**1)City Project**

CREATE TABLE CityProject\_T

(CityProjectID int not null CHECK (CityProjectID > 0),

CentralProjectID int not null CHECK (CentralProjectID > 0),

ORegistrationNumber int not null CHECK (ORegistrationNumber > 0),

TrainingID int not null CHECK (TrainingID > 0),

ProjectTitle nvarchar,

ProjectInitiationDate date,

ProjectCompletionDate date,

TotalFunding decimal(10,2) Check (TotalFunding >0)

CONSTRAINT CityProject\_PK PRIMARY KEY (CityProject),

CONSTRAINT CityProject\_FK1 FOREIGN KEY (CentralProjectID) REFERENCES CentralProject\_T (CentralProjectID),

CONSTRAINT CityProject\_FK2 FOREIGN KEY (TrainingID) REFERENCES AwarenessTraining\_T (TrainingID))

**2)CityProject\_Employee\_T**

CREATE TABLE CityProject\_Employee\_T

(CityProjectID int not null CHECK (CityProjectID > 0),

EmployeeID int not null CHECK (EmployeeID > 0),

CONSTRAINT CityProject\_PK PRIMARY KEY (CityProjectID),

CONSTRAINT CityProject\_FK1 FOREIGN KEY (CentralProjectID) REFERENCES CentralProject\_T (CentralProjectID),

CONSTRAINT CityProject\_FK2 FOREIGN KEY (ContractorID) REFERENCES Contractor\_T (ContractorID),

CONSTRAINT CityProject\_FK3 FOREIGN KEY (MPEmployeeID) REFERENCES Employee\_Permanent\_Manager\_T (MPEmployeeID))

**3)AwarenessTraining\_T**

create table AwarenessTraining\_T

(TrainingID int not null check (TrainingID>0),

TrainingTopic nvarchar(20) not null,

TrainingHours int (check TrainingHours>0),

Constraint AwarenessTraining\_PK Primary Key (TrainingID))

4)**NGO\_T**

create table NGO\_T

(ORegistrationNumber int not null check (ORegistrationNumber>0),

OrganisationName nvarchar(20) not null,

OFoundationDate date,

CONSTRAINT NGO\_PK primary key (ORegistrationNumber))

5) **CityProject\_EmployeeDetails\_T**

CREATE TABLE CityProject\_EmployeeDetails\_T

(CityProjectID int not null CHECK (CityProjectID > 0),

EmployeeID int not null CHECK (EmployeeID > 0),

CONSTRAINT CityProject\_EmployeeDetails\_PK PRIMARY KEY (CityProjectID,EmployeeID),

CONSTRAINT CityProject\_EmployeeDetails\_FK1 FOREIGN KEY (CityProjectID) REFERENCES CityProject\_T (CityProjectID),

CONSTRAINT CityProject\_EmployeeDetails\_FK2 FOREIGN KEY (EmployeeID) REFERENCES Employee\_T (EmployeeID))

6) **Contractor\_T**

CREATE TABLE Contractor\_T

(ContractorID int not null CHECK (ContractorID > 0),

ContractorName nvarchar(50),

ContractorContactNumber bigint CHECK (ContractorContactNumber > 0),

CONSTRAINT ConstructionContractor\_PK PRIMARY KEY (ContractorID))

7) **HouseOwner\_T**

CREATE TABLE HouseOwner\_T

(OwnerID int not null CHECK (OwnerID > 0),

TrainingID int not null CHECK (TrainingID > 0),

OwnerName nvarchar(50),

OwnerContactNumber bigint CHECK (OwnerContactNumber > 0),

CONSTRAINT ConstructionContractor\_PK PRIMARY KEY (ContractorID),

CONSTRAINT Owner\_FK FOREIGN KEY (TrainingID) REFERENCES Training\_T (TrainingID))

8**) ConstructionDesign\_T**

CREATE TABLE ConstructionDesign\_T

(DesignID int not null CHECK (DesignID > 0),

ECEmployeeID int not null CHECK (ECEmployeeID > 0),

StartDate datetime,

CompletionDate datetime

CONSTRAINT ConstructionDesign\_PK PRIMARY KEY (DesignID),

CONSTRAINT ConstructionDesign\_FK1 FOREIGN KEY (ECEmployeeID) REFERENCES Employee\_Contract\_Engineer\_T (ECEmployeeID))

9) **HouseUnit\_T**

CREATE TABLE HouseUnit\_T

(HouseUnitID int not null CHECK (HouseUnitID > 0),

CityProjectID int not null CHECK (CityProjectID > 0),

ContractorID int not null CHECK (ContractorID > 0),

DesignID int not null CHECK (DesignID > 0),

HouseOwnerID int not null CHECK (HouseOwnerID > 0),

UnitAddress nvarchar(50),

UCity nvarchar(20),

UState char(2),

CONSTRAINT HouseUnit\_PK PRIMARY KEY (HouseUnitID),

CONSTRAINT HouseUnit\_FK1 FOREIGN KEY (CityProjectID) REFERENCES CityProject\_T (CityProjectID),

CONSTRAINT HouseUnit\_FK2 FOREIGN KEY (ContractorID) REFERENCES Contractor\_T (ContractorID),

CONSTRAINT HouseUnit\_FK3 FOREIGN KEY (DesignID) REFERENCES ConstructionDesign\_T (DesignID),

CONSTRAINT HouseUnit\_FK4 FOREIGN KEY (HouseOwnerID) REFERENCES HouseOwner\_T (HouseOwnerID))

10**) Design\_Usage\_Details\_T**

CREATE TABLE Design\_Usage\_Details\_T

(DesignContractorID int not null CHECK (DesignContractorID > 0),

DesignID int not null CHECK (DesignID > 0),

ContractorID int not null CHECK (ContractorID > 0),

UsedOnDate date,

CONSTRAINT HouseUnit\_PK PRIMARY KEY (DesignContractorID),

CONSTRAINT HouseUnit\_FK1 FOREIGN KEY (DesignID) REFERENCES ConstructionDesign\_T (DesignID),

CONSTRAINT HouseUnit\_FK2 FOREIGN KEY (ContractorID) REFERENCES Contractor\_T (ContractorID))

11) **Employee\_T**

Create table Employee\_T

(EmployeeID int not null CHECK (EmployeeID >0),

Ename nvarchar(25),

EContactNumber char(10),

EmployeeType char(2) not null CHECK (EmployeeType IN ('P' , 'C')),

CONSTRAINT Employee\_PK PRIMARY KEY (EmployeeID))

12) **Employee\_Permanent\_T**

CREATE TABLE Employee\_Permanent\_T

(PEmployeeID int not null CHECK (PEmployeeID > 0),

AnnualSalary decimal(7,2) CHECK (AnnualSalary > 0),

EmployeePermanentType char(2) not null CHECK (EmployeePermanentType IN ('M'),

UsedOnDate date,

CONSTRAINT HouseUnit\_PK PRIMARY KEY (DesignContractorID),

CONSTRAINT HouseUnit\_FK1 FOREIGN KEY (DesignID) REFERENCES ConstructionDesign\_T (DesignID),

CONSTRAINT HouseUnit\_FK2 FOREIGN KEY (ContractorID) REFERENCES Contractor\_T (ContractorID))

13) **Employee\_Permanent\_Manager\_T**

CREATE TABLE Employee\_Permanent\_Manager\_T

(MPEmployeeID int not null CHECK (MPEmployeeID>0),

CONSTRAINT Employee\_Permanent\_Manager\_PK PRIMARY KEY (MPEmployeeID)

CONSTRAINT Employee\_Permanent\_Manager\_FK FOREIGN KEY (MPEmployeeID)

REFERENCES Employee\_Permanent\_T (PEmployeeID)

14) **Employee\_Contract\_T**

CREATE TABLE Employee\_Contract\_T

(CEmployeeID int not null CHECK(CEmployeeID>0),

HourlyPayRate decimal(5, 2) CHECK(HourlyPayRate >0),

EmployeeContractType char(2) not null CHECK (EmployeeContractType IN ('W' , 'E')),

CONSTRAINT Employee\_Contract\_PK PRIMARY KEY (CEmployeeID),

CONSTRAINT CONSTRAINT Employee\_Contract\_PK PRIMARY KEY (CEmployeeID),

CONSTRAINT Employee\_Contract\_FK1 FOREIGN KEY (CEmployeeID)

REFERENCES Employee\_T (EmployeeID))

15) **Employee\_Contract\_Engineer\_T**

CREATE TABLE Employee\_Contract\_Engineer\_T

(ECEmployeeID int not null CHECK (ECEmployeeID>0),

CONSTRAINT Employee\_Contract\_Engineer\_PK PRIMARY KEY (ECEmployeeID),

CONSTRAINT Employee\_Contract\_Engineer\_FK FOREIGN KEY (ECEmployeeID)

REFERENCES Employee\_Contract\_T (CEmployeeID))

16) **Employee\_Contract\_WDOfficial\_T**

CREATE TABLE Employee\_Contract\_WDOFFICIAL\_T

(WCEmployeeID int not null CHECK (WCEmployeeID>0),

CONSTRAINT Employee\_Contract\_WDOFFICIAL\_PK PRIMARY KEY (WCEmployeeID),

CONSTRAINT Employee\_Contract\_WDOFFICIAL\_FK FOREIGN KEY (WCEmployeeID)

REFERENCES Employee\_Contract\_T (CEmployeeID)

**Views:**

--**For keeping a track of what percentage of the total funding is bounced back to the government of a particular central project.**

**President View**

select CityProject\_T.CentralProjectID, CentralProject\_T.ProjectTitle,

(CentralProject\_T.TotalFunding-derivedtable.totalCityFunding) as balancefunding,

(((CentralProject\_T.TotalFunding-derivedtable.totalCityFunding)/CentralProject\_T.TotalFunding)\*100) as PercentOfTotalFunding

from cityproject\_t, CentralProject\_T,

(select CityProject\_T.CentralProjectID, sum(CityProject\_T.fundingrequired) as totalCityFunding

from CityProject\_T, centralproject\_t

where CentralProject\_T.CentralProjectID=CityProject\_T.CentralProjectID

group by CityProject\_T.CentralProjectID)as derivedtable

where CentralProject\_T.CentralProjectID=CityProject\_T.CentralProjectID

and CityProject\_T.CentralProjectID=derivedtable.CentralProjectID

group by CityProject\_T.CentralProjectID, CentralProject\_T.ProjectTitle,

(CentralProject\_T.TotalFunding-derivedtable.totalCityFunding),

(((CentralProject\_T.TotalFunding-derivedtable.totalCityFunding)/CentralProject\_T.TotalFunding)\*100)

order by PercentOfTotalFunding desc

This view will help the President to know the president of the NGO about the difference between how much pf the total funding has been used in distributing in the city projects as funding required minus the total funding given for a central project as total funding. A percentage of how much of the funding is remaining will also be shown. This is a very important view in term of finance as the remaining funding needs to be returned back to the central government. This amount of funding returned is termed as bounce back funding. It also helps in doing financial analysis on an NGO level about what percentage of the total funding received is used and how much is not.

**---For keeping a track for manager of which contractor has worked on how many house units and used which design on which date. The design which is most commonly used by the contactor will be listed first, demonstrating which design is the most used and which design is the least used.**

**Managerial View**

select Contractor\_T.ContractorName,Design\_Usage\_Details\_T.DesignID, Design\_Usage\_Details\_T.UsedOnDate,

count(HouseUnit\_T.HouseUnitID) as NumberOfHouseUnits,

CentralProject\_T.ProjectTitle

from Contractor\_T, Design\_Usage\_Details\_T, HouseUnit\_T, CentralProject\_T, CityProject\_T,

ConstructionDesign\_T

where Contractor\_T.ContractorID=Design\_Usage\_Details\_T.ContractorID

and HouseUnit\_T.ContractorID=Contractor\_T.ContractorID

and CityProject\_T.CentralProjectID=CentralProject\_T.CentralProjectID

and HouseUnit\_T.DesignID=ConstructionDesign\_T.DesignID

group by Contractor\_T.ContractorName,Design\_Usage\_Details\_T.DesignID, Design\_Usage\_Details\_T.UsedOnDate, CentralProject\_T.ProjectTitle

order by NumberOfHouseUnits desc, CentralProject\_T.ProjectTitle

This view is important for the manager as it will help him to have a look at how many units the contractor has totally worked on under which project. Additionally, this view will allow the manager to check which design designed by an engineer is used on which date by that particular contractor. This track will help the manager to keep a track of which house unit designs are most commonly used by the construction contractor for working on the construction

**Presidential View:**

select Employee\_Contract\_Engineer\_T.ECEmployeeID, derivedtable4.WCEmployeeID, CityProject\_T.MPEmployeeID,

CityProject\_T.CityProjectID,derivedtable3.count1, derivedtable4.count2, count(HouseUnit\_T.HouseUnitID)

from CityProject\_T, Employee\_Contract\_Engineer\_T, Employee\_Contract\_WDOFFICIAL\_T,HouseUnit\_T,

(select Employee\_Contract\_Engineer\_T.ECEmployeeID, count(ConstructionDesign\_T.DesignID) as count1

from Employee\_Contract\_Engineer\_T, ConstructionDesign\_T

where ConstructionDesign\_T.ECEmployeeID=Employee\_Contract\_Engineer\_T.ECEmployeeID

group by Employee\_Contract\_Engineer\_T.ECEmployeeID) as derivedtable3,

(select Employee\_Contract\_WDOFFICIAL\_T.WCEmployeeID,AwarenessTraining\_T.TrainingTopic,

count(AwarenessTraining\_T.TrainingID) as count2

from Employee\_Contract\_WDOFFICIAL\_T,AwarenessTraining\_T

where AwarenessTraining\_T.WCEmployeeID=Employee\_Contract\_WDOFFICIAL\_T.WCEmployeeID

group by Employee\_Contract\_WDOFFICIAL\_T.WCEmployeeID,AwarenessTraining\_T.TrainingTopic ) as derivedtable4

where derivedtable3.ECEmployeeID=Employee\_Contract\_Engineer\_T.ECEmployeeID

and derivedtable4.WCEmployeeID=Employee\_Contract\_WDOFFICIAL\_T.WCEmployeeID

group by Employee\_Contract\_Engineer\_T.ECEmployeeID, derivedtable4.WCEmployeeID,

CityProject\_T.MPEmployeeID, CityProject\_T.CityProjectID, derivedtable3.count1, derivedtable4.count2

**Managerial View:**

select CityProject\_T.CityProjectID, count(Contractor\_T.ContractorID), (HouseUnit\_T.HouseUnitID)

from Contractor\_T, CityProject\_T, HouseUnit\_T,

(select max(derivedtable1.count1) as maxvalue

from

(select count(HouseUnit\_T.HouseUnitID) as count1, CityProject\_T.CityProjectID

from HouseUnit\_T, CityProject\_T

where HouseUnit\_T.CityProjectID=CityProject\_T.CityProjectID

group by CityProject\_T.CityProjectID) as derivedtable1) as derivedtable2

**Managerial View: Which contractor has worked in which state on how many city proposals**

select Contractor\_T.ContractorName, CityProject\_T.Cstate,count(CityProject\_T.CityProjectID)

from Contractor\_T,CityProject\_T, HouseUnit\_T

where Contractor\_T.ContractorID=CityProject\_T.ContractorID

and HouseUnit\_T.ContractorID=Contractor\_T.ContractorID

group by Contractor\_T.ContractorName, CityProject\_T.Cstate

**TRIGGERS:**

**--Trigger created for the log of change in funding required for a particular city project.**

CREATE TABLE FundingUpdates\_Log

(CityProjectID int,

CityName nvarchar(20),

OldFundingRequired decimal(7,2),

NewFundingRequired decimal(7,2),

UpdateDate datetime)

create trigger FundingUpdates on CityProject\_T

for update

as

if update(FundingRequired)begin

insert into FundingUpdates\_Log (CityProjectid, CityName, oldfundingrequired,

newfundingrequired, updatedate)

select inserted.CityProjectID, inserted.CityName,

deleted.FundingRequired, inserted.FundingRequired, GETDATE()

from inserted,

deleted where inserted.CityProjectID = deleted.CityProjectID

end

update CityProject\_t set FundingRequired = 36000 where CityProjectid = 11111