CREATE PROCEDURE [dbo].[addKid]

@kidID char(9),

@firstName nvarchar(20),

@surName nvarchar(20),

@dateOfBirth date,

@street nvarchar(20),

@houseNumber nvarchar(5),

@city nvarchar(20),

@latitude float,

@longitude float,

@kindergardenID int,

@classNumber int,

@parentID char(9)

AS

declare @placeInFamily tinyint

select @placeInFamily=count(\*)

from Kids k

where k.parentID=@parentID

insert into Persons values (@kidID, @firstName, @surName, @dateOfBirth)

insert into Kids values (@kidID, @street, @houseNumber, @city, @latitude, @longitude, @placeInFamily + 1, @kindergardenID, @classNumber, @parentID)

select k.\*

from Kids k

where k.kidID = @kidID

CREATE PROCEDURE [dbo].[addKidToActivity]

@kidID char(9),

@activityID int

AS

DECLARE @kgID int, @clNum int

SELECT @kgID = kindergardenID, @clNum = classNumber

FROM Kids

WHERE kidID = @kidID

INSERT INTO SignedFor VALUES (@kidID, @kgID, @clNum, @activityID)

SELECT \*

FROM SignedFor

WHERE kidID = @kidID and activityID = @activityID

CREATE PROCEDURE [dbo].[addOpinionToAct]

@id int,

@text nvarchar(max),

@grade tinyint

AS

INSERT INTO Opinions (talk, grade, activityID) VALUES (@text, @grade, @id)

SELECT \*

FROM Opinions O

WHERE O.activityID = @id and O.talk = @text and O.grade = @grade

CREATE PROCEDURE [dbo].[addOpinionToKg]

@id int,

@text nvarchar(max),

@grade tinyint

AS

INSERT INTO Opinions (talk, grade, kindergardenNum) VALUES (@text, @grade, @id)

SELECT \*

FROM Opinions O

WHERE O.kindergardenNum = @id and O.talk = @text and O.grade = @grade

CREATE PROCEDURE [dbo].[addParent]

@parentID char(9),

@firstName nvarchar(20),

@surName nvarchar(20),

@DOB date,

@gender char(1),

@pass varchar(20)

AS

insert into Persons values (@parentID, @firstName, @surName, @DOB)

insert into Parents values (@parentID, @gender, @pass)

select p.\*

from Parents p

where p.parentID = @parentID

CREATE PROCEDURE [dbo].[checkIfIDExists]

@parentID char(9)

AS

SELECT P.ID

FROM Persons P

WHERE P.ID = @parentID

CREATE PROCEDURE [dbo].[checkIfKidAlreadyInActivity]

@kidID char(9),

@activityID int

AS

SELECT \*

FROM SignedFor

WHERE kidID = @kidID and activityID = @activityID

CREATE PROCEDURE [dbo].[checkIfParentExists]

@parentID char(9)

AS

SELECT P.parentID

FROM Parents P

WHERE P.parentID = @parentID

CREATE PROCEDURE [dbo].[checkLoginData]

@parentID char(9),

@parentPass varchar(20)

AS

select \*

from Parents p join Persons per on p.parentID = per.ID

where p.parentID = @parentID and p.pass = @parentPass

CREATE PROCEDURE [dbo].[findAllPrivateKGbyDate]

@dateOfBirth date

AS

declare @age int

select @age=DATEDIFF(YEAR, @dateOfBirth, cast(year(getdate()) as char(4))+'-09-01')

SELECT kg.ID, kg.name as KGname, kg.street, kg.houseNumber, kg.latitude, kg.longitude, kg.price, cl.number, cl.name as CLname, cl.maximumKids, (

SELECT count(\*)

FROM kids k

WHERE k.kindergardenID=cl.kindergardenID and k.classNumber=cl.number

) as kidsInClass, p.surName, p.firstName

FROM Kindergardens kg join Classes cl on kg.ID=cl.kindergardenID join Persons p on cl.teacher = p.ID

WHERE kg.private=1 and cl.minimumAge<=@age and @age<=cl.maximumAge

CREATE PROCEDURE [dbo].[findAllPublicKGbyDate]

@dateOfBirth date

AS

declare @age int

select @age=DATEDIFF(YEAR, @dateOfBirth, cast(year(getdate()) as char(4))+'-09-01')

SELECT kg.ID, kg.name as KGname, kg.street, kg.houseNumber, kg.latitude, kg.longitude, kg.price, cl.number, cl.name as CLname, cl.maximumKids, (

SELECT count(\*)

FROM kids k

WHERE k.kindergardenID=cl.kindergardenID and k.classNumber=cl.number

) as kidsInClass, p.surName, p.firstName

FROM Kindergardens kg join Classes cl on kg.ID=cl.kindergardenID join Persons p on cl.teacher = p.ID

WHERE kg.private=0 and cl.minimumAge<=@age and @age<=cl.maximumAge

CREATE PROCEDURE [dbo].[getActInfoForChart]

AS

SELECT A.name, ROUND(AVG(DISTINCT CAST(O.grade AS FLOAT)), 2) as aver, COUNT(DISTINCT SF.kidID) as total

FROM Activities A left outer join Opinions O on A.ID = O.activityID left outer join SignedFor SF on A.ID = SF.activityID

GROUP BY A.name

CREATE PROCEDURE [dbo].[getActNumsInOpinion]

AS

SELECT DISTINCT O.activityID as ID, A.name

FROM Opinions O join Activities A on O.activityID = A.ID

WHERE O.activityID > 0

CREATE PROCEDURE [dbo].[getAllActivitiesInChosenClass]

@kgID int,

@clNum int

AS

SELECT A.ID, A.name, dayInWeek, TP.startTime, DATEADD(MINUTE,A.length,TP.startTime) as endTime

FROM TakesPlace TP join Activities A on TP.activityID = A.ID

WHERE TP.kindergardenID = @kgID and TP.classNumber = @clNum

CREATE PROCEDURE [dbo].[getAllActivitiesInKidsClass]

@kidID char(9)

AS

DECLARE @kgID int, @clNum int

SELECT @kgID = kindergardenID, @clNum = classNumber

FROM Kids

WHERE kidID = @kidID

SELECT A.ID, A.name, O.surName, O.firstName, DATENAME(dw, TP.dayInWeek-2) as dayInWeek, TP.startTime, DATEADD(MINUTE,A.length,TP.startTime) as endTime, A.cost

FROM TakesPlace TP join Activities A on TP.activityID = A.ID join Operators O on TP.operatorID = O.ID

WHERE TP.kindergardenID = @kgID and TP.classNumber = @clNum

CREATE PROCEDURE [dbo].[getAllActivityOpinions]

@activityID int

AS

SELECT talk, grade

FROM Opinions

WHERE activityID = @activityID

CREATE PROCEDURE [dbo].[getAllActNum]

AS

SELECT A.ID

FROM Activities A

CREATE PROCEDURE [dbo].[getAllClassesOfKg]

@kgID int

AS

SELECT number as ID, name

FROM Classes

WHERE kindergardenID = @kgID

CREATE PROCEDURE [dbo].[getAllKgAndName]

AS

SELECT ID, name

FROM Kindergardens

CREATE PROCEDURE [dbo].[getAllKgNum]

AS

SELECT K.ID

FROM Kindergardens K

CREATE PROCEDURE [dbo].[getAllKgOpinions]

@kgID int

AS

SELECT talk, grade

FROM Opinions

WHERE kindergardenNum = @kgID

CREATE PROCEDURE [dbo].[getAllKidsOfParent]

@parentID char(9)

AS

SELECT k.parentID, k.kidID

FROM Kids k where [k.parentID=@parentID](mailto:k.parentID=@parentID)

CREATE PROCEDURE [dbo].[getAllOpinionsOfAct]

@ActID int

AS

SELECT ID, talk as text, grade

FROM Opinions

WHERE activityID = @ActID

CREATE PROCEDURE [dbo].[getAllOpinionsOfKg]

@KgID int

AS

SELECT ID ,talk as text, grade

FROM Opinions

WHERE kindergardenNum = @KgID

CREATE PROCEDURE [dbo].[getKgInfoForChart]

AS

SELECT K.name, ROUND(AVG(DISTINCT CAST(O.grade AS FLOAT)), 2) as aver, COUNT(DISTINCT Ki.kidID) as total

FROM Kindergardens K left outer join Opinions O on K.ID = O.kindergardenNum left outer join Kids Ki on K.ID = Ki.kindergardenID

GROUP BY K.name

CREATE PROCEDURE [dbo].[getKgNumsInOpinion]

AS

SELECT DISTINCT O.kindergardenNum as ID, K.name

FROM Opinions O join Kindergardens K on O.kindergardenNum = K.ID

WHERE O.kindergardenNum > 0

CREATE PROCEDURE [dbo].[insertUpdateT]

@1 int, --ID

@2 nvarchar(20) --name

AS

DECLARE @ans int = 0

IF EXISTS (SELECT \* FROM Training T WHERE T.ID = @1)

BEGIN

UPDATE Training SET name = @2 WHERE ID = @1

SET @ans = 1

END

ELSE

BEGIN

INSERT INTO Training VALUES (@1, @2)

SET @ans = 2

END

SELECT @ans as ans

CREATE PROCEDURE [dbo].[insertUpdateTFA]

@1 char(9), --assistantID

@2 int, --trainingID

@3 date --date

AS

DECLARE @ans int = 0

IF EXISTS (SELECT \* FROM TrainingForAssistant TFA WHERE TFA.assistantID = @1 and TFA.trainingID = @2)

BEGIN

UPDATE TrainingForAssistant SET trainingDate = @3 WHERE assistantID = @1 and trainingID = @2

SET @ans = 1

END

ELSE

BEGIN

INSERT INTO TrainingForAssistant VALUES (@1, @2, @3)

SET @ans = 2

END

SELECT @ans as ans

CREATE PROCEDURE [dbo].[removeOpinion]

@id int

AS

DELETE FROM Opinions WHERE ID = @id

SELECT \*

FROM Opinions O

WHERE O.ID = @id