Pho Hoa Noodle Soup

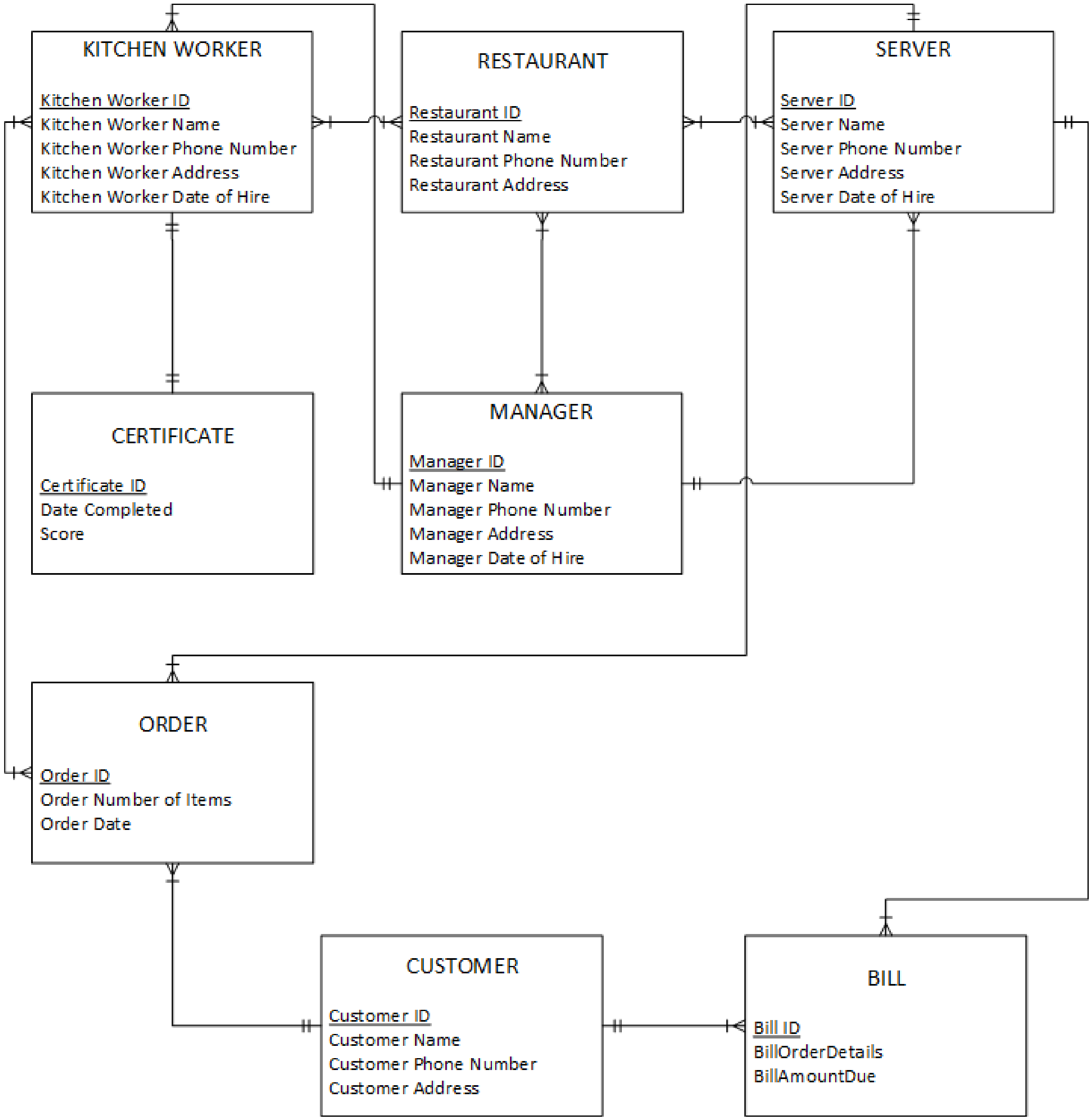
Pho Hoa Noodle Soup is a franchise that produces and supplies Vietnamese Pho noodle soup that was established over thirty-five years ago. The company has many employees and many different restaurant locations all over the world. There are around six locations around the San Jose and Santa Clara area itself. The company is quite profitable, considering that it is a big franchise that has been operating for a decent amount of time. It has many locations in the Bay Area, which is a place where pho is quite popular and consumed often. Since we are now in the fast-paced and technological era, Pho Hoa needs to do something about its business to keep up with this culture. Due to the fact that the company has been operating for awhile now and has grown so greatly, it believes that having a database would help the company be more organized, provide more useful information to the company, and help the company to continue to grow successfully. It would be easier for the company to access, sort through, and analyze certain sales, information, and trends about its company through using a database.

The company has multiple restaurant locations around the globe. Each restaurant has a name, phone number, and an address. Each restaurant has kitchen workers, managers, and servers. Since some restaurants may be a bit understaffed, kitchen workers, managers, and servers can all work at multiple restaurants. Kitchen workers have a kitchen worker ID, name, phone number, address, and date of hire. The kitchen workers must complete an online food handlers certification course in order to prepare the orders for the restaurant. After doing so, they receive a certification that has the certificate number, date completed, and score. Each kitchen worker has only one certificate and each certificate belongs to a single worker. Servers have a server ID, name, phone number, address, and date of hire. The main job of a server is to take the orders placed by the customers. Servers can take multiple orders, but each order can belong to just one server. Kitchen workers can prepare many orders, and multiple kitchen workers can work on one particular order. Managers have a manager ID, name, phone number, address, and date of hire. Managers oversee multiple servers and multiple kitchen workers, but kitchen workers and servers have just one manager.

As mentioned earlier, when customers enter the store, they must place an order. The database will contain information regarding these orders. The order will contain an Order ID, total number of items in the order, and an order date. Customers have a name, phone number, and address. A customer can place multiple orders, but each order can belong to only one specific customer.

Each customer, of course, has a bill they must pay when they dine in or take out food from the restaurant. The bill has a unique bill number and indicates the details of the order and the amount due from the customer. The bill is paid directly to the server. Each bill can belong only to one customer and only to one server. Each server can have multiple bills, and each customer can have multiple bills, though.

ER Model



KITCHEN WORKER

Kitchen Worker ID
Kitchen Worker Name
Kitchen Worker Phone Number
Kitchen Worker Address
Kitchen Worker Date of Hire

RESTAURANT

Restaurant ID
Restaurant Name
Restaurant Phone Number
Restaurant Address

SERVER

Server ID
Server Name
Server Phone Number
Server Address
Server Date of Hire

CERTIFICATE

Certificate ID
Date Completed
Score

MANAGER

Manager ID
Manager Name
Manager Phone Number
Manager Address
Manager Date of Hire

ORDER

Order ID
Order Number of Items
Order Date

CUSTOMER

Customer ID
Customer Name
Customer Phone Number
Customer Address

BILL

Bill ID
BillOrderDetails
BillAmountDue

Pho Hoa Noodle Soup Data Dictionary

CUSTOMER

Name	Data Type	Constraints	Key	Description	Example value
Customer ID	integer	>0	PK	Unique identifier for a customer	1
Customer Name	alphanumeric			First and last name of a customer	Vivian Nguyen
Customer Phone Number	integer			Mobile phone number of a customer	40889239481
Customer Address	alphanumeric			Home address of a customer	123 Candyland Lane.

MANAGER

Name	Data Type	Constraints	Key	Description	Example value
Manager ID	integer	>0	PK	Unique identifier for a manager	1
Manager Name	alphanumeric			First and last name of a manager	Vivian Nguyen
Manager Phone Number	integer			Mobile phone number of a manager	40889239481
Manager Address	alphanumeric			Home city of a manager	San Jose
Manager Date of Hire	date			The date a manager was hired	12/20/2017

RESTAURANT

Name	Data Type	Constraints	Key	Description	Example value
Restaurant ID	integer	>0	PK	Unique identifier for a restaurant	1
Restaurant Name	alphanumeric			Name of the branch of a restaurant	Mckee
Restaurant Phone Number	integer			Phone number of a restaurant	40889239481
Restaurant Address	alphanumeric			City of a restaurant	San Jose

KITCHEN WORKER

Name	Data Type	Constraints	Key	Description	Example value
Kitchen Worker ID	integer	>0	PK	Unique identifier for a kitchen worker	1
Kitchen Worker Name	alphanumeric			First and last name of a kitchen worker	Vivian Nguyen
Kitchen Worker Phone Number	integer			Mobile phone number of a kitchen worker	40889239481
Kitchen Worker Address	alphanumeric			Home city of a kitchen worker	San Jose
Kitchen Worker Date of Hire	date			The date a kitchen worker was hired	12/20/2017
Manager ID	integer	>0	FK	The manager of the kitchen worker; unique identifier for a manager	2

SERVER

Name	Data Type	Constraints	Key	Description	Example value
Server ID	integer	>0	PK	Unique identifier for a server	1
Server Name	alphanumeric			First and last name of a server	Vivian Nguyen
Server Phone Number	integer			Mobile phone number of a server	40889239481
Server Address	alphanumeric			Home city of a server	San Jose
Server Date of Hire	date			The date a server was hired	12/20/2017
Manager ID	integer	>0	FK	The manager of the server; unique identifier for a manager	2

RESTAURANT MANAGER

Name	Data Type	Constraints	Key	Description	Example value
Restaurant ID	integer	>0	PK, FK	Restaurant a manager works at; unique identifier for a restaurant	1
Manager ID	integer	>0	PK, FK	Manager a restaurant has; unique identifier for a manager	2

RESTAURANT KITCHEN

Name	Data Type	Constraints	Key	Description	Example value
Restaurant ID	integer	>0	PK, FK	Restaurant a kitchen worker works at; unique identifier for a restaurant	1
Kitchen Worker ID	integer	>0	PK, FK	Kitchen worker a restaurant has; unique identifier for a kitchen worker	2

RESTAURANT SERVER

Name	Data Type	Constraints	Key	Description	Example value
Restaurant ID	integer	>0	PK, FK	Restaurant a server works at; unique identifier for a restaurant	1
Server ID	integer	>0	PK, FK	Server a restaurant has; unique identifier for a server	2

ORDER KITCHEN

Name	Data Type	Constraints	Key	Description	Example value
Order ID	integer	>0	PK, FK	Order a kitchen worker works on; unique identifier for an order	1
Kitchen Worker ID	integer	>0	PK, FK	Kitchen worker an order has; unique identifier for a kitchen worker	2

CERTIFICATE

Name	Data Type	Constraints	Key	Description	Example value
Certificate ID	integer	>0	PK	Unique identifier for a certificate	1
Date Completed	date			The date the course was completed	12/20/2017
Score	integer			Score received on course	90
Kitchen Worker ID	integer	>0	FK	The kitchen worker that received a certificate; unique identifier for a kitchen worker	2

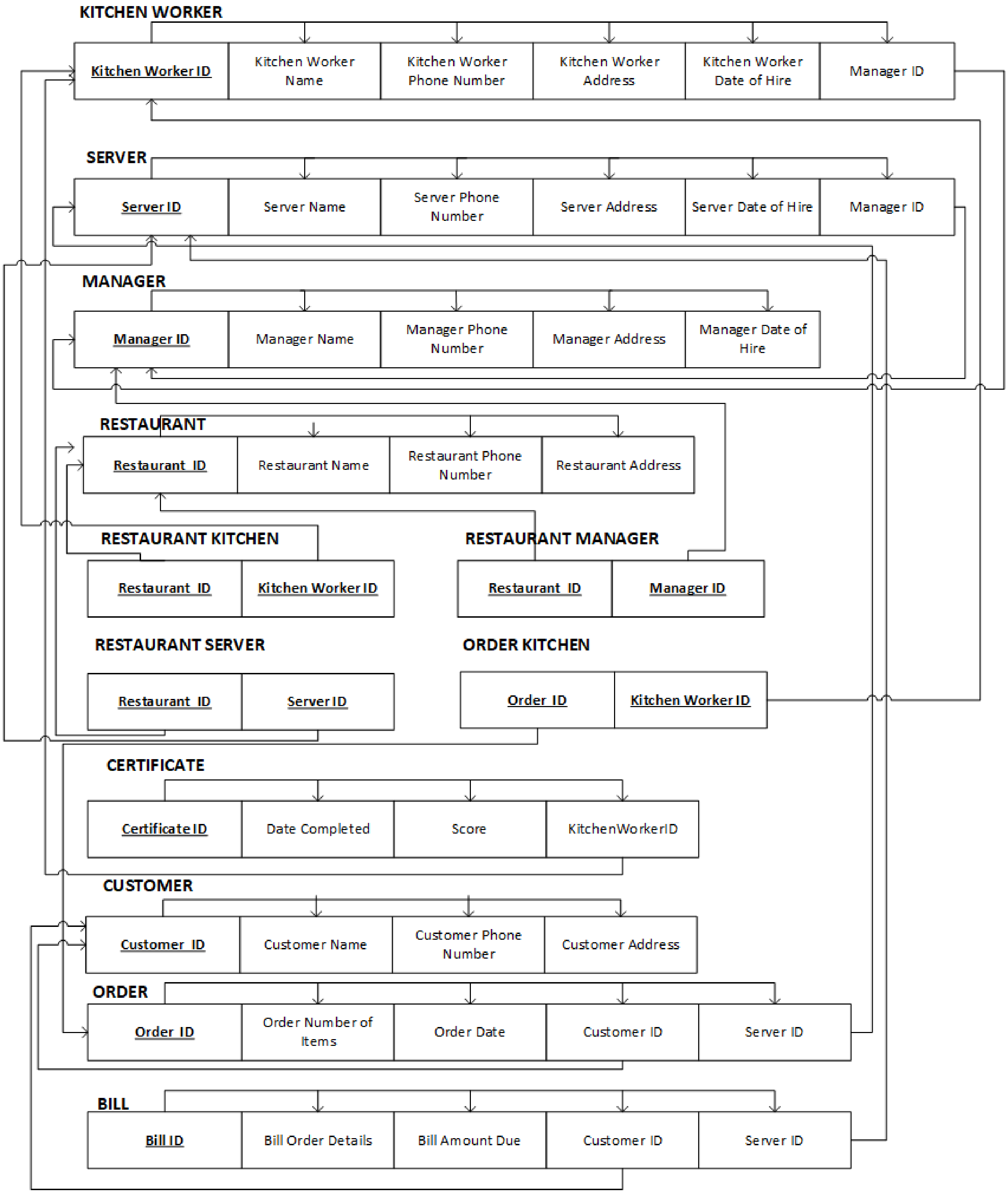
ORDER

Name	Data Type	Constraints	Key	Description	Example value
Order ID	integer	>0	PK	Unique identifier for an order	1
Order Number of Items	integer			Number of items in an order	5
Order Date	date			The date the order was placed	12/20/2017
Customer ID	integer	>0	FK	The customer that placed the order; unique identifier for a customer	3
Server ID	integer	>0	FK	The server that took the order; unique identifier for a server	2

BILL

Name	Data Type	Constraints	Key	Description	Example value
Bill ID	integer	>0	PK	Unique identifier for a bill	1
Bill Order Details	alphanumeric			The number of items on the bill	5 items
Bill Amount Due	decimal(11,2)			The total amount due on the bill	10.69
Customer ID	integer	>0	FK	The customer that paid the bill; unique identifier for a customer	3
Server ID	integer	>0	FK	The server that took care of the bill; unique identifier for a server	2

Relational Data Model



KITCHEN WORKER

Kitchen Worker ID	Kitchen Worker Name	Kitchen Worker Phone Number	Kitchen Worker Address	Kitchen Worker Date of Hire	Manager ID

SERVER

Server ID	Server Name	Server Phone Number	Server Address	Server Date of Hire	Manager ID

MANAGER

Manager ID	Manager Name	Manager Phone Number	Manager Address	Manager Date of Hire

RESTAURANT

Restaurant ID	Restaurant Name	Restaurant Phone Number	Restaurant Address

RESTAURANT KITCHEN

Restaurant ID	Kitchen Worker ID

RESTAURANT MANAGER

Restaurant ID	Manager ID

RESTAURANT SERVER

Restaurant ID	Server ID

ORDER KITCHEN

Order ID	Kitchen Worker ID

CERTIFICATE

Certificate ID	Date Completed	Score	KitchenWorkerID

CUSTOMER

Customer ID	Customer Name	Customer Phone Number	Customer Address

ORDER

Order ID	Order Number of Items	Order Date	Customer ID	Server ID

BILL

Bill ID	Bill Order Details	Bill Amount Due	Customer ID	Server ID

SQL Queries to Create Each Table

```
CREATE TABLE Restaurant_T
        (RestaurantID              NUMERIC (11,0) NOT NULL,
        RestaurantName            VARCHAR(25) NOT NULL,
        RestaurantPhoneNumber            NUMERIC(10, 0),
        RestaurantAddress         VARCHAR(30),
CONSTRAINT Restaurant_PK PRIMARY KEY (RestaurantID));

CREATE TABLE Manager_T
        (ManagerID                NUMERIC(11,0) NOT NULL,
        ManagerName               VARCHAR(25) NOT NULL,
        ManagerPhoneNumber               NUMERIC(10, 0),
        ManagerAddress            VARCHAR(30),
        ManagerDateofHire         DATE DEFAULT GETDATE(),
CONSTRAINT Manager_PK PRIMARY KEY (ManagerID));

CREATE TABLE Server_T
        (ServerID                 NUMERIC (11,0) NOT NULL,
        ServerName                VARCHAR(25) NOT NULL,
        ServerPhoneNumber                NUMERIC(10, 0),
        ServerAddress              VARCHAR(30),
        ServerDateofHire                 DATE DEFAULT GETDATE(),
        ManagerID NUMERIC(11,0),
CONSTRAINT Server_PK PRIMARY KEY (ServerID),
CONSTRAINT Server_FK FOREIGN KEY (ManagerID) REFERENCES Manager_T(ManagerID));

CREATE TABLE KitchenWorker_T
        (KitchenWorkerID                 NUMERIC(11,0) NOT NULL,
        KitchenWorkerName                VARCHAR(25) NOT NULL,
        KitchenWorkerPhoneNumber                NUMERIC(10, 0),
        KitchenWorkerAddress             VARCHAR(30),
        KitchenWorkerDateofHire           DATE DEFAULT GETDATE(),
        ManagerID                         NUMERIC(11,0),
CONSTRAINT KitchenWorker_PK PRIMARY KEY (KitchenWorkerID),
CONSTRAINT KitchenWorker_FK FOREIGN KEY (ManagerID) REFERENCES
Manager_T(ManagerID));

CREATE TABLE Customer_T
        (CustomerID                NUMERIC(11,0) NOT NULL,
        CustomerName               VARCHAR(25) NOT NULL,
        CustomerPhoneNumber NUMERIC(10, 0),
        CustomerAddress            VARCHAR(30),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

CREATE TABLE Order_T
        (OrderID                  NUMERIC(11,0) NOT NULL,
        OrderNumberofItems NUMERIC(11,0) NOT NULL,
        OrderDate                 DATE DEFAULT GETDATE(),
        CustomerID                 NUMERIC(11,0),
        ServerID                  NUMERIC (11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Order_FK2 FOREIGN KEY (ServerID) REFERENCES Server_T(ServerID));

CREATE TABLE Certificate_T
          (CertificateID              NUMERIC(11,0) NOT NULL,
          DateCompleted             DATE DEFAULT GETDATE(),
          Score                     NUMERIC(11,0),
          KitchenWorkerID           NUMERIC(11,0),
CONSTRAINT Certificate_PK PRIMARY KEY (CertificateID),
CONSTRAINT Certificate_FK FOREIGN KEY (KitchenWorkerID) REFERENCES
KitchenWorker_T(KitchenWorkerID));

CREATE TABLE Bill_T
          (BillID                   NUMERIC(11,0) NOT NULL,
          BillOrderDetails          VARCHAR(25),
          BillAmountDue              NUMERIC(11,2),
          CustomerID                NUMERIC(11,0),
          ServerID                   NUMERIC(11,0),
CONSTRAINT Bill_PK PRIMARY KEY (BillID),
CONSTRAINT Bill_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Bill_FK2 FOREIGN KEY (ServerID) REFERENCES Server_T(ServerID));

CREATE TABLE RestaurantKitchen_T
          (RestaurantID              NUMERIC(11,0) NOT NULL,
          KitchenWorkerID           NUMERIC(11,0) NOT NULL,
CONSTRAINT RestaurantKitchen_PK PRIMARY KEY (RestaurantID, KitchenWorkerID),
CONSTRAINT RestaurantKitchen_FK1 FOREIGN KEY (RestaurantID) REFERENCES
Restaurant_T(RestaurantID),
CONSTRAINT RestaurantKitchen_FK2 FOREIGN KEY (KitchenWorkerID) REFERENCES
KitchenWorker_T(KitchenWorkerID));

CREATE TABLE RestaurantServer_T
          (RestaurantID             NUMERIC(11,0) NOT NULL,
          ServerID                   NUMERIC(11,0) NOT NULL,
CONSTRAINT RestaurantServer_PK PRIMARY KEY (RestaurantID, ServerID),
CONSTRAINT RestaurantServer_FK1 FOREIGN KEY (RestaurantID) REFERENCES
Restaurant_T(RestaurantID),
CONSTRAINT RestaurantServer_FK2 FOREIGN KEY (ServerID) REFERENCES Server_T(ServerID));

CREATE TABLE RestaurantManager_T
          (RestaurantID             NUMERIC(11,0) NOT NULL,
          ManagerID                  NUMERIC(11,0) NOT NULL,
CONSTRAINT RestaurantManager_PK PRIMARY KEY (RestaurantID, ManagerID),
CONSTRAINT RestaurantManager_FK1 FOREIGN KEY (RestaurantID) REFERENCES
Restaurant_T(RestaurantID),
CONSTRAINT RestaurantManager_FK2 FOREIGN KEY (ManagerID) REFERENCES
Manager_T(ManagerID));

CREATE TABLE OrderKitchen_T
          (OrderID                  NUMERIC(11,0) NOT NULL,
          KitchenWorkerID           NUMERIC(11,0) NOT NULL,
CONSTRAINT OrderKitchen_PK PRIMARY KEY (OrderID, KitchenWorkerID),
CONSTRAINT OrderKitchen_FK1 FOREIGN KEY (OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT OrderKitchen_FK2 FOREIGN KEY (KitchenWorkerID) REFERENCES
KitchenWorker_T(KitchenWorkerID));
```

SQL Queries for Three External Views and Business Justifications

```
CREATE VIEW View_1 AS SELECT ServerName, CustomerName, BillOrderDetails,
BillAmountDue
FROM Bill_T, Customer_T, Server_T
WHERE Customer_T.CustomerID = Bill_T.CustomerID
AND Bill_T.ServerID = Server_T.ServerID AND BillOrderDetails > '3 items'
```

This view is important because it allows the company to examine and compare the total amount of each bill where there are more than 3 items bought on the bill. It also shows which server took care of which bill, which is important in case a server ever makes a mistake with the finances.

```
CREATE VIEW View_2 AS SELECT OrderID, ServerName, CustomerName
FROM Order_T, Server_T, Customer_T
WHERE Order_T.ServerID = Server_T.ServerID
AND Order_T.CustomerID = Customer_T.CustomerID
```

This view is useful because it shows the company which server took a specific order for a specific customer. This would be helpful for when customers contact top management with complaints or praises regarding a particular server and would allow for management to successfully penalize or reward the specific server accordingly for his/her good or bad work.

```
CREATE VIEW View_3 AS SELECT ManagerName, COUNT(ServerID) as CNT
FROM Manager_T, Server_T
WHERE Manager_T.ManagerID = Server_T.ManagerID
GROUP BY ManagerName HAVING COUNT(ServerID) > 1
```

This view shows the company which managers oversee multiple servers. It is necessary because most of the managers only oversee one server, but this allows the company to identify the ones that oversee multiple servers. The company would be able to see who oversees the most servers out of all the managers.