Database Design and Implementation

Database design and implementation for a fictional Coffee company

The Cafe — A San Francisco Bay Area Premium Coffee Store Chain

Started in 1993 with a mission of revolutionizing how the average person drinks coffee, *The Café*, a coffee shop chain based in the San Francisco Bay Area, owns three locations in San Francisco and two, respectively, in San Mateo and San Jose. Whilst many coffee shops already exist in the SF Bay Area, *The Café* differentiates itself by specializing in serving coffee brewed from only the highest quality and finest of coffee beans. This local coffee shop chain produces coffee with beans imported from all over the world that guarantee an unforgettable taste whilst ensuring balance in the notes of the drinks that ultimately end up in the customer's mouth.

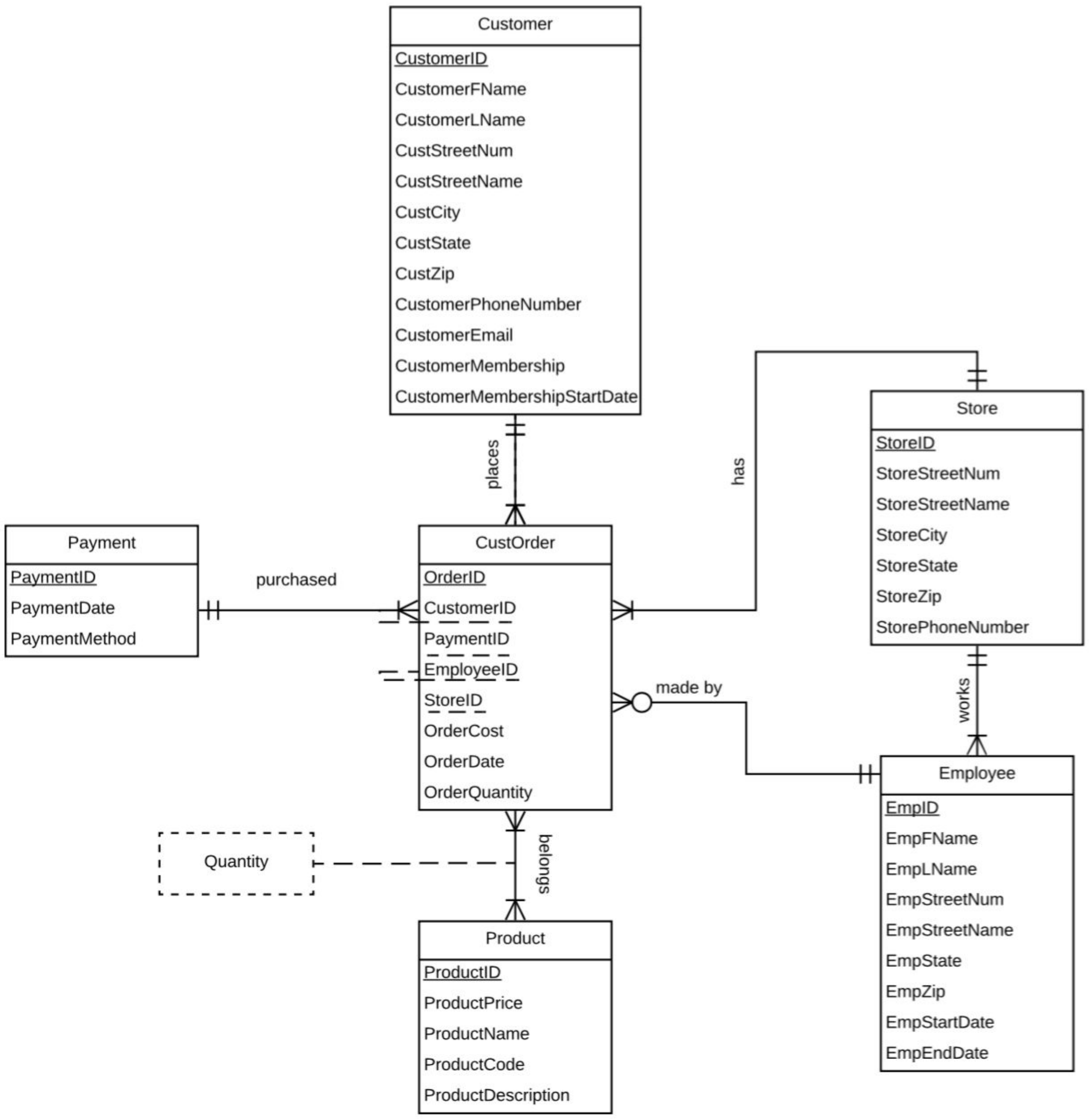
Throughout the years, *The Café* worked on creating a reputation for itself as being a leader in the coffee brewing business. Despite witnessing much success, the coffee shop chain currently deals with a myriad of problems pertaining to its current data management model, with the biggest issues being the effectiveness of its current database design and the efficiency of data storage and retrieval. With this major problem acknowledged, the company decided that it wants to optimize its current practices pertaining to the handling of its data. To achieve the desired end of a more efficient data management environment, we designed database infrastructure centralized on a few major components critical to the day-to-day business operations of *The Café*. Because the coffee shop chain receives a lot of orders, both in-person and over the phone, daily, we decided that it must be able to monitor customer data pertaining to business operations to streamline its business processes, from the beginning of the ordering process to when the customer receives their coffee.

In the database infrastructure designed for *The Café*, we focus on customer, order, product, employee and store data. The database structure of the coffee shop chain starts with *Customer* data. In the database, we created a table called *Customer* that holds information pertaining to the customer, such as their name, address, email, phone number. In addition to the general information about a customer that we receive on a per order basis, each customer database entry receives a unique ID that behaves as the primary key of the table. As a customer places an order, their order gets assigned a unique identifier named OrderID that identifies the order in the *Order* table in the database, which holds *The Café*'s unique order data from each of its locations. After an order has been assigned a unique identifier key, it then gets linked to the customer that placed the order via a foreign key in the *Order* table named *CustID*. Upon placement of an order, the *Order* table references the *Product* table that holds all of the data pertaining to the coffee shop's menu to see what the customer ordered. Aside from the unique order and customer identifiers, this table also holds an order's cost, date, quantity, how it was paid for, the store the order was placed at, the ID of the employee that fulfilled the order and the ID of the customer that a specific order belongs to. After an order has been placed and the relation between said order and customer has been made via the unique identifier keys, the system then identifies the store that the order was placed at and assigns a *StoreID* to the order from the *Store* table.

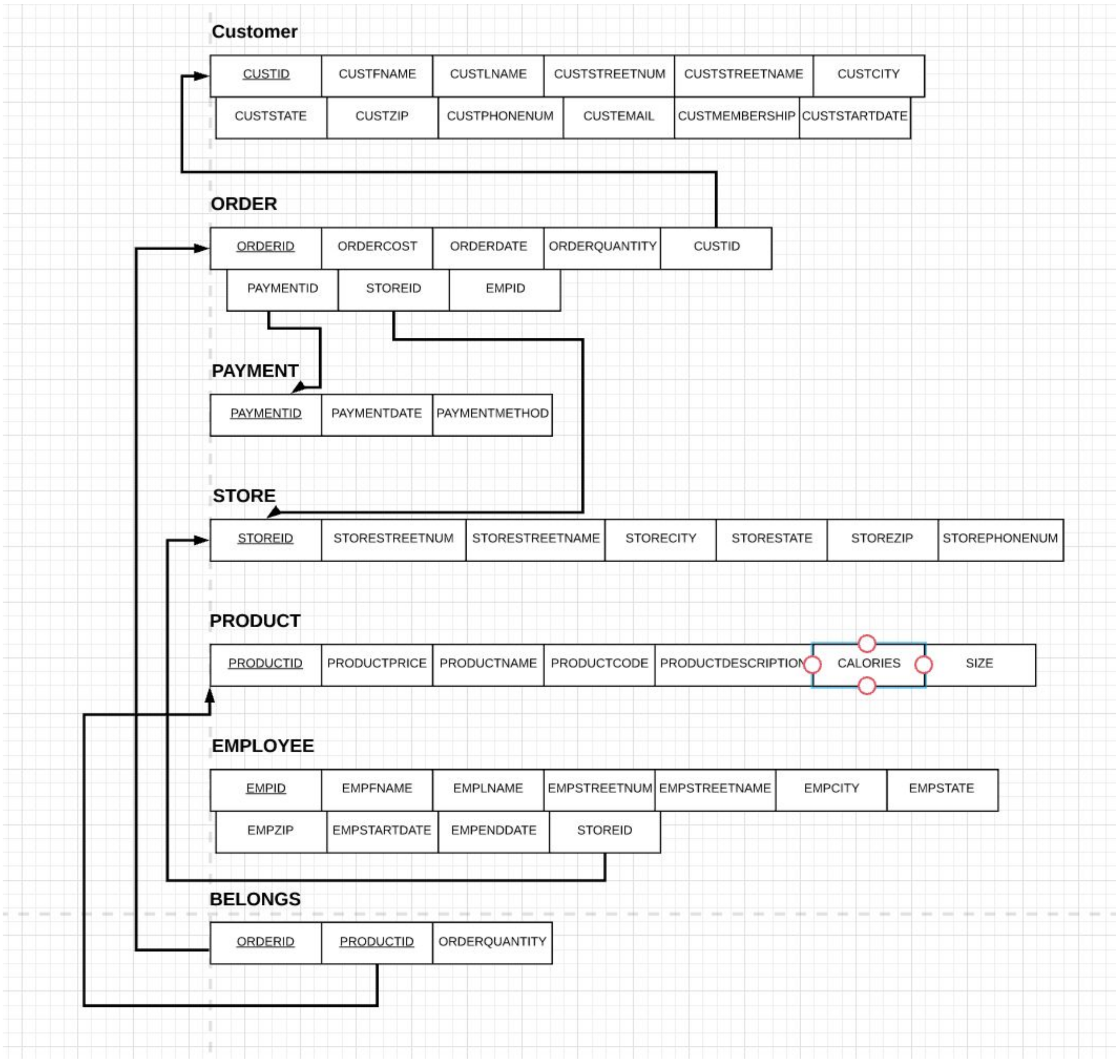
Upon receipt of order, the system then assigns the order to an employee, respective of the location at which the order was placed. *The Cafe* wants to monitor the output of its employees as this allows the company to monitor variables such as who to credit for an order, quality and speed of work, resource waste and how an employee works. As such, we've created table *Employee* for the sole purpose of allowing the company to monitor the work of its employees. This table consists of a unique identifier key assigned to each employee, a foreign identifier key that draws a relationship between the employee and what store they work at via the *StoreID*, identifier information for employees (name, address, phone number), and their start and end dates. After an employee has fulfilled a customer's order, this entire database process starts over again from the *Customer* table.

With this new and improved relational database created for *The Cafe*, it does not look like the premium coffee shop chain will ever run into data inefficiencies ever again. During the intial design stage, Brian Lei and Cody Berlioz desgined the ER diagriam and corresponding logical schema. We analyzed the structure of the business and based off of its structure, we made decisions on what schemas and relationships within those schemas' columns needed to be made. During the database creation and2load stage, Brian Lei helped shape the design of the database and worked on the *Product*, *Employee* and *Belongs* tables and the data and queries for those respective tables. Lanny Tran made the *Payment, Store* and *Order* tables, the data for the *Payment* and *Store* tables, queries for the *Payment* and *Store* tables and helped create the menu for *The Cafe's* menu. Cody Berlioz made the *Customer* and *Order* tables, the data and queries for those tables.

ER Diagram


Customer
CustomerID
CustomerFName
CustomerLName
CustStreetNum
CustStreetName
CustCity
CustState
CustZip
CustomerPhoneNumber
CustomerEmail
CustomerMembership
CustomerMembershipStartDate
places
has
Store
StoreID
StoreStreetNum
StoreStreetName
StoreCity
StoreState
StoreZip
StorePhoneNumber
Payment
PaymentID
PaymentDate
PaymentMethod
purchased
CustOrder
OrderID
CustomerID
PaymentID
EmployeeID
StoreID
OrderCost
OrderDate
OrderQuantity
made by
works
Employee
EmpID
EmpFName
EmpLName
EmpStreetNum
EmpStreetName
EmpState
EmpZip
EmpStartDate
EmpEndDate
Quantity
belongs
Product
ProductID
ProductPrice
ProductName
ProductCode
ProductDescription

Logical Schema



Build Commands

```
create database TheCafe;
use TheCafe;

CREATE TABLE CUSTOMER (
CUSTID INTEGER PRIMARY KEY NOT NULL,
CUSTFNAME VARCHAR (25) NOT NULL,
CUSTLNAME VARCHAR (25) NOT NULL,
CUSTSTREETNUMBER INTEGER NOT NULL,
CUSTSTREETNAME VARCHAR (30) NOT NULL,
CUSTCITY VARCHAR (20) NOT NULL,
CUSTSTATE VARCHAR (25) NOT NULL,
CUSTZIP INTEGER NOT NULL,
CUSTPHONENUMBER TEXT NOT NULL,
CUSTEMAIL VARCHAR (40) NOT NULL,
CUSTMEMBERSHIP VARCHAR (3) NOT NULL,
CUSTSTARTDATE TIMESTAMP NOT NULL
) ;

CREATE TABLE PAYMENT (
PMTID INTEGER PRIMARY KEY NOT NULL,
PMTMETHOD VARCHAR (10) NOT NULL
) ;

CREATE TABLE STORE (
STOREID INTEGER PRIMARY KEY NOT NULL,
STORESTREETNUM INTEGER NOT NULL,
STORESTREET VARCHAR (30) NOT NULL,
STORECITY VARCHAR (20) NOT NULL,
STORESTATE VARCHAR (2) NOT NULL,
STOREZIP INTEGER NOT NULL,
STOREPHONENUM CHAR (13) NOT NULL
) ;

CREATE TABLE EMPLOYEE (
	EMPID INTEGER PRIMARY KEY NOT NULL,
	STOREID INTEGER NOT NULL,
	EMPFNAME VARCHAR (25) NOT NULL,
	EMPLNAME VARCHAR (25) NOT NULL,
	EMPSTREETNUM INTEGER NOT NULL,
	EMPSTREETNAME VARCHAR (25) NOT NULL,
	EMPCITY VARCHAR (25) NOT NULL,
	EMPSTATE VARCHAR (2) NOT NULL,
	EMPZIP INTEGER NOT NULL,
	EMPSTARTDATE TIMESTAMP NOT NULL,
	EMPENDDATE TIMESTAMP NOT NULL,
	CONSTRAINT EMPLOYEE FOREIGN KEY (STOREID) REFERENCES STORE
(STOREID)
	) ;

	CREATE TABLE CUSTORDER (
	ORDERID INTEGER PRIMARY KEY NOT NULL,
	ORDERCOST DECIMAL (15) NOT NULL,
	ORDERDATE DATE NOT NULL,
	ORDERQUANTITY INTEGER NOT NULL,
	CUSTID INTEGER NOT NULL,
	PAYMENTID INTEGER NOT NULL,
	STOREID INTEGER NOT NULL,
	EMPID INTEGER NOT NULL,
	CONSTRAINT CUSTORDER_CUSTOMERID_FK FOREIGN KEY (CUSTID)
REFERENCES CUSTOMER (CUSTID),
	CONSTRAINT CUSTORDER_PAYMENTID_FK FOREIGN KEY (PAYMENTID)
REFERENCES PAYMENT (PMTID),
	CONSTRAINT CUSTORDER_STOREID_FK FOREIGN KEY (STOREID)
REFERENCES STORE (STOREID),
	CONSTRAINT CUSTORDER_EMPLOYEEID_FK FOREIGN KEY (EMPID)
REFERENCES EMPLOYEE (EMPID)
	) ;

	CREATE TABLE PRODUCT (
	PRODUCTID INTEGER PRIMARY KEY NOT NULL,
	PRODUCTPRICE DECIMAL (15,2) NOT NULL,
	PRODUCTNAME TEXT NOT NULL,
	PRODUCTCODE INTEGER NOT NULL,
	PRODUCTDESCRIPTION TEXT NOT NULL,
	PRODUCTCALORIES INTEGER NOT NULL,
	PRODUCTSIZE VARCHAR (10) NOT NULL
	) ;
```

```
        CREATE TABLE BELONGS (
                ORDERID INTEGER,
                PRODUCTID INTEGER,
                ORDERQUANTITY INTEGER,
                CONSTRAINT BELONGS_PK PRIMARY KEY(ORDERID, PRODUCTID),
                CONSTRAINT BELONGS_ORDERID_FK FOREIGN KEY (ORDERID)
REFERENCES CUSTORDER (ORDERID),
                CONSTRAINT BELONGS_PRODUCTID_FK FOREIGN KEY (PRODUCTID)
REFERENCES PRODUCT (PRODUCTID)
        );
```

Data Insertion Commands

```
INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (100, 'Raine', 'Sandever',
23390, 'Lunder', 'Falls Church', 'Virginia', 22047, '571-141-6842', 'rsandever0@si.edu', 'yes',
'20011218') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (101, 'Sky', 'Caddy', 13689,
'Summer Ridge', 'Knoxville', 'Tennessee', 37914, '865-196-0302', 'scaddy1@umn.edu', 'no',
'20020817') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (102, 'Seth', 'Williams', 28538,
'Blue Hill Park', 'Fort Worth', 'Texas', 76105, '817-489-3851', 'shincks@indiegogo.com', 'yes',
'20171117') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (103, 'Marj', 'Jarvis', 10739,
'Delaware', 'Austin', 'Texas', 78744, '361-663-7231', 'mjarvis3@state.gov', 'yes', '20160312') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (104, 'Cesar', 'Pinckard',
```

```sql
20796, 'Amoth', 'Harrisburg', 'Pennsylvania', 17110, '717-261-7552', 'cpinckard4@chron.com',
'no', '20170110') ;


INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (105, 'Elisa', 'Perring', 06924,
'Aberg', 'Annapolis', 'Maryland', 21405, '410-235-4100', 'eperring5@microsoft.com', 'yes',
'20120216') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (106, 'Kahil', 'Willes', 92736,
'Jenifer', 'Austin', 'Texas', 78703, '512-580-9236', 'kwilles6@cdbaby.com', 'no', '20180512');

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (107, 'Pascal', 'Garner', 92611,
'Esch', 'Oklahoma City', 'Oklahoma', 73142, '450-474-1717', 'pgarner7@paginegaille.it', 'yes',
'20110910') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (108, 'Raye', 'Rebichon',
10962, 'Kensington', 'Madison', 'Wisconsin', 53716, '608-257-1121', 'rrebichon8@cmu.edu',
'yes', '20141017') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (109, 'Allister', 'Grestye',
20833, 'Lein', 'Zephyrhills', 'Florida', 33543, '813-267-2091', 'agrestyea@marriot.com', 'no',
'20181018') ;

INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, CUSTSTREETNUMBER,
CUSTSTREETNAME, CUSTCITY, CUSTSTATE, CUSTZIP, CUSTPHONENUMBER,
CUSTEMAIL, CUSTMEMBERSHIP, CUSTSTARTDATE) values (110, 'Kaleena', 'Mcdavid',
29890, 'Valdez', 'Ventura', 'California', 20397, '202-502-3062', 'kmcdavid@com.com', 'no',
'20121010') ;

INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (001, 'Cash');
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (002, 'Check') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (003, 'Visa') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (004, 'MasterCard') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (005, 'Discover') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (006, 'AMex') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (007, 'PayPal') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (008, 'Venmo') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (009, 'Square') ;
INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (010, 'Fund Xfer') ;

INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREET, STORECITY,
STORESTATE, STOREZIP, STOREPHONENUM) VALUES (001, 7512, '19th Ave', 'San
Francisco', 'CA', 94132, '415-124-5689') ;
INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREET, STORECITY,
STORESTATE, STOREZIP, STOREPHONENUM) VALUES (002, 1070, 'Irving st', 'San
Francisco', 'CA', 94122, '415-158-7931') ;

INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREET, STORECITY,
STORESTATE, STOREZIP, STOREPHONENUM) VALUES (003, 3789, 'Geary Street', 'San
Francisco', 'CA', 94121, '415-259-6890') ;

INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREET, STORECITY,
STORESTATE, STOREZIP, STOREPHONENUM) VALUES (004, 1391, 'El Camino Real', 'San
Mateo', 'CA', 94010, '650-679-8403') ;

INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREET, STORECITY,
STORESTATE, STOREZIP, STOREPHONENUM) VALUES (005, 5531, 'San Salvador Street',
'San Jose', 'CA', 950132, '408-832-4381') ;

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(001, '3.00', 'Coffee', 110, 'Regular Coffee', 10, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(002, '5.90', 'Kona Coffee', 111, 'Coffe brewed with Kona Coffee Beans, revererd as the best
coffee beans in America. This coffee has a balanced taste. Not too bitter, not too sweet,', 5,
'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(003, '7.49', 'Blue Moutain Coffee', 112, 'What is Blue Mountain Coffee you ask? Well, it’s coffee
that was made with beans from the Jamaican Blue Mountains. This coffee is made with beans
that are actually from Jamaica!', 5, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(004, '6.99', 'AA Coffee', 113, 'Medium roast coffee brewed with the famous Kenyan AA Coffee
Beans, imported directly from Kenya.', 5, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(005, '5.50', 'Peaberry Coffee', 114, 'Drip brew style coffee, this cup of coffee has notes of brown
sugary goodness from the Peaberry Beans from Tanzania that it’s brewed from whilst
maintaining a hint of bitterness that everyone expects from a good cup of coffee.', 35, 'Medium');
INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(006, '9.50', 'Sulawesi Coffee', 115, 'If you’re looking for a cup of coffee that’s sweet, look no
further! Brewed from Sulawesi Toraja Coffee beans imported directly from Indonesia, this cup of
coffee is as sweet as the coffee on our menu gets.', 150, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(007, '18.50', 'Centeral American Geisha Coffee', 116, 'A rare find, literally. This cup of coffee is
brewed from the Central American Geisha coffee beans, a rare coffee bean find. It has a
tea-like body with an abundance of sparkling flavors for taste. Get this if you want to
demonstrate your inner coffee connoisseur, and especially if you love tea.', 10, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(008, '3.00', 'Tea', '222', 'Hot Tea', 5, 'Medium');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(009, '4.00', 'Hot Chocolate', 333, 'Chocolate with Coca Powder', 100, 'Small');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(010, '1.00', 'Water', 444, 'Hot or Cold Water', 10, 'Large');

INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCTNAME,
PRODUCTCODE, PRODUCTDESCRIPTION, PRODUCTCALORIES, PRODUCTSIZE) values
(011, '2.00', 'Juice', 555, 'Tropicana', 10, 'Medium');

INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNAME, EMPSTREETNUM,
EMPSTREETNAME, EMPCITY, EMPSTATE, EMPZIP, EMPSTARTDATE, EMPENDDATE)
values (001, 001, 'Yule', 'Verdun', 228, 'Cambridge Avenue', 'Lancaster', 'CA', '94132', '190411',
'150813');
```

```sql
INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNAME, EMPSTREETNUM,
EMPSTREETNAME, EMPCITY, EMPSTATE, EMPZIP, EMPSTARTDATE, EMPENDDATE)
values (002, 002, 'Emily', 'Weber', 7747, 'Center Way', 'Bismarck', 'ND', '90112', '051113',
'140615');

INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNAME, EMPSTREETNUM,
EMPSTREETNAME, EMPCITY, EMPSTATE, EMPZIP, EMPSTARTDATE, EMPENDDATE)
values (003, 003, 'Julie', 'Ellwood', 2810, 'Ryan Street', 'Lancaster', 'WA', '89093', '101211',
'140814');

INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNAME, EMPSTREETNUM,
EMPSTREETNAME, EMPCITY, EMPSTATE, EMPZIP, EMPSTARTDATE, EMPENDDATE)
values (004, 004, 'Cedric', 'Pedrazzi', 5, 'Amoth Avenue', 'Albany', 'NY', '79291', '081012',
'010717');

INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNAME, EMPSTREETNUM,
EMPSTREETNAME, EMPCITY, EMPSTATE, EMPZIP, EMPSTARTDATE, EMPENDDATE)
values (005, 005, 'Rocky', 'Viola', 198, 'Tony Court', 'Knoxville', 'TN', '32413', '020212', '071118');

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (010, 4.20, '020815', 2, 100, 002, 002, 003) ;

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (020, 5.19, '041212', 3, 101, 006, 001, 001) ;

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (030, 6.00, '140118', 4, 102, 007, 003, 002);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (040, 3.50, '140818', 1, 103, 010, 004, 004);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (050, 7.50, '170512', 2, 104, 009, 001, 005);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (060, 8.70, '180613', 2, 105, 009, 001, 005);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (070, 10.50, '190714', 3, 106, 009, 001, 005);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY,
CUSTID, PMTID, STOREID, EMPID) values (080, 12.95, '220516', 5, 107, 009, 001, 005);
```

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY, CUSTID, PMTID, STOREID, EMPID) values (090, 5.50, '251015', 7, 108, 009, 001, 005);

INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE, ORDERQUANTITY, CUSTID, PMTID, STOREID, EMPID) values (100, 15.50, '160812', 10, 109, 009, 001, 005);

INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY) values (010, 004, 5);

INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY) values (020, 002, 2);

INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY) values (030, 003, 1);
INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY) values (040, 006, 4);

INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY) values (050, 002, 3);

Database Queries

/*Retrieve all possible Payment Methods*/
SELECT * PMTMETHOD FROM PAYMENT;

/* Verifying the existence of a specific payment method (e.g. Visa) */
SELECT 1 FROM PAYMENT WHERE PMTMETHOD = “Visa”;

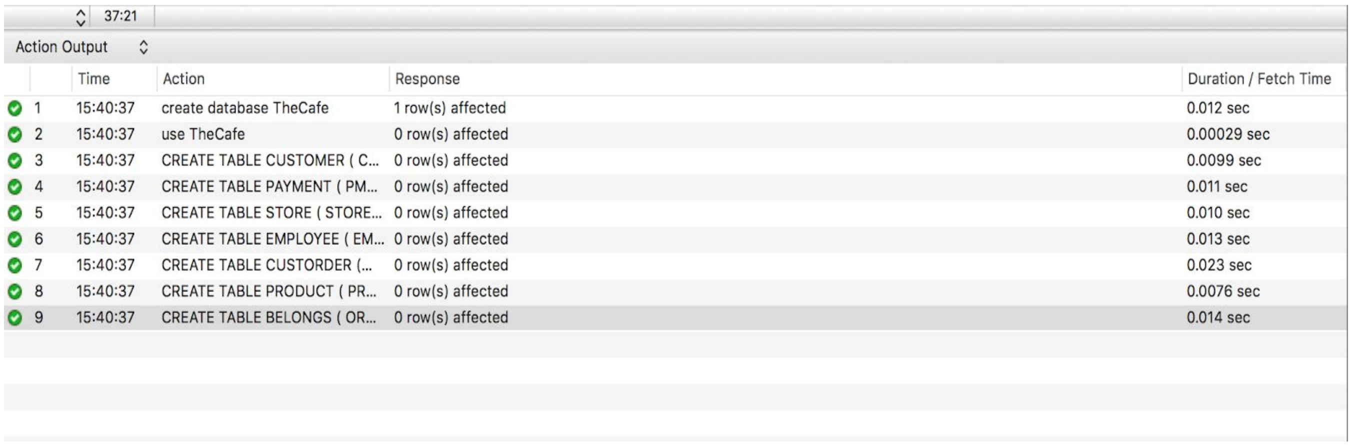
/* List all of The Cafe’s store location street addresses with city and with specific store ID’s*/
SELECT STORESTREET, STORECITY, STOREID FROM STORE;

/*Retrieve Store phone numbers w/ store ID*/
SELECT STOREID, STOREPHONENUM FROM STORE;

/* Get all product over $5*/
SELECT PRODUCTID FROM PRODUCT
WHERE PRODUCTPRICE>5;

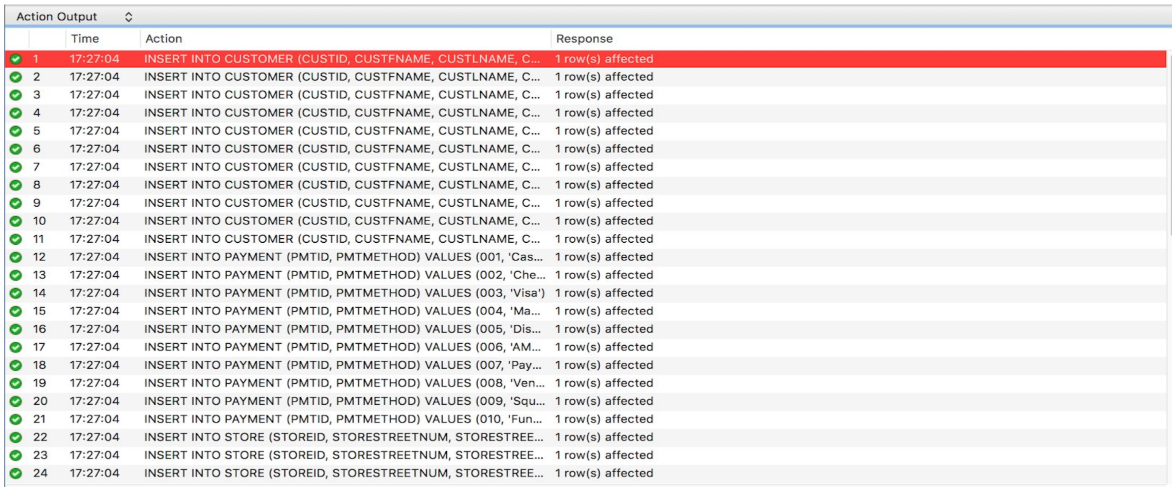
/* Get all customers that are members*/
SELECT CUSTFNAME, CUSTLNAME FROM CUSTOMER
WHERE CUSTMEMBERSHIP = ‘yes’

Database Creation Commands Log

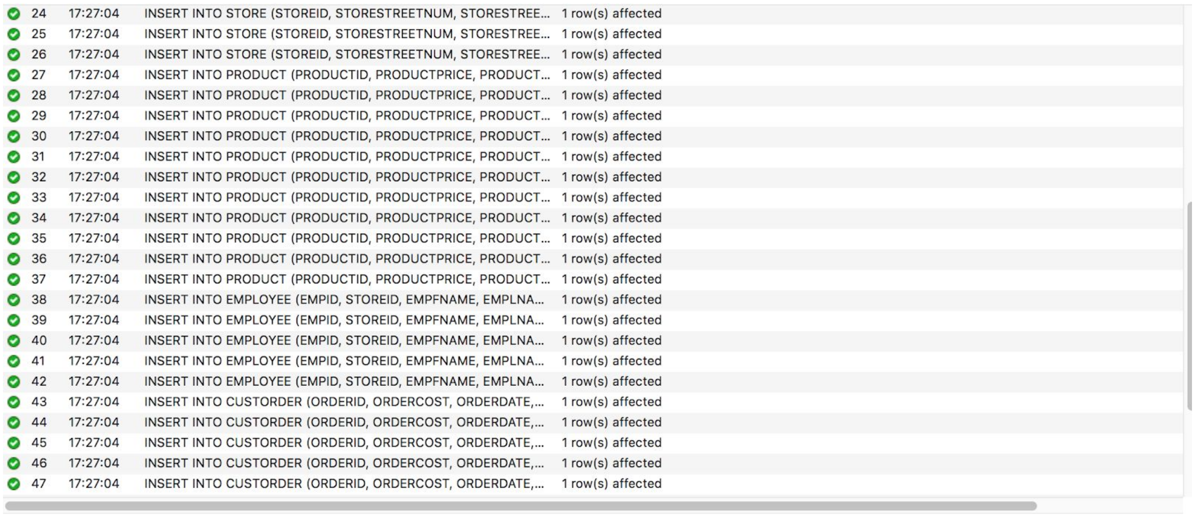


```
     Time      Action                            Response            Duration / Fetch Time
1    15:40:37  create database TheCafe           1 row(s) affected   0.012 sec
2    15:40:37  use TheCafe                       0 row(s) affected   0.00029 sec
3    15:40:37  CREATE TABLE CUSTOMER ( C...      0 row(s) affected   0.0099 sec
4    15:40:37  CREATE TABLE PAYMENT ( PM...      0 row(s) affected   0.011 sec
5    15:40:37  CREATE TABLE STORE ( STORE...     0 row(s) affected   0.010 sec
6    15:40:37  CREATE TABLE EMPLOYEE ( EM...     0 row(s) affected   0.013 sec
7    15:40:37  CREATE TABLE CUSTORDER (...       0 row(s) affected   0.023 sec
8    15:40:37  CREATE TABLE PRODUCT ( PR...      0 row(s) affected   0.0076 sec
9    15:40:37  CREATE TABLE BELONGS ( OR...      0 row(s) affected   0.014 sec
```

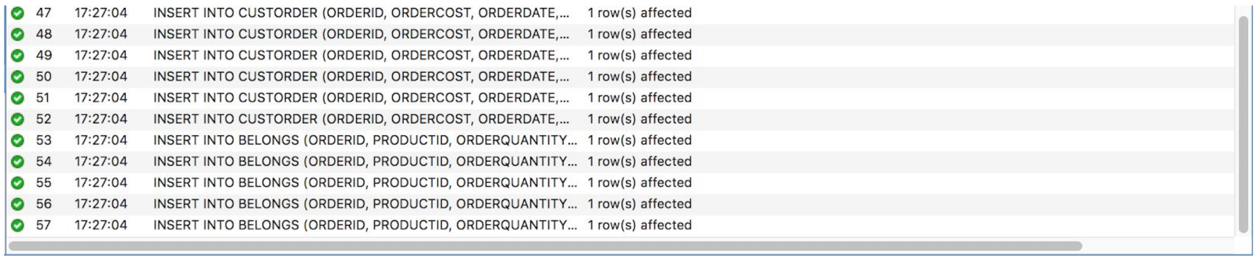
Data Insertion Commands Log



```
     Time      Action                                                         Response
1    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
2    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
3    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
4    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
5    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
6    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
7    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
8    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
9    17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
10   17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
11   17:27:04  INSERT INTO CUSTOMER (CUSTID, CUSTFNAME, CUSTLNAME, C...       1 row(s) affected
12   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (001, 'Cas...    1 row(s) affected
13   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (002, 'Che...    1 row(s) affected
14   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (003, 'Visa')    1 row(s) affected
15   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (004, 'Ma...     1 row(s) affected
16   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (005, 'Dis...    1 row(s) affected
17   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (006, 'AM...     1 row(s) affected
18   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (007, 'Pay...    1 row(s) affected
19   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (008, 'Ven...    1 row(s) affected
20   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (009, 'Squ...    1 row(s) affected
21   17:27:04  INSERT INTO PAYMENT (PMTID, PMTMETHOD) VALUES (010, 'Fun...    1 row(s) affected
22   17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...      1 row(s) affected
23   17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...      1 row(s) affected
24   17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...      1 row(s) affected
```

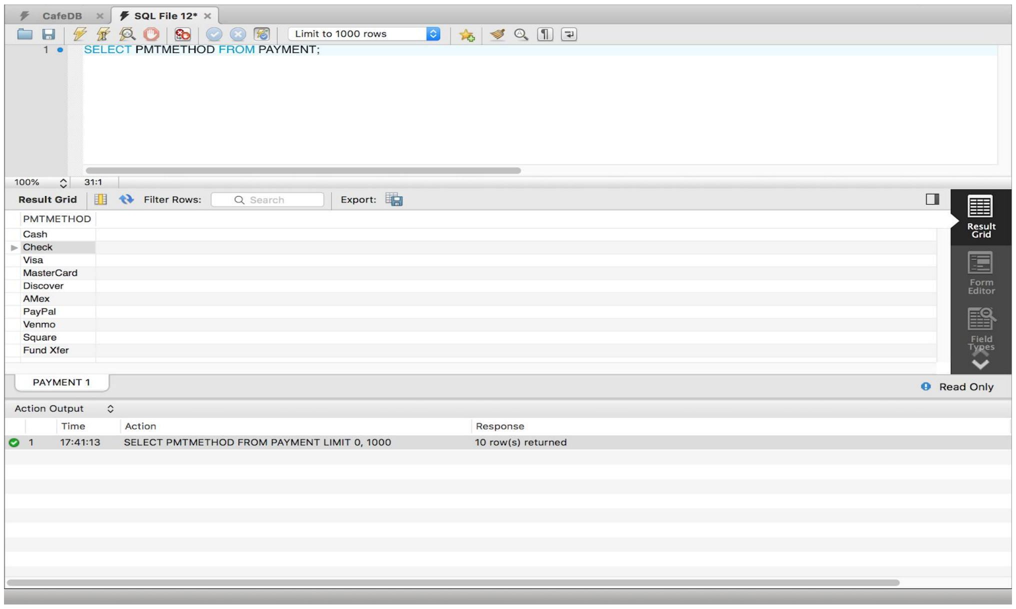


```
24  17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...  1 row(s) affected
25  17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...  1 row(s) affected
26  17:27:04  INSERT INTO STORE (STOREID, STORESTREETNUM, STORESTREE...  1 row(s) affected
27  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
28  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
29  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
30  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
31  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
32  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
33  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
34  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
35  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
36  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
37  17:27:04  INSERT INTO PRODUCT (PRODUCTID, PRODUCTPRICE, PRODUCT...  1 row(s) affected
38  17:27:04  INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNA...  1 row(s) affected
39  17:27:04  INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNA...  1 row(s) affected
40  17:27:04  INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNA...  1 row(s) affected
41  17:27:04  INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNA...  1 row(s) affected
42  17:27:04  INSERT INTO EMPLOYEE (EMPID, STOREID, EMPFNAME, EMPLNA...  1 row(s) affected
43  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
44  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
45  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
46  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
47  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
```



```
47  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
48  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
49  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
50  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
51  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
52  17:27:04  INSERT INTO CUSTORDER (ORDERID, ORDERCOST, ORDERDATE,...  1 row(s) affected
53  17:27:04  INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY...  1 row(s) affected
54  17:27:04  INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY...  1 row(s) affected
55  17:27:04  INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY...  1 row(s) affected
56  17:27:04  INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY...  1 row(s) affected
57  17:27:04  INSERT INTO BELONGS (ORDERID, PRODUCTID, ORDERQUANTITY...  1 row(s) affected
```

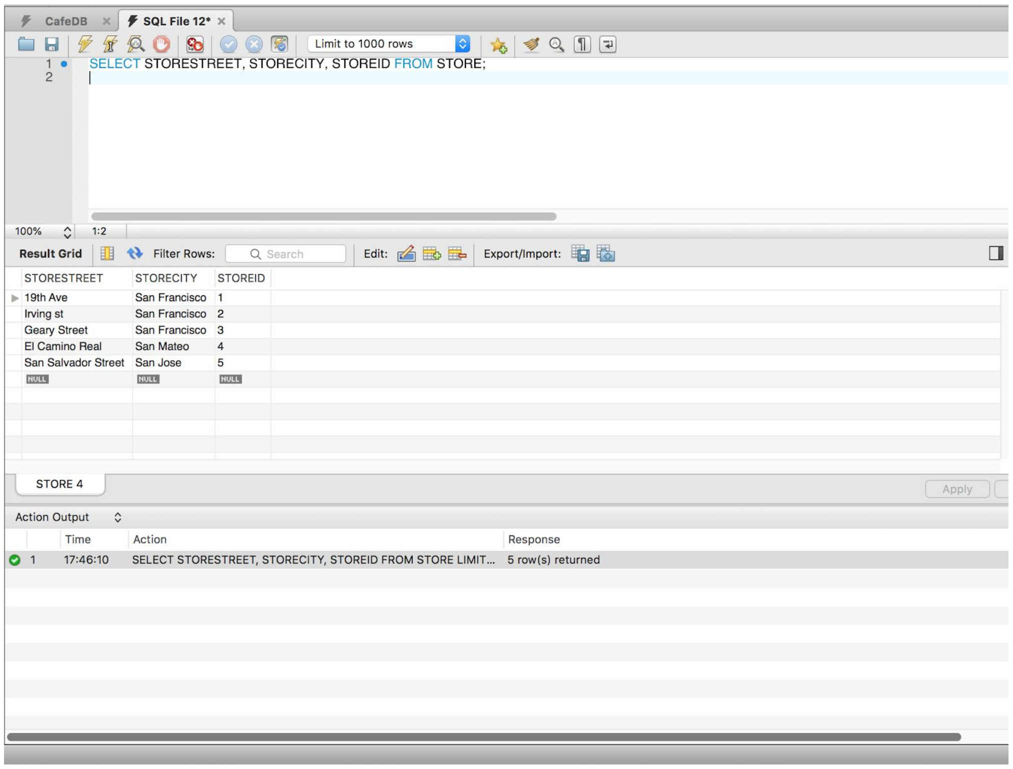
Database Query Logs



```
SELECT PMTMETHOD FROM PAYMENT;
```

PMTMETHOD
Cash
Check
Visa
MasterCard
Discover
AMex
PayPal
Venmo
Square
Fund Xfer

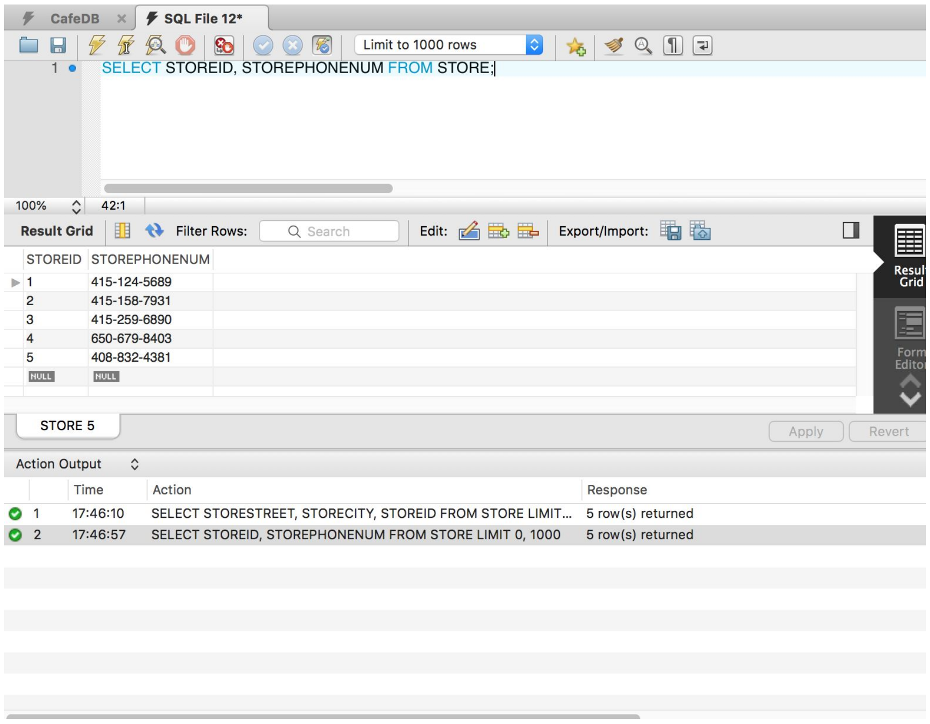
	Time	Action	Response
1	17:41:13	SELECT PMTMETHOD FROM PAYMENT LIMIT 0, 1000	10 row(s) returned



```
SELECT STORESTREET, STORECITY, STOREID FROM STORE;
```

STORESTREET	STORECITY	STOREID
19th Ave	San Francisco	1
Irving st	San Francisco	2
Geary Street	San Francisco	3
El Camino Real	San Mateo	4
San Salvador Street	San Jose	5
NULL	NULL	NULL

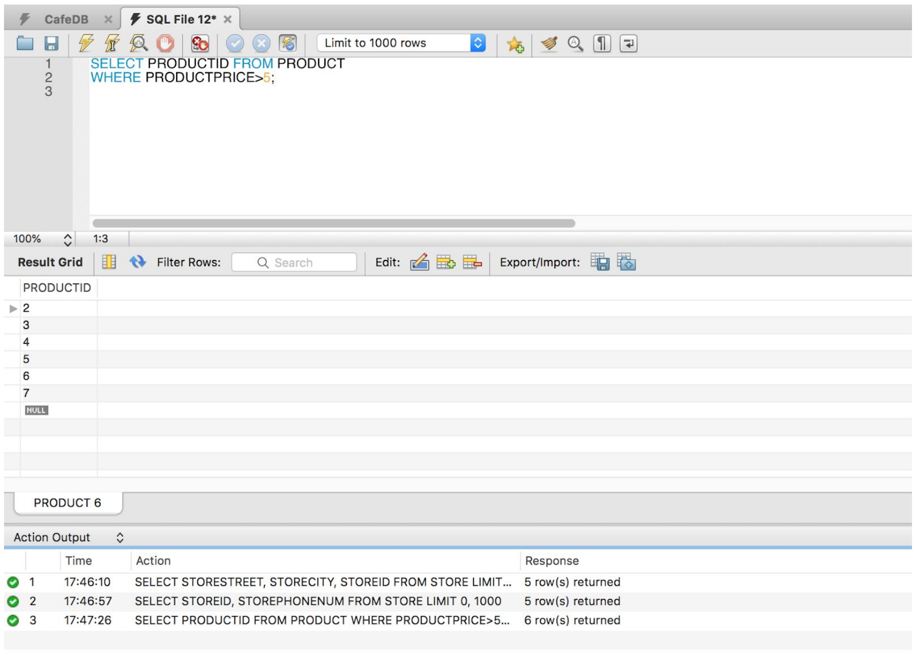
	Time	Action	Response
1	17:46:10	SELECT STORESTREET, STORECITY, STOREID FROM STORE LIMIT...	5 row(s) returned



```
SELECT STOREID, STOREPHONENUM FROM STORE;
```

STOREID	STOREPHONENUM
1	415-124-5689
2	415-158-7931
3	415-259-6890
4	650-679-8403
5	408-832-4381
NULL	NULL

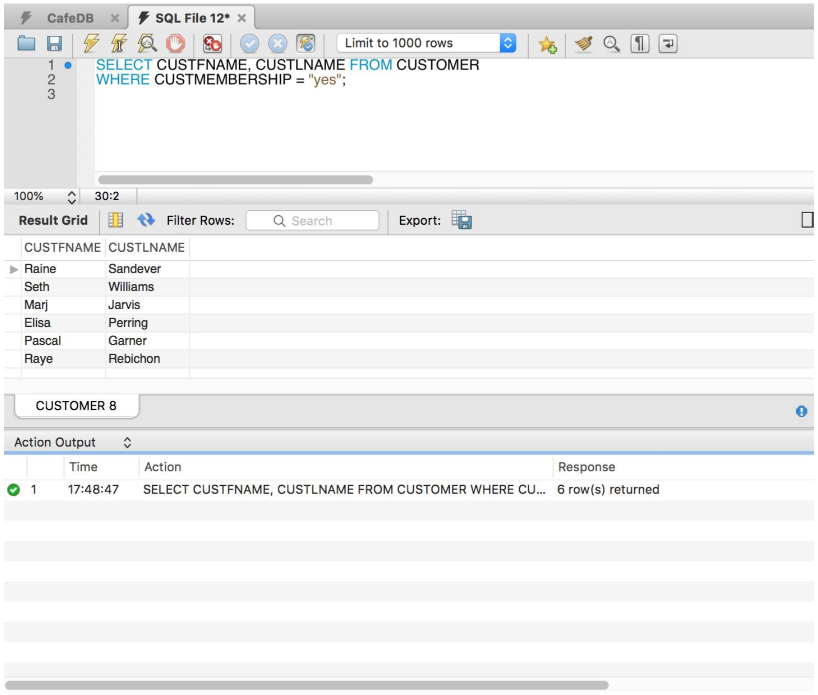
	Time	Action	Response
1	17:46:10	SELECT STORESTREET, STORECITY, STOREID FROM STORE LIMIT...	5 row(s) returned
2	17:46:57	SELECT STOREID, STOREPHONENUM FROM STORE LIMIT 0, 1000	5 row(s) returned



```
SELECT PRODUCTID FROM PRODUCT
WHERE PRODUCTPRICE>5;
```

PRODUCTID
2
3
4
5
6
7
NULL

	Time	Action	Response
1	17:46:10	SELECT STORESTREET, STORECITY, STOREID FROM STORE LIMIT...	5 row(s) returned
2	17:46:57	SELECT STOREID, STOREPHONENUM FROM STORE LIMIT 0, 1000	5 row(s) returned
3	17:47:26	SELECT PRODUCTID FROM PRODUCT WHERE PRODUCTPRICE>5...	6 row(s) returned



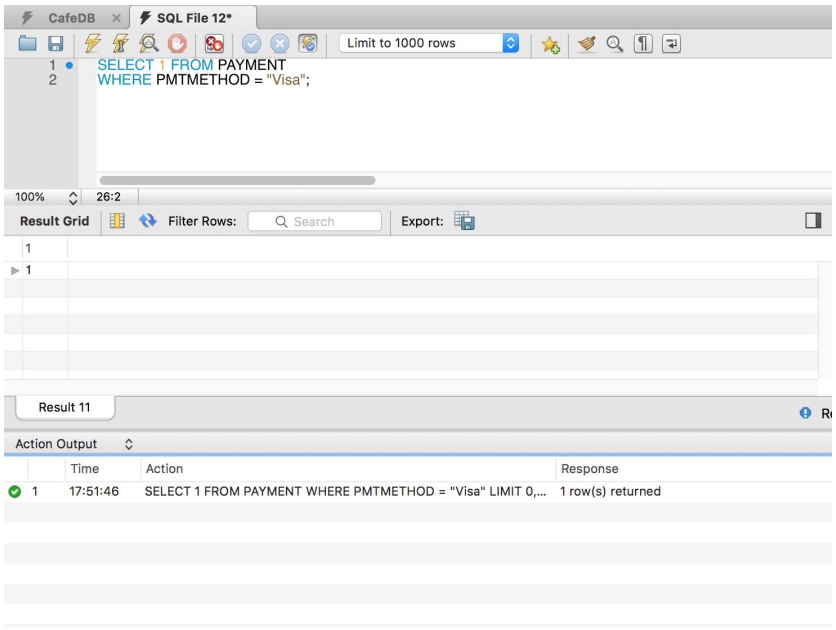
```
SELECT CUSTFNAME, CUSTLNAME FROM CUSTOMER
WHERE CUSTMEMBERSHIP = "yes";
```

```
CUSTFNAME  CUSTLNAME
Raine      Sandever
Seth       Williams
Marj       Jarvis
Elisa      Perring
Pascal     Garner
Raye       Rebichon
```

CUSTOMER 8

Action Output

```
   Time      Action                                             Response
1  17:48:47  SELECT CUSTFNAME, CUSTLNAME FROM CUSTOMER WHERE CU...  6 row(s) returned
```



```
SELECT 1 FROM PAYMENT
WHERE PMTMETHOD = "Visa";
```

```
1
1
```

Result 11

Action Output

```
   Time      Action                                                Response
1  17:51:46  SELECT 1 FROM PAYMENT WHERE PMTMETHOD = "Visa" LIMIT 0,...  1 row(s) returned
```