Database Management Systems (F28DM)
Coursework 2 – Semi-structured Data

Gem Dias, Navin Suresh Cordano, Varun Senthil Kumar

Project Fork on GitLab

Coursework repository was forked by Gem Dias (@gd55).
Group Repository Link: https://gitlab-student.macs.hw.ac.uk/gd55/f28dm-cw2

Work Distribution

Gem:
Data for hires 1-4 in data.xml; customer details (excluding driving license) in schema; custom data types for phone number and email; added keys and keyrefs to schema; fixed data for hires 1 and 2 when keys and keyrefs were added; XPath queries 1 and 2

Navin:
Data to fill the hire numbers 5,6 and 7 in the data.xml file; added the driving license details and hire price per day in schema; custom data type for driving license number; fixed data for hires 3-6 when keys and keyrefs were added; added query number 5 and 6.

Varun:
Data for hires 8-10 in data.xml; vehicle and hire details (hire location and dates) in schema; custom data types for category and transmission type of vehicle; fixed data for hires 7-10 when keys and keyrefs were added; XPath queries 3 and 4

XML File Contents

The XML file contains a root element <hires> to represent all data related to hires. The root element has three types of nested elements – <hire>, <vehicle> and <customer>.

<hire> represents a single hire made for one vehicle by one customer. A vehicle hired is recorded using a key reference to a <vehicle> element and the customer hiring the vehicle is recorded using a key reference to a <customer> element. Details such as the start and end location of the hire, hire dates and the hire price per day are specific to a hire and stored as nested elements of <hire>. Each hire is uniquely identified using the @hireNo attribute.

<vehicle> represents a vehicle that can be hired from a Watt Motors branch. Details about the category of the vehicle (such as small car or luxury car) and the transmission type (automatic or manual) are stored as nested elements of <vehicle>. Each <vehicle> element also has a @id attribute which is used as a key to refer to the vehicle.

<customer> represents a customer of Watt Motors. Several details of the customer such as name and date of birth are recorded as nested elements of <customer>. A <customer> element can store zero to three phone numbers and optionally, an email address. A customer's driving license details are also stored as a nested element with individual details of the driving license nested in the <drivLicense> element. All driving license details except for issue place are required. Each <customer> element has a @id attribute which is used as a key to refer to the vehicle.

The structure described above is enforced by the schema. It includes custom data types for elements such as <category>, <licenseNo> and <email>. It also defines the keys and keyrefs for vehicle and customer IDs. A visual representation of the structure of XML file is included in Fig. 1.

The XML file represents a many-to-many relationship between vehicles and customers as a single vehicle can be hired by many customers and a single customer can hire many vehicles. This relationship is represented by the <hire> element.

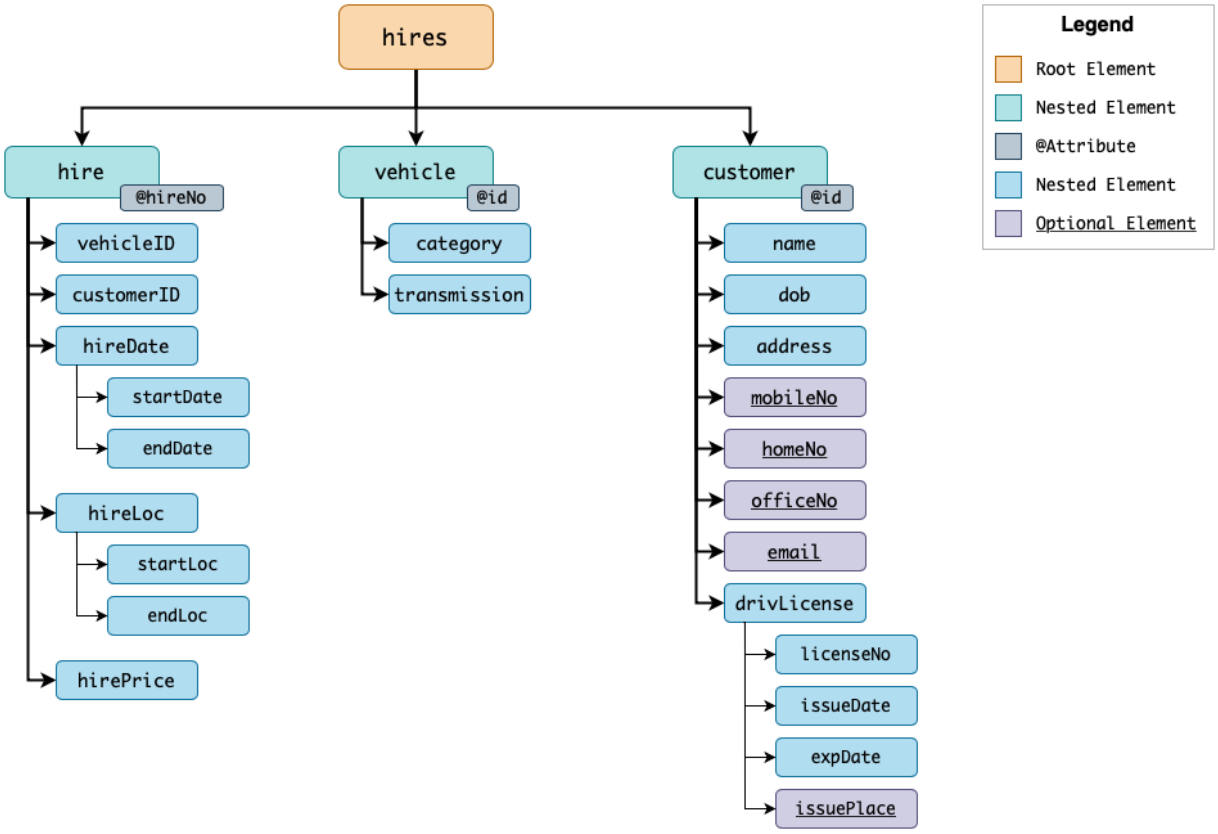

Fig. 1: Visual representation of structure of XML file

XPath Expressions

- Query 1

```
# Gets the emails of customers that live in London
//customer[contains(./address, 'London')]/email/text()
```

Output:

```
Kyleshelton96@gmail.com
Jbrown@gmail.com
```

Written By: Gem Dias

Description:
This query returns the email addresses of customers with addresses in London. This query is useful for sending marketing emails about promotions or deals available at a particular branch of Watt Motors.

- Query 2

```
# Gets the IDs of all vehicles that were hired from or returned to Watt Motors'
Liverpool branch
//hire[.//startLoc='Liverpool' or .//endLoc='Liverpool']/vehicleID
```

Output:

```
<vehicleID>V09199</vehicleID>
<vehicleID>V55341</vehicleID>
```

Written By: Gem Dias

Description:
This query returns the vehicle IDs of all vehicles that were either hired from or returned to Watt Motors' Liverpool branch. This is useful for tracking or locating a particular vehicle.

- Query 3

```
#Query to return the hire details of the booking with hireNo as 7
//hire[@hireNo="7"]
```

Output:

```
<hire hireNo="7">
    <vehicleID>V09199</vehicleID>
    <customerID>C54212</customerID>
    <hireDate>
      <startDate>2021-01-01</startDate>
      <endDate>2021-01-05</endDate>
    </hireDate>
    <hireLoc>
      <startLoc>London</startLoc>
      <endLoc>Manchester</endLoc>
    </hireLoc>
    <hirePrice>100</hirePrice>
  </hire>
```

Written By: Varun Senthil Kumar

Description:
Query to return the hire details of the booking with hireNo as 7

- Query 4

```
# Query returns the count of all vehicles with transmission as manual
count(//vehicle[transmission='Manual'])
```

Output:

```
2
```

Written By: Varun Senthil Kumar

Description:
Query returns the count of all vehicles with transmission as manual

- Query 5

```
# Query will display the numbers of customers that Watt Motors had.
count(/hires/customer)
```

Output:

```
6
```

Written By: Navin Suresh Cordano

Description:
This XPath expression will tell Watt Motors the amount of customer that rented a car within the time that its was open.

- Query 6

```
# Query will display all the hires had their starting location as New York.
/hires/hire[hireLoc/startLoc='New York']
```

Output:

```
<hire hireNo="1">
    <!-- Vehicle reference -->
    <vehicleID>V00348</vehicleID>
    <!-- Customer reference -->
    <customerID>C04141</customerID>

    <!-- Hire date -->
    <hireDate>
      <startDate>2021-01-15</startDate>
      <endDate>2021-01-23</endDate>
    </hireDate>
    <!-- Hire location -->
    <hireLoc>
      <startLoc>New York</startLoc>
      <endLoc>Detroit</endLoc>
    </hireLoc>

    <!-- Hire price per day -->
    <hirePrice>117</hirePrice>
  </hire>
<hire hireNo="6">
    <vehicleID>V09192</vehicleID>
    <customerID>C44231</customerID>
      <hireDate>
        <startDate>2021-01-01</startDate>
        <endDate>2021-03-25</endDate>
      </hireDate>
      <hireLoc>
        <startLoc>New York</startLoc>
        <endLoc>California</endLoc>
      </hireLoc>
    <hirePrice>75</hirePrice>
```

```
    </hire>
<hire hireNo="10">
    <vehicleID>V00173</vehicleID>
    <customerID>C04141</customerID>
    <hireDate>
      <startDate>2021-02-01</startDate>
      <endDate>2021-02-05</endDate>
    </hireDate>
    <hireLoc>
      <startLoc>New York</startLoc>
      <endLoc>Washington</endLoc>
    </hireLoc>
    <hirePrice>300</hirePrice>
  </hire>
```

Written By: Navin Suresh Cordano

Description:
Query will display all the hires had their starting location as New York.

Testing XPath Queries
All of the above XPath queries were tested on the XPath tester available here:
https://www.freeformatter.com/xpath-tester.html