**Club Membership System – Final Project**

**SAI KARTHIK GUTHA**

**STUDENT ID: 4238573**

**Master Of Science in Data Analytics**

**CSDA 5210 Databases and Data Warehouses**

**Webster University**

**Prof. James Ashford**

**December 13, 2023**

**Club Membership System**

**1.** **REQUIREMNTS OF CLUB MEMBERSHIP SYSTEM DATABASE**:

**Operations**: These are the operations will be capturing the club membership database.

* First, to register for membership in the club, whether the person the interested to join in the club.
* To know the membership plans and benefits for an interested in becoming a member.
* Select the membership plan (premium, basic, etc.,) and pay the amount by using types such as cash and cards.
* Event operations are managed and organized by the club members and to participate in all other extra-circular activities happen in the club as per dates announced.

**Information to be tracked**: These are the information will be tracked and recorded in the database.

* Member information to track such as member ID, first name, last name, contact, address.
* Payment information should be tracked such as transaction ID, type, amount for enrolling as member in the club.
* Event information should be tracked as event id, event name, event location, and these events are managed and organized by club members.
* Membership status information should be tracked such as status Type and status description (status name is used as the whether the membership is active or in active) to check the status of membership in the club.

**Relationships and Constraints**: These are the relationships and constraints captured in the database.

* Each member has a unique identifier such as MemberID.
* Each membership status has a unique identifier to check the status of membership of the member such as StatusID.
* Each payment has a unique identifier such as Trans\_ID or PaymentID, to check the status whether the member had paid the for the membership enrolling.
* Each Event has a unique identifier such as EventID to organize the events by the members in the club.

**2.** **ER DIAGRAM:**

A diagram of a diagram

Description automatically generated

*Entities*:

Entity: member

Attributes: memberID (primary key), fname, lname, city, phone number

Entity: membership status

Attributes: ID (primary key), status Type, Dues Amount, Benefits

Entity: event

Attributes: eventID (primary key), event name, date, time, type, event\_registration

Entity: payment

Attributes: trans\_id (primary key), name, date, amount

***Relationships***:

Member to Membership status (one-to-one relationship):

Every member can keep the status of membership of the club in active membership to organize the events in the club. Club members can organize the events.

Member to Payment (one-to-many relationship):

Every member must make payments for the membership for a year or month wise, to maintain and keep that membership active.

Member to Event (many-to-many relationships):

Every member in the club should organize the events in the club and invite attendees to events and must give an opportunity to participate attendees in the events if they are interested in the events.

**3.** **RELATIONAL SCHEMA:**

A screenshot of a diagram

Description automatically generated

The Club membership consists of four tables or relations, namely Member, Membership Status, Events, Payments. Each table has specific attributes or columns that hold information related to member, membership status, payments, and events.

The Member table holds information about each member in the club, including its memberID, member name (first name and last name), contact, address. In the Member table identified by a unique MemberID which is the primary key for this table.

The Membership Status table holds information about each membership of member in the club, including its ID, status Type (active, inactive, suspended), Dues Amount, Benefits. In this table identified by a unique ID which is the primary key for this table.

The Event table holds information about each event that can be organized by the members of the club, including its EventID, Event Name or Type, Event Date, Time, Event registration. In the event table identified by a unique EventID which is the primary key for this table.

The Payment table holds information about each payment that can be managed by the head of the clubs to maintain a club membership active, including its PaymentID (transaction id), Payment Date, Payment Amount. In thi stable identified by a unique PaymentID (transaction id) which is the primary key for this table.

**4. REFERENTIAL INTEGRITY CONSTRAINTS:**

Here is there is a referential integrity constraint for the club membership database, based on the provided attributes:

1. member table:

Primary key: ID

Foreign key: status type (references the primary key of the membership status table)

This constraint ensures that every member record has a valid status type associated with it. A member cannot have a status type that does not exist in the membership\_status table.

2. event table:

Primary key: ID

Foreign key (optional): memberID (references the primary key of the member table)

This constraint allows associating events with specific members (e.g., events organized by a specific member). However, it is optional because some events might not be linked to individual members.

3. payment table:

Primary key: ID

Foreign key: memberID (references the primary key of the member table)

This constraint ensures that every payment record is linked to a valid member. A payment cannot be assigned to a member that does not exist in the member table.

**User-Defined Constraints**:

Unique constraint on member ID: This constraint ensures that no two members have the same ID, preventing duplicate entries in the member table.

Unique constraint on event ID: Like the member ID, this constraint prevents duplicate event entries in the event table.

Check constraint on payment amount: This constraint ensures that the payment amount is always a non-negative value.

**Update/Delete Constraints**:

Restrict update/delete on membership\_status table: Updates or deletions to the membership status table should be restricted to prevent invalid references in the member table.

Cascade delete on event table: When a member record is deleted, all corresponding event records (if any) should be automatically deleted to maintain consistency.

Restrict deleting on payment table: Deleting a member record should not automatically delete associated payment records. Instead, a warning should be issued, prompting the user to confirm their action or oversee the payment records appropriately.

These referential integrity constraints ensure data validity and consistency within the club membership database. They prevent invalid references, maintain accurate relationships between tables, and promote data integrity.

**5. SQL IMPLEMENTATION**:

SQL queries for create and insert records in the table:

**Member**: (create and insert SQL queries of this table).

create table member (MemberID int PRIMARY KEY, Name varchar(25), Contact Varchar(25), Address varchar(255));

INSERT INTO Member VALUES (1, 'John Doe', '123-456-7890', '123 Main St');

INSERT INTO Member VALUES (2, 'John', '789-526-1234', '13th street');

INSERT INTO Member VALUES (3, 'Anthony', '635-489-2587', 'Westend st');

INSERT INTO Member VALUES (4, 'Daniel', '456-896-2587', 'baecon terrace');

INSERT INTO Member VALUES (5, 'Henry joseph', '258-789-6354', 'Albany Avenue');

INSERT INTO Member VALUES (6, 'Johnson', '789-589-4523', 'glenmade ct');

INSERT INTO Member VALUES (7, 'Sai', '789-456-1230', '456 whisper dr');

INSERT INTO Member VALUES (8, 'Albert', '564-897-3210', '145 whiney apts');

INSERT INTO Member VALUES (9, 'Dodge Rivers', '5897-635-2146', '1580 telegraph rd');

INSERT INTO Member VALUES (10, 'Willson', '456-897-2569', '123 campbell st');

**Membership Status**: (create and insert SQL queries in this table).

create table membershipstatus (StatusType VARCHAR2(30), MemberID INTEGER, FOREIGN KEY(MemberID) REFERENCES MEMBER(MemberID));

INSERT INTO MembershipStatus VALUES ('Active', 5);

INSERT INTO MembershipStatus VALUES ('Active', 10);

INSERT INTO MembershipStatus VALUES ('Inactive', 4);

INSERT INTO MembershipStatus VALUES ('Active', 6);

INSERT INTO MembershipStatus VALUES ('Inactive', 7);

INSERT INTO MembershipStatus VALUES ('Active', 1);

INSERT INTO MembershipStatus VALUES ('Active', 3);

INSERT INTO MembershipStatus VALUES ('Inactive', 8);

INSERT INTO MembershipStatus VALUES ('Inactive', 2);

INSERT INTO MembershipStatus VALUES ('Active', 9);

**Event**: (create and insert SQL queries of this table).

create table event ("EventID" number PRIMARY KEY, "Name" varchar2(60), "Date" DATE);

INSERT INTO EVENT VALUES (1, 'Social Gathering', TO\_DATE('2023-12-02', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (2, 'Christmas party', TO\_DATE('2023-12-23', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (3, 'musical concert', TO\_DATE('2023-12-29', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (4, 'NEW YEAR Party', TO\_DATE('2023-12-31', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (5, 'DJ Night', TO\_DATE('2023-12-20', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (6, 'club anniversory', TO\_DATE('2023-12-21', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (7, 'birthday parties', TO\_DATE('2023-12-10', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (8, 'halloween party', TO\_DATE('2023-10-31', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (9, 'fall feast', TO\_DATE('2023-11-30', 'YYYY-MM-DD'));

INSERT INTO EVENT VALUES (10, 'Dinner', TO\_DATE('2023-12-07', 'YYYY-MM-DD'));

**Payment**: (create and insert SQL queries of this table).

create table Payment ("PaymentID" int PRIMARY KEY, MemberID int, "Date" DATE, "Amount" DECIMAL(10, 2), FOREIGN KEY (MemberID) REFERENCES member(MemberID));

INSERT INTO Payment VALUES (0042, 1, TO\_DATE('2023-01-01','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (0733, 3, TO\_DATE('2023-12-20','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (0050, 4, TO\_DATE('2023-11-27','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (2000, 10, TO\_DATE('2023-11-26','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (1000, 9, TO\_DATE('2023-11-25','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (1054, 8, TO\_DATE('2023-11-29','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (0070, 2, TO\_DATE('2023-12-10','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (0200, 6, TO\_DATE('2023-12-04','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (3007, 5, TO\_DATE('2023-12-05','YYYY-MM-DD'), 50.00);

INSERT INTO Payment VALUES (0040, 7, TO\_DATE('2023-11-30','YYYY-MM-DD'), 50.00);

**6. DATA WAREHOUSE REQUIREMENTS**:

We Consider the database using the dimensions and fact table for the star schema.

*Subject of the analysis*: The subject of analysis is the club membership engagement and financial transactions.

*Fields and attributes*:

The fields or attributes that will be useful in the analysis include:

Member Information: ID, name, address, contact

Membership Status: ID, status type, Dues amount, Benefits

Event Details: ID, name, type, date, time, event registration

Payment Records: ID, date, type, amount

*Granularity*:

The granularity of the facts/measures depends on the level of detail required for analysis. In the context of the club membership database:

Member: Analyzing data at the individual member level would involve measures such as total payments made, events attended, and current membership status for each member.

Event: Analysis at the event level could include measures like the total attendance, total revenue generated, and the types of events that attract more members.

Payment: Analyzing payments might involve measures such as total revenue, average payment amount, and the distribution of payment types.

**7. STAR SCHEMA**:

This star schema consists of two types: Dimensions and Fact table.

These are the dimensions of the star schema:

Calendar: This dimension represents the daily updates from the club to check and maintain a membership, which is date, day, week, month, and year.

Member: This dimension represents registering and enrolling in the club using the member details such as ID, name, contact number, address, and membership type (to check the membership status whether it is active or not in the club).

Event: This dimension represents the conducting and managing of the events by members of the club, such as ID, name, type, date, time, and event registration.

Payment: This dimension represents the managing members' payments for the club membership such as transaction ID, date, type.

Membership status: This dimension represents the check the status of the membership of the member, such as ID, Type, Dues amount, benefits.

Fact table for the star schema:

Club membership Fact table: This Fact table represents all dimensions to connect as the foreign keys in the star schema.

Attributes:

Member key (primary key) establishes a relationship with the member dimension.

Event key (Foreign key referencing the key in the event table): Relates to the events attended by members.

Payment key (Foreign key referencing the key in the payment table): Relates to the payments made by members.

MembershipTypekey (Foreign key referencing the Type key in the membership status table): Represents the member's membership status.

Date: Date of the recorded event or payment.

Amount: The amount paid in the case of the payment record.

A diagram of a company

Description automatically generated