Eric Reyneke

Student Number: 576787

Database Development 281 Project

Table of Contents

Page 3: Background and the ERD

Page 4 - 14: Is the code used to create the SchoolDBD and what all the code does.

Page 15: Question asked and answered.

Background

Student primary School was a well-established primary School in Akasia and had rapid growth after the first classes began in 1997. After Covid hit South Africa Student primary School was not able to handle the situation so they brought me in to assess the situation and help where I can. Student primary School had a database that did not conform to coding or database standards. I started by creating a whole new Entity relational diagram (ERD) from scratch and built a new database on SQL server from the ground up.

The ERD that is presented conforms to all the Normalizations steps.

In the 1NF and 2NF I removed all the tables and columns that were depended on primary keys.

I further removed redundant and duplicated data throughout the 1NF. After the Data complied to all requirement of 1NF I removed subsets of data and placed them into their own tables. I then added relationships between these tables and added their respected foreign keys.

Thus, all the duplicated and redundant data was removed and relationships where added. The ERD I created complies with 1NF, 2NF and 3Nf.

Diagram, schematic

Description automatically generated

Code that I used to create SchoolDBD for Student primary School

-- Creating Database SchoolDBD

Go

Create database SchoolDBD

On

(

Name = SchoolDBD,

FileName = 'C:\SchoolDBD\SchoolDBD2.mdf',

Size = 10MB,

MaxSize = 10GB,

FileGrowth = 10%

)

-- Creating Tables

Go

Use SchoolDBD

Create table Teachers

(TeacherID INT Identity(1,1) Primary Key,

FirstName VarChar(20) Not Null,

LastName VarChar(20) Not Null,

Age Int Check(Age between 18 And 65) Not Null,

City VarChar(20) Not Null,

StreetName VarChar(20) Not Null,

Country VarChar(20) Null,

Subject VarChar(20) Not Null,

SportsCoach VarChar(20) Null,

Salary INT Not Null)

Go

Use SchoolDBD

Create Table Parents

(ParentsID INT Identity(1,1) Primary Key,

FirstName VarChar(20) Not Null,

LastName VarChar(20) Not Null,

City VarChar(20) Not Null,

StreetName VarChar(20) Not Null,

Country VarChar(20) Null,

NumOfChildrenInSchool INT Not Null,

Age Int Check(Age>18))

Alter Table Parents

Add Province VarChar(20) Not Null;

Alter Table Parents

Add Gender VarChar(20) Null Default 'Rather not say';

Go

Use SchoolDBD

Create Table Subjects

(SubjectCode Int Identity(1,1) Primary Key,

TeacherID Int References Teachers(TeacherID),

SubjectName VarChar(20) Not Null,

NumStudentsEnrolled INT Not Null,

LecturerNameGivingSubject VarChar(20) Not Null

)

Go

Use SchoolDBD

Create Table ShopProducts

(ProductCode Int Identity(1,1) Primary Key not Null,

ProductPrice Decimal Check(ProductPrice<100) Not null,

ProductDiscription VarChar(20) Not null,

InStock INT Default 0,

)

Alter Table ShopProducts

Add ProductName VarChar(20) Not Null;

Go

Use SchoolDBD

Create Table Tutors

(TutorID INT Identity(1,1) Primary Key,

FirstName VarChar(20) Not Null,

LastName VarChar(20) Not Null,

MiddleName VarChar(20) Null,

SubjectGiving VarChar(20) Not Null

)

Go

Use SchoolDBD

Create Table Classroom

(ClassroomNumber INT Identity(1,1) Primary Key,

SubjectCode Int References Subjects(SubjectCode),

TutorID INT References Tutors(TutorID),

TypeOfProjector VarChar(20) Null,

TypeOfBoard VarChar(20) Not Null,

SubjectGivenInClass VarChar(20) Not Null,

)

Go

Use SchoolDBD

Create Table Students

(StudentNum INT Identity(1,1) Primary Key,

ParentsID Int References Parents(ParentsID),

ClassroomNumber Int References Classroom(ClassroomNumber),

TutorID Int References Tutors(TutorID),

ProductCode Int References ShopProducts(ProductCode),

FirstName VarChar(20) Not Null,

LastName VarChar(20) Not Null,

Age INT Not Null,

Grade INT Check(Grade between 1 and 3) Not Null,

GradeInMath Int Check(GradeInMath Between 1 and 100) Not Null,

GradeInEng Int Check(GradeInEng Between 1 and 100) Not Null,

GradeInAfr Int Check(GradeInAfr Between 1 and 100) Not Null

)

Use SchoolDBD

Go

Create Table Teacher\_Students

(TeacherID INT References Teachers(TeacherID),

StudentNum INT References Students(StudentNum),

Primary Key (TeacherID, StudentNum))

Use SchoolDBD

Go

Create Table Students\_Subject

(StudentNum INT References Students(StudentNum),

SubjectCode INT References Subjects(SubjectCode),

Primary Key (StudentNum, SubjectCode))

-- Inserts

-- Inserts into Parents Table

Use SchoolDBD

Go

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Zubair', 'Jimenez', 'Pretoria', 'Broodboom Street', 'South Africa', 1, 45, 'Gauteng', 'Male')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Miruna', 'Kouma', 'Brits', 'KameelVoet Avn', 'South Africa', 3, 57, 'North West', '')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Eloisa', 'Hooper', 'Pretoria', 'Nar Street', 'South Africa', 1, 37, 'Gauteng', 'Female')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Saira', 'Melia', 'KlerksDorp', 'BetterMe Street', 'South Africa', 2, 67, 'North West', 'Female')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Dawson', 'Hills', 'Cape Town', 'Konoha Avn', 'South Africa', 1, 24, 'West Cape', 'Male')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Sarina', 'Mcdougall', 'johannesburg', 'South Africa', 'WorkHard Street', 3, 49, 'Gauteng', '')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Rosemary', 'Wu', 'Pretoria', 'Living Road', 'South Africa', 1, 30, 'Gauteng', 'Female')

Insert into Parents (FirstName, LastName, City, StreetName, Country, NumOfChildrenInSchool, Age, Province, Gender)

Values ('Evangeline', 'Hayward', 'Durban', 'WHatIsLove Road', 'South Africa', 3, 25, 'KwaZulu-Natal', 'Male')

-- Inserts into ShopProducts Table

Use SchoolDBD

Go

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (16, 'Cold Liquid', 20, 'Coke')

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (0.50, 'Sour Sweet', 76, 'Sour Sweet')

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (14, 'Sweet Sweet', 13, 'Strips')

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (4, 'Milky chocolate', 7, 'Lint chocolate')

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (18, 'Cold Energy dink', 9, 'Play')

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (15, 'Beef pie', 4, 'Chicken mushroom pie')

-- Inserts into Students Table

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (1, 10, 1, 1, 'Arie', 'Dekker', 6, 1, 34, 67, 93)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (2, 10, 2, 5, 'Cherene', 'Whambag', 6, 1, 75, 25, 74)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (2, 10, 2, 3, 'Christian', 'Hooper', 7, 2, 86, 47, 75)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (2, 10, 3, 2, 'Eric', 'Reyneke', 7, 2, 85, 45, 26)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (3, 10, 3, 5, 'Stefan', 'Du Toit', 6, 1, 78, 89, 93)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, ParentsID, ClassroomNumber, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (4, 11, 1, 1, 'Alex', 'Nar', 8, 3, 57, 24, 97)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (4, 11, 2, 1, 'Amore', 'Koos', 6, 1, 74, 27, 16)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (5, 11, 1, 2, 'Dillon', 'Erasmus', 10, 5, 47, 25, 74)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (6, 11, 3, 3, 'Daleen', 'Reyneke', 9, 4, 76, 38, 90)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (6, 11, 2, 4, 'Theo', 'Malan', 9, 4, 95, 87, 58)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (6, 12, 1, 4, 'Michka', 'Van Niekerk', 8, 3, 48, 83, 94)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (7, 12, 2, 4, 'Xander', 'Nienabder', 8, 3, 75, 27, 90)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (8, 12, 1, 2, 'Vierkie', 'Snaams', 6, 1, 67, 75, 40)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (8, 12, 2, 1, 'Petrie', 'Wanie', 6, 1, 74, 9, 46)

Use SchoolDBD

Go

Insert into Students (ParentsID, ClassroomNumber, TutorID, ProductCode, FirstName, LastName, Age, Grade, GradeInMath, GradeInEng, GradeInAfr)

Values (8, 12, 3, 2, 'Alica', 'Smit', 10, 5, 87, 90, 43)

-- Insert Into Teacher Table

Use SchoolDBD

Go

Insert Into Teachers (FirstName, LastName, Age, City, StreetName, Country, Subject, SportsCoach, Salary)

Values ('Akeem', 'Hassan', 64, 'Pretoria', 'Fondel Street', 'South Africa', 'Mathematics', 'Rugby', 50000)

Use SchoolDBD

Go

Insert Into Teachers (FirstName, LastName, Age, City, StreetName, Country, Subject, SportsCoach, Salary)

Values ('Samira', 'Goulding', 54, 'Johannesburg', 'Snasie Street', 'South Africa', 'English', 'Jukskei', 44000)

Use SchoolDBD

Go

Insert Into Teachers (FirstName, LastName, Age, City, StreetName, Country, Subject, SportsCoach, Salary)

Values ('Aria', 'Roman', 20, 'Pretoria', 'Presedent Street', 'South Africa', 'Afrikaans', 'Ice hokkey', 44000)

-- Insert Into Subjects table

Use SchoolDBD

Go

Insert Into Subjects (TeacherID, SubjectName, NumStudentsEnrolled, LecturerNameGivingSubject)

Values (1, 'Mathematics', 15, 'Xander Erasmus')

Use SchoolDBD

Go

Insert Into Subjects (TeacherID, SubjectName, NumStudentsEnrolled, LecturerNameGivingSubject)

Values (2, 'English', 15, 'Theolene Nienaas')

Use SchoolDBD

Go

Insert Into Subjects (TeacherID, SubjectName, NumStudentsEnrolled, LecturerNameGivingSubject)

Values (3, 'Afrikaans', 15, 'Botghe Saabie')

-- Insert Into tutors Table

Use SchoolDBD

Go

Insert Into Tutors (FirstName, LastName, MiddleName, SubjectGiving)

Values ('Manpreet', 'Pugh', 'W', 'Mathematics')

Use SchoolDBD

Go

Insert Into Tutors (FirstName, LastName, MiddleName, SubjectGiving)

Values ('Stanislaw', 'Michael', 'M', 'English')

Use SchoolDBD

Go

Insert Into Tutors (FirstName, LastName, MiddleName, SubjectGiving)

Values ('Hamish', 'Mejia', 'R', 'Afrikaans')

-- Inserts into Classroom Table

Use SchoolDBD

Go

Insert into Classroom (SubjectCode, TutorID, TypeOfProjector, TypeOfBoard, SubjectGivenInClass)

Values (4, 1, 'Sony VPL-FHZ101L/B', 'White Board', 'Mathematics')

Insert into Classroom (SubjectCode, TutorID, TypeOfProjector, TypeOfBoard, SubjectGivenInClass)

Values (5, 2, 'Panasonic PT-RQ32KE', 'White Board', 'English')

Insert into Classroom (SubjectCode, TutorID, TypeOfProjector, TypeOfBoard, SubjectGivenInClass)

Values (6, 3, 'SACER X1327Wi', 'White Board', 'Afrikaans')

-- Queries

-- Which class bought the most cokes at the shop?

Use SchoolDBD

Go

Create View vCokes(ClassNumber, AmountOfCokes)

AS

Select ClassroomNumber, Count(SP.ProductCode)

From Students S

inner join ShopProducts SP ON SP.ProductCode=S.ProductCode

Where SP.ProductCode = 1

Group By ClassroomNumber

Use SchoolDBD

Go

Select \*

From vCokes

-- Show Which student belongs to which parent

Use SchoolDBD

Go

Create view vParentChildren (ParentID, FirstName, LastName, Students)

AS

Select P.ParentsID, P.FirstName, P.LastName, S.FirstName + ' ' + S.LastName

From Parents P

inner join Students S ON P.ParentsID= S.ParentsID

Use SchoolDBD

Go

Select \*

From vParentChildren

-- What is every students avarage?

Use SchoolDBD

Go

Create view vGrade (StudentsFullName, WhatGrade, ActualAvg)

AS

Select FirstName + '' + LastName, Grade, AVG(GradeInAfr + GradeInEng + GradeInMath) / 3

From Students

Group By FirstName + '' + LastName, Grade

Use SchoolDBD

Go

Select \*

From vGrade

-- What class subject is the most teached by tutors?

Use SchoolDBD

Go

Create View vTutors (TutorID, FullName, SubjectGiven, AmountOfStudentsInSubject)

AS

Select T.TutorID, T.FirstName + ' ' + T.LastName, SubjectGiving, Count(S.StudentNum)

From Tutors T

inner join Students S ON T.TutorID=S.TutorID

Group by T.TutorID, T.FirstName + ' ' + T.LastName, SubjectGiving

Having Count(S.StudentNum) > 4

Use SchoolDBD

Go

Select \*

From vTutors

-- How much money has been made of every product in the Shop

Use SchoolDBD

Go

Create View vProfit (ProductName, ProfitMade)

AS

Select ProductName, Sum(ProductPrice) \* Count(S.ProductCode)

From ShopProducts S

inner join Students T ON S.ProductCode=T.ProductCode

Group By ProductName

Use SchoolDBD

Go

Select \*

From vProfit

-- Logins

Use SchoolDBD

Go

Create Login Eric With password = 'EricDatabase121';

Use SchoolDBD

Go

Create Login Christian With password = 'ChristianDatabase121';

-- Stored Procedures

-- This procedure will delete specified studends that left school.

Use SchoolDBD

Go

Create Proc spDeleteStudents

@StudentNumber INT

AS

Begin

Delete Students

Where StudentNum = @StudentNumber

END;

-- This Procedure Updates Grades in the Student Table

Create Proc spUpdateGrades

@StudentNum Int,

@MathGradeAvg Int,

@AfrGradeAvg Int,

@EngGradeAvg Int

AS

Begin

Update Students

Set GradeInMath = @MathGradeAvg,

GradeInAfr = @AfrGradeAvg,

GradeInEng = @EngGradeAvg

Where StudentNum = @StudentNum

End

-- Insert New Poducts into the ShopProducts tabel

Use SchoolDBD

Go

Create Proc spNewProducts

@ProductPrice INT,

@ProductDiscription VarChar(100),

@InStock INT,

@ProductName VarChar(20)

AS

Begin

Insert Into ShopProducts (ProductPrice, ProductDiscription, InStock, ProductName)

Values (@ProductPrice, @ProductDiscription, @InStock, @ProductName)

END;

-- Transactions

-- Is Used to grant access to Eric and Cristiaan

Begin Try

Begin Transaction

Grant Insert, Update ON Subjects

To Eric, Christian

Print 'Access was granted to Eric and Cristian'

Commit Transaction

End Try

Begin Catch

Print 'Transaction failed'

RollBack

End Catch

-- This transaction will show what grade the student should be in from their age.

Begin Try

Begin Transaction

Select FirstName, Age,

Case

When Age = 6 Then 'You should be in grade 1'

When Age = 7 Then 'You should be in grade 2'

When Age = 8 Then 'You should be in grade 3'

When Age = 9 Then 'You should be in grade 4'

When Age = 10 Then 'You should be in grade 5'

End AS 'The Grade you should be in'

From Students

Commit Transaction

End try

Begin Catch

Print 'Transaction was a fail'

Rollback

End Catch

-- Triggers

-- This triger is disigned to display Spcified names data from Students before the table was updated.

Create trigger tShowBeforeUpdated

On Students

For Update

AS

Begin

Declare @SpecifiedName VarChar(20)

Select \* From Deleted Where FirstName = @SpecifiedName

Select \* From Inserted Where FirstName = @SpecifiedName

End;

--This trigger shows the new teacher that was employed.

CREATE TRIGGER tr\_NewTeacher

ON Teachers

AFTER Insert

AS

BEGIN

Declare @TeacherName VarChar(20),

@TeacherLastName VarChar(20),

@GivingSubject VarChar(20),

@Salary INT

SELECT @TeacherName = FirstName, @TeacherLastName = LastName, @GivingSubject = Subject, @Salary = Salary

FROM Inserted

Print 'First Name: '+ @TeacherName

Print 'Last Name: '+ @TeacherLastName

Print 'Subject: ' + @GivingSubject

Print 'Salary: '+ Convert(VarChar, @Salary)

END

GO

-- This trigger revokes the Delete function on Subjects

Create trigger tRevokeDeleting

On Subjects

For Delete

As

Print 'Sorry not allowed to Delete from this tables'

-- BackUp

Use SchoolDBD

Go

BackUp Database SchoolDBD

To DISK = 'C:\SchoolDBD\BackUp';

What all the code does.

* Create Table Teachers: This table was used to store all the data related to the teachers.
* Create Table Parents: This table collected all the data related to the parents.
* Create Table Subjects: This table was used to store all the data related to the subjects.
* Create Table ShopProducts: This table was used to store the data related to the shop.
* Create Table Tutors: This table was used to store the data related to the Tutors.
* Create Table Classroom: This table was used to store the data related to the Classroom.
* Create Table Students: This table was used to store the data related to the Students.
* Create Table Teacher\_Students: This is a bridge entity between the Teachers and Students table.
* Create Table Students\_Subjects: This is a bridge entity between the Subjects and Students table.
* View 1: Shows which class bought the most cokes form the shop.
* View 2: Shows which student belongs to which parent.
* View 3: Shows every student’s average.
* View 4: Shows what subjects is the toughed the most through tutors.
* View 5: Shows how much money is made from each product.
* Create Login: This was used to create a log in for Eric and Christian.
* Create Procedure 1: This procedure will delete specified studends that left school.
* Create Procedure 2: This procedure updated grades in the Students table.
* Create Procedure 3: This procedure Inserts new products in the ShopProducts table.
* Create transaction 1: This transaction is used to grant Eric and Christian access to insert and update table Subjects.
* Create transaction 2: This transaction will show you what grade the student is depending on their ages.
* Create Trigger 1: This trigger is designed to display Specified names from Students before the table was updated.
* Create Trigger 2: This trigger Shows the latest teacher that was employed.
* Create Trigger 3: This trigger revokes the delete function in the subjects table.
* BackUp : BackUp was used to back up and protect the data in the database.

Questions asked and answered in the database.

* Question 1: Which class of student’s bough the most cokes?
* Question 2: Which students belongs to which parents?
* Question 3: What is every student’s average?
* Question 4: What subject is thought the most by the tutors.
* Question 5: How much money is made through every product.