```sql
-- DBMS Assignment 4
-- 1. Arjun Tandon - 2022095
-- 2. Aryan Jain - 2022111
-- 3. Sameer Singh Godara - 2022439
-- 4. Sarthak Sharma - 2022456

-- 1. List the products that are present in all the warehouses.
SELECT productID, COUNT(*) as num_warehouses FROM stores
group by ProductID
HAVING COUNT(*) = (
    SELECT COUNT(*) FROM warehouse
);

-- 2. List the most popular (most ordered) product
SELECT productID, SUM(quantity) as num FROM orderitem
GROUP by productID
HAVING num = (
    SELECT MAX(SUM(quantity)) FROM orderitem
    GROUP by productID
);

-- 3. List all products which are not present at any warehouse at all.
SELECT product.ProductID
FROM product LEFT JOIN stores
ON product.ProductID = stores.ProductID
WHERE stores.WarehouseID IS NULL OR stores.Quantity = 0;

-- 4. List all Warehouses having no products
SELECT warehouseID as wid
FROM stores
GROUP BY warehouseID
HAVING SUM(quantity) = 0
UNION
SELECT warehouse.WarehouseID as wid
FROM warehouse LEFT OUTER JOIN stores
ON warehouse.WarehouseID = stores.WarehouseID
WHERE stores.ProductID IS NULL;


-- 5. Delete all records in 'stores' table where the quantity of the product in a
warehouse is 0.
DELETE FROM stores
WHERE quantity = 0;

-- 6. Update prices of all products in the 'product' table such that, if price of the
product is above 2500,
-- you set it to 2500, else you set it to 2000.
UPDATE product
SET price = CASE
    WHEN price >= 2500 THEN 2500
    ELSE 2000
END
WHERE LOWER(name) LIKE "a%";
```

```sql
-- 7. Write a query to calculate the total quantity of products in each
-- warehouse and add a column in the product table called quantity
-- add the summed quantity to that column in the product table.
ALTER table product
ADD column total_qty INT DEFAULT 0;

UPDATE product p
JOIN (
    SELECT productID, SUM(quantity) as qty
    FROM stores
    GROUP BY productID
) t ON t.productID = p.productID
SET p.total_qty = t.qty;

-- 8. Find the top 5 users who spent the most on orders
SELECT User.UserName, SUM(Orders.TotalAmount) AS TotalSpent
FROM User
JOIN Orders ON User.UserID = Orders.UserID
GROUP BY User.UserID, User.UserName
ORDER BY TotalSpent DESC
LIMIT 5;


-- 9. Find the top 3 products with the highest total quantity sold,
-- including the corresponding brand and the total revenue generated from those products
SELECT P.ProductName, B.BrandName, SUM(OI.Quantity) AS TotalQuantitySold, SUM(OI.Quantity
* P.Price) AS TotalRevenue
FROM Product P
JOIN OrderItem OI ON P.ProductID = OI.ProductID
JOIN Brand B ON P.BrandID = B.BrandID
GROUP BY P.ProductID, P.ProductName, B.BrandName
ORDER BY TotalQuantitySold DESC
LIMIT 3;

-- 10. Retrieve the products present at only one warehouse
SELECT ProductID, COUNT(WareHouseID) AS WarehouseCount
FROM Stores
GROUP BY ProductID
HAVING COUNT(WareHouseID) = 1;

-- Queries violating database constraints

-- 1. Inserting a user without specifying a value for a NOT NULL column
INSERT INTO User (CustomerFlag, Gender)
VALUES (1, 'Male');

-- 2. Inserting a duplicate primary key in the User table
INSERT INTO User (UserID, CustomerFlag, UserName, HouseNumber, Locality, City, State,
PinCode, Country, Gender)
VALUES (1, 1, 'John Doe', '123 Main St', 'Anytown', 'AnyCity', 'AnyState', 12345,
'AnyCountry', 'Male');

-- 3. Inserting a product with a non-existent BrandID in the Product table
INSERT INTO Product (ProductName, Description, Price, BrandID)
```

```sql
VALUES ('New Product', 'Description', 100.00, 100);

-- 4.  Updating a duplicate primary key in the User table
UPDATE User
SET UserID = 1
WHERE UserID = 2;

-- 5. Delete a brand without deleting its products from the Product table (We have not
applied DELETE ON CASCADE)
DELETE FROM Brand
WHERE BrandID = 3;
```

Date

① $\gamma_{ProductID}(\rho_{num\_warehouses \leftarrow COUNT(*)}(\gamma_{productID}(stores))) \div \gamma_{COUNT(*)}(warehouses)$

② $\pi_{ProductID}(\sigma_{sum(quantity) = max(sum(quantity))}(\gamma_{ProductID}(\pi_{ProductID, sum(quantity)}(OrderItem))))$

3) $\pi_{productId}(\sigma_{(store.WarehouseID = NULL \cup stores.quantity = 0)}(product \bowtie stores))$

4) $\pi wid(\sigma_{\Sigma quantity = 0}(stores)) \cup \pi_{warehouseID}(\sigma_{productID, isnull}(warehouse \bowtie stores$

8th $\Pi Username, TotalSpent(\rho User(UserID, UserName), \rho Orders(UserID, TotalAmount), (User \bowtie Orders) \bowtie_i UserID = User.UserID(\gamma UserID, UserName, TotalSpent(User \bowtie Orders))$

10th $\Pi ProductID, WarehouseCount(\gamma ProductID, COUNT(warehouseID) AS warehouseCount(Stores))$