**Gym Chain Database Design Document**

Team 12: Jian Xiao Jiaqi Wang Jingya Zhou

Shuai Shao Yuwei Cao

**Database Purpose**

The purpose of the database is to maintain the data of staffs, members, fitness courses and individual health plans in a gym chain while making daily affairs more smoothly and effectively. It will be used by staff and members under different permissions.

**Business Problem Addressed**

• We can provide targeted advice for members by collecting and analyzing their personal information. The features from information collected can show what is the key point that users care and the degree it matters. So that we can help the members adjust their strategies of body shaping in order to achieve more effectiveness and efficiency.

• Quantify the performance of coaches and staffs and then give them performance improvement advice.

• Maintain sales and cost records in order to generate yearly financial report and then reasonably predict future profit.

**Business Rules**

• Each staff will work at exactly one gym.

• Each gym will have one or more staff.

• Each member will have one membership.

• Each membership will belong to one membership type.

• Each membership type will have zero or more membership.

• Each gym will have one or more equipment.

• Each equipment will be bought by one gym.

• Each staff will have zero or more evaluations.

• Each evaluation will belong to one staff.

• Each member will be taught by zero or more staffs.

• Each member will attend zero or more curriculum.

• Each curriculum will have zero or more member.

• Each curriculum will be conducted by zero or more staffs.

• Each staff will conduct zero or more curriculum.

• Each staff will teach zero or more members.

• Each member will have zero or one diet plan.

• Each diet plan will be used by zero or more member.

• Each member will have zero or one physical condition research.

• Each physical condition research will belong to one member.

• Each member may buy zero or more commodities.

• Each commodity may be bought by zero or more members.

• Each staff may sell zero or more commodities.

• Each commodity may be sold by zero or more staffs.

• Each sale of membership will have one flow record.

• Each sale of equipment will have one flow record.

• Each sale of commodity will have one flow record.

• Each sale of curriculum will have one flow record.

**Design Requirements (Credit to Professor Simon Wang)**

• Use Crows Foot Notation.

• Specify the primary key fields in each table by specifying PK beside the fields.

• Draw a line between the fields in each table to show the relationship between each table. This line should be pointed directly to the fields in each table that are used to form the relationship.

• Specify which table is on the side of relationship by placing a one next to the field where the line starts.

• Specify which table is on the many side of the relationship by placing a crow’s feet symbol next to the field where the line ends.

**Design Decisions**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **How Entity is Related to Other Entities** |
| **Gym** | A gym is the central entity in this database design. All other entities are derived based on their relationship to the gym and in support of the gym. The gym does not have many attributes itself because much of the information is stored in related tables such as Staff and Membership. | The gym has primary key GymID referenced by Staff, Membership and Equipment. |
| **MembershipType** | MembershipType specifies the type of the membership card and the corresponding expense. It is a simple entity used to just provide a common set of categories which can be associated with Membership to correlate them. | The only relationship with MembershipType is its primary key MembershipTypeID is referenced by Membership as a foreign key. This allows the Membership to know the category to which it belongs. |
| **Membership** | Membership contains the message about the duration of the membership, the specific date of the purchase, the gym that the transaction took place. Additionally, income from membership card to the gyms. | Membership relates to Gym, MembershipType and Flow by taking their primary keys as foreign keys. |
| **Staff** | Staff entity includes the content about the basic information of the specific staff, the gym he/she is now working in and salaries he/she earns. | Staff relates to Gym with GymID and its primary key StaffID is referenced by CommodyBought, MemberCoachCurriculum and AnonymousEvaluationSystem as foreign key. |
| **Flow** | Flow provides a coupling relationship between income and expenditure. | Flow provides references for Equipment, MemberCoachCurriculum, CommodityBought and Membership through foreign keys. |
| **Equipment** | Equipment entity tells the basic information about the device. Moreover, the gym that the appliance belongs to. | Equipment references Gym and Flow via foreign keys. |
| **MemberCoachCurriculum** | MemberCoachCurriculum relates the Member to the Curriculum he/she has bought and the Staff who is assigned to teach the course. | MemberCoachCurriculum references the Curriculum, Staff who teaches the class and Member who has bought the lesson. It also has a reference to the Flow to get to know the Course revenue. |
| **AnonymousEvaluationSystem** | AnonymousEvaluationSystem works only to collect the feedback from the customers in order to evaluate staff’s work reasonably. | AnonymousEvaluationSystem is related only to Staff. There is a one-to-many relationship between them. In other words, one staff corresponds to multiple reviews. |
| **Curriculum** | Curriculum is used for involving the specific course info. | Curriculum is related to and referenced only by MemberCoachCurriculum. |
| **Member** | Member is also the central entity in this database design. It contains the information about a specific person who has conducted consumption activities in the gym. | Member contains a foreign key reference to Membership, physical condition, as well as DietPlan. It’s also referenced by MemberCoachCurriculum and CommodityBought. |
| **MemberPhysicalCondition** | Member’s physical Condition is created to show the current health situation of the member. | Member’s physical Condition is referenced by Member. |
| **CommodityBought** | CommodityBought entities are a primary part of the database design. They track all the different sales that take place at a gym along with which Staff made the CommodityBought and which member purchased it. They contain other attributes such as the sale date, total, and mode of payment. | CommodityBought relates to Commodity and have a foreign key reference to Flow in order to determine the purchased specific products and income generated. Additionally, CommodityBought also have foreign keys to Staff and Member to show who sold and purchased the sale. |
| **Commodity** | Commodity contains information specific to a single type of sellable item, such as name, function, price. | The primary key of Commodity is referenced by CommodityBought as a foreign key. This way, the Commodities being sold as well as which CommodityBought they are a part of can be linked together. |
| **DietPlan** | Diet Plan is only created for keeping the Diet menu for fitness use. | Diet Plan is only related and referenced by Member. |