

# 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 maintaining and updating the basic data of staff, member, curriculum and flow in a gym chain to make daily affairs more smoothly and effectively while providing some suggestions for future development. It will be used by staffs and members under different permissions.

## Business Problem Addressed

- By recording the staff and related information, the administrators can have an overall view about the organization structure of this gym chain.
- According to the physical condition of individual members, combining with their diet preferences and fitness goals, give them workout plan and suggestions on the use of supplements.
- Evaluating the performance of staffs by members' anonymous appraises and making it as a reference for the promotion in the future.
- Using flow of each gym to generate yearly financial report, the administrators can have an appropriate plan which can bring more profit next year also adjust the development direction referring to members' reactions.

## Business Rules

- Each staff works at exactly one gym.
- Each gym has one or more staffs.
- Each member has one membership.
- Each gym has zero or more members.
- Each membership belongs to one membership type.
- Each membership type has zero or more members.
- Each gym has one or more equipment.
- Each equipment was bought by one gym.
- Each staff has one position.
- Each position has zero or more staffs.
- Each staff has zero or more evaluations.

- Each evaluation record belongs to one staff.
- Each member can be taught by zero or more coaches.
- Each member can attend zero or more curriculums.
- Each curriculum has zero or more members.
- Each curriculum is conducted by zero or more coaches.
- Each coach can conduct zero or more curriculums.
- Each member can have zero or one diet plan.
- Each diet plan is used by zero or more members.
- Each member has zero or one physical condition research.
- Each physical condition record belongs to one member.
- Each member may buy zero or more commodities.
- Each member may buy commodities via zero or more staffs.
- Each commodity may be bought by zero or more members.
- Each commodity may be sold by zero or more staffs several times.
- Each staff may sell zero or more commodities.
- Each staff may sell commodities to zero or more members.
- Each time of applying a membership generates one flow record.
- Each equipment procurement generates one flow record.
- Each transaction of commodity has one flow record.
- Each signing up for a curriculum has 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
<b>Gym</b>	Gym is the central entity in this database design. All other entities are derived based on their relationship to the gym directly or indirectly. It does not have many attributes of itself because other information is stored in related tables such as Staff and Membership.	As the core entity in the database, the Gym's primary key, GymID, relates it to Membership, Staff and Equipment so that insights can be gained by the relationship.
<b>MembershipType</b>	MembershipType specifies the type of the membership card and the corresponding attributes. It is a simple entity which can be used to provide a common set of options when there is a new membership occurs or just one membership already existing before needs to be altered.	The MembershipType entity is directly related to the Membership through its primary key. Due to the one-to-many relationship, one type of membership can have zero or several records of Membership.
<b>Membership</b>	Membership contains the message about the duration of each individual membership, the specific date of the purchasing, the gym that the transaction took place also a flow record for calculating income and expenses.	Each Membership must have one MembershipType. Deleting the last record in one specific type of membership will not delete the basic information of this membership type. Each Membership must relate to one Member.
<b>Staff</b>	Staff entity includes the basic information of staffs, the gym that he/she is now working for and the feedback from his/her customer.	Each Staff has a unique StaffID and relate to entity Gym by its foreign key GymID. At the same time, each staff must have one Position. Staffs who are coaches can teach zero or several fitness courses.
<b>Position</b>	Position entity serves only to indicate the specific position of each staff.	Position is related and referenced only by Staff. The many-to-one relationship indicates that there may be multiple people in the same position and each one can only in one position.
<b>Flow</b>	Flow provides a coupling relationship between income and expenditure.	Flow connects MemberCoachCurriculum, CommodityBought and

		Membership through their foreign keys.
<b>Equipment</b>	Equipment entity records the basic information about the fitness equipment as well as the gym that the equipment belongs to.	Each equipment has one unique ID no matter whether they are the same equipment. It relates to Gym by its foreign key.
<b>MemberCoachCurriculum</b>	MemberCoachCurriculum is a linked entity for members, coaches and curriculum. It describes any members in any courses taught by specific coaches. Also, a flowID is recorded for future calculating.	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.
<b>AnonymousEvaluationSystem</b>	AnonymousEvaluationSystem works only to collect the feedback from the customers in order to evaluate staff's work reasonably.	AnonymousEvaluationSystem is related to and referenced only by Staff. There is a one-to-many relationship between them. In other words, one staff can have zero or multiple reviews.
<b>Curriculum</b>	Curriculum is used for involving the specific course info.	Curriculum is related to Member and Staff through MemberCoachCurriculum.
<b>Member</b>	Member is another central entity in this database design. It contains the information of each member who has enrolled in one gym.	Member contains foreign keys reference to Membership, physical condition, as well as DietPlan. It's also referenced by MemberCoachCurriculum and CommodityBought.
<b>MemberPhysicalCondition</b>	MemberPhysicalCondition is created to show the current health situation of the members.	Each record relates to one Member and each member can have zero or one condition record.
<b>CommodityBought</b>	CommodityBought track the commodity purchasing taking place at each gym along with the related people's information. The detail of each purchasing is also collected.	CommodityBought is a linked entity for members, staffs and commodities. Each CommodityBought record must related to one member, one staff and one commodity.
<b>Commodity</b>	Commodity contains the basic information of each item available, such as name, function and price.	Commodity only contains a foreign key reference to CommodityBought. Each commodity can be sold zero or several times by different staffs to different members.

<b>DietPlan</b>	DietPlan is created for keeping the Diet record for referencing. Coaches can give individual exercise plan combining member's diet.	Diet Plan is only related and referenced by Member. Each member can have zero or one dietplan. The deleting of one record of member will not influence the existing of dietplan.
-----------------	---	--