**AIM:** SQL to create tables, constraints and manipulate the data for the Logical Schema you designed for your ER Data Model.

**SQL-DDL:**

*CREATE* db texttile\_firm;

*CREATE* *TABLE* Address (

    address\_id INT AUTO\_INCREMENT PRIMARY KEY,

    entity\_type ENUM('WORKER', 'CUSTOMER'),

    street VARCHAR(255),

    city VARCHAR(255),

    postal\_code VARCHAR(20)

);

*CREATE* *TABLE* Worker (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    age INT,

*name* VARCHAR(255),

    aadharNo *FLOAT*,

    gender VARCHAR(1),

    address\_id INT,

    FOREIGN KEY (address\_id) REFERENCES *Address*(address\_id)

);

*CREATE* *TABLE* Machine (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    identifier ENUM('DESIGN', 'PLAIN'),

    capacity *FLOAT*,

    worker\_id INT,

    FOREIGN KEY (worker\_id) REFERENCES Worker(id) ON DELETE *SET* *NULL*

);

*CREATE* *TABLE* Maintenance (

    maintenance\_id INT AUTO\_INCREMENT PRIMARY KEY,

    maintenance\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    cost DECIMAL(10,2),

    machine\_id INT,

    FOREIGN KEY (machine\_id) REFERENCES Machine(id)

);

*CREATE* *TABLE* Produced (

    produced\_id INT AUTO\_INCREMENT PRIMARY KEY,

    production\_date DATE,

    quantity INT,

    identifier VARCHAR(255),

    rate DECIMAL(10,2),

    worker\_id INT,

    FOREIGN KEY (worker\_id) REFERENCES Worker(id)

);

*CREATE* *TABLE* Inventory (

    identifier VARCHAR(255) PRIMARY KEY,

    quantity INT,

    rate DECIMAL(10,2)

);

*CREATE* *TABLE* Customer (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    company\_name VARCHAR(255),

    gstin VARCHAR(20),

    address\_id INT,

    shipping\_address\_id INT,

    FOREIGN KEY (address\_id) REFERENCES *Address*(address\_id)

);

*CREATE* *TABLE* Orders (

    bill\_no VARCHAR(10) PRIMARY KEY,

    identifier VARCHAR(255),

    order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    isPaymentDone BOOLEAN,

    customer\_id INT,

    quantity INT,

    amount DECIMAL(10,2),

    inventory\_type VARCHAR(255),

    FOREIGN KEY (identifier) REFERENCES Inventory(identifier),

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

);

*CREATE* *TABLE* Material (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    identifier ENUM('COLORED', 'NON\_COLORED'),

    purchase\_date DATE,

    rate DECIMAL(10,2),

    quantity INT

);

**DDL-DML:**

*-- Update Worker's Age:*

*UPDATE* Worker

*SET* age *=* 40

*WHERE* id *=* 1;

*-- Update Maintenance Cost:*

*UPDATE* Maintenance

*SET* cost *=* cost *\** 1.1

*WHERE* maintenance\_date *<* '2024-01-01';

*-- Insert New Machine:*

*INSERT INTO* Machine (identifier, capacity, worker\_id)

*VALUES* ('EMBROIDERY', 200, 3);

*-- Delete Worker:*

*DELETE* *FROM* Worker

*WHERE* id *=* 5;

*-- Update Customer's Shipping Address:*

*UPDATE* Customer

*SET* shipping\_address\_id *=* 2

*WHERE* id *=* 1;

*-- Insert New Inventory Item:*

*INSERT INTO* Inventory (identifier, quantity, rate)

*VALUES* ('SILK', 100, 25.50);

*-- Update Order Quantity:*

*UPDATE* Orders

*SET* quantity *=* quantity *+* 10

*WHERE* bill\_no *=* 'ORD123';

*-- Update Production Rate:*

*UPDATE* Produced

*SET* rate *=* rate *\** 1.05

*WHERE* production\_date *BETWEEN* '2023-01-01' *AND* '2023-12-31';

*-- Insert New Customer:*

*INSERT INTO* Customer (company\_name, gstin, address\_id)

*VALUES* ('ABC Textiles', 'GST123456789', 4);

*-- Delete Maintenance Record:*

*DELETE* *FROM* Maintenance

*WHERE* maintenance\_id *=* 10;

*-- Update Material Quantity:*

*UPDATE* Material

*SET* quantity *=* quantity *-* 50

*WHERE* identifier *=* 'COLORED';

*-- Insert New Maintenance Record:*

*INSERT INTO* Maintenance (maintenance\_date, cost, machine\_id)

*VALUES* ('2024-03-15', 500.00, 3);

*-- Update Order Payment Status:*

*UPDATE* Orders

*SET* isPaymentDone *=* TRUE

*WHERE* bill\_no *=* 'ORD456';

*-- Insert New Worker:*

*INSERT INTO* Worker (age, *name*, aadharNo, gender, address\_id)

*VALUES* (35, 'John Doe', 1234567890, 'M', 6);

*-- Update Customer's Company Name:*

*UPDATE* Customer

*SET* company\_name *=* 'XYZ Fabrics'

*WHERE* id *=* 2;

*-- Delete Machine:*

*DELETE* *FROM* Machine

*WHERE* id *=* 5;

*Update* Produced Quantity:

*-- UPDATE Produced*

*SET* quantity *=* quantity *-* 20

*WHERE* production\_date *<* '2023-01-01';

*-- Insert New Material:*

*INSERT INTO* Material (identifier, purchase\_date, rate, quantity)

*VALUES* ('NON\_COLORED', '2024-04-01', 30.00, 500);

*-- Update Customer's GSTIN:*

*UPDATE* Customer

*SET* gstin *=* 'GST987654321'

*WHERE* id *=* 3;

*-- Delete Inventory Item:*

*DELETE* *FROM* Inventory

*WHERE* identifier *=* 'COTTON';

*-- Average Worker Age:*

*SELECT* AVG(age) *AS* avg\_age

*FROM* Worker;

*-- Total Maintenance Cost:*

*SELECT* SUM(cost) *AS* total\_cost

*FROM* Maintenance;

*-- Count of Machines by Worker Gender:*

*SELECT* w.gender, COUNT(m.id) *AS* machine\_count

*FROM* Worker w

*LEFT JOIN* Machine m *ON* w.id *=* m.worker\_id

*GROUP BY* w.gender;

*-- Maximum Production Rate by Machine Type:*

*SELECT* m.identifier, MAX(p.rate) *AS* max\_rate

*FROM* Machine m

*LEFT JOIN* Produced p *ON* m.id *=* p.machine\_id

*GROUP BY* m.identifier;

*-- Average Order Amount by Customer:*

*SELECT* c.company\_name, AVG(o.amount) *AS* avg\_order\_amount

*FROM* Customer c

*JOIN* Orders o *ON* c.id *=* o.customer\_id

*GROUP BY* c.company\_name;

*-- List of Machines with Total Capacity:*

*SELECT* m.identifier, SUM(m.capacity) *AS* total\_capacity

*FROM* Machine m

*GROUP BY* m.identifier;

*-- Number of Orders Placed by Customer:*

*SELECT* c.company\_name, COUNT(o.bill\_no) *AS* order\_count

*FROM* Customer c

*LEFT JOIN* Orders o *ON* c.id *=* o.customer\_id

*GROUP BY* c.company\_name;

*-- Average Maintenance Cost by Machine Type:*

*SELECT* m.identifier, AVG(mt.cost) *AS* avg\_maintenance\_cost

*FROM* Machine m

*LEFT JOIN* Maintenance mt *ON* m.id *=* mt.machine\_id

*GROUP BY* m.identifier;

*-- List of Customers with Total Order Amount:*

*SELECT* c.company\_name, SUM(o.amount) *AS* total\_order\_amount

*FROM* Customer c

*JOIN* Orders o *ON* c.id *=* o.customer\_id

*GROUP BY* c.company\_name;

*-- Average Production Quantity by Worker Age Group:*

*SELECT* *CASE*

*WHEN* w.age *BETWEEN* 20 *AND* 30 *THEN* '20-30'

*WHEN* w.age *BETWEEN* 31 *AND* 40 *THEN* '31-40'

*ELSE* 'Above 40'

*END* *AS* age\_group,

       AVG(p.quantity) *AS* avg\_production\_quantity

*FROM* Worker w

*LEFT JOIN* Produced p *ON* w.id *=* p.worker\_id

*GROUP BY* age\_group;

*-- Maximum Maintenance Cost by Worker Gender:*

*SELECT* w.gender, MAX(mt.cost) *AS* max\_maintenance\_cost

*FROM* Worker w

*LEFT JOIN* Machine m *ON* w.id *=* m.worker\_id

*LEFT JOIN* Maintenance mt *ON* m.id *=* mt.machine\_id

*GROUP BY* w.gender;

*-- List of Machines with Total Maintenance Cost:*

*SELECT* m.identifier, SUM(mt.cost) *AS* total\_maintenance\_cost

*FROM* Machine m

*LEFT JOIN* Maintenance mt *ON* m.id *=* mt.machine\_id

*GROUP BY* m.identifier;

*-- Average Order Quantity by Customer's State:*

*SELECT* a.city, AVG(o.quantity) *AS* avg\_order\_quantity

*FROM* Customer c

*JOIN* *Address* a *ON* c.address\_id *=* a.address\_id

*JOIN* Orders o *ON* c.id *=* o.customer\_id

*GROUP BY* a.city;

*-- Total Production Quantity by Worker's Gender:*

*SELECT* w.gender, SUM(p.quantity) *AS* total\_production\_quantity

*FROM* Worker w

*LEFT JOIN* Produced p *ON* w.id *=* p.worker\_id

*GROUP BY* w.gender;

*-- Average Material Rate by Material Type:*

*SELECT* m.identifier, AVG(m.rate) *AS* avg\_material\_rate

*FROM* Material m

*GROUP BY* m.identifier;

*-- Top 5 Customers with the Highest Total Order Amount:*

*SELECT* c.company\_name, SUM(o.amount) *AS* total\_order\_amount

*FROM* Customer c

*JOIN* Orders o *ON* c.id *=* o.customer\_id

*GROUP BY* c.company\_name

*ORDER BY* total\_order\_amount *DESC*

*LIMIT* 5;

*-- List of Machines with Maintenance Date of Last Maintenance:*

*SELECT* m.identifier, MAX(mt.maintenance\_date) *AS* last\_maintenance\_date

*FROM* Machine m

*LEFT JOIN* Maintenance mt *ON* m.id *=* mt.machine\_id

*GROUP BY* m.identifier;

*-- Number of Orders Placed per Month:*

*SELECT* *DATE\_FORMAT*(o.order\_date, '%Y-%m') *AS* order\_month, COUNT(o.bill\_no) *AS* order\_count

*FROM* Orders o

*GROUP BY* order\_month;

*-- Average Production Rate by Machine Type and Worker Gender:*

*SELECT* m.identifier, w.gender, AVG(p.rate) *AS* avg\_production\_rate

*FROM* Machine m

*LEFT JOIN* Worker w *ON* m.worker\_id *=* w.id

*LEFT JOIN* Produced p *ON* w.id *=* p.worker\_id

*GROUP BY* m.identifier, w.gender;

*-- List of Customers who Placed Orders but Haven't Paid Yet:*

*SELECT* c.company\_name

*FROM* Customer c

*JOIN* Orders o *ON* c.id *=* o.customer\_id

*WHERE* o.isPaymentDone *=* FALSE;