Interim Final Report

Team 02

Quinn Phillips
Adam Warner
Thomas Xiong

S307: Fall, 2021

12/08/2021

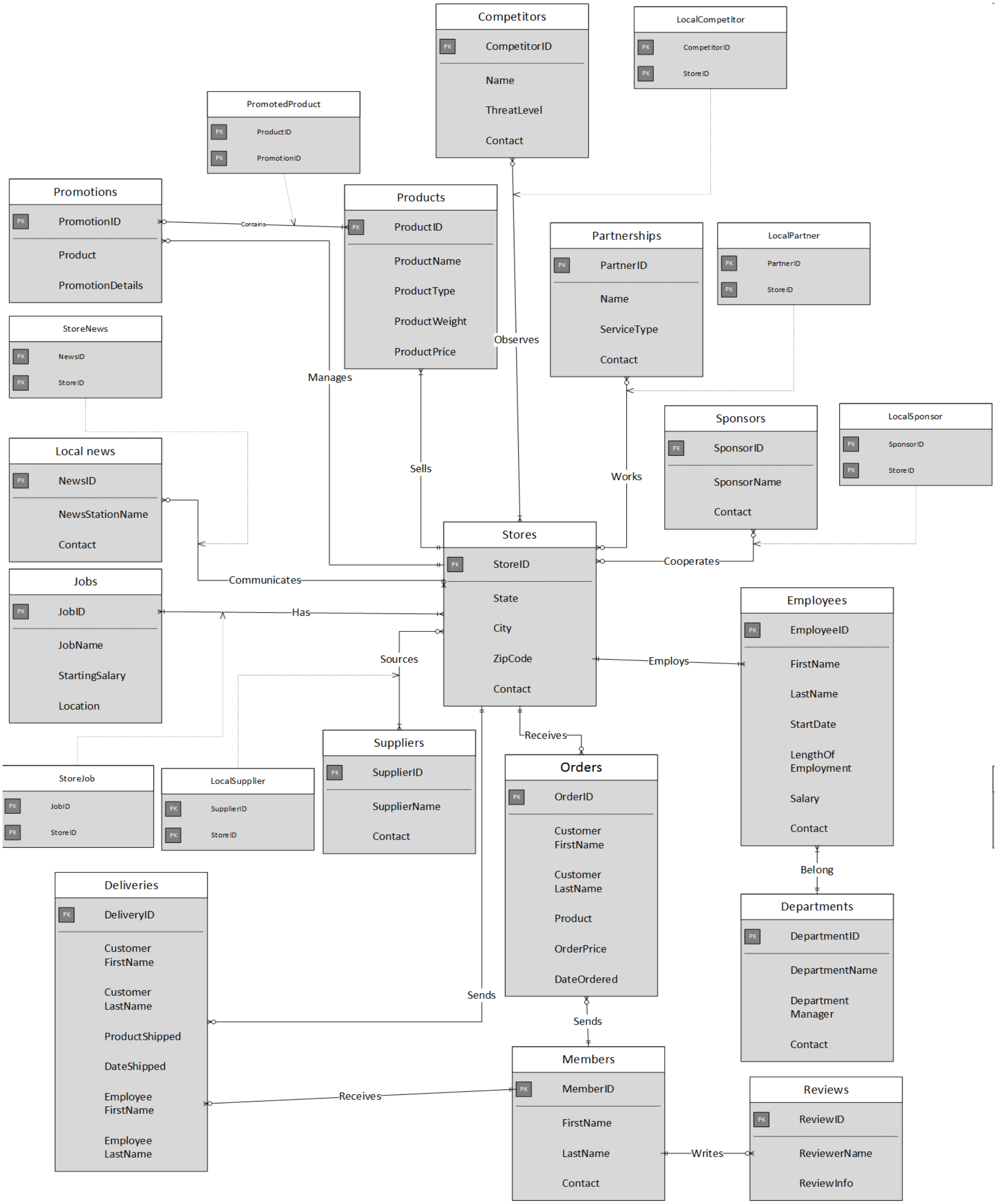
Table of Contents

1 Introduction

Our goal in this project was to create a database for Fresh Thyme that would assist the company in their day to day tasks. This database would allow Fresh Thyme to access key information about their members, suppliers, products and so much more in a quick and easy manner. In addition, this database has a number of constraints that are designed to minimize the instances of database errors. Ideally, any employee, experienced or inexperienced, would be able to access and understand the information coming from this database. Through the use of an ER diagram, a data dictionary, and logical schema, the inner workings of the database will be clearer to more technical employees. This will make it easier for them to change the database as they please in order to react to future situations. Towards the end of the project, we provide the SQL code that we used to implement the database.

2 Design of the Database

2.1 Conceptual Design



Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
SUPPLIERS Entity Type: *Suppliers consist of the vendors responsible for supplying grocery and merchandise items.*				
SupplierName	Alphanumeric	Supplier1, Supplier 2, ...	No	This is the company name of the SUPPLIER/s.
Contact	Alphanumeric	Ex.Phone	No	The phone number of the supplier.

Table 1: SUPPLIER Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
MEMBERS Entity Type*: Members consist of customers who choose to pay set, repeated payments in exchange for greater access and benefits.*				
FirstName	Alphanumeric	John	No	This is the first name of people in the MEMBER program.
LastName	Alphanumeric	Doe	No	This is the last name of people in the MEMBER program.
Contact	Alphanumeric	Ex.Phone	No	The contact information of the MEMBER (phone).

Table 2: MEMBER Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
PRODUCTS Entity Type: *Products are types of goods sold in the Fresh Thyme stores.*				
ProductName	Alphanumeric	Product1, Product2, ...	No	The name of a grocery PRODUCT (ex: Fresh Thyme Marinara Sauce).
ProductType	Alphanumeric	Ex: Hygiene, Produce, Packaged Goods	No	The type of grocery PRODUCT (bakery, dairy, frozen, etc.).
ProductWeight	Decimal	99.999	Yes	The weight of the grocery PRODUCT.
ProductPrice	Decimal	$99,999.99	No	The price of the grocery PRODUCT.

Table 3: PRODUCT Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
STORES Entity Type: *Stores are where the majority of PRODUCTS are sold and where all the EMPLOYEES work.*				
State	Alphanumeric	2 CHAR state abbreviation	No	The state abbreviation in which the STORE is located.
City	Alphanumeric	Ex: Houston, Atlanta, Las Vegas	No	The city in which the STORE is located.
ZipCode	Numeric	99999-9999	No	The zip code in which the STORE is located.
Contact	Alphanumeric	Ex.Phone	No	The STORE contact information (phone).

Table 4: Data Dictionary for STORE Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
DEPARTMENTS Entity Type: *Departments is a subtype of STORES and consists of different groupings of EMPLOYEES.*				
DepartmentName	Alphanumeric	Ex: stocking, retail, management, quality control, etc.	No	The name of the employee DEPARTMENT.
DepartmentManager	Alphanumeric	Doe, John	Yes	The name of DEPARTMENT managers.
Contact	Alphanumeric	Ex.Phone	No	The phone number for each DEPARTMENT.

Table 5: Data Dictionary for DEPARTMENT Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
EMPLOYEES Entity Type: *This represents workers at Fresh Thyme who work*				

together within a STORE.				
FirstName	Alphanumeric	John	No	The first name of an EMPLOYEE.
LastName	Alphanumeric	Doe	No	The last name of an EMPLOYEE.
StartDate	Timestamp	MM/DD/YYYY	No	The start date of the EMPLOYEE.
LengthOfEmployment	Numeric	99	Yes	How long the EMPLOYEE has been working for the STORE.
Salary	Decimal	$999,999.99	No	The current salary of the EMPLOYEE.
Contact	Alphanumeric	Ex.Phone	No	The phone number of the EMPLOYEE

Table 6: Data Dictionary for EMPLOYEE Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
DELIVERIES Entity Type: *This consists of PRODUCTS being sold to customers and the employees responsible for getting the merchandise there on time.*				
CustomerFirstName	Alphanumeric	John	No	The DELIVERY customer’s first name.
CustomerLastName	Alphanumeric	Doe	No	The DELIVERY customer’s last name.
ProductShipped	Timestamp	MM/DD/YYYY	Yes	The PRODUCT shipped in the DELIVERY.
DateShipped	Timestamp	MM/DD/YYYY	Yes	The date of DELIVERY shipment.
EmployeeFirstName	Alphanumeric	John	Yes	The DELIVERY employee’s first name.
EmployeeLastName	Alphanumeric	Doe	Yes	The DELIVERY employee’s last name.

Table 7: Data Dictionary for DELIVERY Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
JOBS Entity Type: *Jobs are the set of positions available for EMPLOYEES.*				
JobName	Alphanumeric	Position,	No	The JOB position

		Position2, ...		title (ex: cashier, custodian, bagger, stocker, food prep. worker, etc.).
StartingSalary	Decimal	$999,999.99	No	The starting salary for the JOB.
Location	Alphanumeric	City, 2 CHAR state abr. EX: Houston, TX	No	The location of the JOB position.

Table 8: Data Dictionary for JOB Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
PROMOTIONS Entity Type: *Promotions are rewards handed out to employees based on tenure or recognition of hard work.*				
Product	Numeric	99999	No	The PRODUCT that currently has a PROMOTION.
PromotionDetails	Alphanumeric	Short Text	No	The details of the PROMOTION (how long, what % off).

Table 9: Data Dictionary for PROMOTION Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
SPONSORS Entity Type: *Sponsors are a set of non-connected companies and entities that feature Fresh Thyme in their marketing or public relations.*				
SponsorName	Alphanumeric	Sponsor1, Sponsor2, ...	No	The name of the SPONSOR company,
Contact	Alphanumeric	Ex.Phone	No	The phone number of the SPONSOR.

Table 10: Data Dictionary for SPONSOR Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
LOCAL NEWS Entity Type: *Local News consists of local news outlets that potentially include relevant stories in their coverage.*				
NewsStationN	Alphanumeric	Name1, Name2,	No	The name of the

ame		...		LOCAL NEWS station.
Contact	Alphanumeric	Ex.Phone	No	The phone number of the LOCAL NEWS station.

Table 11: Data Dictionary for LOCAL NEWS Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
REVIEWS Entity Type: *This is feedback from customers, typically through a star and comment system that is later used in reviewing improvement areas.*				
ReviewerName	Alphanumeric	Doe, John	Yes	The last,first name of the person submitting a REVIEW.
ReviewInfo	Alphanumeric	Short Text	No	The contents of the REVIEW (rating, text).

Table 12: Data Dictionary for REVIEWS Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
COMPETITORS Entity Type: *These are the closest and most direct threats to Fresh Thyme.*				
Name	Alphanumeric	Comp1, Comp2, ...	No	The name of COMPETITOR companies.
ThreatLevel	Numeric	1:100	No	How serious of a threat the COMPETITOR is ranked on a scale of 1-5, 5 being very serious.
Contact	Alphanumeric	Ex.Phone	Yes	The phone number of the COMPETITOR.

Table 13: Data Dictionary for COMPETITOR Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
ORDERS Entity Type: *These are orders through our virtual platform, an emerging alternate option for customers unable to shop in-store.*				
CustomerFirs	Alphanumeric	John	No	The first name of

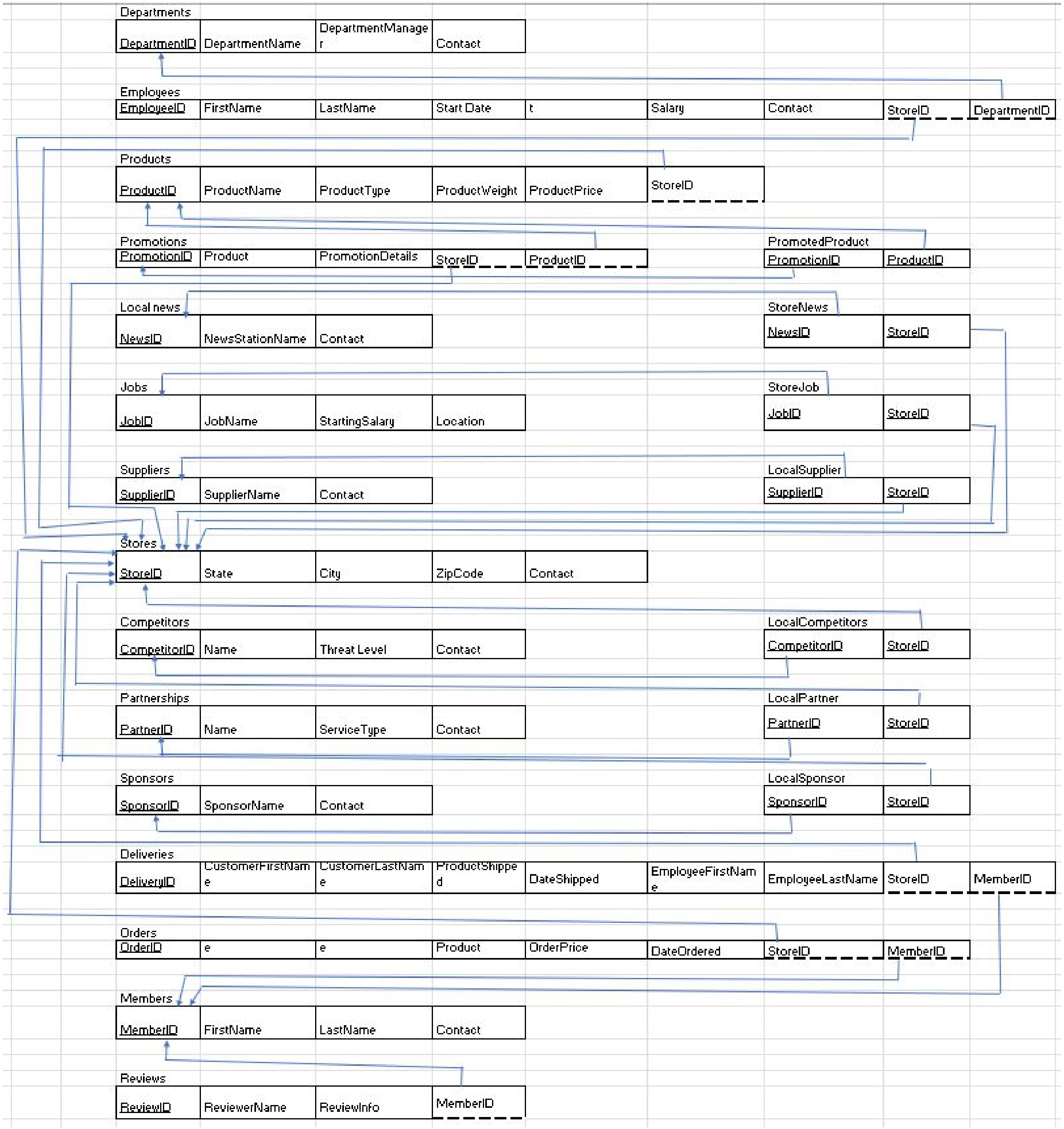
tName				the customer placing the ORDER.
CustomerLastName	Alphanumeric	Doe	No	The last name of the customer placing the ORDER.
Product	Alphanumeric	Product1, Product2, ...	No	The PRODUCT name in the ORDER.
OrderPrice	Decimal	$99,999.99	No	The total cost of the ORDER.
DateOrdered	Timestamp	MM/DD/YYYY	No	The date that the ORDER was placed.

Table 14: Data Dictionary for ORDER Entity Type

Attribute Name	Data Type	Valid Values	NULL Allowed	Description of the attribute
PARTNERSHIPS Entity Type: *Partnerships are sets of mutually beneficial deals with affiliated companies.*				
Name	Alphanumeric	Partner1, Partner2, ...	No	The name of PARTNER companies(instacart).
ServiceType	Alphanumeric	Choose from available values EX: Community Service	No	What type of service the PARTNER company provides for the STORE.
Contact	Alphanumeric	Ex.Phone	No	The phone number of the PARTNER company.

Table 15: Data Dictionary for PARTNERSHIP Entity Type

2.2 Logical Design



Departments

DepartmentID	DepartmentName	DepartmentManager	Contact

Employees

EmployeeID	FirstName	LastName	Start Date	t	Salary	Contact	StoreID	DepartmentID

Products

ProductID	ProductName	ProductType	ProductWeight	ProductPrice	StoreID

Promotions

PromotionID	Product	PromotionDetails	StoreID	ProductID

PromotedProduct

PromotionID	ProductID

Local news

NewsID	NewsStationName	Contact

StoreNews

NewsID	StoreID

Jobs

JobID	JobName	StartingSalary	Location

StoreJob

JobID	StoreID

Suppliers

SupplierID	SupplierName	Contact

LocalSupplier

SupplierID	StoreID

Stores

StoreID	State	City	ZipCode	Contact

Competitors

CompetitorID	Name	Threat Level	Contact

LocalCompetitors

CompetitorID	StoreID

Partnerships

PartnerID	Name	ServiceType	Contact

LocalPartner

PartnerID	StoreID

Sponsors

SponsorID	SponsorName	Contact

LocalSponsor

SponsorID	StoreID

Deliveries

DeliveryID	CustomerFirstName	CustomerLastName	ProductShipped	DateShipped	EmployeeFirstName	EmployeeLastName	StoreID	MemberID

Orders

OrderID	e	e	Product	OrderPrice	DateOrdered	StoreID	MemberID

Members

MemberID	FirstName	LastName	Contact

Reviews

ReviewID	ReviewerName	ReviewInfo	MemberID

Normalization: All relations in the logical schema have been normalized into the third normal form.

2.3 Physical Design

```
-- INDEX's --
-- 1.
SELECT * from product
CREATE Index IX_product
ON product(ProductID ASC, ProductName);
-- DROP INDEX IX_product ON product;
```

We decided to create an index on the ProductID/ProductName because #1. product records will be searched frequently, and #2. there will be well over 100 products in a grocery store database. This non-clustered index acts as a pointer for query/database retrieval which greatly speeds up the retrieval time of the data.

```
-- 2.
SELECT * from delivery
CREATE Index IX_delivery
ON delivery(DeliveryID ASC, ProductShipped);
-- DROP INDEX IX_store ON delivery;
```

We decided to create an index on the DeliveryID/ProductShipped because there will be a large amount of deliveries and #2. This will be a commonly searched field by users, so it is beneficial to implement a non-clustered index to speed up specific delivery data retrieval.

3 Implementation of the Database

3.1 Tables

TABLE STORE

```
Name                             Null?    Type
-------------------------------- -------- ----
STOREID                          NOT NULL NUMBER
STATE                            NOT NULL VARCHAR2(60)
CITY                             NOT NULL VARCHAR2(255)
ZIPCODE                          NOT NULL NUMBER(10,0)
CONTACT                          NOT NULL VARCHAR2(12)
```

TABLE MEMBER

```
Name                             Null?    Type
-------------------------------- -------- ----
MEMBERID                         NOT NULL NUMBER
FIRSTNAME                        NOT NULL VARCHAR2(60)
LASTNAME                         NOT NULL VARCHAR2(255)
CONTACT                          NOT NULL VARCHAR2(60)
```

TABLE PRODUCT

```
Name                             Null?    Type
-------------------------------- -------- ----
PRODUCTID                        NOT NULL NUMBER
NAME                             NOT NULL VARCHAR2(60)
PRODUCTTYPE                      NOT NULL VARCHAR2(255)
PRODUCTWEIGHT                    NOT NULL NUMBER(12,0)
PRODUCTPRICE                     NOT NULL VARCHAR2(12)
STOREID                          NOT NULL NUMBER
```

TABLE SUPPLIER

```
Name                             Null?    Type
-------------------------------- -------- ----
SUPPLIERID                       NOT NULL NUMBER
SUPPLIERNAME                     NOT NULL VARCHAR2(60)
CONTACT                          NOT NULL VARCHAR2(255)
STOREID                          NOT NULL NUMBER
```

TABLE JOB

```
Name                             Null?    Type
-------------------------------- -------- ----
JOBID                            NOT NULL NUMBER
JOBPOSITION                      NOT NULL VARCHAR2(60)
STARTINGSALARY                   NOT NULL VARCHAR2(12)
JOBLOCATION                      NOT NULL VARCHAR2(100)
STOREID                          NOT NULL NUMBER
```

TABLE NEWS

```
Name                             Null?    Type
-------------------------------- -------- ----
NEWSID                           NOT NULL NUMBER
```

```
NEWSNAME                         NOT NULL VARCHAR2(60)
CONTACT                          NOT NULL VARCHAR2(255)
STOREID                          NOT NULL NUMBER
```

TABLE COMPETITOR

```
Name                             Null?    Type
-------------------------------- -------- ----
COMPETITORID                     NOT NULL NUMBER
NAME                             NOT NULL VARCHAR2(60)
THREATLEVEL                      NOT NULL VARCHAR2(255)
CONTACT                          NOT NULL VARCHAR2(12)
STOREID                          NOT NULL NUMBER
SUPPLIERID                       NOT NULL NUMBER
```

TABLE PROMOTION

```
Name                             Null?    Type
-------------------------------- -------- ----
PROMOTIONID                      NOT NULL NUMBER
PRODUCT                          NOT NULL VARCHAR2(60)
PROMOTIONDETAILS                 NOT NULL VARCHAR2(255)
PRODUCTID                        NOT NULL NUMBER
STOREID                          NOT NULL NUMBER
```

TABLE PARTNERSHIP

```
Name                             Null?    Type
-------------------------------- -------- ----
PARTNERID                        NOT NULL NUMBER
NAME                             NOT NULL VARCHAR2(60)
SERVICETYPE                      NOT NULL VARCHAR2(255)
CONTACT                          NOT NULL VARCHAR2(12)
STOREID                          NOT NULL NUMBER
PROMOTIONID                      NOT NULL NUMBER
```

TABLE DEPARTMENT

```
Name                             Null?    Type
-------------------------------- -------- ----
DEPARTMENTID                     NOT NULL NUMBER
DEPARTMENTNAME                   NOT NULL VARCHAR2(60)
DEPARTMENTMANAGER                NOT NULL VARCHAR2(255)
CONTACT                          NOT NULL VARCHAR2(12)
PRODUCTID                        NOT NULL NUMBER
```

TABLE EMPLOYEE

```
Name                             Null?    Type
-------------------------------- -------- ----
EMPLOYEEID                       NOT NULL NUMBER
FIRSTNAME                        NOT NULL VARCHAR2(60)
LASTNAME                         NOT NULL VARCHAR2(255)
```

```
STARTDATE                        NOT NULL DATE
LENGTHOFEMPLOYMENT               -        VARCHAR2(255)
SALARY                           -        VARCHAR2(20)
CONTACT                          NOT NULL VARCHAR2(12)
DEPARTMENTID                     NOT NULL NUMBER
STOREID                          NOT NULL NUMBER
```

TABLE SPONSOR

```
Name                             Null?    Type
-------------------------------- -------- ----
SPONSORID                        NOT NULL NUMBER
SPONSORNAME                      NOT NULL VARCHAR2(60)
CONTACT                          NOT NULL VARCHAR2(255)
STOREID                          NOT NULL NUMBER
PROMOTIONID                      NOT NULL NUMBER
EMPLOYEEID                       NOT NULL NUMBER
```

TABLE DELIVERY

```
Name                             Null?    Type
-------------------------------- -------- ----
DELIVERYID                       NOT NULL NUMBER
CUSTOMERFIRSTNAME                NOT NULL VARCHAR2(60)
CUSTOMERLASTNAME                 NOT NULL VARCHAR2(255)
PRODUCTSHIPPED                   NOT NULL VARCHAR2(50)
DATESHIPPED                      NOT NULL DATE
PRODUCTID                        NOT NULL NUMBER
MEMBERID                         NOT NULL NUMBER
STOREID                          NOT NULL NUMBER
```

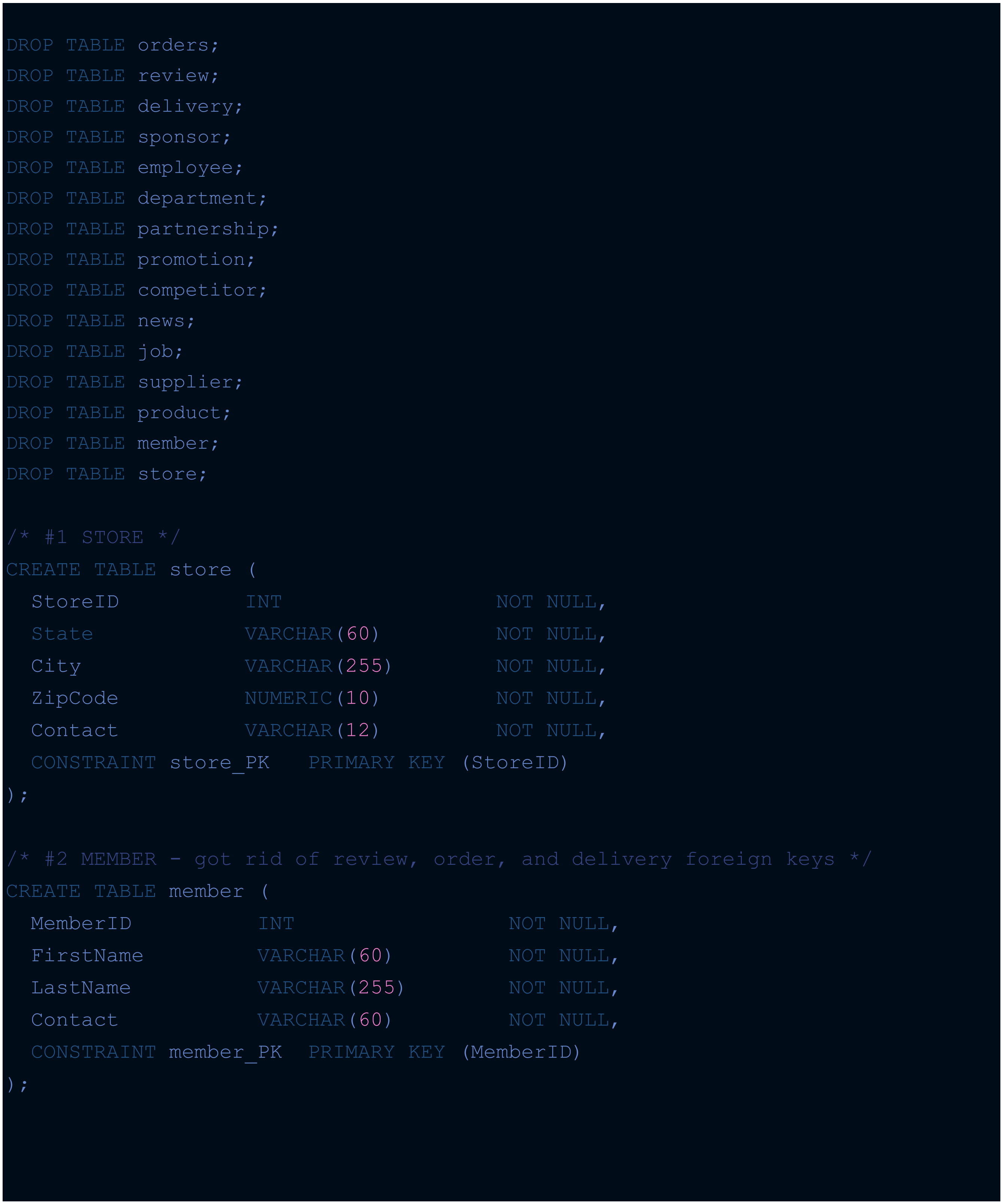
TABLE REVIEW

```
Name                             Null?    Type
-------------------------------- -------- ----
REVIEWID                         NOT NULL NUMBER
REVIEWERNAME                     NOT NULL VARCHAR2(60)
REVIEWINFO                       NOT NULL VARCHAR2(255)
MEMBERID                         NOT NULL NUMBER
PRODUCTID                        NOT NULL NUMBER
STOREID                          NOT NULL NUMBER
```

TABLE ORDERS

```
Name                             Null?    Type
-------------------------------- -------- ----
ORDERID                          NOT NULL NUMBER
CUSTOMERFIRSTNAME                NOT NULL VARCHAR2(60)
CUSTOMERLASTNAME                 NOT NULL VARCHAR2(255)
PRODUCT                          NOT NULL VARCHAR2(50)
ORDERPRICE                       NOT NULL VARCHAR2(12)
```

```
DATEORDERED                          NOT NULL DATE
STOREID                              NOT NULL NUMBER
MEMBERID                             NOT NULL NUMBER
EMPLOYEEID                           NOT NULL NUMBER
```



```
DROP TABLE orders;
DROP TABLE review;
DROP TABLE delivery;
DROP TABLE sponsor;
DROP TABLE employee;
DROP TABLE department;
DROP TABLE partnership;
DROP TABLE promotion;
DROP TABLE competitor;
DROP TABLE news;
DROP TABLE job;
DROP TABLE supplier;
DROP TABLE product;
DROP TABLE member;
DROP TABLE store;

/* #1 STORE */
CREATE TABLE store (
  StoreID           INT                   NOT NULL,
  State             VARCHAR(60)           NOT NULL,
  City              VARCHAR(255)          NOT NULL,
  ZipCode           NUMERIC(10)           NOT NULL,
  Contact           VARCHAR(12)           NOT NULL,
  CONSTRAINT store_PK   PRIMARY KEY (StoreID)
);

/* #2 MEMBER - got rid of review, order, and delivery foreign keys */
CREATE TABLE member (
  MemberID           INT                   NOT NULL,
  FirstName          VARCHAR(60)           NOT NULL,
  LastName           VARCHAR(255)          NOT NULL,
  Contact            VARCHAR(60)           NOT NULL,
  CONSTRAINT member_PK  PRIMARY KEY (MemberID)
);
```

```
/* #3 PRODUCT - got rid of review, department, and promotion foreign keys */
CREATE TABLE product (
  ProductID         INT                   NOT NULL,
  Name             VARCHAR(60)          NOT NULL,
  ProductType       VARCHAR(255)          NOT NULL,
  ProductWeight     NUMERIC(12)           NOT NULL,
  ProductPrice      VARCHAR(12)           NOT NULL,
  StoreID           INT                   NOT NULL,
  CONSTRAINT product_PK   PRIMARY KEY (ProductID),
  CONSTRAINT product_FK4  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #4 SUPPLIER */
CREATE TABLE supplier (
  SupplierID        INT                   NOT NULL,
  SupplierName      VARCHAR(60)           NOT NULL,
  Contact           VARCHAR(255)          NOT NULL,
  StoreID           INT                   NOT NULL,
  CONSTRAINT supplier_PK PRIMARY KEY (SupplierID),
  CONSTRAINT supplier_FK FOREIGN KEY (StoreID) REFERENCES store(StoreID)
);

/* #5 JOB */
CREATE TABLE job (
  JobID             INT                    NOT NULL,
  JobPosition       VARCHAR(60)            NOT NULL,
  StartingSalary    VARCHAR(12)            NOT NULL,
  JobLocation       VARCHAR(100)           NOT NULL,
  StoreID           INT                    NOT NULL,
  CONSTRAINT job_PK  PRIMARY KEY (JobID),
  CONSTRAINT job_FK  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #6 NEWS */
CREATE TABLE news (
  NewsID           INT                   NOT NULL,
  NewsName         VARCHAR(60)           NOT NULL,
  Contact          VARCHAR(255)          NOT NULL,
  StoreID          INT                   NOT NULL,
  CONSTRAINT news_PK  PRIMARY KEY (NewsID),
  CONSTRAINT news_FK  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);
```

```
/* #7 COMPETITOR - got rid of employee foreign key */
CREATE TABLE competitor (
  CompetitorID    INT                   NOT NULL,
  Name            VARCHAR(60)           NOT NULL,
  ThreatLevel     VARCHAR(255)          NOT NULL,
  Contact         VARCHAR(12)           NOT NULL,
  StoreID         INT                   NOT NULL,
  SupplierID      INT                   NOT NULL,
  CONSTRAINT competitor_PK   PRIMARY KEY (CompetitorID),
  CONSTRAINT competitor_FK1  FOREIGN KEY (StoreID)   REFERENCES
store(StoreID),
  CONSTRAINT competitor_FK2  FOREIGN KEY (SupplierID)   REFERENCES
supplier(SupplierID)
);

/* #8 PROMOTION */
CREATE TABLE promotion (
  PromotionID      INT                   NOT NULL,
  Product          VARCHAR(60)           NOT NULL,
  PromotionDetails VARCHAR(255)          NOT NULL,
  ProductID        INT                   NOT NULL,
  StoreID          INT                   NOT NULL,
  CONSTRAINT promotion_PK   PRIMARY KEY (PromotionID),
  CONSTRAINT promotion_FK1  FOREIGN KEY (ProductID)   REFERENCES
product(ProductID),
  CONSTRAINT promotion_FK2  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #9 PARTNERSHIP */
CREATE TABLE partnership (
  PartnerID        INT                     NOT NULL,
  Name             VARCHAR(60)             NOT NULL,
  ServiceType      VARCHAR(255)            NOT NULL,
  Contact          VARCHAR(12)             NOT NULL,
  StoreID          INT                     NOT NULL,
  PromotionID      INT                     NOT NULL,
  CONSTRAINT partnership_PK   PRIMARY KEY (PartnerID),
  CONSTRAINT partnership_FK1  FOREIGN KEY (StoreID)   REFERENCES
store(StoreID),
  CONSTRAINT partnership_FK2  FOREIGN KEY (PromotionID)   REFERENCES
promotion(PromotionID)
);
```

```
/* #10 DEPARTMENT - got rid of employee foreign key */
CREATE TABLE department (
  DepartmentID      INT                 NOT NULL,
  DepartmentName    VARCHAR(60)         NOT NULL,
  DepartmentManager VARCHAR(255)        NOT NULL,
  Contact           VARCHAR(12)         NOT NULL,
  ProductID         INT                 NOT NULL,
  CONSTRAINT department_PK   PRIMARY KEY (DepartmentID),
  CONSTRAINT department_FK2  FOREIGN KEY (ProductID)   REFERENCES
product(ProductID)
);

/* #11 EMPLOYEE */
CREATE TABLE employee (
  EmployeeID         INT                 NOT NULL,
  FirstName          VARCHAR(60)         NOT NULL,
  LastName           VARCHAR(255)        NOT NULL,
  StartDate          DATE                NOT NULL,
  LengthOfEmployment VARCHAR(255),
  Salary             VARCHAR(20),
  Contact            VARCHAR(12)         NOT NULL,
  DepartmentID       INT                 NOT NULL,
  StoreID            INT                 NOT NULL,
  CONSTRAINT employee_PK   PRIMARY KEY (EmployeeID),
  CONSTRAINT employee_FK1  FOREIGN KEY (DepartmentID)   REFERENCES
department(DepartmentID),
  CONSTRAINT employee_FK2  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #12 SPONSOR */
CREATE TABLE sponsor (
  SponsorID        INT                  NOT NULL,
  SponsorName      VARCHAR(60)          NOT NULL,
  Contact          VARCHAR(255)         NOT NULL,
  StoreID          INT                  NOT NULL,
  PromotionID      INT                  NOT NULL,
  EmployeeID       INT                  NOT NULL,
  CONSTRAINT sponsor_PK   PRIMARY KEY (SponsorID),
  CONSTRAINT sponsor_FK1  FOREIGN KEY (StoreID)   REFERENCES store(StoreID),
  CONSTRAINT sponsor_FK2  FOREIGN KEY (PromotionID)   REFERENCES
promotion(PromotionID),
  CONSTRAINT sponsor_FK3  FOREIGN KEY (EmployeeID)   REFERENCES
employee(EmployeeID)
);

/* #13 DELIVERY */
CREATE TABLE delivery (
  DeliveryID         INT                   NOT NULL,
  CustomerFirstName  VARCHAR(60)           NOT NULL,
  CustomerLastName   VARCHAR(255)          NOT NULL,
  ProductShipped     VARCHAR(50)           NOT NULL,
  DateShipped        DATE                  NOT NULL,
  ProductID          INT                   NOT NULL,
  MemberID           INT                   NOT NULL,
  StoreID            INT                   NOT NULL,
  CONSTRAINT delivery_PK   PRIMARY KEY (DeliveryID),
  CONSTRAINT delivery_FK1  FOREIGN KEY (ProductID)   REFERENCES
product(ProductID),
  CONSTRAINT delivery_FK2  FOREIGN KEY (MemberID)   REFERENCES
member(MemberID),
  CONSTRAINT delivery_FK3  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #14 REVIEWS */
CREATE TABLE review (
  ReviewID          INT                    NOT NULL,
  ReviewerName      VARCHAR(60)            NOT NULL,
  ReviewInfo        VARCHAR(255)           NOT NULL,
  MemberID          INT                    NOT NULL,
  ProductID         INT                    NOT NULL,
  StoreID           INT                    NOT NULL,
  CONSTRAINT review_PK   PRIMARY KEY (ReviewID),
  CONSTRAINT review_FK1  FOREIGN KEY (MemberID)   REFERENCES member(MemberID),
  CONSTRAINT review_FK2  FOREIGN KEY (ProductID)   REFERENCES
product(ProductID),
  CONSTRAINT review_FK3  FOREIGN KEY (StoreID)   REFERENCES store(StoreID)
);

/* #15 ORDERs  - changed table name from "order" to orders*/
CREATE TABLE orders (
  OrderID                INT                     NOT NULL,
  CustomerFirstName      VARCHAR(60)             NOT NULL,
  CustomerLastName       VARCHAR(255)            NOT NULL,
  Product                VARCHAR(50)             NOT NULL,
  OrderPrice             VARCHAR(12)            NOT NULL,
  DateOrdered            DATE                   NOT NULL,
  StoreID                INT                    NOT NULL,
  MemberID               INT                    NOT NULL,
  EmployeeID             INT                    NOT NULL,
  CONSTRAINT order_PK   PRIMARY KEY (OrderID),
  CONSTRAINT order_FK1  FOREIGN KEY (StoreID)   REFERENCES store(StoreID),
  CONSTRAINT order_FK2  FOREIGN KEY (MemberID)   REFERENCES member(MemberID),
  CONSTRAINT order_FK3  FOREIGN KEY (EmployeeID)   REFERENCES
employee(EmployeeID)
);

-- INDEX's --
-- 1.
SELECT * from store
CREATE Index IX_store
ON store (StoreID ASC);
-- DROP INDEX IX_store ON store;
-- 2.
SELECT * from store
CREATE Index IX_store
ON store (StoreID ASC);
-- DROP INDEX IX_store ON store;

/* #1 STORE */
insert into store  values (1, 'Kentucky', 'Lexington', 40596, '859-883-0953');
insert into store  values (2, 'Hawaii', 'Honolulu', 96850, '808-717-2849');
insert into store  values (3, 'Georgia', 'Augusta', 30919, '706-661-6928');
insert into store  values (4, 'California', 'Fresno', 93786, '559-949-9938');
insert into store  values (5, 'Georgia', 'Atlanta', 31106, '404-364-0604');
insert into store  values (6, 'North Dakota', 'Grand Forks', 58207, '701-924-
0551');
insert into store  values (7, 'West Virginia', 'Charleston', 25356, '304-115-
8167');
insert into store  values (8, 'Nevada', 'Reno', 89595, '775-334-1104');
insert into store  values (9, 'California', 'San Mateo', 94405, '805-533-
8186');
insert into store  values (10, 'Texas', 'Katy', 77493, '281-417-6443');

/* #2 MEMBER */
insert into member (MemberID, FirstName, LastName, Contact) values (1,
'Janine', 'Stuffins', 'jstuffins0@kickstarter.com');
```

```
insert into member (MemberID, FirstName, LastName, Contact) values (2, 'Kora',
'Spadoni', 'kspadoni1@themeforest.net');
insert into member (MemberID, FirstName, LastName, Contact) values (3, 'Esra',
'Slay', 'eslay2@miitbeian.gov.cn');
insert into member (MemberID, FirstName, LastName, Contact) values (4, 'Cece',
'Allday', 'callday3@nationalgeographic.com');
insert into member (MemberID, FirstName, LastName, Contact) values (5,
'Zechariah', 'Talboy', 'ztalboy4@ebay.co.uk');
insert into member (MemberID, FirstName, LastName, Contact) values (6,
'Edithe', 'Pinsent', 'epinsent5@ifeng.com');
insert into member (MemberID, FirstName, LastName, Contact) values (7,
'Selie', 'MacCurley', 'smaccurley6@opensource.org');
insert into member (MemberID, FirstName, LastName, Contact) values (8,
'Fredi', 'Hamby', 'fhamby7@fda.gov');
insert into member (MemberID, FirstName, LastName, Contact) values (9,
'Heddie', 'Zelley', 'hzelley8@squidoo.com');
insert into member (MemberID, FirstName, LastName, Contact) values (10, 'Bil',
'Howard - Gater', 'bhowardgater9@freewebs.com');

/* #3 PRODUCT */
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (1, 'Lemonade - Natural, 591 Ml', 'Shallots',
4.29, '$59.55', 6);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (2, 'Numi - Assorted Teas', 'Anisette -
Mcguiness', 1.46, '$47.54', 7);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (3, 'Cookie Dough - Oatmeal Rasin', 'Soup -
Campbells Beef Stew', 1.28, '$20.19', 5);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (4, 'Sauce - Salsa', 'Muffin Hinge - 211n', 3.6,
'$39.87', 10);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (5, 'Carbonated Water - Raspberry', 'Coffee
Beans - Chocolate', 2.92, '$2.51', 9);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (6, 'Wine - Muscadet Sur Lie', 'Basil - Dry,
Rubbed', 8.65, '$50.88', 9);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (7, 'Ice - Clear, 300 Lb For Carving',
'Plasticspoonblack', 3.51, '$52.08', 1);
```

```
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (8, 'Apricots Fresh', 'Bagel - Sesame Seed
Presliced', 2.28, '$18.14', 2);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (9, 'Tarragon - Primerba, Paste', 'Blue Curacao
- Marie Brizard', 1.79, '$9.66', 8);
insert into product (ProductID, Name, ProductType, ProductWeight,
ProductPrice, StoreID) values (10, 'Bread - Frozen Basket Variety',
'Thermometer Digital', 2.63, '$5.50', 1);

/* #4 SUPPLIER */
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (1,
'Mills, Kutch and Graham', 'szoane0@smugmug.com', 10);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (2,
'Kris-Parker', 'mcordelle1@yellowpages.com', 9);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (3,
'Block-O''Reilly', 'garrighini2@bbc.co.uk', 2);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (4,
'Schinner and Sons', 'dposen3@dedecms.com', 6);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (5,
'Bechtelar-Cassin', 'dbyatt4@histats.com', 7);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (6,
'Dietrich and Sons', 'vwildt5@123-reg.co.uk', 8);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (7,
'Rice and Sons', 'rstqueintain6@nyu.edu', 8);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (8,
'Johnson-Langworth', 'aholme7@histats.com', 1);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (9,
'Goodwin LLC', 'mcuttles8@vimeo.com', 6);
insert into supplier (SupplierID, SupplierName, Contact, StoreID) values (10,
'Murphy Group', 'breihill9@businesswire.com', 4);

/* #5 JOB */
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (1, 'Systems Administrator I', '$15236.74', 'District of Columbia', 6);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (2, 'Design Engineer', '$16499.28', 'New York', 9);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (3, 'Recruiter', '$15683.35', 'California', 7);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (4, 'Teacher', '$12101.69', 'West Virginia', 8);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (5, 'Internal Auditor', '$13811.23', 'Arizona', 7);
```

```
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (6, 'Marketing Assistant', '$17027.40', 'Utah', 1);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (7, 'Geological Engineer', '$15422.95', 'Nevada', 3);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (8, 'Librarian', '$19522.32', 'Florida', 8);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (9, 'Automation Specialist III', '$18978.10', 'Florida', 10);
insert into job (JobID, JobPosition, StartingSalary, JobLocation, StoreID)
values (10, 'Senior Editor', '$13371.46', 'Ohio', 1);

/* #6 NEWS */
insert into news (NewsID, NewsName, Contact, StoreID) values (1, 'Stanton and
Sons', 'hpennigar0@adobe.com', 8);
insert into news (NewsID, NewsName, Contact, StoreID) values (2, 'Goyette
Group', 'conolan1@creativecommons.org', 8);
insert into news (NewsID, NewsName, Contact, StoreID) values (3, 'Pagac-Mann',
'bbattle2@etsy.com', 1);
insert into news (NewsID, NewsName, Contact, StoreID) values (4, 'Beatty-
Lang', 'ktruelock3@spotify.com', 10);
insert into news (NewsID, NewsName, Contact, StoreID) values (5, 'Hermann-
Bartoletti', 'gching4@nationalgeographic.com', 7);
insert into news (NewsID, NewsName, Contact, StoreID) values (6, 'Homenick-
Considine', 'jsurr5@nasa.gov', 4);
insert into news (NewsID, NewsName, Contact, StoreID) values (7, 'Waters and
Sons', 'acowlam6@etsy.com', 1);
insert into news (NewsID, NewsName, Contact, StoreID) values (8, 'Zemlak-
Tremblay', 'kellen7@cnbc.com', 8);
insert into news (NewsID, NewsName, Contact, StoreID) values (9, 'Medhurst-
Huel', 'kbriamo8@apache.org', 7);
insert into news (NewsID, NewsName, Contact, StoreID) values (10, 'Baumbach
and Sons', 'drobers9@blinklist.com', 6);

/* #7 COMPETITOR */
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID,
SupplierID) values (1, 'Klein, Rowe and Hoppe', 'High', '496-394-7933', 7, 2);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID,
SupplierID) values (2, 'Barrows-Bogisich', 'High', '961-528-5267', 8, 8);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID,
SupplierID) values (3, 'Effertz-O''Kon', 'High', '601-773-2713', 6, 6);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID,
SupplierID) values (4, 'Cole, Stanton and Stracke', 'Medium', '630-354-3243',
2, 6);
```

```
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (5, 'Keebler, Bosco and Marquardt', 'Low', '205-994-5994', 6, 6);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (6, 'Terry, Weissnat and Parisian', 'Medium', '559-970-7281', 9, 6);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (7, 'Crooks, Cruickshank and Jaskolski', 'Medium', '656-461-6154', 7, 4);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (8, 'Shanahan and Sons', 'Low', '105-658-6188', 7, 3);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (9, 'Morar, Willms and Funk', 'Medium', '610-390-9213', 1, 3);
insert into competitor (CompetitorID, Name, ThreatLevel, Contact, StoreID, SupplierID) values (10, 'Reichert-Wolff', 'Low', '243-634-3093', 5, 1);

/* #8 PROMOTION */
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (1, 'Bread - White Epi Baguette', 'egestas metus aenean fermentum donec ut mauris eget massa tempor convallis nulla neque libero convallis eget eleifend', 4, 2);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (2, 'Split Peas - Green, Dry', 'platea dictumst maecenas ut massa quis augue luctus tincidunt nulla mollis molestie lorem quisque ut erat curabitur gravida nisi at', 9, 5);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (3, 'Chocolate - Dark Callets', 'a nibh in quis justo maecenas rhoncus aliquam lacus morbi quis tortor id', 9, 9);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (4, 'Lamb - Loin Chops', 'urna ut tellus nulla ut erat id mauris vulputate elementum nullam varius nulla facilisi cras non velit nec nisi', 4, 10);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (5, 'Appetizer - Spring Roll, Veg', 'nisl duis bibendum felis sed interdum venenatis turpis enim blandit mi in porttitor pede', 8, 4);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (6, 'Squash - Pepper', 'enim leo rhoncus sed vestibulum sit amet cursus id turpis integer aliquet massa', 7, 2);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID, StoreID) values (7, 'Veal - Loin', 'erat id mauris vulputate elementum nullam varius nulla facilisi cras non velit nec nisi vulputate nonummy maecenas', 1, 4);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID,
StoreID) values (8, 'Jameson Irish Whiskey', 'ac nibh fusce lacus purus
aliquet at feugiat non pretium quis lectus suspendisse potenti', 2, 7);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID,
StoreID) values (9, 'Wine - Duboeuf Beaujolais', 'interdum mauris non ligula
pellentesque ultrices phasellus id sapien in sapien iaculis congue vivamus
metus arcu adipiscing molestie hendrerit at', 10, 2);
insert into promotion (PromotionID, Product, PromotionDetails, ProductID,
StoreID) values (10, 'Mix - Cocktail Strawberry Daiquiri', 'velit vivamus vel
nulla eget eros elementum pellentesque quisque porta volutpat erat quisque
erat eros viverra', 9, 10);

/* #9 PARTNERSHIP */
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (1, 'Chef Hat 20cm', 'Electronics', '251-692-4014', 6, 8);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (2, 'Crab - Imitation Flakes', 'Kids', '477-846-1386', 10,
9);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (3, 'French Pastry - Mini Chocolate', 'Kids', '905-694-
4837', 8, 9);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (4, 'Butter - Unsalted', 'Kids', '679-452-1946', 1, 3);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (5, 'Cookie Dough - Peanut Butter', 'Music', '951-821-
4220', 7, 2);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (6, 'Beets - Golden', 'Tools', '336-767-3723', 9, 8);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (7, 'Russian Prince', 'Sports', '364-222-1167', 1, 5);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (8, 'Pea - Snow', 'Tools', '970-857-8226', 5, 9);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (9, 'Shrimp - Prawn', 'Electronics', '233-660-3239', 2,
1);
insert into partnership (PartnerID, Name, ServiceType, Contact, StoreID,
PromotionID) values (10, 'Blueberries - Frozen', 'Kids', '518-704-0965', 2,
5);

/* #10 DEPARTMENT */
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (1, 'Accounting', 'Faith Nice', '639-426-2970', 6);
```

```
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (2, 'Business Development', 'Renata Briscam', '143-
423-9771', 8);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (3, 'Business Development', 'Cully Ragat', '234-
405-9460', 6);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (4, 'Support', 'Pam Girodin', '378-493-0409', 10);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (5, 'Accounting', 'Laird Ovitts', '617-134-2385',
8);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (6, 'Legal', 'Raleigh Mouser', '835-877-6257', 8);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (7, 'Marketing', 'Orton Jeanneau', '732-862-3454',
6);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (8, 'Accounting', 'Dexter Gabbot', '808-159-5956',
4);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (9, 'Business Development', 'Norris Girodon', '181-
280-2530', 9);
insert into department (DepartmentID, DepartmentName, DepartmentManager,
Contact, ProductID) values (10, 'Human Resources', 'Derek Vatini', '534-729-
1772', 5);

/* #11 EMPLOYEE */
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (1, 'Arel',
'Bullas', '2021-03-20', 249, '$43.69', '957-804-7067', 4, 7);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (2,
'Tibold', 'Benes', '2021-11-02', 236, '$54.73', '848-753-9100', 2, 3);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (3,
'Lorraine', 'Bartell', '2020-11-27', 1222, '$37.19', '597-756-5146', 2, 10);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (4,
'Eolande', 'Carnalan', '2021-11-13', 1682, '$47.81', '899-245-9142', 6, 5);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (5,
'Jefferson', 'Christoforou', '2021-05-05', 2, '$49.46', '631-610-9111', 5, 9);
```

```
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (6,
'Gabbi', 'Rippingall', '2021-10-25', 1225, '$51.50', '580-382-8363', 10, 6);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (7,
'Natty', 'Barens', '2020-12-03', 975, '$40.46', '624-652-3953', 2, 10);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (8,
'Hailey', 'Keward', '2021-06-18', 1085, '$32.38', '922-206-1949', 9, 7);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (9,
'Frederico', 'Vasey', '2021-08-04', 111, '$45.33', '732-258-7524', 9, 9);
insert into employee (EmployeeID, FirstName, LastName, StartDate,
LengthOfEmployment, Salary, Contact, DepartmentID, StoreID) values (10,
'Cassey', 'Lapere', '2021-10-06', 218, '$12.02', '847-912-9996', 5, 8);

/* #12 SPONSOR */
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (1, 'Ondricka-Corkery', 'rpolhill0@npr.org', 4, 3, 1);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (2, 'Lynch-Klein', 'rwixey1@wunderground.com', 5, 5, 8);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (3, 'Dare-West', 'smorshead2@netscape.com', 9, 5, 5);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (4, 'Paucek Inc', 'dmonsey3@topsy.com', 9, 4, 8);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (5, 'Funk-Boyer', 'agherarducci4@xinhuanet.com', 9, 5, 9);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (6, 'Gislason, Kohler and Wilderman',
'rketchen5@ycombinator.com', 6, 8, 10);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (7, 'Hodkiewicz, Gleichner and Hauck',
'aleaney6@tumblr.com', 6, 3, 3);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (8, 'Huel-Cremin', 'esimmonite7@google.com.au', 4, 9, 4);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (9, 'Tromp-Hoppe', 'asleford8@hugedomains.com', 9, 9, 10);
insert into sponsor (SponsorID, SponsorName, Contact, StoreID, PromotionID,
EmployeeID) values (10, 'Heidenreich-Wyman', 'jdavaux9@patch.com', 7, 10, 2);

/* #13 DELIVERY */
```

```
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (1,
'Lynett', 'Dillicate', 'Savory', '2020-12-19', 3, 6, 1);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (2,
'Jacquenetta', 'Hurdwell', 'Cucumber - English', '2021-06-16', 1, 10, 1);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (3, 'Nerta',
'Wyllcock', 'Flower - Daisies', '2021-03-07', 10, 2, 7);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (4, 'Kelli',
'Feldmark', 'Basil - Fresh', '2021-02-11', 4, 3, 5);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (5, 'Emmit',
'Bloor', 'Dome Lid Clear P92008h', '2021-08-14', 7, 6, 3);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (6, 'Mag',
'Learned', 'Cheese - Blue', '2021-01-11', 10, 10, 6);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (7, 'Barde',
'Eberts', 'Caviar - Salmon', '2021-05-08', 3, 2, 10);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (8,
'Bartlet', 'Truswell', 'Napkin White - Starched', '2021-05-30', 3, 1, 8);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (9,
'Blayne', 'Franseco', 'Potatoes - Pei 10 Oz', '2021-02-13', 6, 8, 5);
insert into delivery (DeliveryID, CustomerFirstName, CustomerLastName,
ProductShipped, DateShipped, ProductID, MemberID, StoreID) values (10, 'Ive',
'Sibery', 'Sole - Dover, Whole, Fresh', '2021-03-29', 10, 3, 8);

/* #14 REVIEWS */
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (1, 'Carly Teeney', 'natoque penatibus et magnis dis
parturient montes nascetur ridiculus mus vivamus vestibulum sagittis sapien
cum sociis natoque penatibus', 7, 7, 7);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (2, 'Bethena Becks', 'nascetur ridiculus mus vivamus
vestibulum sagittis sapien cum sociis natoque', 4, 7, 5);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (3, 'Goraud Noweak', 'sem mauris laoreet ut rhoncus aliquet
pulvinar sed nisl nunc rhoncus dui vel sem sed sagittis nam congue risus
semper', 3, 2, 9);
```

```
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (4, 'Geoff Millis', 'id lobortis convallis tortor risus
dapibus augue vel accumsan tellus', 1, 4, 8);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (5, 'Maurene Thomesson', 'amet nunc viverra dapibus nulla
suscipit ligula in lacus curabitur at ipsum ac', 3, 5, 10);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (6, 'Opal Merwede', 'elit sodales scelerisque mauris sit amet
eros suspendisse accumsan tortor quis turpis sed ante vivamus tortor duis', 6,
9, 7);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (7, 'Liz Camerello', 'turpis nec euismod scelerisque quam
turpis adipiscing lorem vitae mattis nibh ligula nec sem duis', 5, 3, 6);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (8, 'Ilaire Lanyon', 'diam vitae quam suspendisse potenti
nullam porttitor lacus at turpis donec posuere metus vitae ipsum', 1, 8, 1);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (9, 'Waite Curtain', 'id turpis integer aliquet massa id
lobortis convallis tortor risus dapibus augue vel', 6, 8, 8);
insert into review (ReviewID, ReviewerName, ReviewInfo, MemberID, ProductID,
StoreID) values (10, 'Errol Checchetelli', 'in hac habitasse platea dictumst
etiam faucibus cursus urna ut tellus nulla ut erat id mauris vulputate', 6, 6,
5);

/* #15 ORDERS */
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (1,
'Westleigh', 'Vallack', 'Tomatoes', '$491.18', '2021-04-27', 9, 8, 10);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (2, 'Augustus',
'Burdon', 'Lettuce - Mini Greens, Whole', '$214.17', '2020-11-18', 6, 6, 1);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (3, 'Connie',
'Lardge', 'Wine - Balbach Riverside', '$454.03', '2021-02-24', 1, 10, 5);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (4, 'Amory',
'Lewington', 'Wine - Red, Gamay Noir', '$113.97', '2021-07-16', 4, 3, 3);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (5, 'Rica',
'Gentsch', 'Soup Campbells', '$210.10', '2021-05-25', 6, 2, 5);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (6, 'Greggory',
'Abate', 'Maintenance Removal Charge', '$125.10', '2021-10-09', 10, 3, 9);
```

```
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (7, 'Jessalyn',
'Pindar', 'Icecream Cone - Areo Chocolate', '$216.88', '2021-11-14', 1, 9, 3);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (8, 'Jaymee',
'Loton', 'Longos - Lasagna Beef', '$48.71', '2020-12-03', 4, 9, 1);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (9, 'Josepha',
'Phelan', 'Wine - Touraine Azay - Le - Rideau', '$492.36', '2020-12-16', 7, 7,
3);
insert into orders (OrderID, CustomerFirstName, CustomerLastName, Product,
OrderPrice, DateOrdered, StoreID, MemberID, EmployeeID) values (10, 'Xenos',
'Cicutto', 'Vermacelli - Sprinkles, Assorted', '$487.09', '2021-03-07', 8, 7,
4);
```

3.2 Index

```
-- 1.
--------------------------------------------------------------------
SELECT * from product
CREATE Index IX_product
ON product(ProductID ASC, ProductName);
-- DROP INDEX IX_product ON product;

-- 2.
--------------------------------------------------------------------
SELECT * from delivery
CREATE Index IX_delivery
ON delivery(DeliveryID ASC, ProductShipped);
-- DROP INDEX IX_product ON delivery;
```

4 Conclusion

In completing our project, we wanted to construct a database that gave a comprehensive insight into the inner workings of Fresh Thyme and their business enterprise. The project began with our conceptual design, where we listed every important aspect of Fresh Thyme's business operations and detailed visually how they all connect to each other. We further fleshed this out in our data dictionary, where we listed all of the corresponding attributes for each entity type. These attributes all had special characteristics, where we listed their data type, examples of valid values, and if any null values would be allowed. The data dictionary was used in preparation for our logical design, where we used normalization to remove redundancy in our tables and showed how each of our entities inter-related to each other. Our final step was coding our diagrams into SQL, where we physically created each of our planned tables and included their associated type and null value condition. We also inputted the relevant constraints and references to relevant tables, detailed in our logical design schema.

Like most detailed database projects, we made a few mistakes related to formatting and detailing inter-entity relationships. The conceptual and logical design processes required a few different drafts, as ensuring that each respective entity referenced the correct foreign key meant having to construct detailed overviews that zoomed in on each individual connection within the database. This could often be confusing. The normalization process of removing transitive dependencies also required editing of our original drafts, but resulted in a cleaner and less repetitive diagram. And often with SQL, getting the code to run took a few tries to achieve successfully.

This process tested our critical thinking, SQL, and creative analysis skills as it required construction, visualization, coding, and an overall understanding of how database design works to fulfill each individual part. It was very neat getting to simulate a company's relational and attributional qualities and then seeing them play off of each other in real time when building the code in SQL. We learned how databases are just a sum of their component parts, parts that all interlink in ways that are not always clear. But our project showed that through breaking down the process into those parts, and then placing each part in its respective place within the database, you can create large and adaptable databases that can fulfill the many needs of a complex business enterprise.