PHASE 2: SUBMISSION CHECKLIST/SIGNOFF SHEET

GROUP NAME: iWater

GROUP#: 1

Deliverables:

- [Revised] Description of the organization
- [Revised] ER diagram with min/max specifications
- [Revised] Constraints not in ER diagram
- [Revised] Relational Schema with Referential Integrity
- Queries with brief description
- DML + DDL + SQL statements
- Completed PostgreSQL Implementation (Attached Team SQL script)

Group Assessment

- Group Status Report

We have each reviewed the contents of this deliverable.

Phase Leader	Ammarah Mansoor	_____
Phase Recorder	Omer Sheriff	_____
Phase Checker	Sharukh Syed	_____
Technical Advisor	Amshah Mushtaq	_____

1. Introduction

iWater is an app originating in India that aims to standardize the unorganized drinking water vendor network. Its goal is to provide packaged drinking water to customers, ranging from large company offices to local restaurants and homeowners. Throughout the country, there are many local water vendors who own water purification plants, but they don't have a proper platform to connect and deliver to customers. iWater partners with these local vendors to deliver different quantities and brands of water through customer orders. The expert users are iWater programmers and the parametric end users are vendors and customers who have access to delivering and ordering water products through the iWater application. The contact person are the employees who will be contacted by customers when they are interested in purchasing orders.

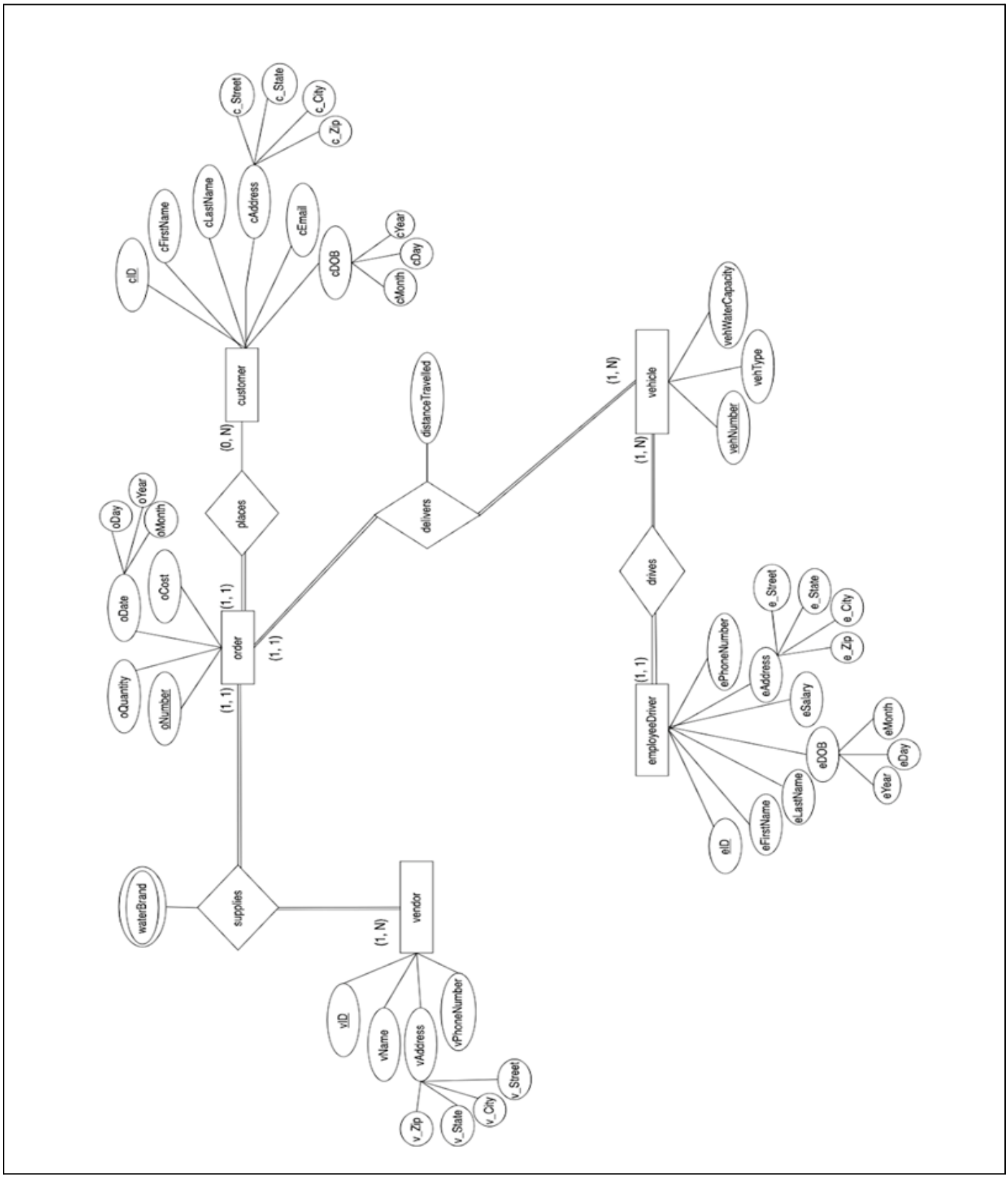
2. Requirements Description

The iWater Distribution Database (IWATERDIST) stores relevant information regarding a company's desire to track the supplying of water from vendors to customers. The IWATERDIST Database contains data about the company vendors, company employees, orders placed by customers, company vehicles supplying the orders, and the type of water distributed by the vendor.

- **Vendors** that supply water to customers are maintained in our database. Each vendor possesses a unique vendor id, vendor name, vendor address, and vendor phone number. Each vendor supplies water orders. One vendor can supply many orders however an order can be supplied by at most one vendor.
- The database keeps a record of the type of water that is **supplied**.
- The company tracks **employeeDrivers** who deliver orders. The information stored include a unique employee id, last name, first name, address, and salary. For employees that work in vehicle delivery, only one employee can drive at most one vehicle. However, a vehicle can be driven by many employees.

- Company **vehicles** driven by employees are used to deliver water orders. Each vehicle possesses a unique vehicle number, vehicle type, water capacity. One vehicle delivers to many customers. However, a customer order is delivered by at most one vehicle.
- The database keeps a record for distance traveled for each **delivery**.
- **Orders** placed by customers are also tracked in our database. Each order has a unique order number, cost, quantity of water ordered, and date the order was placed including the day, month and year.
- Lastly, the database keeps a record of the **customers**. The information stored includes a unique customer identification number, last name, first name, address, and email address. One customer can place many orders but one order belongs to at most one customer

3. ER Diagram


c_Street
c_State
c_City
c_Zip
cAddress
cLastName
cFirstName
cID
cEmail
cDOB
cYear
cDay
cMonth
customer
(0, N)
places
(1, 1)
order
(1, 1)
oDay
oYear
oMonth
oDate
oCost
oQuantity
oNumber
distanceTravelled
delivers
(1, N)
vehicle
vehWaterCapacity
vehType
vehNumber
(1, N)
drives
(1, 1)
employeeDriver
ePhoneNumber
e_Street
e_State
e_City
e_Zip
eAddress
eSalary
eDOB
eMonth
eDay
eYear
eLastName
eFirstName
eID
(1, 1)
supplies
waterBrand
(1, N)
verdor
vID
vName
vAddress
vPhoneNumber
v_Street
v_City
v_State
v_Zip

4. ER Diagram Uncaptured Constraints

The following is a list of constraints that are not captured by the ER diagram of IWATERDIST:

1. Below the constraint values must be a positive integer:
    - Order cost
    - Order quantity
    - Employee salary
    - Distance traveled
2. Below the constraint values must follow a specific format:
    - Customer email must follow a valid email format
    - Customer DOB must follow a valid date of birth format
    - Employee DOB must follow a valid date of birth format
    - Order date must follow a valid date format
3. A vehicle must only be driven by one employee at a time

5. Relational Schema

This section provides the relational schema with referential integrity and the relational table details.

5.1 Relational Schema with Referential Integrity

customer(<u>cID</u>, cFirstName, cLastName, c_street, c_state, c_city, c_zip, cEmail, cDOB)

order(<u>oNumber</u>, oCost, oQuantity, oDate, cID)

Foreign key(cID) references customer (cID)

supplies(<u>oNumber</u>, <u>vID</u>, waterBrand)

Foreign key(oNumber) references order(oNumber)

Foreign key(vID) references vendor(vID)

vendor(<u>vID</u>, vName, vPhoneNumber, v_Street, v_State, v_City, v_Zip)

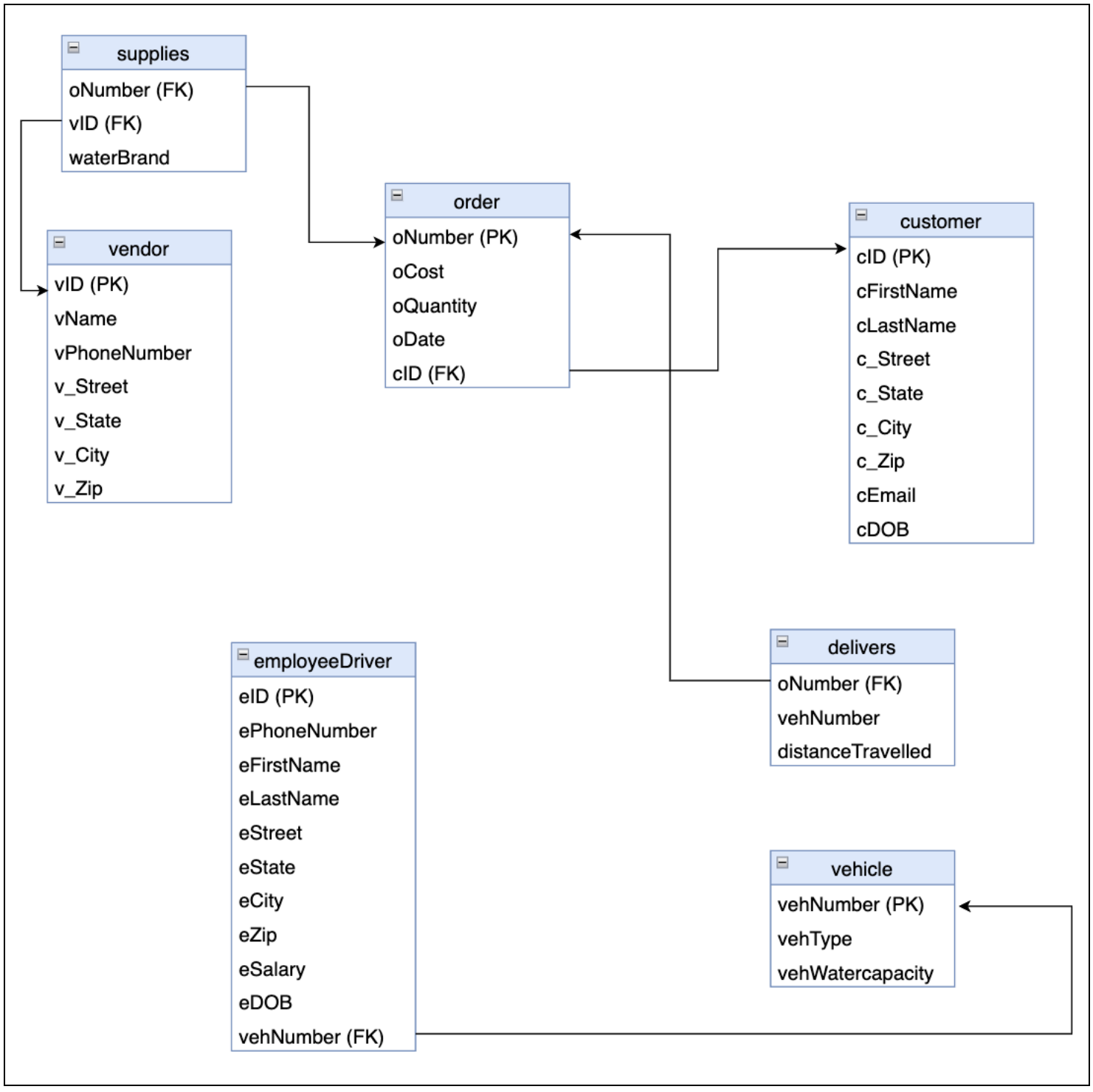
employeeDriver(<u>eID</u>, ePhoneNumber, eFirstName, eLastName, e_Street, e_State, e_City, e_Zip, eSalary, eDOB, vehNumber)

Foreign key(vehNumber) references vehicle(vehNumber)

vehicle(<u>vehNumber</u>, vehType, vehWaterCapacity)

delivers(<u>oNumber</u>, vehNumber, distanceTravelled)

Foreign key(oNumber) reference order(oNumber)


supplies
oNumber (FK)
vID (FK)
waterBrand
vendor
vID (PK)
vName
vPhoneNumber
v_Street
v_State
v_City
v_Zip
order
oNumber (PK)
oCost
oQuantity
oDate
cID (FK)
customer
cID (PK)
cFirstName
cLastName
c_Street
c_State
c_City
c_Zip
cEmail
cDOB
delivers
oNumber (FK)
vehNumber
distanceTravelled
employeeDriver
eID (PK)
ePhoneNumber
eFirstName
eLastName
eStreet
eState
eCity
eZip
eSalary
eDOB
vehNumber (FK)
vehicle
vehNumber (PK)
vehType
vehWatercapacity

5.2 Relational Table Details

The relational schema given in Section 5.1 was mapped into the following tables in the IWATERDIST database. Primary keys have been underlined. Tables that have multiple attributes underlined represent composite keys.

Table Name	**Attribute**	**Description**
vendor	vID	unique vendor ID
	vName	name of vendor
	vPhoneNumber	phone number of vendor
	v_Street	street name of vendor's address
	v_State	state of vendor's address
	v_City	city of vendor's address
	v_Zip	zip code of vendor's address
order	oNumber	unique order number
	oCost	cost of order
	oQuantity	quantity of products in order
	oDate	date of order placed
employeeDriver	eID	unique employee ID
	ePhoneNumber	phone number of employee
	eFirstName	first name of employee
	eLastName	last name of employee
	e_Street	street name of employee's address

	e_State	state of employee's address
	e_City	city of employee's address
	e_Zip	zip code of employee's address
	eSalary	salary of employee
	eDOB	date of birth of employee
vehicle	vehNumber	unique vehicle number
	vehType	type of vehicle
	vehWaterCapacity	water capacity of vehicle
customer	cID	Unique customer id
	cFirstName	First name of customer
	cLastName	Last name of customer
	c_Street	street name of customer address
	c_State	state of customer address
	c_City	city of customer address
	c_Zip	zip code of customer address
	cEmail	Email address of customer
	cDOB	Date of birth of customer
supplies	vID	ID of vendor supplying water
	oNumber	Order number of water ordered by vendor
	waterBrand	Brand of water supplied by vendor

delivers	oNumber	Order number of delivery
	vehNumber	Vehicle number delivering the order
	distanceTravelled	Distance traveled to deliver order

6. Queries

The following table summarizes the queries in the IWATERDIST database.

Query Name	Description	Output	Relations Accessed
vehicleDistanceAndCapacity	List the vehicle numbers, vehicle type and their water capacities. Find the maximum distance traveled by each vehicle. Only include those vehicles with an order quantity greater than 3. Sort the output by vehNumber in ascending order.	vehNumber vehType vehWaterCapacity oQuantity distanceTravelled	Vehicle Delivers Orders
vendorOrderQuantities	Find the maximum order quantity placed for each vendor. List the name of each vendor and the vendor id. Sort the output by vendor id in ascending order.	vID vName oQuantity	Vendor Supplies Orders
aboveAverageSalariedDrivers	For each driver employee whose salary is greater than the average salary, print the employee ID, first name, last name, salary, type of vehicle they drive, and the total distance traveled. Sort the output by salary in ascending order.	eID eFirstName eLastName eSalary vehType distanceTravelled	employeeDriver vehicle delivers
ordersDeliveredByCAEmployees	Find the total numbers of orders delivered by a driver employee to the customers in the state of California. Print driver employee's ID, first name, last name, and the state with the total number of orders delivered.	eID eFirstName eLastName totalOrders	employeeDriver vehicle delivers orders

7. DDL + DML + SQL (Including output screenshots)

The following is an SQL definition of the tables for the IWATERDIST database.

```
/*IWATERDIST Relational Database Schema
customer(cID, cFirstName, cLastName, c_street,  c_state, c_city, c_zip, cEmail, cDOB)
order(oNumber, oCost, oQuantity, oDate, cID)
supplies(oNumber, vID, waterBrand)
vendor(vID, vName, vPhoneNumber, v_Street, v_State, v_City, v_Zip)
employeeDriver(eID, ePhoneNumber, eFirstName, eLastName, e_Street, e_State, e_City, e_Zip, eSalary,
eDOB, vehNumber)
vehicle(vehNumber, vehType, vehWaterCapacity)
delivers(oNumber, vehNumber, distanceTravelled)
*/

DROP TABLE IF EXISTS supplies CASCADE;
DROP TABLE IF EXISTS orders CASCADE;
DROP TABLE IF EXISTS customer CASCADE;
DROP TABLE IF EXISTS employeeDriver CASCADE;
DROP TABLE IF EXISTS delivers CASCADE;
DROP TABLE IF EXISTS vehicle CASCADE;
DROP TABLE IF EXISTS vendor CASCADE;

--1. customer
CREATE TABLE customer(
        cID varchar(10) PRIMARY KEY,
        cFirstName varchar(50),
        cLastName varchar(50),
        c_Street varchar(50),
        c_City varchar(50),
        c_State varchar(50),
        c_Zip int,
        cEmail varchar(50) check (cEmail like '%_@__%.%'),
        cDOB date check (TO_CHAR(cDOB, 'YYYY-MM-DD') like '____-__-__')
);

--1. INSERT customer query
INSERT INTO customer (cID, cFirstName, cLastName, c_Street, c_City, c_State, c_Zip, cEmail, cDOB)
VALUES
('C001', 'John', 'Doe', '123 Main St', 'Anytown', 'CA', 12345, 'john.doe@email.com', '1990-01-01'),
('C002', 'Jane', 'Doe', '456 Elm St', 'Othertown', 'NY', 67890, 'jane.doe@email.com', '1995-02-02'),
('C003', 'Bob', 'Smith', '789 Oak St', 'Somewhere', 'FL', 23456, 'bob.smith@email.com', '1985-03-03'),
('C004', 'Mary', 'Johnson', '321 Maple St', 'Nowhere', 'TX', 34567, 'mary.johnson@email.com',
'1980-04-04'),
```

('C005', 'David', 'Lee', '654 Pine St', 'Anywhere', 'WA', 45678, 'david.lee@email.com', '1992-05-05'),
('C006', 'Karen', 'Brown', '987 Cedar St', 'Everywhere', 'GA', 56789, 'karen.brown@email.com',
'1987-06-06'),
('C007', 'Steven', 'Nguyen', '741 Birch St', 'Nowheresville', 'AZ', 67890, 'steven.nguyen@email.com',
'1998-07-07'),
('C008', 'Sarah', 'Davis', '852 Spruce St', 'Anywhereville', 'IL', 78901, 'sarah.davis@email.com',
'1993-08-08'),
('C009', 'Michael', 'Garcia', '963 Oakwood St', 'Everytown', 'CA', 89012, 'michael.garcia@email.com',
'1988-09-09'),
('C010', 'Elizabeth', 'Wilson', '147 Maplewood St', 'Anyville', 'NY', 90123, 'elizabeth.wilson@email.com',
'1983-10-10'),
('C011', 'Christopher', 'Hernandez', '258 Cedarwood St', 'Nowheretown', 'FL', 12345,
'christopher.hernandez@email.com', '1991-11-11'),
('C012', 'Amanda', 'Lopez', '369 Pinewood St', 'Otherville', 'TX', 23456, 'amanda.lopez@email.com',
'1986-12-12'),
('C013', 'Matthew', 'Martinez', '987 Elmwood St', 'Everyplace', 'WA', 34567,
'matthew.martinez@email.com', '1994-01-13'),
('C014', 'Ashley', 'Gonzalez', '654 Birchwood St', 'Anyplace', 'GA', 45678, 'ashley.gonzalez@email.com',
'1989-02-14'),
('C015', 'Daniel', 'Perez', '321 Cedar St', 'Somewheresville', 'AZ', 56789, 'daniel.perez@email.com',
'1984-03-15'),
('C016', 'Megan', 'Taylor', '741 Oakwood St', 'Nowhereville', 'IL', 67890, 'megan.taylor@email.com',
'1920-04-20'),
('C017', 'Brandon', 'Anderson', '852 Sprucewood St', 'Anyplaceville', 'CA', 78901,
'brandon.anderson@email.com', '1999-04-16'),
('C018', 'Rachel', 'Thomas', '147 Pinewood St', 'Nowheretownsville', 'NY', 89012,
'rachel.thomas@email.com', '1996-05-17'),
('C019', 'Tyler', 'Jackson', '258 Cedarwood St', 'Anywhereville', 'FL', 90123, 'tyler.jackson@email.com',
'1991-06-18'),
('C020', 'Emily', 'White', '369 Pine St', 'Everytown', 'TX', 12345, 'emily.white@email.com', '1986-07-19'),
('C021', 'Jacob', 'Harris', '987 Oakwood St', 'Anywhereville', 'WA', 23456, 'jacob.harris@email.com',
'1994-08-20'),
('C022', 'Olivia', 'Clark', '654 Cedarwood St', 'Nowheresville', 'GA', 34567, 'olivia.clark@email.com',
'1989-09-21'),
('C023', 'William', 'Lewis', '321 Oakwood St', 'Anyville', 'AZ', 45678, 'william.lewis@email.com',
'1984-10-22'),
('C024', 'Sophia', 'Robinson', '741 Maplewood St', 'Everywhere', 'IL', 56789,
'sophia.robinson@email.com', '1998-11-23'),
('C025', 'Ethan', 'Walker', '852 Cedarwood St', 'Nowheretown', 'CA', 67890, 'ethan.walker@email.com',
'1993-12-24');

--2. orders
CREATE TABLE orders(
	oNumber int PRIMARY KEY,

```
        oCost int check (oCost > 0),
        oQuantity int check (oQuantity > 0),
        oDate date check (TO_CHAR(oDate, 'YYYY-MM-DD') like '____-__-__'),
        cID varchar(10),
        FOREIGN KEY (cID) REFERENCES customer(cID)
 );

--2. INSERT orders query
INSERT INTO orders (oNumber, oCost, oQuantity, oDate, cID) VALUES
(1001, 500, 2, '2022-01-01', 'C001'),
(1002, 250, 1, '2022-01-01', 'C002'),
(1003, 1000, 5, '2022-01-02', 'C003'),
(1004, 750, 3, '2022-01-03', 'C004'),
(1005, 1200, 6, '2022-01-03', 'C005'),
(1006, 800, 4, '2022-01-04', 'C006'),
(1007, 300, 2, '2022-01-05', 'C007'),
(1008, 900, 3, '2022-01-05', 'C008'),
(1009, 150, 1, '2022-01-06', 'C009'),
(1010, 2000, 10, '2022-01-06', 'C010'),
(1011, 400, 2, '2022-01-07', 'C011'),
(1012, 100, 1, '2022-01-07', 'C012'),
(1013, 600, 3, '2022-01-08', 'C013'),
(1014, 700, 2, '2022-01-09', 'C014'),
(1015, 900, 3, '2022-01-09', 'C015'),
(1016, 500, 2, '2022-01-10', 'C016'),
(1017, 250, 1, '2022-01-11', 'C017'),
(1018, 750, 3, '2022-01-11', 'C018'),
(1019, 1000, 5, '2022-01-12', 'C019'),
(1020, 1200, 6, '2022-01-12', 'C020'),
(1021, 800, 4, '2022-01-13', 'C021'),
(1022, 300, 2, '2022-01-14', 'C022'),
(1023, 900, 3, '2022-01-14', 'C023'),
(1024, 150, 1, '2022-01-15', 'C024'),
(1025, 2000, 10, '2022-01-15', 'C025'),
(1026, 150, 1, '2022-01-13', 'C001'),
(1027, 400, 4, '2022-01-07', 'C023'),
(1028, 450, 2, '2022-01-07', 'C024'),
(1029, 200, 3, '2022-01-15', 'C008'),
(1030, 800, 5, '2022-01-11', 'C014'),
(1031, 350, 3, '2022-01-08', 'C002'),
(1032, 350, 2, '2022-01-09', 'C007'),
(1033, 250, 5, '2022-01-03', 'C007'),
(1034, 200, 1, '2022-01-05', 'C001'),
(1035, 600, 3, '2022-01-12', 'C012'),
```

```
(1036, 400, 2, '2022-01-10', 'C013'),
(1037, 400, 2, '2022-01-14', 'C015'),
(1038, 600, 4, '2022-01-14', 'C016'),
(1039, 150, 1, '2022-01-10', 'C017'),
(1040, 250, 3, '2022-01-09', 'C023');

--3. supplies
CREATE TABLE supplies(
        vID varchar(20),
        waterBrand varchar(50),
        oNumber int,
        FOREIGN KEY (oNumber) REFERENCES orders(oNumber)
 );

--3. INSERT supplies query
INSERT INTO supplies (vID, waterBrand, oNumber) VALUES
('V001', 'Aquafina', 1001),
('V002', 'Nestle Pure Life', 1002),
('V003', 'Dasani', 1003),
('V004', 'Evian', 1004),
('V005', 'Fiji', 1005),
('V001', 'Smartwater', 1006),
('V002', 'Poland Spring', 1007),
('V003', 'Deer Park', 1008),
('V004', 'Voss', 1009),
('V005', 'Perrier', 1010),
('V001', 'Crystal Geyser', 1011),
('V002', 'San Pellegrino', 1012),
('V003', 'Arrowhead', 1013),
('V004', 'Ozarka', 1014),
('V005', 'Zephyrhills', 1015),
('V001', 'Ice Mountain', 1016),
('V001', 'Kirkland Signature', 1017),
('V002', 'Mountain Valley', 1018),
('V002', 'VOSS', 1019),
('V003', 'Gerolsteiner', 1020),
('V004', 'Topo Chico', 1021),
('V004', 'Glacier', 1022),
('V005', 'Saratoga', 1023),
('V005', 'Canadian Springs', 1024),
('V001', 'Acqua Panna', 1025),
('V004', 'Aquafina', 1026),
('V003', 'Saratoga', 1027),
('V002', 'Canadian Springs', 1028),
```

```
('V002', 'Deer Park', 1029),
('V003', 'Ice Mountain', 1030),
('V004', 'Arrowhead', 1031),
('V005', 'Aquafina', 1032),
('V005', 'San Pellegrino', 1033),
('V005', 'Arrowhead', 1034),
('V001', 'San Pellegrino', 1035),
('V005', 'Zephyrhills', 1036),
('V002', 'Ice Mountain', 1037),
('V004', 'Kirkland Signature', 1038),
('V001', 'Canadian Springs', 1039),
('V005', 'Dasani', 1040);

--4. vendor
CREATE TABLE vendor(
        vID varchar(20)  PRIMARY KEY,
        vName varchar (50),
        vPhoneNumber varchar (12),
        v_Street varchar (50),
        v_City varchar (50),
        v_State varchar (50),
        v_Zip int
 );

--4. INSERT vendor query
INSERT INTO vendor (vID, vName, vPhoneNumber, v_Street, v_City, v_State, v_Zip) VALUES
('V001', 'General services', '555-1234', '123 Main St', 'Los Angeles', 'CA', 90001),
('V002', 'Coca services', '555-2345', '456 Oak Ave', 'Atlanta', 'GA', 30301),
('V003', 'Hormel supplies', '555-3456', '789 Elm St', 'Purchase', 'NY', 10577),
('V004', 'Ferrero transportation', '555-4567', '321 Maple Dr', 'Plano', 'TX', 75023),
('V005', 'Nabisco, USA', '555-5678', '654 Pine St', 'Burlington', 'MA', 01803);

--5. delivers
CREATE TABLE delivers(
        vehNumber varchar(20),
        distanceTravelled int check (distanceTravelled > 0),
        oNumber int,
        FOREIGN KEY (oNumber) REFERENCES orders(oNumber)
 );

--5.INSERT delivers query
INSERT INTO delivers (vehNumber, distanceTravelled, oNumber) VALUES
('VH001', 50, 1001),
('VH002', 30, 1002),
```

('VH003', 40, 1003),
('VH004', 60, 1004),
('VH005', 20, 1005),
('VH006', 70, 1006),
('VH007', 55, 1007),
('VH008', 35, 1008),
('VH009', 45, 1009),
('VH010', 25, 1010),
('VH011', 50, 1011),
('VH012', 30, 1012),
('VH013', 40, 1013),
('VH014', 60, 1014),
('VH015', 20, 1015),
('VH016', 70, 1016),
('VH017', 55, 1017),
('VH018', 35, 1018),
('VH019', 45, 1019),
('VH020', 25, 1020),
('VH021', 50, 1021),
('VH022', 30, 1022),
('VH023', 40, 1023),
('VH024', 60, 1024),
('VH025', 20, 1025),
('VH001', 50, 1026),
('VH023', 40, 1027),
('VH012', 30, 1028),
('VH013', 40, 1029),
('VH002', 30, 1030),
('VH010', 25, 1031),
('VH007', 55, 1032),
('VH002', 30, 1033),
('VH013', 40, 1034),
('VH001', 50, 1035),
('VH011', 50, 1036),
('VH007', 55, 1037),
('VH019', 45, 1038),
('VH013', 40, 1039),
('VH002', 30, 1040);

--6. vehicle
CREATE TABLE vehicle(
	vehNumber varchar(20) PRIMARY KEY,
	vehType varchar(50),
	vehWaterCapacity int

```
);

--6. INSERT vehicle query
INSERT INTO vehicle (vehNumber, vehType, vehWatercapacity) VALUES
('VH001', 'Tanker Truck', 10000),
('VH002', 'Tanker Truck', 10000),
('VH003', 'Flatbed Truck', 5000),
('VH004', 'Flatbed Truck', 5000),
('VH005', 'Dump Truck', 8000),
('VH006', 'Dump Truck', 8000),
('VH007', 'Semi-trailer Truck', 15000),
('VH008', 'Semi-trailer Truck', 15000),
('VH009', 'Box Truck', 5000),
('VH010', 'Box Truck', 5000),
('VH011', 'Tanker Trailer', 20000),
('VH012', 'Tanker Trailer', 20000),
('VH013', 'Flatbed Trailer', 10000),
('VH014', 'Flatbed Trailer', 10000),
('VH015', 'Cement Mixer Truck', 6000),
('VH016', 'Cement Mixer Truck', 6000),
('VH017', 'Garbage Truck', 10000),
('VH018', 'Garbage Truck', 10000),
('VH019', 'Refrigerated Truck', 8000),
('VH020', 'Refrigerated Truck', 8000),
('VH021', 'Tow Truck', 6000),
('VH022', 'Tow Truck', 6000),
('VH023', 'Bucket Truck', 8000),
('VH024', 'Bucket Truck', 8000),
('VH025', 'Fire Truck', 12000);

--7. employeeDriver
CREATE TABLE employeeDriver(
        eID varchar(10) PRIMARY KEY,
        ePhoneNumber varchar(12),
        eFirstName varchar(50),
        eLastName varchar(50),
        e_Street varchar(50),
        e_City varchar(50),
        e_State varchar(50),
        e_Zip varchar(5),
        eSalary float(20) check (eSalary > 0),
        eDOB date check (TO_CHAR(eDOB, 'YYYY-MM-DD') like '____-__-__'),
        vehNumber varchar(20),
        FOREIGN KEY (vehNumber) REFERENCES vehicle(vehNumber)
```

);

--7. INSERT employeeDriver query
INSERT INTO employeeDriver (eID, ePhoneNumber, eFirstName, eLastName, e_Street, e_City, e_State, e_Zip, eSalary, eDOB, vehNumber) VALUES
('EMP001', '5551234', 'John', 'Doe', '123 Main St', 'Anytown', 'CA', '12345', '50000', '1990-01-01', 'VH001'),
('EMP002', '5552345345', 'Jane', 'Doe', '456 High St', 'Othertown', 'CA', '23456', '60000', '1985-05-15', 'VH002'),
('EMP003', '5553456898', 'Bob', 'Smith', '789 Maple Ave', 'Sometown', 'CA', '34567', '55000', '1992-02-28', 'VH003'),
('EMP004', '5554567567', 'Alice', 'Johnson', '321 Elm St', 'Anytown', 'CA', '45678', '65000', '1980-10-02', 'VH004'),
('EMP005', '5555678678', 'David', 'Lee', '654 Oak Dr', 'Othertown', 'CA', '56789', '50000', '1995-12-31', 'VH005'),
('EMP006', '5556789789', 'Cathy', 'Kim', '987 Cedar Ln', 'Sometown', 'CA', '67890', '75000', '1975-07-10', 'VH006'),
('EMP007', '5557890567', 'Mark', 'Davis', '135 Pine St', 'Anytown', 'CA', '78901', '80000', '1972-11-20', 'VH007'),
('EMP008', '5558901454', 'Karen', 'Chen', '246 Walnut Ave', 'Othertown', 'CA', '89012', '55000', '1990-06-04', 'VH008'),
('EMP009', '5559012789', 'Steven', 'Nguyen', '369 Elmwood Dr', 'Sometown', 'CA', '90123', '70000', '1988-03-08','VH009'),
('EMP010', '5550101676', 'Emily', 'Garcia', '777 Oak St', 'Anytown', 'CA', '01234', '60000', '1984-09-16', 'VH010'),
('EMP011', '5551111345', 'Michael', 'Wang', '888 Maple Ave', 'Othertown', 'CA', '12345', '85000', '1978-12-25', 'VH011'),
('EMP012', '5552222123', 'Amanda', 'Rodriguez', '333 Cedar St', 'Sometown', 'CA', '23456', '45000', '1994-08-18','VH012'),
('EMP013', '5553333897', 'Eric', 'Chang', '555 Pine Ave', 'Anytown', 'CA', '34567', '65000', '1986-04-23', 'VH013'),
('EMP014', '5554444777', 'Jasmine', 'Kim', '222 Walnut St', 'Othertown', 'CA', '45678', '75000', '1982-07-30', 'VH014'),
('EMP015', '9876543211', 'Samantha', 'Garcia', '123 Main St', 'Los Angeles', 'CA', '90001', '50000', '1993-08-12', 'VH015'),
('EMP016', '9876543212', 'Jonathan', 'Fernandez', '456 Pine St', 'San Francisco', 'CA', '94102', '65000', '1990-06-21', 'VH016'),
('EMP017', '9876543213', 'Karen', 'Kim', '789 Oak St', 'New York', 'NY', '10001', '55000', '1994-05-02', 'VH017'),
('EMP018', '9876543214', 'Avery', 'Campbell', '1010 5th Ave', 'Seattle', 'WA', '98101', '70000', '1988-12-06', 'VH018'),
('EMP019', '9876543215', 'Connor', 'Liu', '1111 8th St', 'Austin', 'TX', '78701', '62000', '1992-11-19', 'VH019'),
('EMP020', '9876543216', 'Dylan', 'Nguyen', '2222 1st St', 'Houston', 'TX', '77002', '58000', '1991-09-14',

'VH020'),
('EMP021', '9876543217', 'Aria', 'Wang', '3333 4th St', 'Chicago', 'IL', '60601', '75000', '1987-02-28', 'VH021'),
('EMP022', '9876543218', 'Lila', 'Gupta', '4444 Main St', 'Boston', 'MA', '02108', '80000', '1985-07-17', 'VH022'),
('EMP023', '9876543219', 'Nathan', 'Lee', '5555 Pine St', 'Denver', 'CO', '80202', '63000', '1992-04-09', 'VH023'),
('EMP024', '9876543210', 'Ivy', 'Patel', '6666 Oak St', 'Portland', 'OR', '97205', '57000', '1991-01-22', 'VH024'),
('EMP025', '9876543221', 'Jayden', 'Hernandez', '7777 Maple St', 'Phoenix', 'AZ', '85004', '68000', '1989-09-30','VH025');

vehicleDistanceAndCapacity

SELECT V.vehNumber, V.vehType, V.vehWaterCapacity, O.oQuantity,
max(D.distanceTravelled) AS maxdistance
FROM vehicle V, delivers D, orders O
WHERE V.vehNumber = D.vehNumber AND O.oNumber = D.oNumber AND O.oQuantity > 3
GROUP BY V.vehNumber, V.vehWaterCapacity, O.oQuantity
ORDER BY vehNumber;

```
1  SELECT V.vehNumber, V.vehType, V.vehWaterCapacity, O.oQuantity,
2  max(D.distanceTravelled) AS maxdistance
3  FROM vehicle V, delivers D, orders O
4  WHERE V.vehNumber = D.vehNumber AND O.oNumber = D.oNumber AND O.oQuantity > 3
5  GROUP BY V.vehNumber, V.vehWaterCapacity, O.oQuantity
6  ORDER BY vehNumber;
```

Data Output | Messages | Notifications

	vehnumber character varying (20)	vehtype character varying (50)	vehwatercapacity integer	oquantity integer	maxdistance integer
1	VH002	Tanker Truck	10000	5	30
2	VH003	Flatbed Truck	5000	5	40
3	VH005	Dump Truck	8000	6	20
4	VH006	Dump Truck	8000	4	70
5	VH010	Box Truck	5000	10	25
6	VH019	Refrigerated Truck	8000	4	45
7	VH019	Refrigerated Truck	8000	5	45
8	VH020	Refrigerated Truck	8000	6	25
9	VH021	Tow Truck	6000	4	50
10	VH023	Bucket Truck	8000	4	40
11	VH025	Fire Truck	12000	10	20

vendorOrderQuantities

SELECT V.vID, V.vName, max(oQuantity) AS maxOrderQ
FROM vendor V, supplies S, orders O
WHERE V.vID = S.vID AND S.oNumber = O.oNumber
GROUP BY V.vName, V.vID
ORDER BY V.vID;

```
1  SELECT V.vID, V.vName, max(oQuantity) AS maxOrderQ
2  FROM vendor V, supplies S, orders O
3  WHERE V.vID = S.vID AND S.oNumber = O.oNumber
4  GROUP BY V.vName, V.vID
5  ORDER BY V.vID;
6
7
```

Data Output | Messages | Notifications

	vid [PK] character varying (20)	vname character varying (50)	maxorderq integer
1	V001	General services	10
2	V002	Coca services	5
3	V003	Hormel supplies	6
4	V004	Ferrero transportation	4
5	V005	Nabisco, USA	10

aboveAverageSalariedDrivers

SELECT e.eID, e.eFirstName, e.eLastName, e.eSalary, v.vehType, SUM(d.distanceTravelled) as distanceTravelled
FROM employeeDriver e, vehicle v, delivers d
WHERE v.vehNumber = e.vehNumber AND d.vehNumber = v.vehNumber AND e.eSalary >
(SELECT AVG(eSalary)
FROM employeeDriver)
GROUP BY e.eID, e.eFirstName, e.eLastName, e.eSalary, v.vehType
ORDER BY e.eSalary ASC;

```
12  SELECT e.eID, e.eFirstName, e.eLastName, e.eSalary, v.vehType, SUM(d.distanceTravelled) as distanceTravelled
13  FROM employeeDriver e, vehicle v, delivers d
14  WHERE v.vehNumber = e.vehNumber AND d.vehNumber = v.vehNumber AND e.eSalary > (SELECT AVG(eSalary)
15                                                                                 FROM employeeDriver)
16  GROUP BY e.eID, e.eFirstName, e.eLastName, e.eSalary, v.vehType
17  ORDER BY e.eSalary ASC;
18
```

Data Output | Messages | Notifications

	eid character varying (10)	efirstname character varying (50)	elastname character varying (50)	esalary real	vehtype character varying (50)	distancetravelled bigint
1	EMP013	Eric	Chang	65000	Flatbed Trailer	160
2	EMP016	Jonathan	Fernandez	65000	Cement Mixer Truck	70
3	EMP004	Alice	Johnson	65000	Flatbed Truck	60
4	EMP025	Jayden	Hernandez	68000	Fire Truck	20
5	EMP018	Avery	Campbell	70000	Garbage Truck	35
6	EMP009	Steven	Nguyen	70000	Box Truck	45
7	EMP014	Jasmine	Kim	75000	Flatbed Trailer	60
8	EMP021	Aria	Wang	75000	Tow Truck	50
9	EMP006	Cathy	Kim	75000	Dump Truck	70
10	EMP007	Mark	Davis	80000	Semi-trailer Truck	165
11	EMP022	Lila	Gupta	80000	Tow Truck	30
12	EMP011	Michael	Wang	85000	Tanker Trailer	100

OrdersDeliveredByCAEmployees

SELECT E.eID, E.eFirstName, E.eLastName, E.e_state, COUNT(O.oNumber) AS Totalorders
FROM employeeDriver E, vehicle V, delivers D, orders O
WHERE E.vehNumber = V.vehNumber AND V.vehNumber= D.vehNumber AND D.oNumber = O.oNumber AND E.e_state = 'CA'
GROUP BY E.eID
ORDER BY Totalorders DESC;

Query | Query History

```
1  SELECT E.eID, E.eFirstName, E.eLastName, E.e_state, COUNT(O.oNumber) AS Totalorders
2  FROM employeeDriver E, vehicle V, delivers D, orders O
3  WHERE E.vehNumber = V.vehNumber AND V.vehNumber= D.vehNumber AND D.oNumber = O.oNumber AND E.e_state = 'CA'
4  GROUP BY  E.eID
5  ORDER BY Totalorders DESC;
```

Data Output | Messages | Notifications

	eid [PK] character varying (10)	efirstname character varying (50)	elastname character varying (50)	e_state character varying (50)	totalorders bigint
1	EMP013	Eric	Chang	CA	4
2	EMP002	Jane	Doe	CA	4
3	EMP001	John	Doe	CA	3
4	EMP007	Mark	Davis	CA	3
5	EMP012	Amanda	Rodriguez	CA	2
6	EMP010	Emily	Garcia	CA	2
7	EMP011	Michael	Wang	CA	2
8	EMP014	Jasmine	Kim	CA	1
9	EMP015	Samantha	Garcia	CA	1
10	EMP016	Jonathan	Fernandez	CA	1
11	EMP003	Bob	Smith	CA	1
12	EMP004	Alice	Johnson	CA	1
13	EMP005	David	Lee	CA	1
14	EMP006	Cathy	Kim	CA	1
15	EMP008	Karen	Chen	CA	1
16	EMP009	Steven	Nguyen	CA	1

GROUP STATUS REPORT

GROUP #: 1 **GROUP NAME: iWater** **PHASE #:** 2

Dates & attendance at group meetings in this phase:

Date	Meeting Duration	Members Present
Thursday, February 23	2:30 PM - 4:30 PM	All group members present.
Wednesday, March 15	5:00 PM - 7:00 PM	All group members present.
Thursday, March 16	2:30 PM - 5:00 PM	All group members present.
Monday, March 20	7:00PM - 9:00 PM	All group members present.

Overview of progress on project as of March 21:

Phase 2 Deliverable is complete. The IWATERDIST database is complete and functional. Listed below are the reports, queries, and DDL responsibilities completed by each group member.

Contributions of Group Members

Leader: Ammarah Mansoor

- Discussed revisions with Professor Silva
- Revised Requirements Description from Phase 1
- QUERY IMPLEMENTATION: vehicleDistanceAndCapacity
- TABLE DDL: order
- TABLE DDL: delivers

Recorder: Omer Sheriff

- Helped checker revise DDL and DML
- Revised Relational Table from Phase 1
- QUERY IMPLEMENTATION: vendorOrderQuantities
- TABLE DDL: customer

Phase Checker: Shahrukh Syed

- Revised DDL and DML
- Revised Relational Schema with Referential Integrity from Phase 1
- QUERY IMPLEMENTATION: ordersDeliveredByCAEmployees
- TABLE DDL: employeeDriver
- TABLE DDL: vehicle

Technical Advisor: Amshah Mushtaq

- Prepared group status report
- Revised document to include verified DDL, DML, and SQL
- Revised ER Diagram from Phase 1
- QUERY IMPLEMENTATION: aboveAverageSalariedDrivers
- TABLE DDL: supplies
- TABLE DDL: vendor