



IM2675 – Database Management Systems Group Assignment

Inventory Management System

| | |
|-----------------------|------------|
| R.M.U.L.M. Nawarathne | BM22610584 |
| Gunaseena K W S H | BM22159526 |
| Fernando W S I | BM22150202 |
| Alfred H.K.Y.D | BM22556134 |
| W.S.S Dilrukshi | BM22644794 |
| K.G.H.M Senevirathna | BM22622914 |
| P.M.T.K rathnayake | Bm22624376 |
| Dananjana K W T | Bm22509390 |

Submitted to:

Ms. Thisaru Dias

Date of submission:

28th April 2024

Table of Contents

| | |
|--|----|
| Introduction | 3 |
| ER Diagram | 4 |
| Relational Schema | 5 |
| Queries in SQL - Part 1- Create Tables | 7 |
| Queries in SQL - Part 2- Insert Value | 11 |
| Queries in SQL - Part 3- | 15 |
| Completed Database with Sample Data- | 16 |
| Annextures- | 22 |

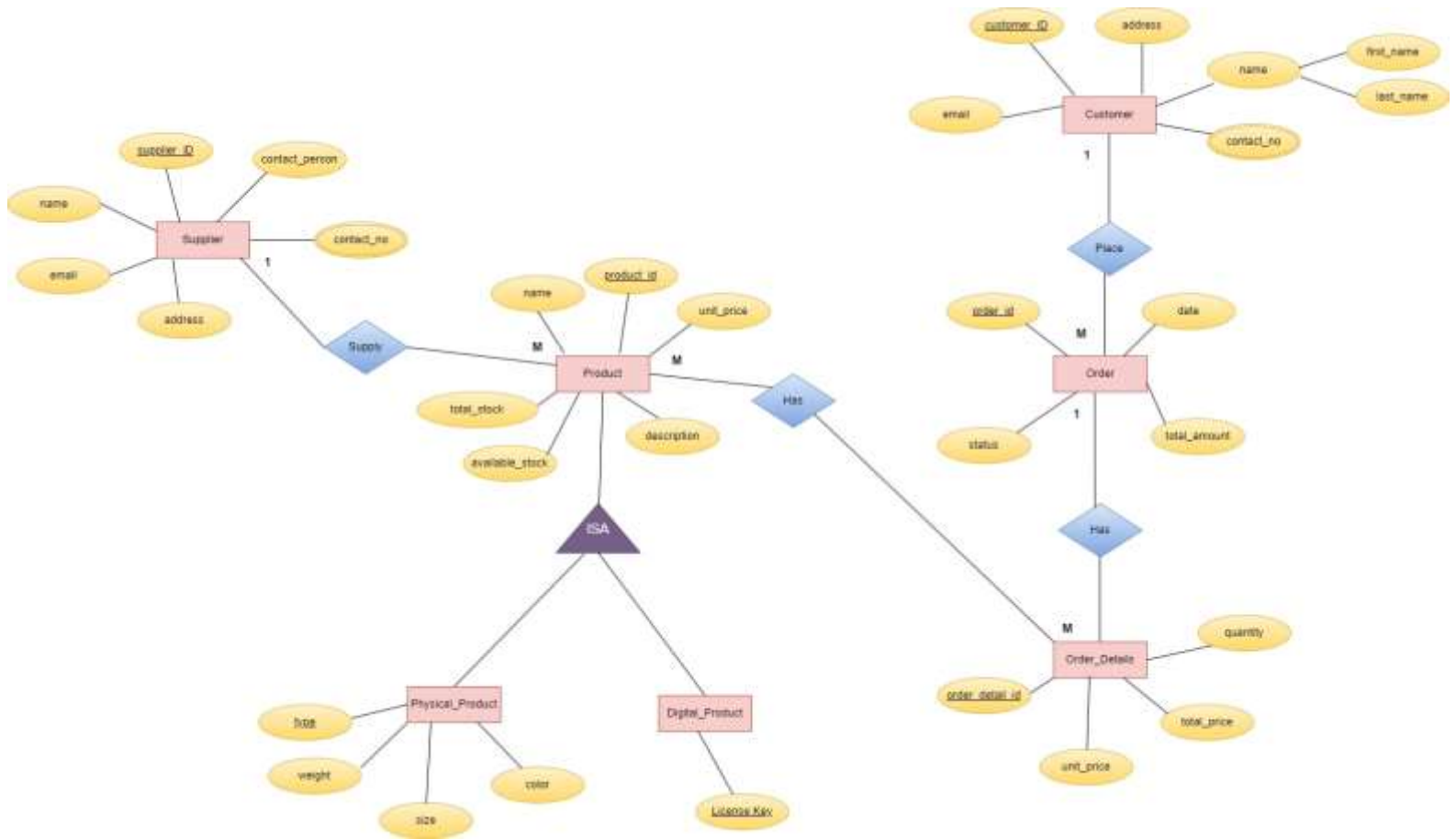
1. Introduction

We are entrusted with creating an Inventory Management System (IMS) for a retail company in this made-up scenario. The system will monitor a number of things, such as suppliers, orders, inventory levels, and product details. In order to minimize excess inventory and guarantee that the retail business has the right amount of inventory on hand to fulfill customer demand, the inventory management system (IMS) was developed.

One of the main components of the IMS would be a centralized database that would hold all pertinent data, including product names, descriptions, available quantities, reorder points, supplier information, and previous order histories. It would also include features for managing supplier connections, tracking shipments, creating purchase orders when inventory levels fall below a predetermined threshold, and updating inventory levels in real-time as transactions take place.

The retail company can increase order fulfillment procedures, decrease stockouts, prevent overstocking, improve inventory accuracy, and ultimately optimize its supply chain management for increased profitability and customer happiness by putting in place an effective IMS.

2. ER Diagram



ER Diagram Link =

https://drive.google.com/file/d/1jou6qGEWUIAnkGkGTz568c8rswY3fmkA/view?usp=drive_link

3. Relational Schema

3.1 Map Entities

a. Regular Entities

- supplier(supplie_id, name, contact_person, email, address)
- physical_product(product_id,type, name, description, total_stock, available_stock, color, weight, size, supplier_id)
- digital_product(product_id, lisenskey, name, description, total_stock, available_stock, supplier_id)
- customer(customer_id, first_name, last_name, email, address)
- order(order_id, date, status, total_amount, available_stock, customer_id)
- order_details(order_detail_id, quantity, unit_price, total_price, order_id)

b. Multivalued attributes

- supplier_contract(supplier_id, contact_no)
- customer_contract(customer_id, contact_no)

3.2 Map relationships

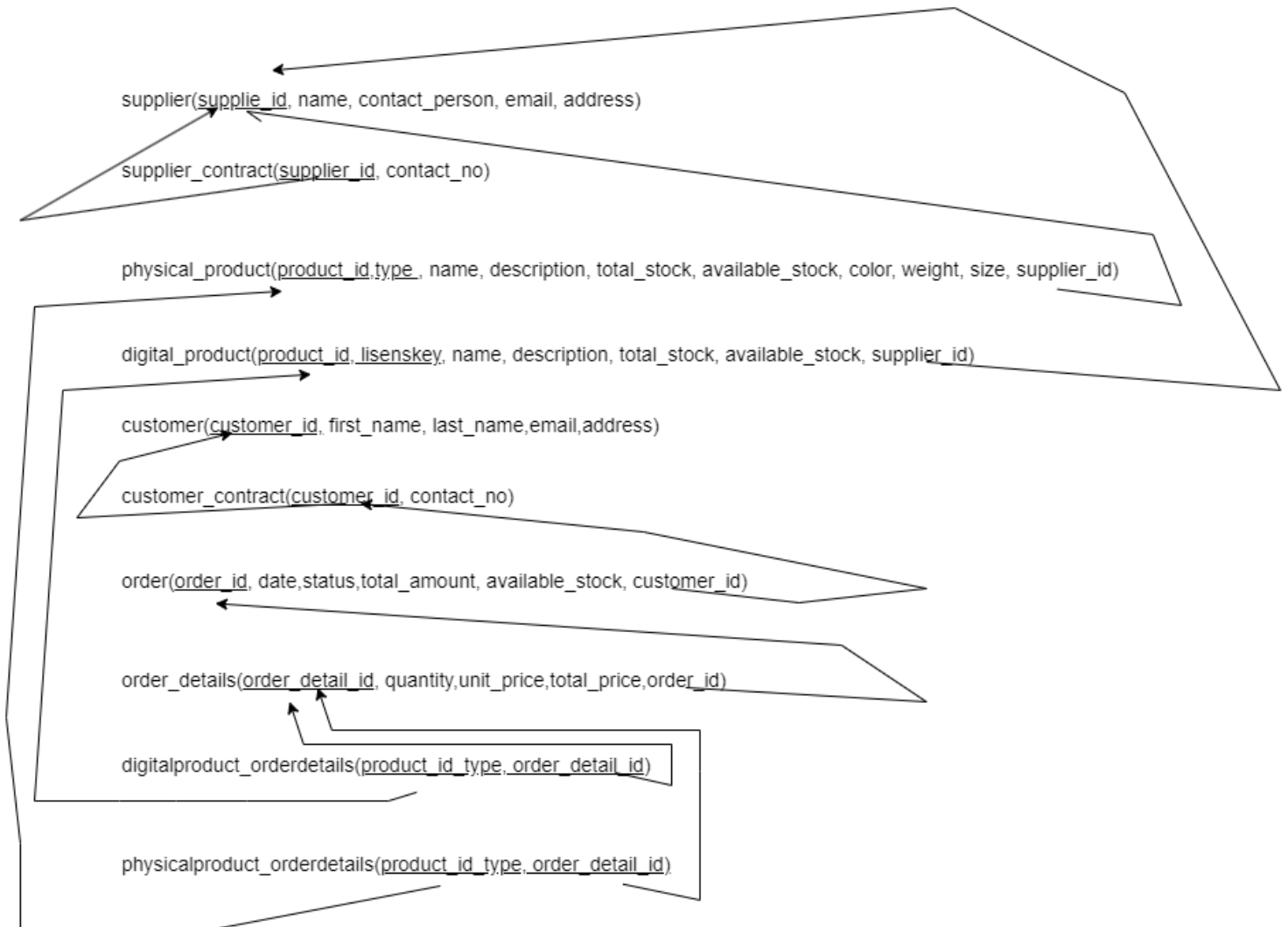
a. 1 : M

- physical_product(product_id,type, name, description, total_stock, available_stock, color, weight, size, supplier_id)
- digital_product(product_id, lisenskey, name, description, total_stock, available_stock, supplier_id)
- order(order_id, date, status, total_amount, available_stock, customer_id)
- order_details(order_detail_id, quantity, unit_price, total_price, order_id)

b. M : N

- digitalproduct_orderdetails(product_id type, order_detail_id)
- physicalproduct_orderdetails(product_id type, order_detail_id)

3.3 Final Schema



4. Queries in SQL - Part 1– Create Tables

01).

```
CREATE TABLE supplier (
    supplier_id VARCHAR(10) PRIMARY KEY,
    name VARCHAR(255),
    contact_person VARCHAR(100),
    email VARCHAR(100),
    address VARCHAR(255)
);
```

02).

```
CREATE TABLE supplier_contact (
    supplier_id VARCHAR(10),
    contact_no VARCHAR(20),

    CONSTRAINT pk_suppliercontact PRIMARY KEY (supplier_id, contact_no),
    CONSTRAINT fk_supplier_contact_supplier FOREIGN KEY (supplier_id) REFERENCES    supplier
(supplier_id)
);
```

03).

```
CREATE TABLE physical_product (
    product_id VARCHAR(10),
    type VARCHAR(255),
    name VARCHAR(255),
    description VARCHAR(255),
    unit_price DECIMAL(10, 2),
    available_stock INT,
    total_stock INT,
    supplier_id VARCHAR(10),
    size VARCHAR(50),
    weight DECIMAL(10, 2),
    color VARCHAR(50),
    CONSTRAINT pk_physicalproduct PRIMARY KEY (product_id, type),
    FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
);
```

04).

```
CREATE TABLE digital_product (
    product_id VARCHAR(10),
    licenskey VARCHAR(255),
    name VARCHAR(255),
    description VARCHAR(255),
    unit_price DECIMAL(10, 2),
    available_stock INT,
    total_stock INT,
    supplier_id VARCHAR(10),
    CONSTRAINT pk_digitalproduct PRIMARY KEY (product_id, licenskey),
    FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
);
```

05).

```
CREATE TABLE customer (
    customer_id VARCHAR(10) PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    email VARCHAR(100),
    address VARCHAR(255)
);
```

06).

```
CREATE TABLE customer_contact (
    customer_id VARCHAR(10),
    contact_no VARCHAR(20),

    CONSTRAINT pk_cuscontact PRIMARY KEY (customer_id, contact_no),
    CONSTRAINT fk_customer_contact_customer FOREIGN KEY (customer_id) REFERENCES customer
(customer_id)
);
```


07).

```
CREATE TABLE `order` (
  order_id VARCHAR(10) PRIMARY KEY,
  `date` DATE,
  customer_id VARCHAR(10),
  total_amount DECIMAL(10, 2),
  status VARCHAR(20),
  FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

08).

```
CREATE TABLE order_details (
  order_detail_id VARCHAR(10) PRIMARY KEY,
  quantity INT,
  unit_price DECIMAL(10, 2),
  total_price DECIMAL(10, 2),
  FOREIGN KEY (order_id) REFERENCES `order`(order_id)
);
```

09).

```
CREATE TABLE physicalproduct_orderdetails (
  order_detail_id VARCHAR(10),
  product_id VARCHAR(10),
  type VARCHAR(225),

  CONSTRAINT pk_guest PRIMARY KEY (order_detail_id, product_id, type),
  FOREIGN KEY (type) REFERENCES physical_product (type),
  FOREIGN KEY (order_detail_id) REFERENCES order_details (order_detail_id),
  FOREIGN KEY (product_id) REFERENCES physical_product (product_id)
);
```

10).

```
CREATE TABLE digitalproduct_orderdetails (  
  order_detail_id VARCHAR(10),  
  product_id VARCHAR(10),  
  licenskey VARCHAR(225),  
  
  CONSTRAINT pk_guest PRIMARY KEY (order_detail_id, product_id, licenskey),  
  FOREIGN KEY (licenskey) REFERENCES digital_product (licenskey),  
  FOREIGN KEY (order_detail_id) REFERENCES order_details (order_detail_id),  
  FOREIGN KEY (product_id) REFERENCES physical_product (product_id)  
);
```

5. Queries in SQL - Part 2 – Insert Value

01) Inserting data into supplier table

```
INSERT INTO supplier (supplier_id, name, contact_person, email, address)
VALUES
('Sup001', 'John Abrhum', 'John Doe', 'john.doe@gamil.com', '123 Main Street'),
('Sup002', 'Nikesh Nithil', 'Jane Smith', 'jane.smith@gamil.com', '456 Elm Street'),
('Sup003', 'Kamal Anirudh', 'Michael Johnson', 'michael.johnson@gamil.com', '789 Oak Street'),
('Sup004', 'Patrick Jane', 'Emily Brown', 'emily.brown@gamil.com', '101 Pine Street'),
('Sup005', 'Jane hightower', 'David Wilson', 'david.wilson@gamil.com', '202 Maple Street'),
('Sup006', 'Brad Smith', 'Sarah Lee', 'sarah.lee@gamil.com', '303 Cedar Street'),
('Sup007', 'Kevin Fernando', 'Kevin Miller', 'kevin.miller@gamil.com', '404 Birch Street'),
('Sup008', 'Adam Martinez', 'Amanda Davis', 'amanda.davis@gamil.com', '505 Walnut Street'),
('Sup009', 'Rebeca Silvester', 'Robert Taylor', 'robert.taylor@gamil.com', '606 Ash Street'),
('Sup010', 'Anne Sebestian', 'Jessica Martinez', 'jessica.martinez@gamil.com', '707 Oakwood Avenue');
```

02). Inserting data into supplier_contact table

```
INSERT INTO supplier_contact (supplier_id, contact_number)
VALUES
('Sup001', '123-456-7890'),
('Sup002', '234-567-8901'),
('Sup003', '345-678-9012'),
('Sup004', '456-789-0123'),
('Sup005', '567-890-1234'),
('Sup006', '678-901-2345'),
('Sup007', '789-012-3456'),
('Sup008', '890-123-4567'),
('Sup009', '901-234-5678'),
('Sup010', '012-345-6789');
```

03). Inserting data into physical_product table

```
INSERT INTO physical_product (product_id, type, name, description, unit_price, available_stock, total_stock,
supplier_id, size, weight, color)
VALUES
('Phy001', 'Electronics', 'Laptop', 'High-performance laptop', 999.99, 50, 100, 'Sup001', '15 inch', 2.5, 'Black'),
('Phy002', 'Electronics', 'Smartphone', 'Latest smartphone model', 699.99, 100, 150, 'Sup002', '6.5 inch', 0.5,
'Silver'),
('Phy003', 'Clothing', 'T-shirt', 'Cotton t-shirt', 19.99, 200, 300, 'Sup003', 'Large', 0.3, 'White'),
('Phy004', 'Clothing', 'Jeans', 'Denim jeans', 39.99, 150, 200, 'Sup004', '32W x 34L', 0.7, 'Blue'),
('Phy005', 'Books', 'Novel', 'Best-selling novel', 14.99, 300, 400, 'Sup005', 'N/A', 0.8, 'Various'),
('Phy006', 'Electronics', 'Tablet', 'High-resolution tablet', 299.99, 80, 120, 'Sup006', '10 inch', 1.0, 'Space Gray'),
('Phy007', 'Home Appliances', 'Microwave', 'Countertop microwave', 129.99, 50, 80, 'Sup007', 'Medium', 15.0,
'White'),
('Phy008', 'Home Appliances', 'Coffee Maker', 'Automatic coffee maker', 79.99, 70, 100, 'Sup008', 'N/A', 5.0,
```

```
'Black'),
('Phy009', 'Toys', 'Action Figure', 'Popular action figure', 24.99, 120, 180, 'Sup009', 'N/A', 0.2, 'Various'),
('Phy010', 'Toys', 'Board Game', 'Classic board game', 29.99, 100, 150, 'Sup010', 'N/A', 1.5, 'Various');
```

04). Inserting data into digital_product table

```
INSERT INTO digital_product (product_id, licenskey, name, description, unit_price, available_stock,
total_stock, supplier_id)
```

```
VALUES
```

```
('Dig001', 'ABC123', 'Software', 'Digital software product', 49.99, 200, 300, 'Sup001'),
('Dig002', 'DEF456', 'E-book', 'Digital book in PDF format', 9.99, 500, 600, 'Sup002'),
('Dig003', 'GHI789', 'Online Course', 'Digital course with video lectures', 79.99, 100, 150, 'Sup003'),
('Dig004', 'JKL012', 'Subscription', 'Digital subscription service', 19.99, 300, 400, 'Sup004'),
('Dig005', 'MNO345', 'Music Album', 'Digital music album in MP3 format', 14.99, 150, 200, 'Sup005'),
('Dig006', 'PQR678', 'Movie', 'Digital movie download', 9.99, 200, 250, 'Sup006'),
('Dig007', 'STU901', 'Game', 'Digital video game', 39.99, 100, 150, 'Sup007'),
('Dig008', 'VWX234', 'Audio Book', 'Digital audio book', 29.99, 80, 120, 'Sup008'),
('Dig009', 'YZA567', 'Software Tool', 'Digital software tool', 69.99, 150, 200, 'Sup009'),
('Dig010', 'BCD890', 'E-magazine', 'Digital magazine subscription', 4.99, 300, 400, 'Sup010');
```

05). Inserting data into customer table

```
INSERT INTO customer (customer_id, first_name, last_name, email, address)
```

```
VALUES
```

```
('Cus001', 'Alice', 'Johnson', 'alice.johnson@example.com', '123 Pine Street'),
('Cus002', 'Bob', 'Smith', 'bob.smith@example.com', '456 Oak Street'),
('Cus003', 'Charlie', 'Williams', 'charlie.williams@example.com', '789 Maple Street'),
('Cus004', 'David', 'Brown', 'david.brown@example.com', '101 Elm Street'),
('Cus005', 'Emma', 'Jones', 'emma.jones@example.com', '202 Cedar Street'),
('Cus006', 'Frank', 'Taylor', 'frank.taylor@example.com', '303 Birch Street'),
('Cus007', 'Grace', 'Anderson', 'grace.anderson@example.com', '404 Walnut Street'),
('Cus008', 'Henry', 'Martinez', 'henry.martinez@example.com', '505 Ash Street'),
('Cus009', 'Ivy', 'Garcia', 'ivy.garcia@example.com', '606 Oakwood Avenue'),
('Cus010', 'Jack', 'Lopez', 'jack.lopez@example.com', '707 Elmwood Drive');
```

06). Inserting data into customer_contact table

```
INSERT INTO customer_contact (customer_id, contact_number)
```

```
VALUES
```

```
('Cus001', '123-456-7890'),
('Cus002', '234-567-8901'),
('Cus003', '345-678-9012'),
('Cus004', '456-789-0123'),
('Cus005', '567-890-1234'),
('Cus006', '678-901-2345'),
```

```
(('Cus007', '789-012-3456'),
('Cus008', '890-123-4567'),
('Cus009', '901-234-5678'),
('Cus010', '012-345-6789'));
```

07). Inserting data into order_table table

```
INSERT INTO order (order_id, date, customer_id, total_amount, status)
VALUES
('Ord001', '2024-04-01', 'Cus001', 299.99, 'Shipped'),
('Ord002', '2024-04-02', 'Cus002', 199.99, 'Delivered'),
('Ord003', '2024-04-03', 'Cus003', 399.99, 'Pending'),
('Ord004', '2024-04-04', 'Cus004', 599.99, 'Shipped'),
('Ord005', '2024-04-05', 'Cus005', 499.99, 'Delivered'),
('Ord006', '2024-04-06', 'Cus006', 99.99, 'Shipped'),
('Ord007', '2024-04-07', 'Cus007', 199.99, 'Delivered'),
('Ord008', '2024-04-08', 'Cus008', 149.99, 'Pending'),
('Ord009', '2024-04-09', 'Cus009', 799.99, 'Shipped'),
('Ord010', '2024-04-10', 'Cus010', 349.99, 'Delivered');
```

08). Inserting data into order_details table

```
INSERT INTO order_details (order_detail_id, order_id, quantity, unit_price, total_price)
VALUES
('Od001', 'Ord001', 1, 999.99, 999.99),
('Od002', 'Ord002', 2, 699.99, 1399.98),
('Od003', 'Ord003', 3, 19.99, 59.97),
('Od004', 'Ord004', 1, 39.99, 39.99),
('Od005', 'Ord005', 2, 14.99, 29.98),
('Od006', 'Ord006', 1, 299.99, 299.99),
('Od007', 'Ord007', 1, 129.99, 129.99),
('Od008', 'Ord008', 2, 79.99, 159.98),
('Od009', 'Ord009', 3, 24.99, 74.97),
('Od010', 'Ord010', 1, 29.99, 29.99);
```

09). Inserting data into physicalproduct_orderdetails table

```

INSERT INTO physicalproduct_orderdetails (order_detail_id, product_id, type)
VALUES
('Od001', 'Phy001', 'Electronics'),
('Od002', 'Phy002', 'Electronics'),
('Od003', 'Phy003', 'Clothing'),
('Od004', 'Phy004', 'Clothing'),
('Od005', 'Phy005', 'Books'),
('Od006', 'Phy006', 'Electronics'),
('Od007', 'Phy007', 'Home Appliances'),
('Od008', 'Phy008', 'Home Appliances'),
('Od009', 'Phy009', 'Toys'),
('Od010', 'Phy010', 'Toys');

```

10). Inserting data into digitalproduct_orderdetails table

```

INSERT INTO digitalproduct_orderdetails (order_detail_id, product_id, licenskey)
VALUES
('Od001', 'Dig001', 'ABC123'),
('Od002', 'Dig002', 'DEF456'),
('Od003', 'Dig003', 'GHI789'),
('Od004', 'Dig004', 'JKL012'),
('Od005', 'Dig005', 'MNO345'),
('Od006', 'Dig006', 'PQR678'),
('Od007', 'Dig007', 'STU901'),
('Od008', 'Dig008', 'VWX234'),
('Od009', 'Dig009', 'YZA567'),
('Od010', 'Dig010', 'BCD890');

```

6. Queries in SQL – Part 3

| Query | SQL Statement to Retrieve the Query |
|--|--|
| 01) Retrieve all suppliers whose names contain "John" (Refer : ANNEX 1) | SELECT * FROM supplier WHERE name LIKE '%John%'; |
| 02) Retrieve all physical products with unit prices between \$20 and \$50 (Refer : ANNEX 2) | SELECT * FROM physical_product WHERE unit_price BETWEEN 20 AND 50; |
| 03) Retrieve all orders with a total amount greater than \$400 (Refer : ANNEX 3) | SELECT * FROM order WHERE total_amount > 400; |
| 04) Retrieve all customers with last names starting with "J" (Refer : ANNEX 4) | SELECT * FROM customer WHERE last_name LIKE 'J%'; |
| 05) Retrieve all digital products ordered by product name in descending order (Refer : ANNEX 5) | SELECT * FROM digital_product ORDER BY name DESC; |
| 06) Retrieve the lowest unit price among physical products (Refer : ANNEX 6) | SELECT MIN(unit_price) AS lowest_price FROM physical_product; |
| 07) Retrieve all orders placed on April 4, 2024 (Refer : ANNEX 7) | SELECT * FROM order WHERE date = '2024-04-04'; |
| 08) Retrieve all suppliers located on streets with names starting with "M" (Refer : ANNEX 8) | SELECT * FROM supplier WHERE address LIKE '%M%'; |
| 09) Retrieve all physical products with quantities on hand greater than 100 (Refer : ANNEX 9) | SELECT * FROM physical_product WHERE quantity_on_hand > 100; |
| 10) Retrieve all information about suppliers whose addresses contain the letter "M" anywhere in the address field. (Refer : ANNEX 10) | SELECT MAX(total_price) AS highest_total_price FROM order_details; |

7. Completed Database with Sample Data

01) Sample data for supplier table

| supplier_id | name | contact_person | email | address |
|-------------|-------------|------------------|----------------------------|--------------------|
| Sup001 | Supplier 1 | John Doe | john.doe@gamil.com | 123 Main Street |
| Sup002 | Supplier 2 | Jane Smith | jane.smith@gmail.com | 456 Elm Street |
| Sup003 | Supplier 3 | Michael Johnson | michael.johnson@gamil.com | 789 Oak Street |
| Sup004 | Supplier 4 | Emily Brown | emily.brown@gamil.com | 101 Pine Street |
| Sup005 | Supplier 5 | David Wilson | david.wilson@gmail.com | 202 Maple Street |
| Sup006 | Supplier 6 | Sarah Lee | sarah.lee@gmail.com | 303 Cedar Street |
| Sup007 | Supplier 7 | Kevin Miller | kevin.miller@gmail.com | 404 Birch Street |
| Sup008 | Supplier 8 | Amanda Davis | amanda.davis@gamil.com | 505 Walnut Street |
| Sup009 | Supplier 9 | Robert Taylor | robert.taylor@gmail.com | 606 Ash Street |
| Sup010 | Supplier 10 | Jessica Martinez | jessica.martinez@gmail.com | 707 Oakwood Avenue |

02) Sample data for supplier_contact table

| supplier_id | contact_no |
|-------------|--------------|
| Sup001 | 123-456-7890 |
| Sup002 | 234-567-8901 |
| Sup003 | 345-678-9012 |
| Sup004 | 456-789-0123 |
| Sup005 | 567-890-1234 |
| Sup006 | 678-901-2345 |
| Sup007 | 789-012-3456 |
| Sup008 | 890-123-4567 |
| Sup009 | 901-234-5678 |
| Sup010 | 012-345-6789 |

03) Sample data for physical_product table

| product_id | type | name | description | unit_price | available_stock | total_stock | supplier_id | size | weight | color |
|------------|-----------------|---------------|-------------------------|------------|-----------------|-------------|-------------|-----------|--------|------------|
| Phy001 | Electronics | Laptop | High-performance laptop | 999.99 | 50 | 100 | sup001 | 15 inch | 2.5 | Black |
| Phy002 | Electronics | Smartphone | Latest smartphone model | 699.99 | 100 | 150 | sup002 | 6.5 inch | 0.5 | Silver |
| Phy003 | Clothing | T-shirt | Cotton t-shirt | 19.99 | 200 | 300 | sup003 | Large | 0.3 | White |
| Phy004 | Clothing | Jeans | Denim jeans | 39.99 | 150 | 200 | sup004 | 32W x 34L | 0.7 | Blue |
| Phy005 | Books | Novel | Best-selling novel | 14.99 | 300 | 400 | sup005 | N/A | 0.8 | Various |
| Phy006 | Electronics | Tablet | High-resolution tablet | 299.99 | 80 | 120 | sup006 | 10 inch | 1 | Space Gray |
| Phy007 | Home Appliances | Microwave | Countertop microwave | 129.99 | 50 | 80 | sup007 | Medium | 15 | White |
| Phy008 | Home Appliances | Coffee Maker | Automatic coffee maker | 79.99 | 70 | 100 | sup008 | N/A | 5 | Black |
| Phy009 | Toys | Action Figure | Popular action figure | 24.99 | 120 | 180 | sup009 | N/A | 0.2 | Various |
| Phy010 | Toys | Board Game | Classic board game | 29.99 | 100 | 150 | sup010 | N/A | 1.5 | Various |

04) Sample data for gidital_product table

| product_id | license_y | name | description | unit_price | available_stock | total_stock | supplier_id |
|------------|-----------|---------------|------------------------------------|------------|-----------------|-------------|-------------|
| Dig001 | ABC123 | Software | Digital software product | 49.99 | 200 | 300 | sup001 |
| Dig002 | DEF456 | E-book | Digital book in PDF format | 9.99 | 500 | 600 | sup002 |
| Dig003 | GHI789 | Online Course | Digital course with video lectures | 79.99 | 100 | 150 | sup003 |
| Dig004 | JKL012 | Subscription | Digital subscription service | 19.99 | 300 | 400 | sup004 |
| Dig005 | MNO345 | Music Album | Digital music album in MP3 format | 14.99 | 150 | 200 | sup005 |
| Dig006 | PQR678 | Movie | Digital movie download | 9.99 | 200 | 250 | sup006 |
| Dig007 | STU901 | Game | Digital video game | 39.99 | 100 | 150 | sup007 |
| Dig008 | VWX234 | Audio Book | Digital audio book | 29.99 | 80 | 120 | sup008 |
| Dig009 | YZA567 | Software Tool | Digital software tool | 69.99 | 150 | 200 | sup009 |
| Dig010 | BCD890 | E-magazine | Digital magazine subscription | 4.99 | 300 | 400 | sup010 |

05) Sample data for order table

| order_id | date | customer_id | Total_amount | status |
|----------|----------|-------------|--------------|-----------|
| Ord001 | 4/1/2024 | Cus001 | 299.99 | Shipped |
| Ord002 | 4/2/2024 | Cus002 | 199.99 | Delivered |
| Ord003 | 4/3/2024 | Cus003 | 399.99 | Pending |
| Ord004 | 4/4/2024 | Cus004 | 599.99 | Shipped |
| Ord005 | 4/5/2024 | Cus005 | 499.99 | Delivered |
| Ord006 | 4/6/2024 | Cus006 | 99.99 | Shipped |
| Ord007 | 4/7/2024 | Cus007 | 199.99 | Delivered |
| Ord008 | 4/8/2024 | Cus008 | 149.99 | Pending |
| Ord009 | 4/9/2024 | Cus009 | 799.99 | Shipped |

| | | | | |
|--------|-----------|--------|--------|-----------|
| Ord010 | 4/10/2024 | Cus010 | 349.99 | Delivered |
|--------|-----------|--------|--------|-----------|

06) Sample data for order_details table

| order_details_id | quantity | order_id | unit_price | total_price |
|------------------|----------|----------|------------|-------------|
| Od001 | 1 | Ord001 | 999.99 | 999.99 |
| Od002 | 2 | Ord002 | 699.99 | 1399.98 |
| Od003 | 3 | Ord003 | 19.99 | 59.97 |
| Od004 | 1 | Ord004 | 39.99 | 39.99 |
| Od005 | 2 | Ord005 | 14.99 | 29.98 |
| Od006 | 1 | Ord006 | 299.99 | 299.99 |
| Od007 | 1 | Ord007 | 129.99 | 129.99 |
| Od008 | 2 | Ord008 | 79.99 | 159.98 |
| Od009 | 3 | Ord009 | 24.99 | 74.97 |
| Od010 | 1 | Ord010 | 29.99 | 29.99 |

07) Sample data for customer table

| customer_id | first_name | last_name | email | address |
|-------------|------------|-----------|--|--------------------|
| Cus001 | Alice | Johnson | alice.johnson@gmail.com | 123 Pine Street |
| Cus002 | Bob | Smith | bob.smith@gmail.com | 456 Oak Street |
| Cus003 | Charlie | Williams | charlie.williams@gmail.com | 789 Maple Street |
| Cus004 | David | Brown | david.brown@gmail.com | 101 Elm Street |
| Cus005 | Emma | Jones | emma.jones@gmail.com | 202 Cedar Street |
| Cus006 | Frank | Taylor | frank.taylor@gmail.com | 303 Birch Street |
| Cus007 | Grace | Anderson | grace.anderson@gmail.com | 404 Walnut Street |
| Cus008 | Henry | Martinez | henry.martinez@gmail.com | 505 Ash Street |
| Cus009 | Ivy | Garcia | ivy.garcia@gmail.com | 606 Oakwood Avenue |
| Cus010 | Jack | Lopez | jack.lopez@gmail.com | 707 Elmwood Drive |

08) Sample data for customer_contact table

| customer_id | contact_no |
|-------------|--------------|
| Cus001 | 077-456-7890 |
| Cus002 | 075-567-8901 |
| Cus003 | 091-678-9012 |
| Cus004 | 081-789-0123 |
| Cus005 | 072-890-1234 |
| Cus006 | 075-901-2345 |
| Cus007 | 076-012-3456 |
| Cus008 | 075-123-4567 |
| Cus009 | 072-234-5678 |
| Cus010 | 076-345-6789 |

09) Sample data for physicalproduct_orderdeatils table

| order_detail_id | product_id | type |
|-----------------|------------|-----------------|
| Od001 | Phy001 | Electronics |
| Od002 | Phy002 | Electronics |
| Od003 | Phy003 | Clothing |
| Od004 | Phy004 | Clothing |
| Od005 | Phy005 | Books |
| Od006 | Phy006 | Electronics |
| Od007 | Phy007 | Home Appliances |
| Od008 | Phy008 | Home Appliances |
| Od009 | Phy009 | Toys |
| Od010 | Phy010 | Toys |

10) Sample data for digitalproduct_orderdetails table

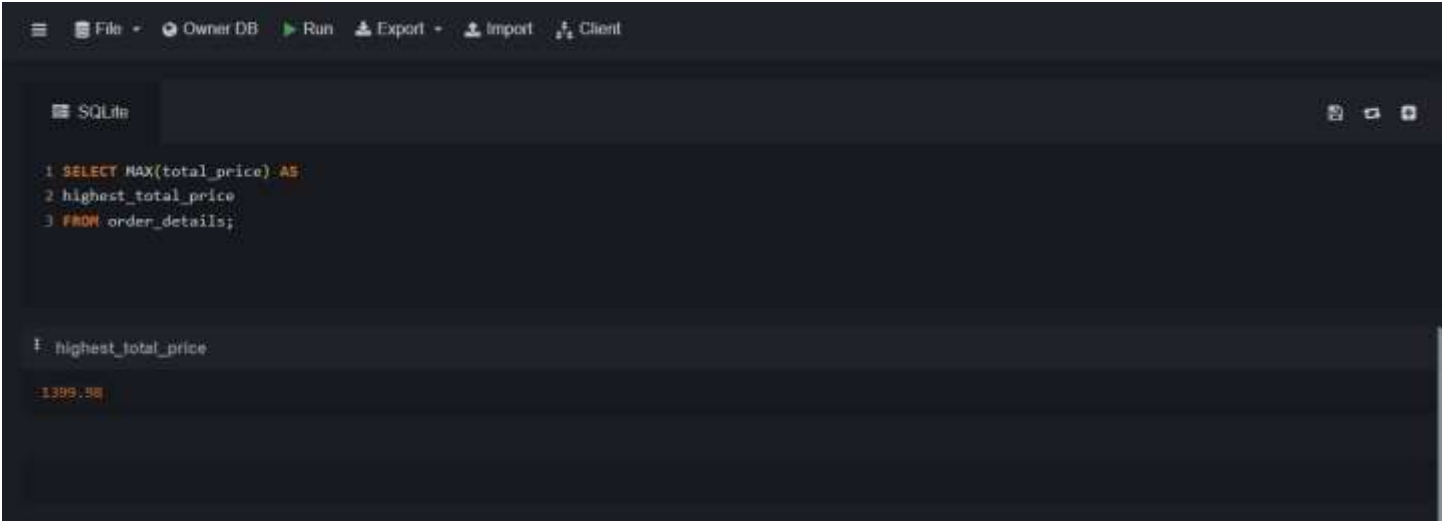
| order_details_id | product_id | lisecskey |
|------------------|------------|-----------|
| Od001 | Dig001 | ABC123 |
| Od002 | Dig002 | DEF456 |
| Od003 | Dig003 | GHI789 |

| | | |
|-------|--------|--------|
| Od004 | Dig004 | JKL012 |
| Od005 | Dig005 | MNO345 |
| Od006 | Dig006 | PQR678 |
| Od007 | Dig007 | STU901 |
| Od008 | Dig008 | VWX234 |
| Od009 | Dig009 | YZA567 |
| Od010 | Dig010 | BCD890 |

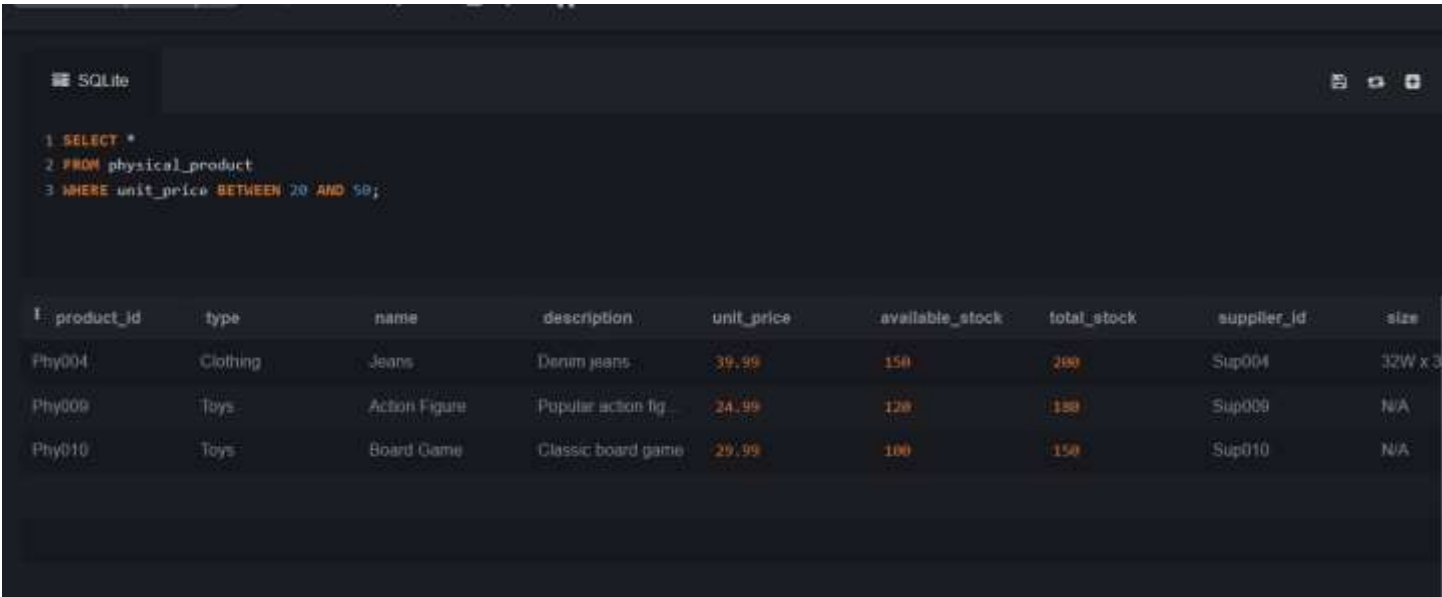
8. Annextures

Link for Annex = https://drive.google.com/drive/folders/1feSlplpApDYy65vDhC4ySGWLEySXX4nD?usp=drive_link

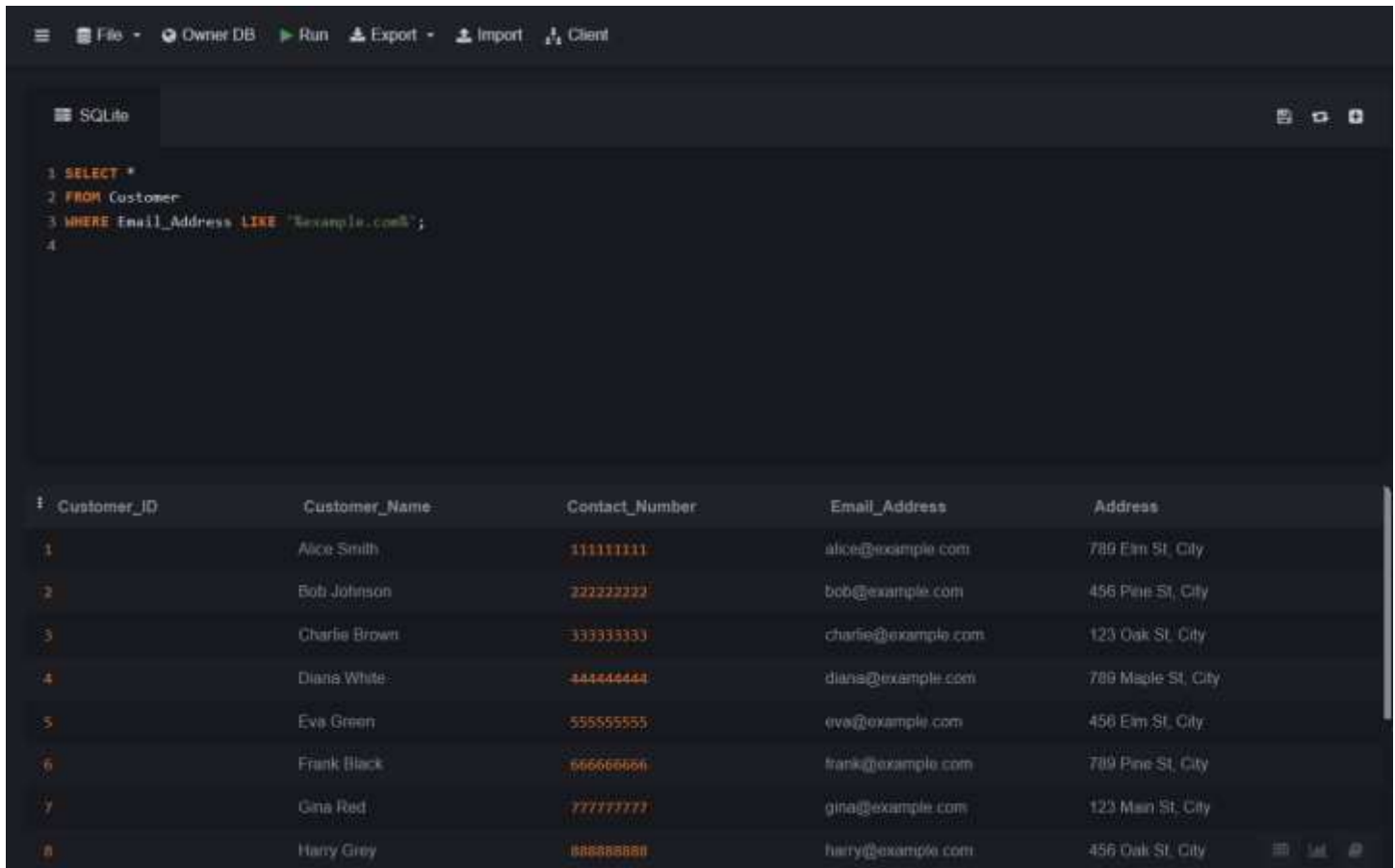
ANNEX 1



ANNEX 2



ANNEX 3



The screenshot shows a SQLite database application interface. At the top, there is a menu bar with options: File, Owner DB, Run, Export, Import, and Client. Below the menu bar, the title bar reads 'SQLite'. The main area displays a SQL query:

```

1 SELECT *
2 FROM Customer
3 WHERE Email_Address LIKE '%example.com';
4

```

Below the query, the results are displayed in a table with the following columns: Customer_ID, Customer_Name, Contact_Number, Email_Address, and Address. The table contains 8 rows of data.

| Customer_ID | Customer_Name | Contact_Number | Email_Address | Address |
|-------------|---------------|----------------|---------------------|--------------------|
| 1 | Alice Smith | 1111111111 | alice@example.com | 789 Elm St, City |
| 2 | Bob Johnson | 2222222222 | bob@example.com | 456 Pine St, City |
| 3 | Charlie Brown | 3333333333 | charlie@example.com | 123 Oak St, City |
| 4 | Diana White | 4444444444 | diana@example.com | 789 Maple St, City |
| 5 | Eva Green | 5555555555 | eva@example.com | 456 Elm St, City |
| 6 | Frank Black | 6666666666 | frank@example.com | 789 Pine St, City |
| 7 | Gina Red | 7777777777 | gina@example.com | 123 Main St, City |
| 8 | Harry Grey | 8888888888 | harry@example.com | 456 Oak St, City |

ANNEX 4

SQLite

```

1 SELECT *
2 FROM customer
3 WHERE last_name LIKE 'J%';

```

| customer_id | first_name | last_name | email |
|-------------|------------|-----------|-------------------------|
| Cus001 | Alice | Johnson | alice.johnson@gmail.com |
| Cus005 | Emma | Jones | emma.jones@gmail.com |

ANNEX 5

SQLite

```

1 SELECT *
2 FROM digital_product
3 ORDER BY name DESC;

```

| product_id | licensekey | name | description | unit_price | available_stock | total_sales |
|------------|------------|---------------|---------------------------------|------------|-----------------|-------------|
| Dig004 | JKL012 | Subscription | Digital subscription service | 19.99 | 300 | 4 |
| Dig009 | YZA567 | Software Tool | Digital software tool | 69.99 | 150 | 2 |
| Dig001 | ABC123 | Software | Digital software product | 49.99 | 200 | 3 |
| Dig003 | GHI789 | Online Course | Digital course with video le... | 79.99 | 100 | 1 |
| Dig005 | MNO345 | Music Album | Digital music album in MP... | 14.99 | 150 | 2 |
| Dig006 | PQR578 | Movie | Digital movie download | 9.99 | 200 | 2 |
| Dig007 | STU801 | Game | Digital video game | 39.99 | 100 | 1 |
| Dig010 | BCD890 | E-magazine | Digital magazine subscrip... | 4.99 | 300 | 4 |
| Dig002 | DEF456 | E-book | Digital book in PDF format | 9.99 | 500 | 6 |
| Dig008 | VWX234 | Audio Book | Digital audio book | 29.99 | 80 | 1 |

ANNEX 6

SQLite

```
1 SELECT MIN(unit_price) AS lowest_price
2 FROM physical_product;
```

lowest_price

14.99

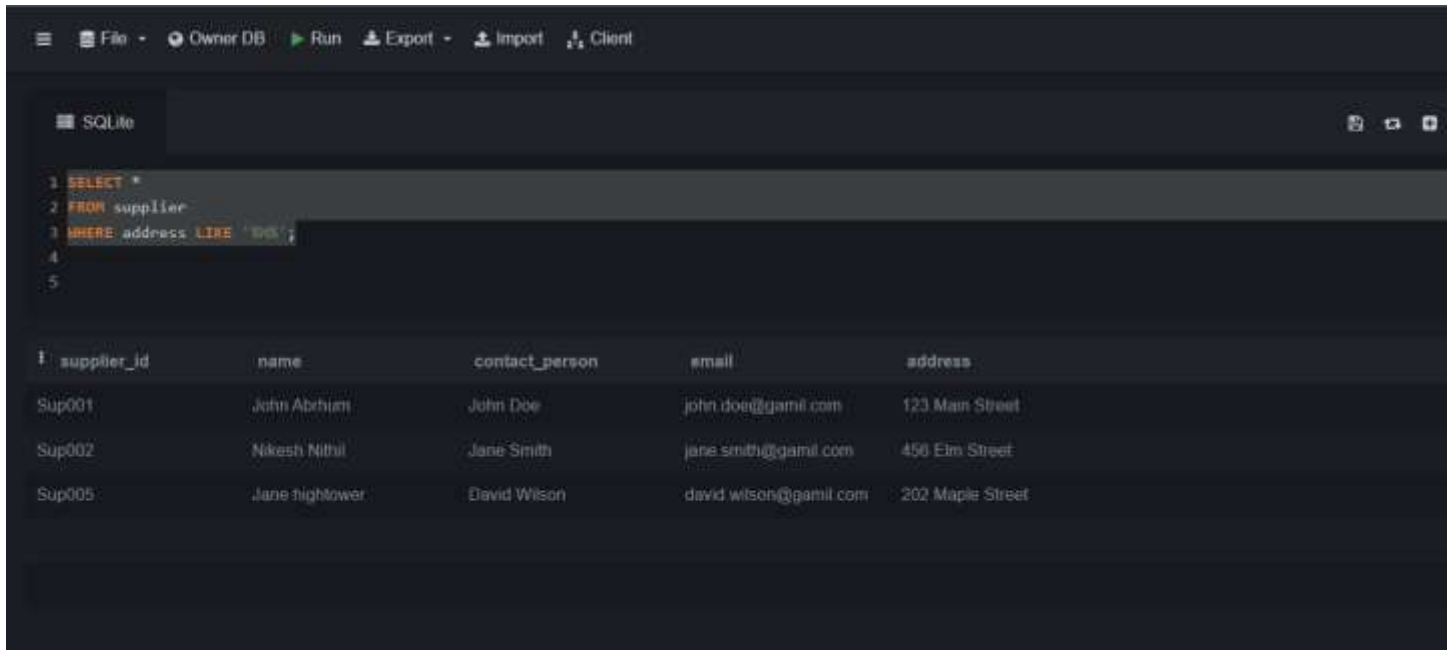
ANNEX 7

SQLite

```
1 SELECT *
2 FROM "order"
3 WHERE date = '2024-04-04';
```

| order_id | date | customer_id | total_amount |
|----------|------------|-------------|--------------|
| Ord004 | 2024-04-04 | Cus004 | 599.99 |

ANNEX 8



The screenshot shows a SQLite database client interface. At the top, there is a menu bar with options: File, Owner DB, Run, Export, Import, and Client. Below the menu bar, the title bar reads "SQLite". The main area displays a SQL query:

```
1 SELECT *
2 FROM supplier
3 WHERE address LIKE '100%';
4
5
```

Below the query, the results are displayed in a table with the following columns: supplier_id, name, contact_person, email, and address. The table contains three rows of data:

| supplier_id | name | contact_person | email | address |
|-------------|----------------|----------------|------------------------|------------------|
| Sup001 | John Abraham | John Doe | john.doe@gmail.com | 123 Main Street |
| Sup002 | Nikesh Nithil | Jane Smith | jane.smith@gmail.com | 456 Elm Street |
| Sup005 | Jane Hightower | David Wilson | david.wilson@gmail.com | 202 Maple Street |

ANNEX 9

SQLite

```
1 SELECT *
2 FROM physical_product
3 WHERE available_stock > 100;
```

| product_id | type | name | description | unit_price | available_stock | total_stock | supplier_id | size |
|------------|----------|---------------|-----------------------|------------|-----------------|-------------|-------------|---------|
| Phy003 | Clothing | T-shirt | Cotton t-shirt | 19.99 | 200 | 300 | Sup003 | Large |
| Phy004 | Clothing | Jeans | Denim jeans | 39.99 | 150 | 200 | Sup004 | 32W x 3 |
| Phy005 | Books | Novel | Best-selling novel | 14.99 | 300 | 400 | Sup005 | N/A |
| Phy009 | Toys | Action Figure | Popular action fig... | 24.99 | 120 | 180 | Sup009 | N/A |

ANNEX 10

SQLite

```
1 SELECT MAX(total_price) AS
2 highest_total_price
3 FROM order_details;
```

| highest_total_price |
|---------------------|
| 1389.98 |