Name: Sumeet S Kedare

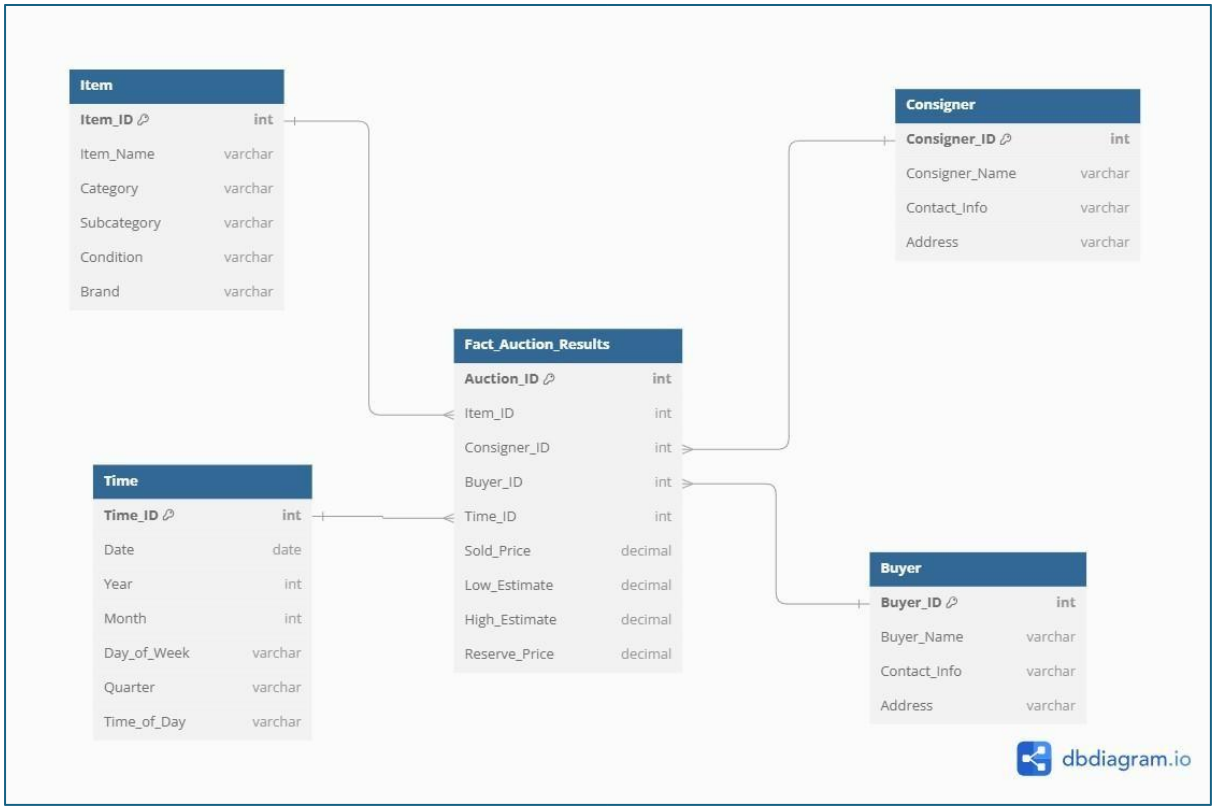
Roll No: 31 **Class-Batch:** TY3-B

DWM Experiment No. 1

Aim: Design Star and Snowflake Schema for the given objective.

Problem Statement: An Auction company wants to design data warehouse to record sold price of items with their low estimate, high estimate and reserve price. There are four dimensions - Item, Consigner, Buyer and Time. Explain all aspects of the diagram. Design Star and Snowflake schema for above case.

Star Schema:



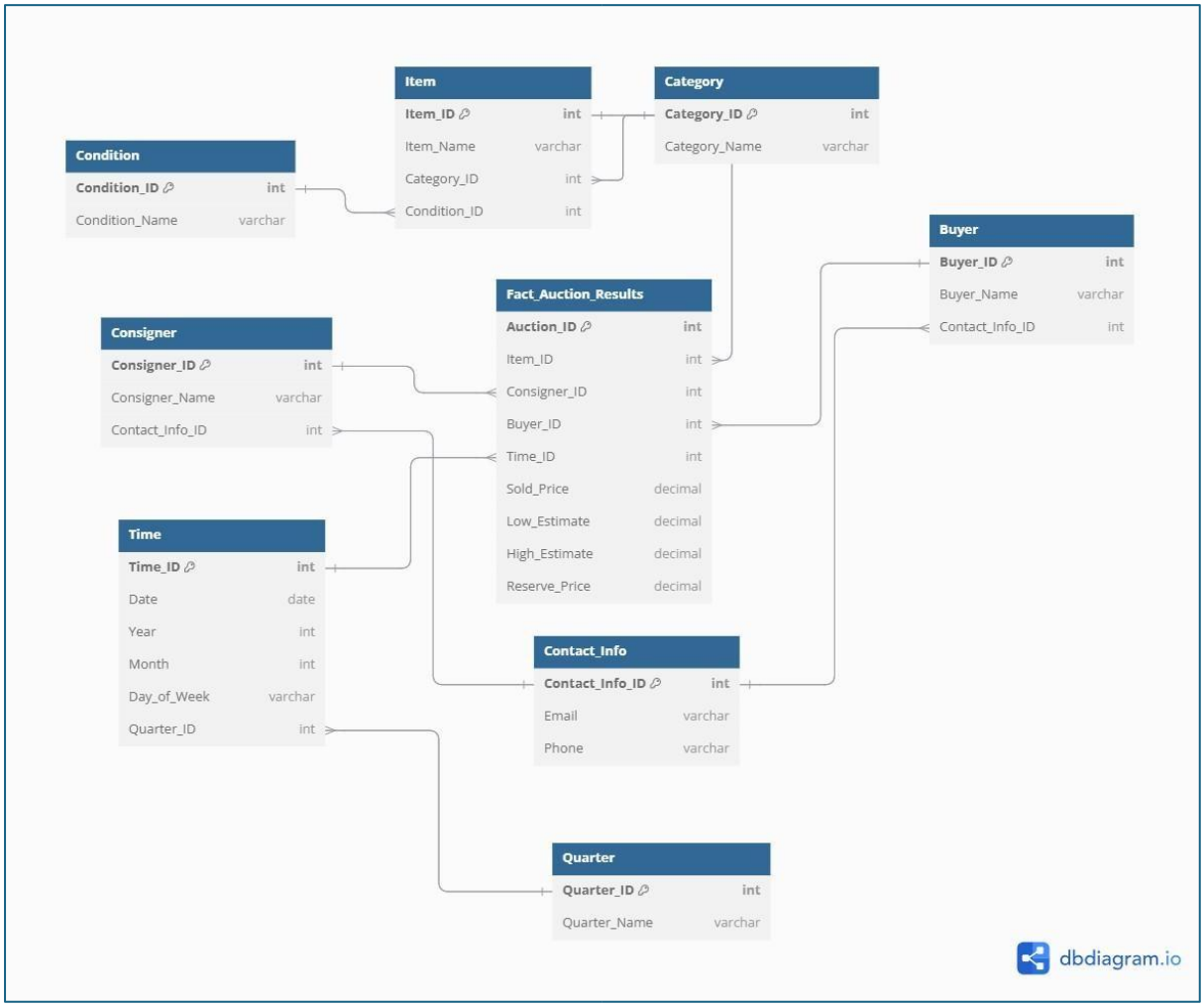
SQL Query:

```
CREATE TABLE Item (
    Item_ID INT PRIMARY KEY,
    Item_Name VARCHAR(255),
```

```
    Category VARCHAR(255),
    Subcategory VARCHAR(255),
    Condition VARCHAR(255),
    Brand VARCHAR(255)
);
CREATE TABLE Consigner (
    Consigner_ID INT PRIMARY KEY,
    Consigner_Name VARCHAR(255),
    Contact_Info VARCHAR(255),
    Address VARCHAR(255)
);
CREATE TABLE Buyer (
    Buyer_ID INT PRIMARY KEY,
    Buyer_Name VARCHAR(255),
    Contact_Info VARCHAR(255),
    Address VARCHAR(255)
);
CREATE TABLE Time (
    Time_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day_of_Week VARCHAR(50),
    Quarter VARCHAR(50),
    Time_of_Day VARCHAR(50)
);
CREATE TABLE Fact_Auction_Results (
    Auction_ID INT PRIMARY KEY,
    Item_ID INT,
    Consigner_ID INT,
    Buyer_ID INT,
    Time_ID INT,
    Sold_Price DECIMAL(10,2),
    Low_Estimate DECIMAL(10,2),
    High_Estimate DECIMAL(10,2),
    Reserve_Price DECIMAL(10,2),
    FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID),
    FOREIGN KEY (Consigner_ID) REFERENCES Consigner(Consigner_ID),
    FOREIGN KEY (Buyer_ID) REFERENCES Buyer(Buyer_ID),
```

FOREIGN KEY (Time_ID) REFERENCES Time(Time_ID)
);

Snowflake Schema:



SQL Query:

```
CREATE TABLE Condition (
    Condition_ID INT PRIMARY KEY,
Condition_Name VARCHAR(255)
);
CREATE TABLE Category (
    Category_ID INT PRIMARY KEY,
Category_Name VARCHAR(255)
);
CREATE TABLE Item (
    Item_ID INT PRIMARY KEY,
    Item_Name VARCHAR(255),
    Category_ID INT,
```

```
    Condition_ID INT,
    FOREIGN KEY (Category_ID) REFERENCES Category(Category_ID),
    FOREIGN KEY (Condition_ID) REFERENCES Condition(Condition_ID)
);

CREATE TABLE Contact_Info (
    Contact_Info_ID INT PRIMARY KEY,
    Email VARCHAR(255),
    Phone VARCHAR(50)
);
CREATE TABLE Consigner (
    Consigner_ID INT PRIMARY KEY,
    Consigner_Name VARCHAR(255),
    Contact_Info_ID INT,
    FOREIGN KEY (Contact_Info_ID) REFERENCES Contact_Info(Contact_Info_ID)
);
CREATE TABLE Buyer (
    Buyer_ID INT PRIMARY KEY,
    Buyer_Name VARCHAR(255),
    Contact_Info_ID INT,
    FOREIGN KEY (Contact_Info_ID) REFERENCES Contact_Info(Contact_Info_ID)
);
CREATE TABLE Quarter (
    Quarter_ID INT PRIMARY KEY,
    Quarter_Name VARCHAR(50)
);
CREATE TABLE Time (
    Time_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day_of_Week VARCHAR(50),
    Quarter_ID INT,
    FOREIGN KEY (Quarter_ID) REFERENCES Quarter(Quarter_ID)
);
CREATE TABLE Fact_Auction_Results (
    Auction_ID INT PRIMARY KEY,
    Item_ID INT,
    Consigner_ID INT,
    Buyer_ID INT,
    Time_ID INT,
```

```
    Sold_Price DECIMAL(10,2),
    Low_Estimate DECIMAL(10,2),
    High_Estimate DECIMAL(10,2),
    Reserve_Price DECIMAL(10,2),
    FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID),
    FOREIGN KEY (Consigner_ID) REFERENCES Consigner(Consigner_ID),
    FOREIGN KEY (Buyer_ID) REFERENCES Buyer(Buyer_ID),
    FOREIGN KEY (Time_ID) REFERENCES Time(Time_ID)
);
```

Review Questions:

Q1. In a star schema, how is the fact table typically related to the dimension tables?

Ans: In a star schema, the fact table is centrally located and is directly related to multiple dimension tables through foreign key relationships. Each record in the fact table contains measurable business data (such as sales, prices, or counts) and includes foreign keys that reference primary keys in the surrounding dimension tables. The dimension tables store descriptive attributes (such as product details, time periods, customer information, or geographical data) that provide context to the numerical values in the fact table. This structure enables efficient querying and reporting, as it allows for fast joins between the fact table and dimension tables while maintaining a straightforward and highly denormalized design.

Q2. What is the main difference between a star schema and a snowflake schema?

Ans: The main difference between a star schema and a snowflake schema lies in the level of normalization of the dimension tables. In a star schema, dimension tables are denormalized, meaning they store all descriptive attributes in a single table without further breaking them down into sub-tables. This results in a simpler structure with fewer joins, making queries faster and easier to understand. In contrast, a snowflake schema normalizes the dimension tables by dividing them into multiple related tables to eliminate redundancy. While this reduces data storage requirements and improves data integrity, it increases query complexity due to the additional joins needed to retrieve data. Essentially, a star schema prioritizes query performance and simplicity, while a snowflake schema focuses on reducing redundancy and improving data integrity.