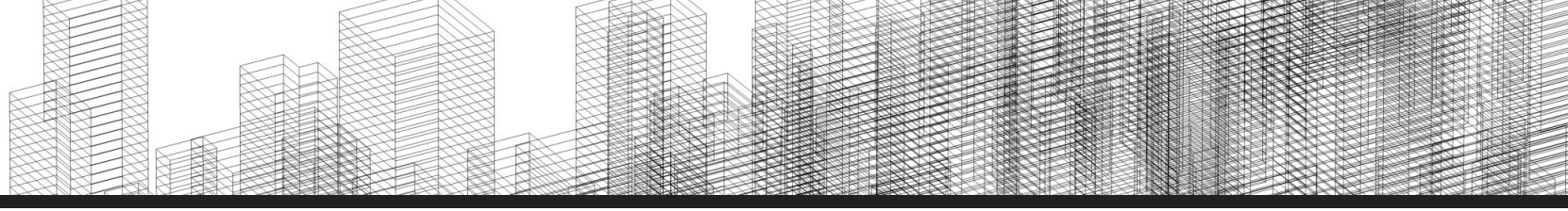


Aragon Pharmacy

Raisa Stepanova-Timina, Anastassia Titova, Andrey Lyamkin, Leonid Digerman

Work ethics

- Skilled team work
- Scheduled daily meetings
- In detail discussion for each subject
- Each team member having own version of new task
- Elaborate unity of each team member’s files into one
- Rotation of the Team Lead
- Daily Scrum

What we learned

- Strong Teamwork
- Creation and manipulation Database:
  - Creation and manipulation of Tables, Constraints, Data Types, Indexes, Views, Functions and Triggers
  - Upload and manipulation of Data
- Time management and Organization
- Regex Patterns
- Binding rules with user defined types in sql server
- How SQL store dates and how to format dates.
- Datetime Data Type

The goal for creating Database

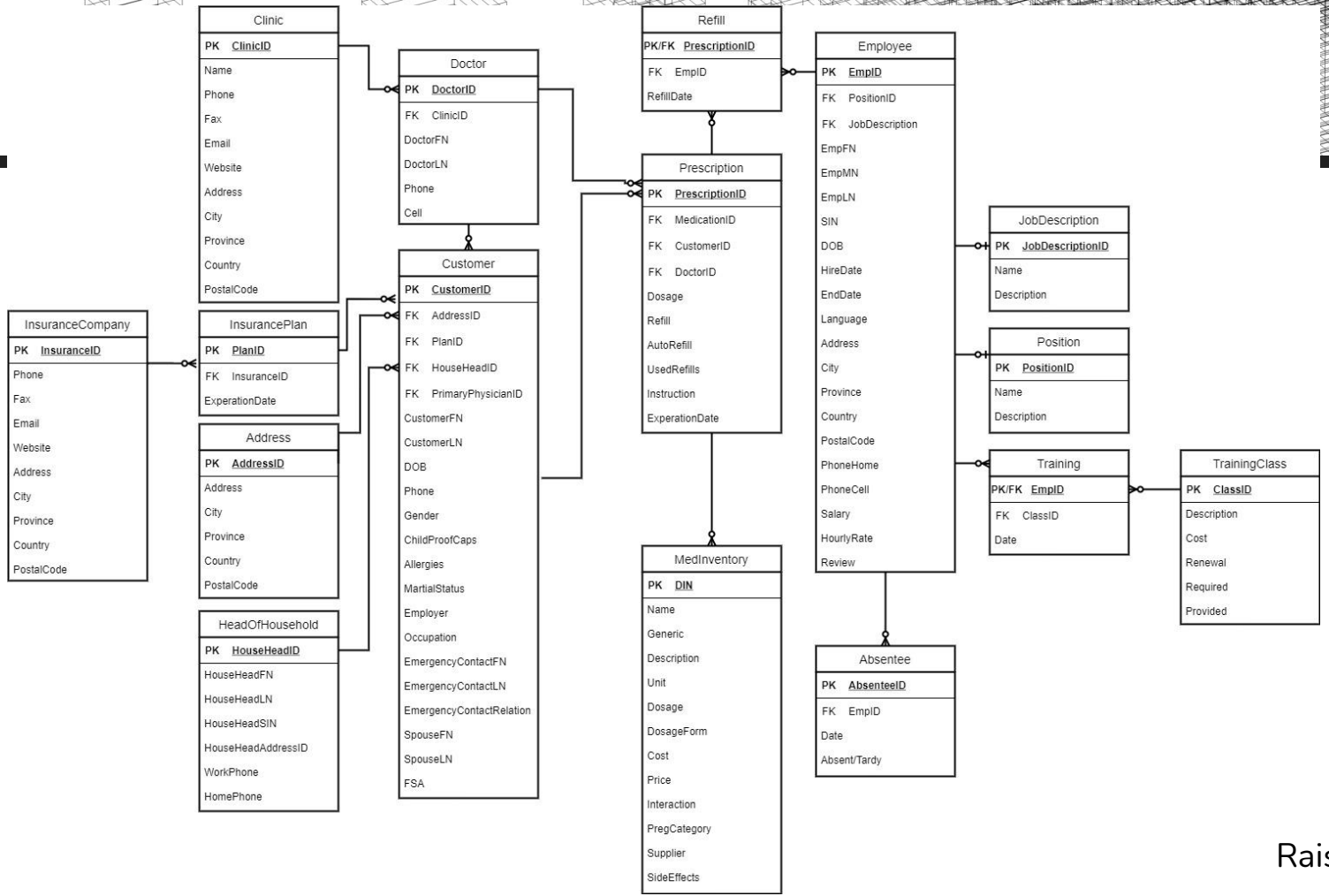
- For Client:
  - To collect all the data from different sources
  - To adapt the database for Canada
  - To automate and secure data manipulations
- Personal:
  - To practice creating and manipulating database
  - To acquire new skills manipulating database
  - To increase the communication level in team
  - To improve the competence of working in team

Steps we took

- Created Design
- Created Dictionary and User defined Data Types
- Created Schema
- Created Tables
- Created Constraints
- Created Data
- Updated Data
- Uploaded Data
- Created Indexes
- Created Functions
- Created Views
- Created Triggers

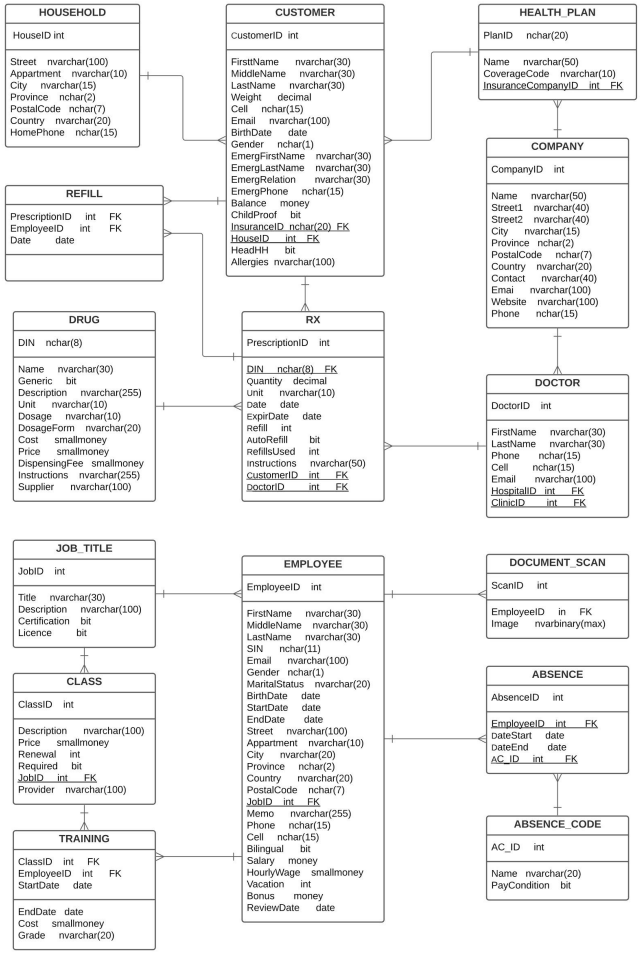
Design Process

- Steps One to Six were created individually
- United four diagrams into one
- Updated Dictionary according to new diagram and given data

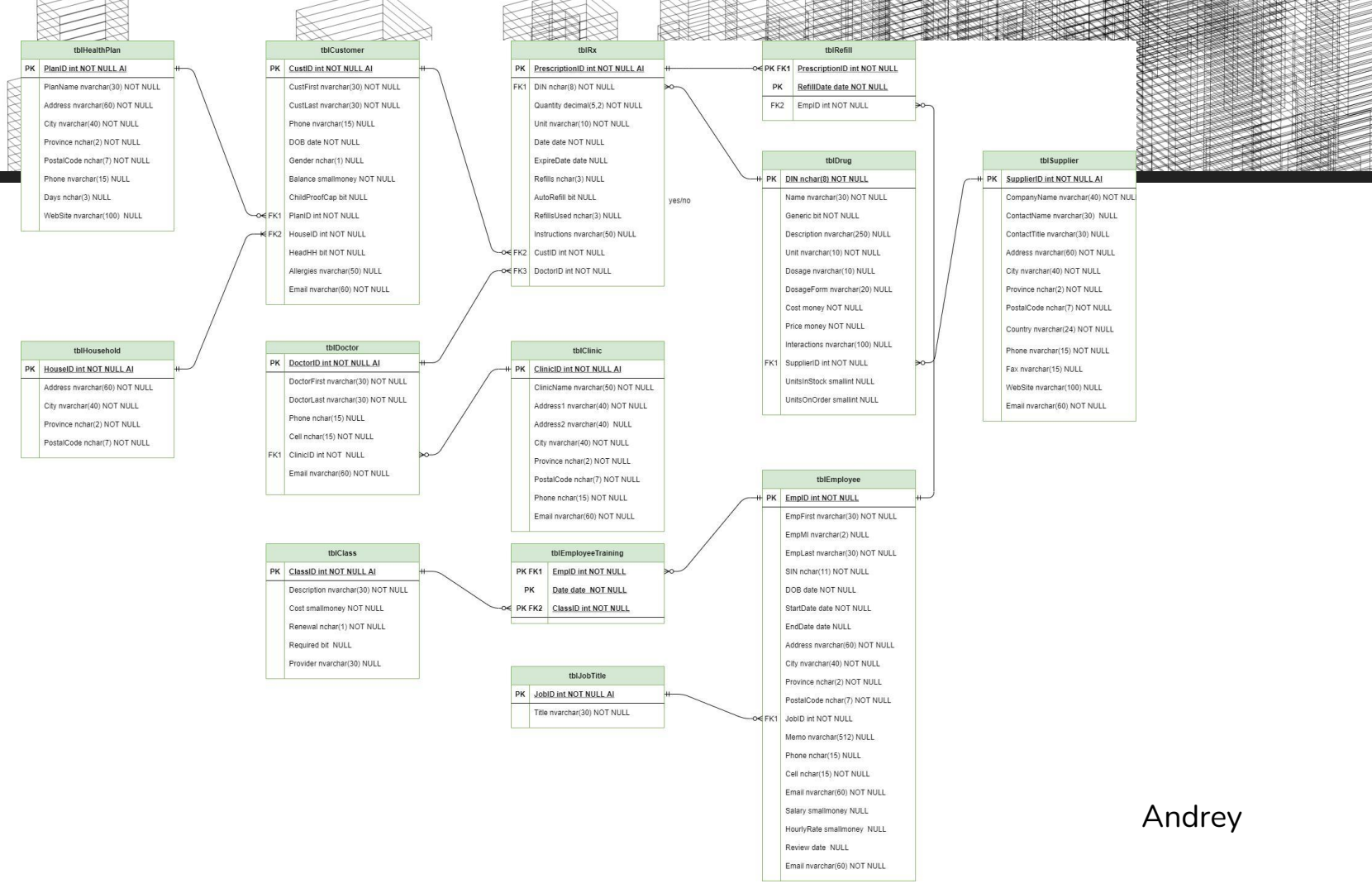

Clinic
PK ClinicID
Name
Phone
Fax
Email
Website
Address
City
Province
Country
PostalCode
Doctor
PK DoctorID
FK ClinicID
DoctorFN
DoctorLN
Phone
Cell
Refill
PK/FK PrescriptionID
FK EmpID
RefillDate
Employee
PK EmpID
FK PositionID
FK JobDescription
EmpFN
EmpMN
EmpLN
SIN
DOB
HireDate
EndDate
Language
Address
City
Province
Country
PostalCode
PhoneHome
PhoneCell
Salary
HourlyRate
Review
Prescription
PK PrescriptionID
FK MedicationID
FK CustomerID
FK DoctorID
Dosage
Refill
AutoRefill
UsedRefills
Instruction
ExperationDate
JobDescription
PK JobDescriptionID
Name
Description
Position
PK PositionID
Name
Description
Customer
PK CustomerID
FK AddressID
FK PlanID
FK HouseHeadID
FK PrimaryPhysicianID
CustomerFN
CustomerLN
DOB
Phone
Gender
ChildProofCaps
Allergies
MartialStatus
Employer
Occupation
EmergencyContactFN
EmergencyContactLN
EmergencyContactRelation
SpouseFN
SpouseLN
FSA
InsuranceCompany
PK InsuranceID
Phone
Fax
Email
Website
Address
City
Province
Country
PostalCode
InsurancePlan
PK PlanID
FK InsuranceID
ExperationDate
Address
PK AddressID
Address
City
Province
Country
PostalCode
HeadOfHousehold
PK HouseHeadID
HouseHeadFN
HouseHeadLN
HouseHeadSIN
HouseHeadAddressID
WorkPhone
HomePhone
MedInventory
PK DIN
Name
Generic
Description
Unit
Dosage
DosageForm
Cost
Price
Interaction
PregCategory
Supplier
SideEffects
Training
PK/FK EmpID
FK ClassID
Date
TrainingClass
PK ClassID
Description
Cost
Renewal
Required
Provided
Absentee
PK AbsenteeID
FK EmpID
Date
Absent/Tardy

Raisa

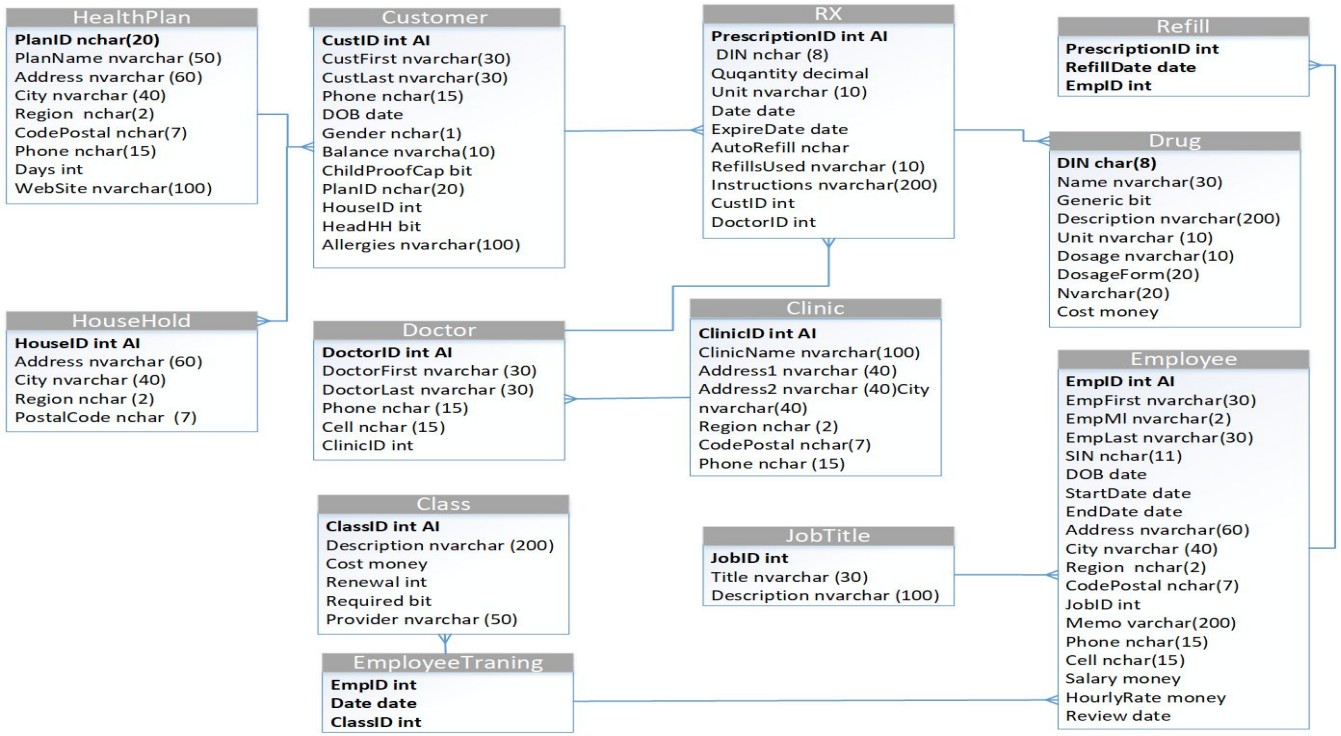
ARAGON PHARMACY - ERD



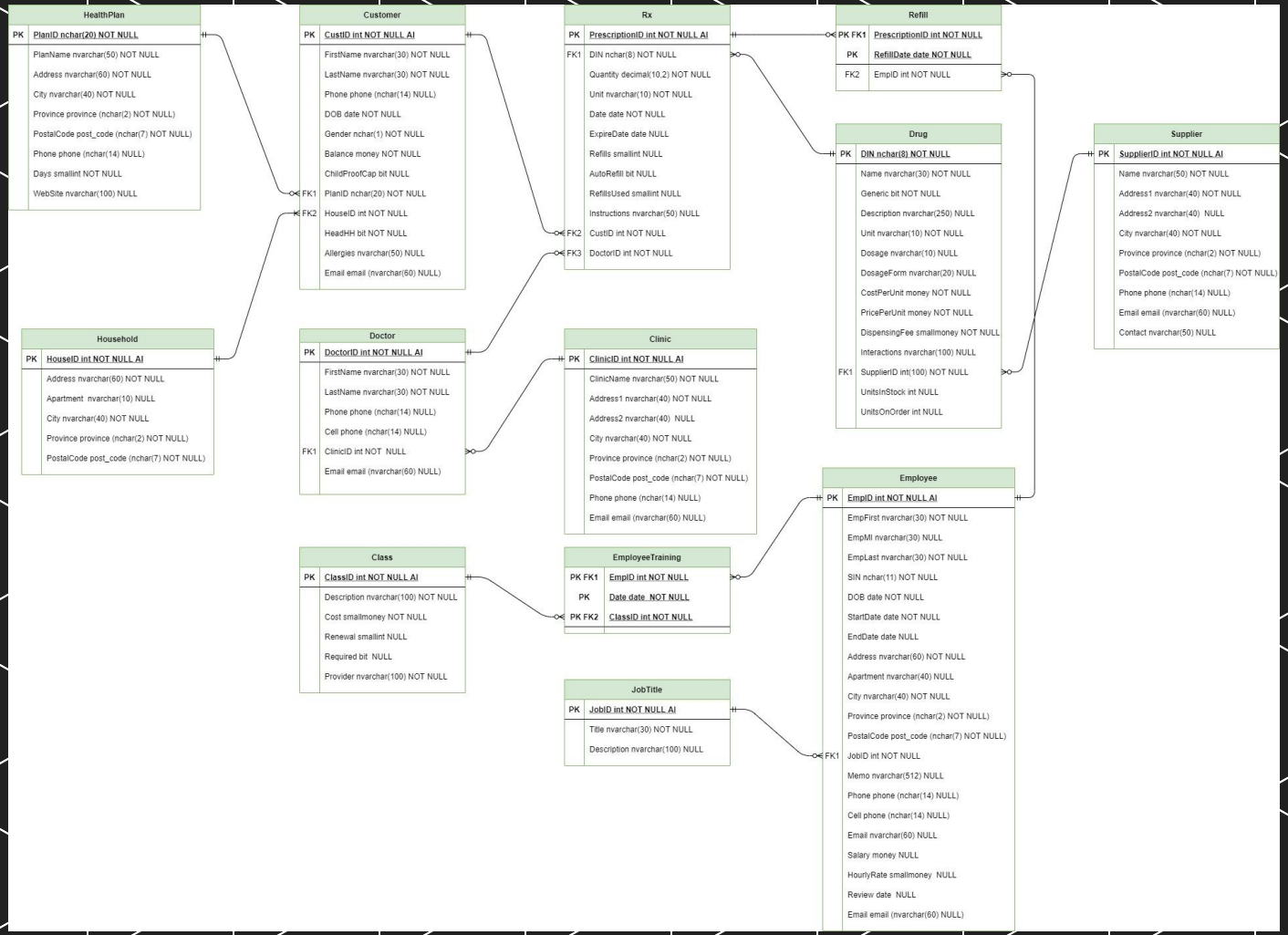
Anastassia


tblHealthPlan
PK PlanID int NOT NULL AI
PlanName nvarchar(30) NOT NULL
Address nvarchar(60) NOT NULL
City nvarchar(40) NOT NULL
Province nchar(2) NOT NULL
PostalCode nchar(7) NOT NULL
Phone nvarchar(15) NULL
Days nchar(3) NULL
WebSite nvarchar(100) NULL
tblCustomer
PK CustID int NOT NULL AI
CustFirst nvarchar(30) NOT NULL
CustLast nvarchar(30) NOT NULL
Phone nvarchar(15) NULL
DOB date NOT NULL
Gender nchar(1) NULL
Balance smallmoney NOT NULL
ChildProofCap bit NULL
FK1 PlanID int NOT NULL
FK2 HouseID int NOT NULL
HeadHH bit NOT NULL
Allergies nvarchar(50) NULL
Email nvarchar(60) NOT NULL
tblRx
PK PrescriptionID int NOT NULL AI
FK1 DIN nchar(8) NOT NULL
Quantity decimal(5,2) NOT NULL
Unit nvarchar(10) NOT NULL
Date date NOT NULL
ExpireDate date NULL
Refills nchar(3) NULL
AutoRefill bit NULL
RefillsUsed nchar(3) NULL
Instructions nvarchar(50) NULL
FK2 CustID int NOT NULL
FK3 DoctorID int NOT NULL
yes/no
tblRefill
PK FK1 PrescriptionID int NOT NULL
PK RefillDate date NOT NULL
FK2 EmpID int NOT NULL
tblDrug
PK DIN nchar(8) NOT NULL
Name nvarchar(30) NOT NULL
Generic bit NOT NULL
Description nvarchar(250) NULL
Unit nvarchar(10) NOT NULL
Dosage nvarchar(10) NULL
DosageForm nvarchar(20) NULL
Cost money NOT NULL
Price money NOT NULL
Interactions nvarchar(100) NULL
FK1 SupplierID int NOT NULL
UnitsInStock smallint NULL
UnitsOnOrder smallint NULL
tblSupplier
PK SupplierID int NOT NULL AI
CompanyName nvarchar(40) NOT NUL
ContactName nvarchar(30) NULL
ContactTitle nvarchar(30) NULL
Address nvarchar(60) NOT NULL
City nvarchar(40) NOT NULL
Province nchar(2) NOT NULL
PostalCode nchar(7) NOT NULL
Country nvarchar(24) NOT NULL
Phone nvarchar(15) NOT NULL
Fax nvarchar(15) NULL
WebSite nvarchar(100) NULL
Email nvarchar(60) NOT NULL
tblHousehold
PK HouseID int NOT NULL AI
Address nvarchar(60) NOT NULL
City nvarchar(40) NOT NULL
Province nchar(2) NOT NULL
PostalCode nchar(7) NOT NULL
tblDoctor
PK DoctorID int NOT NULL AI
DoctorFirst nvarchar(30) NOT NULL
DoctorLast nvarchar(30) NOT NULL
Phone nchar(15) NULL
Cell nchar(15) NOT NULL
FK1 ClinicID int NOT NULL
Email nvarchar(60) NOT NULL
tblClinic
PK ClinicID int NOT NULL AI
ClinicName nvarchar(50) NOT NULL
Address1 nvarchar(40) NOT NULL
Address2 nvarchar(40) NULL
City nvarchar(40) NOT NULL
Province nchar(2) NOT NULL
PostalCode nchar(7) NOT NULL
Phone nchar(15) NOT NULL
Email nvarchar(60) NOT NULL
tblEmployee
PK EmpID int NOT NULL
EmpFirst nvarchar(30) NOT NULL
EmpMI nvarchar(2) NULL
EmpLast nvarchar(30) NOT NULL
SIN nchar(11) NOT NULL
DOB date NOT NULL
StartDate date NOT NULL
EndDate date NULL
Address nvarchar(60) NOT NULL
City nvarchar(40) NOT NULL
Province nchar(2) NOT NULL
PostalCode nchar(7) NOT NULL
FK1 JobID int NOT NULL
Memo nvarchar(512) NULL
Phone nchar(15) NULL
Cell nchar(15) NOT NULL
Email nvarchar(60) NOT NULL
Salary smallmoney NULL
HourlyRate smallmoney NULL
Review date NULL
Email nvarchar(60) NOT NULL
tblClass
PK ClassID int NOT NULL AI
Description nvarchar(30) NOT NULL
Cost smallmoney NOT NULL
Renewal nchar(1) NOT NULL
Required bit NULL
Provider nvarchar(30) NULL
tblEmployeeTraining
PK FK1 EmpID int NOT NULL
PK Date date NOT NULL
PK FK2 ClassID int NOT NULL
tblJobTitle
PK JobID int NOT NULL AI
Title nvarchar(30) NOT NULL

Andrey


HealthPlan
PlanID nchar(20)
PlanName nvarchar (50)
Address nvarchar (60)
City nvarchar (40)
Region nchar(2)
CodePostal nchar(7)
Phone nchar(15)
Days int
WebSite nvarchar(100)
Customer
CustID int AI
CustFirst nvarchar(30)
CustLast nvarchar(30)
Phone nchar(15)
DOB date
Gender nchar(1)
Balance nvarcha(10)
ChildProofCap bit
PlanID nchar(20)
HouseID int
HeadHH bit
Allergies nvarchar(100)
RX
PrescriptionID int AI
DIN nchar (8)
Quqantity decimal
Unit nvarchar (10)
Date date
ExpireDate date
AutoRefill nchar
RefillsUsed nvarchar (10)
Instructions nvarchar(200)
CustID int
DoctorID int
Refill
PrescriptionID int
RefillDate date
EmpID int
Drug
DIN char(8)
Name nvarchar(30)
Generic bit
Description nvarchar(200)
Unit nvarchar (10)
Dosage nvarchar(10)
DosageForm(20)
Nvarchar(20)
Cost money
HouseHold
HouseID int AI
Address nvarchar (60)
City nvarchar (40)
Region nchar (2)
PostalCode nchar (7)
Doctor
DoctorID int AI
DoctorFirst nvarchar (30)
DoctorLast nvarchar (30)
Phone nchar (15)
Cell nchar (15)
ClinicID int
Clinic
ClinicID int AI
ClinicName nvarchar(100)
Address1 nvarchar (40)
Address2 nvarchar (40)City
nvarchar(40)
Region nchar (2)
CodePostal nchar(7)
Phone nchar (15)
Employee
EmpID int AI
EmpFirst nvarchar(30)
EmpMI nvarchar(2)
EmpLast nvarchar(30)
SIN nchar(11)
DOB date
StartDate date
EndDate date
Address nvarchar(60)
City nvarchar (40)
Region nchar(2)
CodePostal nchar(7)
JobID int
Memo varchar(200)
Phone nchar(15)
Cell nchar(15)
Salary money
HourlyRate money
Review date
Class
ClassID int AI
Description nvarchar (200)
Cost money
Renewal int
Required bit
Provider nvarchar (50)
JobTitle
JobID int
Title nvarchar (30)
Description nvarchar (100)
EmployeeTraning
EmpID int
Date date
ClassID int

Leonid

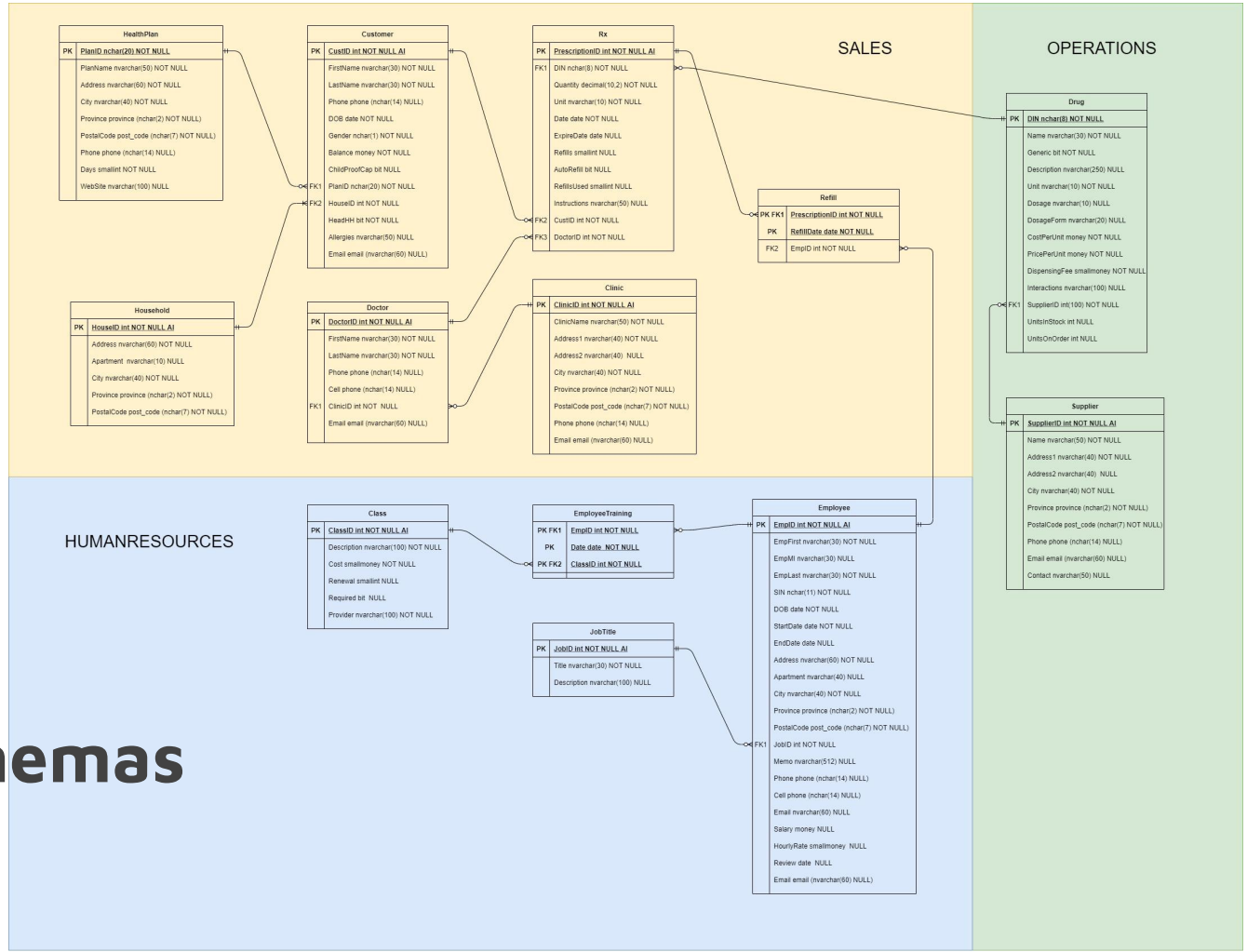

HealthPlan
PK PlanID nchar(20) NOT NULL
PlanName nvarchar(50) NOT NULL
Address nvarchar(60) NOT NULL
City nvarchar(40) NOT NULL
Province province (nchar(2) NOT NULL)
PostalCode post_code (nchar(7) NOT NULL)
Phone phone (nchar(14) NULL)
Days smallint NOT NULL
WebSite nvarchar(100) NULL
Customer
PK CustID int NOT NULL AI
FirstName nvarchar(30) NOT NULL
LastName nvarchar(30) NOT NULL
Phone phone (nchar(14) NULL)
DOB date NOT NULL
Gender nchar(1) NOT NULL
Balance money NOT NULL
ChildProofCap bit NULL
FK1 PlanID nchar(20) NOT NULL
FK2 HouseID int NOT NULL
HeadHH bit NOT NULL
Allergies nvarchar(50) NULL
Email email (nvarchar(60) NULL)
Rx
PK PrescriptionID int NOT NULL AI
FK1 DIN nchar(8) NOT NULL
Quantity decimal(10,2) NOT NULL
Unit nvarchar(10) NOT NULL
Date date NOT NULL
ExpireDate date NULL
Refills smallint NULL
AutoRefill bit NULL
RefillsUsed smallint NULL
Instructions nvarchar(50) NULL
FK2 CustID int NOT NULL
FK3 DoctorID int NOT NULL
Refill
PK FK1 PrescriptionID int NOT NULL
PK RefillDate date NOT NULL
FK2 EmpID int NOT NULL
Drug
PK DIN nchar(8) NOT NULL
Name nvarchar(30) NOT NULL
Generic bit NOT NULL
Description nvarchar(250) NULL
Unit nvarchar(10) NOT NULL
Dosage nvarchar(10) NULL
DosageForm nvarchar(20) NULL
CostPerUnit money NOT NULL
PricePerUnit money NOT NULL
DispensingFee smallmoney NOT NULL
Interactions nvarchar(100) NULL
FK1 SupplierID int(100) NOT NULL
UnitsInStock int NULL
UnitsOnOrder int NULL
Supplier
PK SupplierID int NOT NULL AI
Name nvarchar(50) NOT NULL
Address1 nvarchar(40) NOT NULL
Address2 nvarchar(40) NULL
City nvarchar(40) NOT NULL
Province province (nchar(2) NOT NULL)
PostalCode post_code (nchar(7) NOT NULL)
Phone phone (nchar(14) NULL)
Email email (nvarchar(60) NULL)
Contact nvarchar(50) NULL
Household
PK HouseID int NOT NULL AI
Address nvarchar(60) NOT NULL
Apartment nvarchar(10) NULL
City nvarchar(40) NOT NULL
Province province (nchar(2) NOT NULL)
PostalCode post_code (nchar(7) NOT NULL)
Doctor
PK DoctorID int NOT NULL AI
FirstName nvarchar(30) NOT NULL
LastName nvarchar(30) NOT NULL
Phone phone (nchar(14) NULL)
Cell phone (nchar(14) NULL)
FK1 ClinicID int NOT NULL
Email email (nvarchar(60) NULL)
Clinic
PK ClinicID int NOT NULL AI
ClinicName nvarchar(50) NOT NULL
Address1 nvarchar(40) NOT NULL
Address2 nvarchar(40) NULL
City nvarchar(40) NOT NULL
Province province (nchar(2) NOT NULL)
PostalCode post_code (nchar(7) NOT NULL)
Phone phone (nchar(14) NULL)
Email email (nvarchar(60) NULL)
Employee
PK EmpID int NOT NULL AI
EmpFirst nvarchar(30) NOT NULL
EmpMI nvarchar(30) NULL
EmpLast nvarchar(30) NOT NULL
SIN nchar(11) NOT NULL
DOB date NOT NULL
StartDate date NOT NULL
EndDate date NULL
Address nvarchar(60) NOT NULL
Apartment nvarchar(40) NULL
City nvarchar(40) NOT NULL
Province province (nchar(2) NOT NULL)
PostalCode post_code (nchar(7) NOT NULL)
FK1 JobID int NOT NULL
Memo nvarchar(512) NULL
Phone phone (nchar(14) NULL)
Cell phone (nchar(14) NULL)
Email nvarchar(60) NULL
Salary money NULL
HourlyRate smallmoney NULL
Review date NULL
Email email (nvarchar(60) NULL)
Class
PK ClassID int NOT NULL AI
Description nvarchar(100) NOT NULL
Cost smallmoney NOT NULL
Renewal smallint NULL
Required bit NULL
Provider nvarchar(100) NOT NULL
EmployeeTraining
PK FK1 EmpID int NOT NULL
PK Date date NOT NULL
PK FK2 ClassID int NOT NULL
JobTitle
PK JobID int NOT NULL AI
Title nvarchar(30) NOT NULL
Description nvarchar(100) NULL

User Defined Data Types

- Created data types:
  - Province
    - nchar(2)
  - Phone
    - nchar(14)
  - Post Code
    - (nchar(7)
  - Email
    - nvarchar(60)

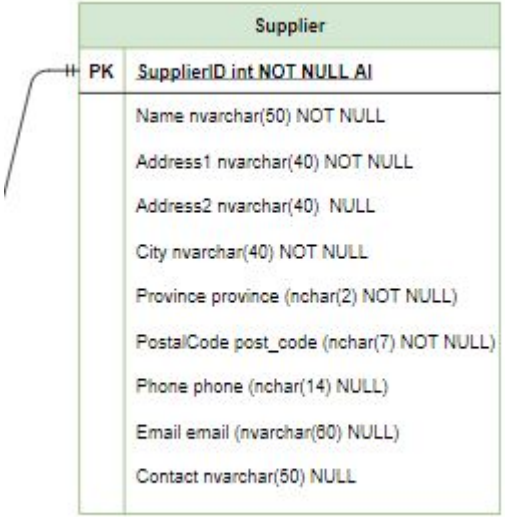
```
/* create Postal Code data type */
create type post_code
from nchar(7) not null
;
go

/* Create Email data type */
create type email
from nvarchar(60) null
;
go
```

Created Schemas



Created Tables

- Adapted fields for Canadian Regulations
- Added table Supplier



Constraints

- Constraints by Type:
  - Foreign key
  - Default: money, cost, refills = 0, province = ‘QC’, AutoRefill = false, HeadHH = false
  - Unique: Name, SIN, Title
  - Check:
    - Province in (‘QC’, ‘MB’....)
    - PostalCode like '[A-Z][0-9][A-Z] [0-9][A-Z][0-9]'
    - Phone like '([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'
    - Gender in ('M','F')

```sql
-- 2. Check Province in ('AB','BC','MB','NB','NL','NT','NS','NU','ON','PE','QC','SK','YT')
alter table HumanResources.Employee
    add constraint ck_Province_Employee check (Province in ('AB','BC','MB','NB','NL','NT','NS','NU','ON','PE','QC','SK','YT'))
;
go
```

```sql
-- 7. check Postal Code is entered as 'LNL NLN' */
alter table HumanResources.Employee
    add constraint ck_PostalCode_Employee check (PostalCode like '[A-Z][0-9][A-Z] [0-9][A-Z][0-9]')
;
go
```

Uploaded Data

- Adapted data to be uploaded to the tables
- Created missing files

```sql
/* ***** Table No. 1 - HumanResources.Class ***** */
BULK INSERT HumanResources.Class
FROM 'C:\data\Class.csv'
   WITH (
      FIELDTERMINATOR = ',',
      ROWTERMINATOR = '\n',
      FIRSTROW = 2
);
GO

-- reset identity to 0
DBCC CHECKIDENT ('HumanResources.Class', RESEED, 0);
GO

select * from HumanResources.Class
;
go
```

Functions

In the project scope:

- HumanResources.getSubstituteListFn
- HumanResources.yearsOfServiceFn

Extra for simplifying Queries and for View creation purposes:

- Sales.CustomerFullNameFn
- HumanResources.EmployeeFullNameFn

```sql
/* 2.  HumanResources.EmployeeFullNameFn
 Display Employee full name */
if OBJECT_ID('HumanResources.EmployeeFullNameFn', 'Fn') is not null
 drop function HumanResources.EmployeeFullNameFn
 ;
 go

create function HumanResources.EmployeeFullNameFn
 (
     @EmpID as int
 )
 returns nvarchar(150)
  as
     begin
         declare @FullName as nvarchar(150)
         select @FullName = concat_ws(', ', EmpLast, EmpMI, EmpFirst)
         from [HumanResources].[Employee]
         where EmpID = @EmpID
         return @FullName
     end
 ;
 go
```

Views

9 - views in the project scope

Addition:

- Views to analyze Salary
- Views to analyze Customers location and behaviour
- Operations.CheckStockOrderUnitsView

	Product Name	Unit	Units In Stock	Units On Order	Units In Stock Percentage	Units On Order Percentage
1	Ampicillin	Pill	340	60	85.00	15.00
2	Levothyroxine	Pill	480	69	87.43	12.57
3	Didanosine	Pill	260	0	100.00	0.00
4	Tvalaxec	Bottle	50	0	100.00	0.00
5	Rivastigmine tartrate	Pill	510	78	86.73	13.27

```sql
create view Operations.CheckStockOrderUnitsView
as
select
    D.Name AS 'Product Name',
    D.Unit as 'Unit',
    D.UnitsInStock as 'Units In Stock',
    D.UnitsOnOrder as 'Units On Order',
    FORMAT((CAST(D.UnitsInStock AS DECIMAL(5,2)) * 100/NULLIF((D.UnitsOnOrder + D.UnitsInStock),0)),'N2') AS 'Units In Stock Percentage',
    FORMAT((CAST(D.UnitsOnOrder AS DECIMAL(5,2)) * 100/NULLIF((D.UnitsOnOrder + D.UnitsInStock),0)),'N2') AS 'Units On Order Percentage'
from Operations.Drug AS D
;
go
```

Indexes

Created Indexes for:

- Table Drug, Field:
  - Name
- Table Supplier, Field:
  - City
- Table Customer, Field:
  - LastName
- Table HealthPlan, Field:
  - PlanName
- Table Employee, Field:
  - EmpLast
  - City
- Table Clinic, Field:
  - ClinicName
- Table Class, Field:
  - Provider
- Table Doctor, Field:
  - LastName

Example of Index

```sql
/* without index */
select LastName from Sales.Customer
;
go
```

Aggregate Status	
Connection failures	
Elapsed time	00:00:00.048
Finish time	2/18/2021 6:49:12 PM
Name	DESKTOP-RLUTOES
Rows returned	82
Start time	2/18/2021 6:49:12 PM
State	Open

```sql
/* after created index (LastName) */
create nonclustered index ncl_LastName_Customer on
  Sales.Customer (LastName);
go
select LastName from Sales.Customer;
go
```

Aggregate Status	
Connection failures	
Elapsed time	00:00:00.033
Finish time	2/18/2021 6:56:21 PM
Name	DESKTOP-RLUTOES
Rows returned	82
Start time	2/18/2021 6:56:21 PM
State	Open

Triggers

```
create trigger HumanResources.CheckRevewDateTR
on HumanResources.Employee
for insert, update
as
    begin
        -- declare variables
        declare @EmpID as int,
            @RevewDate as date
        -- compute the return value
        select @RevewDate = Review,
                @EmpID = EmpID
             inserted

            aking decision (comparing the Modified date with the current date
            abs(DateDiff(day, @RevewDate, getDate() )) > 0) or (@RevewDate is null)
            begin
                -- set the modified date to the current date
                update HumanResources.Employee
                set Review = getDate()
                where  EmpID = @EmpID
                print ' ***** The date was modified ***** '
            end
    end
;
go
```

```
(1 row affected)
 ***** The date was modified *****

(1 row affected)

Completion time: 2021-02-18T16:29:39.7260159-05:00
```

Triggers

```sql
begin
-- audit the EmployeeData old record
    insert into Sales.UpdateHouseHold
    (
        LogType,
        UpdateHouseID ,
        UpdateAddress,
        UpdateApartment,
        UpdateCity,
        UpdateProvince ,
        UpdatePostalCode
    )
    select
        'old',
        del.HouseID ,
        del.Address,
        del.Apartment,
        del.City,
        del.Province ,
        del.PostalCode
    from deleted as del
    union
    select
        'new',
        ins.HouseID ,
        ins.Address,
        ins.Apartment,
        ins.City,
        ins.Province ,
        ins.PostalCode
    from inserted as ins
end
```

LogType	UpdateHouseID	UpdateAddress	UpdateApartment	UpdateCity	UpdateProvince	UpdatePostalCode	ModifiedDate
old	2	21987 147 Street NE	NULL	Montreal	QC	H4E 2H7	2021-02-18 16:41:26.240
new	2	1555 Rue Duchesneau	NULL	Laval	QC	H7A 0A5	2021-02-18 16:41:26.240

Datetime Data Type - When How Why

When is easy. Everytime we have a date

How is it stored in a database? **How** is it formatted in a database?

Answer from the web: you can not format date in the DB

Why???

SQL Server stores the date/time values as one or more integers

The first integer represents the day and the second integer represents the time.

The days can range from January 1, 1753, through December 31, 9999.

The times can range from 00:00:00.000 through 23:59:59.997.

The default value being 1900-01-01 00:00:00.000.

"The proof of the pudding is in the eating"
English proverb

We have today's date.

Displaying Date as binary and integer, Time as binary and integer.

```sql
DECLARE @a DATETIME = '2021-02-19 10:00:00.187'
SELECT
    SUBSTRING(CONVERT(VARBINARY(8), @a), 1, 4) AS DateBinary,
    CAST(SUBSTRING(CONVERT(VARBINARY(8), @a), 1, 4) AS INT) AS DateInt,
    SUBSTRING(CONVERT(VARBINARY(8), @a), 5, 4) AS TimeBinary,
    CAST(SUBSTRING(CONVERT(VARBINARY(8), @a), 5, 4) AS INT) AS TimeInt;
```

The output of the previous code:

Results | Messages

	DateBinary	DateInt	TimeBinary	TimeInt
1	0x0000ACD4	44244	0x00A4CBB8	10800056

44244 / 365 = 121 years

1900 + 121 = 2021 year

44244 - 44195(all full years in days since 1900) = 49 - 31(january) = 18 February

10800056 /3000 mlsec/60 sec/60 min = 10.00 hours

Formatted Date

```sql
17 create view HumanResources.EmployeeTrainingView
18 as
19     select E.EmpID as 'EmployeeID',
20            HumanResources.EmployeeFullNameFn(E.EmpID) as 'Employee Name',
21            C.Description as 'Class Description',
22            format(ET.Date, 'MM/dd/yyyy', 'en-US') as 'Training Date',
23            format(DATEADD(year, C.Renewal, ET.Date ), 'MM/dd/yyyy', 'en-US') as 'Renewal Date',
24            C.Required as 'Class Required',
```

Results | Messages

	EmployeeID	Employee Name	Class Description	Training Date	Renewal Date	Class Required	Price	Class Provider
1	5	Gabel, S, Joan	NULL	NULL	NULL	NULL	NULL	NULL
2	19	Millbrandt, W, Janice	Adult CPR	09/01/2020	09/01/2020	1	15.00	Red Cross
3	19	Millbrandt, W, Janice	Adult CPR Recertification	02/05/2020	02/05/2021	1	10.00	Red Cross
4	19	Millbrandt, W, Janice	Defibrillator Use	06/15/2021	06/15/2022	1	25.00	Johnston Health Systems
5	6	Fujikawa, K, Karl	NULL	NULL	NULL	NULL	NULL	NULL
6	12	Foxhall, L, Desmond	NULL	NULL	NULL	NULL	NULL	NULL
7	2	Starkey, T, Phillip	NULL	NULL	NULL	NULL	NULL	NULL
8	13	Foxhall, M, Ava	Adult CPR Recertification	02/05/2020	02/05/2021	1	10.00	Red Cross

The hunt for ‘Global’ constraint

At the very beginning of our project we were looking for a constraint that you write once and apply to several tables.

The solution that we found was :

Binding **RULES** with **USER DEFINED DATA TYPES.**

The code

```
/*  Create data type ssn for Social Insurance Number (sin is a reserved word)*/
CREATE TYPE ssn
FROM varchar(11) NOT NULL ;
go

/*  Create test table */
create table dbo.Test
(
    SIN ssn
)
;
go
```

1. Create User Defined Data Type ssn
2. Create table Test

The code

```
/* create ssn rule like NNN-NNN-NNN */
Create rule ssn_rule
as @sin  like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]'
;
go

/* Bind the ssn rule and ssn data type */
EXEC sp_bindrule ssn_rule, 'ssn'
;
go
```

1. Create Rule for ssn
2. Bind the Rule and User Defined Data Type ssn

The end is a new beginning

Important

The **RULE** feature will be **removed** in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

It is possible to achieve the same result with functions and triggers. And we will certainly implement it in the next database

Conclusion

- Teamwork is the key to successful project
- Research is always required for better result