

Министерство образования и науки РФ Федеральное государственное бюджетное
образовательное учреждение высшего профессионального образования
«Омский государственный университет им. Ф.М. Достоевского»
Кафедра компьютерных технологий и сетей

**Разработка программы для преобразования из
реляционной базы данных в многомерную для интернет-
магазина**

Выполнила: Кузина С.А.
Научный руководитель: Опарина Т.М.

Омск 2017

Цель и задачи

Цель работы – разработать приложение для интеграции реляционной базы данных интернет-магазина в многомерную.

Задачи, которые необходимо решить:

- Проанализировать средства создания интернет-магазинов.
- Проанализировать существующие способы аналитики продаж, определить необходимые данные для построения OLAP-куба.
- Сформулировать требования к приложению.
- Выбрать средства разработки.
- Разработать приложение.
- Проанализировать скорость выполнения запросов.

Анализ средств создания интернет-магазина

Способ	База данных	Достоинства	Недостатки
Конструкторы сайтов (SaaS)	MySQL	<ul style="list-style-type: none">- простота в использовании- скорость создания магазина	<ul style="list-style-type: none">- шаблонный дизайн- возможности строго ограничены
CMS (Content management system), ориентированные на создание интернет-магазинов	MySQL	<ul style="list-style-type: none">- индивидуальный дизайн- возможности не ограничены или почти не ограничены, в зависимости от выбора CMS	<ul style="list-style-type: none">- требуется больше времени для создания магазина, по сравнению с SaaS
Универсальные CMS	MySQL/ SQLite/ PostgreSQL/ MS SQL	<ul style="list-style-type: none">- индивидуальный дизайн- возможности не ограничены	<ul style="list-style-type: none">- требуется больше времени для создания магазина, по сравнению с предыдущими вариантами

Преимущества использования многомерных баз данных

- Упрощается формулировка запросов
- Уменьшается количество операций соединения таблиц при обработке запросов

Модель	Месяц	Объем
Honda	сентябрь	12
Honda	октябрь	24
Honda	ноябрь	5
Volvo	сентябрь	2
Volvo	октябрь	18
Audi	октябрь	19

Реляционная модель
представления данных


Модель	Сентябрь	Октябрь	Ноябрь
Honda	12	24	5
Volvo	2	18	No
Audi	No	19	No









Многомерная модель представления
данных






Аналитика продаж










Виды анализа	Цели анализа	Периодичность				
		Д	Н	М	К	Г
Динамика товарооборота, прибыли, сезонности продаж, удельных показателей продаж	Отслеживание тенденций продаж.	+	+	+	+	+
	Оперативная корректировка ассортимента, цен.					
	Оценка эффективности проведенных мероприятий.					
	Построение прогнозов продаж.					
Анализ товарных запасов	Выявление пробелов в ассортименте.		+	+		
	Оценка избыточности/недостатка товарных запасов.					
Сравнение продаж отчетного периода с предыдущими	Отслеживание динамики продаж по различным направлениям (точки продаж, товары, товарные группы, бренды и т. д.)			+	+	+
ABC-, XYZ-анализ	Определение приоритетных направлений развития.			+	+	+
	Перераспределение ресурсов.					5



Реляционная схема БД Интернет-Магазина







ProductCategories			
	CategoryId	int	
	CategoryName	varchar(30)	
 Add field			

Products			
	ProductId	int	
	ProductName	varchar(50)	
	ProductCategoryId	int	
	ProductCost	double	
	ProductDescr	varchar(200)	
	SupplierId	int	
 Add field			






Suppliers			
	SupplierId	int	
	SupplierName	varchar(50)	
	SupplierDescr	varchar(200)	
 Add field			





Orders			
	OrderId	int	
	UserId	int	
	OrderCost	double	
	OrderStatus	varchar(10)	
	OrderComment	varchar(200)	
	OrderDate	date	
	ShippingId	int	
	PaymentSystemId	int	
 Add field			

OrderProducts			
	OrderId	int	
	ProductId	int	
	Count	bigint	
 Add field			

Users			
	UserId	int	
	UserName	varchar(30)	
	UserPassword	varchar(50)	
	UserEmail	varchar(30)	
 Add field			

Shipping			
	ShippingId	int	
	ShippingCity	varchar(20)	
	ShippingAddress	varchar(200)	
	ShippingCost	double	
	ShippingDate	date	
	ShippingCompanyId	int	
 Add field			

ShippingCompanies			
	ShippingCompanyId	int	
	ShippingCompanyName	varchar(100)	
	ShippingCost	double	
 Add field			

PaymentSystem			
	PaymentSystemId	int	
	PaymentSystemName	varchar(20)	
 Add field			

Многомерная структура БД

Таблицы Измерений:

- {Поставщик: Поставщик | Группа товаров | Товар}
- {Страна: Страна | Город}
- {Время: Год | Квартал | Месяц | Неделя | День}

Таблица Фактов:

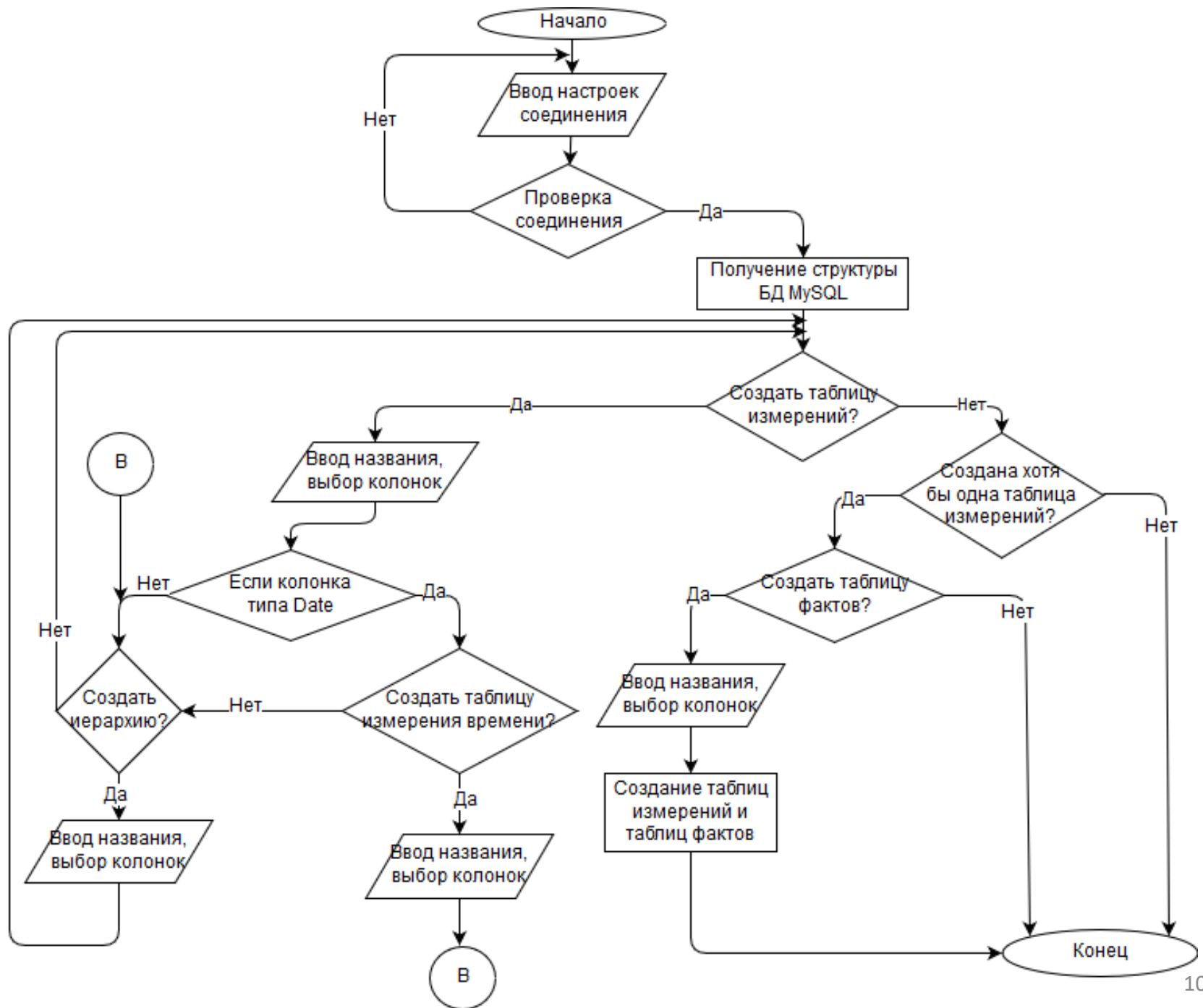
- ПоставщикКлюч;
- СтранаКлюч;
- ВремяКлюч;
- Сумма продаж.

Средства разработки

- Java SE 1.8
- Maven
- MySQL
- Oracle Database
- Oracle OLAP Java API

Организация данных

- Данные интернет-магазина хранятся в СУБД MySQL.
- С помощью Database Configuration Assistant будет создана база данных TEST (Oracle DB).
- Создание пользователя от имени SYS в SQL Developer:
 - CREATE USER test IDENTIFIED BY 123;
 - GRANT all privileges TO test;



Установка соединения

Настройки программы

Настройки соединения с MySQL

Сервер:

Порт:

База данных:

Имя пользователя:

Пароль:

Настройки соединения с Oracle Database

Сервер:

Порт:

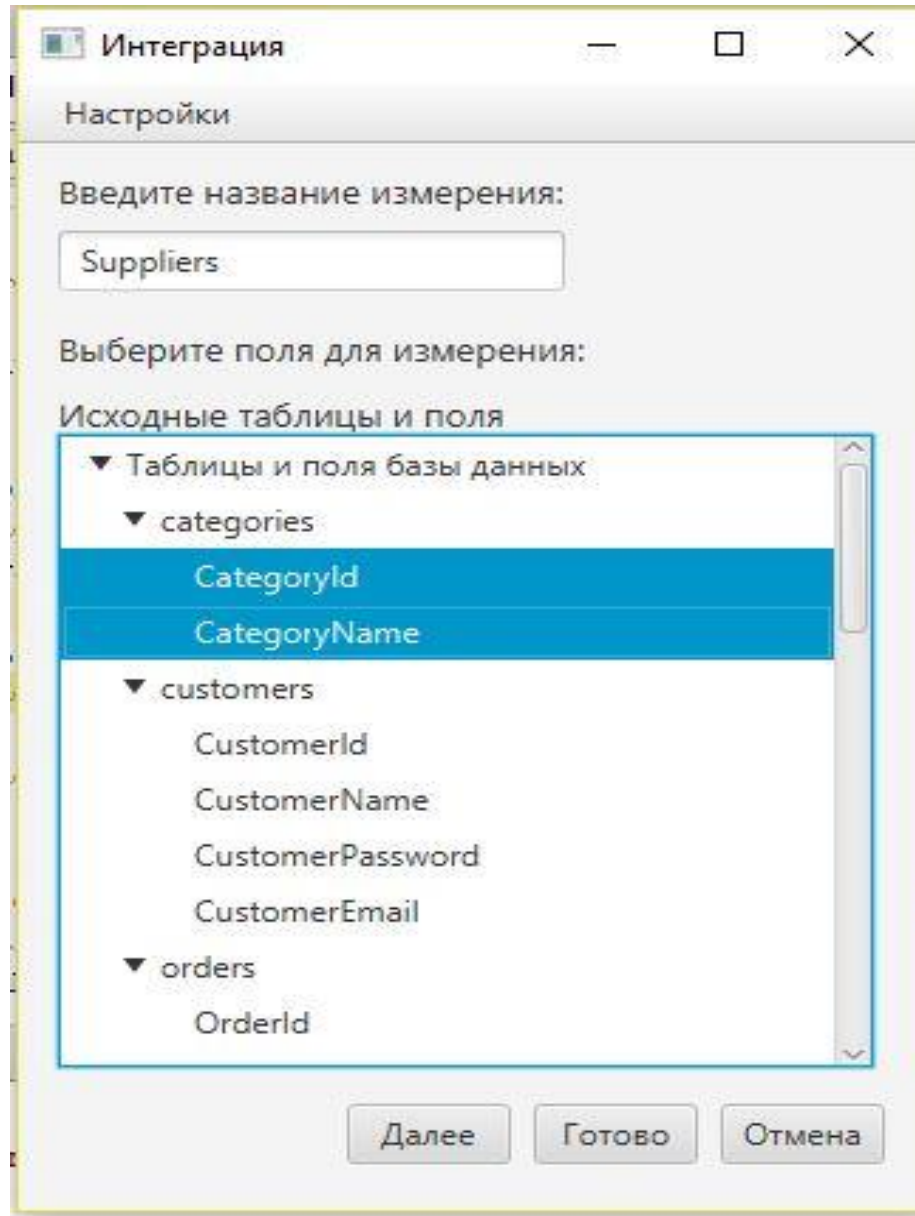
База данных:

Имя пользователя:

Пароль:

Ок Отмена

Создание таблицы измерения



Интеграция

Настройки

Введите название измерения:

Suppliers

Выберите поля для измерения:

Исходные таблицы и поля

- ▼ Таблицы и поля базы данных
 - ▼ categories
 - CategoryId
 - CategoryName
 - ▼ customers
 - CustomerId
 - CustomerName
 - CustomerPassword
 - CustomerEmail
 - ▼ orders
 - OrderId

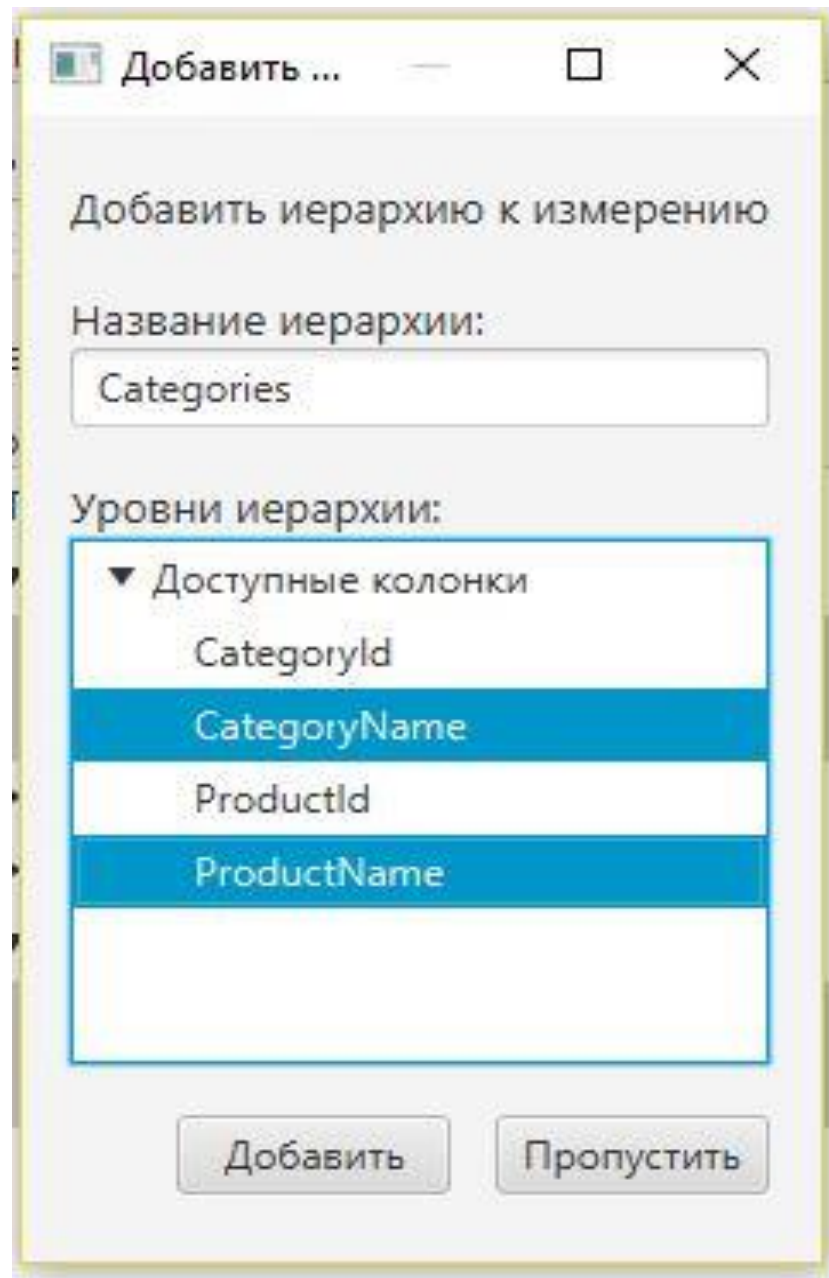
Далее Готово Отмена

Соответствие типов

```
public static Map<Integer,String> types = new  
HashMap<Integer, String>();  
types.put(1, "number");  
types.put(2, "varchar2");  
types.put(3, "date");  
types.put(4, "float");
```

```
// Для типа Varchar важно указать длину  
String fieldType = types.get(type);  
String fieldLength = "";  
if (type == 2) {  
    int length = typeLength.get(type);  
    fieldLength = "(" + length + " ";  
}
```

Создание иерархии



Добавить иерархию к измерению

Название иерархии:

Categories

Уровни иерархии:

▼ Доступные колонки

CategoryId
CategoryName
ProductId
ProductName

Добавить Пропустить

Создание таблицы фактов

Добавить ...

Создать таблицу фактов

Имя таблицы:

Sales

Колонки таблицы:

- CustomerId
- ProductCount
- ProductCost
- OrderCost
- OrderStatus
- ShippingId
- OrderDate
- products
- shipping

Готово

Созданные таблицы в Oracle Database

Oracle SQL Developer : test

File Edit View Navigate Run Versioning Tools Help

Connections x Reports x export.sql x test3 x test x TIMES x

Connections

- test
 - Tables (Filtered)
 - AW\$SALES_AW
 - COUNTRIES
 - COUNTRIESKEY
 - SHIPPINGCOUNTRY
 - SHIPPINGCITY
 - SALES_FACT
 - COUNTRIESKEY
 - SUPPLIERSKEY
 - TIMESKEY
 - ORDERCOST
 - SUPPLIERS
 - SUPPLIERSKEY
 - CATEGORYID
 - PRODUCTNAME
 - CATEGORYNAME
 - PRODUCTID
 - TIMES
 - TIMESKEY
 - YEAR
 - YEARID
 - MONTH
 - MONTHID
 - DAY
 - DAYID
 - Views
 - Editing Views

Worksheet Query Builder

```
select * from SUPPLIERS
```

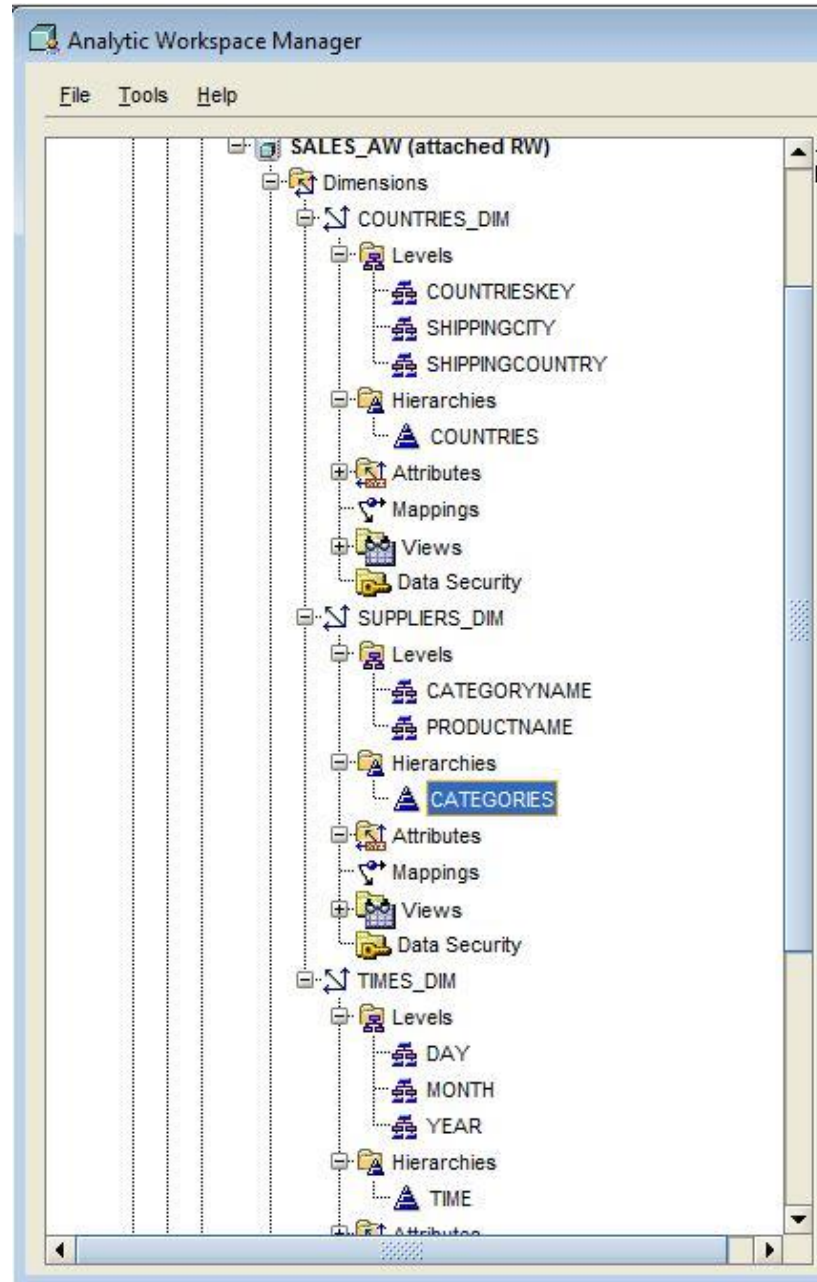
Script Output x Query Result x

SQL | All Rows Fetched: 91 in 0,017 seconds

	SUPPLIERSKEY	CATEGORYID	PRODUCTNAME	CATEGORYNAME	PRODUCTID
1	1	1	Чипсы яблочные	Еда	1
2	2	1	Мюсли мультизлак	Еда	2
3	3	1	Хлебцы мультизлак	Еда	3
4	4	2	Вода питьевая	Напитки	4
5	5	2	Сок апельсиновый	Напитки	5
6	6	2	Лимонад	Напитки	6
7	7	3	Тени для век	Косметика	7
8	8	3	Тушь для ресниц	Косметика	8
9	9	3	Крем для лица	Косметика	9

Messages - Log x

Созданные измерения и иерархии



Количество данных, в строках	Скорость выполнения запроса	
	Реляционная БД, мс	Многомерная БД, мс
10	9	9
20	31	31
30	55	54
40	91	89
50	130	128
60	178	175
70	247	243
80	279	275
90	365	359
100	378	374
200	680	674
300	1152	1146
400	1224	1218
500	1328	1321
600	1465	1456
700	1539	1530
800	1570	1560
900	1684	1674
1000	1790	1779

Заключение

- В результате выполнения работы была разработана программа для интеграции реляционной базы данных в многомерную.
- Программа подходит для любых баз данных, работающих в MySQL.
- Решены все поставленные задачи.
- Возможно дальнейшее развитие проекта.

Спасибо за внимание!