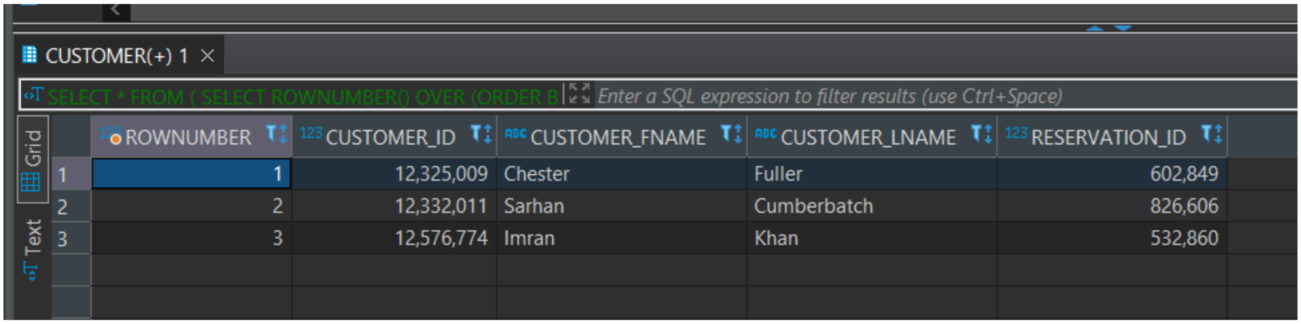
Purpose:

To display any number of rows depending on how many is needed. The query is descending in order of Total_payments, which is the total payments each customer has made.

Window function (ROW_NUMBER():

```
SELECT * FROM (
SELECT ROWNUMBER() OVER (ORDER BY Customer.Customer_ID) as RowNumber,
Customer.Customer_ID, Customer.Customer_FNAME, Customer.Customer_LNAME,
Reservation.Reservation_ID
FROM Customer
INNER JOIN Reservation ON Customer.Customer_ID = Reservation.Customer_ID) as temp
WHERE temp.RowNumber BETWEEN 1 and 3
```

Output:



```
SELECT * FROM ( SELECT ROWNUMBER() OVER (ORDER B
```

Enter a SQL expression to filter results (use Ctrl+Space)

	ROWNUMBER	CUSTOMER_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	RESERVATION_ID
1	1	12,325,009	Chester	Fuller	602,849
2	2	12,332,011	Sarhan	Cumberbatch	826,606
3	3	12,576,774	Imran	Khan	532,860

Purpose:

This query selects specific rows of data from the customer and reservation tables by using a row number function to order the results by customer ID, and then it filters the output to display only a specific range of rows.