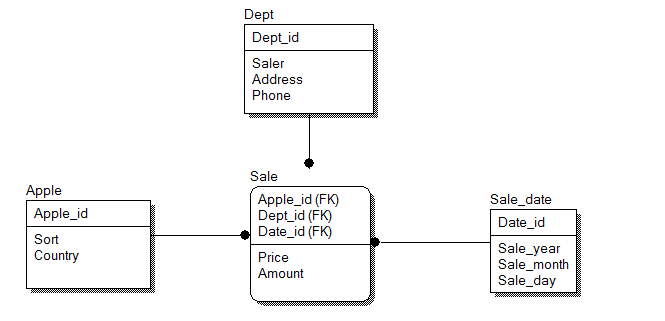
Отчет (в процессе)

1. Строение базы данных



Сущность Apple содержит поля: Apple\_id (PK), обозначающее идентификационный номер яблока, Sort -- cорт яблока, Country -- страна экспортер.

Сущность Dept содержит поля: Dept\_id (PK) -- идентификационный номер места продажи, Saler -- ФИО продавца, Address -- адрес точки продажи, Phone -- телефон продавца.

Сущность Sale\_date содержит поля: Date\_id (PK) -- идентификационный номер даты, Sale\_year -- год продажи, Sale\_month -- месяц продажи, Sale\_day -- день продажи.

Сущность Sale содержит поля: Apple\_id (FK) -- идентификационный номер яблока, Dept\_id (FK) -- идентификационный номер места продажи, Date\_id (FK) -- идентификационный номер даты, Price -- цена проданного яблока, Amount -- количество проданных яблок.

1. Описание XML документа.

Пример XML:

<Appledb path = "database\Apple.db3">

<Fixed>

<Apple>

<Apple\_id descr = "Ид яблока">1</Apple\_id>

</Apple>

</Fixed>

<Fact>

<Sale>

<Apple\_id relativeTo = "Apple"></Apple\_id>

<Date\_id relativeTo = "Sale\_date"></Date\_id>

<Dept\_id relativeTo = "Dept"></Dept\_id>

</Sale>

</Fact>

<Selection>

<Dept>

<Address descr = "Адрес точки продажи"></Address>

</Dept>

<Sale\_date>

<Sale\_year descr = "Год продажи"></Sale\_year>

<Sale\_month descr = "Месяц продажи"></Sale\_month>

<Sale\_day descr = "День продажи"></Sale\_day>

</Sale\_date>

</Selection>

</Appledb>

Первый тег Appledb указывает путь к базе данных с помощью атрибута path.

Фиксированная таблица описывается следующим образом:

<Apple>

<Apple\_id descr = "Ид яблока">1 2 3</Apple\_id>

</Apple>

Где теги <Apple></Apple> -- содержат имя таблицы в базе данных, <Apple\_id></Apple\_id> -- содержат идентификационный номер атрибута в данной таблицы, descr = “Ид яблока” – описание столбца, текст между тегами – значения которые фиксируются. Все фиксированные таблицы должны быть заключены между тегами <Fixed></Fixed>.

Таблица фактов описывается следующим образом:

<Sale>

<Apple\_id relativeTo = "Apple"></Apple\_id>

<Date\_id relativeTo = "Sale\_date"></Date\_id>

<Dept\_id relativeTo = "Dept"></Dept\_id>

</Sale>

Где теги <Sale></Sale> -- содержат имя таблицы в базе данных, <Apple\_id></Apple\_id>, <Date\_id></Date\_id> и так далее – содержат поля внешних ключей по которым будет происходить связь между таблицами, relativeTo = “Apple” и так далее – показывает с какой таблицей происходит связь. Таблица фактов должна быть заключена между тегами <Fact></Fact>.

Таблицы выборки описываются в виде:

<Dept>

<Address descr = "Адрес точки продажи"></Address>

</Dept>

Где теги <Dept></Dept> -- содержат имя таблицы в базе данных, <Address></Address> -- название столбца в данной таблице, descr =”Адрес точки продажи” – описание выводимого столбца. Все таблицы выборки должны быть заключены тегами <Selection></Selection>.

1. Описание считывания XML файла

Считывание файла XML происходит с помощью класса XmlTextReader (C#), используя следующий код:

while (reader.Read())

{

if (reader.NodeType == XmlNodeType.Element && reader.Name == "Appledb")

{

reader.MoveToNextAttribute();

section1.path = reader.Value;

}

if (reader.NodeType == XmlNodeType.Element && (reader.Name == "Fixed" || reader.Name == "Selection" || reader.Name == "Fact"))

{

if (reader.Name == "Fixed")

section1.xmlReadTable("Fixed", ref reader, ref section1.fixedTable, ref section1.fixedKol);

else if (reader.Name == "Selection")

section1.xmlReadTable("Selection", ref reader, ref section1.selectTable, ref section1.selectKol);

else if (reader.Name == "Fact")

section1.xmlReadTable("Fact", ref reader, ref section1.factTable, ref section1.factKol);

}

}

Если мы встречаем начальный тег “Appledb”, то из него извлекаем только аттрибут, который содержит путь к базе данных. Если же начальный тег равен “Fixed” или “Selection” или “Fact”, то вызываем функцию xmlReadTable, которая выглядит следующим образом:

public void xmlReadTable(string sost, ref XmlTextReader reader, ref allTable[] table, ref int kol)

{

reader.Read();

reader.Read();

while (reader.Name != sost)

{

string t = reader.Name;

if (t != "")

{

while (reader.Read() && reader.Name != t)

{

if (reader.NodeType == XmlNodeType.Element)

{

table[kol].table = t;

table[kol].column = reader.Name;

reader.MoveToNextAttribute();

table[kol].description = reader.Value;

}

else if (reader.NodeType == XmlNodeType.Text && reader.Value != "")

{

string[] split = reader.Value.Split(new Char[] { ' ' });

foreach (string s in split)

table[kol].val.Add(Convert.ToInt32(s));

}

else if (reader.NodeType == XmlNodeType.EndElement)

kol++;

}

}

reader.Read();

}

}

На вход функция получает: sost –имя встреченного тега, reader – ссылка на xmlTextReader, table – ссылка на массив структуры allTable, kol – ссылка на количество элементов в массиве структуры allTable. Чтение происходит пока мы не достигнем закрывающего тега с именем sost. Считываем внутренние теги, если тег является начальным, то заполняем table, указывая имя таблицы, столбец и описание. Если между тегами присутствует текст, то он должен представлять собой численные значения, разделенные пробелом, таким образом мы их все и считываем. Если тег является закрывающим, значит мы прочли полностью тег и можем увеличить значение kol на единицу.