# Patient Appointment Application Project

Owner: Kyle Porter, Bao Dinh

Date: 06/05/17

# **Summary**

The company is creating an application that will allow patient appointments to be scheduled. The application will include the following database.

# **Expectations**

1. The application will store data in a SQL database.
2. The application must use views, stored procedures, and functions.
3. The application must be simple and keep costs at a minimum.
4. A working prototype should be available in a short period of time.

# Document History

Here is an account of changes to this document and any blocking issues.

## Change Log

|  |  |  |
| --- | --- | --- |
| Who | When | What |
| Kyle Porter | 05/26/17 | Added table names. |

## Open Issues

|  |  |  |
| --- | --- | --- |
| Who | When | What |
|  |  |  |

## Closed Issues

|  |  |  |
| --- | --- | --- |
| Who | When | What |
|  |  |  |
|  |  |  |

# Solution Naming Conventions

## Database (OLTP)

|  |  |  |  |
| --- | --- | --- | --- |
| Object Type | Convention | Example | Description |
| Table | TableName | Products | A table used to hold data |
| View | vViewName | vProducts | A saved select statement that provides data |
| Stored Proc. | pProcedureName | pInsProducts | A set of saved statements for data transactions |
| Primary Key Constraint | pkObjectName | pkProducts | A constraint to enforce Primary Key functionality |
| Foreign Key Constraint | fkObjectNameToObjectName | pkProductsToCategories | A constraint to enforce Foreign Key functionality |
| Check Constraint | ckObjectNamePurpose | ckProductStandardPriceGreaterThanZero | A constraint to enforce Domain integrity |
| Unique Constraint | uObjectName | uProductName | A constraint to enforce Unique values |

# **OLTP Database Tables**

## Table Name: Patients

Description: Contains patient data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Example | Notes |
| PatientID | int | 1 | PK, Identity (1,1) |
| FirstName | nvarchar(50) | Bob | Not Null |
| LastName | nvarchar(50) | Smith | Not Null |
| PhoneNumber | nvarchar(20) | 253-241-3421 | Not Null |
| Email | nvarchar(50) | imail@email.com | Not Null |
| Address | nvarchar(50) | Times Square 25 | Not Null |
| City | nvarchar(50) | New York | Not Null |
| State | char(2) | WA | Not Null |
| Zip | nvarchar(10) | 98303 | Not Null |

## Table Name: Doctors

Description: Contains doctor data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Example | Notes |
| DoctorID | int | 1 | PK, Identity (1,1) |
| FirstName | nvarchar(50) | Bob | Not Null |
| LastName | nvarchar(50) | Smith | Not Null |
| PhoneNumber | nvarchar(20) | 253-241-3421 | Not Null |
| Email | nvarchar(50) | imail@email.com | Not Null |

## Table Name: Appointments

Description: Contains patient data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Example | Notes |
| AppointmentID | int | 1 | PK, Identity (1,1) |
| PatientID | int | 2 | FK References dbo.Patients.PatientID |
| DoctorID | int | 3 | FK References dbo.Doctors.DoctorID |
| Datetime | datetime | 1/13/2014 12:00:00 AM | Not Null |

### Constraints

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Primary Key | Foreign Key | Not Null | Check | Unique | Default |
| ProductID | Yes | No | Yes | No | Yes | Yes |
| ProductName | No | No | Yes | No | Yes | No |
| ProductStandardPrice | No | No | No | Yes | No | No |
| CategoryID | No | Yes | Yes | No | No | No |

## Table Name: Clinics

Description: Contains clinic data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Example | Notes |
| ClinicID | int | 1 | PK, Identity (1,1) |
| ClinicName | nvarchar(100) | Cat1 | Unique |
| ClinicPhone |  |  |  |
| ClinicEmail |  |  |  |
| ClinicAddress |  |  |  |
| ClinicCity |  |  |  |
| ClinicState |  |  |  |
| ClinicZip |  |  |  |

### Constraints

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Primary Key | Foreign Key | Not Null | Check | Unique | Default |
| CategoryID | Yes | No | Yes | No | Yes | Yes |
| CategoryName | No | No | Yes | No | Yes | No |

# **OLTP Database Views**

## View Name: vProducts

Description: Selects product data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Source | Example | Notes |
| ProductID | Products.ProductID | 1 |  |
| ProductName | Products.ProductName | ProdA |  |
| ProductStandardPrice | Products.ProductStandardPrice | 9.99 |  |
| CategoryID | Products.CategoryID | 1 |  |

## View Name: vCategories

Description: Selects product data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Source | Example | Notes |
| CategoryID | Categories.CategoryID | 1 |  |
| CategoryName | Categories.CategoryName | Cat1 |  |

## View Name: vProductsByCategories

Description: Selects joined category and product data

### Columns

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Source | Example | Notes |
| CategoryID | Categories.CategoryID | 1 | Join Column |
| CategoryName | Categories.CategoryName | Cat1 |  |
| ProductID | Products.ProductID | 1 |  |
| ProductName | Products.ProductName | ProdA |  |
| ProductStandardPrice | Products.ProductStandardPrice | 9.99 |  |

# **OLTP Database Functions**

## View Name: fSelCategoryID

Description: Selects category ID based on category name

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @CategoryName | nvarchar(100) | Input | Cat1 |  |

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Type | Example | Notes |
| CategoryID | int | Result | 1 |  |

# **OLTP Database Stored Procedures**

## Stored Procedure Name: pValGreaterThanZero

Description: Validates a value is greater than zero

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @Value | float | Input | 1 |  |

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Type | Example | Notes |
| NA | float | Return | 100 | 100 = No error  -100 = Validation error |

## Stored Procedure Name: pSelProducts

Description: Selects product data

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @ProductID | int | Input | 1 | Defaults to All |
| @ProductName | nvarchar(100) | Input | ProdA | Defaults to All |
| @ProductStandardPrice | money | Input | 9.99 | Defaults to All |
| @CategoryID | int | Input | 1 | Defaults to All |

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Type | Example | Notes |
| NA |  | Return | 100 | 100 = No error  -100 = General error |
| ProductID | int | Column | 1 | From vProducts.ProductID |
| ProductName | nvarchar(100) | Column | ProdA | From vProducts.ProductName |
| ProductStandardPrice | money | Column | 9.99 | From vProducts.ProductStandardPrice |
| CategoryID | int | Column | 1 | From vProducts.CategoryID |

## Stored Procedure Name: pInsProducts

Description: Inserts product data

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @ProductID | int | Output | @ProdID | Returns Inserted ID using @@Identity |
| @ProductName | nvarchar(100) | Input | ProdA | Validated by uProductName |
| @ProductStandardPrice | money | Input | 9.99 | Validated by pValGreaterThanZero |
| @CategoryID | int | Input | 1 | Validated by fSelCategoryID |

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Type | Example | Notes |
| NA |  | Return | 100 | 100 = No error  -100 = General error  -200 = Product Name not unique  -300 = Price not greater than zero  -400 = Category ID not found |
| @ProductID | int | Output | 1 | Returns Inserted ID using @@Identity |

## Stored Procedure Name: pUpdProducts

Description: Updates product data

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @ProductID | int | Input | @ProdID |  |
| @ProductName | nvarchar(100) | Input | ProdA | Validated by uProductName |
| @ProductStandardPrice | money | Input | 9.99 | Validated by pValueGreaterThanZero |
| @CategoryID | int | Input | 1 | Validated by fSelCategoryID |

### Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Type | Example | Notes |
| NA |  | Return | 100 | 100 = No error  -100 = General error  -200 = Product Name not unique  -300 = Price not greater than zero  -400 = Category ID not found |

## Stored Procedure Name: pDelProducts

Description: Updates product data

### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter Name | Data Type | Type | Example | Notes |
| @ProductID | int | Input | @ProdID |  |

### Results

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Type | Example | Notes |
| NA | Return Code | 100 | 100 = No error  -100 = General error |