**Practice Mini Case Study for Schema Integration**

Kelsey Thompson

Colorado State University Global

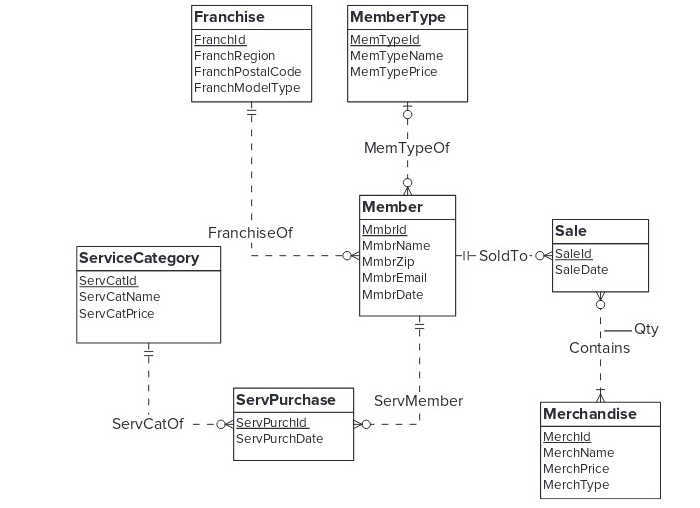
MIS407: Database Concepts

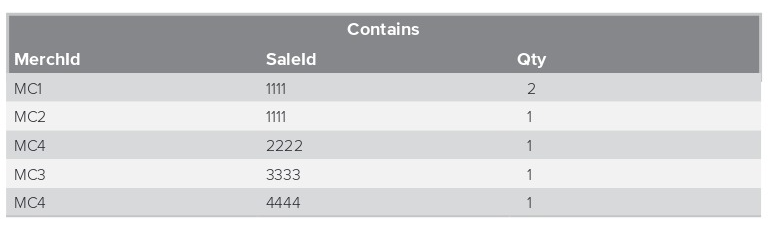
Dr. Mario Missakian

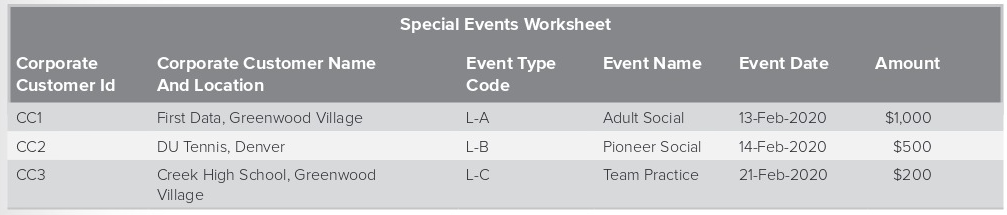
May 30, 2024

**Practice Mini Case Study for Schema Integration**

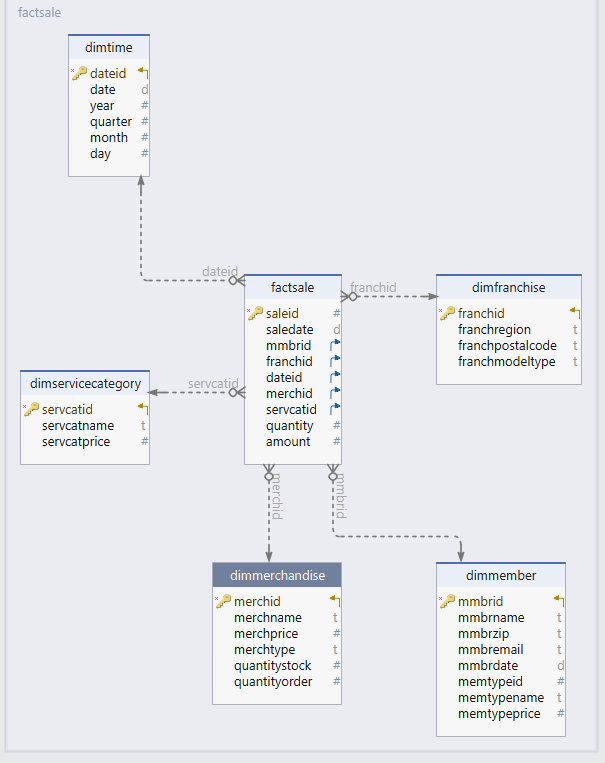
For noted references, here is the ORIGINAL database for Fitness Unlimited. This organization stands out as a prominent provider of fitness centers, offering a diverse range of fitness programs and membership choices. To effectively manage its operations, Fitness Unlimited maintains a comprehensive retail database to monitor the sales of services and merchandise. In this assignment, I will be customizing the database design to suit the specific requirements of Fitness Unlimited's business model.







Here is my revised ERD, followed by the assignment deliverables:



1. Identify dimensions, map dimensions to data sources, and specify dimension hierarchies. For each dimension, identify its data sources and attributes in each data source. For hierarchical dimensions, indicate the levels from broad to narrow.
2. The dimensions signify the WHAT, WHO, WHEN, WHERE. The map dimensions to data sources are described by listing the entities and attributes. These are listed by specifying the hierarchies of the dimension.

1. The WHAT is represented by the Products, MemberType and the Services. The tables are ServiceCategory and Merchandise. The ServiceCategory contains narrow to broad levels labeled as the ServCatld, ServCatName and the ServCatPrice columns. The Merchandise entity has the following attributes, in narrow to broad levels: Merchid, MerchName. MerchPrice and MerchType.

2. The WHO are the gym members represented in the Member table. The attributes are as follows and are mostly narrow levels specific to each customer (excluding the zip and date): MemberID, MemberName, MemberZip, MemberEmail and MemberDate.

3. The WHEN represents the sale of the merchandise/services dates. The WHEN for the products is found in the Sales Table with narrow to broad levels of attributes which are labeled SaleID and SaleDate.

Secondly, the WHEN for the service dates is represented by the ServPerch Table with the following (there is no hierarchy): SerPurchID and SerPuchDate.

4) WHERE can be described within the Franchise table with a clear hierarchy of broad to narrow. The attributes are listed as FranchiseID, FranchRegion, FranchPostalCode and FranchModelType.

2. Specify measures, related data sources, and measure aggregation properties.

To specify measures, there are dollar amounts, class/type names, zip codes or names.

* However, more specifically, the WHAT is measured among the products (sales), services (service purchase table), events (worksheet) and columns in the Sales and Contains tables, as well as the Special Events Worksheet. These can be aggregated into a single table with the relational data, regarding sales. I will merge the following: SaleID, SalesDate, MerchID, ServicePurchID, MemberID
* For the WHO, the gym members are represented in the Member table and by the CustomerCopID in the Special Events Worksheet. These columns can be aggregated:  MemberID, MemberName, TypeID, EventType and the FranchiseID
* For the WHEN, the merchandise/services dates are represented by the salesID and Date (month, day, year), SevicePurchaseDate, EventDate of the merchandise/services dates. These are found in the Sales Table, ServPurchase and the Special Events Worksheet.
* For the WHERE, the Franchise table contains the FranchiseID, FranchRegion, FranchPostalCode and FranchModelType and the Sales table contains the SalesID. These will be related to the SalesID for aggregation. In the Sales Table, I will utilize the COUNT function.

1. Identify the grain in your dimensional design using the business needs as a guideline. Compute sparsity if the fact table size is given.

* The sparsity is mentioned in the previous, answered question.
* The business needs (WHAT, WHO, WHEN, WHERE) are as follows: The data warehouse is designed to analyze merchandise sales and service purchases. Merchandise sales are computed based on quantity and selling price, tracked by franchise, type, and customer over time. For services, only the purchase price is recorded. Customer details include zip code, membership date, and member type. The sales office requires flexibility for data mining analysis (individual customer, product, franchise, and date) and typical reporting (customer location, franchise location, product or service type, and week).
* Based on these business needs, the grain should specify each individual row and what that row represents. For example, in the Sales table, each row will represent a transaction corresponding to the purchase of a product or service on a specific date. The primary key is SaleID (M-1). For the WHEN, each row will represent a date ID linked to multiple purchases on the same date. For the WHO, each row will represent a customer (MemberID), which will be an M-1 relationship since each row represents one unique customer. The same applies to the Franchise, where each row will represent one unique location (FranchID).

1. Extend an analysis to design a star schema (or variation) to support inventory analysis. For each table, define the table name, primary key, and columns. Apply design transformations, especially the flatten and merge transformations where appropriate.

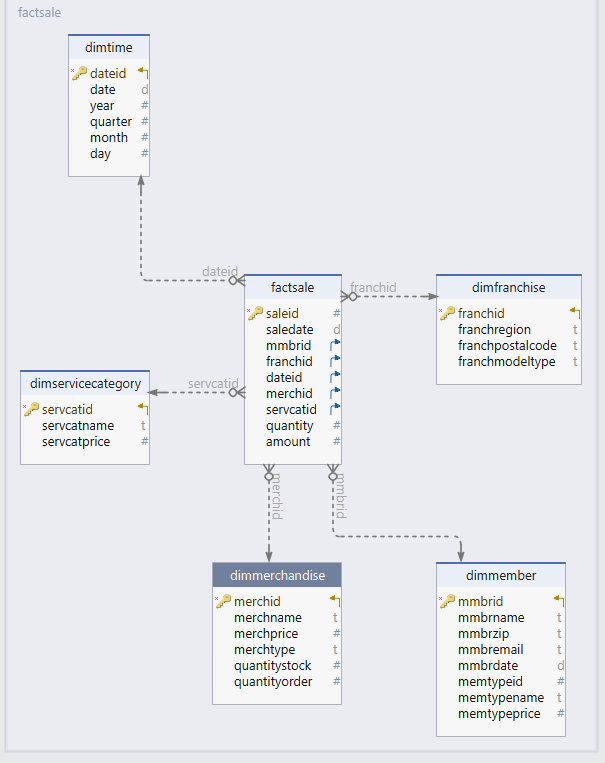
**Dimension Tables:**

1. **DimMember:**
   * MmbrId (Primary Key)
   * MmbrName
   * MmbrZip
   * MmbrEmail
   * MmbrDate
   * MemTypeId
   * MemTypeName
   * MemTypePrice
2. **DimFranchise:**
   * FranchId (Primary Key)
   * FranchRegion
   * FranchPostalCode
   * FranchModelType
3. **DimTime:**
   * DateId (Primary Key)
   * Date
   * Year
   * Quarter
   * Month
   * Day
4. **DimMerchandise:**
   * MerchId (Primary Key)
   * MerchName
   * MerchPrice
   * MerchType
   * QuantityStock
   * QuantityOrder
5. **DimServiceCategory:**
   * ServCatId (Primary Key)
   * ServCatName
   * ServCatPrice

**Fact Table:**

* **FactSale:**
  + SaleId (Primary Key)
  + SaleDate
  + MmbrId (Foreign Key referencing DimMember)
  + FranchId (Foreign Key referencing DimFranchise)
  + DateId (Foreign Key referencing DimTime)
  + MerchId (Foreign Key referencing DimMerchandise)
  + ServCatId (Foreign Key referencing DimServiceCategory)
  + Quantity
  + Amount

This schema design follows the star schema model, with one central fact table (FactSale) surrounded by dimension tables (DimMember, DimFranchise, DimTime, DimMerchandise, DimServiceCategory).



For personal reference, I wrote the code for the database:

-- Dimension Table: DimMember

CREATE TABLE DimMember (

MmbrId INT PRIMARY KEY,

MmbrName VARCHAR(100),

MmbrZip VARCHAR(10),

MmbrEmail VARCHAR(100),

MmbrDate DATE,

MemTypeId INT,

MemTypeName VARCHAR(50),

MemTypePrice DECIMAL(10, 2)

);

-- Dimension Table: DimFranchise

CREATE TABLE DimFranchise (

FranchId INT PRIMARY KEY,

FranchRegion VARCHAR(50),

FranchPostalCode VARCHAR(10),

FranchModelType VARCHAR(50)

);

-- Dimension Table: DimTime

CREATE TABLE DimTime (

DateId INT PRIMARY KEY,

Date DATE,

Year INT,

Quarter INT,

Month INT,

Day INT

);

-- Dimension Table: DimMerchandise

CREATE TABLE DimMerchandise (

MerchId INT PRIMARY KEY,

MerchName VARCHAR(100),

MerchPrice DECIMAL(10, 2),

MerchType VARCHAR(50)

);

-- Dimension Table: DimServiceCategory

CREATE TABLE DimServiceCategory (

ServCatId INT PRIMARY KEY,

ServCatName VARCHAR(50),

ServCatPrice DECIMAL(10, 2)

);

-- Fact Table: FactSale

CREATE TABLE FactSale (

SaleId INT PRIMARY KEY,

SaleDate DATE,

MmbrId INT,

FranchId INT,

DateId INT,

MerchId INT,

ServCatId INT,

Quantity INT,

Amount DECIMAL(10, 2),

CONSTRAINT FK\_MmbrId FOREIGN KEY (MmbrId) REFERENCES DimMember(MmbrId),

CONSTRAINT FK\_FranchId FOREIGN KEY (FranchId) REFERENCES DimFranchise(FranchId),

CONSTRAINT FK\_DateId FOREIGN KEY (DateId) REFERENCES DimTime(DateId),

CONSTRAINT FK\_MerchId FOREIGN KEY (MerchId) REFERENCES DimMerchandise(MerchId),

CONSTRAINT FK\_ServCatId FOREIGN KEY (ServCatId) REFERENCES DimServiceCategory(ServCatId)

);