1. Introduction

1.1 Business Overview and Description

'Educate with Nutrition' is a Non-Governmental Organization (NGO) working towards abolishing malnutrition in school going, children. The NGO works in conjunction with the 'Midday Meal Scheme,' an initiative of the Government of India designed to provide a meal to every registered student in a school. The organization cooks and serves meals to school children and is committed to yield adequate nutritious food following the individual student's health fitness. The NGO operates with funds generated by donors and school donations. Various employees part of the NGO work towards the mission of the organization under different roles. Nutritionists are responsible for analyzing individual student's health fitness and recommending a suitable diet plan designed by them, promptly as and when the student's profile goes renewed. The staff category of employees possess duties of meal preparation and serving the right meal to a right student, whereas a Manager is responsible for managing the overall functioning of the organization which holds fund management and addressing the concerns raised by schools in case of anomalies.

1.2 Database Description

The database "Educate with Nutrition" is a central repository for information pertinent to the NGO - Educate with Nutrition, in support of a mission to serve nutritious food to school going children. The database is designed to include all the information required to aid the organization's smooth functioning. The database is modeled on the physical structure of the NGO in consideration of the processes and functionalities carried out by the organization. Some major business rules captured through database design are:

- The information about various schools associated with NGO, the details of each school are stored with a unique identifier. There are many students recorded in the database who study in one of the registered schools; many students learn at a school.
- A student falls under an age group, which is described by a description, maximum age, minimum age, and gender categories.
- Every student receives meal servings daily, tracked with an identifier, date and time of the serving, every serving is associated to a single student.
- A meal is comprised of various quantities of food items and served to students on a rotational basis specific to the diet requirements of an individual student.
- The organization consists of employees in different roles and is interested in storing details of the employees, and details specific to their positions.
- Nutritionists design different diet plans for every age group with other information like BMI range, BMI category, and various quantities recommended calories, protein, Iron, calcium, Vitamin A and Vitamin C, those which are essential for a child's healthy development.
- The organization requires to store information of every student's growth history.

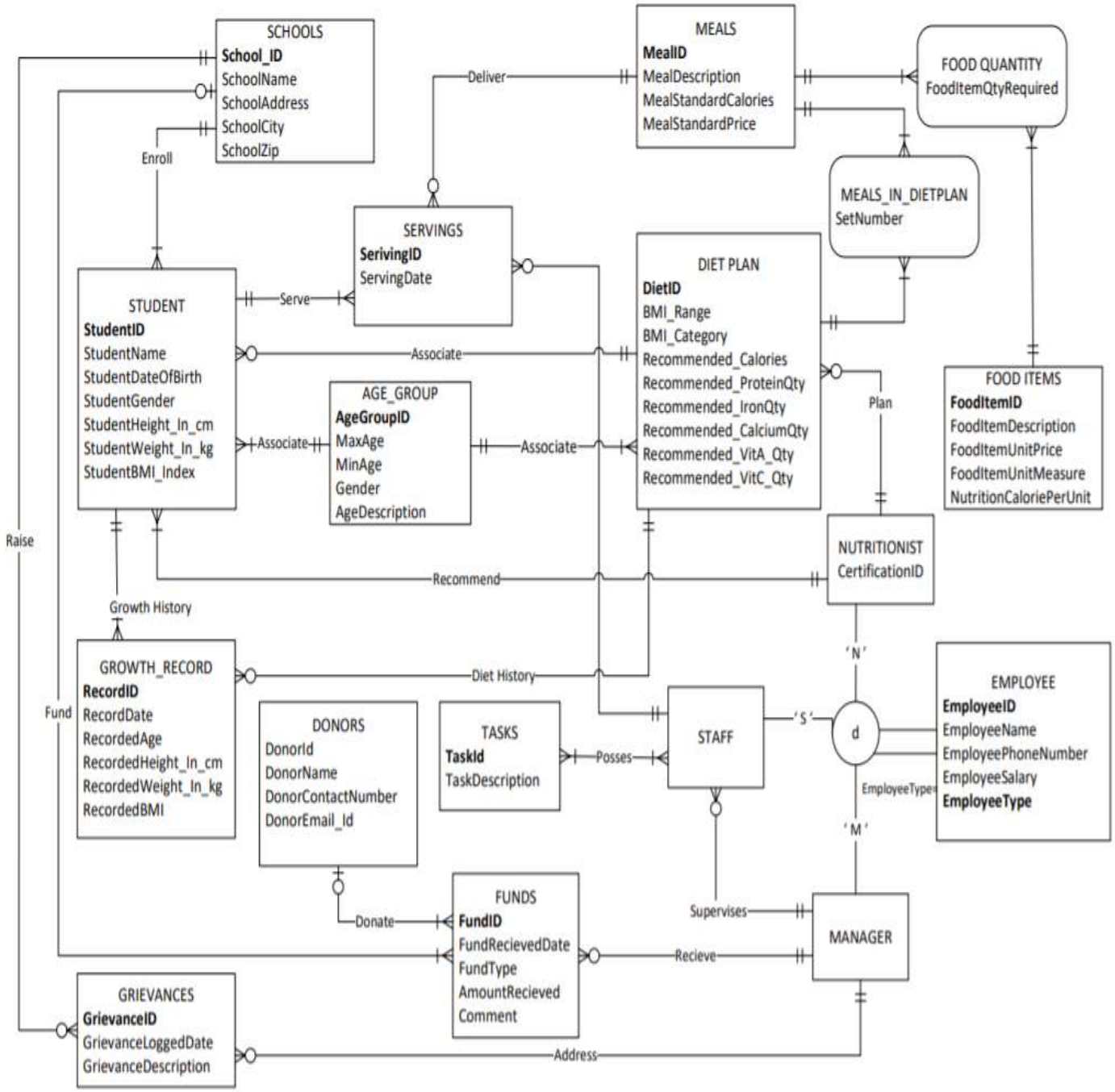
- The NGO generates funds from sources like donor funds and school donations which are tracked by the database. A school or donor make one or more donations to the organization's fund
- Information about the concerns raised by schools is stored as grievances and addressed by a manager.

The database is designed to apprehend comprehensive information on schools, students, different employees part of the organization, diet plans, meals which are a combination of various food items and funds received by the organization as major entities and others such as age groups, donors and grievance, those which have justified importance associated. Every object in the database is described by a diverse set of attributes to capture necessary details which helps access and reference data across the database. The detailed specifications of the attribute components of each entity are described in their corresponding data dictionary records.
Each table in the database is linked to one or more tables by carefully designed relationships. And, the tables are created with proper constraints to impose restrictions on the possible data in a record for a given table. Certain aspects of the organization share many characteristics; for example, students of certain age under a BMI category are served with meals associated with recommended diet plans; the tables are created to incorporate such common elements at all levels, hence increasing the efficiency of the data structure. One of the critical requirement to keep track of student's growth history, a trigger is designed to make the database effectively and accurately accommodate new values for the student's profile and at the same time move the existing data to the growth table and log the update.
This database project has been created with a goal to help the NGO to better function by organizing their data in a more accessible and interpretable form.

1.3 Value addition to the business by designing the database

The database will help the NGO organize its processes by providing a logical representation of various information required to administer and track its day to day activities. The database helps strengthen the organization's accessibility to data. In turn, it helps the end users share the data fast and effectively across the organization. The database will promote the integrity of organizations operations, the design of database with constraints and regulations increases data consistency and empowers the end user of the database to make informed decision that helps the NGO to support its mission of eliminating malnutrition by serving right meal to a student based on stored data of student nutrition requirement and adjust menus as necessary. In conclusion, the database adds immense value to the proper functioning of the organization and its progress towards accomplishing its objectives.

2. Entity Relationship Diagram



3. Logical Schema

EMPLOYEE	**EmployeeID**	EmployeeName	EmployeePhoneNumber	EmployeeSalary
	EmployeeType			
DONORS	**DonorID**	DonorName	DonorContactNumber	DonorEmail_Id
MANAGER	***M_EmployeeID***			
NUTRITIONIST	***N_EmployeeID***	CertificationID		
STAFF	***S_EmployeeID***	***SupervisorID***		
TASKS	**Task Id**	Task Description		
STAFF_TASKS	***S_EmployeeID***	***Task Id***		
DIET_PLAN	**DietID**	BMI_Range	BMI_Category	Recommended_Calories
	Recommended_ProteinQty	Recommended_IronQty	Recommended_CalciumQty	Recommended_VitA_Qty
	Recommended_VitC_Qty	***N_EmployeeID***	***AgeGroupID***	
FOODITEMS	**FoodItemID**	FoodItemDescription	FoodItemUnitPrice	FoodItemUnitMeasure
	NutritionCaloriePerUnit			
MEALS	**MealID**	MealDescription	MealStandardCalories	MealStandardPrice
FOOD_QUANTITY	***FoodItemID***	***MealID***	FoodItemQtyRequired	
MEALS_IN_DIETPLAN	***DietID***	***MealID***	SetNumber	
SERVINGS	**SerivingID**	ServingDate	***MealID***	***Packaging_StaffID***
	StudentID			
SCHOOLS	**School_ID**	SchoolName	SchoolAddress	SchoolCity
	SchoolZip			
STUDENT	**StudentID**	StudentName	StudentDateOfBirth	Student Gender
	StudentHeight_In_cm	StudentWeight_In_kg	***School_ID***	***DietID***
	AgeGroupID	***N_EmployeeID***		
AGE_GROUP	**AgeGroup ID**	MaxAge	MinAge	Gender
	AgeDescription			
GROWTH_RECORD	**RecordID**	RecordDate	RecordedAge	RecordedHeight_In_cm
	RecordedWeight_In_kg	RecordedBMI	***DietID***	***StudentID***
GRIEVANCES	**GrievanceID**	GrievanceLoggedDate	GrievanceDescription	***AddressingMGR_ID***
	School_ID			
FUNDS	**FundID**	FundRecievedDate	FundType	AmountRecieved
	Comment	***RecievingMGR_ID***	*DonorID*	*SchoolID*

4. Data Dictionary:

EMPLOYEE

Name	DataType	Constraints	Key	Description	Example Value
EmployeeID	bigint	> 0	PK	Unique identifier for an employee	4536
EmployeeName	nvarchar(100)			Name of an employee	Ram Rao
EmployeePhoneNumber	char (10)	Like REPLICATE ('[0-9]' ,10)		Contact number of an employee	9901299119
EmployeeSalary	decimal (9,2)	> 0.0		Annual Salary for an employee	95000.05
EmployeeType	char (1)	('N' , 'S' ,'M')		Discriminator for employee type, Nutritionist('N) or Staff ('S') or Manager('M')	N

DONORS

Name	DataType	Constraints	Key	Description	Example Value
DonorID	bigint	> 0	PK	Unique identifier for a donor	136
DonorName	nvarchar(100)			Name of a donor	Mohan Raj
DonorContactNumber	char (10)	Like REPLICATE ('[0-9]' ,10)		Contact number of a donor	9901299119
DonorEmail_Id	nvarchar(50)			Email Id of a donor	mohan12@yahoo.in

EMPLOYEE_MANAGER

Name	DataType	Constraints	Key	Description	Example Value
M_EmployeeID	bigint	>0	PK, FK	Unique identifier for a Manager employee	1122

EMPLOYEE_NUTRITIONIST

Name	DataType	Constraints	Key	Description	Example Value
N_EmployeeID	bigint	>0	PK, FK	Unique identifier for a Nutritionist employee	1182
CertificationID	nvarchar(25)			Certification number of the Nutritionist	ND1345

EMPLOYEE_STAFF

Name	DataType	Constraints	Key	Description	Example Value
S_EmployeeID	bigint	>0	PK, FK	Unique identifier for a Staff employee	1100
SupervisorID	bigint	>0	FK	Supervisor of a staff employee; unique identifier of a manager	1122

TASKS

Name	DataType	Constraints	Key	Description	Example Value
TaskID	int	>0	PK	Unique identifier for a task	12
TaskDescription	nvarchar(50)			Description of a task	Packaging of meals

STAFF_TASKS

Name	DataType	Constraints	Key	Description	Example Value
S_EmployeeID	bigint	>0	PK, FK	Staff associated with the task; unique identifer for a staff employee; composite identifier for a staff association to task	1100
Task Id	nvarchar(10)	>0	PK, FK	Task associated to a staff;unique identifier for a task;composite identifier for a staff association to task	PK12

DIET_PLAN

Name	DataType	Constraints	Key	Description	Example Value
DietID	int	>0	PK	Unique identifier for a diet plan	5
BMI_Range	nvarchar(25)			Specifies the BMI range for a certain BMI category	18.5 up to 25.0
BMI_Category	nvarchar(15)	('Underweight' ,'Normal','Overweight', 'Obese')		Specifies the BMI category	Underweight
Recommended_Calories	int	>0		Nutritional requirement in a diet plan	450
Recommended_ProteinQty	nvarchar(10)			Protien requirement in a diet plan	28.8 g
Recommended_IronQty	nvarchar(10)			Iron requirement in a diet plan	6.2 mg
Recommended_CalciumQty	nvarchar(10)			Calcium requirement in a diet plan	125.5 mg
Recommended_VitA_Qty	nvarchar(10)			Vitamin A requirement in a diet plan	256.6 mg
Recommended_VitC_Qty	nvarchar(10)			Vitamin C requirement in a diet plan	32.7 mg
N_EmployeeID	bigint	>0	FK	Nutritionist who designs the diet plan;Unique identifier for a Nutritionist employee	1182
AgeGroupID	int	>0	FK	Age group associated to a diet plan ; unique identifier for a age group	3

FOODITEMS

Name	DataType	Constraints	Key	Description	Example Value
FoodItemID	int	>0	PK	Unique Identifier for a food item	4
FoodItemDescription	nvarchar(50)			Description for a food item	Sprouts salad
FoodItemUnitPrice	decimal(5,2)			Price for unit quantity of a food item	7.25
FoodItemUnitMeasure	nvarchar(10)			Unit measurement of a food item	cup
NutritionCaloriePerUnit	int			Nutritional value of a food item in calories	115

MEALS

Name	DataType	Constraints	Key	Description	Example Value
MealID	int	>0	PK	Unique Identifier for a meal	3
MealDescription	nvarchar(50)			Description for a meal	Iron rich sprouts meal
MealStandardCalories	int			Nutritional value of a meal in calories	400
MealStandardPrice	decimal(5,2)			Cost of a meal	18.25

FOOD_QUANTITY

Name	DataType	Constraints	Key	Description	Example Value
FoodItemID	int	>0	PK,FK	Quantity of Food item associated to a meal; unique identifier for a food item ; Composite identifier for quantity of food item in a meal	4
MealID	int	>0	PK,FK	Meal consisting quantity of a food item ; unique identifier for a meal; Composite identifier for quantity of food item in a meal	3
FoodItemQtyRequired	int	>0		Quantity of a food item in a meal	2

MEALS_IN_DIETPLAN

Name	DataType	Constraints	Key	Description	Example Value
DietID	int	>0	PK,FK	Diet Plan associated to a meal ; unique identifier for a diet plan ; Composite identifier for a meal in diet	5
MealID	int	>0	PK,FK	Meal part of a diet plan ; unique identifier for a diet plan ; Composite identifier for a meal in diet	3
SetNumber	int			Groups Diet and Meal combination to a set	1

SERVINGS

Name	DataType	Constraints	Key	Description	Example Value
SerivingID	bigint	>0	PK	Unique identifier for a serving	1122337
ServingDate	datetime			Date and time of the meal packaged for a serving	12/25/2018 11:00AM
MealID	int	>0	FK	Meal served ; unique identifier for a meal	5
Packaging_StaffID	bigint	>0	FK	Staff employee who packs the meal ; unique identifier for staff employee	1100
StudentID	bigint	>0	FK	Student receiving the servings; unique identifier for a student	22334534

SCHOOLS

Name	DataType	Constraints	Key	Description	Example Value
School_ID	int	>0	PK	Unique identifier for a school	10
SchoolName	nvarchar(50)			Name of a school	St Johns School
SchoolAddress	nvarchar(50)			Address of a school	2250 Market St
SchoolCity	nvarchar(25)			City where a school is located	Bangalore
SchoolZip	char(6)			Zipcode of a school	560061

STUDENT

Name	DataType	Constraints	Key	Description	Example Value
StudentID	bigint	>0	PK	Unique identifier for a student	234557
StudentName	nvarchar(50)			Name of a student	Reena Sen
StudentDateOfBirth	date			Date of birth of a student	10/9/2009
StudentGender	char(6)	('Female' , 'Male')		Gender of a student	Female
StudentHeight_In_cm	decimal(5,2)	>0.0		Height of a student in centimeters	133.45
StudentWeight_In_kg	decimal(5,2)	>0.0		Weight of a student in Kilograms	28.5
School_ID	int	>0	FK	School in which the student is enrolled; unique identifier for a school	SJ1250
DietID	int	>0	FK	Diet plan recommended for a student; unique identifier for a diet plan	5
AgeGroupID	int	>0	FK	Age group associated to a diet plan ; unique identifier for a age group	3
N_EmployeeID	bigint	>0	FK	Emp of a meters	1500

AGE_GROUP

Name	DataType	Constraints	Key	Description	Example Value
AgeGroupID	int	>0	PK	Unique identifier for a age group	6
MaxAge	int	>0		Maximum age allowed in a group	4
MinAge	int	>0		Minimum age allowed in a group	2
Gender	char(6)	('Female' , 'Male')		Gender of a age group	Female
AgeDescription	nvarchar(25)			Description of a age group	Female aged between 2 and 3 yrs

GROWTH_RECORD

Name	DataType	Constraints	Key	Description	Example Value
RecordID	bigint	>0	PK	Unique identifier for a growth record	123456
RecordDate	datetime			Date and time of the growth record creation	10/25/2017 11:00AM
RecordedAge	nvarchar(25)			Age of a student at the time of growth record creation	5 yrs 2 months
RecordedHeight_In_cm	decimal(5,2)	>0.0		creation	113.45
RecordedWeight_In_kg	decimal(5,2)	>0.0		Weight of a student in Kilograms at the time of growth record creation	20.5
RecordedBMI	decimal(5,2)	>0.0		BMI of a student at the time of growth record creation	20.25
DietID	int	>0	FK	Diet plan recorded for a student in growth record ; unique identifier for a diet plan	4
StudentID	bigint	>0	FK	Student associated to the growth record; unique identifier for a student	234557

GRIEVANCES

Name	DataType	Constraints	Key	Description	Example Value
GrievanceID	bigint	>0	PK	Unique identifier for a grievance raised	12234
GrievanceLoggedDate	datetime			Date time when a grievance was raised	10/25/2017 11:00AM
GrievanceDescription	nvarchar(100)			Description of a grievance	Meal missed for 2 students
AddressingMGR_ID	bigint	>0	FK	Manager adressing the grievance; unique identifier for a manager employee	11203
School_ID	int	>0	FK	School raising the grievance ; unique Identifier for a school	SJ1250

FUNDS

Name	DataType	Constraints	Key	Description	Example Value
FundID	bigint	>0	PK	Unique Identifier for a fund	102345
FundRecievedDate	datetime			Date and time of a fund received	10/25/2017 11:00AM
FundType	char(20)	('Donor Fund', 'School Fund','Anonymous')		Discriminator for the fund type 'Donor Fund' for funds received by donors, 'School Fund' for funds recieved from schools and 'Anonymous' for unkown donations.	Donor Fund
AmountRecieved	decimal(10,2)	>0.0		Amount received as fund	2000
DonorID	bigint		FK	Unique identifier for a donor	136
SchoolID	nvarchar(15)		FK	Unique identifier for a school	SJ1250
Comment	nvarchar(25)			The form of the fund received	Check-22302
RecievingMGR_ID	bigint	>0	FK	Acknowledging manager for a fund ; unique identifier for a manger employee	11205

4. SQL Statements for table creation

1. Employee_T

```
CREATE TABLE Employee_T
(EmployeeID bigint not null CHECK (EmployeeID > 0),
EmployeeName nvarchar(100)  ,
EmployeePhoneNumber char(10)CHECK (EmployeePhoneNumber Like REPLICATE ('[0-9]' ,10)),
EmployeeSalary decimal(9,2) CHECK (EmployeeSalary>0.0),
EmployeeType char(1) CHECK (EmployeeType IN ('S','N','M')) not null,
CONSTRAINT Employee_PK PRIMARY KEY (EmployeeID))
```

2. Donor_T

```
CREATE TABLE Donor_T
( DonorID bigint not null CHECK (DonorID >0),
DonorName nvarchar(100)  ,
DonorContactNumber char(10)CHECK (DonorContactNumber Like REPLICATE ('[0-9]' ,10)),
DonorEmail_Id nvarchar(50),
CONSTRAINT Donor_PK PRIMARY KEY (DonorID))
```

3. Employee_Manager_T

```
CREATE TABLE Employee_Manager_T
(M_EmployeeID bigint not null CHECK (M_EmployeeID > 0)
CONSTRAINT Employee_Manager_PK PRIMARY KEY (M_EmployeeID),
CONSTRAINT Employee_Manager_FK FOREIGN KEY (M_EmployeeID) REFERENCES Employee_T
(EmployeeID))
```

4. Employee_Nutritionist_T

```
CREATE TABLE Employee_Nutritionist_T
(N_EmployeeID bigint not null CHECK (N_EmployeeID > 0),
CertificationID nvarchar(25) not null,
CONSTRAINT Employee_Nutritionist_PK PRIMARY KEY (N_EmployeeID),
CONSTRAINT Employee_Nutritionist_FK FOREIGN KEY (N_EmployeeID) REFERENCES
Employee_T(EmployeeID))
```

5. Employee_Staff_T

```
CREATE TABLE Employee_Staff_T
(S_EmployeeID bigint not null CHECK (S_EmployeeID > 0),
SupervisorID bigint not null CHECK (SupervisorID > 0)
CONSTRAINT Employee_Staff_PK PRIMARY KEY (S_EmployeeID),
CONSTRAINT Employee_Staff_FK FOREIGN KEY (S_EmployeeID) REFERENCES Employee_T (EmployeeID),
CONSTRAINT Employee_Staff_FK2 FOREIGN KEY (SupervisorID) REFERENCES
Employee_Manager_T(M_EmployeeID))
```

6. Task_T

```
CREATE TABLE Task_T
(TaskID int not null CHECK (TaskID > 0),
TaskDescription nvarchar(50),
CONSTRAINT Task_PK PRIMARY KEY(TaskID))
```

7. Staff_Task_T

```
CREATE TABLE Staff_Task_T
(TaskID int not null CHECK (TaskID > 0),
S_EmployeeID bigint not null CHECK (S_EmployeeID > 0),
CONSTRAINT Staff_Task_PK1 PRIMARY KEY (TaskID,S_EmployeeID),
CONSTRAINT Staff_Task_FK1 FOREIGN KEY (S_EmployeeID) REFERENCES Employee_T(EmployeeID),
CONSTRAINT Staff_Task_FK2 FOREIGN KEY (TaskID) REFERENCES Task_T(TaskID))
```

8. FoodItems_T

```
CREATE TABLE FoodItems_T
(FoodItemID int not null CHECK (FoodItemID > 0),
FoodItemDescription nvarchar(50),
FoodItemUnitPrice decimal(5,2),
FoodItemUnitMeasure nvarchar(10),
NutritionCaloriePerUnit int,
CONSTRAINT FoodItems_PK PRIMARY KEY (FoodItemID))
```

9. Schools_T

```
CREATE TABLE Schools_T
(School_ID int not null CHECK (School_ID > 0),
SchoolName nvarchar(50),
SchoolAddress nvarchar(50),
SchoolCity nvarchar(25),
SchoolZip char(6),
CONSTRAINT Schools_PK PRIMARY KEY (School_ID))
```

10. Age_Group_T

```
CREATE TABLE Age_Group_T
(AgeGroupID int not null CHECK (AgeGroupID > 0),
MaxAge    int not null CHECK (MaxAge > 0),
MinAge    int not null CHECK (MinAge > 0),
Gender    char(6) CHECK(Gender IN('Female' , 'Male')),
AgeDescription nvarchar(50),
CONSTRAINT AgeGroup_PK PRIMARY KEY (AgeGroupID))
```

11. Funds_T

```
CREATE TABLE Funds_T
(FundID   bigint not null CHECK (FundID > 0),
FundRecievedDate datetime,
FundType char(20) CHECK(FundType IN('Donor Fund', 'School Fund','Anonymous' )),
AmountRecieved decimal(10,2) CHECK(AmountRecieved >0.0),
DonorID bigint,
SchoolID int,
Comment nvarchar(25),
RecievingMGR_ID bigint not null CHECK (RecievingMGR_ID>0)
CONSTRAINT Funds_PK PRIMARY KEY (FundID),
CONSTRAINT Funds_FK1 FOREIGN KEY (DonorID) REFERENCES Donor_T(DonorID),
CONSTRAINT Funds_FK2 FOREIGN KEY (SchoolID) REFERENCES Schools_T(School_ID),
CONSTRAINT Funds_FK3 FOREIGN KEY (RecievingMGR_ID) REFERENCES
Employee_Manager_T(M_EmployeeID))
```

12. Grievances_T

```
CREATE TABLE Grievances_T
(GrievanceID    bigint not null CHECK (GrievanceID > 0),
GrievanceLoggedDate datetime,
GrievanceDescription nvarchar(100),
AddressingMGR_ID bigint not null CHECK (AddressingMGR_ID > 0) ,
SchoolID int not null CHECK (SchoolID > 0),
CONSTRAINT Grievances_PK PRIMARY KEY (GrievanceID),
CONSTRAINT Grievances_FK1 FOREIGN KEY (SchoolID) REFERENCES Schools_T(School_ID),
CONSTRAINT Grievances_FK2 FOREIGN KEY (AddressingMGR_ID) REFERENCES
Employee_Manager_T(M_EmployeeID))
```

13. Meals_T

```
CREATE TABLE Meals_T
( MealID int not null CHECK (MealID >0),
MealDescription nvarchar(50)  ,
MealStandardCalories int ,
MealStandardPrice decimal(5,2),
CONSTRAINT Meals_PK PRIMARY KEY (MealID))
```

14. Food_Quantity_T

```
CREATE TABLE Food_Quantity_T
( FoodItemID int not null CHECK (FoodItemID >0),
MealID int not null CHECK (MealID >0),
FoodItemQtyRequired int not null CHECK (FoodItemQtyRequired >0)
CONSTRAINT FoodQuantity_PK PRIMARY KEY (MealID,FoodItemID)
CONSTRAINT FoodQuantity_FK1 FOREIGN KEY (MealID) REFERENCES Meals_T(MealID),
CONSTRAINT FoodQuantity_FK2 FOREIGN KEY (FoodItemID) REFERENCES FoodItems_T(FoodItemID))
```

15. Diet_Plan_T

```
CREATE TABLE Diet_Plan_T
( DietID int not null CHECK (DietID >0),
BMI_Range nvarchar(25),
BMI_Category nvarchar(15)
CHECK(BMI_Category IN ('Underweight' ,'Normal','Overweight', 'Obese')),
Recommended_Calories int CHECK (Recommended_Calories>0),
Recommended_ProteinQty nvarchar(10),
Recommended_IronQty    nvarchar(10),
Recommended_CalciumQty nvarchar(10),
Recommended_VitA_Qty nvarchar(10),
Recommended_VitC_Qty nvarchar(10),
N_EmployeeID bigint not null CHECK(N_EmployeeID>0),
AgeGroupID      int not null CHECK(AgeGroupID>0),
CONSTRAINT DietPlan_PK PRIMARY KEY (DietID),
CONSTRAINT DietPlan_FK1 FOREIGN KEY (N_EmployeeID) REFERENCES
Employee_Nutritionist_T(N_EmployeeID),
CONSTRAINT DietPlan_FK2 FOREIGN KEY (AgeGroupID) REFERENCES Age_Group_T(AgeGroupID))
```

16. Student_T

```
CREATE TABLE Student_T
( StudentID      bigint not null CHECK (StudentID >0),
StudentName      nvarchar(50),
StudentDateOfBirth      date,
StudentGender char(6) CHECK(StudentGender IN ('Female' ,'Male')),
StudentHeight_In_cm     decimal(5,2)  CHECK (StudentHeight_In_cm >0.0),
StudentWeight_In_kg     decimal(5,2)CHECK (StudentWeight_In_kg >0.0),
School_ID int not null CHECK (School_ID > 0),
DietID    int    not null CHECK (DietID > 0),
AgeGroupID       int not null CHECK (AgeGroupID > 0),
N_EmployeeID bigint    not null CHECK (N_EmployeeID > 0),
CONSTRAINT Student_PK PRIMARY KEY (StudentID),
CONSTRAINT Student_FK1 FOREIGN KEY (School_ID) REFERENCES Schools_T(School_ID),
CONSTRAINT Student_FK2 FOREIGN KEY (DietID) REFERENCES Diet_Plan_T(DietID),
CONSTRAINT Student_FK3 FOREIGN KEY (N_EmployeeID) REFERENCES
Employee_Nutritionist_T(N_EmployeeID),
CONSTRAINT Student_FK4 FOREIGN KEY (AgeGroupID) REFERENCES Age_Group_T(AgeGroupID))
```

17. Servings_T

```
CREATE TABLE Servings_T
( SerivingID bigint not null CHECK (SerivingID >0),
ServingDate      datetime,
MealID int not null CHECK (MealID > 0),
Packaging_StaffID bigint not null CHECK (Packaging_StaffID > 0),
StudentID bigint not null CHECK (StudentID > 0),
CONSTRAINT Serving_PK PRIMARY KEY (SerivingID),
CONSTRAINT Serving_FK1 FOREIGN KEY (MealID) REFERENCES Meals_T(MealID),
CONSTRAINT Serving_FK2 FOREIGN KEY (StudentID) REFERENCES Student_T(StudentID),
CONSTRAINT Serving_FK3 FOREIGN KEY (Packaging_StaffID) REFERENCES
Employee_Staff_T(S_EmployeeID));
```

18. Growth_Record_T

```
CREATE TABLE Growth_Record_T
( RecordID bigint not null CHECK (RecordID >0),
RecordDate datetime,
RecordedAge nvarchar(25),
RecordedHeight_In_cm decimal(5,2)   CHECK(RecordedHeight_In_cm >0.0),
RecordedWeight_In_kg decimal(5,2)   CHECK(RecordedWeight_In_kg >0.0),
RecordedBMI      decimal(5,2) CHECK (RecordedBMI > 0.0),
DietID    int    not null CHECK (DietID > 0),
StudentID bigint not null CHECK (StudentID > 0),
CONSTRAINT Growth_Record_PK PRIMARY KEY (RecordID),
CONSTRAINT Growth_Recordg_FK1 FOREIGN KEY (DietID) REFERENCES Diet_Plan_T(DietID),
CONSTRAINT Growth_Record_FK2 FOREIGN KEY (StudentID) REFERENCES Student_T(StudentID));
```

19. Meals_In_DietPlan_T

```
CREATE TABLE Meals_In_DietPlan_T
(DietID   int not null CHECK (DietID > 0),
MealID    int not null CHECK (MealID > 0),
SetNumber int ,
CONSTRAINT Meals_In_DietPlan_PK PRIMARY KEY (DietID,MealID),
CONSTRAINT Meals_In_DietPlan_FK1 FOREIGN KEY (DietID) REFERENCES Diet_Plan_T(DietID),
CONSTRAINT Meals_In_DietPlan_FK2 FOREIGN KEY (MealID) REFERENCES Meals_T(MealID));
```

6. Views and Procedures

6.1 Materialised View 1

View user's role in the Organization: Manager responsible to track expenditure and funds to the organization.

Justification for Usefulness of the View: There are many schools registered with the NGO to provide nutritious meals to its students and for which the schools fund the organization some amount every month. Every meal served has a cost associated with it, and hence there is an expenditure incurred to serve meals to the students. This view 'ExpenseDetails_View' is designed to generate a report for a manager, on the total cost concerning each school and to contrast the corresponding fundings received from them. This data helps the manager to well organize the fundings from different sources, and support the organization to operate better. The report from this view is expected to be requested at the beginning of every month to analyze the information from the previous month, and the view is designed accordingly.

SQL Statements for View and Procedure :

```
CREATE Table ExpenseDetails_View
( School_ID int not null,
SchoolName nvarchar(50),
Total_Expenditure decimal(7,2),
FundsRecieved decimal (7,2) )

create procedure RefreshExpenseDetails_View as
delete from ExpenseDetails_View
insert into ExpenseDetails_View

Select schools_T.School_ID,Schools_T.SchoolName,
 Sum( Meals_T.MealStandardPrice) as Total_Expenditure  ,derivedtable.TotalFund
from Student_T , Servings_T, Meals_T , Schools_T ,
(Select Funds_T.SchoolID, sum(AmountRecieved) as TotalFund    from Funds_T
where FundType like'School%' and month(FundRecievedDate) = month(GETDATE())-1
Group by SchoolID ) as derivedtable
where Student_T.StudentID= Servings_T.StudentID and Servings_T.MealID=Meals_T.MealID
and Student_T.School_ID=schools_T.School_ID and schools_T.School_ID=
derivedtable.SchoolID
and month(ServingDate) = month(GETDATE())-1
group by schools_T.School_ID, Schools_T.SchoolName ,derivedtable.TotalFund

execute RefreshExpenseDetails_View
Select * from ExpenseDetails_View
```

Results | Messages

	School_ID	SchoolName	Total_Expenditure	FundsRecieved
1	13	St Jhons School	124.50	3700.00
2	14	Achala Vidya Mandir	118.00	2701.00
3	15	Gurukulam School	67.00	3602.00
4	16	Clarence Public School	62.50	3703.00
5	17	Lawrence School	62.50	3504.00
6	18	St Marks Public School	62.50	3005.00
7	19	Gem School	62.00	3700.00
8	20	Baldwin Public School	61.00	2707.00

6.2 Materialised View 2

View user's role in the Organization: Staff member with responsibility of packaging the meals.

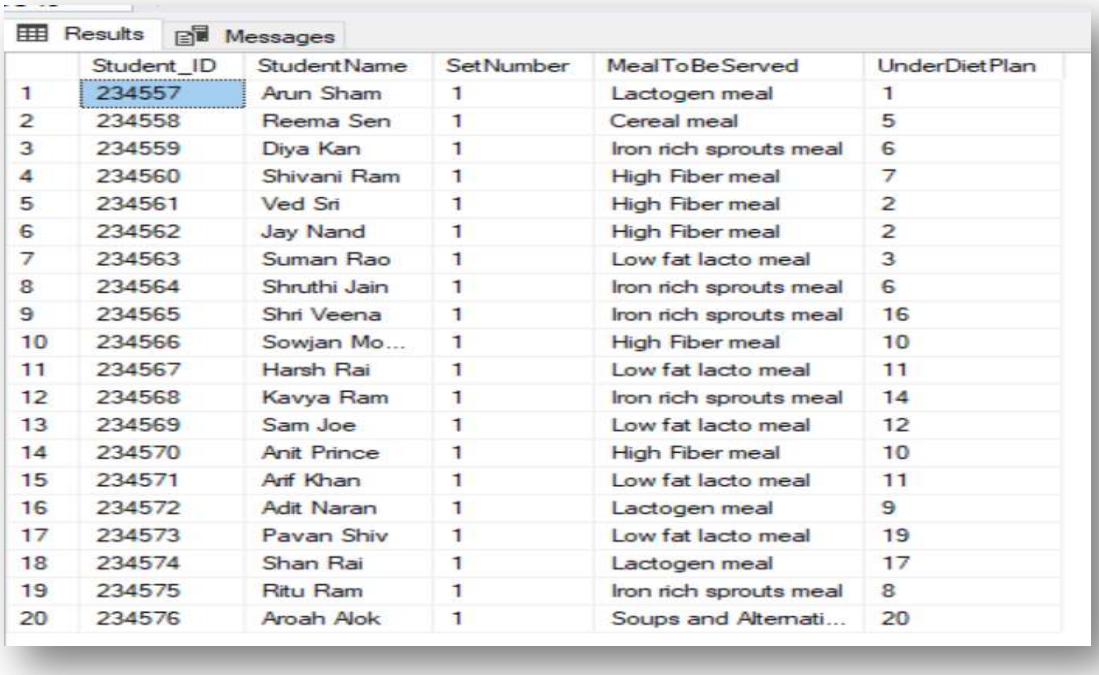
Justification for Usefulness of the View: The organization's daily activities include meal packaging to serve the students. Students are under different diet plan and are to be served distinctive meals under a specific diet plan. To facilitate the way toward turning different meals to Students the combination of a diet and meal routine are gathered under different sets. Every day, a set number is chosen, to serve the relating meal under the respective diet plan. Every serving is associated to a specific student, and there is a staff who is responsible for packaging the meals and labeling it to that particular student. This view 'MealPlanner_View' is designed to help the staff member to know which meal a student has to receive on a given day for a determined set number.

SQL Statements for View and Procedure :

```
CREATE Table MealPlanner_View
( Student_ID bigint not null,
StudentName nvarchar(50),
SetNumber int ,
MealToBeServed  nvarchar(50),
UnderDietPlan int )

create procedure RefreshMealPlanner_View as
delete from MealPlanner_View
insert into MealPlanner_View

select Student_T.StudentID, Student_T.StudentName,Meals_In_DietPlan_T.SetNumber,
Meals_T.MealDescription  ,Student_T.DietID
from Student_T, Meals_T, Meals_In_DietPlan_T
where Student_T.DietID = Meals_In_DietPlan_T.DietID
and Meals_T.MealID = Meals_In_DietPlan_T.MealID
group by Student_T.StudentID, Student_T.StudentName,
Meals_T.MealDescription,Student_T.DietID ,SetNumber

execute RefreshMealPlanner_View

Select * from MealPlanner_View
```

[This View , requires the User to query against the view with a where clause to specify the required SetNumber as below]

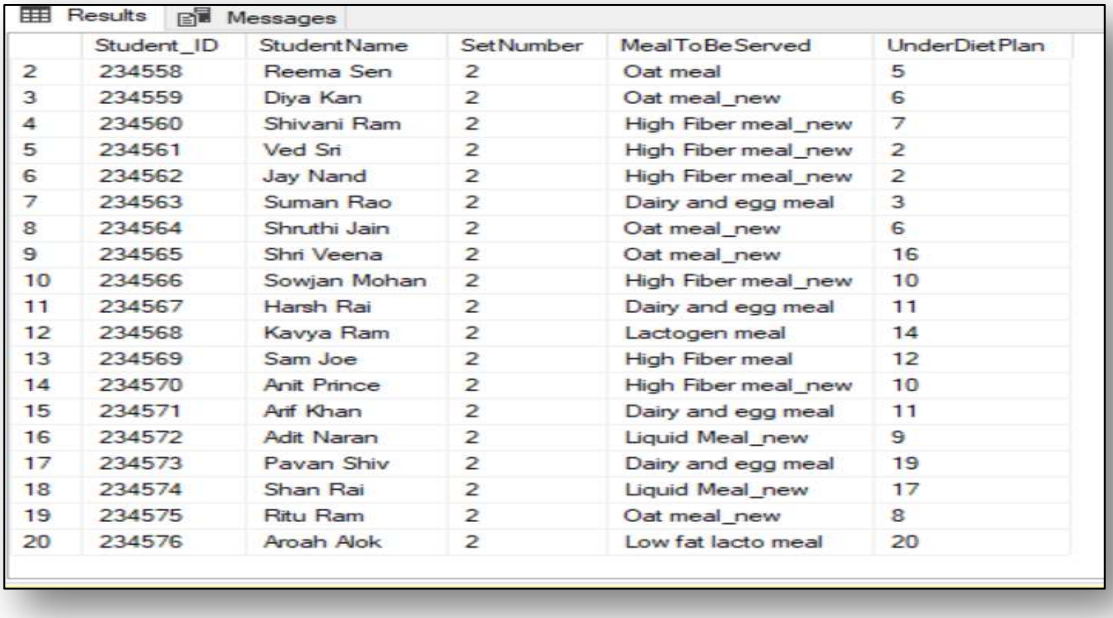
```
Select * from MealPlanner_View where SetNumber=2 -Filter required SetNumber
```



Results | Messages

	Student_ID	StudentName	SetNumber	MealToBeServed	UnderDietPlan
1	234557	Arun Sham	1	Lactogen meal	1
2	234558	Reema Sen	1	Cereal meal	5
3	234559	Diya Kan	1	Iron rich sprouts meal	6
4	234560	Shivani Ram	1	High Fiber meal	7
5	234561	Ved Sri	1	High Fiber meal	2
6	234562	Jay Nand	1	High Fiber meal	2
7	234563	Suman Rao	1	Low fat lacto meal	3
8	234564	Shruthi Jain	1	Iron rich sprouts meal	6
9	234565	Shri Veena	1	Iron rich sprouts meal	16
10	234566	Sowjan Mo...	1	High Fiber meal	10
11	234567	Harsh Rai	1	Low fat lacto meal	11
12	234568	Kavya Ram	1	Iron rich sprouts meal	14
13	234569	Sam Joe	1	Low fat lacto meal	12
14	234570	Anit Prince	1	High Fiber meal	10
15	234571	Arif Khan	1	Low fat lacto meal	11
16	234572	Adit Naran	1	Lactogen meal	9
17	234573	Pavan Shiv	1	Low fat lacto meal	19
18	234574	Shan Rai	1	Lactogen meal	17
19	234575	Ritu Ram	1	Iron rich sprouts meal	8
20	234576	Aroah Alok	1	Soups and Alternati...	20

```
Select * from MealPlanner_View where SetNumber=2 -Filter required SetNumber
```



Results | Messages

	Student_ID	StudentName	SetNumber	MealToBeServed	UnderDietPlan
2	234558	Reema Sen	2	Oat meal	5
3	234559	Diya Kan	2	Oat meal_new	6
4	234560	Shivani Ram	2	High Fiber meal_new	7
5	234561	Ved Sri	2	High Fiber meal_new	2
6	234562	Jay Nand	2	High Fiber meal_new	2
7	234563	Suman Rao	2	Dairy and egg meal	3
8	234564	Shruthi Jain	2	Oat meal_new	6
9	234565	Shri Veena	2	Oat meal_new	16
10	234566	Sowjan Mohan	2	High Fiber meal_new	10
11	234567	Harsh Rai	2	Dairy and egg meal	11
12	234568	Kavya Ram	2	Lactogen meal	14
13	234569	Sam Joe	2	High Fiber meal	12
14	234570	Anit Prince	2	High Fiber meal_new	10
15	234571	Arif Khan	2	Dairy and egg meal	11
16	234572	Adit Naran	2	Liquid Meal_new	9
17	234573	Pavan Shiv	2	Dairy and egg meal	19
18	234574	Shan Rai	2	Liquid Meal_new	17
19	234575	Ritu Ram	2	Oat meal_new	8
20	234576	Aroah Alok	2	Low fat lacto meal	20

6.3 Materialised View 3

View user's role in the Organization: Nutritionist associated with the organization.

Justification for Usefulness of the View: The organization's primary motto to abolish malnutrition in school children by providing a nutritious meal as per the individual student's nutritional requirement; therefore the nutritionists associated with the NGO are responsible for recommending a diet to each student and to trace the average calories a student is regularly getting. This view 'AvgCaloriesServed_View' is designed to provide the Nutritionists the average calories served to an individual student in contrast to the recommended calories through the suggested diet plan. In the case of a higher variance, the Nutritionist can take further actions to change the diet plan for a student. The focal point of this view is to generate average calories served concerning most recent data; hence the view is designed to consider data not more than six months before the date of querying the view.

SQL Statements for View and Procedure :

```
CREATE Table AvgCaloriesServed_View
( Student_ID bigint not null,
StudentName nvarchar(50),
AvgCaloriesServed int ,
Recommended_Calories int )

create procedure RefreshAvgCaloriesServed_View as
delete from AvgCaloriesServed_View
insert into AvgCaloriesServed_View

select Student_T.StudentID, Student_T.StudentName,
(sum(Meals_T.MealStandardCalories) / count(Servings_T.SerivingID)) ,
Diet_Plan_T.Recommended_Calories
from Student_T, Meals_T, Servings_T, Diet_Plan_T
where Student_T.DietID = Diet_Plan_T.DietID
and Meals_T.MealID = Servings_T.MealID
and Servings_T.StudentID = Student_T.StudentID
and month(ServingDate) > month(Getdate())-6
group by Student_T.StudentID, Student_T.StudentName,Diet_Plan_T.Recommended_Calories

execute RefreshAvgCaloriesServed_View

Select * from AvgCaloriesServed_View
```

Results | Messages

	Student_ID	StudentName	AvgCaloriesServed	Recommended_Calories
1	234558	Reema Sen	323	300
2	234557	Arun Sham	360	350
3	234559	Diya Kan	376	375
4	234564	Shruthi Jain	381	375
5	234565	Shri Veena	385	380
6	234560	Shivani Ram	403	400
7	234561	Ved Sri	403	400
8	234562	Jay Nand	403	400
9	234566	Sowjan Mohan	401	400
10	234563	Suman Rao	400	450

6.4 Materialised View 4

View user's role in the Organization: Staff responsible for planning and supervising cooking activities.

Justification for Usefulness of the View: The organization's day to day activities includes meal preparations to serve the students. Students are under various diet plans and are to be served different meals under a particular diet plan. To ease the process of rotating different meals to students the combination of a diet and meal is grouped under different sets. Daily, a set number is decided, to serve the corresponding meal under the respective diet plan. This view 'FoodPreparationPlanner_View' is designed to help the staff responsible for managing cooking activity to receive the required count of different food items necessary to package the meals planned to serve on a given day under the decided SetNumber. The count received through this view aids the cooking activity by providing an accurate quantity of a food item required and makes the process very organized and helps avoid food shortage or wasteasge by an inappropriate measure for food preparation.

SQL Statements for View and Procedure :

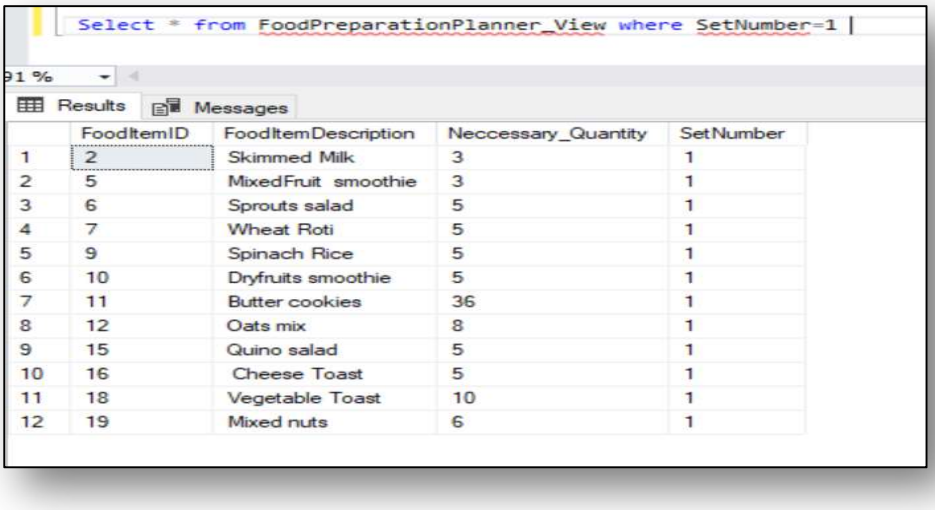
```
CREATE Table FoodPreparationPlanner_View
( FoodItemID int not null,
FoodItemDescription nvarchar(50),
Neccessary_Quantity int ,
SetNumber int )

create procedure RefreshFoodPreparationPlanner_View as
delete from FoodPreparationPlanner_View
insert into FoodPreparationPlanner_View

Select  FoodItems_T.FoodItemID,FoodItemDescription,
Sum( FoodItemQtyRequired * derivedtable.Requirement) as Neccessary_Quantity
,derivedtable.SetNumber
from Meals_T , Food_Quantity_T , FoodItems_T ,
(Select  Meals_In_DietPlan_T.MealID,SetNumber, Count(Meals_In_DietPlan_T.MealID) as
Requirement From Student_T , Meals_In_DietPlan_T
where Student_T.DietID = Meals_In_DietPlan_T.DietID and SetNumber =1 ---Decides the plan
Group by Meals_In_DietPlan_T.MealID,SetNumber) as derivedtable
where Meals_T.MealID= Food_Quantity_T.MealID and
Food_Quantity_T.FoodItemID=FoodItems_T.FoodItemID and
derivedtable.MealID=Food_Quantity_T.MealID
Group By FoodItemDescription  , FoodItems_T.FoodItemID,derivedtable.SetNumber
UNION
Select  FoodItems_T.FoodItemID,FoodItemDescription,
Sum( FoodItemQtyRequired * derivedtable.Requirement) as Neccessary_Quantity
,derivedtable.SetNumber
from Meals_T , Food_Quantity_T , FoodItems_T ,
(Select  Meals_In_DietPlan_T.MealID,SetNumber, Count(Meals_In_DietPlan_T.MealID) as
Requirement From Student_T , Meals_In_DietPlan_T
where Student_T.DietID = Meals_In_DietPlan_T.DietID and SetNumber =2 ---Decides the plan
Group by Meals_In_DietPlan_T.MealID,SetNumber) as derivedtable
where Meals_T.MealID= Food_Quantity_T.MealID and
Food_Quantity_T.FoodItemID=FoodItems_T.FoodItemID and
derivedtable.MealID=Food_Quantity_T.MealID
Group By FoodItemDescription  , FoodItems_T.FoodItemID,derivedtable.SetNumber

execute RefreshFoodPreparationPlanner_View
```

```
Select * from FoodPreparationPlanner_View --Queries Results for all SetNumbers
```

[This View , requires the User to query against the view with a where clause to specify the required SetNumber as below]

```
Select * from FoodPreparationPlanner_View where SetNumber=1 --Filters required set
```

FoodPreparationPlanner_View queried for records belonging to SetNumber 1



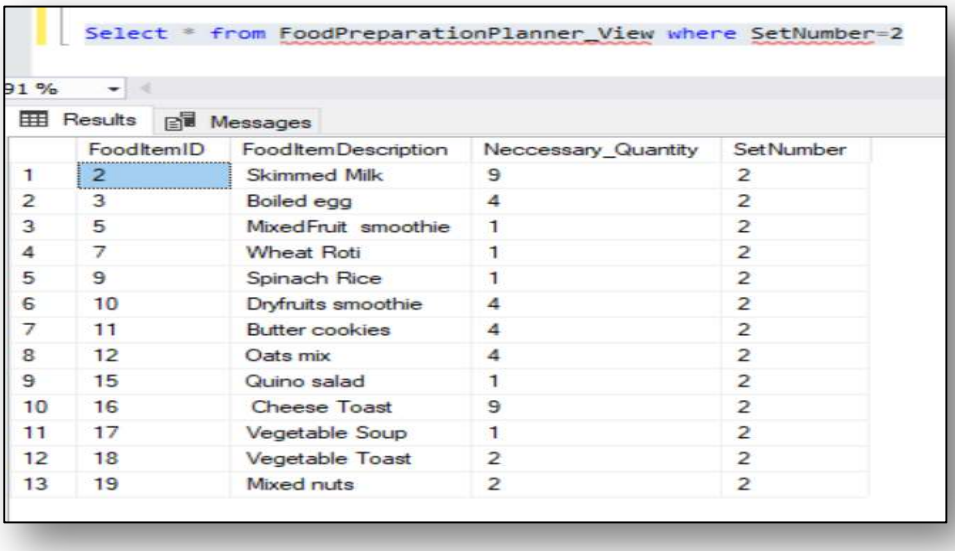
```
Select * from FoodPreparationPlanner_View where SetNumber=1
```

91 %

Results | Messages

	FoodItemID	FoodItemDescription	Neccessary_Quantity	SetNumber
1	2	Skimmed Milk	3	1
2	5	MixedFruit smoothie	3	1
3	6	Sprouts salad	5	1
4	7	Wheat Roti	5	1
5	9	Spinach Rice	5	1
6	10	Dryfruits smoothie	5	1
7	11	Butter cookies	36	1
8	12	Oats mix	8	1
9	15	Quino salad	5	1
10	16	Cheese Toast	5	1
11	18	Vegetable Toast	10	1
12	19	Mixed nuts	6	1

FoodPreparationPlanner_View queried for records belonging to SetNumber 1



```
Select * from FoodPreparationPlanner_View where SetNumber=2
```

91 %

Results | Messages

	FoodItemID	FoodItemDescription	Neccessary_Quantity	SetNumber
1	2	Skimmed Milk	9	2
2	3	Boiled egg	4	2
3	5	MixedFruit smoothie	1	2
4	7	Wheat Roti	1	2
5	9	Spinach Rice	1	2
6	10	Dryfruits smoothie	4	2
7	11	Butter cookies	4	2
8	12	Oats mix	4	2
9	15	Quino salad	1	2
10	16	Cheese Toast	9	2
11	17	Vegetable Soup	1	2
12	18	Vegetable Toast	2	2
13	19	Mixed nuts	2	2

7. SQL Statements for database triggers

7.1 Trigger to Log Update on Student_T table

The organization requires to update the student's height and weight regularly and tracks the growth statistics of a student. As per the design requirement, on updating new values to the height and weight of a student the previous values are to be moved to the growth_record table to store the history on a student's growth.

A database trigger is therefore designed to capture these updates on the Student_T table and perform the following actions.

- Copy the existing data values to Growth_Record_T with all corresponding data.
- Create a log of the update details in a separate database table StudentDataUpdates_Log
- Update existing Student_T tables record with the new values.

StudentDataUpdates_Log is a table that has been created to capture the changes to the Old and New values of a Student's height and weight. It is assumed to handle through the application design to create this log as a notification to Nutritionists in the organization to be notified about the update, to act on new values of a student by analysing the record and recommend a new diet plan to a student.

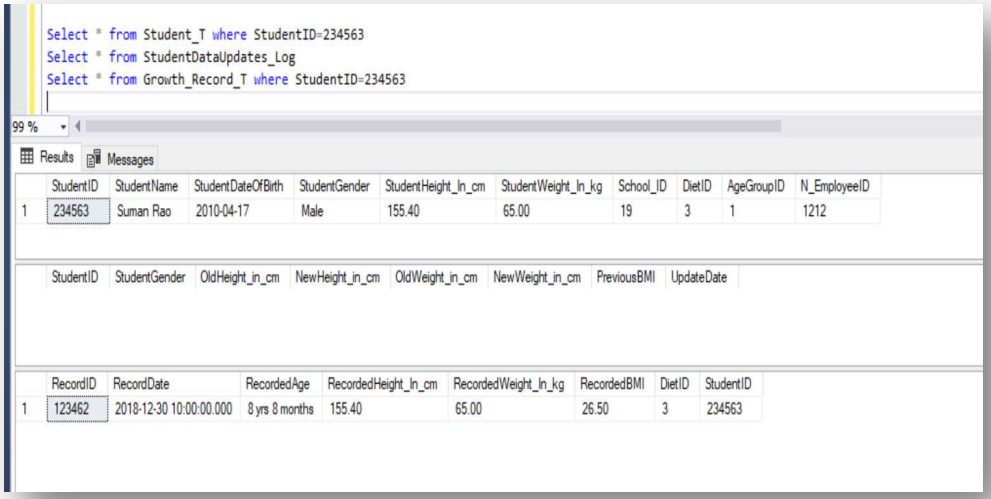
SQL Statements for the trigger

```
CREATE TABLE StudentDataUpdates_Log
(StudentID bigint, StudentGender char(6) ,
 OldHeight_in_cm decimal(5,2) ,NewHeight_in_cm decimal(5,2),
 OldWeight_in_cm decimal(5,2) ,NewWeight_in_cm decimal(5,2),
 PreviousBMI decimal(5,2),
 UpdateDate datetime)

create trigger StudentDataUpdate on Student_T
for update
as
if (update(StudentHeight_In_cm ) or update(StudentWeight_In_kg))
begin

--------To Push data to Growth_Record-------------
insert into Growth_Record_T(RecordID,RecordDate,RecordedAge ,
RecordedHeight_In_cm,RecordedWeight_In_kg , RecordedBMI,
 DietID ,StudentID)
 Select ((Select Max(RecordID) From Growth_Record_T) + 1) ,Getdate(),
((CAST ((DATEDIFF(MONTH,CASE WHEN DAY(inserted.StudentDateOfBirth) > DAY(GETDATE())
THEN DATEADD(MONTH,1,inserted.StudentDateOfBirth)
ELSE inserted.StudentDateOfBirth
END,GETDATE()) / 12) AS Varchar)+'yrs') +
(CAST((DATEDIFF(MONTH,CASE WHEN DAY(inserted.StudentDateOfBirth) > DAY(GETDATE())
THEN DATEADD(MONTH,1,inserted.StudentDateOfBirth)
ELSE inserted.StudentDateOfBirth
END,GETDATE()) % 12) AS VARCHAR)+'months')) ,
inserted.StudentHeight_In_cm, inserted.StudentWeight_In_kg ,
(round(((deleted.StudentWeight_In_kg /(deleted.StudentHeight_In_cm *
deleted.StudentHeight_In_cm))*10000), 2)) ,
inserted.DietID,inserted.StudentID
from inserted, deleted
where inserted.StudentID = deleted.StudentID
```

```
---To Log Update------------
insert into StudentDataUpdates_Log (StudentID,StudentGender,
OldHeight_in_cm,NewHeight_in_cm,
OldWeight_in_cm, NewWeight_in_cm ,PreviousBMI,UpdateDate)
select inserted.StudentID, inserted.StudentGender,
deleted.StudentHeight_In_cm, inserted.StudentHeight_In_cm,
deleted.StudentWeight_In_kg, inserted.StudentWeight_In_kg,
(round(((deleted.StudentWeight_In_kg /(deleted.StudentHeight_In_cm *
deleted.StudentHeight_In_cm))*10000), 2)),
 GETDATE()

from inserted, deleted
where inserted.StudentID = deleted.StudentID
end
```

Tables status before Update:

```
Select * from Student_T where StudentID=234563
Select * from StudentDataUpdates_Log
Select * from Growth_Record_T where StudentID=234563
```

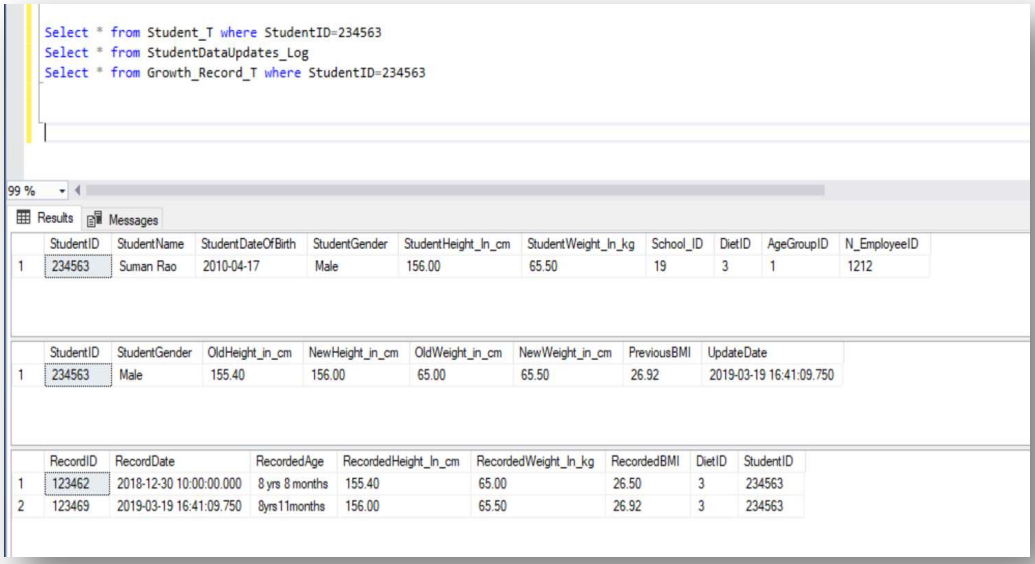


An Update to Student_T :

```
Update Student_T set StudentHeight_In_cm=156 , StudentWeight_In_kg= 65.50 where
StudentID=234563
```

Tables status after Update :

```
Select * from Student_T where StudentID=234563
Select * from StudentDataUpdates_Log
Select * from Growth_Record_T where StudentID=234563
```



```
Select * from Student_T where StudentID=234563
Select * from StudentDataUpdates_Log
Select * from Growth_Record_T where StudentID=234563
```

99 %

Results | Messages

	StudentID	StudentName	StudentDateOfBirth	StudentGender	StudentHeight_In_cm	StudentWeight_In_kg	School_ID	DietID	AgeGroupID	N_EmployeeID
1	234563	Suman Rao	2010-04-17	Male	156.00	65.50	19	3	1	1212

	StudentID	StudentGender	OldHeight_in_cm	NewHeight_in_cm	OldWeight_in_cm	NewWeight_in_cm	PreviousBMI	UpdateDate
1	234563	Male	155.40	156.00	65.00	65.50	26.92	2019-03-19 16:41:09.750

	RecordID	RecordDate	RecordedAge	RecordedHeight_In_cm	RecordedWeight_In_kg	RecordedBMI	DietID	StudentID
1	123462	2018-12-30 10:00:00.000	8 yrs 8 months	155.40	65.00	26.50	3	234563
2	123469	2019-03-19 16:41:09.750	8yrs11months	156.00	65.50	26.92	3	234563