|  |  |  |
| --- | --- | --- |
|  | | |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования  «МИРЭА — Российский технологический университет»  РТУ МИРЭА | | |
|  | | |
| Институт информационных технологий (ИТ) | | |
| Кафедра инструментального и прикладного программного обеспечения (ИППО) | | |
| **ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2** | | |
| **по дисциплине**  «Хранилища данных и OLAP-технологии» | | |
|  | | |
| Выполнил студент группы ИКБО-02-15 | | Апальков П.Ю. |
| Принял | | Абаев Р. Г. |
| Работа выполнена | « » 2018 г. | *(подпись студента)* |
| Работа зачтена | « » 2018 г. | *(подпись руководителя)* |
| Москва 2018 | | |

**Содержание**

[Задание 4](#_Toc531542169)

[Выполнение работы 4](#_Toc531542170)

[Заключение 13](#_Toc531542171)

# Генерация хранилища данных и заполнение его записями

# Задание

В рамках данной лабораторной работы необходимо произвести генерацию структуры хранилища данных и заполнить его записями. В качестве исходных данных выступает база «northwind\_hd».

# Выполнение работы

Процесс создания хранилища данных начинается с проектирования его архитектуры. Существуют три типа технологии OLAP:

• многомерная OLAP (Multidimensional OLAP — MOLAP);

• реляционная OLAP (Relational OLAP — ROLAP);

• гибридная OLAP (Hybrid OLAP — HOLAP).

MOLAP — это классическая форма OLAP, так что её часто называют просто OLAP. Она использует суммирующую БД, специальный вариант процессора пространственных БД и создаёт требуемую пространственную схему данных с сохранением как базовых данных, так и агрегатов.

ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы.

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегатов.

Для создаваемого хранилища используется тип ROLAP: исходные данные и данные хранилища расположены в реляционной БД.

Основой для дизайна хранилища данных является денормализованная модель (хранилище с измерениями), составляющими которой являются таблица фактов (fact table) и таблицы измерений (dimension tables).

Таблица фактов является основной таблицей хранилища данных и, как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

- факты, связанные с транзакциями (Transaction facts). Они основаны на отдельных событиях (типичными примерами которых являются телефонный звонок или снятие денег со счета с помощью банкомата);

- факты, связанные с «моментальными снимками» (Snapshot facts). Основаны на состоянии объекта (например, банковского счета) в определенные моменты времени, например на конец дня или месяца. Типичными примерами таких фактов являются объем продаж за день или дневная выручка;

- факты, связанные с элементами документа (Line-item facts). Основаны на том или ином документе (например, счете за товар или услуги) и содержат подробную информацию об элементах этого документа (например, количестве, цене, проценте скидки);

- факты, связанные с событиями или состоянием объекта (Event or state facts). Представляют возникновение события без подробностей о нем (например, просто факт продажи или факт отсутствия таковой без иных подробностей).

Таблицы измерений содержат неизменяемые либо редко изменяемые данные. В подавляющем большинстве случаев эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении. Таблицы измерений также содержат как минимум одно описательное поле (обычно с именем члена измерения) и, как правило, целочисленное ключевое поле (обычно это суррогатный ключ) для однозначной идентификации члена измерения.

Одно измерение куба может содержаться как в одной таблице (в том числе и при наличии нескольких уровней иерархии), так и в нескольких связанных таблицах, соответствующих различным уровням иерархии в измерении. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название «звезда» (star schema). Если же хотя бы одно измерение содержится в нескольких связанных таблицах, такая схема хранилища данных носит название «снежинка» (snowflake schema).

С целью повышения скорости выполнения запросов к хранилищу данных в рамках данной лабораторной работы предпочтение отдается схеме «звезда».

Запустим MySQL Workbench и выберем на стартовой странице пункт «Create New EER Model» (рис.1).

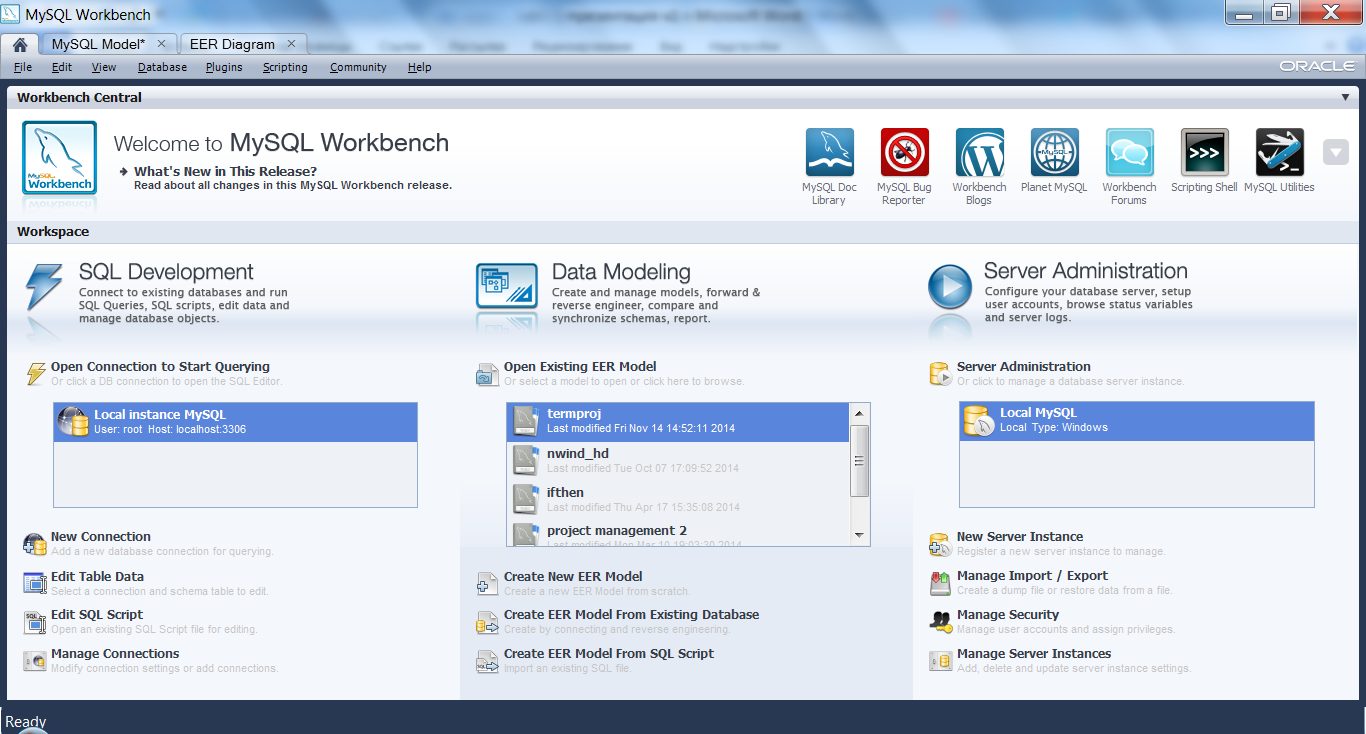


Рис.1. Начало работы с MySQL Workbench.

Откроется новая вкладка. Сохраните модель, выбрав пункт «File»-«Save model» главного меню под именем nwind\_hd. Также следует изменить название схемы на nwind\_hd, щёлкнув 2 раза по изображению цилиндра и введя новое название в появившемся поле (рис.2).

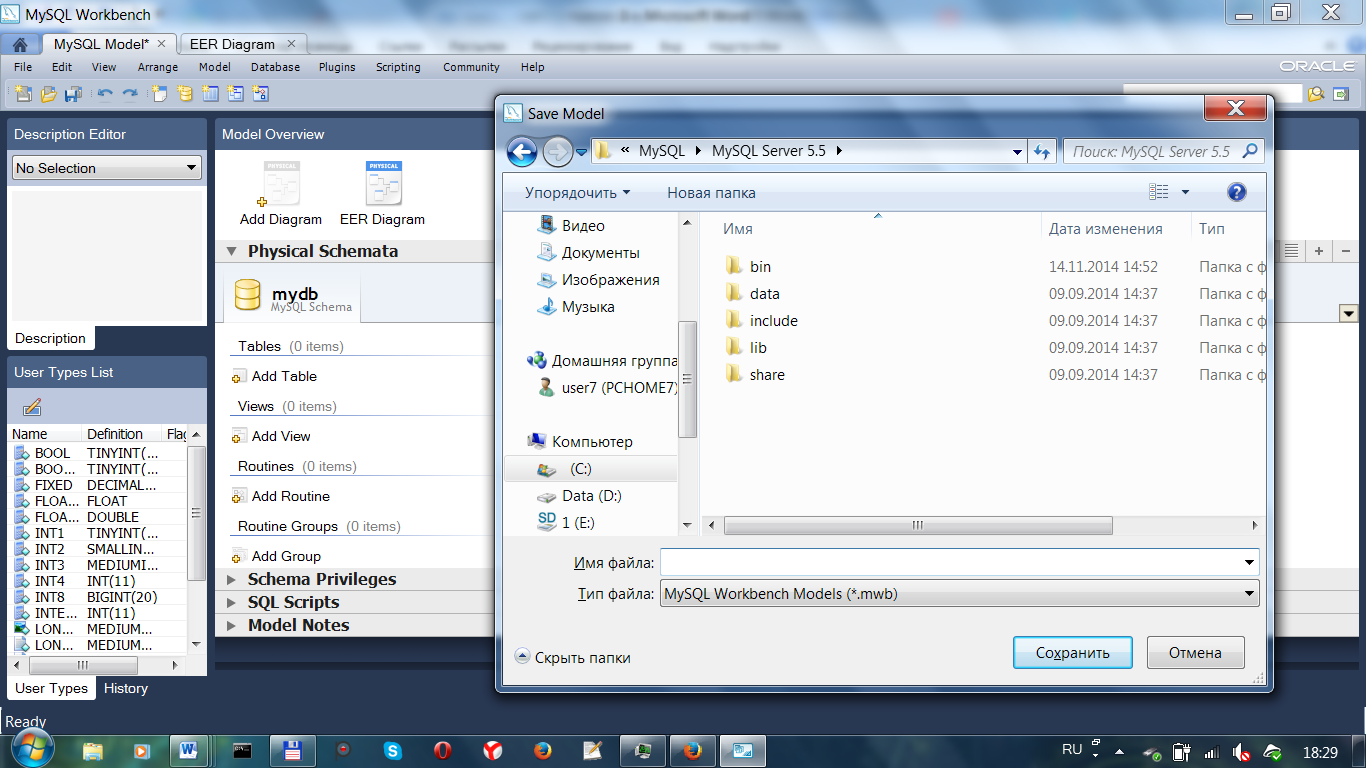


Рис.2. Окно сохранения модели.

Во вкладке выбираем «Add Diagram» («Создать новую диаграмму»). (рис.3)

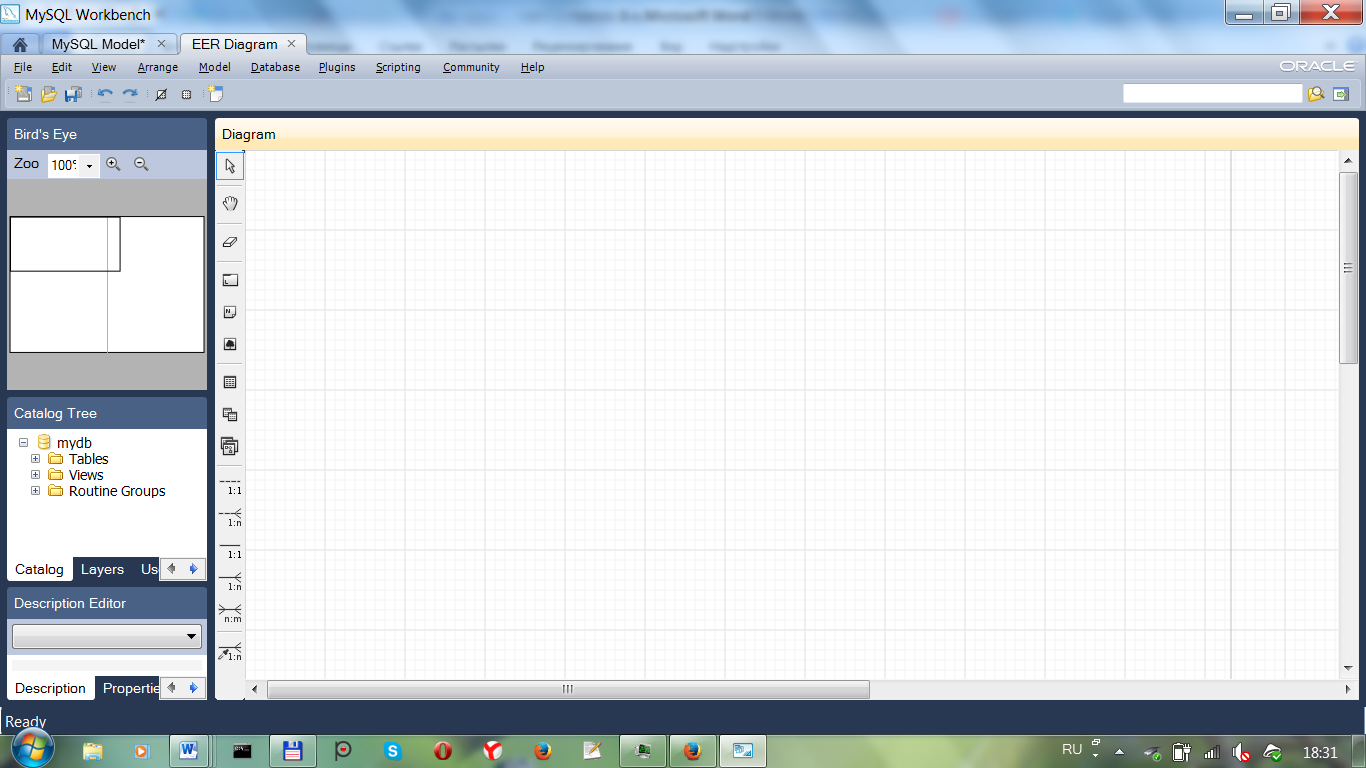


Рис.3. Добавление новой диаграммы.

C помощью инструментов добавления таблиц («Add Table» с вкладки «MySQL Model») и установления связей между ними (иконки в левой части рабочей области вкладки «Диаграмма») постройте основу хранилища – схему базы данных, как показано на рисунке 4:

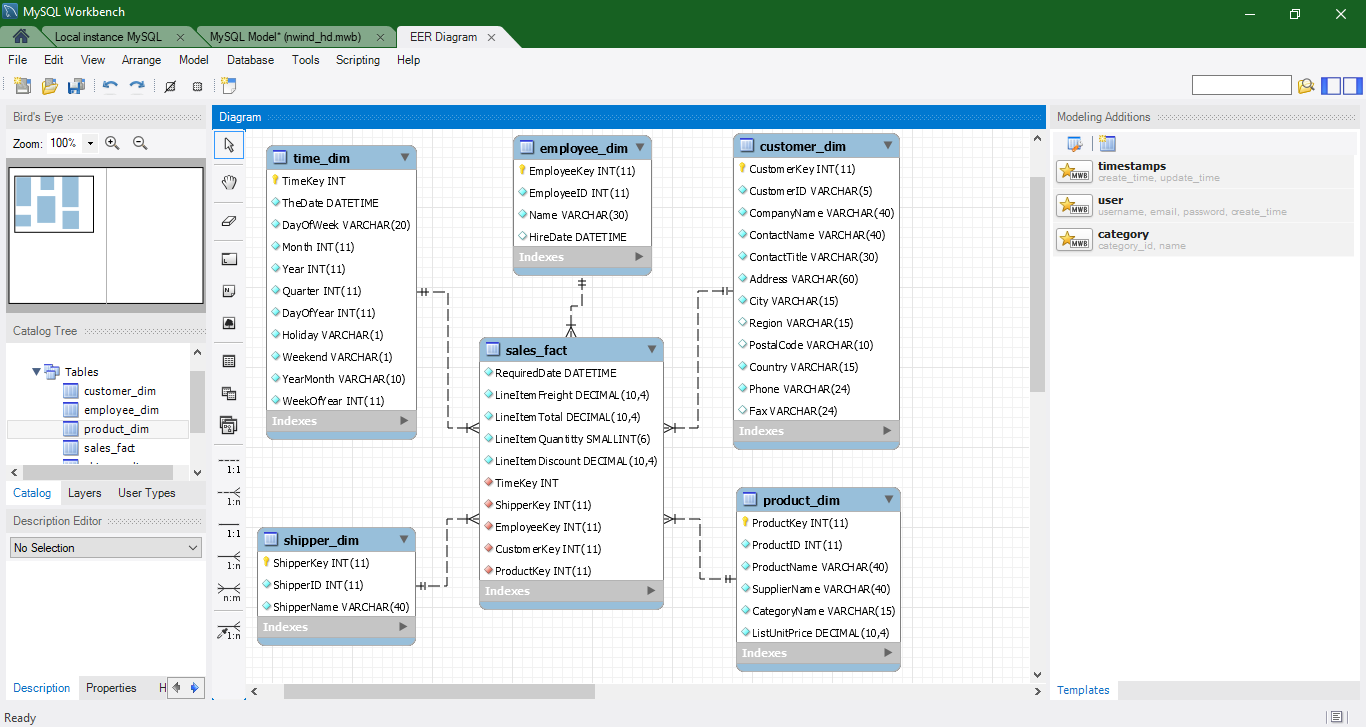


Рис.4. Схема хранилища данных.

Для этого нужно создать соответствующие колонки и правильно расставить флажки для каждой из них в окошке, появляющемся после выбора таблицы.

На диаграмме используются следующие обозначения:

- флажок NN (not null) установлен

- флажок NN снят

🔑 – флажок PK (primary key) установлен. Необходимо также установить флажок AI (auto increment) для каждого ключа, чтобы избежать ошибок на следующих этапах выполнения работы.

- данные колонки создавать не надо, они появятся сами после установления связей. Связи устанавливаются при помощи кнопки 1:n следующим образом: сначала выбирается таблица фактов, а затем ключ из определённой таблицы измерений.

Рассмотрим подробнее структуру таблиц приведенной выше схемы.

Таблицы измерений:

* Перевозчик: ключ – id Перевозчика – имя Перевозчика;
* Продукт: ключ – id продукта – наименование его – имя поставщика – наименование категории – цена единицы продукта;
* Покупатель: ключ – id – имя компании – контактное лицо – должность – адрес – город – регион – почтовый индекс – страна – телефон – факс;
* Работник: ключ – id – имя работника – дата приема на работу;
* Время: ключ – несколько полей, содержащих значения даты доставки заказа в различном формате (подробное описание данного измерения приведено далее).

Таблица фактов:

* 5 ключей таблиц измерений;
* `northwind\_hd`.`orders`.`requireddate` - предполагаемая дата заказа (из исходной таблицы Заказы);
* Плата за перевозку грузов морем – рассчитывается как `northwind\_hd`.`orders`.`Freight` \* `northwind\_hd`.`order details`.`Quantity` / (SELECT SUM(Quantity) FROM `northwind\_hd`.`order details` as od WHERE od.OrderID = `northwind\_hd`.`orders`.`OrderID`);
* Суммарная стоимость заказа – рассчитывается как произведение цены единицы товара на его количество из order\_details;
* Количество перевозимого товара – order\_details quantity;
* Скидка – рассчитывается как произведение размера скидки на цену единицы товара на его количество из order\_details.

Особое внимание здесь следует уделить измерению «Время». Члены измерений могут быть объединены одной или несколькими иерархиями. При построении OLAP-кубов по оси с датой заказа мы можем хотеть группировать точки (т.е. дни доставки заказов) по иерархии, например, *Год-Месяц-День*.

В рамках данной работы для измерения «Время» используется следующая иерархия (см.рисунок):

* Дата в длинном формате – ShippedDate (дата доставки заказа) из Orders (Заказы) исходной БД;
* Год – квартал – месяц - номер месяца - номер недели - день недели - день года – праздник - выходной – их значения рассчитываются на основе даты в длинном формате с помощью стандартных функций mysql date\_format(date,specification) (типа printf()), quarter(date) и week(date).

Для создания БД хранилища необходимо сгенерировать SQL-скрипт и выполнить его. Для этого в главном меню выберите пункт «Database»-«Forward Engineering». Перед Вами откроется диалоговое окно генерации и выполнения скрипта. Сначала Вам будет предложено установить параметры генерации таблиц, затем – выбрать объекты схемы, которые будут созданы в базе данных (рис.5-6).

Далее будет сгенерирован текст SQL-скрипта (рис.7). На следующем этапе выберите уже существующий экземпляр подключения к серверу MySQL или создайте новый (рис.8). Чтобы выполнить запрос, формирующий структуру хранилища, нажмите «Execute». В результате Вы увидите отчет о состоянии произведенного к серверу запроса (рис.9).

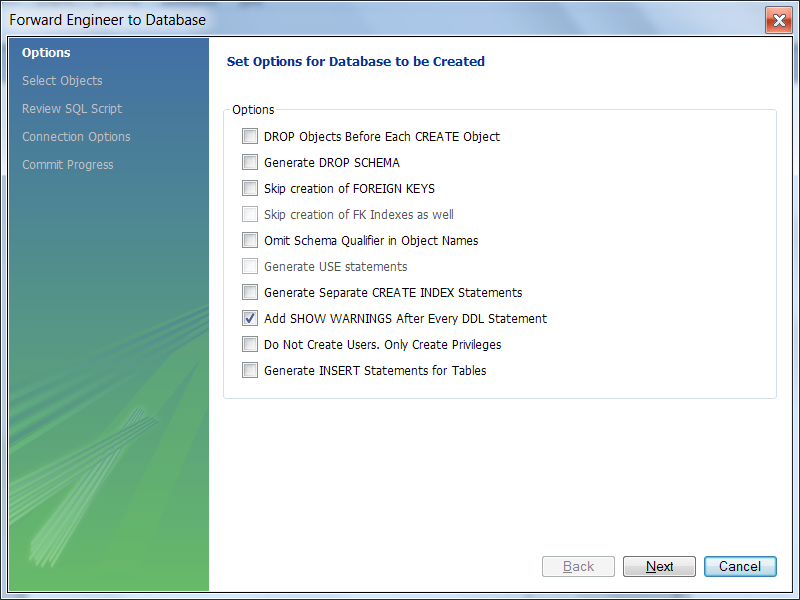


Рис.5. Настройка параметров выполнения скрипта.

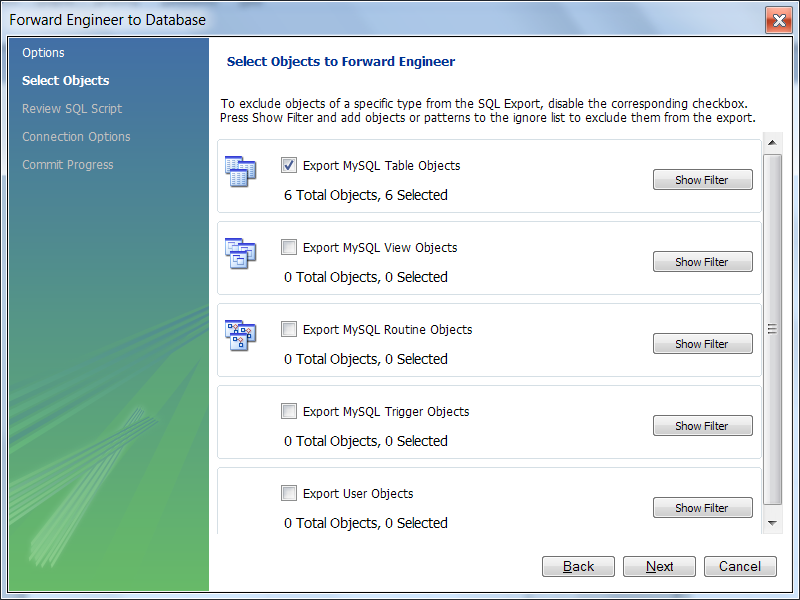


Рис.6. Выбор экспортируемых в базу объектов.

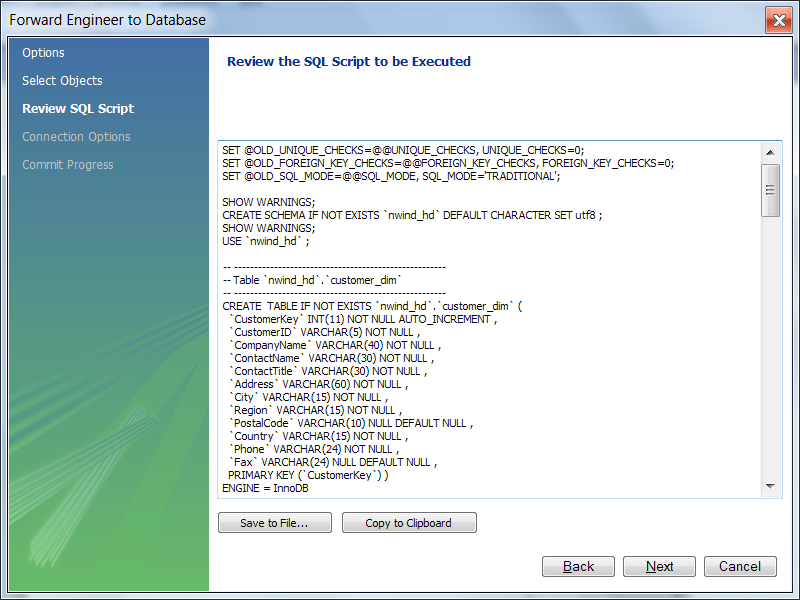


Рис.7. Генерация текста скрипта.

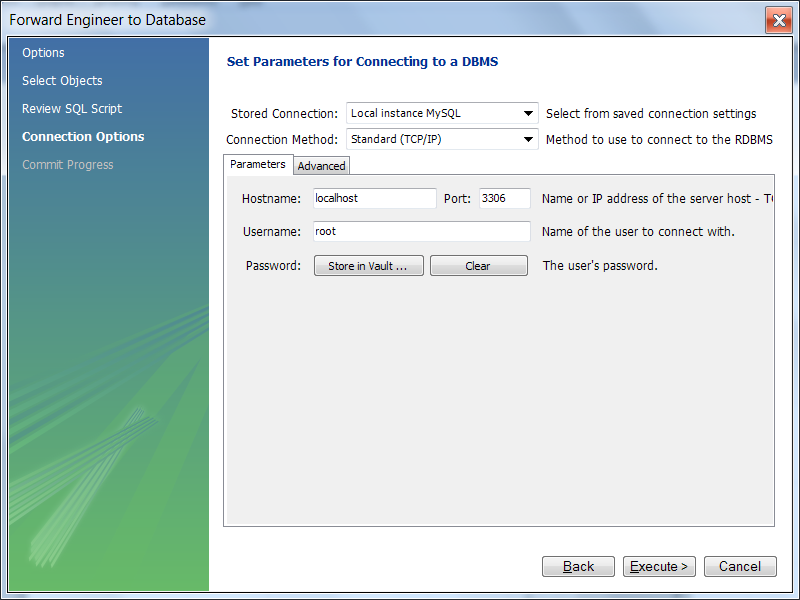


Рис.8. Установление подключения к серверу баз данных.

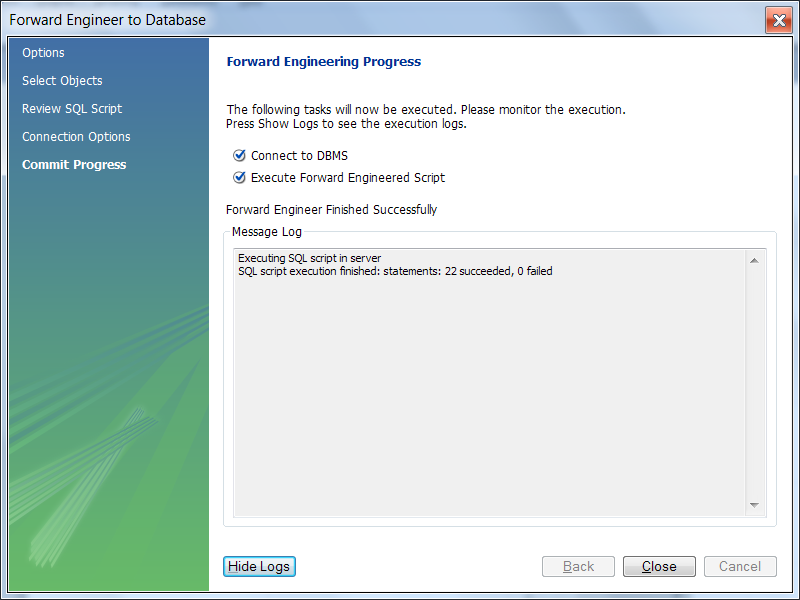


Рис.9. Результат выполнения скрипта.

Следующим шагом является заполнение хранилища данными. Принимая во внимание его архитектуру (в соответствии со схемой «звезда» таблицы измерений связаны с таблицей фактов связями типа «один-ко-многим»), следует сначала заполнить записями таблицы измерений, а затем – таблицу фактов.

# Заключение

В ходе выполнения данной работы было сформировано хранилище данных для базы данных «northwind\_hd». Перенос данных из исходной базы в хранилище был осуществлен с помощью SQL-скрипта с использованием стандартных функций MySQL преобразования данных. В отчете было приведено подробное описание всех выполненных в рамках работы процедур, тексты скриптов, изображения, иллюстрирующие шаги выполнения работы.