

## Лабораторная работа №1

### Часть 1. Создание ER-диаграммы

В лабораторной работе выполняется *концептуальное проектирование* БД с использованием *ER-модели* представления данных (модели «сущность-связь»). Требуется разработать *ER-модель* данных с учетом семантических ограничений заданной предметной области и представить модель в виде *ER-диаграммы*.

#### Порядок выполнения работы

1) По полученному заданию *представить «реальный мир»* (предметную область). То, что входит в эту предметную область, – подлежит моделированию, то, что не входит, – не подлежит. Для этого этапа допустимо словесное или умозрительное представление данных. Задание формулируется только общим направлением (например, названием организации: «библиотека», «столовая» и т.п.), т.к. моделирование предметной области также входит в задачи данной работы. Допустимо моделирование только некоторых аспектов данных в предложенной области (например, только успеваемость школьников в направлении «школа» без учета других особенностей (например, турпоходов, олимпиад, школьной библиотеки, столовой и т.п.)).

2) Сформировать *типы объектов* (для учебной модели требуется не менее 6 сильных типов объектов, рекомендации по выбору объектов смотри в пункте «Пример ER-диаграммы»).

3) Для каждого типа объекта определить 2 – 3 *атрибута*.

4) Сформировать *типы связей* между типами объектов (рекомендации по выбору связей смотри в пункте «Пример ER-диаграммы»). Требуется создать не менее одной связи следующих мощностей:

- мощности «один-ко-многим» для описания иерархии объектов (вхождений),
- мощности «многие-ко-многим» - для описания вариантов взаимодействия разных объектов,
- мощности «многие-ко-многим» с дополнительными атрибутами связи – для описания мелких производственных отношений.

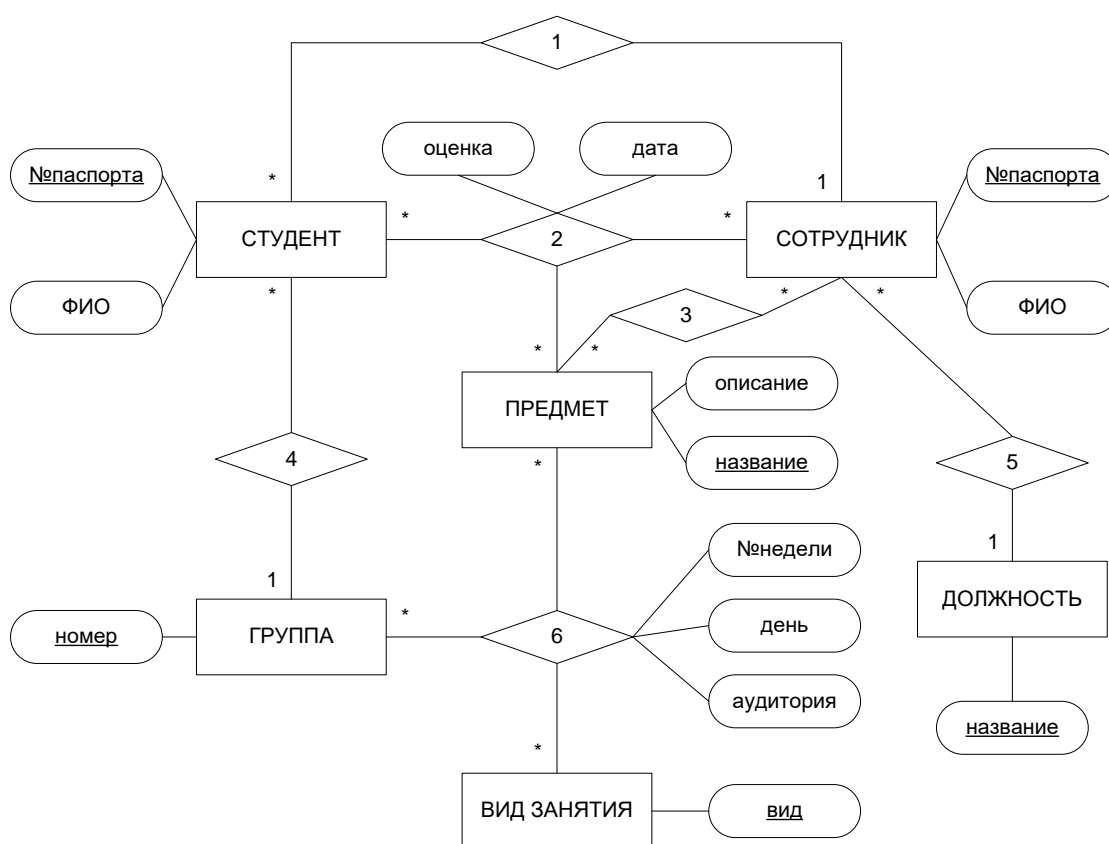
На всех связях ER-диаграммы требуется указать мощности.

5) Полученную предварительную ER-диаграмму еще раз *проверить* по выше изложенным пунктам 2 – 4, т.к. процесс формирования концептуальной модели в рамках общей методологии проектирования БД носит *итерационный* характер.

б) Оформить *отчет*, включающий в себя исходное задание и конечную концептуальную ER-диаграмму (допустимо указывать дополнительные текстовые пояснения, если семантика фрагментов диаграммы не ясна из названий).

### Пример ER-диаграммы

В качестве примера по созданию ER-диаграммы рассмотрим организацию «Университет». В модели «Университет» нашим основным интересом будет описание учебного процесса – состав учебных групп, оценки студентов, расписание занятий (остальная часть реального университета нас не интересует и не будет реализована в рамках ER-диаграммы).



При формировании *типов объектов* рекомендуется:

1) Выделить *единичные объекты* предметной области (естественно не все, но так, чтобы «ассортимент» различных объектов был как можно шире). Этот этап необходим для более полного осмысления предметной области и также может быть выполнен умозрительно. Например, в модели «университет» можно представить себе такие единичные объекты, как «студент Иванов», «группа 050505», «предмет БД» и т.п. Подобные действия можно выполнить также с помощью интервью, проводимого с заказчиком (представителем предметной области).

2) Выделить *классы объектов* (множества качественно (атрибутивно)

сходных объектов). Для модели «университет» можно представить себе такие классы объектов, как «Студент», «Группа», «Предмет». В рамках общей методологии проектирования *не рекомендуется* (в связи с усложнением диаграммы и дальнейших операций над данными, а также низкой информативностью):

- выделять в отдельные классы такие объекты, которые всегда (за время существования БД) будут присутствовать в только единичном экземпляре; для нашего примера таким объектом можно назвать объект «университет», т.к. он всегда будет содержать только один экземпляр – «БГУИР»;
- формировать классы объектов, различающиеся по одному или нескольким признакам при прочих одинаковых атрибутах и общей семантической направленности объектов; например, делить преподавателей на классы «преподаватель физики», «преподаватель математики» и т.п.;
- создавать иерархии объектов путем выделения подмножеств экземпляров объектов с одинаковыми свойствами; например, «сотрудники» – «преподаватели» – «преподаватели кафедры»; дублирование информации также недопустимо с точки зрения возможного нарушения целостности БД;
- выделять в отдельные классы объекты, *сильно зависящие* от других классов (обычно двух и более), т.е. более выражающие связь между объектами.

3) Перевести полученные классы в объекты на ER-диаграмме, причем у каждого объекта обязательно должны быть заданы атрибуты. *Не рекомендуется* создавать атрибуты, имеющие зависимость от числа других объектов (например «оценка по физике», «оценка по математике»), так как это снижает надежность структуры БД и ведет к сложной обработке данных, в таких случаях лучше работает связь с другим объектом. Также, если один объект имеет атрибут, семантически схожий с другим существующим типом объекта, то данный атрибут следует заменить (исходя из соображений целостности БД) на связь между указанными объектами. В учебной модели желательно исключить атрибуты, хранящие статистическую информацию (например «рейтинг студента»), особенно если эти значения можно будет рассчитать другим путем.

При формировании типов связей между типами объектов рекомендуется:

1) Оценить, как могут быть взаимосвязаны между собой экземпляры объектов разных типов объектов. При формировании связей желательно учесть следующее:

- между объектами могут быть заданы разные связи, по одной для каждой функциональности;
- схожие по функциональности связи между одинаковыми участниками могут быть объединены в одну с добавлением соответствующих атрибутов связи;
- при наличии более сложной связи (по числу участников) можно

исключать похожие (дублирующие) связи меньших степеней (для одних и тех же участников и для одной и той же функциональности);

- атрибуты связи, которые семантически схожи с некоторым объектом модели, следует удалить и внести этот объект в качестве дополнительного участника связи;
- необходимо добавить такие дополнительные атрибуты связи, которые будут расширять возможные комбинации отношений объектов (например «дата экзамена»).

2) Установить на связи *структурные ограничения* с учетом текущей ситуации, а также возможных расширений БД при ее эксплуатации. Учесть, что в рамках общей методологии проектирования:

- связи мощности «один-к-одному» возникают в моделях крайне редко, поэтому лучше заменить такую связь связью мощности «один-ко-многим»;
- связи с числом участников более двух лучше задавать как связь мощности «много» со стороны всех участников (дополнительные ограничения можно пояснить на основе выработанных бизнес-правил и учесть на следующих этапах проектирования).

Для модели «Университет» можно выделить следующие связи (с учетом их функциональной направленности и выделенных типов объектов):

1 – «студент – декан», описывает подчиненность студента декану (здесь декан – это подмножество из числа сотрудников; возможность установки связи может быть описана как дополнительное бизнес-правило: «сотрудник должен иметь должность декан»);

2 – «журнал оценок», описывает оценки всех студентов по всем предметам, выставленные преподавателями (здесь преподаватель – подмножество от числа сотрудников; возможность выставления оценки может быть описана как дополнительное бизнес-правило: «сотрудник имеет должность из списка преподавателей»);

3 – «знание предмета», описывает предметы, занятия по которым может вести преподаватель;

4 – «состав группы», описывает вхождение студентов в учебные группы;

5 – «должность сотрудника», описывает должность, которую занимает сотрудник;

6 – «расписание предметов», описывает проведение занятий по аудиториям.

### Варианты заданий

1. Автозаправка.
2. Авторынок (автосалон).
3. Автостоянка.
4. Автошкола.

5. Аэропорт.
6. Банк.
7. Больница.
8. Военкомат.
9. Гостиница.
10. Грузоперевозки.
11. Детский сад.
12. Железнодорожный вокзал (автовокзал, такси и т.п.).
13. Завод.
14. Зоопарк.
15. Кафе (бар).
16. Кинотеатр.
17. Локальная компьютерная сеть (кабельное телевидение).
18. Магазин (музыкальный, продовольственный и т.п.).
19. Налоговая инспекция.
20. Общежитие.
21. Оператор связи.
22. Организация концертов.
23. Поликлиника.
24. Прокат видеодисков.
25. Ресторан.
26. СТО.
27. Столовая.
28. Студия звукозаписи (киностудия).
29. Туристическое агентство.
30. Футбольный клуб.
31. Школа.
32. Бассейн.

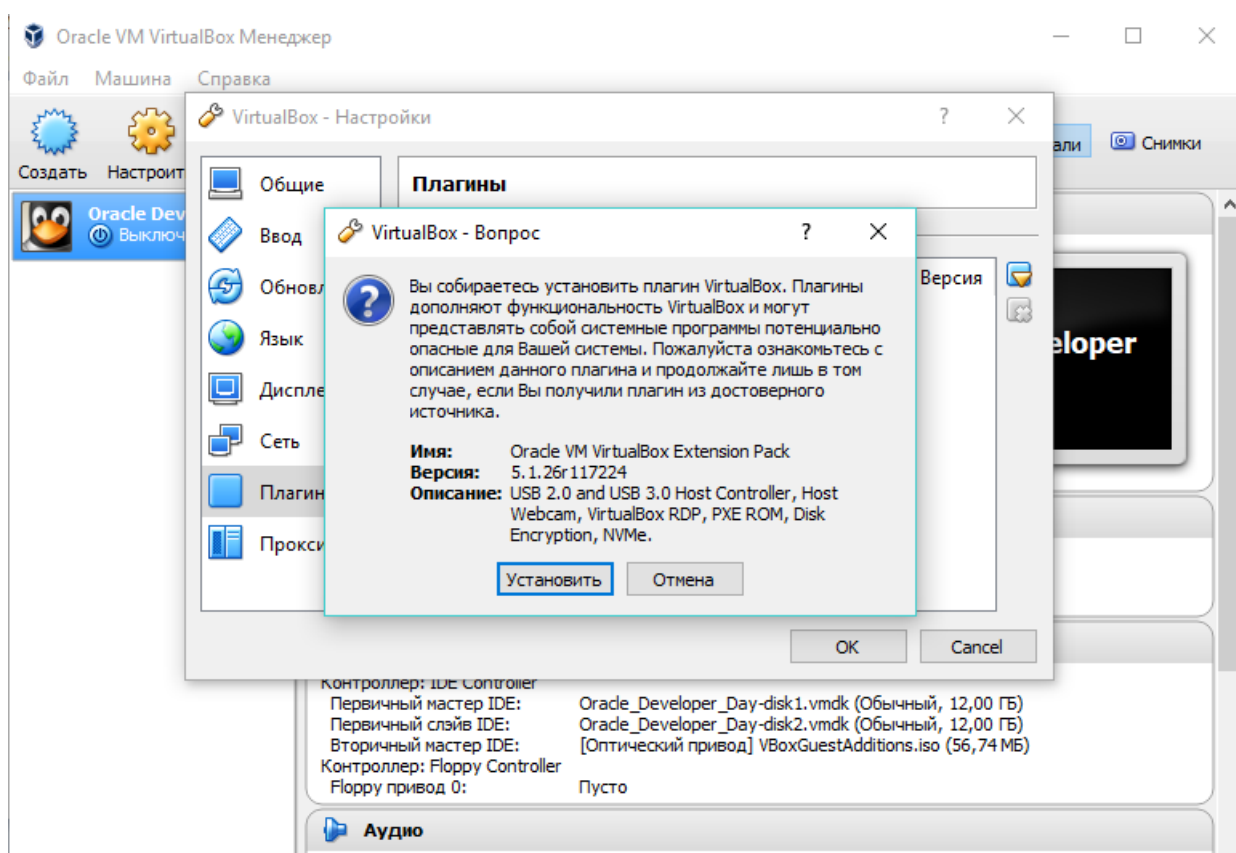
## Часть 2. Установка и изучение демонстрационного программного обеспечения Oracle

Здесь описан вариант установки программного обеспечения Oracle с использованием образа виртуальной машины под ОС MS Windows 10.

### Порядок выполнения работы

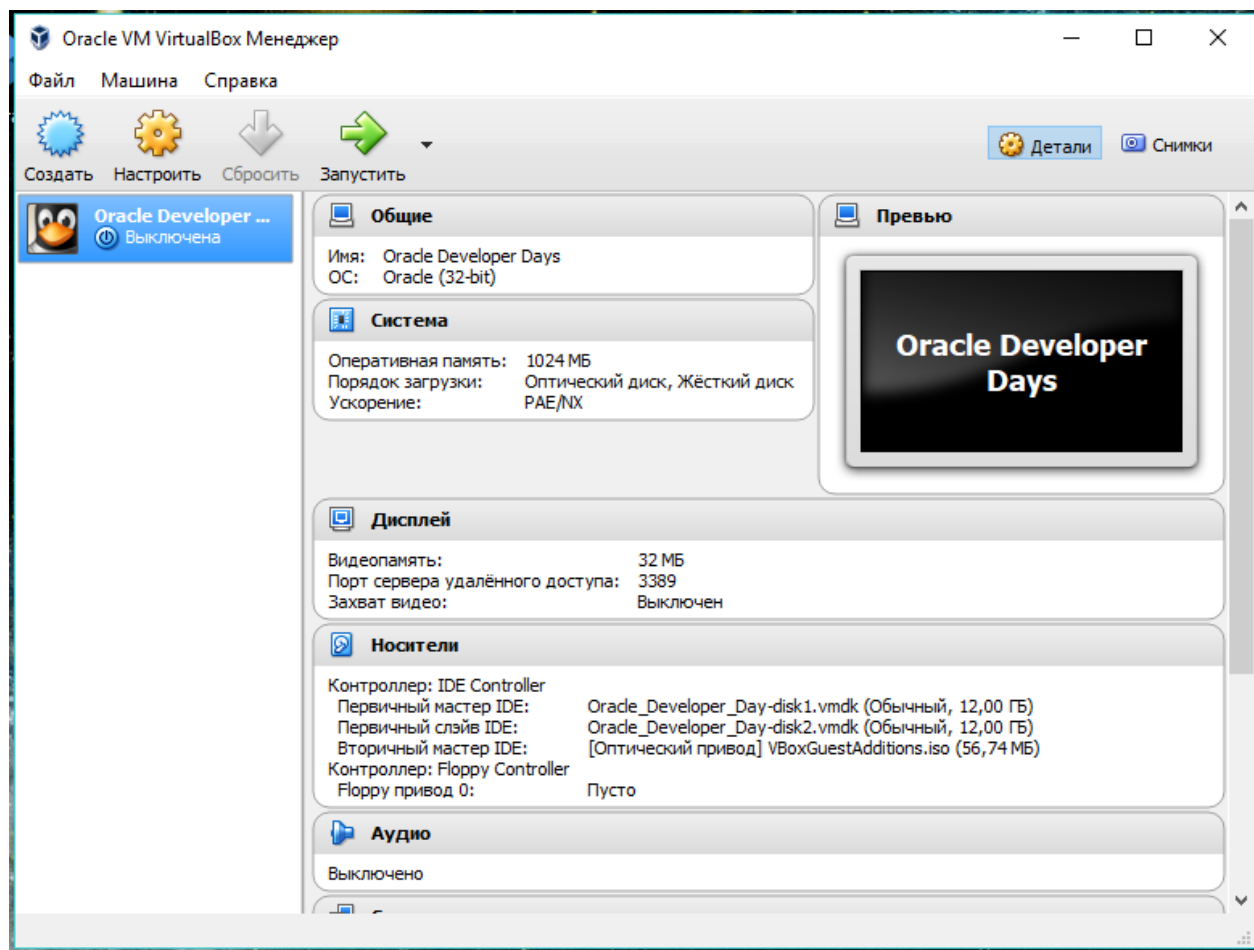
#### 1) Установить Oracle VM VirtualBox:

- Запустить VirtualBox-5.1.26-117224-Win.exe, выполнить установку (достаточно установки по умолчанию).
- Установить плагин Oracle\_VM\_VirtualBox\_Extension\_Pack-5.1.26-117224.vbox-extpack (VB: Файл → Настройки... → Плагины).



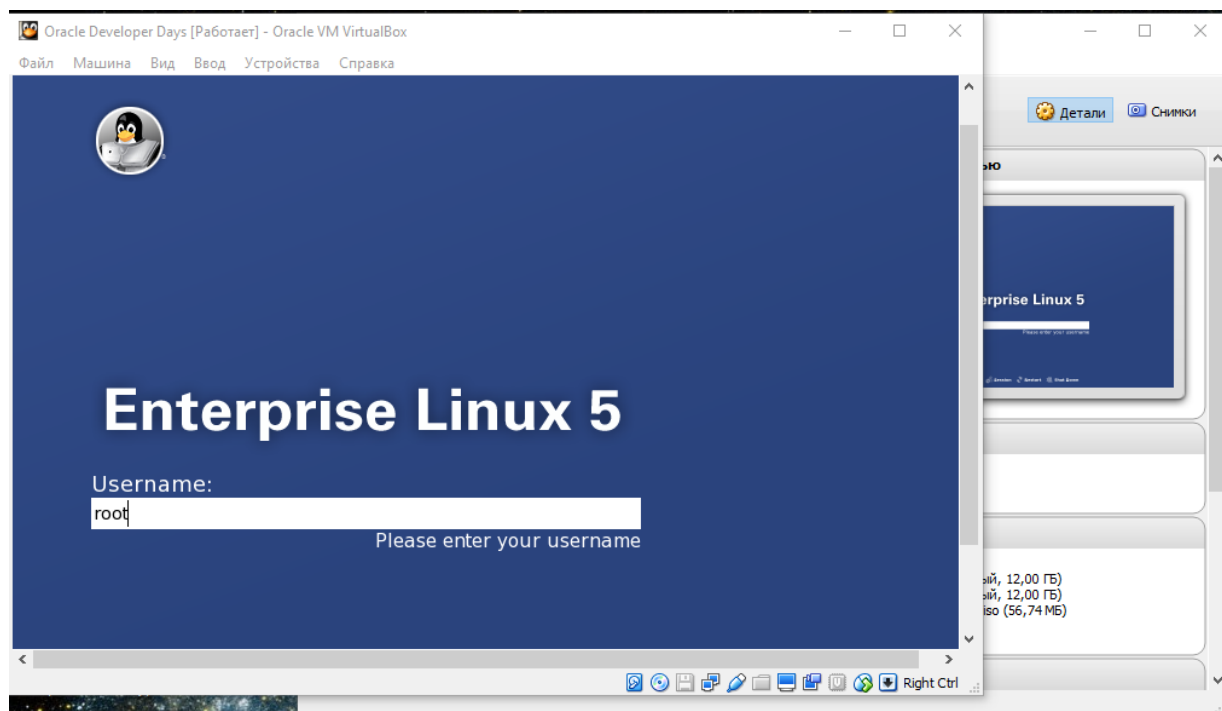
#### 2) Импортировать образ виртуальной машины:

- VB: Файл → Импорт конфигураций... → выбрать Oracle\_Developer\_Day.ova → Импорт... (значения по умолчанию);
- Можно дополнительно настроить виртуальную машину:
  - размер ОЗУ (VB: Машина → Настроить... → Система);
  - размер видеопамяти (VB: Машина → Настроить... → Дисплей);
  - включить поддержку USB (VB: Машина → Настроить... → USB).

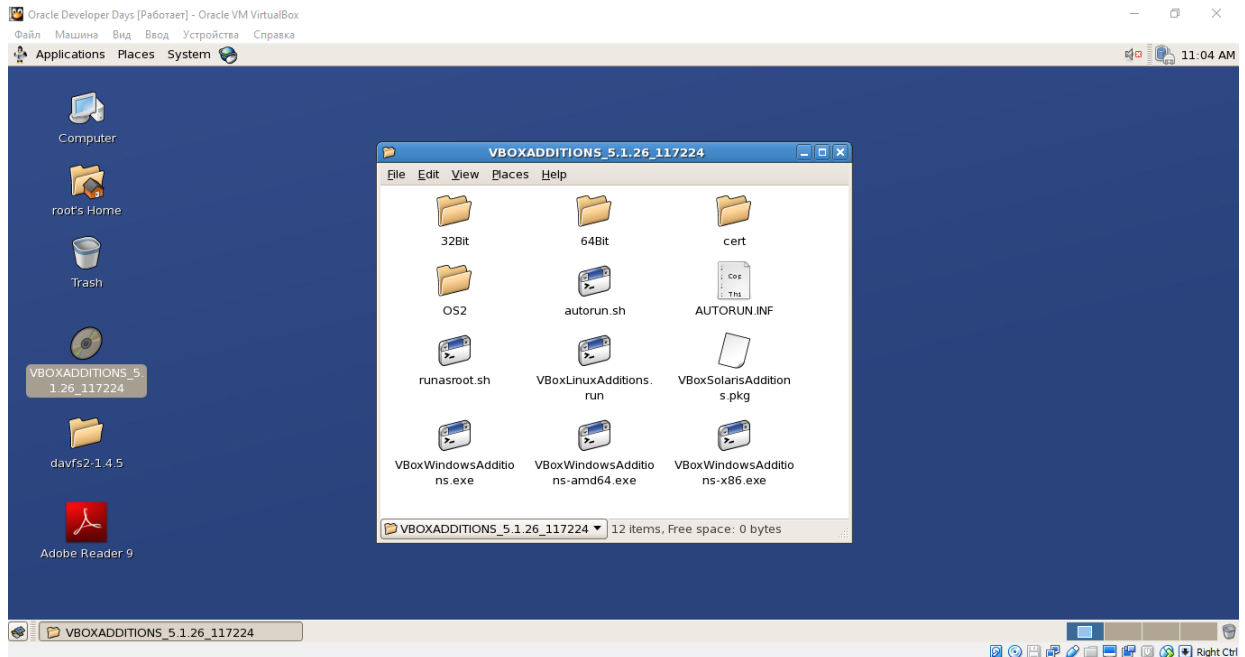


3) Запустить виртуальную машину и настроить ее для работы:

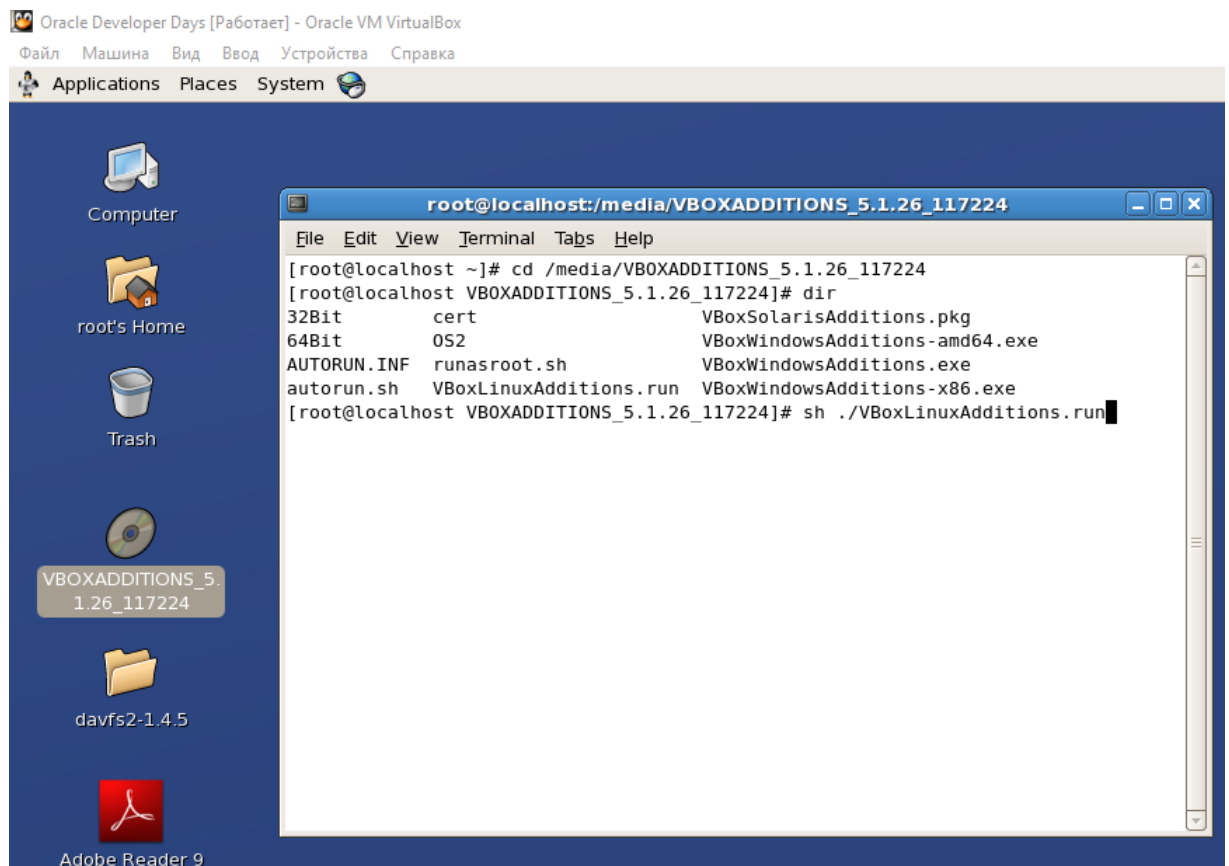
- Войти в Linux под администратором:
  - username: root
  - password: oracle



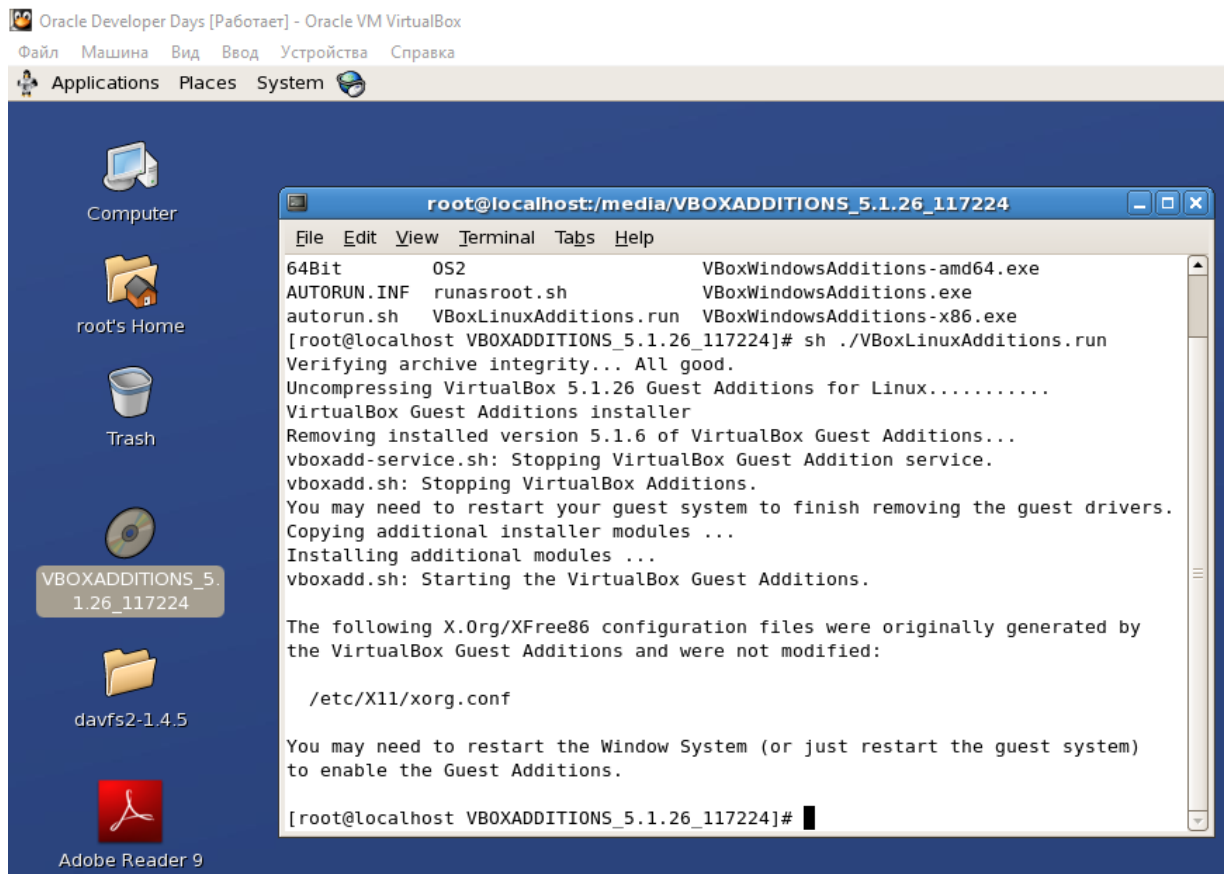
- Если не подключен, то подключить CD с дополнения для гостевой ОС (VM: Устройства → Подключить образ диска Дополнений гостевой ОС).



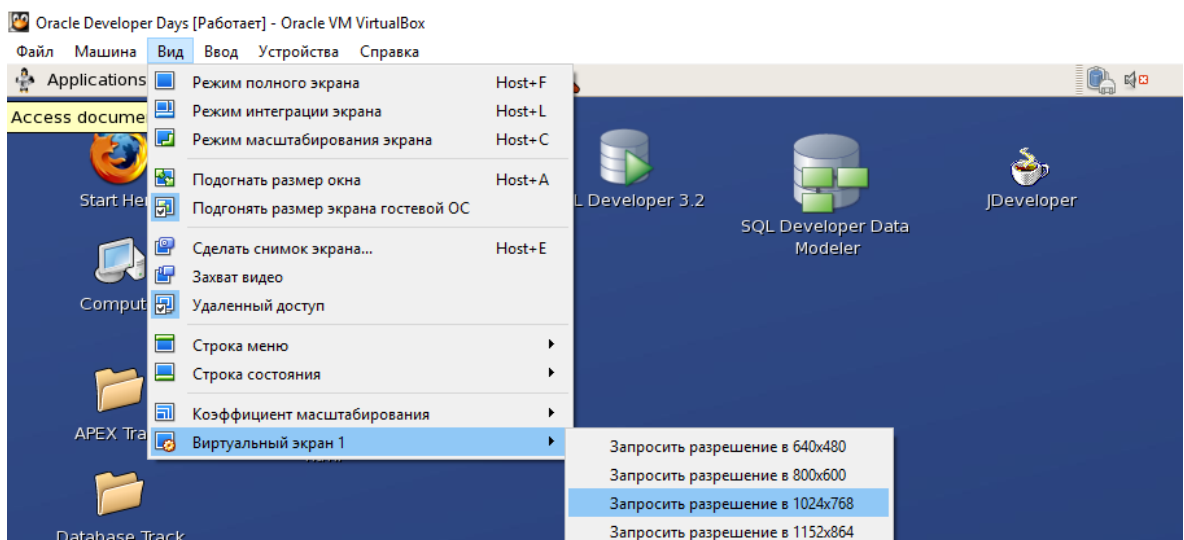
- Запустить терминал (Linux: Applications → Accessories → Terminal) и установить дополнения для гостевой ОС (набрать команды и подождать окончания установки (см. рисунки ниже)).







- Перегрузить или выключить виртуальную машину (например, Linux: System → Shut Down...).
- 4) Запустить виртуальную машину для практической работы:
- Войти под рабочим пользователем:
    - username: oracle
    - password: oracle
  - Установить требуемое для работы разрешение дисплея виртуальной машины (например, VM: Вид → Виртуальный экран 1 → Запросить разрешение в ...).



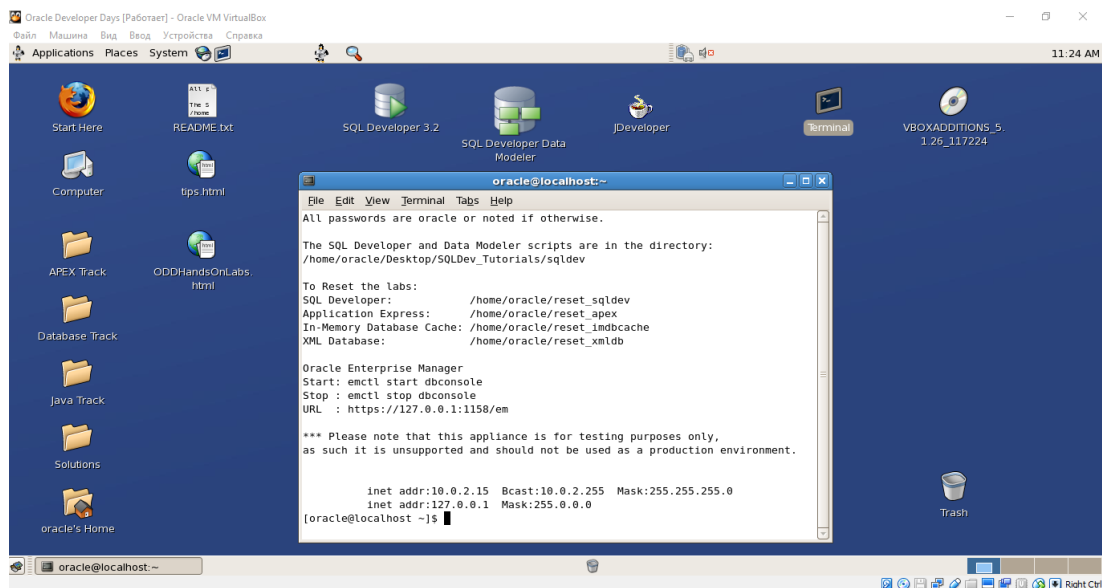
- Затем под рабочим пользователем можно изучить саму ОС Linux (приложения, файловая система, настройки и т.п.) и возможно установить необходимую конфигурацию системы (учтите, что все изменения настроек сохраняются при следующем старте машины, однако бояться не стоит – если что-то «упадет», то можно будет просто переставить и заново настроить виртуальную машину).
- Offline документация по программному обеспечению, установленному на виртуальной машине, находится здесь:



Нас будет интересовать в основном документация по *SQL Developer* и *SQL Developer Data Modeler*.

*Oracle SQL Developer* – мощный инструмент для написания SQL-запросов, разработки PL/SQL пакетов, процедур, функций и триггеров. *SQL Developer* также позволяет выполнять экспорт и импорт данных и структур, миграцию с других баз данных, разработку отчетов и публикацию в Oracle Apex и многое другое. Этот инструмент написан на языке Java и является кросс-платформенным.

*Oracle SQL Developer Data Modeler* – комплексное решение, позволяющее разработчикам проектировать реляционные модели взаимосвязей объектов для последующего преобразования их в полноценные базы данных. Продукт поддерживает логическое, реляционное, многомерное моделирование и моделирование типов данных, предлагая возможности многоуровневого проектирования и построения концептуальных диаграмм сущностей и связей.



5) Для окончания работы – выключить виртуальную машину через меню Linux: System → Shut Down...