**DBMS for Supply Chain Management**

**Hood College, IT 530, Fall 2020**

**Description**

Project team of BCSY Automotive Ltd., consisting of members Beke, Charu, Sagoon and Yang, must set up the supply chain management system for an automobile plant which introduces new electric car models in Maryland. The supply chain management system is required to track part and supplier information, production operations, quality control and personnel management. The current flat file database cannot support the increased demand for the business. With increasing complexity of supply chains, and subsequent growth in supplier base, the current database system is also insufficient for providing rapid responses to the user operations. The new database system will aid the organization to gain better insights into their performance and help them make improved decisions and remain competitive in the industry.

**Scope**

The proposed system utilizes a relational database model to provide easy access to data while ensuring data accuracy and integrity. The database captures the flow of the supply chain in the form of metadata and detailed information. Metadata reflects those data needed to identify parts and suppliers and detailed data will describe the parts delivered to the related production line and supplier performance in terms of quality.

The database also captures the metadata and detailed information to manage workers for each production line and their respective supervisors. The detailed data will contain the information of the production line, parts and the workers.

The database will also capture the metadata and detailed information to track the finished cars. The sales department monitors the finished cars volume so that they can distribute the cars to dealers based on a plan.

The supply chain management system provides web-based access to the underlying relational database. Data entry forms provide the means to enter and edit the information of parts, suppliers, supplier performance, employees, production and sales. The user interface provides reports of the status for the parts, suppliers, employees, production and sales as well as the supplier performance and quality control. The web-based interface will utilize PHP processes to access the database and provide web access. Client-side processing will utilize JavaScript and CSS as necessary for entry validation and visualization.

**Data Requirement Analysis**

For each supplier, the Supplier relation contains a supplier name and a supplier location. It also contains a unique ID to identify the supplier. Contact information such as supplier phone and email is also recorded in the relation.

For each supplier, the Supplier Rating relation contains rating ID and supplier ID which together uniquely identify the supplier ratings. It records quality rating, delivery rating, customer service rating and overall performance rating. The Supplier Rating is totally dependent upon the Supplier.

For each part, the Parts relation contains a unique part number to identify the part. It also contains the part name, and part type. The relation contains information of parts that will be supplied by a supplier and used by a production line.

For each production line, the Production Line relation contains a unique production line number to track the production. It also contains the production line name.

For each employee, the Employee relation contains a unique emp ID to identify the employee. It also records employee name, phone, and email. The relation also tracks the supervisor number for an employee. It also records the department ID which the employee works in.

For each dependent, the Dependent relation requires both employee ID and the dependent name to uniquely identify the dependent. It also contains the date of birth, gender, relation\_type. The Dependent relation is totally dependent upon the Employee.

For each department, the Department relation contains department ID to identify the department. It also contains a department name.

For each car, the Car relation contains a unique VIN (Vehicle Identification Number) to identify the car. It also contains model number, color, year of manufacture, manufacturer price.

For each model, the Model relation contains a model number to identify the car model. It also contains model name, model year and body type.

For each dealer, the Dealership relation contains a unique dealer ID to identify the dealer. The relation also contains dealer name, dealer phone, dealer email, dealer shipping address and billing address.

For each car sale, the Car Sale relation contains a unique sales ID to identify the sale. It also contains sales date and sales amount. It also records the VIN to track each car in the Sale as well as the Dealer ID to track the dealership which buys the car. The Sale relation also records which employee handles the sale.

For each payment, the Invoice relation contains a unique invoice number to identify the invoice. It also contains the discount, tax, billed amount, invoice date and payment\_duedate.

**Other Requirement Analysis**

The HR department will collect and enter the initial employee data for both the business department and production department into the system.

The production department acquires the parts from suppliers and assigns employees to operate the production line, and will enter the data for parts, supplier, supplier rating, supervisors and workers for each production line and car into the system.

The supply chain process for car production and dispatch will occur if: first, the part number matches the production line which it should be delivered to, and the car matches the sales ID to the related dealer.

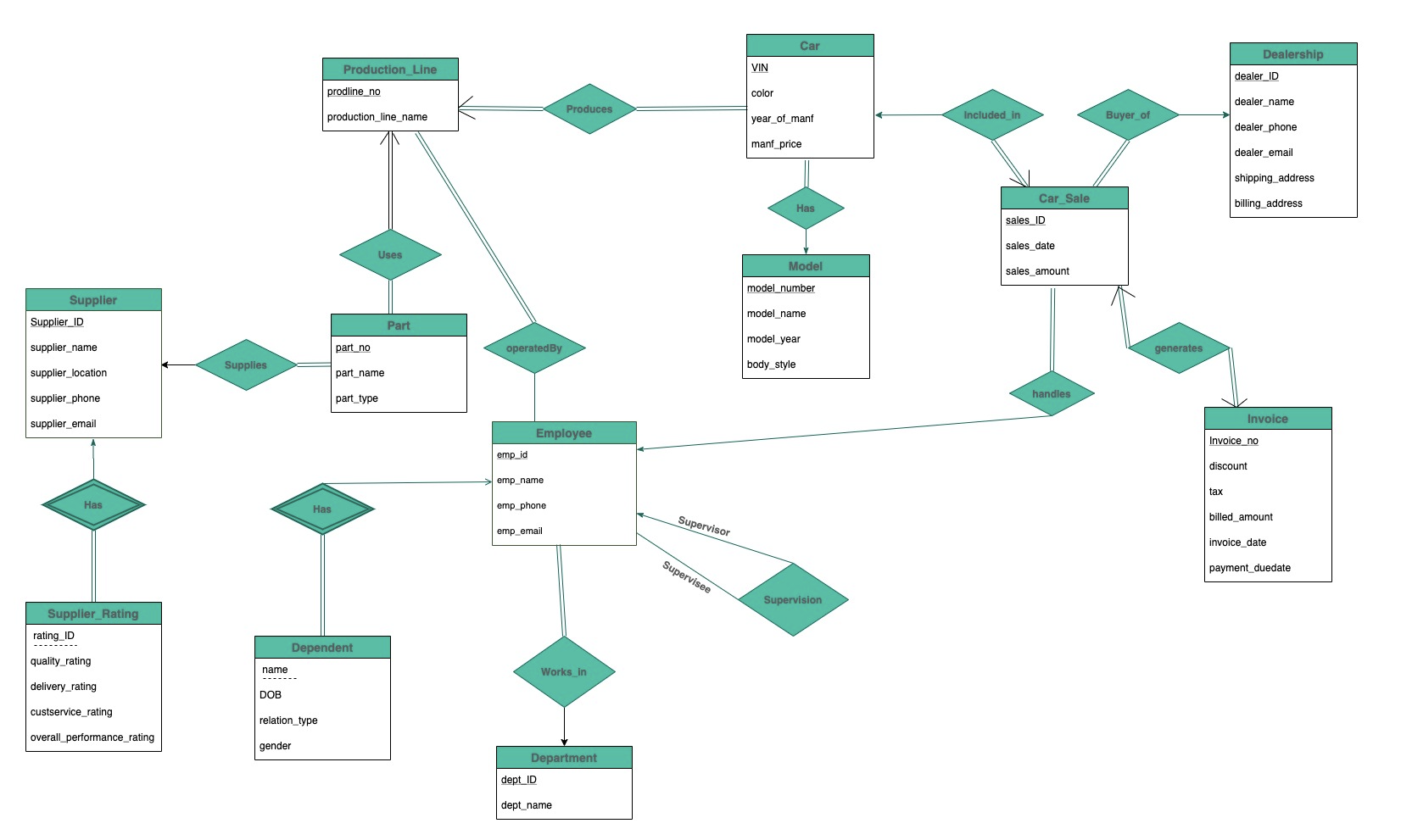
The sales department will enter the sales records which match the sale of each car to each dealer.

The finance department will review the sale and determine the invoice to be billed to each dealer.

The system will update the tables for the production and sales records based on the production volume input.

The initial system will manage the supply chain data from inbound logistics from part suppliers to outbound logistics to dealers.

**ER DIAGRAM:**

****

**ER Diagram Assumptions:**

* Each supplier may have one or more supplier ratings and each supplier rating must be given to at most one supplier.
* The Supplier Rating relation is a weak entity set because it is totally dependent on the Supplier. A Supplier’s rating does not matter if that supplier is discontinued.
* Each supplier can supply one or more parts and also each part can be supplied by at most one supplier.
* Each part must be used by at most one Production line and each production line must use one or more Parts.
* Each production line must be operated by one or more employees and an employee may operate on at one or more production lines.
* Each employee must work in at most one department and each department can have many employees.
* Each employee may have many dependents and each dependent must be associated with at most one employee. The dependent details do not matter if the employee does not work in the company anymore.
* Each employee may have at most one supervisor and each supervisor may supervise one or more employees.
* Each production line must produce one or more cars and each car must be produced by at most one production line.
* Each Car must have at most one model and each Model may have many cars.
* Each car can be included in at most one car sale and each car sale can include at most one car.
* Each employee may handle many car sales and each car sale must be handled by at most one employee.
* Each car sale must generate at most one invoice and each invoice must be generated for at most one car sale.
* Each dealership may be the buyer of one or more car sales and each car sale must have at most one dealership as a buyer.
* Each car can have only one VIN.
* Employee\_ID, supplier\_ID and dealer\_ID are unique within the system.
* A car with a unique VIN can be manufactured but not sold to a dealer.
* Inventory, payment and transportation modules of supply chain management are not handled in this system.

**RELATIONAL SCHEMAS AND EXPLANATIONS**

* 1. SUPPLIER

Supplier (supplier\_ID: INT (10), supplier\_name: VARCHAR (25), supplier\_location: VARCHAR (50), supplier\_phone: VARCHAR (14), supplier\_email: VARCHAR (30))

The Supplier table stores information about each supplier, such as their ID, name, location, phone number and email address. The supplier ID is the primary key for the Supplier table.

* 1. PART

Part (part\_no: INT (9), part\_name: VARCHAR (20), part\_type: VARCHAR (20), supplier\_ID: INT (10), prodline\_no: VARCHAR(10))

The Part table contains the information about each part, such as part number, part name, part type. The part number is the primary for the Part table.

* 1. SUPPLIER\_RATING

Supplier\_Rating (rating\_ID: INT (4), supplier\_ID: INT(10), quality\_rating: INT(2), delivery\_rating: INT(2), custservice\_rating: INT(2), overall\_performance\_rating: DECIMAL (4, 2))

The Suppler\_Rating table contains the information about supplier performance, such as rating ID, supplier ID, quality rating, delivery rating, customer service rating, and overall performance rating. The rating ID and supplier ID are both used as the primary key.

* 1. PRODUCTION\_LINE

Production\_line (prodline\_no: VARCHAR(10),

production\_line\_name: VARCHAR (25))

The Production\_Line table contains the production line number as the primary key, and also production line name.

* 1. MODEL

Model (model\_number: VARCHAR (8), model\_name: VARCHAR (25), model\_year: YEAR, body\_style: VARCHAR(25))

The Model table contains the model number as primary key, model name, model year, and body style.

* 1. CAR

Car (VIN: VARCHAR (17), color: VARCHAR (10), year\_of\_manf: YEAR,

manf\_price: DECIMAL (10, 2), model\_number: VARCHAR(8), prodline\_no: VARCHAR(10))

The Car table stores the information about each car, such as VIN, color, year of manufacturing, year, manufacturing price and production line number was manufactured in. The VIN is the primary key for the Car table.

* 1. EMPLOYEE

Employee (empID: INT(5), emp\_name: VARCHAR (30), emp\_phone: VARCHAR (14), emp\_email: VARCHAR (30), supervisor\_no: INT(5), dept\_ID: VARCHAR(6))

The Employee table contains the information about each employee, such as employee ID, employee name, employee phone, employee email and supervisor number. The employee ID is the primary key.

* 1. DEPENDENT

Dependent (empID: INT(5), dependent\_name: VARCHAR(30), DOB: DATE, relation\_type: VARCHAR(20), gender: VARCHAR(5))

The Dependent table contains the employee ID as well as dependent name, both together serve as the primary key and holds Date of Birth, relation type, gender as well.

* 1. DEPARTMENT

Department(dept\_ID: VARCHAR(6), dept\_name: VARCHAR(25))

The Department table contains the department ID as the primary key, and department name.

* 1. OPERATES

Operates(empID: INT(5), prodline\_no: VARCHAR(10))

The Operates table maps the employee ID with the production Line number that they operate in. Both attributes together serve as the primary key.

* 1. DEALERSHIP

Dealership (dealer\_ID: INT (6), dealer\_name: VARCHAR (30), dealer\_phone: VARCHAR (14), dealer\_email: VARCHAR (30), shipping\_address: VARCHAR (50), billing\_address: VARCHAR (50))

The Dealership table contains the dealer ID as primary key, dealer name, dealer phone, dealer email, shipping address and billing address.

* 1. CAR\_SALE

Car\_Sale (sales\_ID: INT(6), VIN: VARCHAR(17), dealer\_ID: INT(6), sales\_empID: INT(5), invoice\_num: INT(7), sales\_amount: DECIMAL(10, 2), discount: DECIMAL(10, 2), tax: DECIMAL(10, 2), billed\_amount: DECIMAL(10, 2), sales\_date: DATE, invoice\_date: DATE, payment\_duedate: DATE)

The Car\_Sale table contains the sales ID as primary key, and the information for each sale such as VIN, dealer, sales employee ID, invoice number, sales amount, discount, tax, billed amount, sales date, invoice data, payment due date.

**IMPLEMENTATION OF THE SCHEMA IN MYSQL**

use XXXdb;

/\* Database Table Creation \*/

/\* Drop any existing tables. Any errors are ignored. \*/

DROP TABLE IF EXISTS Car\_Sale;

DROP TABLE IF EXISTS Dealership;

DROP TABLE IF EXISTS Car;

DROP TABLE IF EXISTS Part;

DROP TABLE IF EXISTS Production\_line;

DROP TABLE IF EXISTS Dependent;

DROP TABLE IF EXISTS Employee;

DROP TABLE IF EXISTS Department;

DROP TABLE IF EXISTS Model;

DROP TABLE if EXISTS Supplier\_Rating;

DROP TABLE IF EXISTS Supplier;

/\* Add each table. \*/

/\* supplier\_ID must have a not null constraint because it needs to be unique\*/

CREATE TABLE Supplier(

supplier\_ID INT(10) PRIMARY KEY,

supplier\_name VARCHAR(25) NOT NULL UNIQUE,

supplier\_location VARCHAR(50),

supplier\_phone VARCHAR(14),

supplier\_email VARCHAR(30)) ENGINE=InnoDB;

/\* The primary key needs to be rating\_id and supplier\_id to make each row unique in the Supplier\_Rating table. If a Supplier\_ID is

updated or deleted in Supplier, we also want to update or delete the corresponding rating data in this Supplier\_Rating table. \*/

CREATE TABLE Supplier\_Rating(

rating\_ID INT(4),

supplier\_ID INT(10) NOT NULL,

quality\_rating INT(2),

delivery\_rating INT(2),

custservice\_rating INT(2),

overall\_performance\_rating DECIMAL (4, 2),

PRIMARY KEY (rating\_ID, supplier\_ID),

Foreign key(supplier\_ID) references Supplier(supplier\_ID)

ON UPDATE CASCADE

ON DELETE CASCADE) ENGINE=InnoDB;

/\* model\_number must have a not null constraint because it needs to be unique\*/

CREATE TABLE Model(

model\_number VARCHAR(8) PRIMARY KEY,

model\_name VARCHAR(25),

model\_year YEAR,

body\_style VARCHAR(25)) ENGINE=InnoDB;

/\* dept\_ID must have a not null constraint because it needs to be unique.\*/

CREATE TABLE Department(

dept\_ID VARCHAR(6) PRIMARY KEY,

dept\_name VARCHAR(25))ENGINE=InnoDB;

/\* empID must have a not null constraint because it needs to be unique. If an empID which is also a supervisor\_no

for another employee is deleted in the Employee table, we set the deleted supervisor\_no to null, and if supervisor\_no is updated, it is updated as well.

The dept\_ID must have a not null constraint because each Employee must have a dept\_ID (total participation). If a dept\_ID is updated, we also want to update the respective data in this table.\*/

CREATE TABLE Employee(

empID INT(5) PRIMARY KEY,

emp\_name VARCHAR(30),

emp\_phone VARCHAR (14),

emp\_email VARCHAR(30),

supervisor\_no INT(5),

dept\_ID VARCHAR(6) NOT NULL,

Foreign key (supervisor\_no) references Employee (empID)

ON DELETE SET NULL

On UPDATE CASCADE,

Foreign key (dept\_ID) references Department (dept\_ID)

ON UPDATE CASCADE

On DELETE CASCADE) ENGINE=InnoDB;

/\* The primary key needs to be dependent\_name and empID to make each row unique in the Dependent table. If an empID is updated or deleted in Employee, we also want to update or delete the corresponding empID and dependent data in this Dependent table. \*/

CREATE TABLE Dependent(

empID INT(5) NOT NULL,

dependent\_name VARCHAR(30),

DOB DATE,

relation\_type VARCHAR(20),

gender VARCHAR(5),

Primary key (empID, dependent\_name),

Foreign key (empID) references Employee (empID)

ON UPDATE CASCADE

ON DELETE CASCADE) ENGINE=InnoDB;

/\* prodline\_no must have a not null constraint because it needs to be unique.\*/

CREATE TABLE Production\_line(

prodline\_no VARCHAR(10) PRIMARY KEY,

production\_line\_name VARCHAR(25),

Foreign key (empID) references Employee(EmpID)

ON UPDATE CASCADE) ENGINE=InnoDB;

/\* part\_no must have a not null constraint because it needs to be unique.

Prodline\_no and Supplier\_ID must have a not null constraints because each Part must have a Prodline\_no and a Supplier(total participation).

If a prodline\_no or supplier\_ID is updated or deleted, we want to set to update or delete in this Part table.

\*/

CREATE TABLE Part(

part\_no INT(9) PRIMARY KEY,

part\_name VARCHAR(20),

part\_type VARCHAR(20),

supplier\_ID INT(10) NOT NULL,

Prodline\_no VARCHAR(10) NOT NULL,

Foreign Key (Prodline\_no) references Production\_line(Prodline\_no)

ON DELETE CASCADE

ON UPDATE CASCADE,

Foreign Key (supplier\_ID) references Supplier(supplier\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=InnoDB;

/\*The primary key needs to be empID and prodline\_no to make each row unique in the Operates table. If a prodline\_no or empID is updated or deleted in the Production\_line or Employee respectively, we also want to update or delete the corresponding data in this Operates table.\*/

CREATE TABLE Operates(

empID INT(5),

prodline\_no VARCHAR(10),

Primary key (prodline\_no, empID),

Foreign key (prodline\_no) references Production\_line (prodline\_no)

ON DELETE CASCADE

ON UPDATE CASCADE,

Foreign key (empID) references Employee (empID)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=InnoDB;

/\* VIN must have a not null constraint because it needs to be unique. The model\_number and prodline\_no must have a not null constraint because each Car must have a model\_number and a prodline\_no(total participation).

If a model\_number is updated or deleted, we also want to update or delete the respective data in the Car table. If a prodline\_no is updated, we want to update in this table.\*/

CREATE TABLE Car(

VIN VARCHAR(17) PRIMARY KEY,

color VARCHAR(10),

year\_of\_manf YEAR,

manf\_price DECIMAL(10, 2),

model\_number VARCHAR(8) NOT NULL,

prodline\_no VARCHAR(10) NOT NULL,

Foreign key(model\_number) references Model(model\_number)

ON DELETE CASCADE

ON UPDATE CASCADE,

Foreign key(prodline\_no) references Production\_line(prodline\_no)

ON UPDATE CASCADE) ENGINE=InnoDB;

/\* dealer\_ID must have a not null constraint because it needs to be unique\*/

CREATE TABLE Dealership(

dealer\_ID INT(6) PRIMARY KEY,

dealer\_name VARCHAR(30),

dealer\_phone VARCHAR (14),

dealer\_email VARCHAR(30),

shipping\_address VARCHAR(50),

billing\_address VARCHAR(50)) ENGINE=InnoDB;

/\* sales\_ID must have a not null constraint because it must be unique. VIN must have

a not null and Unique constraint because each Car\_Sale must include a VIN (total participation) and each VIN must be unique.

dealer\_ID and sales\_empID must have a not null constraint because each Car\_Sale must have a dealer\_ID and sales\_empID(total participation). If a dealer\_ID, VIN, or sales\_empID is updated or deleted, we also want to update or delete the respective data in the Car\_Sale table. \*/

CREATE TABLE Car\_Sale(

sales\_ID INT(6) PRIMARY KEY,

VIN VARCHAR(17) NOT NULL UNIQUE,

dealer\_ID INT(6) NOT NULL,

sales\_empID INT(5) NOT NULL,

invoice\_num INT(7) NOT NULL UNIQUE,

sales\_amount DECIMAL(10, 2),

discount DECIMAL(10, 2),

tax DECIMAL(10, 2),

billed\_amount DECIMAL(10, 2),

sales\_date DATE,

invoice\_date DATE,

payment\_duedate DATE,

Foreign key (VIN) references Car (VIN)

ON DELETE CASCADE

ON UPDATE CASCADE,

Foreign key (dealer\_ID) references Dealership(dealer\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE,

Foreign key (sales\_empID) references Employee (empID)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE=InnoDB;

**LOADING DATA**

insert into Supplier (supplier\_ID, supplier\_name, supplier\_location, supplier\_phone, supplier\_email) values ( 2014940110, 'Johnson Controls', '6487 US-19, Zebulon, GA 30295', '(770) 567-0968', 'info@Johnson.Controls.com');

insert into Supplier (supplier\_ID, supplier\_name, supplier\_location, supplier\_phone, supplier\_email) values ( 2014940210, 'Delphi', '3900 E Holland Rd, Saginaw, MI 48601', '(989) 401-2102', '[info@delphi.com](mailto:info@delphi.com)');

insert into Supplier (supplier\_ID, supplier\_name, supplier\_location, supplier\_phone, supplier\_email) values ( 2014940310, 'Denso', '24777 Denso Dr, Southfield, MI 48033', '(248) 620-1170', '[info@denso.com](mailto:info@denso.com)');

insert into Supplier (supplier\_ID, supplier\_name, supplier\_location, supplier\_phone, supplier\_email) values ( 2014940410, 'Inergy', '2710 Bellingham Dr # 400, Troy, MI 48083', '(248) 743-5700', '[info@inergy.com](mailto:info@inergy.com)');

insert into Supplier (supplier\_ID, supplier\_name, supplier\_location, supplier\_phone, supplier\_email) values (2014941410, ‘Treves’, ‘29145 US-72, Hollywood, AL 35752’,(256) 259-2088, ‘[info@treves.com](mailto:info@treves.com)’);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating) values ( 1001, 2014940110, 10, 9,9, 9.33);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating) values ( 1002, 2014940210, 10, 10, 10, 10);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating) values ( 1003, 2014940310, 10, 10, 10, 10);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating) values ( 1004,2014940410, 8, 10, 9, 9);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating)values (1005, 2014940510, 8, 9, 8, 8.33);

insert into Supplier\_Rating (Rating\_ID, Supplier\_ID, Quality\_rating, Delivery\_Rating, CustService\_rating, overall\_performance\_rating) values( 1006, 2014940610, 10, 9, 9, 9.33);

insert into Model (model\_number, model\_name, model\_year, body\_style) values( 'F30', '3 series', 2011, 'Sedan');

insert into Model (model\_number, model\_name, model\_year, body\_style) values( 'F45', '2 series', 2014, 'Active Tourer');

insert into Model (model\_number, model\_name, model\_year, body\_style) values( 'F49', 'X1', 2015, 'SUV');

insert into Model (model\_number, model\_name, model\_year, body\_style) values( 'F49 PHEV', 'X1 Hybrid', 2015, 'SUV Hybrid');

insert into Model (model\_number, model\_name, model\_year, body\_style) values( ‘G38’,’5 series’,2017,’Sedan’);

insert into Department (dept\_ID, dept\_name) values ('BCSY-1', 'CEO Office');

insert into Department (dept\_ID, dept\_name) values ('BCSY-2', 'Sales');

insert into Department (dept\_ID, dept\_name) values ('BCSY-3', 'Finance');

insert into Department (dept\_ID, dept\_name) values ('BCSY-4', 'Human Resource');

insert into Department (dept\_ID, dept\_name) values ('BCSY-5', 'Operation and Production');

insert into Department (dept\_ID, dept\_name) values ('BCSY-6', 'Supply Chain');

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(61107, 'Beke Esther', '(301) 111-112', 'beke@bcsy.com', null, 'BCSY-1');

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(61106, 'Yang Chen', '(301) 111-111', 'yang@bcsy.com ', 61107, 'BCSY-2');

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(61108, 'Charu Ravi', '(301) 111-113', 'charu@bcsy.com ', 61107, 'BCSY-3');

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(61109, 'Sagoon Thapa', '(301) 111-114', 'sagoon@bcsy.com ', 61107, 'BCSY-4');

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values( 68319, ‘Kayling Rogers’, (301) 111-115, ‘kayling@bcsy.com’, 61107 , ‘BCSY-5’ );

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(66928, ‘Blaze Lee’,(301) 111-116, ‘blaze@bcsy.com’,61107, ‘BCSY-5’);

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values( 67832, ‘Clare Martins’,(301) 111-117,’clare@bcsy.com’,61107,‘BCSY-5’);

insert into Employee (empID, emp\_name, emp\_phone, emp\_email, supervisor\_no, dept\_ID) values(65646, ‘Jonas McCain’,(301) 111-118, ‘jonas@bcsy.com’, 61107,‘BCSY-5’);

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 68319, 'Charlie Rogers', '2008-06-11', 'Child', 'M');

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 68319, 'Alan Rogers', '2000-08-01', 'Child', 'M');

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 67832, 'Megan Martins', '2010-06-28', 'Child', 'F');

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 65646, 'Susan McCain', '1992-02-09', 'Wife', 'F');

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 61108, 'Rahul Ravi', '1988-07-18', 'Husband', 'M');

insert into Dependent (empID, dependent\_name, DOB, relation\_type, gender ) values ( 65646, ‘Susan McCain’, ‘1992-02-09’, ‘Wife’, ‘F’);

insert into Production\_line (prodline\_no, Production\_line\_name) values ( 'Assy\_A', 'Assembly Powertrain Line');

insert into Production\_line (prodline\_no, Production\_line\_name) values ( 'Assy\_F', 'Assembly Transmission Line');

insert into Production\_line (prodline\_no, Production\_line\_name) values ( 'Assy\_K', 'Assembly Exterior Line');

insert into Production\_line (prodline\_no, Production\_line\_name) values ( 'Assy\_I', 'Assembly Interior Line');

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351257, 'Battery', 'Powertrain', 2014940110 , 'Assy\_A');

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351258, 'Radiator', 'Powertrain', 2014940210, 'Assy\_A');

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351259, 'Alternator', 'Powertrain', 2014940310, 'Assy\_A');

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351260, 'Fuel Tank', 'Powertrain', 2014940410 , 'Assy\_A');

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351261, ‘Axle’, ‘Transmission’,2014940510, ‘Assy\_F’);

insert into Part (part\_no, part\_name, part\_type, supplier\_ID, Prodline\_no) values (897351271, ‘Traction Motor’, ‘Transmission’, 2014940510, ‘Assy\_F’);

insert into Operates (empID,prodline\_no) values (68319,'Assy\_A');

insert into Operates (empID,prodline\_no) values (66928,'Assy\_F');

insert into Operates (empID,prodline\_no) values (67832,'Assy\_K');

insert into Operates (empID,prodline\_no) values (69062,'Assy\_A');

insert into Operates (empID,prodline\_no) values (66564, 'Assy\_A');

insert into Operates (empID,prodline\_no) values (65646, 'Assy\_I');

insert into Operates (empID,prodline\_no) values (63679, 'Assy\_F');

insert into Operates (empID,prodline\_no) values (64989, 'Assy\_K');

insert into Operates (empID,prodline\_no) values (65271, 'Assy\_I');

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values ('WBSBR93454P464716', 'Red', 2020, 41250, 'F30', 'Assy\_A');

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values ('WBAPH5G55ANL80668', 'Silver', 2020, 41250, 'F30', 'Assy\_F');

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values ('WBAKA4C5XCDS99012', 'Silver', 2020, 41250, 'F30', 'Assy\_K');

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values ('WBA3X9C50ED153351', 'Black', 2020, 35900, 'F45', 'Assy\_A');

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values

(‘WBAUL7C50BVJ78714’, ‘Gold’, 2020, 35200, ‘F49’, ‘Assy\_A’);

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values (‘WBAEG2300M0019756’, ‘Green’, 2020, 37200, ‘F49PHEV’, ‘Assy\_I’);

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values (‘WBAVW11038A276930’, ‘Blue’, 2020, 54200, ‘G38’, ‘Assy\_A’);

insert into Car (VIN, Color, year\_of\_manf, manf\_price, model\_number, prodline\_no) values (‘WBAGK9340YDE35021’, ‘White’, 2020, 54200, ‘G38’, ‘Assy\_K’);

insert into Dealership ( dealer\_ID, dealer\_name, dealer\_phone, dealer\_email, shipping\_address, billing\_address) values( 18337,'Crystal Auto Sales', '(240) 702-0984', 'sales@crystalauto.com', '19705 Waters Rd B, Germantown, MD 20874', '19705 Waters Rd B, Germantown, MD 20874');

insert into Dealership ( dealer\_ID, dealer\_name, dealer\_phone, dealer\_email, shipping\_address, billing\_address) values( 12464,'Direct Auto Access', '(240) 801-4735', 'sales@directauto.com ' , '12429 Middlebrook Road, Germantown, MD 20874', '12466 Halton Road, Germantown, MD 20874');

insert into Dealership ( dealer\_ID, dealer\_name, dealer\_phone, dealer\_email, shipping\_address, billing\_address) values( 12765,'Criswell Maserati', '(240) 252-2637', 'sales@maserati.com' , '19570 Amaranth Dr suite a, Germantown, MD 20874', '19705 Waters Rd B, Germantown, MD 20874');

insert into Dealership ( dealer\_ID, dealer\_name, dealer\_phone, dealer\_email, shipping\_address, billing\_address) values( 18337,'Maryland Metal', '(301) 284-0531', 'sales@metal.com' , '11940 Little Seneca Pkwy, Clarksburg, MD 20871', '10550 River Road, Clarksburg, MD 20871');

insert into Dealership ( dealer\_ID, dealer\_name, dealer\_phone, dealer\_email, shipping\_address, billing\_address) values( 10408, ‘Maryland Metal’, ‘(301) 284-0531’, ‘sales@metal.com’, ‘11940 Little Seneca Pkwy, Clarksburg, MD 20871’, ‘10550 River Road, Clarksburg, MD 20871’);

insert into Car\_Sale ( sales\_ID, VIN, dealer\_ID, sales\_empID, invoice\_num, sales\_amount, discount, tax, billed\_amount, sales\_date, invoice\_date, payment\_duedate) values ( 201107, 'WBSBR93454P464716', 18337, 61107, 6919827, 41250, 2063, 2351,41539, '2020-10-01', '2020-10-16', '2020-11-15');

insert into Car\_Sale ( sales\_ID, VIN, dealer\_ID, sales\_empID, invoice\_num, sales\_amount, discount, tax, billed\_amount, sales\_date, invoice\_date, payment\_duedate) values ( 201108, 'WBAPH5G55ANL80668', 18337, 61107, 5377679, 41250, 2063, 2351, 41539, '2020-10-02', '2020-10-17', '2020-11-16');

insert into Car\_Sale ( sales\_ID, VIN, dealer\_ID, sales\_empID, invoice\_num, sales\_amount, discount, tax, billed\_amount, sales\_date, invoice\_date, payment\_duedate) values ( 201109, 'WBAKA4C5XCDS99012',12464, 61107, 6685472, 41250, 2063, 2351, 41539, '2020-10-03', '2020-10-18', '2020-11-17');

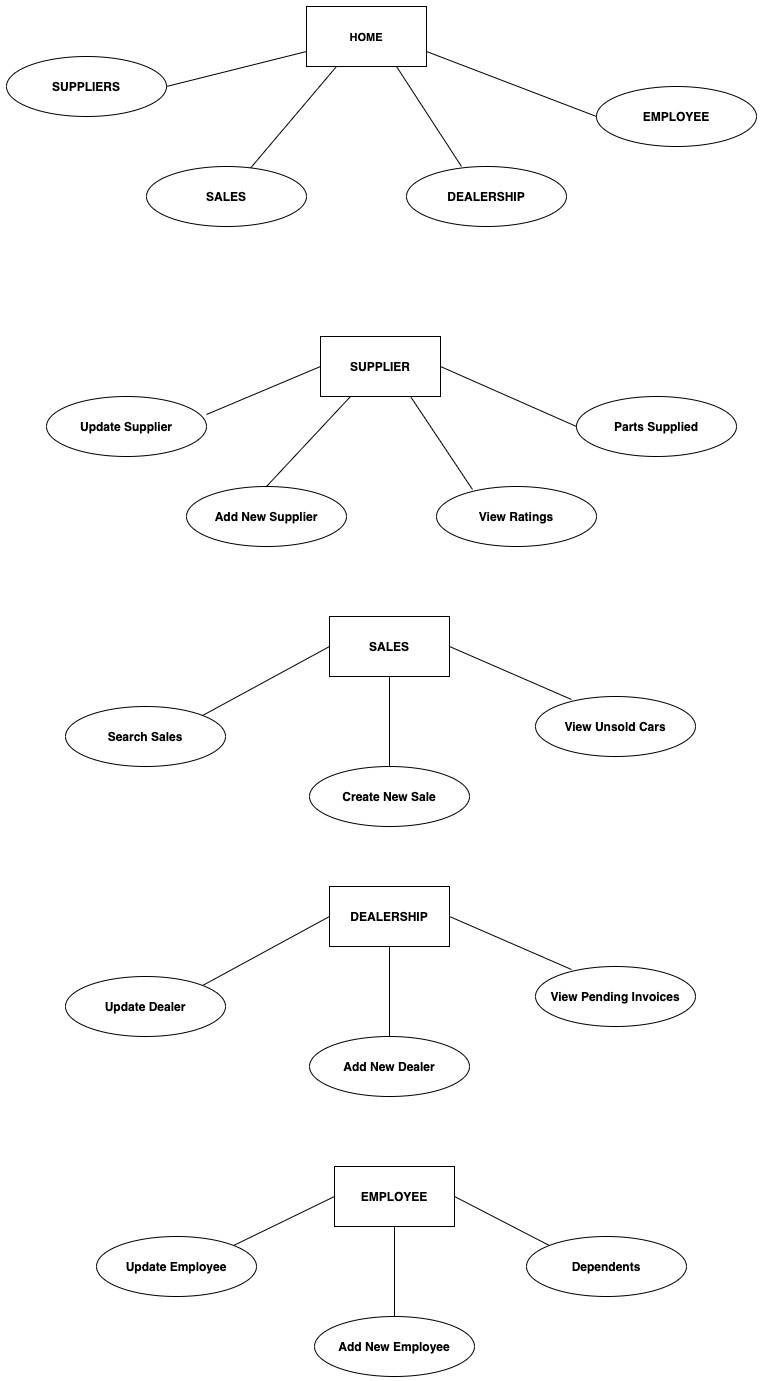
insert into Car\_Sale ( sales\_ID, VIN, dealer\_ID, sales\_empID, invoice\_num, sales\_amount, discount, tax, billed\_amount, sales\_date, invoice\_date, payment\_duedate) values ( 201110, 'WBA3X9C50ED153351', 12464, 61107, 6285206, 41250, 2063, 2351, 36151, '2020-10-04', '2020-10-19', '2020-11-17');

insert into Car\_Sale ( sales\_ID, VIN, dealer\_ID, sales\_empID, invoice\_num, sales\_amount, discount, tax, billed\_amount, sales\_date, invoice\_date, payment\_duedate) values ( 201111, ‘5UXZW0C54CL957682’, 12765,68736, 5566653, 35900, 1795,2046, 36151, ‘2020-10-05,2020-10-20’, ‘2020-11-19’);

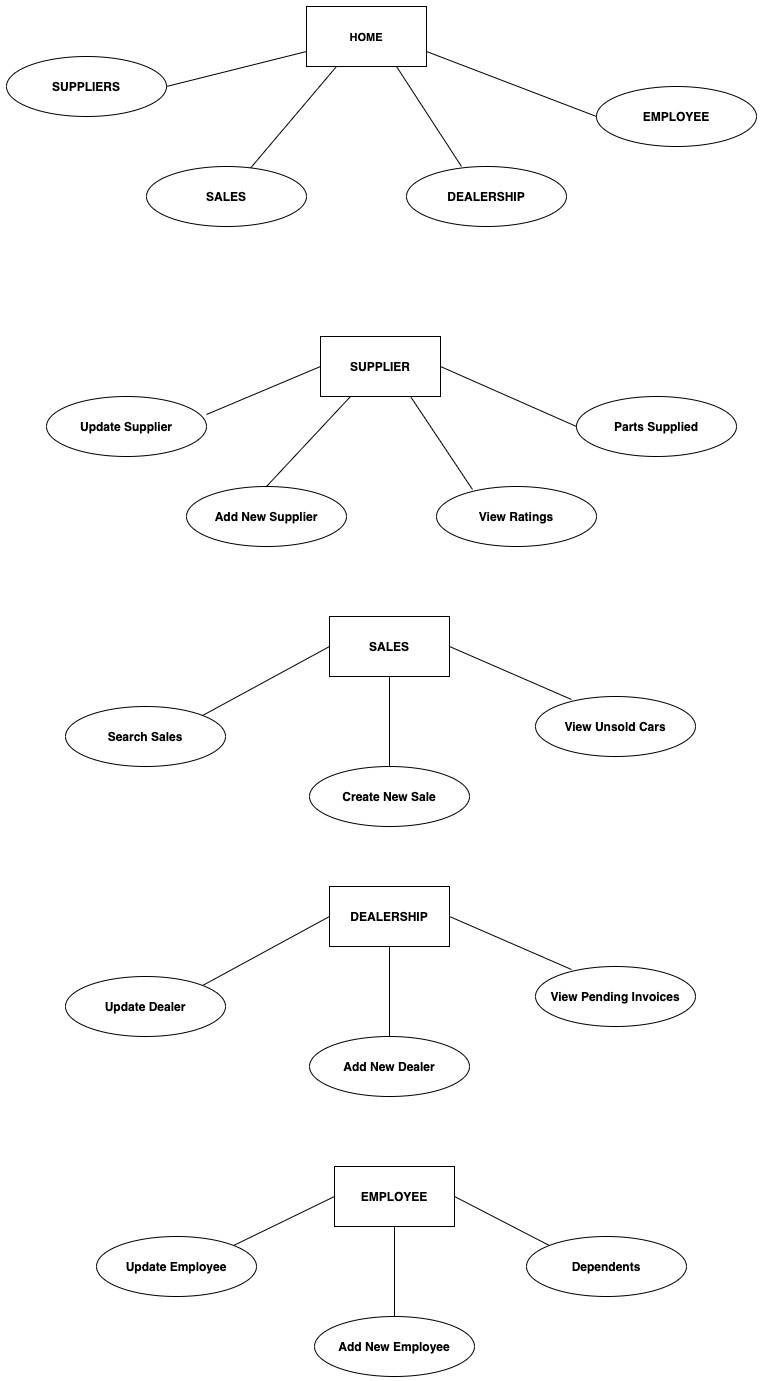
**USER INTERFACE DESIGN:**

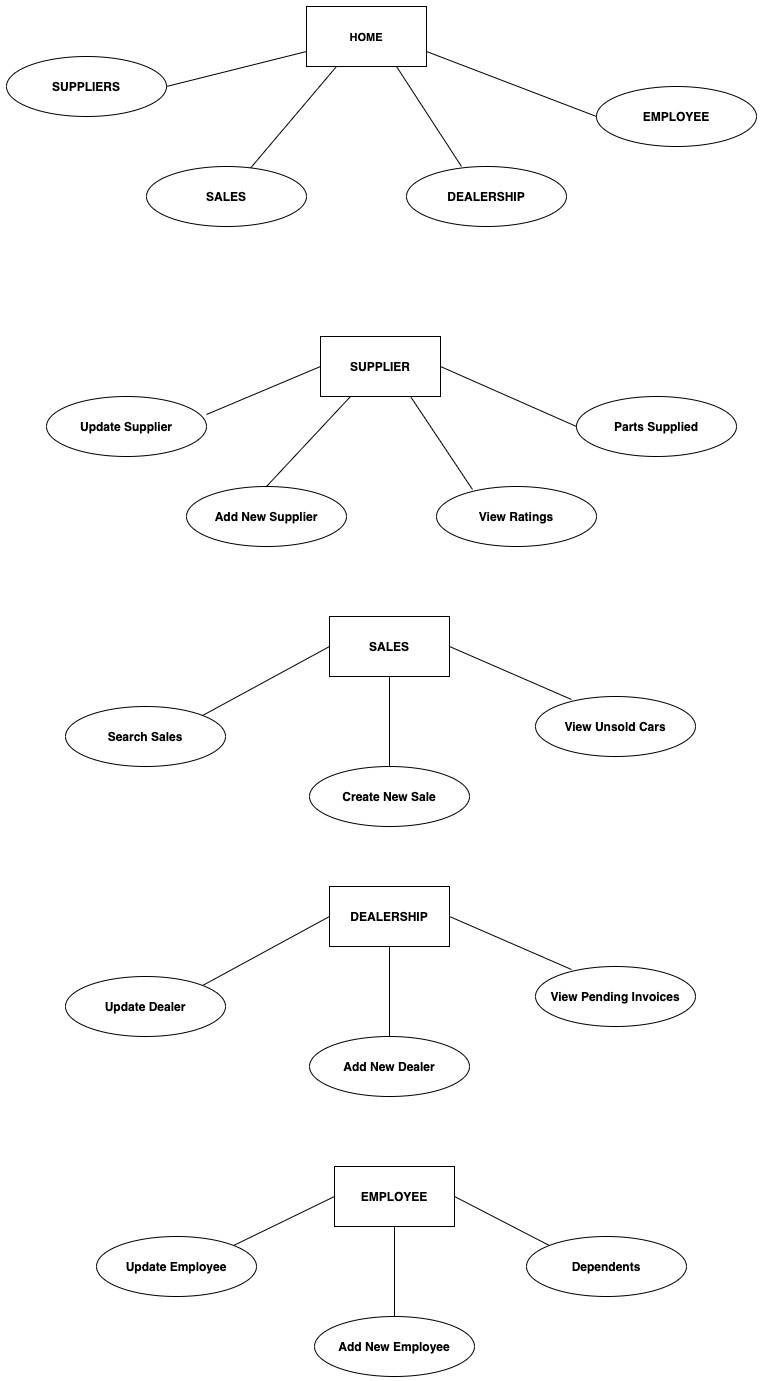
**WEBSITE DIAGRAM:**

High-level Diagram**:**



Lower-level Diagrams:





**WEBSITE FUNCTIONALITY DESCRIPTION:**

**Supplier Information Page:**

For supplier information in our database, the users are presented with the tabs of Update Supplier, Add New Supplier, View Supplier Ratings, View Parts Supplied.

The Update Supplier tab presents the information of Supplier ID, Supplier Name, Supplier Location, Supplier Phone, Supplier Email. Users can search for a certain supplier with Supplier ID and update their information.

The Add New Supplier tab enables users to enter supplier information in terms of Supplier ID, Supplier Name, Location, Phone and Email.

The View Supplier Ratings tab presents an overview for all suppliers rating with Supplier ID, Supplier Name, Rating ID, Quality Rating, Delivery Rating, Customer Service Rating and Overall Performance Rating.

The View Parts Supplied tab enables users to search for Supplier ID and present the Parts supplied information such as Supplier ID, Supplier Name, Part No, Part Name, Part Type, Proline no.

**Sales Information Page:**

For Sales information in our database, users are presented with the tabs of Search, Create New Sale, View Unsold Cars.

The Search tab presents the sales records in terms of Sales ID, VIN, Model Name, Model Year, Color, Dealer ID, Dealer Name, Billed amount, Sales Date, Sales Employee ID, Sales Employee Name. Users can search for sales records by the time period.

The Create New Sale tab enables users to enter a new sales details in terms of Sales ID,

VIN, Dealer ID, Invoice Number, Sale Amount, Discount, Tax, Billed Amount, Sales Employee ID, Sale Date, Invoice Date, Payment Due Date.

The View Unsold Cars tab presents the Unsold Cars in terms of VIN, Color, Year of Manufactured, Price, Model Number, Model Name, Production Line No, and Production Line Name.

**Dealership Information Page:**

For Dealership information in our database, users are presented with the tabs of Update Dealers, Add New Dealer, View Pending Invoices.

The Update Dealers tab presents the dealers information with Dealer ID, Dealer Name, Dealer Phone, Dealer Email, Shipping Address, Billing Address. Users can search for a certain dealer information by Dealer ID and update their details.

The Add New Dealer tab enables users to enter new dealer details with Dealer ID, Dealer Name, Dealer Phone, Dealer Email, Shipping Address, Billing Address.

The View Pending Invoices tab enables users to search for invoice information for a particular dealer ID. Any invoice payment information which is due soon for the dealer is pulled from Sales.

**Employees Information Page:**

For employee information in our database, users are presented with the tabs of Update Employee, Add New Employee, and Dependents.

The tab of Update Employee presents the information of Employee ID, Employee Name, Employee Phone, Employee Email, Department ID, Department Name, Supervisor no, Supervisor Name. Users can search for a certain employee information by employee ID and update the employee details.

The tab of Add New Employee enables users to enter employee details in terms of Employee ID, Employee Name, Employee Phone, Employee Email, Supervisor no, and Department ID.

The tab of Dependents presents the dependent records and users can add a new dependent information for a particular employee.

**WEBSITE ACCESS LINK:**

<http://pluto.hood.edu/~yc9/Home.html>