

DREAMEFV DATABASE

Data Dictionary and Documentation

Created by Michal Shaffer 2020

Contents:

Table of Contents

General Information	6
Introduction.....	6
Purpose	6
Audience.....	6
Naming Conventions	6
Schemas and Tables	7
dbo Schema	7
Account.....	7
Affiliation	7
Address	8
AddressAffiliation	9
Campaign	9
Contact	10
ContactType.....	11
TypeStatus.....	11
enroll Schema.....	12
Attendance	12
Course.....	13
Grade.....	13
Section.....	14
SectionEnrollment	15
Term.....	15
fund Schema.....	16
Opportunity	16
Payment.....	17
RecDonation	18
volunteer Schema.....	18
Detail.....	19
Hour	19

Job	20
JobDetail	20
Shift.....	21
SkillAvail.....	21
Triggers.....	22
DateMod Trigger	22
Sequences (and custom ID fields)	22
User Defined Functions.....	22
dbo.fnCheckTypeStatus	22
dbo.fnCheckContactExists	23
Views.....	23
dbo Schema	23
vw_AccountWithAddresses	23
vw_AccountWithContacts.....	23
vw_AccountWithRollup	23
vw_CampaignWithRollup	23
vw_ContactwithAffiliations.....	24
vw_ContactwithAddresses	24
vw_ContactwithRollup.....	24
enroll Schema.....	24
vw_CourseSections	24
vw_StudentEnrollments.....	24
fund Schema.....	24
vw_Opportunities.....	24
volunteer Schema	24
vw_SkillsbyContact.....	24
Jobs	25
DataCleanup	25
Stored Procedures	25
dbo Schema	25
SP_Create_Account.....	25
SP_Create_Address.....	25
SP_Create_Affiliation	26

SP_Create_Campaign.....	26
SP_Create_Contact.....	27
SP_Create_ContactType	28
SP_Delete_Account.....	28
SP_Delete_Address.....	29
SP_Delete_Affiliation	29
SP_Delete_Contact.....	29
SP_Update_Account.....	30
SP_Update_Address	30
SP_Update_Campaign	31
SP_Update_Contact	31
SP_Update_ContactType.....	32
usp_GetErrorInfo	32
enroll Schema.....	32
SP_Create_Attendance.....	32
SP_Create_Course	33
SP_Create_Grade	33
SP_Create_Section	34
SP_Create_SectionEnrollment.....	34
SP_Create_Term	35
SP_Delete_Course	35
SP_Delete_Grade	36
SP_Delete_Section.....	36
SP_Update_Attendance	36
SP_Update_Course.....	37
SP_Update_Section	37
SP_Update_SectionEnrollment	37
SP_Update_Term.....	38
fund Schema.....	38
SP_Create_Opportunity	38
SP_Create_Payment	39
SP_Create_RecDonation.....	39
SP_Update_Opportunity	40
SP_Update_Payment.....	41

volunteer Schema.....	41
SP_Create_Hour.....	41
SP_Create_Job.....	42
SP_Create_Shift.....	42
SP_Delete_Hour.....	43
SP_Delete_Jobs.....	43
SP_Delete_Shift.....	43
SP_Update_Jobs.....	44
Database Diagrams.....	45
dbo Schema.....	45
enroll Schema.....	46
fund Schema.....	47
volunteer Schema.....	48
full database diagram.....	49

General Information

Introduction

The DreamEFV Data Dictionary is a useful and necessary tool in collecting, reporting, and integrating enrollment, fundraising, and volunteer data at DREAM charter schools. It includes the names and descriptions of the various tables, views, functions, and other objects in the database. Each table definition is comprised of rows and columns, where the row corresponds to the values in the table, while the column corresponds to the data elements in the table. Each data element has a set of information that explains more about that piece of information, including data type and description.

The database is a SQL Server database created with T-SQL and using T-SQL definitions, functions, and datatypes.

Purpose

This document provides a comprehensive list of all DreamEFV database tables, views, functions etc for the complete understanding of the structure and processes of the database.

Audience

This document is intended for database administrators or developers who require database definitions or process clarification.

Naming Conventions

DreamEFV follows Sql Server standard naming conventions version 1.5 published by Shane Lively and Michael Sarsany.
(https://www.isbe.net/Documents/SQL_server_standards.pdf)

Schemas and Tables

dbo Schema

General or shared data

Account

Contains information of households, institutions, foundations, etc.

Column Name	Data Type	Description	Related objects
Id	nvarchar(9)	Identity constructed from the prefix 'AC' and a 7-digit numerical ID generated from a custom sequence dbo.SeqAccountId	Sequence, Primary Key
AccountName	nvarchar(100)	The name of the foundation, household, institution, or organization	
Type	int	Type of account with constraint that calls a function to confirm choice is from relevant values in dbo.Typestatus table.	Table, UDF, Constraint
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_dbo.Account_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Affiliation

Contains relationships between contacts and accounts. Ex student contact to school account or funder contact to organization account.

Column Name	Data Type	Description	Related Objects
-------------	-----------	-------------	-----------------

Id	int (identity)		Primary Key
AccountId	nvarchar(9)	The id from dbo.Account	FK > dbo.Account
ContactId	nvarchar(9)	the ID from dbo.Contact	FK > dbo.Contact
Role	nvarchar(30)	role of title description of the contact in the account	
isActive	bit		
isPrimaryAffiliation	bit		Default
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR__DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Address

Contains Address information for contacts and accounts.

Column Name	Data Type	Description	Related Objects
Id	int (identity)		Primary Key
Type	nvarchar(20)	Type of address like Home or Work from dbo.TypeStatus table	Table, UDF, Constraint
AddressCo	nvarchar(100)		
AddressLine1	nvarchar(100)		
AddressLine2	nvarchar(100)		
City	nvarchar(20)		
State	nvarchar(20)		
Zip	int		
Country	nvarchar(20)		
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_AddressCo_DateMod	Trigger

DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default
-------------	----------	---	---------

AddressAffiliation

Contains affiliations connecting an address to an account or contact

Column Name	Data Type	Description	Related Objects
Id			Primary Key
AddressId	int	The ID from dbo.Address	FK > dbo.Address
AccountId	nvarchar(9)	The id from dbo.Account	FK > dbo.Account
ContactId	nvarchar(9)	the ID from dbo.Contact	FK > dbo.Contact
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_Contains affiliations connecting an address to an account or contact_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Campaign

Contains fundraising and volunteer campaigns and events

Column Name	Data Type	Description	Related Objects
Id	nvarchar(9)	Identity constructed from the prefix 'CA' and a 7-digit numerical ID generated from a custom sequence dbo.SeqCampaignId	Sequence, Primary Key
Name	nvarchar(30)	Name of the campaign or event	
Type	int	Type of campaign or event like 'fundraiser abc' or 'math	Table, UDF, Constraint

		tutoring' from dbo.TypeStatus table	
StartDate	datetime	Date of the start of the campaign or event	
EndDate	datetime	Date of the end of the campaign or event	
Status	int	Status of campagin like 'planning' or 'in progress' from dbo.TypeStatus table	Table, UDF, Constraint
isActive	bit	shows whether the campaign is currently active	
AccountId	nvarchar(9)	The institution or department that is hosting the campaign or event	FK > dbo.Account
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_Name_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Contact

Contains all conatcts including funders, employees, students, parents, volunteers, etc

Column Name	Data Type	Description	Related Objects
Id	nvarchar(9)	Identity constructed from the prefix 'CU' and a 7-digit numerical ID generated from a custom sequence dbo.SeqCustomerId	Sequence, Primary Key
Title	nvarchar(10)		
FirstName	nvarchar(30)		
LastName	nvarchar(30)		
MiddleName	nvarchar(30)		
Suffix	nvarchar(10)		
Email	nvarchar(50)		
Email2	nvarchar(50)		
WorkPhone	nvarchar(20)	Phone numbers do not have constraints to check the format to allow for international phone numbers	

HomePhone	nvarchar(20)		
CellPhone	nvarchar(20)		
Fax	nvarchar(20)		
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_Suffix_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

ContactType

Contains information on what type or role a contact has at DREAM

Column Name	Data Type	Description	Related Objects
Id	int		Primary Key
ContactId	nvarchar(9)	The ID from dbo.Address	FK > dbo.Contact
AccountId	nvarchar(9)	The id from dbo.Account	FK > dbo.Account
Type	int	The Type of contact ex. student, parent, funder, employee etc	Table, UDF, Constraint
isActive	bit	shows whether the Contact is currently active as that type	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_dbo.ContactType_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

TypeStatus

Contains types and status for all tables with a table name and ID for reference

Column Name	Data Type	Description	Related Objects
Id	int (identity)		Primary Key
ForTableId	int	Id of the table the row can be used in	
ForTableName	nvarchar(30)	name of the table the row can be used in	
isType	bit	shows if the value is for a type field or a status field	
value	nvarchar(20)	the value	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR__DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

enroll Schema

Tables related to enrollment and school courses

Attendance

Contains records of student attendance at specific course sections

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
SectionEnrollmentId	int	The id of the section that the student is enrolled in and attending	FK > enroll.SectionEnrollment
AttendanceDate	datetime	Start date and time of the class	Table, UDF, Constraint
ArrivedTime	time	time that student arrived at class	
LeftTime	time	time that student left class	
isPresent	bit	shows if a student was present	

isLate	bit	shows if a student was late	
Note	nvarchar(100)		
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Course

Contains details of an offered academic course

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
AcademicInstitutionId	nvarchar(20)	The id from Accounts of the academic institution	FK > dbo.Account
CourseName	nvarchar(30)	The name of the course	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Course_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Grade

Contains records of student's enrollment for a specific term in a sepcific school year/grade

Column Name	Data Type	Description	Related objects
-------------	-----------	-------------	-----------------

Id	int (identity)		Primary Key
StudentId	nvarchar(20)	the Id of the student from contacts	FK > dbo.Contact
GradeName	nvarchar(30)	The name of the year/grade the student is in	
TermId	int	the id of the term that the student is enrolled in	FK > enroll.Term
isActive	bit	shows if the student is currently in this grade	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Grade_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Section

Contains all the sections that are offered per course

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
CourseId	int	the Id of the Course the section is for	FK > enroll.Course
TermId	int	The id of the term that the section is being offered in	FK > enroll.Term
InstructorId	nvarchar(20)	The Id of the instructor or teacher from dbo.Contact	FK > dbo.Account
ScDays	varchar(10)	Days of the week that the section is held ex. 'M,W,TH'	
ScTime	time	time of day the section starts	
ScMins	int	duration of section in minutes	
MaxStudents	int	the maximum seats in the class section	
Status	nvarchar(20)	the status of the section ex. current, active, past, cancelled, etc	Table, UDF, Constraint
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Section_DateMod	Trigger

DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default
-------------	----------	---	---------

SectionEnrollment

Contains the enrollment of students per each course section

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
StudentId	nvarchar(20)	the Id of the student from contacts	FK > dbo.Contact
SectionId	int	the id of the section that the student is enrolled in	FK > enroll.Section
Grade	nvarchar(10)	the score the student received for the section ex, A, A+, 98, Pass, Fail	
Status	nvarchar(20)	the status of the enrollment ex. active, upcoming, registered, waitlisted, dropped, completed	Table, UDF, Constraint
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.SectionEnrollment_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Term

Contains dates and information about academic terms

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary

			Key
TermName	nvarchar(20)	The name of the term ex. Fall2020, Summer2020	
StartDate	datetime	The date the term starts	
EndDate	datetime	The date the term ends	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Term_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

fund Schema

Tables related to fundraising events and lead and funder advancement

Opportunity

Contains records of donations that are requested or promised

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
CampaignId	nvarchar(20)	the Id of the campaign that this opportunity belongs to ex. A fundraising event	FK > dbo.Campaign
ContactId	nvarchar(20)	the Id of the contact providing the funds	FK > dbo.Contact
RecDonation	bit	shows whether this is part of a recurring donation schedule	
RecDonationId	int	the Id of the recurring donation schedule this opportunity belongs to	FK > fund.RecDonation
RecDonationInstal	int	the installation number of this opportunity in the recurring donation schedule	
Status	nvarchar(20)	The status of the opportunity ex. Promised, Complete,	Table, UDF, Constraint

		Dropped	
ReqAmount	decimal	the amount that was promised or requested	
WonAmount	decimal(18,2)	the amount that was received	Sum of payments recieved toward this opportunity - not calculated
DateExpected	datetime	date the full payments are expected	
DateClosed	datetime	date the opportunity was closed	
Name	varchar(100)	the name of the opportunity	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_fund.Opportunity_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Payment

Contains payments that corospond to opportunities

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
OpportunityId	int	the id of the opportunity donation that the payment belongs to	FK > fund.Opportunity
Type	nvarchar(20)	The type of payment ex.	Table, UDF, Constraint
Status	nvarchar(20)	The Status of the payment ex. Pending, completed, chargeback, cancelled	Table, UDF, Constraint
Paid	bit		
AmountReceived	decimal(18,2)	amount received	
WrittenOff	bit	shows whether the payment was cancelled or not	
Notes	varchar(max)	check number or other notes	
DateModified	datetime	Updated with current timestamp when a row is updated by using	Trigger

		table trigger TR_fund.Payment_DateMod	
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

RecDonation

Contains information about donations that are recurring with or without an end date

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
CampaignId	nvarchar(20)	the id of the fundraising event or campaign from dbo.Campaign	FK > dbo.Campaign
ContactId	nvarchar(20)	the Id of the funder from dbo.Contact	FK > dbo.Contact
TotalAmount	decimal(18,2)	The total amount promised	
InstalAmount	decimal(18,2)	The amount of each installment promised	
InstalPeriod	nvarchar(20)	the frequency of the donations ex. Monthly, weekly, bimonthly etc	
NumInstal	int	number of installments in this recurring donation schedule	
StartDate	datetime	date of the first installment	
EndDate	datetime	date of the last installment	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_fund.RecDonation_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

volunteer Schema

Tables related to volunteer events, shifts, and hours

Detail

Contains volunteering related information about volunteers

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
ContactId	int	The id of the volunteer from dbo.Contact	FK > dbo.Contact
SkillAvail	int	Id of the skill the volunteer has	FK > volunteer.SkillAvail
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Hour

Contains volunteer's hours information

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
ContactId	int	The id of the volunteer from dbo.Contact	FK > dbo.Contact
ShiftId	int	The id of the shift the volunteer is signed up for from Shift	FK > volunteer.Shift
CheckInDate	datetime	date and time of check-in for event	
CheckOutDate	datetime	date and time of check out for event	
Status	nvarchar(20)	Status of the hours worked ex. Upcoming, cancelled, checked in, confirmed	Table, UDF, Constraint
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger	Trigger

		TR_enroll.Attendance_DateMod	
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Job

Contains information on volunteer jobs within each campaign event

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
CampaignId	nvarchar(20)	the id of the event or volunteering program from dbo.Campaign	FK > dbo.Campaign
Name	varchar(30)	The name of the Job	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

JobDetail

Contains skills needed for Jobs

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
JobId	int	the ID of the job	FK > volunteer.Job
SkillNeededId	int	the ID of the skills needed for the job	FK > volunteer.SkillAvail
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Shift

Contains shift date and time information for volunteer jobs

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
JobId	int	the Id of the job that the shift belongs to	FK > volunteer.Job
StartDate	datetime	the date and time that the shift starts	FK > volunteer.Shift
EndDate	datetime	the date and time that the shift ends	
VolunteersNeeded	int	the number of volunteers needed for the shift	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

SkillAvail

Contains records of skills or availability that the volunteers have

Column Name	Data Type	Description	Related objects
Id	int (identity)		Primary Key
Name	varchar(30)	The name of the skill or availability	
DateModified	datetime	Updated with current timestamp when a row is updated by using table trigger TR_enroll.Attendance_DateMod	Trigger
DateCreated	datetime	Generated at row insertion by default constraint with the current timestamp	Default

Triggers

DateMod Trigger

Every table in the database has a trigger named TR_TableName_DateMod. This trigger performs on UPDATE and sets the DateModified field to the current timestamp on the updated rows.

Sequences (and custom ID fields)

Sequences are used to generate sequential unique numbers, primarily used for unique ID fields.

Sequence Name	Data Type	Start	Increment	Description
dbo.SeqAccountID	int	1000001	1	7-digit integer incrementer to use for Account IDs.
dbo.SeqCampaignID	int	1000001	1	7-digit integer incrementer to use for Contact IDs.
dbo.SeqContactID	int	1000001	1	7-digit integer incrementer to use for Campaign IDs.

User Defined Functions

dbo.fnCheckTypeStatus

Input: @field varchar(20), @table_id int, @isType bit

Process: Checks if the type or status value from @field is valid for the table and type

Output: True or False

Usage: Used in tables with a 'Type' or 'Status' field to check that the data is valid for that table. Called using a check constraint on the table columns.

dbo.fnCheckContactExists

Input: @contactid varchar(9)

Process: Checks if the contact id exists in dbo.Contact table

Output: True or False

Usage: Used in stored procedures to check if contact exists. Used for error handling.

Views

dbo Schema

vw_AccountWithAddresses

Description: Shows account data with all address information for each Account

vw_AccountWithContacts

Description: Shows account data with all contacts affiliated with each account

vw_AccountWithRollup

Description: Shows each account data with summary of number of active campaigns, total campaigns, contacts, addresses, etc.

vw_CampaignWithRollup

Description: Shows each campaign data with summary of number of opportunities, requested amounts, and won amounts from funding as well as the total jobs, shifts, and volunteer sign-ups from volunteer schema.

vw_ContactwithAffiliations

Description: Shows contact data with each contact's affiliations.

vw_ContactwithAddresses

Description: Shows contact data with each contact's address affiliations.

vw_ContactwithRollup

Description: Shows each contact data with summary of number of addresses, affiliations, opportunities, volunteer hours, payments, enrollments, etc.

enroll Schema

vw_CourseSections

Description: Shows all course section details by course name and institution.

vw_StudentEnrollments

Description: Shows all student enrollments with student grade information and section data.

fund Schema

vw_Opportunities

Description: Shows opportunity information with contact and campaign data and number of payments.

volunteer Schema

vw_SkillsbyContact

Description: Shows all contacts with their skills for easy lookup-by-skill.

Jobs

DataCleanup

Process:

Step 1: Check for unattached addresses – uses SQL script to delete addresses without affiliations.

Step 2: Check for unattached skills - uses SQL script to delete volunteer skills without attached jobs or contacts.

Schedule: Weekly, Occurs every week on Sunday at 12:00:00 AM. Schedule will be used between 5/14/2020 and 5/17/2022.

Notifications: Writes to Windows Application event log when the job completes.

Stored Procedures

dbo Schema

SP_Create_Account

Description: Creates an Account

Inputs: @accountname as varchar(100) = null

, @type as varchar(20)

Process: Inserts data into account table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if account already exists

SP_Create_Address

Description: Creates an Address for an account or contact

Inputs: @accountid as varchar(9) = null

, @contactid as varchar(9) = null

, @type as varchar(20)

```
, @addressCo as varchar(100)
, @address1 as varchar(100)
, @address2 as varchar(100)
, @city as varchar(20)
, @state as varchar(20)
, @zip as int
, @country as varchar(20)
```

Process: Checks address table to see if address already exists. If exists, saves ID. If not exists, inserts data into address table, saves generated ID. Inserts address ID into addressaffiliation table with the account or contact ID provided.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if address affiliation already exists for provided contact or account.

SP_Create_Affiliation

Description: Creates an affiliation for contact or account

Inputs: @accountid as varchar(9)
 , @contactid as varchar(9)
 , @role as varchar(30) = null
 , @isprimaryaffiliation as bit

Process: Inserts data into affiliation table.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Calls fnCheckContactExists function and raises custom error if contact does not exist. Raises custom error if affiliation already exists.

SP_Create_Campaign

Description: Creates a Campaign

Inputs: @Name varchar(100)

- , @Type varchar(20)
- , @StartDate datetime = null
- , @EndDate datetime = null
- , @Status varchar(20)
- , @isActive bit
- , @AccountId varchar(9)

Process: Inserts data into campaign table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if campaign already exists or that account.

SP_Create_Contact

Description: Creates a new contact

Inputs: @first as varchar(30)

- , @last as varchar(30)
- , @title as varchar(10) = null
- , @email as varchar(50)
- , @mobile as varchar(20) = null
- , @middle as varchar(20) = null
- , @suffix as varchar(10) = null
- , @email2 as varchar(50) = null
- , @workphone as varchar(20) = null
- , @homephone as varchar(20) = null
- , @fax as varchar(20) = null
- , @accountname as varchar(100) = null
- , @accounttype as varchar(50)
- , @addresstype as varchar(20)
- , @addressCo as varchar(100)
- , @address1 as varchar(100)

, @address2 as varchar(100)
, @city as varchar(20)
, @state as varchar(20)
, @zip as int
, @country as varchar(20)
, @contactType as varchar(20)

Process: Inserts data into contact table, saves newly generated ID. Inserts ID and type into contacttype table. Checks if account exists, if yes – saves ID. If not exists, inserts data into account table and saves newly generated ID. Inserts contact and account IDs into affiliation table. Updates account name with all account contact's information if not student. Calls dbo.SP_Create_Address to add address information for contact.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if customer already exists.

SP_Create_ContactType

Description: Adds a type for a contact

Inputs: @contactid varchar(9)
, @contactType as varchar(20)
, @isActive bit = 1

Process: Inserts data into contact type table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if contact does not exist (uses function fnCheckContactExists) or if type already exists for the contact.

SP_Delete_Account

Description: Delete an Account

Inputs: @accountid as varchar(9)

Process: Deletes record from account table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if account has active affiliations or campaigns.

SP_Delete_Address

Description: Delete an address

Inputs: @addressid int

Process: Deletes any address affiliations with that addressID from addressaffiliation table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises error if address does not exist.

SP_Delete_Affiliation

Description: Deletes affiliation

Inputs: @accountid as varchar(9)

, @contactid as varchar(9)

Process: Sets affiliation as inactive for contact and account given. If affiliation was the primary affiliation, the next possible affiliation for the household is updated to be primary affiliation.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if affiliation doesn't exist.

SP_Delete_Contact

Description: Deletes Contact

Inputs: @contactid as varchar(9)

Process: Deletes records from contacttype, addressaffiliations, affiliation, and contact tables for this contact id.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if contact has active affiliation, opportunities, enrollments, teaching positions, or volunteer hours.

SP_Update_Account

Description: Updates Account information

Inputs: @accountid as varchar(9)
 , @accountname as varchar(100) = null
 , @type as varchar(20) = null

Process: Updates account name and/or type

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if account doesn't exist or update causes a duplicate account.

SP_Update_Address

Description: Updates address information

Inputs: @addressid int
 , @type as varchar(20) = null
 , @addressCo as varchar(100) = null
 , @address1 as varchar(100) = null
 , @address2 as varchar(100) = null
 , @city as varchar(20) = null
 , @state as varchar(20) = null
 , @zip as int = null
 , @country as varchar(20) = null

Process: Updates address information in address table

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if address does not exist.

SP_Update_Campaign

Description: Updates Campaign

Inputs: @CampaignID varchar(9)
 , @Name varchar(100) = null
 , @Type varchar(20) = null
 , @StartDate datetime = null
 , @EndDate datetime = null
 , @Status varchar(20) = null
 , @isActive bit = null

Process: Updates campaign with new data

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if campaign does not exist.

SP_Update_Contact

Description: Updates contact's information

Inputs: @contactid as varchar(9)
 , @first as varchar(30) = null
 , @last as varchar(30) = null
 , @title as varchar(10) = null
 , @email as varchar(50) = null
 , @mobile as varchar(20) = null
 , @middle as varchar(20) = null
 , @suffix as varchar(10) = null
 , @email2 as varchar(50) = null
 , @workphone as varchar(20) = null
 , @homephone as varchar(20) = null

, @fax as varchar(20) = null
, @contactType as varchar(20) = null

Process: updates contact record with new data, replaces oldest contact type record if contacttype input is not null.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if contact does not exist.

SP_Update_ContactType

Description: Set a contact type as active or inactive

Inputs: @typeid varchar(9)

, @isActive bit

Process: updates contact type as active or inactive

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if contact type record does not exist.

usp_GetErrorInfo

Description: Gets Error information if called in the catch block of other stored procedure

Inputs: None

Process: Prints out data on Error

Output: ErrorNumber, ErrorSeverity, ErrorState, ErrorProcedure, ErrorLine, ErrorMessage

enroll Schema

SP_Create_Attendance

Description: Creates an attendance record for a student

Inputs: @studentid varchar(9)
 , @enrollmentid int
 , @attendancedate datetime = null
 , @ispresent bit = 1
 , @islate bit = 0
 , @arrivedtime time = null
 , @lefttime time = null
 , @note varchar(100) = null

Process: Sets attendancedate to current date if null. Updates attendancedate with time of class start. If isPresent = 1 the @arrivedTime is set to the class start time. Inserts attendance record for student.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if attendance record already exists for that student, in the section, on that day. Uses fnCheckContactExists function to raise custom error if student does not exist.

SP_Create_Course

Description: Creates a course

Inputs: @schoolid varchar(9)
 , @CourseName varchar(30)

Process: Creates course record in enroll.course table.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if course already exists for that school.

SP_Create_Grade

Description: Create a grade record for a student

Inputs: @studentid varchar(9)
 , @gradename varchar(30)

, @termid int
, @isActive bit = 0

Process: Inserts new grade for student. If @isActive = 1 (inserted grade is the active one) all other grade records for that student are updated to be inactive.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if grade already exists for that student or student does not exist (using fnCheckContactExists function).

SP_Create_Section

Description: Create a section for a course

Inputs: @courseid int

, @termid int
, @instructorid varchar(9) = null
, @maxstudents int = 50
, @status varchar(20) = 'Planned'
, @scdays varchar(20) = 'M,T,W,TH'
, @sctime time
, @scmins int = 50

Process: Inserts new section for a course

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if a section for that course exists with the same days, start time, and duration.

SP_Create_SectionEnrollment

Description: Creates an enrollment record for a student in a section.

Inputs: @sectionid int

, @studentid varchar(9)

, @grade varchar(10) = null
, @status varchar(20) = 'Current'

Process: Creates enrollment record for student.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if student does not exist (using fnCheckContactExists function), if section does not exist or is cancelled, or if enrollment already exists for that student in that section.

SP_Create_Term

Description: Create a term

Inputs: @name varchar(10)
, @StartDate datetime
, @EndDate datetim

Process: Crrates a term with start and end dates.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if a term with that name already exists.

SP_Delete_Course

Description: Delete a course

Inputs: @courseid int

Process: Deletes sections for the course and the course record if the sections are all in cancelled status.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error course does not exist or if active sections exist.

SP_Delete_Grade

Description: Deletes a grade (School year) record for a student.

Inputs: @gradeid int

Process: if record is the active grade, the latest grade is set to active. Grade record is deleted.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if grade record does not exist.

SP_Delete_Section

Description: Deletes a section for a course.

Inputs: @sectionid int

Process: Deletes section enrollments with status 'withdrawn' and deletes the section record.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if section does not exist or has active enrollments.

SP_Update_Attendance

Description: Updates attendance record for a student.

Inputs: @attendanceid int

, @ispresent bit = null

, @islate bit = null

, @arrivedtime time = null

, @lefttime time = null

, @note varchar(100) = null

Process: Sets sctime variable to the section start time. If @ispresent = 1 the arrivedtime is set to the sctime variable. The attendance record is updated.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if the attendance record does not exist.

SP_Update_Course

Description: Updates a course name

Inputs: @courseid int
 , @CourseName varchar(30)

Process: updates course name

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if course does not exist or if update causes a duplicate course.

SP_Update_Section

Description: Update a section

Inputs: @sectionid int
 , @instructorid varchar(9) = null
 , @maxstudents int = null
 , @status varchar(20) = null
 , @scdays varchar(20) = null
 , @sctime varchar(20) = null

Process: updates section information

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if section does not exist or if it has active enrollments and the new status is 'cancelled'.

SP_Update_SectionEnrollment

Description: Update an enrollment record

Inputs: @enrollmentid int

, @grade varchar(10) = null

, @status varchar(20) = null

Process: Updates an enrollment with a new grade (score) and/or status.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if enrollment record does not exist.

SP_Update_Term

Description: Update a term

Inputs: @termid int

, @StartDate datetime = null

, @EndDate datetime = null

Process: updates a term's start and end dates.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if term does not exist

fund Schema

SP_Create_Opportunity

Description: Create an opportunity record

Inputs: @campaignid varchar(9)

, @contactid varchar(9)

, @name varchar(100) = null

, @Status nvarchar(20) = 'Pending'

, @ReqAmount decimal(18,2)

, @WonAmount decimal(18,2) = null

```
, @dateexpected datetime = null  
, @dateclosed datetime = null  
, @RecDonation bit = 0  
, @RecDonationId int = null  
, @RecDonationInstal int = null
```

Process: Sets opportunity name to contact name – date expected or current date if name is null. Inserts record into opportunity table.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if enrollment record does not exist.

Raises custom error if contact does not exist (using fnCheckContactExists function).

SP_Create_Payment

Description: Creates a Payment

Inputs: @opportunityid int

```
, @type varchar(20)  
, @status varchar(20) = 1  
, @paid bit  
, @AmountReceived decimal(18,2)  
, @WrittenOff bit = 0  
, @Notes varchar(max) = null
```

Process: Inserts a record into payment table. Updates opportunity with the payment amount. Updates opportunity as closed won if full amount was paid.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if opportunity does not exist.

SP_Create_RecDonation

Description: Creates a recurring donation record and corresponding opportunity records.

Inputs: @CampaignId varchar(9)

- , @ContactId varchar(9)
- , @TotalAmount decimal (18,2) = null
- , @InstalAmount decimal(18,2) = null
- , @InstalPeriod varchar(20) = "Once a Month"
- , @StartDate datetime = null
- , @EndDate datetime = null
- , @NumInstal int = null
- , @OpenEnded bit = 0

Process: Sets startdate and current date if null. If TotalAmount was entered: installation amount is calculated by total amount divided by number of installations. If TotalAmount was not entered, the number of installations is calculated by amount of months between start and end dates. Recurring donation record is inserted and generated ID is saved. A loop creates an opportunity record for each donation, using NumInstal to see how many opportunities to create.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if TotalAmount and InstalAmount and EndDate are all null (this would not allow the procedure to calculate instalments properly.) Raises custom error if contact does not exist (using fnCheckContactExists function).

SP_Update_Opportunity

Description: Update an opportunity

Inputs: @opportunityid int

- , @Status nvarchar(20) = null
- , @ReqAmount decimal(18,2) = null
- , @dateexpected datetime = null
- , @dateclosed datetime = null

Process: Updates an opportunity requested amount, status, or dates.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if opportunity does not exist.

SP_Update_Payment

Description: Update a payment record

Inputs: @paymentid int

- , @type varchar(20) = null
- , @status varchar(20) = null
- , @paid bit = null
- , @WrittenOff bit = null
- , @Notes varchar(max) = null

Process: If status = void the opportunity is updated to reduce the won amount and the payment is updated with the new status.

Otherwise, the payment is updated. Note: The payment amount cannot be altered. You must void the payment and create a new one.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if payment does not exist.

volunteer Schema

SP_Create_Hour

Description: Creates volunteer hours record for a shift

Inputs: @ContactId varchar(9)

- , @ShiftId int
- , @skill1 varchar(20) = null
- , @skill2 varchar(20) = null
- , @skill3 varchar(20) = null

```
, @CheckInDate datetime = null  
, @CheckOutDate datetime = null  
, @Status varchar(20) = 'Interested'
```

Process: inserts volunteer hours record. Adds new skills to skillavail table and puts skill affiliations into volunteer.detail table.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if shift or contact does not exist (using fnCheckContactExists function) or if hours already exist for that volunteer on that shift.

SP_Create_Job

Description: Create a job for a campaign

Inputs: @CampaignId varchar(9)
 , @Name varchar(100)
 , @skill1 varchar(20) = null
 , @skill2 varchar(20) = null
 , @skill3 varchar(20) = null

Process: Inserts job record and saves generated job ID. Adds any job skills to SkillAvail table if they don't exist and saves IDs. IF they do exist, IDs are pulled. Job ID and skill IDs are inserted into jobdetail table if they don't already exist.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if campaign does not exist or job already exists with that name for that campaign.

SP_Create_Shift

Description: Create a shift for a campaign job

Inputs: @JobId int
 , @StartDate datetime = null
 , @EndDate datetime = null

, @VolNeeded int = null

Process: Adds shift record.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors.

SP_Delete_Hour

Description: Delete a record of volunteer hours

Inputs: @hourid int

Process: Sets volunteer hour status to 'Cancelled' if the start date has not yet passed.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if hour record does not exist or if the start date has already passed.

SP_Delete_Jobs

Description: Delete job from campaign

Inputs: @jobid int

Process: Deletes job if campaign is not completed or shift hasn't started.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if job does not exist or if campaign is completed, or if a shift for this job has already started.

SP_Delete_Shift

Description: Delete a shift from a job

Inputs: @shiftid int

Process: Deletes shift if it hasn't yet started.

Output: None

Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if shift does not exist or has already started.

SP_Update_Jobs

Description: Update a job for a campaign.

Inputs: @jobid int

, @skill1 varchar(20) = null

, @skill2 varchar(20) = null

, @skill3 varchar(20) = null

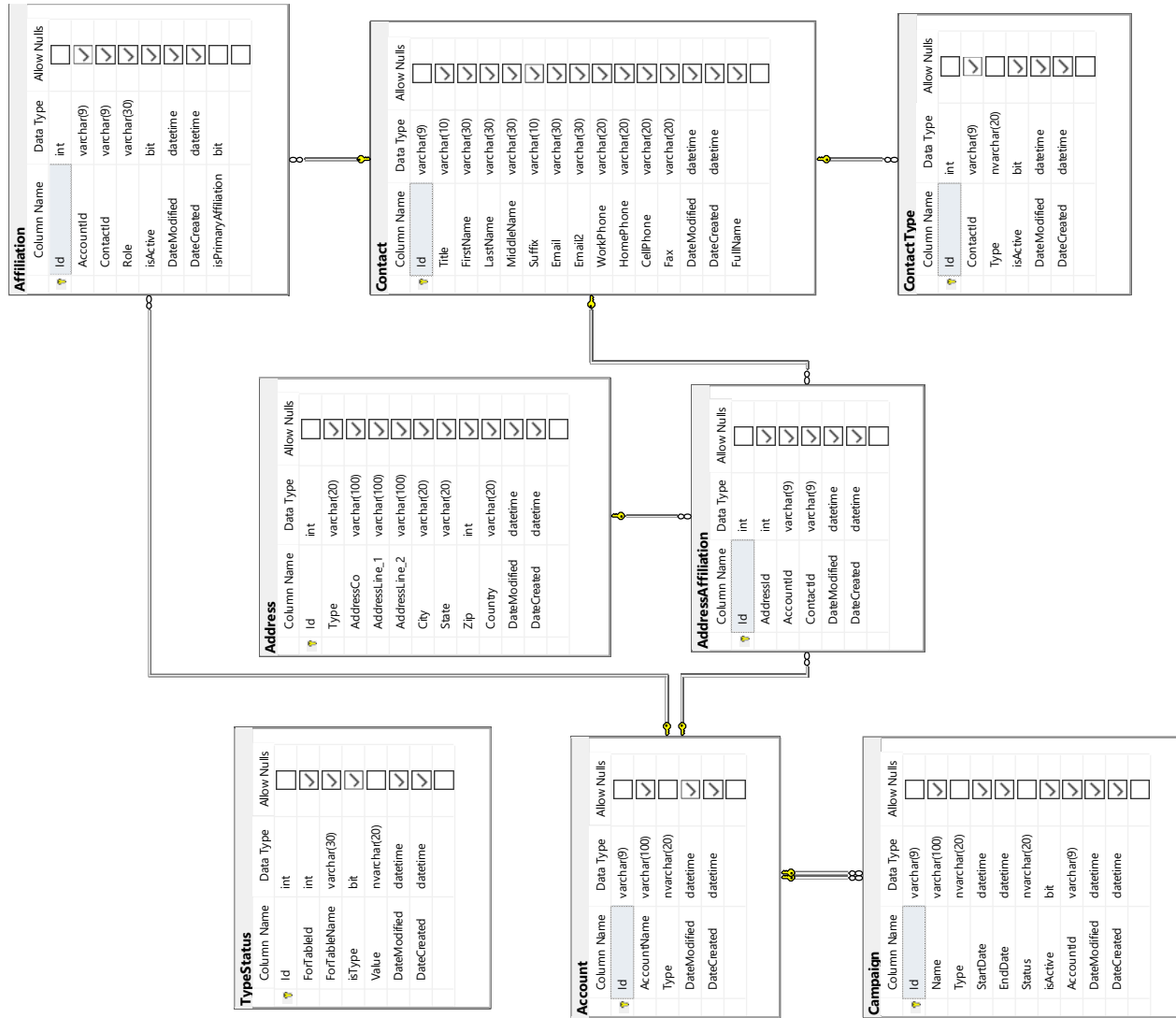
Process: Updates the skills for a job, creates new records in skillavail if the skills do not yet exist. Saves skill id. Adds record to jobdetail with jobid and skillids.

Output: None

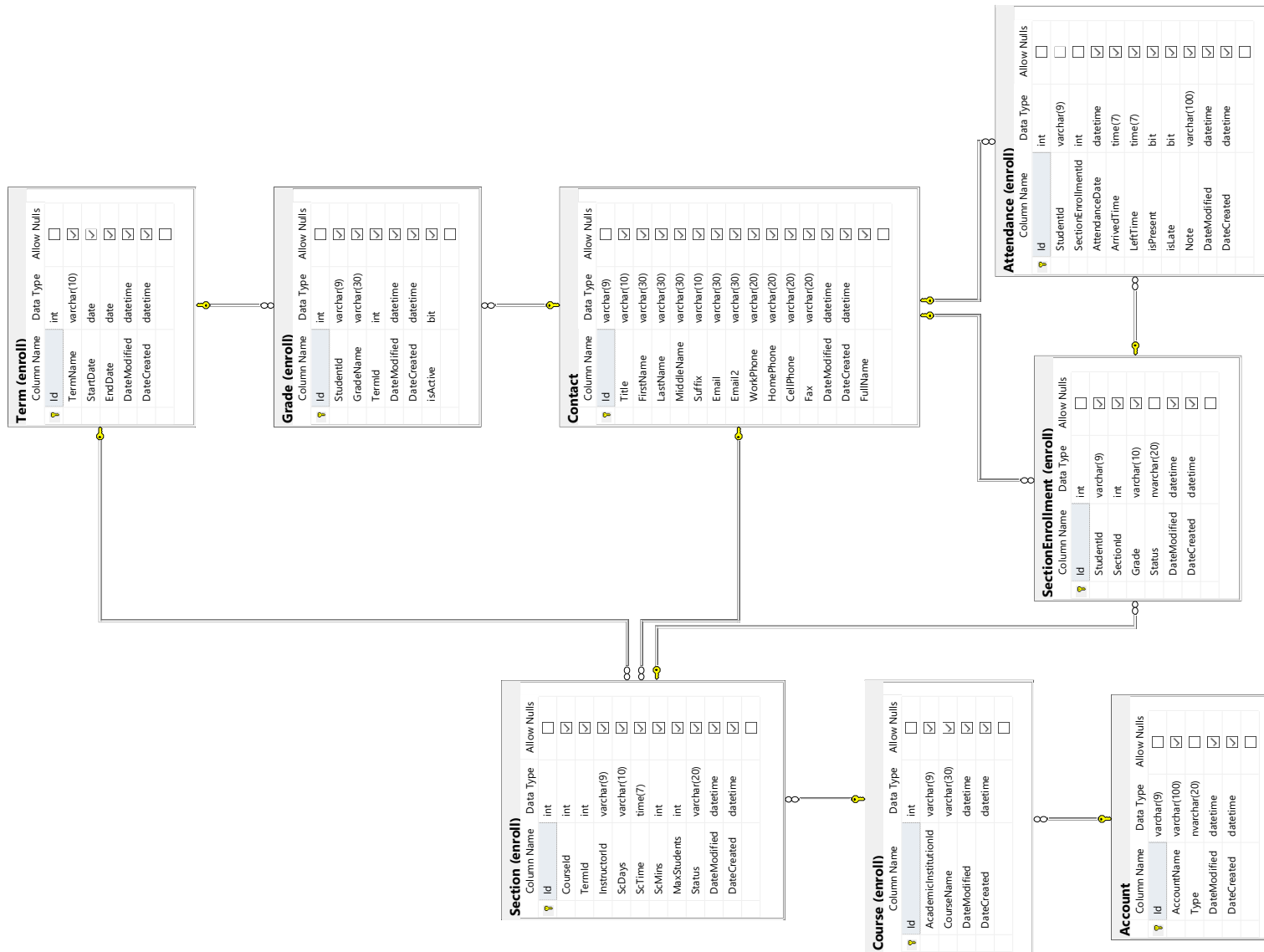
Error Handling: Calls usp_GetErrorInfo in try/catch block to display errors. Raises custom error if job does not exist.

Database Diagrams

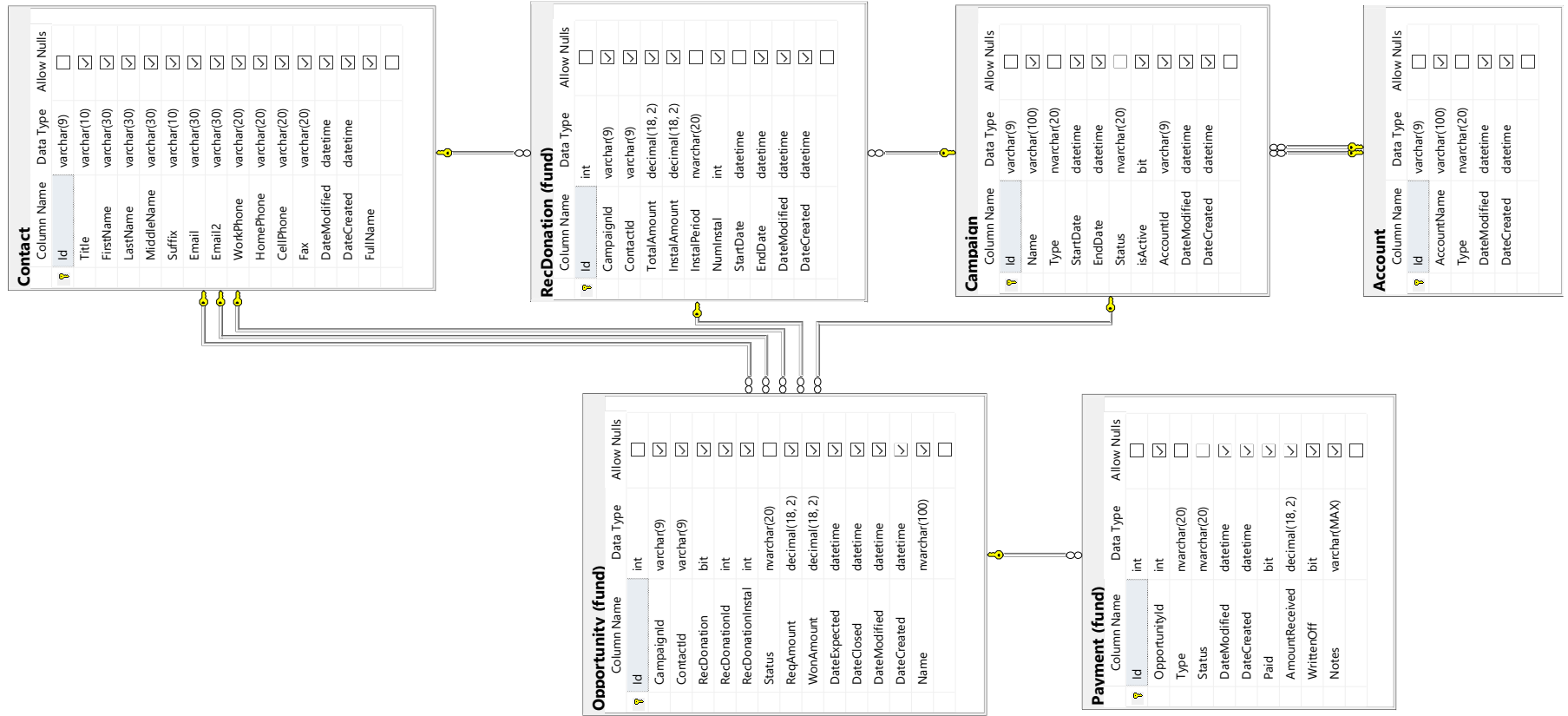
dbo Schema



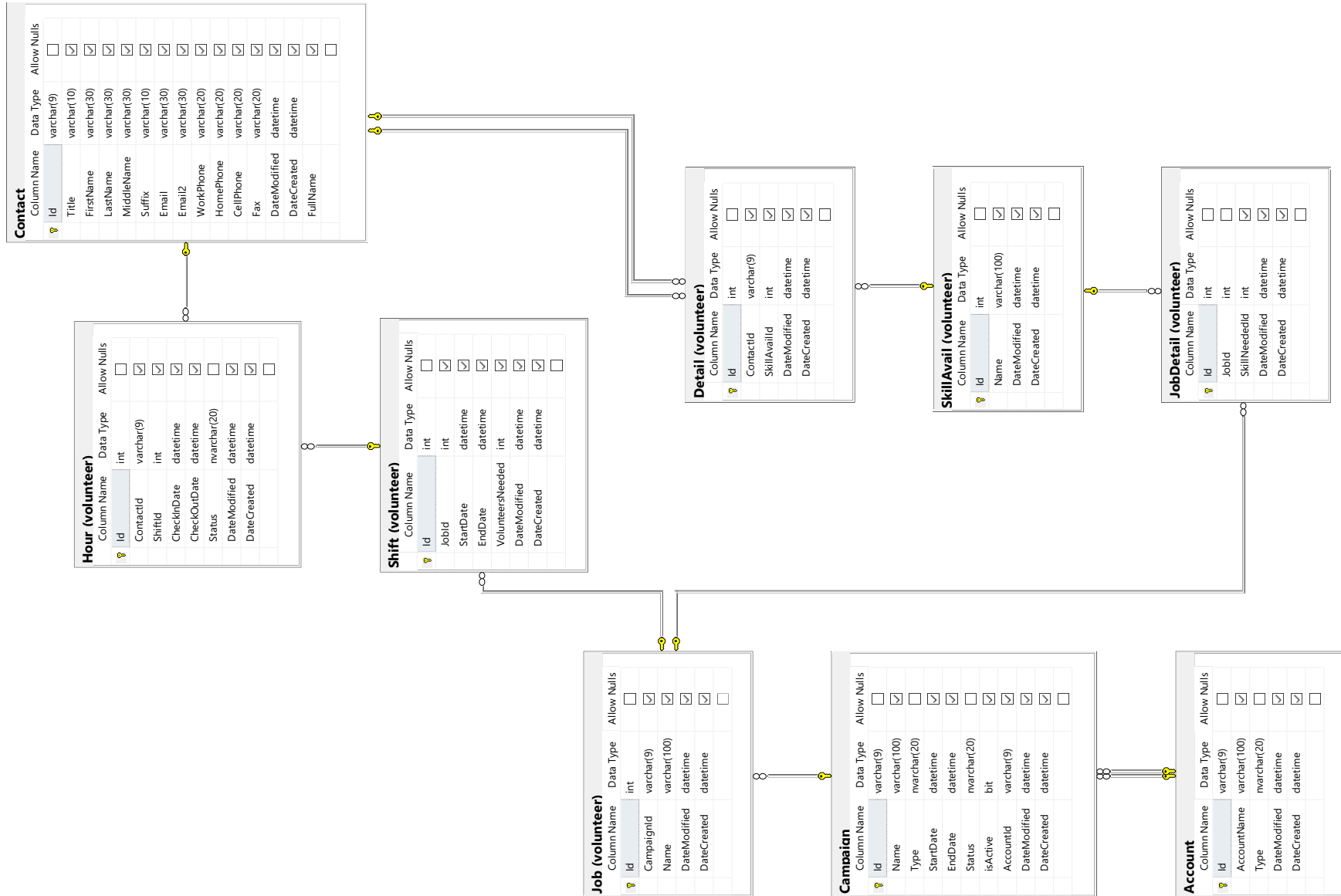
enroll Schema



fund Schema



volunteer Schema



full database diagram

