SQL Project

Begin by providing a brief background of an ice-cream shop that led to the need for the SQL based solution. This could be an overview of the business environment, the challenges the organization was facing, or the specific needs that prompted the analysis.

# Objectives:

Clearly define the objectives of the case study. Outline what the case study aims to achieve, such as improving decision-making processes, enhancing data accessibility, or optimizing business operations through SQL queries.

# Dataflow of the database:



customers



category



Id (pk) name

brandid (fk) categoryid (fk) price

brand

Id (pk) name

id (pk) name

phonenumber nearorfar

icecream

orderdetails

id (pk) orderid (fk)

icecreamid (fk) quantity

orders

id (pk) customerid (fk) orderdate

weekday

Id (pk) name

**-- Creating the database, tables and inserting the values in the tables**

CREATE DATABASE icecreamshop; USE icecreamshop;

CREATE TABLE category ( id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

INSERT INTO category (id, name) VALUES

(1,"cone"),

(2,"cup"),

(3,"stick/kulfi"),

(4,"packet"),

(5,"others")

;

CREATE TABLE brand ( id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

INSERT INTO brand (id, name) VALUES

(1,"creamery"),

(2,"havmor"),

(3,"motherDairy"),

(4,"vadilal"),

(5,"frozenFrenzy"),

(6,"berryBliss")

;

CREATE TABLE iceCream ( id INT NOT NULL,

name VARCHAR(50) NOT NULL,

brandid INT NOT NULL, categoryid INT NOT NULL, price INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (brandid) REFERENCES brand (id),

FOREIGN KEY (categoryid) REFERENCES category (id)

);

-- data has been imported through "table data import wizard"

CREATE TABLE customer ( id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

phoneNumber INT NOT NULL, nearorfar VARCHAR(50) NOT NULL

);

-- data has been imported through "table data import wizard"

CREATE TABLE orders ( id INT NOT NULL,

customerid INT NOT NULL, orderDate DATE NOT NULL, weekday INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (customerid) REFERENCES customer (id)

);

-- data has been imported through "table data import wizard"

CREATE TABLE ordersdetails ( id INT NOT NULL,

orderid INT NOT NULL,

icecreamid INT NOT NULL, quantity INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (orderid) REFERENCES orders (id),

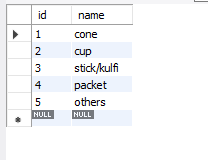
FOREIGN KEY (icecreamid) REFERENCES icecream (id)

);

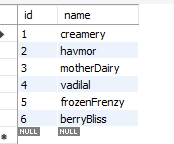
-- data has been imported through "table data import wizard"

# -- Queries for the tables

* All the tables SELECT \* FROM category;



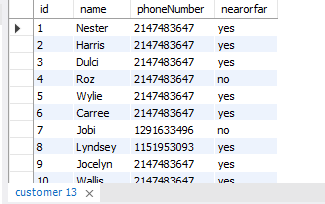
SELECT \* FROM brand;



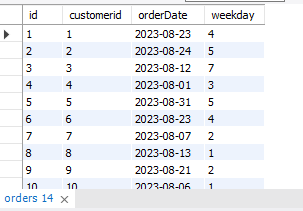
SELECT \* FROM icecream;



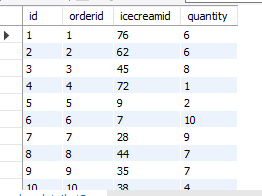
SELECT \* FROM customer;



SELECT \* FROM orders;



SELECT \* FROM ordersdetails;



# -- Sales overview

* total number of orders throughout the month SELECT SUM(quantity) AS totalQuantity FROM ordersdetails;



* total sales throughout the month SELECT SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN icecream ic ON ic.id = od.icecreamid;



-- Remark:

-- in the given month the shop has acquired the total sales of Rs. 75,952 while selling total 1392 products of the ice creams

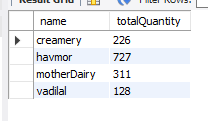
# -- Brand wise Sales overview

**-- quantity**

SELECT b.name, SUM(od.quantity) AS totalQuantity FROM ordersdetails od

JOIN icecream ic ON ic.id = od.icecreamid JOIN brand b ON b.id = ic.brandid

GROUP BY b.id;

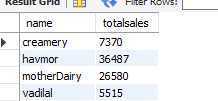


# -- Sales

SELECT b.name, SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN icecream ic ON ic.id = od.icecreamid JOIN brand b ON b.id = ic.brandid

GROUP BY b.id;



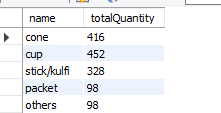
# -- Category wise Sales overview

**-- quantity**

SELECT c.name, SUM(od.quantity) AS totalQuantity FROM ordersdetails od

JOIN icecream ic ON ic.id = od.icecreamid JOIN category c ON c.id = ic.categoryid

GROUP BY c.id;

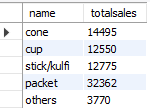


# -- sales

SELECT c.name, SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN icecream ic ON ic.id = od.icecreamid JOIN category c ON c.id = ic.categoryid

GROUP BY c.id;



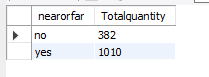
# -- Locality wise Sales overview (near or far)

**-- quantity**

SELECT c.nearorfar, SUM(od.quantity) AS Totalquantity FROM ordersdetails od

JOIN orders o ON o.id = od.orderid

JOIN customer c ON c.id = o.customerid GROUP BY nearorfar;



# -- sales

SELECT c.nearorfar, SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN orders o ON o.id = od.orderid

JOIN customer c ON c.id = o.customerid

JOIN icecream ic ON ic.id = od.icecreamid GROUP BY c.nearorfar;



-- havmor has the most number of the sales to figure out the best category in it we have

SELECT b.name, c.name, SUM(od.quantity) AS totalorders, SUM(od.quantity\*ic.price) AS Sales

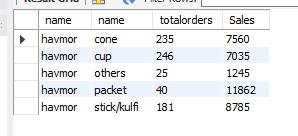
FROM brand b

JOIN icecream ic ON ic.brandid = b.id

JOIN category c ON c.id = ic.categoryid

JOIN ordersdetails od ON od.icecreamid = ic.id WHERE b.name = "havmor"

GROUP BY c.name;



**-- sales based on the ice cream products** SELECT ic.name, SUM(ic.price\*od.quantity) AS sales FROM icecream ic

JOIN ordersdetails od ON od.icecreamid = ic.id GROUP BY ic.name

ORDER BY sales DESC;



**-- quantity based on the ice cream products** SELECT b.name, ic.name, SUM(od.quantity) AS sales FROM icecream ic

JOIN ordersdetails od ON od.icecreamid = ic.id JOIN brand b ON b.id = ic.brandid

GROUP BY ic.name

ORDER BY sales DESC;



# -- sales and quantity based on the weekday

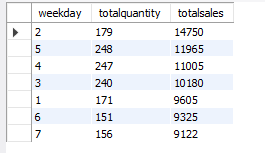
SELECT o.weekday, SUM(od.quantity) AS totalquantity, SUM(od.quantity\*ic.price) AS totalsales

FROM ordersdetails od

JOIN orders o ON od.orderid = o.id

JOIN icecream ic ON ic.id = od.icecreamid GROUP BY weekday

ORDER BY totalsales DESC;



# -- BEST customers

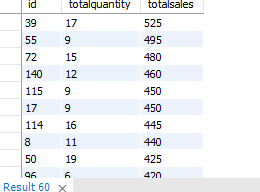
SELECT cm.id, SUM(od.quantity) AS totalquantity, SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN orders o ON od.orderid = o.id

JOIN icecream ic ON ic.id = od.icecreamid

JOIN customer cm ON cm.id = o.customerid GROUP BY cm.id

ORDER BY totalsales DESC;



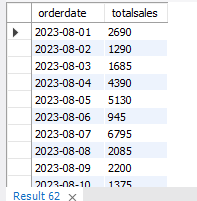
# -- Sales in the date range

SELECT o.orderdate, SUM(od.quantity\*ic.price) AS totalsales FROM ordersdetails od

JOIN orders o ON o.id = od.orderid

JOIN icecream ic ON ic.id = od.icecreamid WHERE o.orderdate BETWEEN '2023-08-01' AND '2023-08-10'

GROUP BY orderdate ORDER BY orderdate;



# CONCLUSION:

In this case study, we explored the design and implementation of a relational database using SQL to address the needs of icecream shop. Through a structured approach, we identified key entities, defined their relationships, and ensured data integrity through appropriate constraints and normalization.