XML & SQL REVISION LABWORK

Details:

Introduction

In this revision session you will complete a small XML system and a related SQL database for booking a parking space. Part of the systems will be provided and you will need to complete them.

Booking data data will be set up and in an XML system and that data will be validated using XSDs before being loaded into a database that will be created using SQL and where it will be maintained in the web application.

The task

- Full details of the XML and SQL tasks are contained within the design specification below

DESIGN SPECIFICATION

Software environment

1. The SQL database must be implemented using SQL Server Management Studio with SQL Server 2012
2. The XML system must be implemented using Visual Studio 2010

This **system design** covers just two elements:

1. an XML data system where the parking space and customer data is created and validated before input to the database
2. An SQL database, designed to hold parking space, customer and booking tables

System design

This web system [not included here] enables owners of parking spaces to create a record of their space and allow for it to be booked by customers for a session. This session can be day time [8am-6pm] or night time [6pm-8am]. An owner can only have one space and that space can be booked by a customer for any single available [ie not booked] session.

The following table contains the data dictionary for the database and XML files:

Table: parkingspace – details of the space and its owner

column name	data type	null	constraint or description
spaceid	int	NO	PK
spacename	nvarchar(50)	NO	Name of owner
spaceaddress1	nvarchar(100)	NO	
spaceaddress2	nvarchar(100)	YES	
spaceaddress3	nvarchar(100)	NO	
spacepostcode	nvarchar(7)	NO	
spacedescription	nvarchar(100)	NO	Details of size, access etc

spacetelno	nvarchar(11)	NO	Owner contact no
spaceemail	nvarchar(50)	YES	Owner contact email
spacecomment	nvarchar(100)	YES	Owner comments
spacedaycost	money	NO	Cost for 8am to 6pm
spacenightcost	money	NO	Cost for 6pm to 8am
spaceupdatedate	datetime	NO	Default in database to current date

Table: customer – details of the people booking a space

column name	data type	null	constraint or description
customerid	int	NO	PK
custname	nvarchar(50)	NO	
custaddress1	nvarchar(100)	NO	
custaddress2	nvarchar(100)	YES	
custaddress3	nvarchar(100)	NO	
custpostcode	nvarchar(7)	NO	
custemail	nvarchar(50)	YES	
custtelno	nvarchar(11)	NO	
custcomment	nvarchar(100)	YES	
custupdatedate	datetime	NO	Default in database to current date

Table: booking – details of each space booking

column name	data type	null	constraint or description
bookingid	int	NO	PK – database system generated; identity, seed = 1000 increment = 3
spaceid	int	NO	FK – references parkingspace table
customerid	int	NO	FK - references customer table
bookingdate	smalldatetime	NO	Default to today's date
bookingstatus	nvarchar(10)	NO	values 'reserved', 'confirmed' or 'paid'
bookingcost	money	NO	Cost of this booking taken from the relevant value in the parkingspace record
bookingtype	nvarchar(5)	NO	Values 'day' or 'night'
bookingrequired	datetime	NO	Must be > bookingdate
bookingcomment	nvarchar(100)	YES	
bookingupdatedate	datetime	NO	Default to current date

1. **XML data system**

Create an XML system to hold two data sets, each of which will be held in a separate XML file:

- Parkingspace – holds space and owner details

- Customer – holds space customer details

Use the detail contained in the data dictionary above to define your XML file structures.

The XML files must each be validated against their own XSD [XML Schema] before the data is loaded into the database.

- **XML files**
    - Create an XML file structure for the parkingspace and customer data sets, using the detail contained in the data dictionary for that data set [table]
    - Create a suitable set of test data for each XML file – you will need enough rows to ensure data is related and to fully test your XSD and order processing website, with a *minimum* of
        - 2 parkingspaces
        - 3 customers

- **XSD validation -** create separate XML Schemas to validate data sets as follows
    - parkingspace data
        - must only include the elements described in the data dictionary
        - spaceid must be numeric, unique and between 11 and 99 inclusive
        - name must be an alphanumeric string and maximum 50 characters
        - address lines, description and comment must be alphanumeric strings and maximum 100 characters
        - postcode must be an alphanumeric string and maximum 7 characters
        - telno must be numeric, start '0121', with a total of 11 digits
        - daycost must be numeric and between 4.99 and 9.99 inclusive
        - nightcost must be between 3.00 and 6.99 inclusive
        - email, address2 and updatedate can be omitted, but if included must be valid strings

    - customer data
        - must only include the elements described in the data dictionary
        - customerid must be numeric, unique and between 101 and 199 inclusive
        - name must be an alphanumeric string and maximum 50 characters
        - address lines must be alphanumeric strings and maximum 100 characters
        - postcode must be an alphanumeric string and maximum 7 characters
        - telno must be numeric, start '07', with a total of 11 digits
        - email, address2 and updatedate can be omitted, but if included must be valid strings

    - correct all validation errors, after taking screen shots showing these errors and before the data is loaded into the database

2. database design

There are two tasks here:

- create a new database named **parkingdatabase**
- Create the three tables in the database structure defined by the data dictionary **using SQL create and then alter scripts**.

- Then import valid XML data into the database
    - NOTE that XML and SQL valid date formats are NOT compatible, so after validating against the XSD and before importing into the database, you need to change the date values from yyyy-mm-dd [XML] to mm/dd/yyyy [SQL]
    - in SSMS, create a modified version of the purpose built SQL script [that was used in the class lab work] for both of the XML files, customer and space
    - execute the scripts to load the XML data into the database
    - check that the data has loaded correctly by viewing tables in SSMS

*************** END OF DESIGN SPECIFICATION ******************