ERD :-

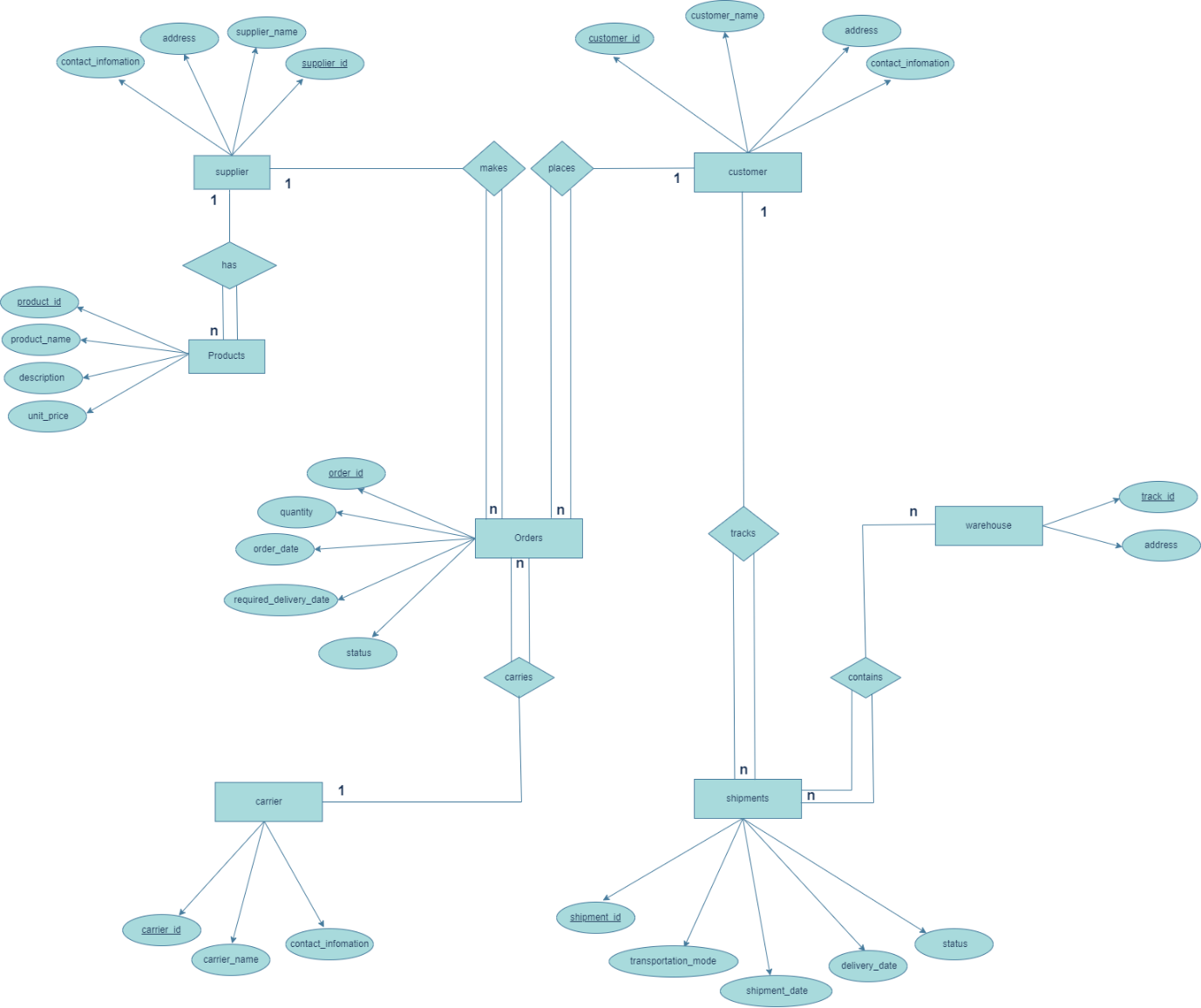


Table :-

CREATE TABLE supply\_chain.suppliers (

id integer PRIMARY KEY,

name VARCHAR(255) NOT NULL,

address VARCHAR(255) NOT NULL,

contact\_information VARCHAR(255) NOT NULL

);

CREATE TABLE supply\_chain.products (

id integer PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description VARCHAR(255) NOT NULL,

unit\_price DECIMAL(10, 2) NOT NULL,

supplier\_id INT NOT NULL,

FOREIGN KEY (supplier\_id) REFERENCES supply\_chain.suppliers(id)

);

CREATE TABLE supply\_chain.customer (

id integer PRIMARY KEY,

name VARCHAR(255) NOT NULL,

address VARCHAR(255) NOT NULL,

contact\_information VARCHAR(255) NOT NULL

);

CREATE TABLE supply\_chain.carriers (

id integer PRIMARY KEY,

name VARCHAR(255) NOT NULL,

address VARCHAR(255) NOT NULL,

contact\_information VARCHAR(255) NOT NULL

);

CREATE TABLE supply\_chain.orders (

id integer PRIMARY KEY,

customer\_id INT NOT NULL,

supplier\_id INT NOT NULL,

carrier\_id INT NOT NULL,

quantity INT NOT NULL,

order\_date DATE NOT NULL,

required\_delivery\_date DATE NOT NULL,

status VARCHAR(255) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES supply\_chain.customer(id),

FOREIGN KEY (supplier\_id) REFERENCES supply\_chain.suppliers(id),

FOREIGN KEY (carrier\_id) REFERENCES supply\_chain.carriers(id)

);

CREATE TABLE supply\_chain.shipments (

id integer PRIMARY KEY,

customer\_id INT NOT NULL,

transportation\_mode VARCHAR(255) NOT NULL,

shipment\_date DATE NOT NULL,

delivery\_date DATE NOT NULL,

status VARCHAR(255) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES supply\_chain.customer(id)

);

CREATE TABLE supply\_chain.warehouse (

id integer PRIMARY KEY,

address VARCHAR(255) NOT NULL

);

CREATE TABLE supply\_chain.bridge (

shipment\_id INT NOT NULL,

track\_id INT NOT NULL,

FOREIGN KEY (shipment\_id) REFERENCES supply\_chain.shipments(id),

FOREIGN KEY (track\_id) REFERENCES supply\_chain.warehouse(id)

);

Data :-

1 Supplier :-

insert into supply\_chain.suppliers(id,name,address,contact\_information)

values

(1,'Divansh','Jaipur',7094721500),

(2,'Gunjan','Ahmedabad',8838521582),

(3,'Aayush','Surat',9934571542),

(4,'Siddharth','Udaipur',6034621574),

(5,'Jinansh','Ahmedabad',9074721599),

(6,'Jeevan','Ahmedabad',8034521872),

(7,'Vivek','Rajkot',9334021577),

(8,'Rituraj','Jaipur',6084521572),

(9,'Ahmad','Surat',8934561572),

(10,'Ishaan','Rajkot',7634921572);

2 Products :-

INSERT INTO supply\_chain.products (id, name, description, unit\_price, supplier\_id)

VALUES

(1, 'headphones', 'Premium headphones ', 1000, 1),

(2, 'Smartwatches', 'Boat smart watches', 5000, 2),

(3, 'TV', 'HD Quality Television in the world', 30000, 3),

(4, 'Keyboard', 'Mechanical gaming keyboard', 150, 4),

(5, 'Mouse', 'Wireless ergonomic mouse', 80, 5),

(6, 'Laptop', 'Ultra-thin powerful laptop', 2000, 6),

(7, 'Chair', 'Office chair', 300, 7),

(8, 'Desk', 'Adjustable desk', 500, 8),

(9, 'Monitor', '27-inch HD Monitor', 600, 9),

(10, 'Printer', 'All-in-one scanner', 400, 10),

(11, 'Speakers', 'Surround sound speakers', 800, 1),

(12, 'Microwave', 'Compact microwave oven', 1500, 1),

(13, 'Refrigerator', 'Double-door fridge', 3000, 2),

(14, 'Vacuum Cleaner', 'Robotic vacuum cleaner', 700, 3),

(15, 'Blender', 'High-speed blender', 120, 4);

3 Customer

insert into supply\_chain.customer (id,name,address,contact\_information)

values

(1,'Aarya','Mumbai',9878567443),

(2,'Kanak','Mount Abu',9878567444),

(3,'Urja','Telangana',9878567445),

(4,'Shruti','Noida',9878567446),

(5,'Ananya','Sikkim',9878567447),

(6,'Dev','Chennai',9878567448),

(7,'Alankrita','Uttrakhand',9878567449),

(8,'Rishabh','Jaipur',9878567453),

(9,'Samarth','Ahmedabad',9878557443),

(10,'Falguni','Hyderabad',9872567443);

4 Carrier

INSERT INTO supply\_chain.carriers (id, name, address, contact\_information) VALUES

(1, 'Rajat', 'Jaipur', '9876543210'),

(2, 'Anjali', 'Ahmedabad', '8765432109'),

(3, 'Vikram', 'Surat', '7654321098'),

(4, 'Meera', 'Udaipur', '6543210987'),

(5, 'Sohan', 'Ahmedabad', '5432109876'),

(6, 'Priya', 'Jaipur', '4321098765'),

(7, 'Kiran', 'Surat', '3210987654'),

(8, 'Arjun', 'Udaipur', '2109876543'),

(9, 'Neha', 'Ahmedabad', '1098765432'),

(10, 'Amit', 'Jaipur', '0123456789');

5 Order :-

INSERT INTO supply\_chain.orders (id, customer\_id, supplier\_id, carrier\_id, quantity, order\_date, required\_delivery\_date, status)

VALUES

(1, 1, 1, 1, 100, '2023-01-15', '2023-01-25', 'Pending'),

(2, 1, 1, 2, 150, '2023-02-05', '2023-02-15', 'In Progress'),

(3, 2, 2, 1, 75, '2023-03-10', '2023-03-20', 'Shipped'),

(4, 2, 2, 3, 200, '2023-04-01', '2023-04-10', 'Delivered'),

(5, 3, 3, 1, 50, '2023-05-12', '2023-05-22', 'Pending'),

(6, 3, 3, 3, 120, '2023-06-02', '2023-06-12', 'In Progress'),

(7, 3, 3, 4, 80, '2023-07-15', '2023-07-25', 'Shipped'),

(8, 4, 4, 5, 180, '2023-08-05', '2023-08-15', 'Delivered'),

(9, 5, 5, 6, 90, '2023-09-10', '2023-09-20', 'Pending'),

(10,5, 6, 7, 250, '2023-10-01', '2023-10-10', 'In Progress');

6 Shipments :-

INSERT INTO supply\_chain.shipments (id, customer\_id, transportation\_mode, shipment\_date, delivery\_date, status)

VALUES

(1, 1, 'Air', '2023-01-20', '2023-01-25', 'In Transit'),

(2, 1, 'Surface', '2023-02-10', '2023-02-18', 'Delivered'),

(3, 1, 'Air', '2023-03-15', '2023-03-22', 'In Transit'),

(4, 2, 'Surface', '2023-04-05', '2023-04-12', 'Delivered'),

(5, 2, 'Air', '2023-05-18', '2023-05-25', 'In Transit'),

(6, 3, 'Surface', '2023-06-08', '2023-06-15', 'Delivered'),

(7, 3, 'Air', '2023-07-22', '2023-07-29', 'In Transit'),

(8, 3, 'Surface', '2023-08-12', '2023-08-19', 'Delivered'),

(9, 4, 'Air', '2023-09-25', '2023-10-02', 'In Transit'),

(10, 4, 'Surface', '2023-10-15', '2023-10-22', 'Delivered');

7 Warehouse

INSERT INTO supply\_chain.warehouse (id, address) VALUES

(1, 'Ahmedabad'),

(2, 'Surat'),

(3, 'Rajkot');

8 bridge

INSERT INTO supply\_chain.bridge (shipment\_id,track\_id)

VALUES

(1,1),

(2,2),

(3,3);

* Simple Queries :-

[Q-1] Query to Retrieve the Number of Orders for a Specific Customer :-

[Ans] SELECT c.id, COUNT(\*) AS order\_count

FROM supply\_chain.orders o, supply\_chain.customer c

WHERE o.customer\_id = c.id

AND c.id = 1

GROUP BY c.id;

[Q-2] Query to Retrieve all products that have the highest unit price :-

[Ans] SELECT \* from supply\_chain.products where unit\_price in (select MAX(unit\_price) from supply\_chain.products);

[Q-3] Query to Sum the Quantities of Orders for a Specific Supplier :-

[Ans] SELECT supplier\_id, SUM(quantity) AS total\_quantity

FROM supply\_chain.orders

WHERE supplier\_id = 6

GROUP BY supplier\_id;

[Q-4] Query to Retrieve Shipped Shipments Sorted by Shipment Date :-

[Ans] SELECT \*

FROM supply\_chain.shipments

WHERE status = 'Delivered'

ORDER BY shipment\_date ASC;

[Q-5] Query to Calculate the Total Quantity of Products in Transit:

[Ans]SELECT SUM(o.quantity) AS Total\_quantity\_in\_transit

FROM supply\_chain.orders o,

supply\_chain.shipments s where o.id = s.id

and s.status = 'In Transit';

* Complex Queries :-

[Q-1] Query to Find Customers with Multiple Orders :-

[Ans]

SELECT c.id AS customer\_id, c.name AS customer\_name, COUNT(o.id) AS order\_count

FROM supply\_chain.customer c

JOIN supply\_chain.orders o ON c.id = o.customer\_id

GROUP BY c.id, c.name

HAVING COUNT(o.id) > 1;

[Q-2] Query to Retrieve Products Ordered by a Specific Customer with Shipment Details :-

[Ans]

SELECT c.name AS customer\_name, s.id AS shipment\_id,s.status,s.delivery\_date, w.address AS warehouse\_address

FROM supply\_chain.customer c

JOIN supply\_chain.shipments s ON c.id = s.customer\_id

JOIN supply\_chain.bridge b ON s.id = b.shipment\_id

JOIN supply\_chain.warehouse w ON b.track\_id = w.id

WHERE c.id = 1;

[Q-3] Query to Retrieve Products with the Lowest Total Order Quantity:

[Ans]

SELECT p.id AS product\_id, p.name AS product\_name, SUM(o.quantity) AS total\_order\_quantity

FROM supply\_chain.products p

LEFT JOIN supply\_chain.orders o ON p.supplier\_id = o.supplier\_id

GROUP BY p.id, p.name

ORDER BY total\_order\_quantity

LIMIT 1;

[Q-4] Query to Find the Customers Who Have Not Placed Orders:

[Ans] SELECT c.id AS customer\_id, c.name AS customer\_name

FROM supply\_chain.customer c

LEFT JOIN supply\_chain.orders o ON c.id = o.customer\_id

WHERE o.id IS NULL;

[Q-5] Query to Retrieve Warehouse Information with the Total Quantity of Products Stored :-

[Ans]

SELECT w.id AS warehouse\_id, w.address AS warehouse\_address, SUM(o.quantity) AS total\_products\_stored

FROM supply\_chain.warehouse w, supply\_chain.bridge b, supply\_chain.shipments s, supply\_chain.orders o

WHERE w.id = b.track\_id

AND b.shipment\_id = s.id

AND s.id = o.id

GROUP BY w.id;

* Relational Algebra
* Simple Queries :-

[Q-1] Relational Algebra for Retrieving the Number of Orders for a Specific Customer :-

[Ans] π customer\_id, count(orders) (σ\_customer\_id=1 (orders ⨝ customer) )

[Q-2] Relational Algebra for Retrieving all products that have the highest unit price :-

[Ans] π (σ\_unit\_price=π\_max(unit\_price) (products))

[Q-3] Query to Sum the Quantities of Orders for a Specific Supplier :-

[Ans] π supplier\_id, sum(quantity) (σ\_supplier\_id=6 (orders))

[Q-4] Query to Retrieve Shipped Shipments Sorted by Shipment Date :-

[Ans] π shipment\_id, transportation\_mode, shipment\_date, delivery\_date, status

(σ status=’Delivered’ (σ shipment\_date ASC (shipments)))

[Q-5] Query to Calculate the Total Quantity of Products in Transit:

[Ans] π\_sum(quantity) (σ\_status='In Transit' (orders ⨝ shipments ))

* Complex Queries :-

[Q-1] Query to Find Customers with Multiple Orders :-

[Ans]

[Q-2] Query to Retrieve Products Ordered by a Specific Customer with Shipment Details :-

[Ans]

[Q-3] Query to Retrieve Products with the Lowest Total Order Quantity:

[Ans]

[Q-4] Query to Find the Customers Who Have Not Placed Orders:

[Ans]

[Q-5] Query to Retrieve Warehouse Information with the Total Quantity of Products Stored :-

[Ans]