Navid Hariri, Summeeyah Haq, Juyoung Suh, Junjie Xiao, Shonda O’Conner

information systems 441  Professor Davoudian

Cool cars company

Contents

[Introduction 5](#_Toc26380726)

[Case Scenario 5](#_Toc26380727)

[Business Needs 5](#_Toc26380728)

[Requirements 5](#_Toc26380729)

[Functions 6](#_Toc26380730)

[Activities 6](#_Toc26380731)

[Purpose 7](#_Toc26380732)

[Relationship Matrix 7](#_Toc26380733)

[Enhanced Entity Relationship Diagram 8](#_Toc26380734)

[Business Rule 9](#_Toc26380735)

[Referential Integrity Constraint 11](#_Toc26380736)

[Functional Dependency Diagram and Normalization 12](#_Toc26380737)

[Structured Query Language 15](#_Toc26380738)

[Creation 15](#_Toc26380739)

[Customer 15](#_Toc26380740)

[Domestic Customer 15](#_Toc26380741)

[International Customer 15](#_Toc26380742)

[Employee 16](#_Toc26380743)

[Mechanic 16](#_Toc26380744)

[Sales Person 16](#_Toc26380745)

[Employee Skill 17](#_Toc26380746)

[Order 17](#_Toc26380747)

[Vehicle 17](#_Toc26380748)

[Order Line 18](#_Toc26380749)

[Service 18](#_Toc26380750)

[Supplier 19](#_Toc26380751)

[Part 19](#_Toc26380752)

[Insertion 19](#_Toc26380753)

[Customer 19](#_Toc26380754)

[Domestic Customer 20](#_Toc26380755)

[International Customer 21](#_Toc26380756)

[Employee 21](#_Toc26380757)

[Employee Skills 22](#_Toc26380758)

[Sales Person 24](#_Toc26380759)

[Mechanic 24](#_Toc26380760)

[Vehicles 25](#_Toc26380761)

[Order 26](#_Toc26380762)

[Order Line 26](#_Toc26380763)

[Supplier 27](#_Toc26380764)

[Part 28](#_Toc26380765)

[Service 29](#_Toc26380766)

[Queries and Result 30](#_Toc26380767)

[List All Domestic Customers. 30](#_Toc26380768)

[List all the employee IDs, names, and dates of employees who were hired before the end of May 2015. 30](#_Toc26380769)

[List all employee ID and names of employees who have more than 2 skills. 31](#_Toc26380770)

[List all Japanese customers. 31](#_Toc26380771)

[List customer ID, customer name, customer type, and how many orders each customer placed. 32](#_Toc26380772)

[Which vehicle ID, make, model, year, are priced above the average cost of our cars, and how much are they? 32](#_Toc26380773)

[Find out vehicle make, model which were made after 2015, and all the service info from the vehicle and service table. 33](#_Toc26380774)

[Which mechanic has an hourly rate of higher than $40, and works with parts that cost less than $3,000 and what are those parts called? List Mechanic employee ID and hourly rate. 33](#_Toc26380775)

[Find out the sales person's average rate of commission of total vehicle cost. 34](#_Toc26380776)

[How many cars are priced less than the average price of our cars? 34](#_Toc26380777)

[What percentage of cars are priced less than the average price? Round to 2 decimal places. 34](#_Toc26380778)

[List the customer name, customer type, customer phone number and order number for all customers. Include customer information for customer that don’t have an order. 35](#_Toc26380779)

[Show the information required to make an invoice for order number 10. 35](#_Toc26380780)

[Show all customers who have placed an order. 36](#_Toc26380781)

[What is the name and address for the customer who placed order number 6? 36](#_Toc26380782)

[Show orders for vehicle id 2, display Order Id and Order Total. 36](#_Toc26380783)

# Introduction

Case Scenario

Cool Car Company is a company that has been around for many years specializing in selling vehicles. However, as the years went on, competition had risen, and other companies were responding faster than Cool Car Company in adopting more recent technologies, which resulted in less errors and cost and higher sales for them. Much of this issue seemed to have been caused by the company’s usage of the outdated traditional file processing system. The traditional file processing system has its disadvantages of program data dependence, duplication of data, limited sharing, lengthy development time, and high program maintenance. Having identified this as their primary issue, Cool Car Company went forward to implement a database approach for the company. The database approach has its many advantages to help the company such as program data independence, planned redundancy of data, improved data consistency, better data sharing and more.

## Business Needs

With the product line being the place where most companies compete with one another, Cool Car Company finds it crucial to be able to keep track of their revenue and expenses correctly and effortlessly. They are requesting a system to be able to identify the types of customers they have, the salespeople who sell these vehicles, the types of vehicle they sell, and the services provided for those vehicles by mechanics.

## Requirements

The management system should deploy a centralized database as a core to store, manage, and operate data. The database itself should be independent which only the database management system (DBMS) has the accessibility to the database. The DBMS connects with users’ applications as well. The entities in the database all should relate to the business. Each entity should have a unique name. The entities in the business should be logically related. The relationship between the entities should represent the exact business activities.

## Functions

The system will function to show what types of customers Cool Car Company has and what types of employees are selling or working on their vehicle. Also, the system will give detailed descriptions about the vehicles make, model, year, and cost as well as the orders made by customers. It will also function to show the services provided to each vehicle by the, mechanic

## Activities

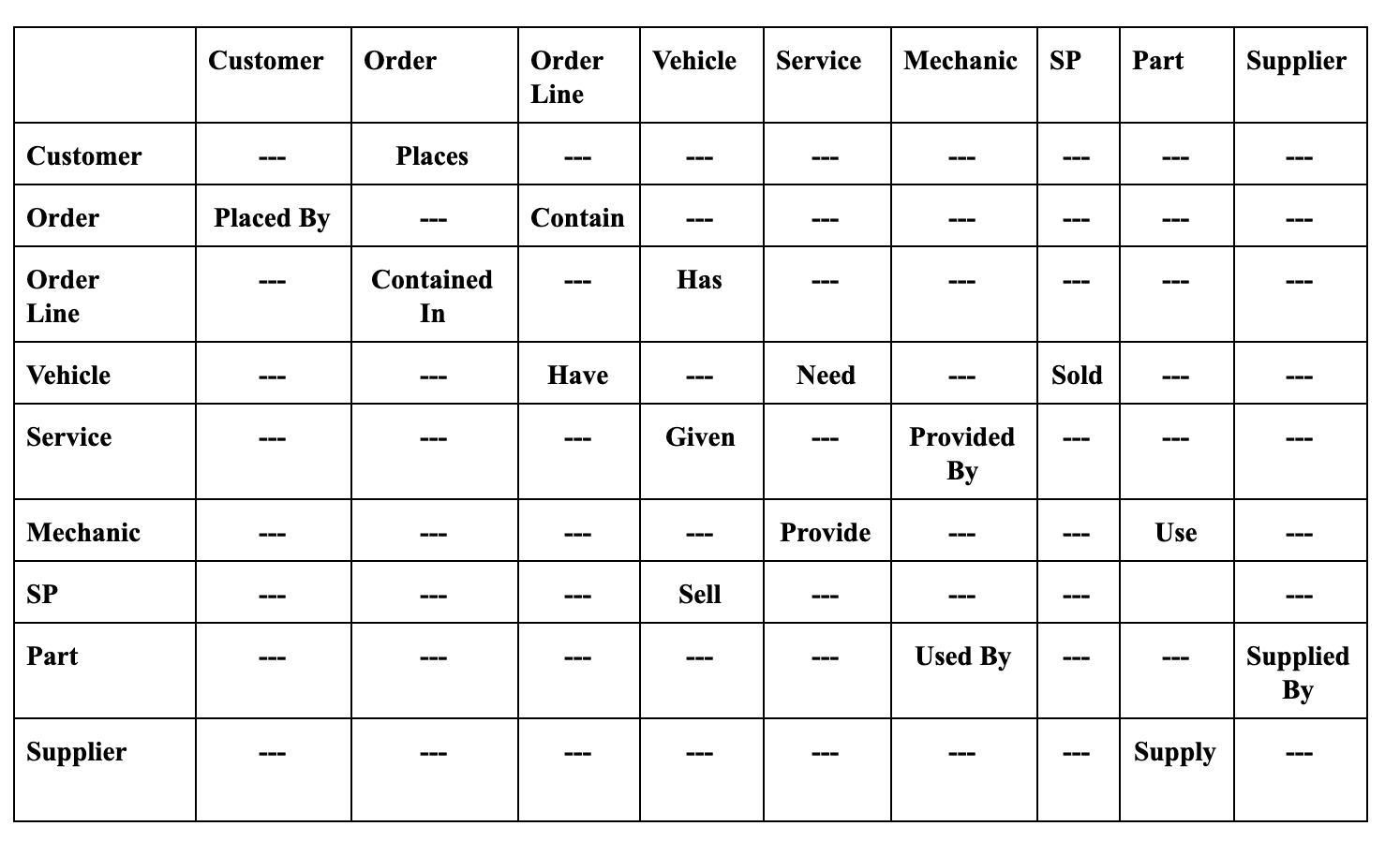
The database administrator has access to data definition language in order to create, alter, and drop tables. These privileges are granted to the database administrator as he or she would be most knowledgeable about constructing and organizing the database. He and the other developers also have access to the data manipulation commands to insert, update, modify, and query the database. The other developers need to be able to take user data from the frontend and insert, update, or delete records in the database.  
Entities

The entities that are needed to create the system are Customer supertype with subtypes of International and Domestic subtypes. Employee supertype with Mechanic and Salesperson subtypes. Vehicle, Order, and Order Line, which will serve as an associative entity between Vehicle and Orders. Service entity will serve as an associative entity of Vehicle and Mechanic, Part, and Supplier.

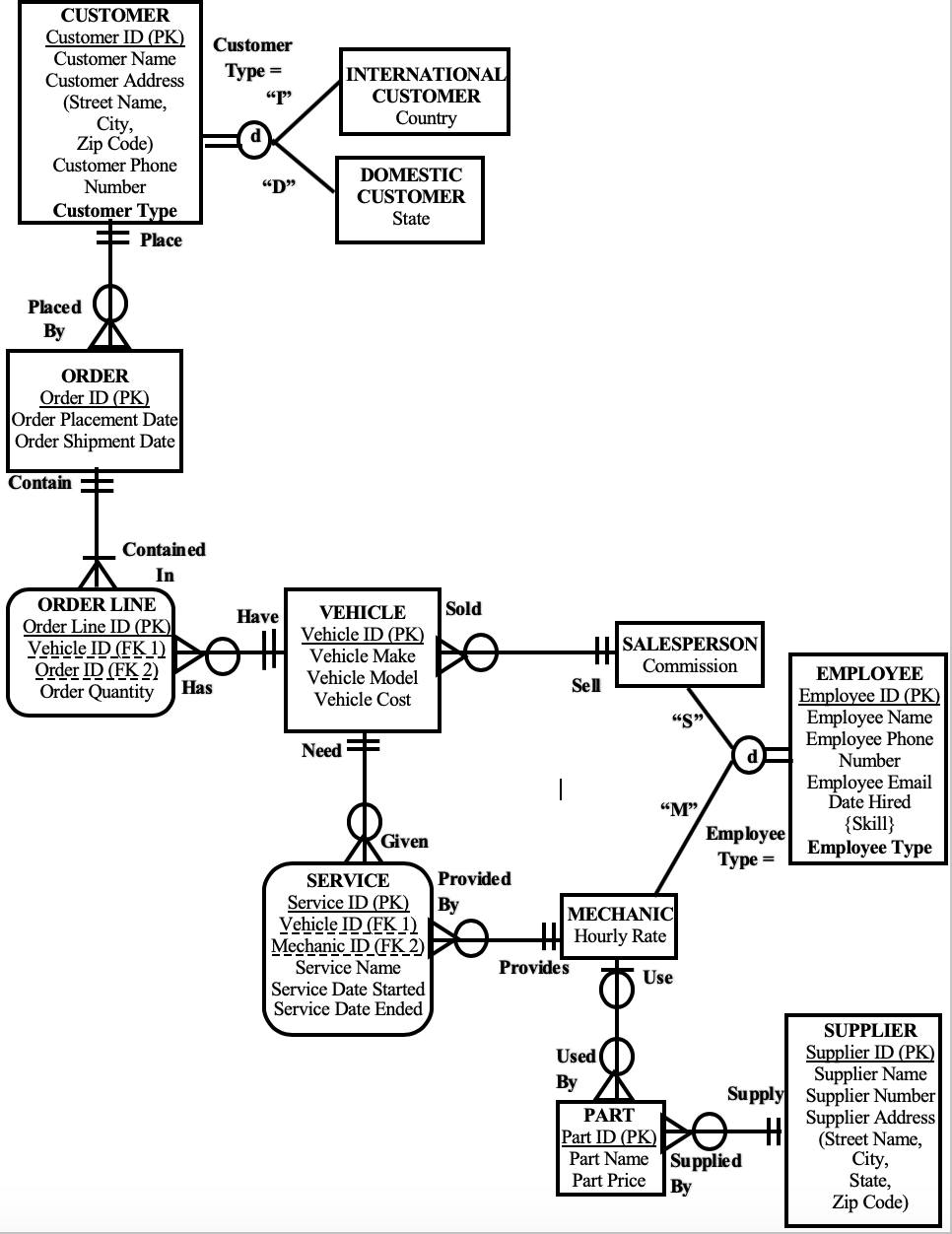
## Purpose

The purpose of this technological shift to the database approach is to keep track of revenue and expenses accurately and effortlessly, which will then help Cool Cars Company continue to be competitive in the market and continue to have highly satisfied and happy customers.

# Relationship Matrix

****

# Enhanced Entity Relationship Diagram

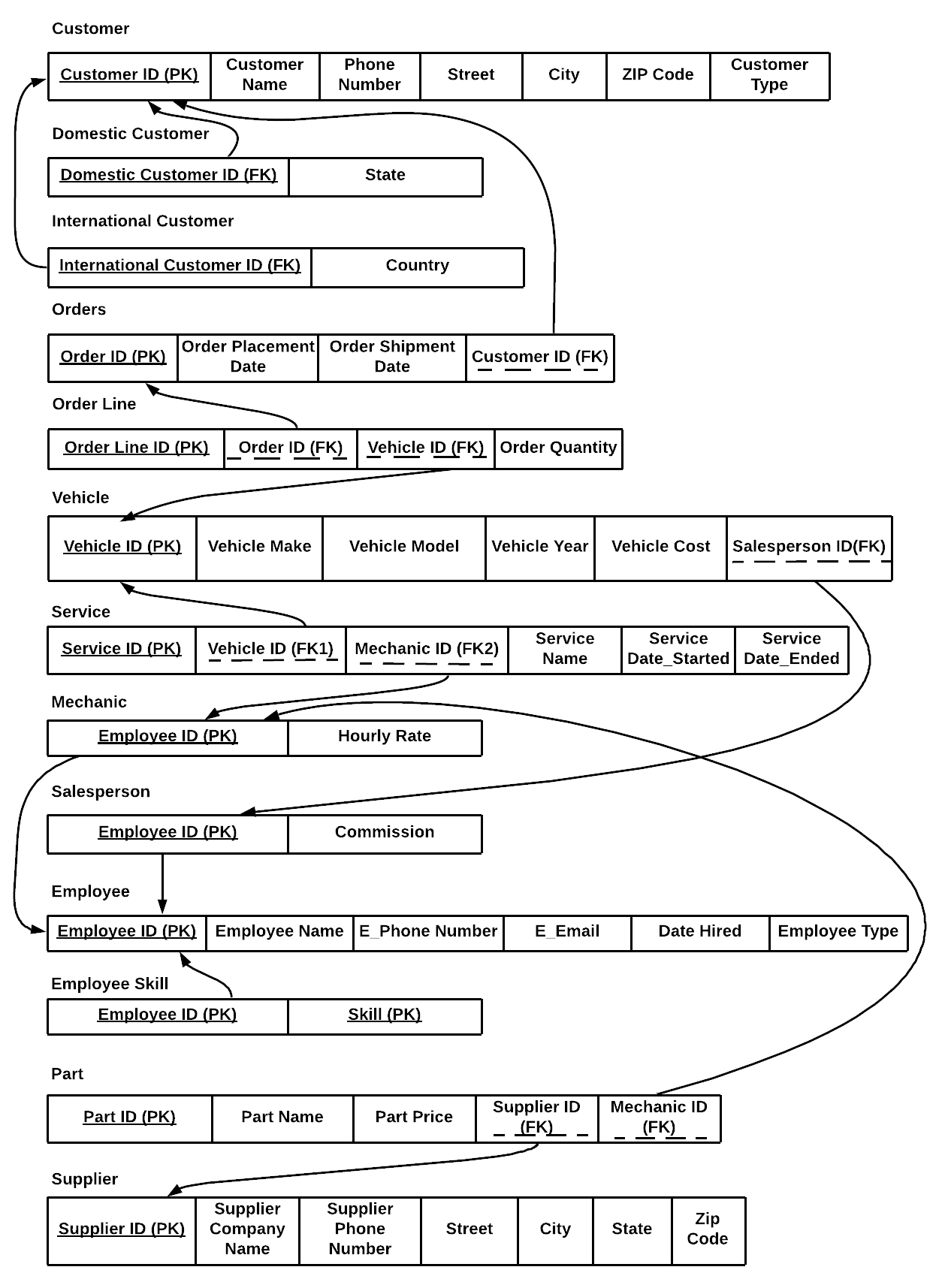
****

# Business Rule

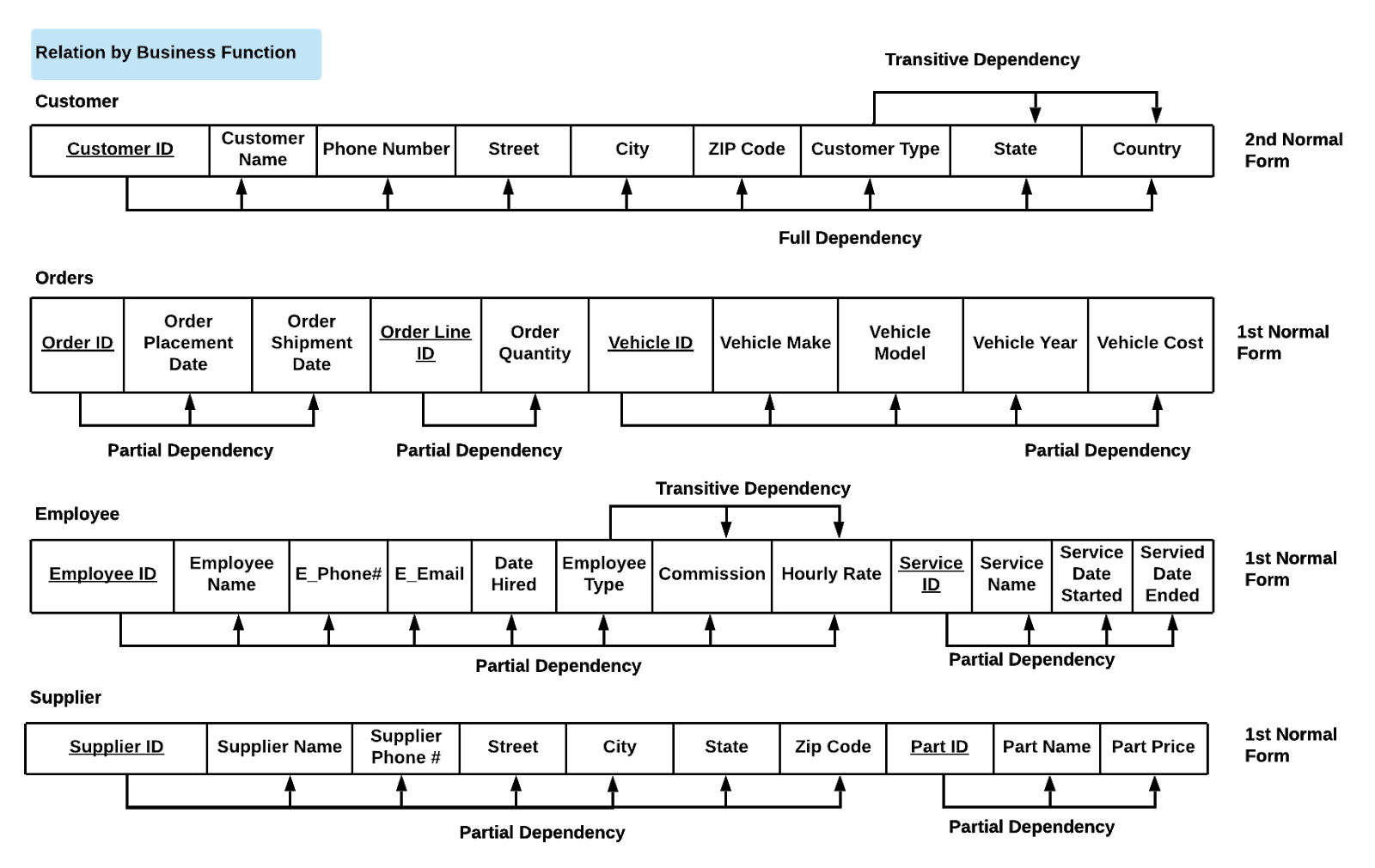
1. Each CUSTOMERmust be a DOMESTIC CUSTOMER or an INTERNATIONAL CUSTOMER but never both at the same time.
2. Each CUSTOMER may place one or more ORDERS**.** Each ORDER is placed by one and only one CUSTOMER.
3. Each ORDERmust contain one or more ORDER LINES. Each ORDER LINE must contain in one and only one ORDER.
4. Each ORDERmust include one or more VEHICLES**.** Each VEHICLEmay be included in one or more orders.
5. Each ORDER LINE has one and only one VEHICLE**.** Each VEHICLEmay have one or more ORDER LINES.
6. Each EMPLOYEEmust be a SALESPERSONor a MECHANIC but never both at the same time.
7. Each VEHICLE may be worked on by one or more MECHANICS. Each MECHANIC may work on one or more VEHICLE.
8. Each VEHICLEmay need many types of SERVICE**.** Each SERVICEmust be given to one and only one VEHICLE.
9. EachSERVICE must be provided by one and only one MECHANIC**.** Each MECHANICmay provide many SERVICES.
10. Each SALESPERSON may sell one or more VEHICLE**.** Each VEHICLEmust be sold by one SALESPERSON**.**
11. Each MECHANIC may use one or more PARTS**.** Each PART may be used by one and only one MECHANIC.

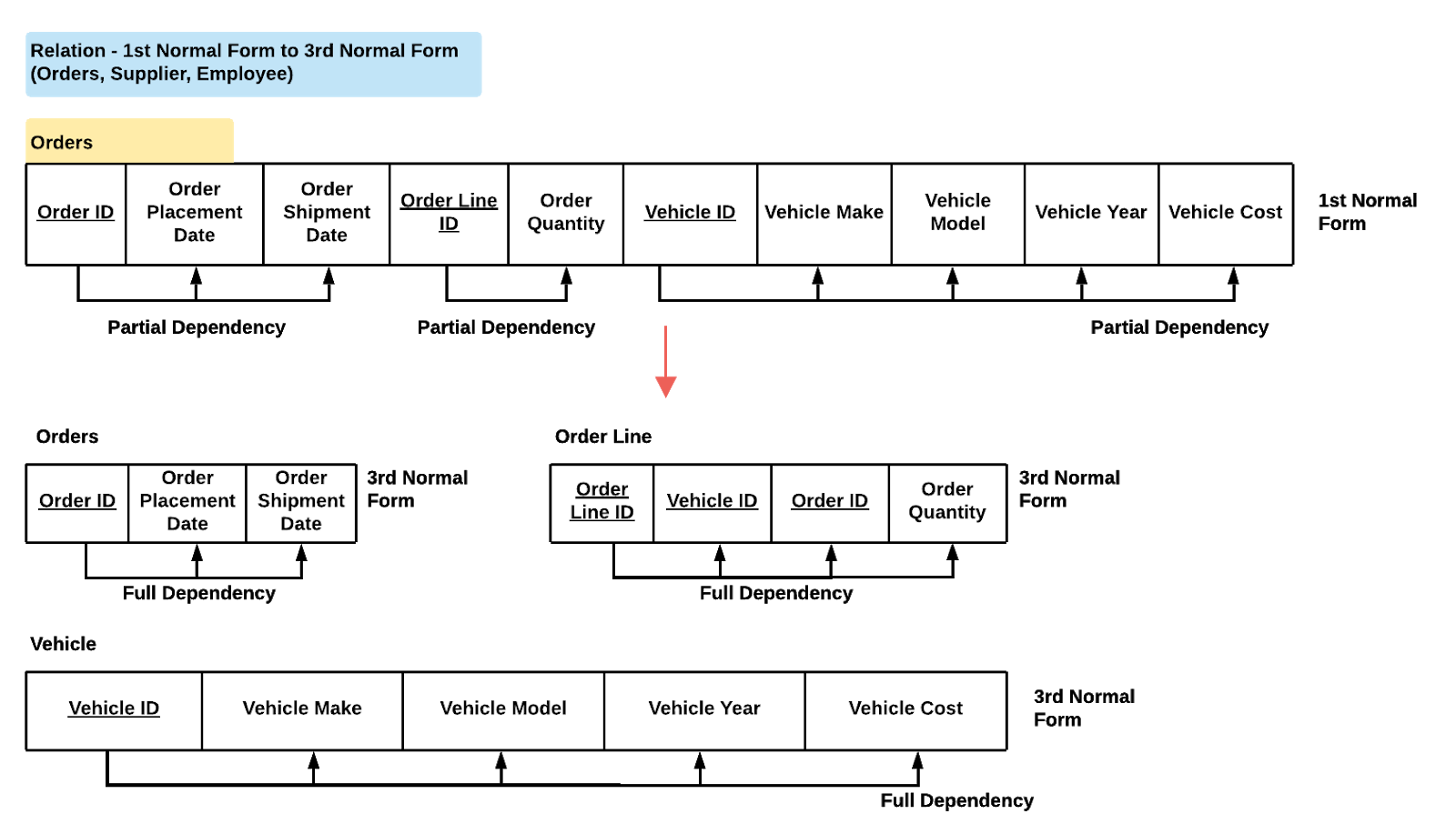
Each PARTmust be supplied by one and only one SUPPLIER**.** Each SUPPLIERmay supply one or more PARTS.

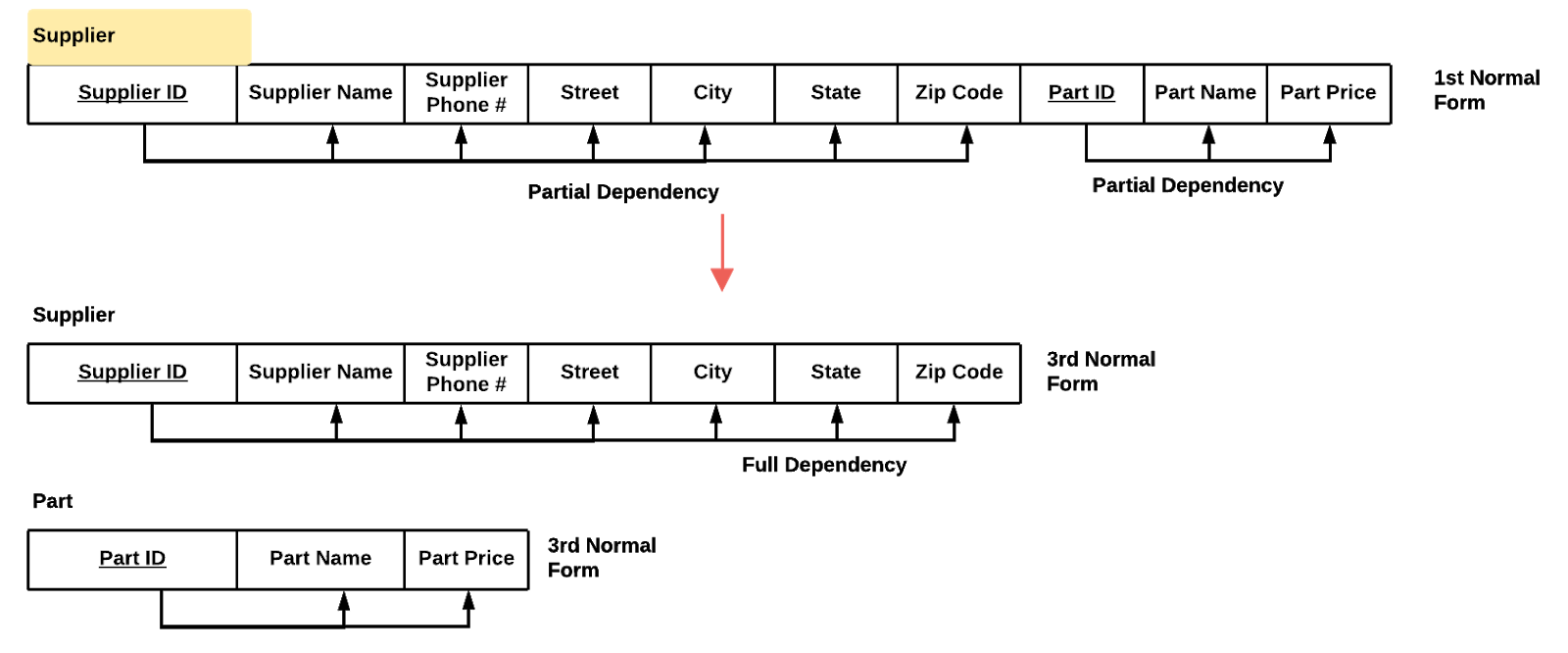
# Referential Integrity Constraint

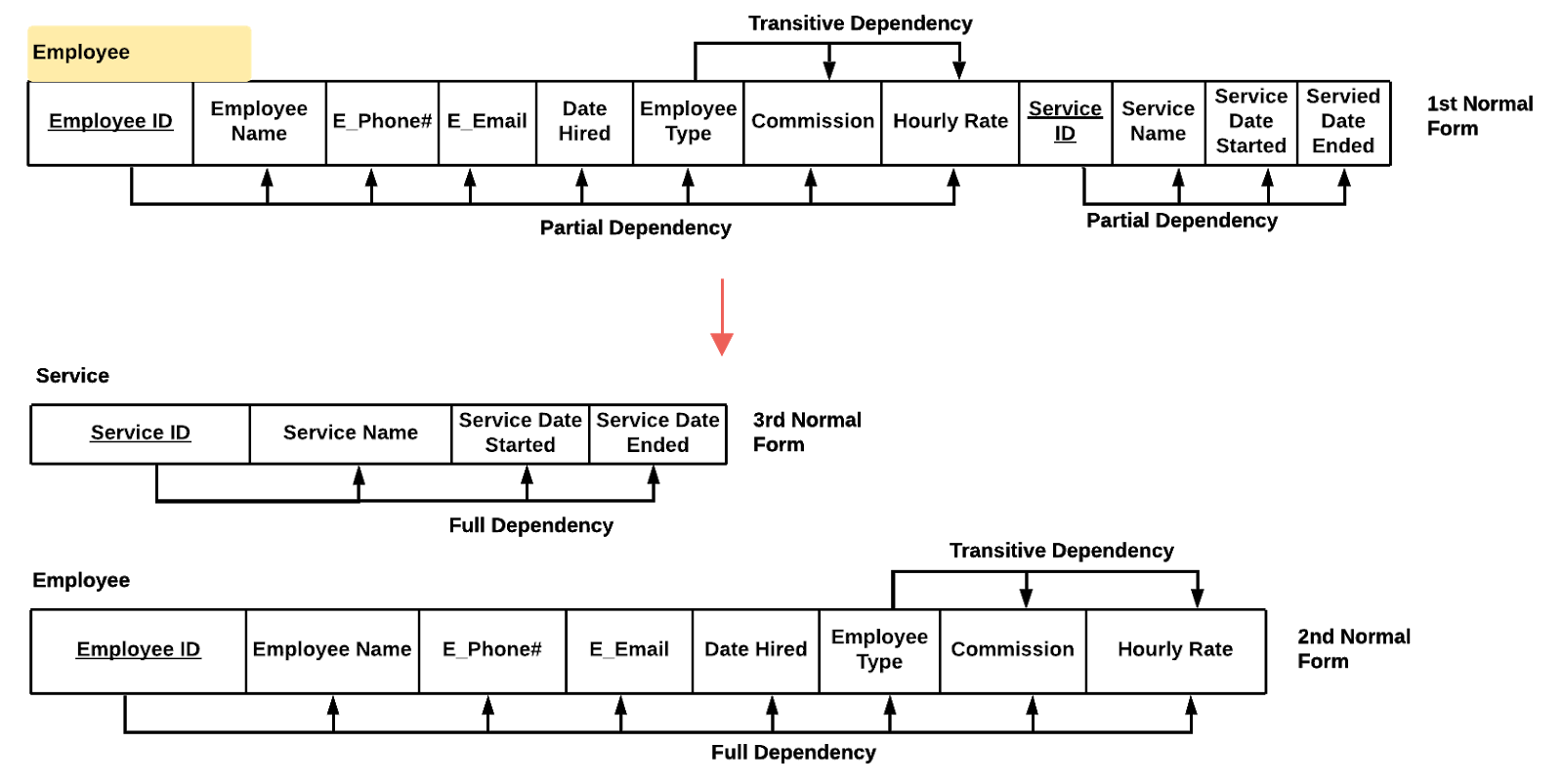


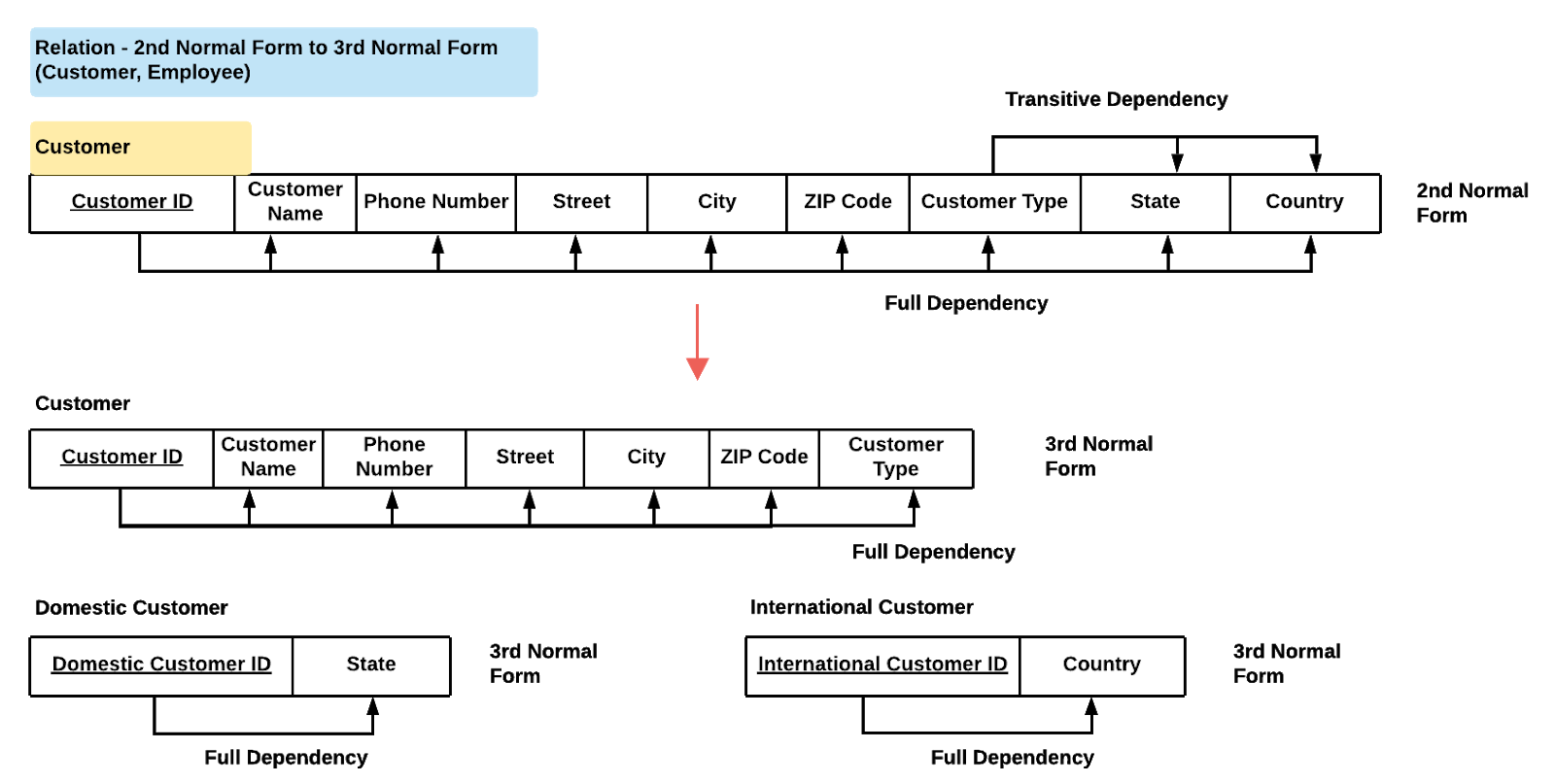
# Functional Dependency Diagram and Normalization

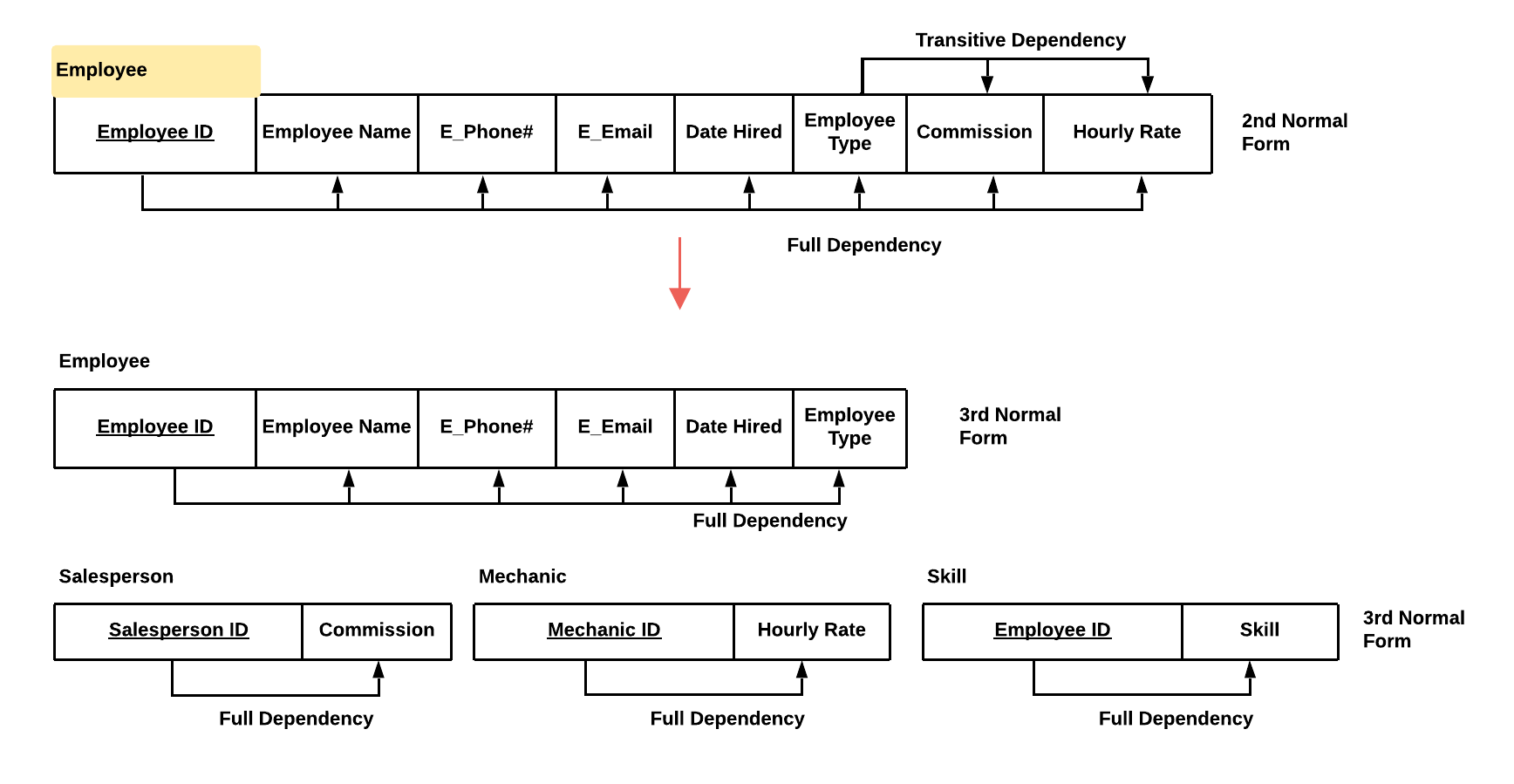
****

****

****

****

****



# Structured Query Language

## Creation

### Customer

-- customer table

CREATE TABLE IF NOT EXISTS customer

(

customer\_id INTEGER NOT NULL,

customer\_name TEXT NOT NULL,

customer\_phone\_number VARCHAR(30) NOT NULL,

customer\_street TEXT NOT NULL,

customer\_city TEXT NOT NULL,

customer\_zip\_code VARCHAR(20) NOT NULL,

customer\_type TEXT NOT NULL,

PRIMARY KEY (customer\_id),

CHECK ( customer\_type = 'international' or customer\_type = 'domestic')

);

### Domestic Customer

-- domestic customer table subtype of customer

CREATE TABLE IF NOT EXISTS domestic\_customer

(

domestic\_customer\_id INTEGER NOT NULL,

domestic\_customer\_state VARCHAR(2) NOT NULL,

PRIMARY KEY (domestic\_customer\_id),

FOREIGN KEY (domestic\_customer\_id) REFERENCES customer (customer\_id)

);

### International Customer

-- international customer table with subtype of customer

CREATE TABLE IF NOT EXISTS international\_customer

(

international\_customer\_id INTEGER NOT NULL,

international\_customer\_country TEXT NOT NULL,

PRIMARY KEY (international\_customer\_id),

FOREIGN KEY (international\_customer\_id) REFERENCES customer (customer\_id)

);

### Employee

-- create employee table

CREATE TABLE IF NOT EXISTS employee

(

employee\_ID INTEGER NOT NULL,

employee\_name TEXT NOT NULL,

employee\_phone\_number VARCHAR(30) NOT NULL,

employee\_email TEXT NOT NULL,

employee\_date\_hired DATE DEFAULT CURRENT\_DATE NOT NULL,

employee\_type TEXT NOT NULL,

PRIMARY KEY (employee\_ID),

CHECK ( employee\_type = 'sales' or employee\_type = 'mechanic')

);

### Mechanic

-- create mechanic table subtype of employee

CREATE TABLE IF NOT EXISTS mechanic

(

employee\_id INTEGER NOT NULL,

mechanic\_hourly\_rate FLOAT NOT NULL,

PRIMARY KEY (employee\_id),

FOREIGN KEY (employee\_id) REFERENCES employee (employee\_ID)

);

### Sales Person

-- sales person table subtype of employee

CREATE TABLE IF NOT EXISTS sales\_person

(

employee\_id INTEGER NOT NULL,

sales\_person\_commission FLOAT NOT NULL,

PRIMARY KEY (employee\_id),

FOREIGN KEY (employee\_id) REFERENCES employee (employee\_ID)

);

### Employee Skill

-- employee skill multi valued table

CREATE TABLE IF NOT EXISTS employee\_skill

(

employee\_id INTEGER NOT NULL,

employee\_skill TEXT NOT NULL,

PRIMARY KEY (employee\_id, employee\_skill),

FOREIGN KEY (employee\_id) REFERENCES employee (employee\_ID)

);

### Order

-- orders table

CREATE TABLE IF NOT EXISTS orders

(

order\_id INTEGER NOT NULL,

order\_placement\_date DATE DEFAULT CURRENT\_DATE NOT NULL,

order\_shipment\_date DATE,

customer\_id INTEGER NOT NULL,

primary key (order\_id),

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id)

);

### Vehicle

-- vehicle table

CREATE TABLE IF NOT EXISTS vehicle

(

vehicle\_id INTEGER NOT NULL,

vehicle\_make TEXT NOT NULL,

vehicle\_model TEXT NOT NULL,

vehicle\_year INT NOT NULL,

vehicle\_cost FLOAT NOT NULL,

sales\_person\_id INTEGER,

PRIMARY KEY (vehicle\_id),

FOREIGN KEY (sales\_person\_id) REFERENCES sales\_person (employee\_id)

);

### Order Line

-- order line table

CREATE TABLE IF NOT EXISTS order\_line

(

order\_line\_id INTEGER NOT NULL,

order\_id INTEGER NOT NULL,

vehicle\_id INTEGER NOT NULL,

order\_quantity INT NOT NULL,

PRIMARY KEY (order\_line\_id),

FOREIGN KEY (order\_id) REFERENCES orders (order\_id),

FOREIGN KEY (vehicle\_id) REFERENCES vehicle (vehicle\_id)

);

### Service

-- service table

CREATE TABLE IF NOT EXISTS service

(

service\_id INTEGER NOT NULL,

vehicle\_id INTEGER NOT NULL,

service\_mechanic\_id INTEGER NOT NULL,

service\_name TEXT NOT NULL,

service\_date\_started DATE DEFAULT CURRENT\_DATE NOT NULL,

service\_date\_end DATE,

PRIMARY KEY (service\_id),

FOREIGN KEY (vehicle\_id) REFERENCES vehicle (vehicle\_id),

FOREIGN KEY (service\_mechanic\_id) REFERENCES mechanic (employee\_id)

);

### Supplier

-- supplier table

CREATE TABLE IF NOT EXISTS supplier

(

supplier\_id INTEGER NOT NULL,

company\_name TEXT NOT NULL,

supplier\_phone\_number VARCHAR(30) NOT NULL,

supplier\_street TEXT NOT NULL,

supplier\_city TEXT NOT NULL,

supplier\_state VARCHAR(2),

supplier\_zip\_code INT NOT NULL,

PRIMARY KEY (supplier\_id)

);

### Part

-- part table

CREATE TABLE IF NOT EXISTS part

(

part\_id INTEGER NOT NULL,

part\_name TEXT NOT NULL,

part\_price FLOAT NOT NULL,

supplier\_id INTEGER NOT NULL,

mechanic\_id INTEGER NOT NULL,

primary key (part\_id),

FOREIGN KEY (supplier\_id) REFERENCES supplier (supplier\_id),

FOREIGN KEY (mechanic\_id) REFERENCES mechanic (employee\_id)

);

## Insertion

### Customer

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Aaron Fuller', 7383670449, 'Pennsylvania Avenue', 19002, 'Grant', 'domestic');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Antonio Gross', 6674435742, 'Main customer\_street South', 20744, 'Higganum', 'domestic');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Celia Young', 4238425859, 'Church customer\_street', 30741, 'Wood River', 'domestic');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Wilson Goodwin', 5548581699, 'Fairview Avenue', 'Roy', 45840, 'domestic');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Leo Gill', 7184528143, 'Wall customer\_street', 'Montross', 07026, 'domestic');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Reginald Hogan', 2833381181, 'Hudson customer\_street', 'Marco Island', 11542, 'international');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Elijah Joseph', 9239375532, 'Highland Avenue', 'Virgelle', 27526, 'international');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Katie Brady', 3958198655, 'Dogwood Drive', 'Bienville', 16506, 'international');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Mona Drake', 2179153374, 'Heritage Drive', 'Carthage', 08527, 'international');

INSERT INTO customer(customer\_name, customer\_phone\_number, customer\_street, customer\_city, customer\_zip\_code,

customer\_type)

VALUES ('Derrick Summers', 9768634277, '4th customer\_street', 'Floyd', 07726, 'international');

### Domestic Customer

INSERT INTO domestic\_customer(domestic\_customer\_id, domestic\_customer\_state)

VALUES (1, 'CA');

INSERT INTO domestic\_customer(domestic\_customer\_id, domestic\_customer\_state)

VALUES (2, 'NY');

INSERT INTO domestic\_customer(domestic\_customer\_id, domestic\_customer\_state)

VALUES (3, 'NV');

INSERT INTO domestic\_customer(domestic\_customer\_id, domestic\_customer\_state)

VALUES (4, 'MA');

INSERT INTO domestic\_customer(domestic\_customer\_id, domestic\_customer\_state)

VALUES (5, 'CA');

### International Customer

INSERT INTO international\_customer(international\_customer\_id, international\_customer\_country)

VALUES (6, 'Chile');

INSERT INTO international\_customer(international\_customer\_id, international\_customer\_country)

VALUES (7, 'France');

INSERT INTO international\_customer(international\_customer\_id, international\_customer\_country)

VALUES (8, 'Greece');

INSERT INTO international\_customer(international\_customer\_id, international\_customer\_country)

VALUES (9, 'Japan');

INSERT INTO international\_customer(international\_customer\_id, international\_customer\_country)

VALUES (10, 'Mexico');

### Employee

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Abraham Kennedy', 5974784359, 'tmaek@me.com', '2017-04-01', 'sales');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Lorene Cannon', 6128891387, 'wmszeliga@mac.com', '2011-04-01', 'sales');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Ronald Saunders', 7203219940, 'johndo@yahoo.ca', '2012-04-01', 'sales');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Nicole Webb', 3698455909, 'quinn@gmail.com', '2013-04-01', 'sales');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Beatrice Morrison', 3188742111, 'shang@comcast.net', '2017-04-01', 'sales');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Charlene Williams', 6958416474, 'stecoop@optonline.net', '2014-04-01', 'mechanic');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Arlene Lyons', 6194572578, 'speeves@verizon.net', '2015-04-01', 'mechanic');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Antonia Barton', 4488908506, 'mwitte@optonline.net', '2012-04-01', 'mechanic');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Cristina Chandler', 3754965474, 'squirrel@yahoo.ca', '2014-04-01', 'mechanic');

INSERT INTO employee(employee\_name, employee\_phone\_number, employee\_email, employee\_date\_hired, employee\_type)

VALUES ('Ronnie Ferguson', 2932422652, 'skippy@yahoo.com', '2018-04-01', 'mechanic');

### Employee Skills

INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (1, 'Product Knowledge');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (1, 'Strategic Prospecting');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (1, 'Active Listening');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (2, 'Product Knowledge');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (2, 'Communication');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (2, 'Qualification Questioning');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (3, 'Objection Handling');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (3, 'Strategic Prospecting');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (3, 'Rapport Building on the Call');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (3, 'Qualification Questioning');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (4, 'Product Knowledge');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (4, 'Buyer-Seller Agreement');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (5, 'Objection Handling');  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (5, 'Objection Prevention');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (5, 'Communication');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (6, 'Diagnostic');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (6, 'Customer service');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (6, 'Problem-solving');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (6, 'Technical aptitude');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (7, 'Customer service');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (7, 'Diagnostic');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (7, 'Technical aptitude');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (7, 'Problem-solving');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (8, 'Customer service');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (8, 'Diagnostic');

INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (9, 'Problem-solving');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (9, 'Technical aptitude');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (10, 'Customer service');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (10, 'Diagnostic');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (10, 'Technical aptitude');  
  
INSERT INTO employee\_skill (employee\_id, employee\_skill)  
VALUES (10, 'Problem-solving');

### Sales Person

INSERT INTO sales\_person (employee\_id, sales\_person\_commission)

VALUES (1, 5634.12);

INSERT INTO sales\_person (employee\_id, sales\_person\_commission)

VALUES (2, 5799.83);

INSERT INTO sales\_person (employee\_id, sales\_person\_commission)

VALUES (3, 8514.22);

INSERT INTO sales\_person (employee\_id, sales\_person\_commission)

VALUES (4, 8730.48);

INSERT INTO sales\_person (employee\_id, sales\_person\_commission)

VALUES (5, 4355.93);

### Mechanic

INSERT INTO mechanic (employee\_id, mechanic\_hourly\_rate)

VALUES (6, 40.74);

INSERT INTO mechanic (employee\_id, mechanic\_hourly\_rate)

VALUES (7, 37.45);

INSERT INTO mechanic (employee\_id, mechanic\_hourly\_rate)

VALUES (8, 50.87);

INSERT INTO mechanic (employee\_id, mechanic\_hourly\_rate)

VALUES (9, 43.71);

INSERT INTO mechanic (employee\_id, mechanic\_hourly\_rate)

VALUES (10, 30.86);

### Vehicles

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'TT1', 2015, 32000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'TT2', 2017, 34000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'TT3', 2018, 39000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'ST1', 2014, 42000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'ST2', 2016, 45000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'GT1', 2012, 72000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'GT2', 2014, 112000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'GT3', 2016, 202000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'FF', 2019, 332000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost)

VALUES ('Cool', 'FFS', 2020, 352000);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('Giorgia', 'NINJA', 2017, 417826.20, 1);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('Pugito', 'Tumbler', 2018, 145842.20, 2);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('Cocanita', 'Titan', 2018, 18721.22, 3);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('EA', 'Cheetah', 2019, 18721.22,3);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('UBOAT', 'Ship', 2019, 24892.20, 4);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('HIT', 'Rocket', 2019, 24892.20, 4);

INSERT INTO vehicle (vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost, sales\_person\_id)

VALUES ('SHIP', 'Delta', 2019, 5782.20, 5);

### Order

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (1, '2018-03-15', '2018-03-17', 1);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (2, '2018-03-15', '2018-03-17', 1);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (3, '2018-04-25', '2018-05-3', 2);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (4, '2018-05-06', '2018-05-06', 3);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (5, '2018-08-29', '2018-09-03', 3);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (6, '2018-010-17', '2018-10-27', 4);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (7, '2018-11-05', '2018-11-07', 5);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (8, '2018-12-05', '2018-12-07', 6);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (9, '2019-04-16', '2019-04-17', 7);

INSERT INTO orders (order\_id, order\_placement\_date, order\_shipment\_date, customer\_id)

VALUES (10, '2019-04-17', '2019-04-20', 8);

### Order Line

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (1, 4, 8);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (2, 1, 24);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (3, 1, 32);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (4, 4, 8);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (5, 6, 2);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (6, 2, 40);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (7, 3, 3);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (8, 5, 1);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (9, 3, 5);

INSERT INTO order\_line (order\_id, vehicle\_id, order\_quantity)

VALUES (10, 5, 4);

### Supplier

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('Audi', 6387622055, '71 Livingston St', 'Charlotte', 'NC', 28205);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('Bosche', 7017801479, '87 Oak Valley Road', 'Brainerd', 'MN', 56401);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('Titan', 9468600631, '7897 Roosevelt St.', 'Massapequa', 'NY', 11758);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('YOKOHOMA', 5543431211, '784 High Noon St.', 'Solon', 'OH', 44139);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('YAMAZAKI', 9297542285, '7922 Roosevelt Ave.', 'Ridgewood', 'NJ', 07450);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('S1 moto sport club', 3887319886, '8904 Armstrong Lane', 'Montclair', 'NJ', 07042);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('Bazuka', 5262312635, '44 Morris Ave', 'Westport', 'CT', 06880);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('AMD', 6999955820, '935 St Margarets Drive', 'Millington', 'TN', 38053);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('GRID', 3964893814, '8877 Snake Hill supplier\_street', 'Fall River', 'MA', 02720);

INSERT INTO supplier (company\_name, supplier\_phone\_number, supplier\_street, supplier\_city, supplier\_state,

supplier\_zip\_code)

VALUES ('TOYO', 2136244983, '8135 South Ave', 'Ellicott supplier\_city', 'MD', 21042);

### Part

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Audi sport suspension', 1800, 1, 6);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Bosch moto oil', 89, 2, 6);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Drift LSD', 3200, 3, 7);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('performance tires', 1400, 4, 7);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Anti-roll bar', 900, 5, 8);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Brambo brake system', 3000, 6, 8);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('B Turbo', 5600, 7, 9);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('ECU unit', 4000, 8, 9);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Suports package', 7000, 9, 10);

INSERT INTO part (part\_name, part\_price, supplier\_id, mechanic\_id)

VALUES ('Grip tires', 900, 10, 10);

### Service

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (6, 6, 'suspension check', '2019-3-12', '2019-3-12');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (2, 6, 'oil change', '2019-3-21', '2019-3-21');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (3, 7, 'oil change', '2019-3-25', '2019-3-25');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (2, 7, 'air filter upgrade', '2019-4-02', '2019-4-22');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (1, 8, 'coolant change', '2019-4-12', '2019-4-13');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (6, 8, 'transmission replace', '2019-4-15', '2019-4-19');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (1, 9, 'suspension change', '2019-4-22', '2019-4-24');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (4, 9, 'CV joint change', '2019-5-8', '2019-5-12');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

VALUES (5, 10, 'turbo maintenance', '2019-5-18', '2019-5-19');

INSERT INTO service(vehicle\_id, service\_mechanic\_id, service\_name, service\_date\_started, service\_date\_end)

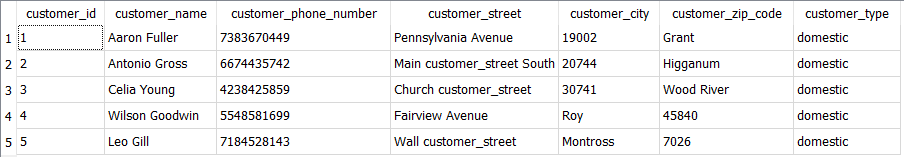
VALUES (3, 10, 'oil change', '2019-6-22', '2019-6-22');

## Queries and Results

### List All Domestic Customers.

SELECT \*  
FROM customer

WHERE customer\_type = 'domestic';



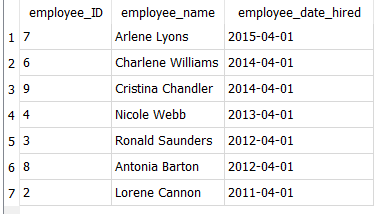
### List all the employee IDs, names, and dates of employees who were hired before the end of May 2015.

SELECT employee\_id, employee\_name, employee\_date\_hired

FROM employee

WHERE employee\_date\_hired <= '2015-05-30'

ORDER BY employee\_date\_hired DESC;

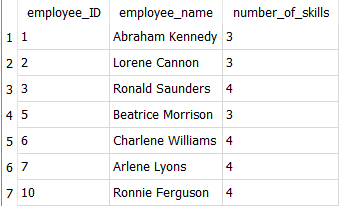


### List all employee ID and names of employees who have more than 2 skills.

SELECT employee.employee\_id, employee.employee\_name, COUNT(employee\_skill.employee\_id) AS number\_of\_skills

FROM employee,  
 employee\_skill WHERE employee\_skill.employee\_id = employee.employee\_id GROUP BY employee.employee\_id

HAVING count(employee\_skill.employee\_id) > 2;



### List all Japanese customers.

SELECT customer\_id,  
 international\_customer\_country,  
 customer\_name,  
 customer\_phone\_number,  
 customer\_street,  
 customer\_city,  
 customer\_zip\_code,  
 customer\_type

FROM international\_customer

INNER JOIN customer ON international\_customer.international\_customer\_id = customer.customer\_id

WHERE international\_customer\_country = 'Japan';



### List customer ID, customer name, customer type, and how many orders each customer placed.

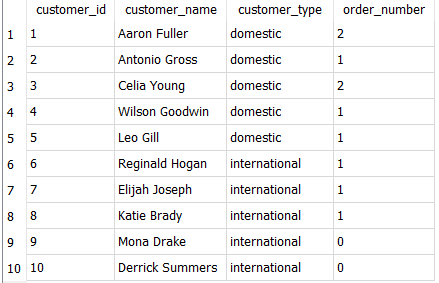
SELECT customer.customer\_id, customer\_name, customer\_type, COUNT(orders.customer\_id) AS order\_number

FROM customer

LEFT JOIN orders

ON customer.customer\_id = orders.customer\_id

GROUP BY customer.customer\_id;



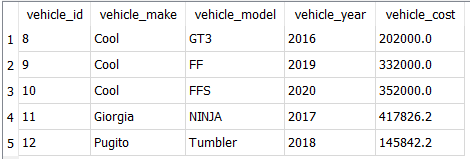
### Which vehicle ID, make, model, year, are priced above the average cost of our cars, and how much are they?

SELECT vehicle\_id, vehicle\_make, vehicle\_model, vehicle\_year, vehicle\_cost FROM vehicle

WHERE vehicle\_cost >

(  
 SELECT AVG(vehicle\_cost)  
 FROM vehicle

);



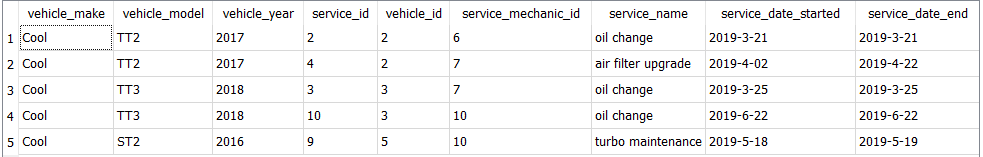
### Find out vehicle make, model which were made after 2015, and all the service info from the vehicle and service table.

SELECT vehicle.vehicle\_make, vehicle.vehicle\_model, vehicle.vehicle\_year, service.\*  
FROM vehicle

INNER JOIN service ON vehicle.vehicle\_id = service.vehicle\_id

WHERE vehicle\_year > '2015'

ORDER BY service.vehicle\_id;



### Which mechanic has an hourly rate of higher than $40, and works with parts that cost less than $3,000 and what are those parts called? List Mechanic employee ID and hourly rate.

SELECT employee\_id, mechanic\_hourly\_rate, part\_name, part\_price

FROM mechanic

LEFT JOIN part ON employee\_id = mechanic\_id

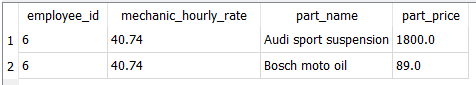
WHERE mechanic\_hourly\_rate > 40

AND employee\_id =  
 (  
 SELECT mechanic\_id

FROM part

WHERE part\_price < 3000

)  
ORDER BY employee\_id;



### Find out the sales person's average rate of commission of total vehicle cost.

SELECT AVG(sales\_person\_commission) / AVG(vehicle\_cost)  
FROM sales\_person, vehicle;



### How many cars are priced less than the average price of our cars?

SELECT COUNT(\*)  
FROM vehicle

WHERE vehicle\_cost <

(  
 SELECT AVG(vehicle\_cost)  
 FROM vehicle

)



### What percentage of cars are priced less than the average price? Round to 2 decimal places.

SELECT (ROUND(((CAST((SELECT COUNT(\*)  
FROM vehicle

WHERE vehicle\_cost < (SELECT AVG(vehicle\_cost) FROM vehicle)) AS FLOAT) /  
 CAST((SELECT COUNT(\*) FROM vehicle) AS FLOAT))) \* 100, 2)) AS cars\_priced\_less\_than\_average;



### List the customer name, customer type, customer phone number and order number for all customers. Include customer information for customer that don’t have an order.

SELECT customer.customer\_id, customer\_name, customer\_phone\_number, customer\_type,

order\_id

FROM customer LEFT OUTER JOIN orders

ON customer.customer\_id = orders.customer\_id



### Show the information required to make an invoice for order number 10.

SELECT customer.customer\_id, customer\_name, customer\_street, customer\_city,

customer\_zip\_code, orders.order\_id, order\_placement\_date, vehicle\_make, vehicle\_model,

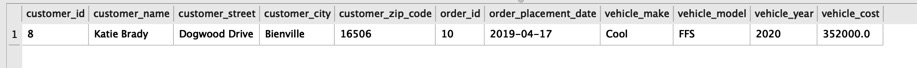
vehicle\_year, vehicle\_cost

FROM customer, orders, vehicle

WHERE orders.customer\_id = customer.customer\_id

AND orders.order\_id = vehicle.vehicle\_id

AND orders.order\_id = 10



Show all customers who have placed an order.

SELECT customer\_name

FROM customer

WHERE customer\_id IN

(SELECT DISTINCT customer\_id from orders)



### What is the name and address for the customer who placed order number 6?

SELECT customer\_name, customer\_street, customer\_zip\_code

FROM customer

WHERE customer.customer\_id =

(SELECT customer\_id FROM orders WHERE order\_id = 6)



### Show orders for vehicle id 2, display Order Id and Order Total.

SELECT order\_line.order\_id, sum (vehicle\_cost\*order\_quantity) AS order\_otal

FROM vehicle INNER JOIN order\_line

ON order\_line.vehicle\_id = vehicle.vehicle\_id

WHERE order\_line.order\_id

IN (SELECT order\_line.order\_id FROM order\_line

WHERE order\_line.vehicle\_id = 2)

GROUP BY order\_line.order\_id

