Silver Oaks Cooperative School Database

**Mission Statement:** To enhance the efficiency of the enrollment and administrative process of the school by creating a database to allow entry and tracking of information of key entities. At the end we should be able to generate important documents like membership directory, class rosters, service hour reports, attendance reports, etc.

**Mission Objective:**

1. Get the details of the top 3 Members who are actively engaged in Coop tasks at the school
2. Get details of those students, not having emergency contact details in database.
3. Get the details of teachers, teaching particular classes at the school.
4. Get student details having less than average attendance. Fetch corresponding Medical details and analyze the correlation between low attendance and poor medical conditions if any.
5. Get details of all the members and tasks performed by them.

**ER Schema:**

Entities, Attributes, and Primary Keys

Member(**memId**,memName, -memFirstName,-memLastName, placeOfEmployment, memAddress, -memStreet, -memCity, -memState, -memZip, memCellNumber, memHomeNumber, memEmailId, proxy, *proxyId*)

Student( **stdtId,** stdName, -stdFirstName, -stdLastName,stdDOB, sex, stdEnrollmentDate, stdAttendance, stdGrade,stdAddress, -stdStreet, -stdCity, -stdState, -stdZip,enrollmentAgreement, fingerprintReceipt, physicalExamForm,childCareReport, liabilityRelease, medicalAdministratorAuthorisation, immuneCertificate)

StudentMedical(**stdMedId**, ***stdId***, medConditions, medications, tetanusShotDate, foodAllergy, drugAllergy, signsSymptons, responseDescp, preventionDescp, comments, medicalFormVerified)

HealthcareProvider(**providerId**, providerName, providerNumber, providerStreet, providerCity, providerState, providerZip)

EmergencyContact(**emrContactId**, ***stdId*,** emrFirstName, emrLastName, emrHomeNumber, emrWorkNumber, emrContactStreet, emrContactCity, emrContactState, emrContactZip)

ServiceTask(**taskId***,* serviceTask, organiser)

Classes(**classId** , className, classDescription)

Staff( **staffId** , staffFirstName , staffLastName , staffPosition, staffPhone, staffdateOfBirth, staffStreet , staffCity , staffState, staffAttendence)

Relationships, Attributes, Degrees, Participating Entities and Constraints

Membership(***stdId****,* ***memId***, relationshipToChild, lastUpdate, additionalInformation, coopType, coopDays, mediaConsent, specialPreferences)

1 student to 1 or more members

1 member to 1 or many

Authorize(***stdId****,* ***providerId****,* ***memId****,* memPermission, autsignature, autDate)

1 student and 1 member to 1 provider

1 member and 1 provider to 1 or many students

1 student and 1 provider to 1 or more member

Enroll **(*stdId***, ***classId***)

1 student to 1 class

1 class to many students

Teach**(*staffId, classId*** )

1 staff to 1 or many classes

1 class to 1 or many staff

Perform(***taskId, memId,*** memServiceHours, datePerformed)

1 task to 1 and more members

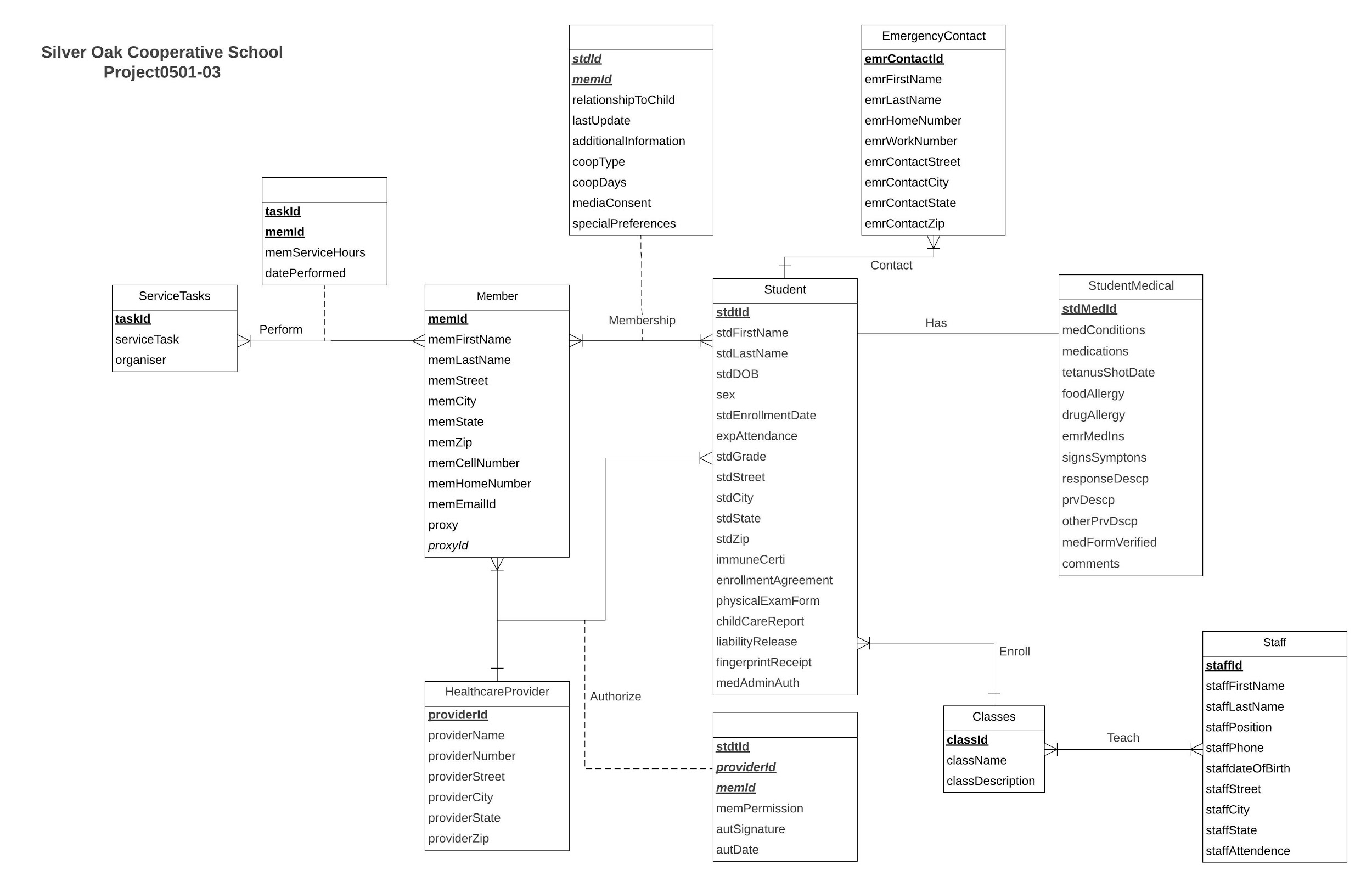
1 member to 1 and more tasks

Contact(***stdId , emrContactId)***

1 student 1 or more emergency contacts

1 emergency contact to 1student???

ER Diagram:



Functional Dependencies

memId → memFirstName, memLastName, placeOfEmployment, memStreet, memCity, memState, memZip, memCellNumber, memHomeNumber, memEmailId, proxy,proxyId

stdtId **→** stdFirstName, stdLastName,stdDOB, sex, stdEnrollmentDate, stdAttendance, stdGrade, stdStreet, stdCity, stdState, stdZip, enrollmentAgreement, fingerprintReceipt, physicalExamForm,childCareReport, liabilityRelease, medicalAdministratorAuthorisation, immuneCertificate

stdMedId, stdId → medConditions, medications, tetanusShotDate, foodAllergy, drugAllergy, signsSymptons, responseDescp, preventionDescp, comments, medicalFormVerified

providerId→ providerName, providerNumber, providerStreet, providerCity, providerState, providerZip

emrContactId, stdId**→** emrFirstName, emrLastName, emrHomeNumber, emrWorkNumber, emrContactStreet, emrContactCity, emrContactState, emrContactZip

taskId*→* serviceTask, organiser

classId → className, classDescription

staffId → staffFirstName , staffLastName , staffPosition, staffPhone, staffdateOfBirth, staffStreet , staffCity , staffState, staffAttendence

stdId, memId → relationshipToChild, lastUpdate, additionalInformation, coopType, coopDays, mediaConsent, specialPreferences

taskId, memId → memServiceHours, datePerformed

stdId, providerId, memId→ memPermission, autsignature, autDate

Normalization:

Member(**memId**,memFirstName,memLastName, placeOfEmployment, memStreet, memCity, memState, memZip, memCellNumber, memHomeNumber, memEmailId, proxy,*proxyId*)= 3NF

Student( **stdtId,**stdFirstName, stdLastName,stdDOB, sex, stdEnrollmentDate, stdAttendance, stdGrade, stdStreet, stdCity, stdState, stdZip,enrollmentAgreement, fingerprintReceipt, physicalExamForm,childCareReport, liabilityRelease, medicalAdministratorAuthorisation, immuneCertificate) = 3NF

StudentMedical(**stdMedId**, ***stdId***, medConditions, medications, tetanusShotDate, foodAllergy, drugAllergy, signsSymptons, responseDescp, preventionDescp, comments, medicalFormVerified, severity) = 3NF

HealthcareProvider(**providerId**, providerName, providerNumber, providerStreet, providerCity, providerState, providerZip) = 3NF

EmergencyContact(**emrContactId**, ***stdId*,** emrFirstName, emrLastName, emrHomeNumber, emrWorkNumber, emrContactStreet, emrContactCity, emrContactState, emrContactZip) = 3NF

ServiceTasks(**taskId***,* serviceTask, organiser) = 3NF

Classes(**classId** , className, classDescription) = 3NF

Staff( **staffId** , staffFirstName , staffLastName , staffPosition, staffPhone, staffdateOfBirth, staffStreet , staffCity , staffState, staffAttendence) = 3NF

Membership(***stdId****,* ***memId***, relationshipToChild, lastUpdate, additionalInformation, coopType, coopDays, mediaConsent, specialPreferences) = 3NF

Authorize(***stdId****,* ***providerId****,* ***memId****,* memPermission, autsignature, autDate) = 3NF

Enroll **(*stdId***, ***classId***) = 3NF

Teach**(*staffId*** ***, classId*** ) = 3NF

Has(***stdId***, ***stdMedId) =*** 3NF

Perform(***taskId, memId,*** memServiceHours, datePerformed) = 3NF

Contact(***stdId , emrContactId)*** *=3NF*

Business Rules:

[R01] When a student leaves the school and is deleted from the database, the corresponding contact records should be deleted.

[R02] When a student’s information is updated in the database, the corresponding contact records should be updated accordingly.

[R03] When a student leaves the school and is deleted from the database, the corresponding membership records should be deleted.

[R04] When a student’s information is updated in the database, the corresponding membership records should be updated accordingly.

[R05] When a student leaves the school and is deleted from the database, the corresponding authorize records should be deleted.

[R06] When a student’s information is updated in the database, the corresponding authorize records should be updated accordingly.

[R07] When a member leaves the school and is deleted from the database, the corresponding perform records should be deleted.

[R08] When a member information is updated in the database, the corresponding task perform records should be updated accordingly.

[R09] When a member leaves the school and is deleted from the database, the corresponding membership records should be deleted.

[R10] When a member information is updated in the database, the corresponding membership records should be updated accordingly.

[R11] When a member leaves the school and is deleted from the database, the corresponding authorize records should be deleted.

[R12] When a member information is updated in the database, the corresponding authorize records should be updated accordingly.

[R13] When a student leaves the school and is deleted from the database, the corresponding student medical records should be deleted.

[R14] When a student’s information is updated in the database, the corresponding student medical records should be updated accordingly.

[R15] When a healthcare provider is deleted from the database, the corresponding authorize records should be deleted.

[R16] When a healthcare provider changes id number, the corresponding authorize records should be updated accordingly.

[R17] When a staff leaves the school and is deleted from the database, the corresponding teach records should be deleted.

[R18] When a staff’s information is updated in the database, the corresponding teach records should be updated accordingly.

[R19] When the class is no longer taught and deleted from the database, the corresponding teach records should be deleted.

[R20] When the class information is updated in the database, the corresponding teach records should be updated accordingly.

[R21] When the class is no longer taught and deleted from the database, the corresponding enroll records should be deleted.

[R22] When the class information is updated in the database, the corresponding enroll records should be updated accordingly.

[R22] When the service task is no longer done and deleted in the database, the corresponding perform record should be deleted.

[R23] When the service task information is updated in the database, the corresponding perform records should be updated accordingly.

[R24] When the emergency contact is no longer affiliated with the student and deleted in the database, the corresponding contact record should be deleted.

[R25] When the emergency contact information is updated in the database, the corresponding contact record should be updated accordingly.

[R26] When the student is no longer a part of the school and deleted from the database, the corresponding enroll record should be deleted.

[R27] When the student information is updated in the database, the corresponding enroll record should be updated accordingly.

In addition, for each student enrolled, half-coopers spend one full day every four weeks assisting in a classroom as the teacher’s aide. Full coopers spend one full day every two weeks assisting in a classroom as the teacher’s aide for each student enrolled. If a family has more than one student enrolled, they may select different coop levels for each student.

Referential Integrities:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Relation** | **Foreign Key** | **Base Relation** | **Primary Key** | **Business Rule** | **Constraint: ON DELETE** | **Business Rule** | **Constraint: ON UPDATE** |
| Perform | memId | Member | memId | 07 | CASCADE | 08 | CASCADE |
| Perform | taskId | ServiceTasks | taskId | 22 | CASCADE | 23 | CASCADE |
| Authorize | providerId | HealthcareProvider | providerId | 15 | CASCADE | 16 | CASCADE |
| Authorize | memId | Member | memId | 11 | CASCADE | 12 | CASCADE |
| Authorize | stdId | Student | stdId | 05 | CASCADE | 06 | CASCADE |
| Membership | memId | Member | memId | 09 | CASCADE | 10 | CASCADE |
| Membership | stdId | Student | stdId | 03 | CASCADE | 04 | CASCADE |
| Contact | stdId | Student | stdId | 01 | CASCADE | 02 | CASCADE |
| Contact | emrContactId | Contact | emrContactId | 24 | CASCADE | 25 | CASCADE |
| Enroll | classId | Classes | classId | 21 | CASCADE | 22 | CASCADE |
| Enroll | stdId | Student | stdId | 26 | CASCADE | 27 | CASCADE |
| Teach | staffId | Staff | staffId | 17 | CASCADE | 18 | CASCADE |
| Teach | classId | Classes | classId | 19 | CASCADE | 20 | CASCADE |
| Has | stdId | Student | stdId | 13 | CASCADE | 14 | CASCADE |

**Sample Data:**

Member(10000001,’Jack’,’Lee’, ‘Baltimore’, ‘Rowalt Dr.’, ‘College Park’, ‘Maryland’, 20740, 307-938-9782, 123-321-3981, ‘jack.lee@gmail.com’, ‘Alan watson’ ,0000890)

Student( 00000001, ’Aady’ , ’Lee’, 2006-12-17, ‘M’, 2012-03-19, ‘80%’, ‘Saplings’, ‘Rowalt Dr’, ‘College Park’, ‘Maryland’, 20740, ‘Y’, ‘Y’, ‘Y’,’Y’, ‘Y’, ‘Y, ‘Y’)

StudentMedical(20000001, 00000001,’ Asthma’, ‘Albuterol’, 2012-02-19, ‘Peanuts’, ‘NA’, ‘Wheezing, Coughing’, ‘Take puffs from reliever inhaler’, Vaccination for influenza, Take prescribed medication’, ‘NA’, ‘Y’)

HealthcareProvider(30000001, ‘United Health Care’, 19999999999 , ‘Mowatt Lane’, ‘College Park’, ‘Maryland’,20742)

EmergencyContact(40000001, 00000001**,** ‘Jack’, ‘lee’, 307-938-9782, 123-321-3981, ‘Rowalt Dr.’, ‘College Park’, ‘Maryland’, 20740)

ServiceTasks(50001112*,* ‘Ann Taylor’)

Classes(60000001 , ‘Business Process Analysis’, ‘2 credit course meets monday and wednesday from 630- 745 PM’’)

Staff( 70000000, ‘James’, ‘Brody’, ‘Teacher’, 307-966-9342, 1981-07-19, ‘Mowatt Lane’, ‘College Park’, ‘Maryland’, 90%)

Membership(00000001*,* 10000001, ‘Parent’, 11-05-2013, ’NA’, ‘Full coop’, ‘T Th’, ‘Y’, ‘N’)

Authorize(00000001*,* 30000001*,* 10000001*,* ‘Y’, ‘ Jack Lee’ , 11-05-2013)

Enroll (00000001, 50000001)

Teach(70000000*,* 50000001)

Perform(50001112*,* 10000001, ‘cleaning the playground’ ,5, 2012-07-19,)

Contact(00000001*,* 40000001*)*