**Lab Number: 2**

**Title:** Designing various Data Warehouse Schema (Star Schema, Snowflake Schema, Fact Constellation) using MySQL Query.

**Objective:** To get familiarized with the schemas used for creating a data warehouse.

**IDE/Tools used:** MySQL, Apache server.

**Query Programming language: SQL**

**Theory:**

**Data warehouse:** A data warehouse is like a central hub for all an organization’s data. It collects information from various sources, organizes it, and makes it easier to analyze. Unlike systems used for daily operations, a data warehouse focuses on helping businesses understand their data, identify patterns, and make informed decisions.

**Schema in Data warehouse:** A schema acts as a blueprint for organizing data in a data warehouse. It outlines the structure, including what tables are used, how they relate, and what kind of data each contains. A well-designed schema makes it simple to find and analyze data.

**Key Parts of a Data Warehouse Schema**

* **Fact Tables**: These store the numerical data or metrics you want to analyze, like sales amounts or quantities sold.
* **Dimension Tables**: These provide additional details about the numbers, such as "who," "what," "when," or "where" (e.g., customer, product, date, or location).
* **Attributes**: These are the specific pieces of information in the dimension tables, such as product names, regions, or time periods.
* **Keys**:
* **Primary Keys**: Unique identifiers in dimension tables, like a customer ID.
* **Foreign Keys**: Links that connect dimension tables to the fact table.
* **Metadata**: This serves as a guide, explaining the purpose and relationships of the data in the warehouse.

**Types of Data Warehouse Schemas**

* **Star Schema:** The star schema is the simplest design, featuring a central fact table with key metrics (e.g., sales) surrounded by dimension tables (e.g., product, customer, time). The straightforward layout makes it easy to query and ideal for fast analysis.
* **Snowflake Schema:** The snowflake schema extends the star schema by normalizing dimension tables into smaller sub-tables (e.g., splitting "product" into "product" and "category"). This reduces redundancy but increases complexity and can slow down queries.
* **Fact Constellation (Galaxy Schema):** The galaxy schema involves multiple fact tables sharing dimension tables, enabling analysis of related datasets (e.g., sales and inventory). While it supports complex queries, it requires careful design and management.

**Implementation:**

**Q.1) Star Schema:**

**Create a database**

CREATE DATABASE AUTOXYZ;

**Using SQL Create a source table for data warehouse.**

DROP TABLE IF EXISTS company;

CREATE TABLE company (id int AUTO\_INCREMENT PRIMARY KEY,

item\_name varchar(255),

brand varchar(255),

sold\_by varchar(255),

category varchar(255),

day int,

month varchar(255),

quarter varchar(255),

years int,

location\_name varchar(255),

state varchar(255),

pin\_code int,

branch\_name varchar(255),

branch\_manager varchar(255),

qty\_sold int,

amt\_sold int

);

INSERT INTO company(item\_name,

brand,

sold\_by,

category,

day,

month,

quarter,

years,

location\_name,

state,

pin\_code,

branch\_name,

branch\_manager,

qty\_sold,

amt\_sold)

VALUES("Car","Model X","Tesla","Four wheeler",13,"June","Q2",2021,"New Baneshwor","Bagmati","123","Baneshwor 1","Ashish",2,15000),

("Car","Model Y","Tesla","Four wheeler",15,"October","Q4",2022,"Old Baneshwor","Bagmati","123","Baneshwor 3","Manoj",1,5000);

SELECT \* FROM company;

**Using SQL Create time dimension table for data warehouse.**

DROP TABLE IF EXISTS timedim;

CREATE TABLE timedim (t\_id int AUTO\_INCREMENT PRIMARY KEY,

day int,

month varchar(255),

quarter varchar(255),

years int

);

INSERT INTO timedim(

day,

month,

quarter,

years

)

SELECT day,month,quarter,years FROM company;

SELECT \* FROM timedim;

**Using SQL Create item dimension table for data warehouse.**

DROP TABLE IF EXISTS itemdim;

CREATE TABLE itemdim (i\_id int AUTO\_INCREMENT PRIMARY KEY,

item\_name varchar(255),

brand varchar(255),

sold\_by varchar(255),

category varchar(255)

);

INSERT INTO itemdim(

item\_name,

brand,

sold\_by,

category

)

SELECT item\_name,brand,sold\_by,category FROM company;

SELECT \* FROM itemdim;

**Using SQL Create location dimension table for data warehouse.**

DROP TABLE IF EXISTS locationdim;

CREATE TABLE locationdim (l\_id int AUTO\_INCREMENT PRIMARY KEY,

location\_name varchar(255),

state varchar(255),

pin\_code int

);

INSERT INTO locationdim(

location\_name,

state,

pin\_code

)

SELECT location\_name,state,pin\_code FROM company;

SELECT \* FROM locationdim;

**Using SQL Create branch dimension table for data warehouse.**

DROP TABLE IF EXISTS branchdim;

CREATE TABLE branchdim (b\_id int AUTO\_INCREMENT PRIMARY KEY,

branch\_name varchar(255),

branch\_manager varchar(255)

);

INSERT INTO branchdim(

branch\_name,

branch\_manager

)

SELECT branch\_name,branch\_manager FROM company;

SELECT \* FROM branchdim;

**Using SQL Create sales fact table for data warehouse using Foreign Key.**

DROP TABLE IF EXISTS salesFact;

CREATE TABLE salesFact (t\_id int,

i\_id int,

l\_id int,

b\_id int,

qty\_sold int,

amt\_sold int,

FOREIGN Key(t\_id) REFERENCES timedim(t\_id),

FOREIGN Key(i\_id) REFERENCES itemdim(i\_id),

FOREIGN Key(l\_id) REFERENCES locationdim(l\_id),

FOREIGN Key(b\_id) REFERENCES branchdim(b\_id));

INSERT INTO salesFact(t\_id,

i\_id,

l\_id,

b\_id,

qty\_sold,

amt\_sold)

SELECT t\_id,i\_id,l\_id,b\_id,qty\_sold,amt\_sold FROM company c

LEFT OUTER JOIN timedim t ON t.day = c.day AND t.month = c.month AND t.quarter = c.quarter AND t.years = c.years

LEFT OUTER JOIN itemdim i ON i.item\_name = c.item\_name AND i.brand = c.brand AND i.sold\_by = c.sold\_by AND i.category = c.category

LEFT OUTER JOIN locationdim l ON l.location\_name = c.location\_name AND l.state = c.state AND l.pin\_code = c.pin\_code

LEFT OUTER JOIN branchdim b ON b.branch\_name = c.branch\_name AND b.branch\_manager = c.branch\_manager;

SELECT \* FROM salesFact;

**Query to select the records where years = 2022**

SELECT \* FROM salesFact s LEFT OUTER JOIN timedim t ON t.t\_id = s.t\_id WHERE years=2022;

**Q.2) Snowflake Schema :**

**Create a database named "snowcompany".**

CREATE DATABASE snowcompany;

**Using SQL Create a source table for data warehouse.**

DROP TABLE IF EXISTS company;

CREATE TABLE company (id int AUTO\_INCREMENT PRIMARY KEY,

item\_name varchar(255),

brand varchar(255),

sold\_by varchar(255),

category varchar(255),

day int,

month varchar(255),

quarter varchar(255),

years int,

location\_name varchar(255),

state varchar(255),

pin\_code int,

branch\_name varchar(255),

branch\_manager varchar(255),

department\_name varchar(255),

department\_code int,

supplier\_name varchar(255),

supplier\_address varchar(255),

supplier\_type varchar(255),

qty\_sold int,

amt\_sold int

);

INSERT INTO company(item\_name,

brand,

sold\_by,

category,

day,

month,

quarter,

years,

location\_name,

state,

pin\_code,

branch\_name,

branch\_manager,

department\_name,

department\_code,

supplier\_name,

supplier\_address,

supplier\_type,

qty\_sold,

amt\_sold)

VALUES("Car","Model X","Tesla","Four wheeler",13,"June","Q2",2021,"New Baneshwor","Bagmati","123","Baneshwor 1","Ashish","sales",013,"C&C Auto","Koteshwor","Auto Four Wheeler",2,15000),

("Car","Model Y","Tesla","Four wheeler",15,"October","Q4",2022,"Old Baneshwor","Bagmati","123","Baneshwor 3","Manoj","finance",420,"T&T Auto","Tripureshwor","Auto",1,5000);

SELECT \* FROM company;

**Using SQL Create time dimension table for data warehouse.**

DROP TABLE IF EXISTS timedim;

CREATE TABLE timedim (t\_id int AUTO\_INCREMENT PRIMARY KEY,

day int,

month varchar(255),

quarter varchar(255),

years int

);

INSERT INTO timedim(

day,

month,

quarter,

years

)

SELECT day,month,quarter,years FROM company;

SELECT \* FROM timedim;

**Using SQL Create location dimension table for data warehouse.**

DROP TABLE IF EXISTS locationdim;

CREATE TABLE locationdim (l\_id int AUTO\_INCREMENT PRIMARY KEY,

location\_name varchar(255),

state varchar(255),

pin\_code int

);

INSERT INTO locationdim(

location\_name,

state,

pin\_code

)

SELECT location\_name,state,pin\_code FROM company;

SELECT \* FROM locationdim;

**Using SQL Create department dimension table for data warehouse.**

DROP TABLE IF EXISTS departmentdim;

CREATE TABLE departmentdim(dept\_id int AUTO\_INCREMENT PRIMARY KEY,

dept\_name varchar(255),

dept\_code int);

INSERT INTO departmentdim(dept\_name,dept\_code)

SELECT department\_name,department\_code FROM company;

SELECT \* FROM departmentdim;

**Using SQL Create branch dimension table for data warehouse.**

DROP TABLE IF EXISTS branchdim;

CREATE TABLE branchdim (b\_id int AUTO\_INCREMENT PRIMARY KEY,

branch\_name varchar(255),

branch\_manager varchar(255),

depart\_id int,

FOREIGN KEY(depart\_id) REFERENCES departmentdim(dept\_id)

);

INSERT INTO branchdim(

branch\_name,

branch\_manager,

depart\_id

)

SELECT branch\_name,branch\_manager,dept\_id FROM company c

JOIN departmentdim d ON c.department\_name = d.dept\_name AND

c.department\_code = d.dept\_code;

SELECT \* FROM branchdim;

**Using SQL Create Supplier dimension table for data warehouse.**

DROP TABLE IF EXISTS supplierdim;

CREATE TABLE supplierdim(supp\_id int AUTO\_INCREMENT PRIMARY KEY,

supp\_name varchar(255),

supp\_address varchar(255),

supp\_type varchar(255));

INSERT INTO supplierdim(supp\_name,supp\_address,supp\_type)

SELECT supplier\_name,supplier\_address,supplier\_type FROM company;

SELECT \* FROM supplierdim;

**Using SQL Create item dimension table for data warehouse.**

DROP TABLE IF EXISTS itemdim;

CREATE TABLE itemdim (i\_id int AUTO\_INCREMENT PRIMARY KEY,

item\_name varchar(255),

brand varchar(255),

sold\_by varchar(255),

category varchar(255),

supplier\_id int,

FOREIGN KEY(supplier\_id) REFERENCES supplierdim(supp\_id)

);

INSERT INTO itemdim(

item\_name,

brand,

sold\_by,

category,

supplier\_id

)

SELECT item\_name,brand,sold\_by,category,supp\_id FROM company c

LEFT OUTER JOIN supplierdim s ON c.supplier\_name = s.supp\_name AND

c.supplier\_address = s.supp\_address AND

c.supplier\_type = s.supp\_type;

SELECT \* FROM itemdim;

**Using SQL Create sales fact table for data warehouse using Foreign Key.**

DROP TABLE IF EXISTS salesFact;

CREATE TABLE salesFact (t\_id int,

i\_id int,

l\_id int,

b\_id int,

qty\_sold int,

amt\_sold int,

FOREIGN Key(t\_id) REFERENCES timedim(t\_id),

FOREIGN Key(i\_id) REFERENCES itemdim(i\_id),

FOREIGN Key(l\_id) REFERENCES locationdim(l\_id),

FOREIGN Key(b\_id) REFERENCES branchdim(b\_id));

INSERT INTO salesFact(t\_id,

i\_id,

l\_id,

b\_id,

qty\_sold,

amt\_sold)

SELECT t\_id,i\_id,l\_id,b\_id,qty\_sold,amt\_sold FROM company c

LEFT OUTER JOIN timedim t ON t.day = c.day AND t.month = c.month AND t.quarter = c.quarter AND t.years = c.years

LEFT OUTER JOIN itemdim i ON i.item\_name = c.item\_name AND i.brand = c.brand AND i.sold\_by = c.sold\_by AND i.category = c.category

LEFT OUTER JOIN locationdim l ON l.location\_name = c.location\_name AND l.state = c.state AND l.pin\_code = c.pin\_code

LEFT OUTER JOIN branchdim b ON b.branch\_name = c.branch\_name AND b.branch\_manager = c.branch\_manager;

SELECT \* FROM salesFact;

**Query to show the records of sales fact table with the department dimension table**

SELECT \* FROM salesFact s LEFT OUTER JOIN branchdim b ON b.b\_id = s.b\_id

LEFT OUTER JOIN departmentdim d ON b.depart\_id = d.dept\_id;

**Q.3) Fact Constellation:**

-- Create a new database named "nikitafact"

**CREATE DATABASE nikitafact;**

-- Use the database "nikitafact"

USE nikitafact;

-- Create source table for the data warehouse

**CREATE TABLE nikita\_sales\_data (**

id INT AUTO\_INCREMENT PRIMARY KEY,

item\_name VARCHAR(255),

brand VARCHAR(255),

sold\_by VARCHAR(255),

category VARCHAR(255),

day INT,

month VARCHAR(255),

quarter VARCHAR(255),

year INT,

location\_name VARCHAR(255),

state VARCHAR(255),

pin\_code INT,

driver\_name VARCHAR(255),

duty\_time VARCHAR(255),

qty\_sold INT,

amt\_sold INT

);

-- Insert data into the source table

INSERT INTO nikita\_sales\_data(

item\_name, brand, sold\_by, category, day, month, quarter, year,

location\_name, state, pin\_code, driver\_name, duty\_time, qty\_sold, amt\_sold

)

VALUES

('Car', 'Model X', 'Tesla', 'Four wheeler', 13, 'June', 'Q2', 2024, 'New Baneshwor', 'Bagmati', 123, 'Jack', 'Evening', 2, 13000),

('Car', 'Model Y', 'Tesla', 'Four wheeler', 21, 'November', 'Q4', 2024, 'Koteshwor', 'Bagmati', 123, 'Candace', 'Morning', 1, 8000);

-- Select all records from the source table

SELECT \* FROM nikita\_sales\_data;

-- Create time dimension table for the data warehouse

CREATE TABLE timedim (

t\_id INT AUTO\_INCREMENT PRIMARY KEY,

day INT,

month VARCHAR(255),

quarter VARCHAR(255),

year INT

);

-- Insert values into the time dimension table

INSERT INTO timedim(day, month, quarter, year)

SELECT day, month, quarter, year FROM nikita\_sales\_data;

-- Select all records from the time dimension table

SELECT \* FROM timedim;

-- Create location dimension table for the data warehouse

**CREATE TABLE locationdim (**

l\_id INT AUTO\_INCREMENT PRIMARY KEY,

location\_name VARCHAR(255),

state VARCHAR(255),

pin\_code INT

);

-- Insert values into the location dimension table

INSERT INTO locationdim(location\_name, state, pin\_code)

SELECT location\_name, state, pin\_code FROM nikita\_sales\_data;

-- Select all records from location dimension table

SELECT \* FROM locationdim;

-- Create item dimension table for the data warehouse

**CREATE TABLE itemdim (**

i\_id INT AUTO\_INCREMENT PRIMARY KEY,

item\_name VARCHAR(255),

brand VARCHAR(255),

sold\_by VARCHAR(255),

category VARCHAR(255)

);

-- Insert values into item dimension table

INSERT INTO itemdim(item\_name, brand, sold\_by, category)

SELECT item\_name, brand, sold\_by, category FROM nikita\_sales\_data;

-- Select all records from item dimension table

SELECT \* FROM itemdim;

-- Create vehicle dimension table for the data warehouse

CREATE TABLE vehicledim (

vehicle\_id INT AUTO\_INCREMENT PRIMARY KEY,

driver\_name VARCHAR(255),

duty\_time VARCHAR(255)

);

-- Insert values into vehicle dimension table

INSERT INTO vehicledim(driver\_name, duty\_time)

SELECT driver\_name, duty\_time FROM nikita\_sales\_data;

-- Select all records from vehicle dimension table

SELECT \* FROM vehicledim;

-- Create sales fact table for the data warehouse

**CREATE TABLE salesfact (**

item\_id INT,

time\_id INT,

location\_id INT,

qty\_sold INT,

FOREIGN KEY (item\_id) REFERENCES itemdim(i\_id),

FOREIGN KEY (time\_id) REFERENCES timedim(t\_id),

FOREIGN KEY (location\_id) REFERENCES locationdim(l\_id)

);

-- Insert values into sales fact table

INSERT INTO salesfact(item\_id, time\_id, location\_id, qty\_sold)

SELECT i\_id, t\_id, l\_id, qty\_sold

FROM nikita\_sales\_data c

LEFT JOIN timedim t ON t.day = c.day AND t.month = c.month AND t.quarter = c.quarter AND t.year = c.year

LEFT JOIN itemdim i ON i.item\_name = c.item\_name AND i.brand = c.brand AND i.sold\_by = c.sold\_by AND i.category = c.category

LEFT JOIN locationdim l ON l.location\_name = c.location\_name AND l.state = c.state AND l.pin\_code = c.pin\_code;

-- Select all records from sales fact table

SELECT \* FROM salesfact;

-- Create delivery fact table for the data warehouse

**CREATE TABLE deliveryfact (**

item\_id INT,

location\_id INT,

vehicle\_id INT,

FOREIGN KEY (item\_id) REFERENCES itemdim(i\_id),

FOREIGN KEY (location\_id) REFERENCES locationdim(l\_id),

FOREIGN KEY (vehicle\_id) REFERENCES vehicledim(vehicle\_id)

);

-- Insert values into delivery fact table

INSERT INTO deliveryfact(item\_id, location\_id, vehicle\_id)

SELECT i\_id, l\_id, vehicle\_id

FROM nikita\_sales\_data c

LEFT JOIN itemdim i ON i.item\_name = c.item\_name AND i.brand = c.brand AND i.sold\_by = c.sold\_by AND i.category = c.category

LEFT JOIN locationdim l ON l.location\_name = c.location\_name AND l.state = c.state AND l.pin\_code = c.pin\_code

LEFT JOIN vehicledim v ON v.driver\_name = c.driver\_name AND v.duty\_time = c.duty\_time;

-- Select all records from delivery fact table

SELECT \* FROM deliveryfact;

-- Query to select item\_id, location\_id, vehicle\_id, driver\_name, duty\_time records where duty time = "Morning"

SELECT d.item\_id, d.location\_id, v.vehicle\_id, v.driver\_name, v.duty\_time

FROM deliveryfact d

LEFT JOIN vehicledim v ON d.vehicle\_id = v.vehicle\_id

WHERE v.duty\_time = 'Morning';**Screenshots of generated Data Warehouse Schema:**

**Discussion:**

The data warehouse organizes information using schemas. These schemas, which rely on fact and dimension tables, fall into three main types. Among them, the star schema is the simplest, featuring multiple dimension tables linked to a single fact table. The snowflake schema, on the other hand, organizes dimension tables in a normalized format. Lastly, the third type involves multiple fact tables. Each schema helps manage and filter data effectively, making them essential for structuring ecosystems and enhancing query performance.

**Conclusion:**

In this lab, we explored different data warehouse schemas by implementing them on the MySQL platform, hosted locally on an Apache server, and using SQL for querying. We also observed that each schema design has unique strengths, tailored to meet specific application requirements and resource constraints.