

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет имени Г.В. Плеханова»**

**Институт цифровой экономики и информационных технологий
Кафедра Прикладной информатики и информационной безопасности**

Курсовая работа

по дисциплине: «Базы данных и экспертные системы»

на тему: «**Разработка базы данных для учета оборудования и
программного обеспечения в компании (закупки, комплектация,
размещение, ремонты, техническое обслуживание и пр.)**»

Выполнил
обучающийся группы 15.11Д-ИБ08/206
очной формы обучения ВШКМиС
Смирнов Андрей Владимирович
(Фамилия Имя Отчество обучающегося)

Научный руководитель:
Кандидат технических наук, доцент, Гаврилов А. В.
(ученая степень, уч. звание\должность, ФИО)

Дата защиты 10.06.2022

Оценка: _____

Подпись руководителя: _____

Москва – 2022 г.

Содержание

Введение.....	2
Глава 1. Проектирование базы данных	5
1.1 Предметная область и бизнес требования.....	5
1.2 Концептуальная ER-модель ПО	11
1.3 Дatalogическая реляционная модель БД.....	32
Глава 2. Реализация БД	38
2.1 DDL-Сценарий создания БД.....	38
2.2 Заполнение таблиц через SQL-скрипт	49
2.3 SQL-запросы, Select	58
2.4 Триггеры и их реализация.....	61
2.5 Хранимые процедуры и их реализация.....	61
Заключение	63
Список литературы	63

Введение.

Базы данных — это совокупность структур, предназначенных для хранения большого количества разноплановой информации и вспомогательных инструментов, осуществляющих управление данными, формой их представления, возможности их структурирования, взаимосвязи и прочего.

Цель баз данных — за счёт проведения учёта и организации информации, упростить обработку больших объемов информации, минимизировать объем базы данных за счёт разбития повторяющейся информации на несколько связанных таблиц, образовать связи между информацией в таблицах между собой, с последующим приведением их к легче воспринимаемому виду для последующей обработки человеком и более быстрому принятию соответственных решений на основе структурированной информации.

В данной курсовой работе будет разработана база данных для учета оборудования и программного обеспечения в компании: закупки оборудования и ПО, их комплектация, размещение в производственных помещениях. Также будет учтен ремонт и техническое обслуживание оборудования. В качестве примера компании для реализации БД был выбран офис ИТ компании. На его примере можно в полной мере продемонстрировать хранение всей сопутствующей и сопровождающей информации в виде реляционной базы данных.

Актуальность темы курсовой работы – необходимость в цифровизации, упрощении работы с большими массивами информации: их структуризация, перенос в одну базу данных для быстрого анализа и принятия соответствующих управленческих решений. В некотором роде, обеспечение свойств информации – конфиденциальности, в том в виде, что в отличие от полностью бумажных или Excel-подобных способов хранения информации, БД предоставляет доступ к хранимой информации только лицам, имеющим на это право. Целостность информации, в том, что вся информация в базе данных, в отличие от бумажных и других разрозненных методов хранения информации, может быть утеряна с меньшим шансом, учитывая человеческий фактор, чем в других формах хранения. Доступность в том плане, что любой пользователь, имеющий право на получение доступа к информации, может его получить. Управляемость, т.к СУБД предоставляет инструменты для удобного использования и корректировки и прочих действий с информацией, хранимой в базе данных.

Целью курсовой работы является анализ особенностей функционирования ИТ офиса: изучение его структуры, технических и программных требований для организации бесперебойной работы. И на основе этого анализа разработать схему базы данных и реализовать саму БД.

В качестве инструмента для построения даталогической модели буду использовать ПО Oracle Data Modeler из-за его простоты, а также возможности быстрого перевода даталогической модели в DDL-скрипт, требующий минимальных доработок. В качестве инструмента для проектирования непосредственно самой БД – Microsoft SQL Server Management Studio.

Задачи курсовой работы:

Глава 1. Проектирование БД

1.Изучение предметной области для реализации подробных бизнес-требования к хранимым в БД данным.

2.Создание концептуальной модели с использованием ПО Oracle Data Modeler с учётом уже сформированных бизнес-требований. В дальнейшем – описание созданных сущностей и связей согласно методическим указаниям.

3.Создание даталогической реляционной модели базы данных с использованием ПО Oracle Data Modeler на основе концептуальной модели с последующим документированием получившихся на этом этапе таблиц БД, их ключей и прочего согласно требованиям методологических указаний.

Глава 2. Реализация БД.

1.Создание сценария в виде DDL-скрипта на основе созданной даталогической модели с использованием ПО Oracle Data Modeler. Последующее использование его и создание БД с использованием ПО SQL Server Management Studio.

2.Заполнение таблиц созданной БД с помощью инструментов ПО SQL Server Management Studio.

3.Доказательство работы БД через использование запросов в среде SQL Server Management Studio.

4.Описание двух триггеров к разработанной БД в среде SQL Server Management Studio.

5.Выполнение и описание двух хранимых процедур к разработанной БД в среде SQL Server Management Studio.

Глава 1. Проектирование базы данных

1.1 Предметная область и бизнес требования.

Предметная область – IT офис компании, представляет из себя здание с определенной структурой:

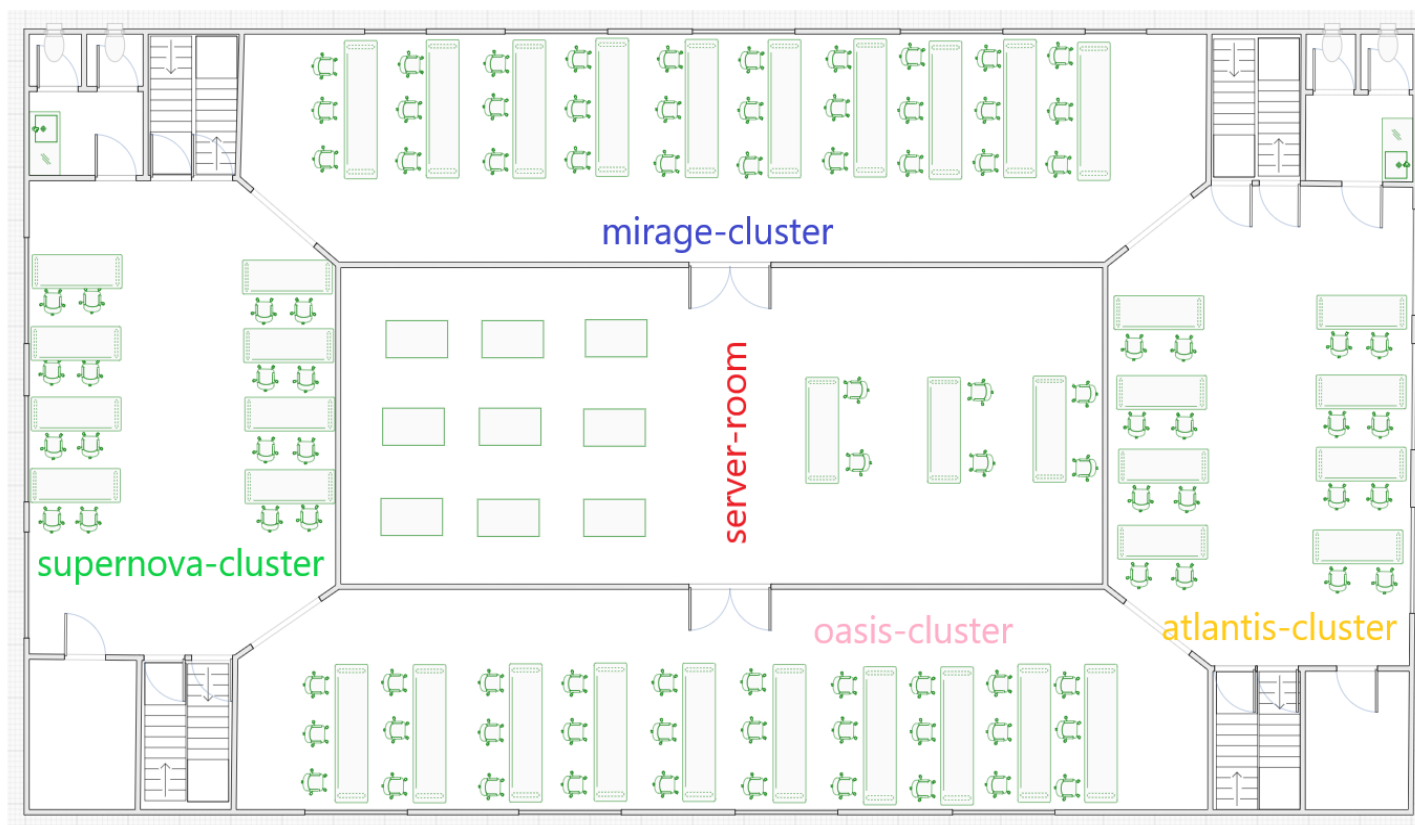


Рисунок 1. Схема этажа с основным персоналом

На данном этаже расположены так называемые кластеры (clusters) – офисы открытого типа, каждый из которых имеет своё название и закреплён

за определенным отделом. Помимо этого, каждый ряд столов в кластере обозначен буквой в алфавитном порядке, а каждое рабочее место – цифрой. Таким образом, за каждым сотрудником закреплено определенное место, которое определяется как **название кластера - номер ряда - номер компьютера**

Выглядит это следующим образом:



Рисунок 2. Организация рабочих мест основного персонала

Помимо офисных помещений, по центру этажа расположена серверная с рабочими местами для системных администраторов.

Каждое рабочее место в офисе укомплектовано одинаково: моноблоком iMac, бесперебойником клавиатурой и мышью.

На следующем этаже расположены кабинеты и архивные помещения:

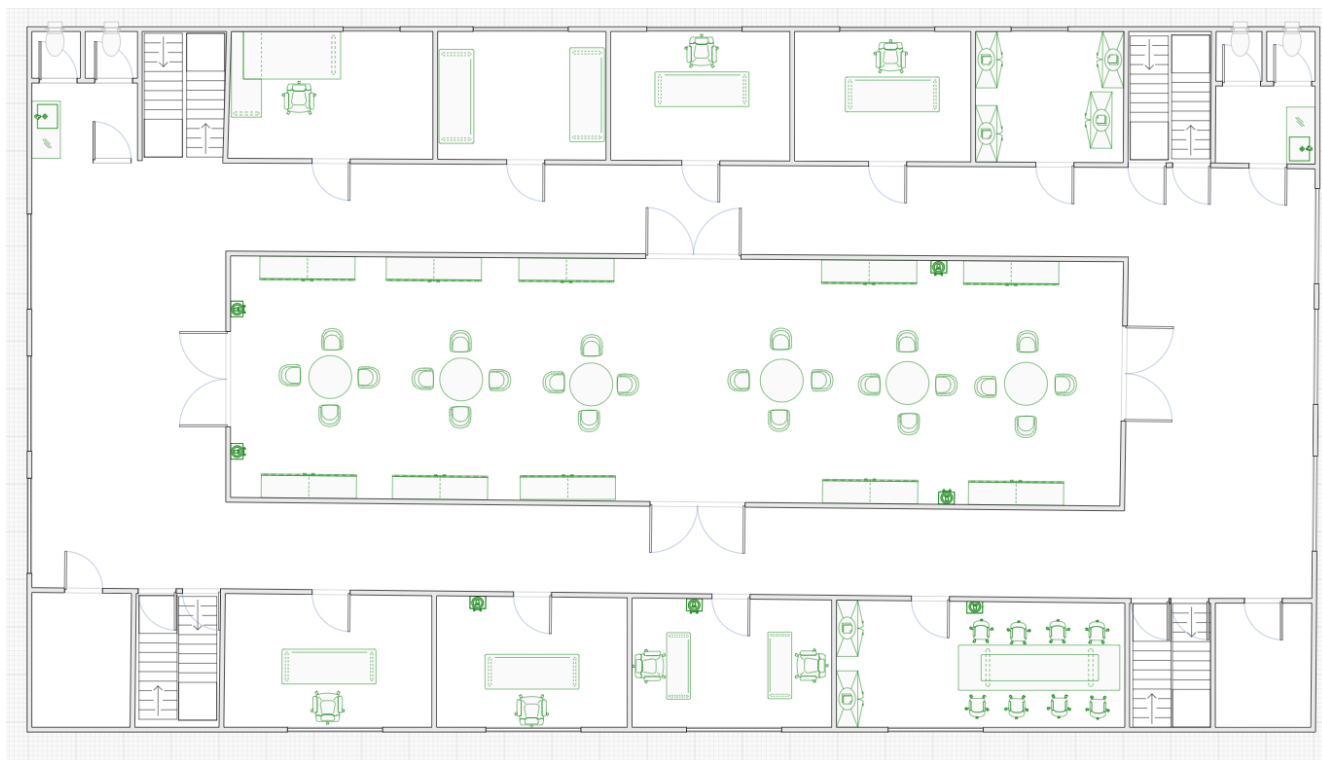


Рисунок 3. Схема этажа с кабинетами для руководства

Как и в офисах, рабочие места в кабинетах оборудованы моноблоками iMac, бесперебойником, клавиатурой и мышью. Но, помимо этого, в некоторых кабинетах могут быть предусмотрены принтеры и телевизоры для проведения презентаций.

На основе приведенной структуры предприятия составим бизнес-требования к хранимым данным:

Таблица 1. Бизнес-требования и реквизиты.

Хранимая информация	Реквизиты
Информация об офисе	Уникальный идентификатор
	Город
	Почтовый индекс
	Адрес
	Количество работников
Информация о работниках	Уникальный идентификатор
	Имя
	Фамилия
	Отчество
	Пол
	Дата рождения

	Паспортные данные
	Адрес проживания
	Личный номер телефона
	Рабочий номер телефона
	Должность
	Идентификатор офиса
	Зарплата
Информация о рабочих местах в кластерах	Уникальный идентификатор
	Идентификатор офиса
	Название кластера
	Номер ряда
	Номер рабочего места
	Идентификатор ПК
	Идентификатор клавиатуры
	Идентификатор мыши
	Идентификатор бесперебойника
	Идентификатор работника
Информация о рабочих местах в кабинетах	Уникальный идентификатор
	Идентификатор офиса
	Номер кабинета
	Идентификатор ПК
	Идентификатор клавиатуры
	Идентификатор мыши
	Идентификатор бесперебойника
	Идентификатор работника
	Идентификатор принтера
	Идентификатор телевизора
Информация о серверных стойках	Уникальный идентификатор
	Модель коммутатора
	Модель сервера
Список моделей iMac	Уникальный идентификатор
	ЦПУ
	ГПУ
	ССД
	ОП
	Дисплей
	Стоимость

	Поставщик
Список моделей клавиатур	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей компьютерных мышей	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей бесперебойников	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей принтеров	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей телевизоров	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей серверов	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список моделей коммутаторов	Уникальный идентификатор
	Компания-производитель
	Стоимость
	Поставщик
Список ПК в офисе	Уникальный идентификатор
	Модель
	Офис
	Статус
	Проблема
Список клавиатур в офисе	Уникальный идентификатор
	Модель
	Офис

	Статус
	Проблема
Список компьютерных мышей в офисе	Уникальный идентификатор
	Модель
	Офис
	Статус
	Проблема
Список бесперебойников в офисе	Уникальный идентификатор
	Модель
	Офис
	Статус
	Проблема
Список принтеров в офисе	Уникальный идентификатор
	Модель
	Офис
	Статус
	Проблема
Список телевизоров в офисе	Уникальный идентификатор
	Модель
	Офис
	Статус
	Проблема
Список поставщиков оборудования	Уникальный идентификатор
	Вид техники
	Компания-производитель
	Веб-сайт
	Телефон
Список поставщиков программного обеспечения	Уникальный идентификатор
	Компания-разработчик
	Веб-сайт
	Телефон

1.2 Концептуальная ER-модель ПО

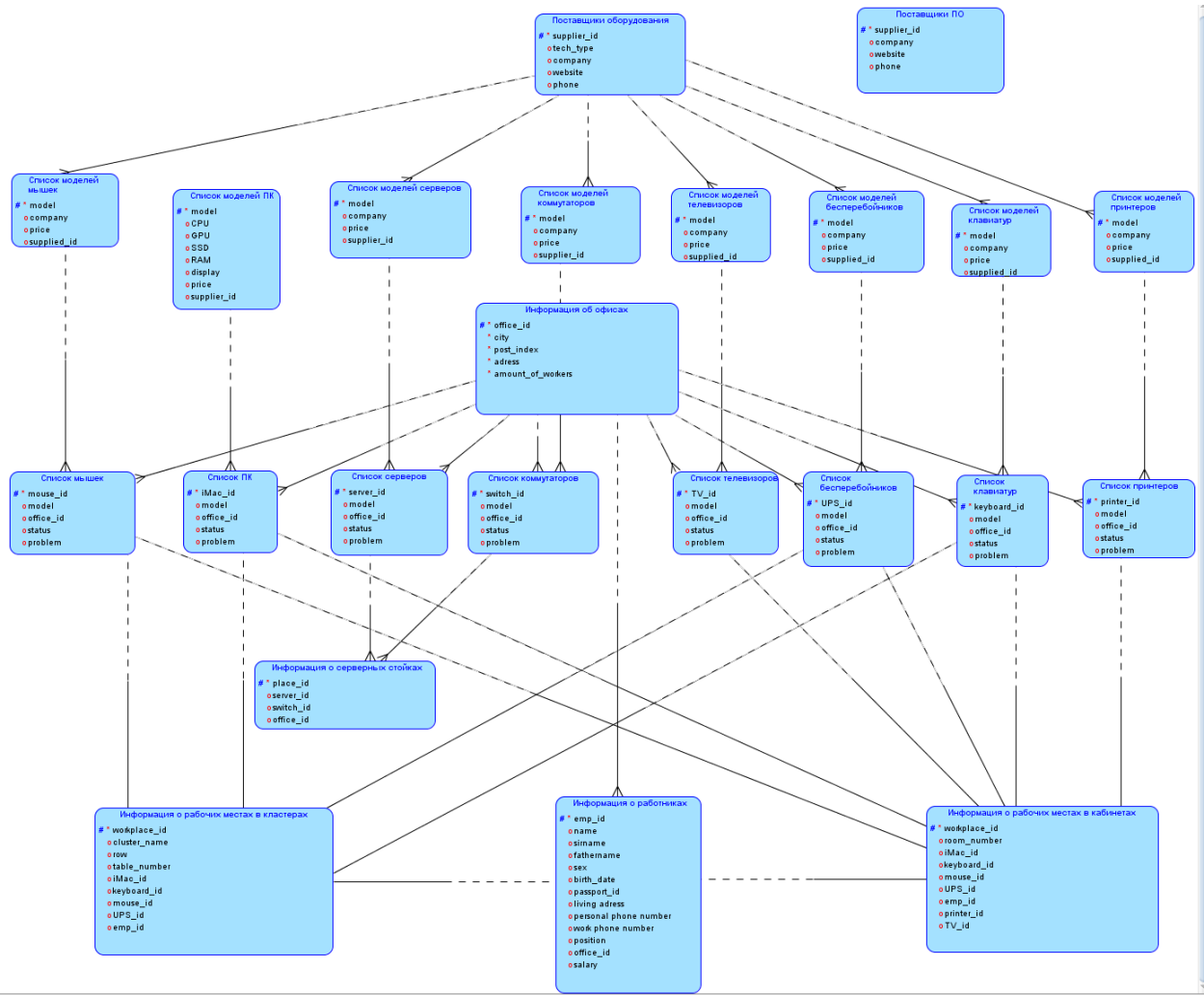


Рисунок 4.Инфологическая модель предметной области, выполненная при помощи программного продукта Oracle SQL Developer Data Modeler

1.Сущность Информация об офисах

Таблица 2. Сущность Информация об офисах

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id офиса	Office_id	Smallint		Primary UID
2.	Город	City	Varchar	20	Mandatory
3.	Почтовый индекс	Post_index	Varchar	6	Mandatory
4.	Адрес	Adress	Varchar	100	Mandatory
5.	Кол-во работников	Amount_of_workers	Smallint		Mandatory

2.Сущность Информация о работниках

Таблица 3. Сущность Информация о работниках

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id работника	Emp_id	Smallint		Primary UID
2.	Имя	Name	Varchar	20	Mandatory
3.	Фамилия	Surname	Varchar	20	Mandatory
4.	Отчество	Fathername	Varchar	20	
5.	Пол	Sex	Varchar	1	Mandatory
6.	Дата рождения	Birthday	Datetime		Mandatory
7.	Паспортные данные	Passport_id	Varchar	8	Mandatory
8.	Адрес проживания	Living address	Varchar	100	Mandatory
9.	Личный номер телефона	Personal phone	Varchar	14	Mandatory
10.	Рабочий номер телефона	Work phone	Varchar	14	
11.	Должность	Position	Varchar	20	Mandatory
12.	Оклад	Salary	Integer		Mandatory

3.Сущность Информация о рабочих местах в кластерах

Таблица 4. Сущность Информация о рабочих местах в кластерах

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id рабочего места	Workplace_id	Smallint		Primary UID
2.	Название кластера	Cluster_name	Varchar	20	Mandatory
3.	Номер ряда	Row	Varchar	1	Mandatory
4.	Номер места	Table	Varchar	1	Mandatory

4.Сущность Информация о рабочих местах в кабинете

Таблица 5. Сущность Информация о рабочих местах в кабинете

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id рабочего места	Workplace_id	Smallint		Primary UID
2.	Номер кабинета	Cabinet_number	varchar	3	Mandatory

5.Сущность Поставщики ПО

Таблица 6. Сущность Поставщики ПО

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id поставщика	Supplier_id	Smallint		Primary UID
2.	Название компании	Company	Varchar	20	Mandatory
3.	Вебсайт	Website	Varchar	512	Mandatory
4.	Телефон	Phone	Varchar	15	

6.Сущность Поставщики оборудования

Таблица 7. Сущность Поставщики оборудования

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id поставщика	Supplier_id	Smallint		Primary UID
2.	Тип техники	Tech_type	Varchar	15	Mandatory
3.	Фирма производитель	Company	Varchar	20	Mandatory
4.	Вебсайт	Website	Varchar	512	Mandatory
5.	Телефон	Phone	Varchar	15	

7.Сущность Список моделей ПК

Таблица 8. Сущность Список моделей ПК

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	ЦПУ	CPU	Varchar	30	Mandatory
3.	ГПУ	GPU	Varchar	30	Mandatory
4.	ССД	SSD	Varchar	20	Mandatory
5.	РАМ	RAM	Varchar	20	Mandatory
6.	Дисплей	Display	varchar	40	Mandatory
7.	Стоимость	Price	Smallint		Mandatory

8.Сущность Список моделей мышек

Таблица 9. Статус договора

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

9.Сущность Список моделей серверов

Таблица 10. Список моделей серверов

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

10.Сущность Список моделей коммутаторов

Таблица 11. Список моделей коммутаторов

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

11.Сущность Список моделей телевизоров

Таблица 12. Список моделей телевизоров

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

12.Сущность Список моделей бесперебойников

Таблица 13. Список моделей бесперебойников

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

13.Сущность Список моделей клавиатур

Таблица 14. Список моделей клавиатур

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

14.Сущность Список моделей принтеров

Таблица 15. Список моделей принтеров

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Модель	Model	varchar	20	Primary UID
2.	Фирма производитель	Company	Varchar	20	Mandatory
3.	Стоимость	Price	Smallint		Mandatory

15.Сущность Список мышек

Таблица 16. Список мышек

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id мышки	Mouse_id	Smallint		Primary UID
2.	Модель мышки	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

16.Сущность Список ПК

Таблица 17. Список ПК

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id iMac	iMac_id	Smallint		Primary UID
2.	Модель iMac	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

17.Сущность Список серверов

Таблица 18. Список серверов

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id сервера	Server_id	Smallint		Primary UID
2.	Модель сервера	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

18.Сущность Список коммутаторов

Таблица 19. Список коммутаторов

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id коммутатора	Switch_id	Smallint		Primary UID
2.	Модель коммутатора	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

19.Сущность Список телевизоров

Таблица 20. Список телевизоров

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен , на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id телевизора	TV_id	Smallint		Primary UID
2.	Модель телевизора	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

20.Сущность Список бесперебойников

Таблица 21. Список бесперебойников

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id бесперебойника	UPS_id	Smallint		Primary UID
2.	Модель бесперебойника	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

21.Сущность Список клавиатур

Таблица 22. Список клавиатур

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id клавиатуры	Keyboard_id	Smallint		Primary UID
2.	Модель клавиатуры	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

22.Сущность Список принтеров

Таблица 23. Список принтеров

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id принтера	Printer_id	Smallint		Primary UID
2.	Модель принтера	Model	Varchar	20	Mandatory
3.	Статус	Status	Varchar	10	Mandatory
4.	Проблема	Problem	Problem	512	

23. Сущность Информация о серверных стойках

Таблица 24. Информация о серверных стойках

Порядковый номер в списке атрибутов	Имя атрибута	Предпочтительное сокращение (Preferred Abbreviation)	Тип данных/домен, на котором основан атрибут	Параметры атрибута в зависимости от типа данных (размер/точность, масштаб)	Ограничения
1.	Id Места расположения	Place_id	Smallint		Primary UID

Связи

1. Связь «**Поставщик мышек**» между сущностями «Список моделей мышек» и «Поставщики оборудования» имеет мощность **1:N**, т.к поставщики у разных мышек могут быть разные, но при этом у конкретной модели мышки может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей мышек» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей мышек» **является обязательным** (у каждой мышки должен быть ее поставщик), сущности «Поставщики оборудования» **не обязательным**
2. Связь «**Поставщик iMac**» между сущностями «Список моделей ПК» и «Поставщики оборудования» имеет мощность **1:N**, т.к поставщики у разных ПК могут быть разные, но при этом у конкретной модели ПК может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей ПК» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей ПК» **является обязательным** (у каждого ПК должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**

3. Связь **«Поставщик серверов»** между сущностями «Список моделей серверов» и «Поставщики оборудования» имеет мощность **1:N**, т.к поставщики у разных серверов могут быть разные, но при этом у конкретной модели сервера может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей серверов» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей серверов» **является обязательным** (у каждого сервера должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**
4. Связь **«Поставщик коммутаторов»** между сущностями «Список моделей коммутаторов» и «Список поставщиков» имеет мощность **1:N**, т.к поставщики у разных коммутаторов могут быть разные, но при этом у конкретной модели коммутатора может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей коммутаторов» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей коммутаторов» **является обязательным** (у каждого коммутатора должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**
5. Связь **«Поставщик телевизоров»** между сущностями «Список моделей телевизоров» и «Список поставщиков» имеет мощность **1:N**, т.к поставщики у разных телевизоров могут быть разные, но при этом у конкретной модели телевизора может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей телевизоров» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей телевизоров» **является обязательным** (у каждого телевизора

должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**

6. Связь «**Поставщик бесперебойников**» между сущностями «Список моделей бесперебойников» и «Список поставщиков» имеет мощность **1:N**, т.к поставщики у разных бесперебойников могут быть разные, но при этом у конкретной модели бесперебойника может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей бесперебойников» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей бесперебойников» **является обязательным** (у каждого бесперебойника должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**
7. Связь «**Поставщик клавиатур**» между сущностями «Список моделей клавиатур» и «Список поставщиков» имеет мощность **1:N**, т.к поставщики у разных клавиатур могут быть разные, но при этом у конкретной модели клавиатуры может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей клавиатур» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список моделей клавиатур» **является обязательным** (у каждой клавиатуры должен быть ее поставщик), сущности «Поставщики оборудования» **не обязательным**
8. Связь «**Поставщик принтеров**» между сущностями «Список моделей принтеров» и «Список поставщиков» имеет мощность **1:N**, т.к поставщики у разных принтеров могут быть разные, но при этом у конкретной модели принтера может быть только 1 поставщик. Связь **не идентифицирующая**, т.к сущность «Список моделей принтеров» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда

следует, что участие в связи сущности «Список моделей принтеров» **является обязательным** (у каждого принтера должен быть его поставщик), сущности «Поставщики оборудования» **не обязательным**

9. Связь «**Модель мышки**» между сущностями «Список мышек» и «Список моделей мышек» имеет мощность **1:N**, т.к одна и та же модель может быть у многих мышек, но при этом конкретно взятая мышка может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список мышек» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список мышек» **является обязательным** (мышка должна быть какой-то модели), сущности «Список моделей мышек» **не обязательным**
10. Связь «**Модель iMac**» между сущностями «Список ПК» и «Список моделей ПК» имеет мощность **1:N**, т.к одна и та же модель может быть у многих iMac, но при этом конкретно взятый iMac может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список ПК» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список ПК» **является обязательным** (iMac должен быть какой-то модели), сущности «Список моделей ПК» **не обязательным**
11. Связь «**Модель сервера**» между сущностями «Список серверов» и «Список моделей серверов» имеет мощность **1:N**, т.к одна и та же модель может быть у многих серверов, но при этом конкретно взятый сервер может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список серверов» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список серверов» **является**

обязательным (сервер должен быть какой-то модели), сущности «Список моделей серверов» **не обязательным**

12.Связь «**Модель коммутатора**» между сущностями «Список коммутаторов» и «Список моделей коммутаторов» имеет мощность **1:N**, т.к одна и та же модель может быть у многих коммутаторов, но при этом конкретно взятый коммутатор может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список коммутаторов» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список коммутаторов» **является обязательным** (коммутатор должен быть какой-то модели), сущности «Список моделей коммутаторов» **не обязательным**

13.Связь «**Модель телевизора**» между сущностями «Список телевизоров» и «Список моделей телевизоров» имеет мощность **1:N**, т.к одна и та же модель может быть у многих телевизоров, но при этом конкретно взятый телевизор может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список телевизоров» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список телевизоров» **является обязательным** (телевизор должен быть какой-то модели), сущности «Список моделей телевизоров» **не обязательным**

14.Связь «**Модель бесперебойника**» между сущностями «Список бесперебойников» и «Список моделей бесперебойников» имеет мощность **1:N**, т.к одна и та же модель может быть у многих бесперебойников, но при этом конкретно взятый бесперебойник может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список бесперебойников» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет

к избыточности ключа. Отсюда следует, что участие в связи сущности «Список бесперебойников» **является обязательным** (бесперебойник должен быть какой-то модели), сущности «Список моделей бесперебойников» **не обязательным**

15.Связь «**Модель клавиатуры**» между сущностями «Список клавиатур» и «Список моделей клавиатур» имеет мощность **1:N**, т.к одна и та же модель может быть у многих клавиатур, но при этом конкретно взятая клавиатура может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список клавиатур» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список клавиатур» **является обязательным** (клавиатура должна быть какой-то модели), сущности «Список моделей клавиатур» **не обязательным**

16.Связь «**Модель принтера**» между сущностями «Список принтеров» и «Список моделей принтеров» имеет мощность **1:N**, т.к одна и та же модель может быть у многих принтеров, но при этом конкретно взятый принтер может быть только одной модели. Связь **не идентифицирующая**, т.к дочерняя сущность «Список принтеров» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список принтеров» **является обязательным** (принтер должен быть какой-то модели), сущности «Список моделей принтеров» **не обязательным**

17.Связь «**Офис местонахождения мышки**» между сущностями «Список мышек» и «Информация об офисах» имеет мощность **1:N**, т.к мышки могут находиться в разных офисах, но при этом конкретно взятая мышка может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список мышек» имеет первичный ключ, достаточный для идентификации.

Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список мышек» **является обязательным** (мышка должна находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

18.Связь «**Офис местонахождения iMac**» между сущностями «Список ПК» и «Информация об офисах» имеет мощность **1:N**, т.к iMac могут находиться в разных офисах, но при этом конкретно взятый iMac может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список ПК» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список ПК» **является обязательным** (iMac должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

19.Связь «**Офис местонахождения сервера**» между сущностями «Список серверов» и «Информация об офисах» имеет мощность **1:N**, т.к сервера могут находиться в разных офисах, но при этом конкретно взятый сервер может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список серверов» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список серверов» **является обязательным** (сервер должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

20.Связь «**Офис местонахождения коммутатора**» между сущностями «Список коммутаторов» и «Информация об офисах» имеет мощность **1:N**, т.к коммутаторы могут находиться в разных офисах, но при этом конкретно взятый коммутатор может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список коммутаторов» имеет первичный ключ, достаточный для

идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список коммутаторов» **является обязательным** (коммутатор должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

21.Связь «**Офис местонахождения телевизора**» между сущностями «Список телевизоров» и «Информация об офисах» имеет мощность **1:N**, т.к телевизоры могут находиться в разных офисах, но при этом конкретно взятый телевизор может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список телевизоров» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список телевизоров» **является обязательным** (телевизор должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

22.Связь «**Офис местонахождения бесперебойника**» между сущностями «Список бесперебойников» и «Информация об офисах» имеет мощность **1:N**, т.к бесперебойники могут находиться в разных офисах, но при этом конкретно взятый бесперебойник может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список бесперебойников» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список бесперебойников» **является обязательным** (бесперебойник должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

23.Связь «**Офис местонахождения клавиатуры**» между сущностями «Список клавиатур» и «Информация об офисах» имеет мощность **1:N**, т.к клавиатуры могут находиться в разных офисах, но при этом конкретно взятая клавиатура может быть только в одном офисе. Связь

не идентифицирующая, т.к дочерняя сущность «Список клавиатур» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список клавиатур» **является обязательным** (клавиатура должна находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

24.Связь «**Офис местонахождения принтера**» между сущностями «Список клавиатур» и «Информация об офисах» имеет мощность **1:N**, т.к клавиатуры могут находиться в разных офисах, но при этом конкретно взятая клавиатура может быть только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Список клавиатур» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список клавиатур» **является обязательным** (клавиатура должен находиться в каком-то из офисов), сущности «Информация об офисах» **не обязательным**

25.Связь «**Офис сотрудника**» между сущностями «Информация об офисах» и «Информация о работниках» имеет мощность **1:N**, т.к. в одном и том же офисе может работать множество сотрудников, но при этом конкретно взятый сотрудник может работать только в одном офисе. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о работниках» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Информация об офисах» **является обязательным** (работник должен быть закреплен за каким-то из офисов), сущности «Информация о работниках» **не обязательным**

26.Связь «**Местонахождение мышки в кластере**» между сущностями «Информация о рабочих местах в кластерах» и «Список мышек» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет

закреплена только 1 мышка. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кластерах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список мышек» **является обязательным** (за рабочим местом обязательно должна быть закреплена мышка)

27.Связь **«Местонахождение iMac в кластере»** между сущностями «Информация о рабочих местах в кластерах» и «Список ПК» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплён только 1 iMac. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кластерах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список iMac» **является обязательным** (за рабочим местом обязательно должен быть закреплён iMac)

28.Связь **«Местонахождение бесперебойника в кластере»** между сущностями «Информация о рабочих местах в кластерах» и «Список бесперебойников» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплён только 1 бесперебойник. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кластерах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список бесперебойников» **является обязательным** (за рабочим местом обязательно должен быть закреплён бесперебойник)

29.Связь **«Местонахождение клавиатуры в кластере»** между сущностями «Информация о рабочих местах в кластерах» и «Список клавиатур» имеет мощность **1:1**, т.к. предполагается, что за рабочим

местом будет закреплена только 1 клавиатура. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кластерах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список клавиатур» **является обязательным** (за рабочим местом обязательно должна быть закреплена клавиатура)

30.Связь **«Местонахождение мышек в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список мышек» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплена только 1 мышка. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список мышек» **является обязательным** (за рабочим местом обязательно должна быть закреплена мышка)

31.Связь **«Местонахождение iMac в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список ПК» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплён только 1 iMac. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список ПК» **является обязательным** (за рабочим местом обязательно должен быть закреплён iMac)

32.Связь **«Местонахождение телевизоров в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список телевизоров» имеет мощность **1:1**, т.к. предполагается, что за рабочим

местом будет закреплен только 1 телевизор. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список телевизоров» **является необязательным** (допускается, что на каких-то рабочих местах принтер будет отсутствовать)

33.Связь **«Местонахождение бесперебойников в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список бесперебойников» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплен только 1 бесперебойник. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список бесперебойников» **является обязательным** (допускается, что на каких-то рабочих местах телевизор будет отсутствовать)

34.Связь **«Местонахождение бесперебойников в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список бесперебойников» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплен только 1 бесперебойник. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список бесперебойников» **является обязательным** (за рабочим местом обязательно должен быть закреплен бесперебойник)

35.Связь **«Местонахождение принтеров в кабинетах»** между сущностями «Информация о рабочих местах в кабинетах» и «Список принтеров» имеет мощность **1:1**, т.к. предполагается, что за рабочим

местом будет закреплен только 1 принтер. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Список принтеров» **является необязательным** (допускается, что на каких-то рабочих местах принтер будет отсутствовать)

36.Связь «**Рабочее место сотрудника в кластере**» между сущностями «Информация о рабочих местах в кластерах» и «Информация о работниках» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплен только 1 сотрудник. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кластерах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Информация о работниках» **является необязательным** (допускается, что рабочее место будет свободно)

37.Связь «**Рабочее место сотрудника в кабинете**» между сущностями «Информация о рабочих местах в кабинетах» и «Информация о работниках» имеет мощность **1:1**, т.к. предполагается, что за рабочим местом будет закреплен только 1 сотрудник. Связь **не идентифицирующая**, т.к дочерняя сущность «Информация о рабочих местах в кабинетах» имеет первичный ключ, достаточный для идентификации. Идентифицирующая связь приведет к избыточности ключа. Отсюда следует, что участие в связи сущности «Информация о работниках» **является необязательным** (допускается, что рабочее место будет свободно)

1.3 Даталогическая реляционная модель БД

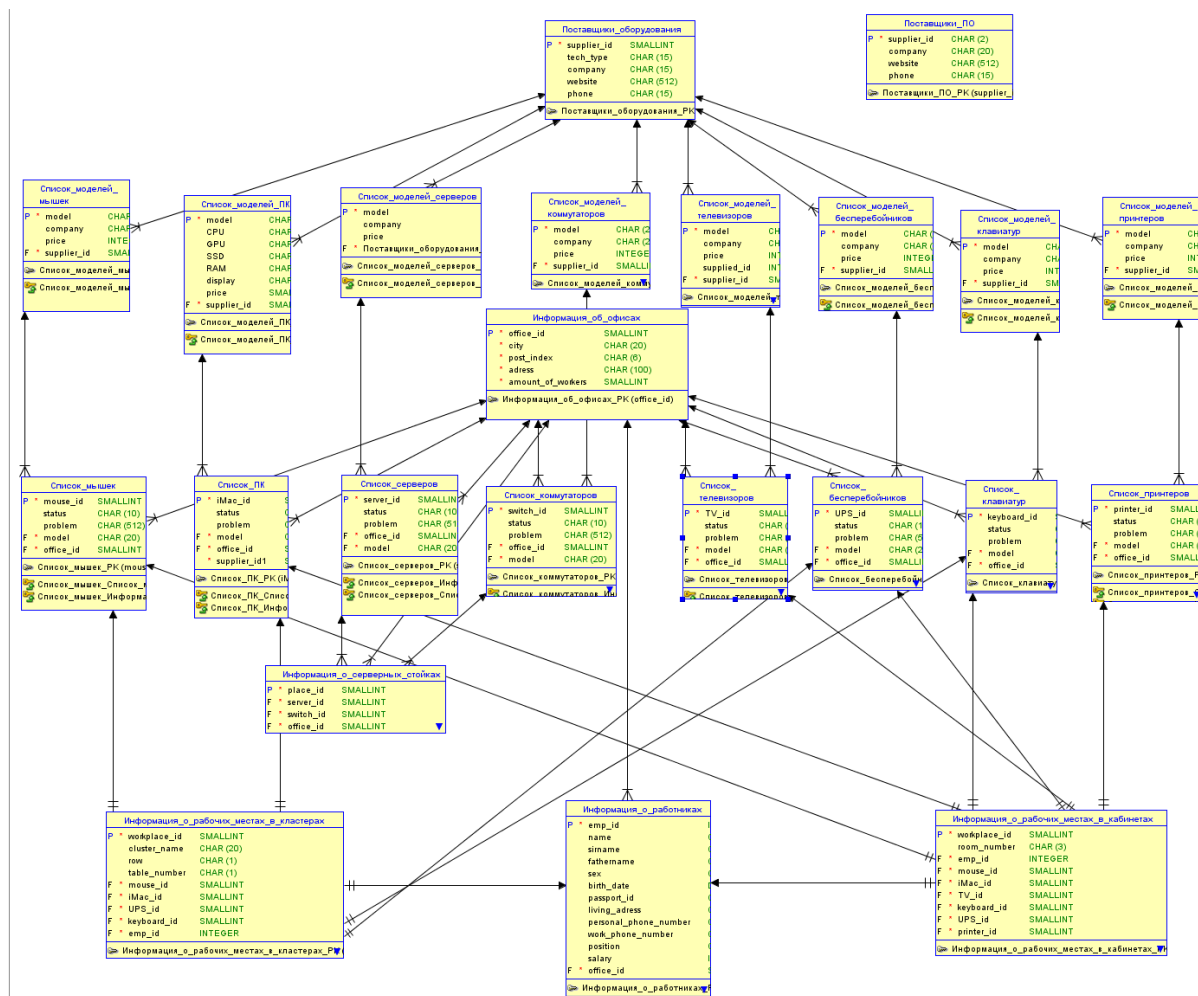


Рисунок 5. Даталогическая модель (выполнена с использованием программного продукта Oracle SQL Developer Data Modeler)

1. Таблица employee_info

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Emp_id	Integer	Primary Key	Not Null
Name	Varchar	-	Not Null
Sirname	Varchar	-	Not Null
Fathername	Varchar	-	Null
Sex	Varchar	-	Not Null
Birth_date	Date	-	Not Null
Passport_id	Varchar	-	Not Null
Living_address	Varchar	-	Not Null
Personal_phone_number	Varchar	-	Not Null
Work_phone_number	Varchar	-	Null
Position	Varchar	-	Not Null
Salary	Integer	-	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 20. employee_info

2. Таблица clusters_workplaces_info

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Workplace_id	Smallint	Primary Key	Not Null
Cluster_name	Varchar	-	Not Null
Row	Varchar	-	Not Null
Table_number	Varchar	-	Not Null
Mouse_id	Smallint	Foreign_key	Not Null
iMac_id	Smallint	Foreign_key	Not Null
UPS_id	Smallint	Foreign_key	Not Null
Keyboard_id	Smallint	Foreign_key	Not Null
Emp_id	Smallint	Foreign_key	Null

Таблица 21. clusters_workplaces_info

3. Таблица cabinets_workplaces_info

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Workplace_id	Smallint	Primary Key	Not Null
Room_number	Varchar	-	Not Null
Emp_id	Smallint	Foreign_key	Null
Mouse_id	Smallint	Foreign_key	Not Null
iMac_id	Smallint	Foreign_key	Not Null
TV_id	Smallint	Foreign_key	Null
Keyboard_id	Smallint	Foreign_key	Not Null
UPS_id	Smallint	Foreign_key	Not Null
Printer_id	Smallint	Foreign_key	Null

Таблица 22. cabinets_workplaces_info

4. Таблица server_tables_info

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Place_id	Smallint	Primary Key	Not Null
Server_id	Smallint	Foreign_key	Not Null
Switch_id	Smallint	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 23. server_tables_info

5. Таблица mouse_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Mouse_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 24. mouse_ids

6. Таблица iMac_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
iMac_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 25. iMac_ids

7. Таблица server_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
server_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 26. server_ids

8. Таблица switch_id

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
switch_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 27. switch_ids

9. Таблица TV_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
TV_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 28. TV_ids

10. Таблица UPS_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
UPS_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 29. UPS_ids

11. Таблица keyboard_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
keyboard_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 30. keyboard_ids

12. Таблица printer_ids

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
printer_id	Smallint	Primary Key	Not Null
Status	Varchar	-	Not Null
Problem	Varchar	-	Null
Model	Varchar	Foreign_key	Not Null
Office_id	Smallint	Foreign_key	Not Null

Таблица 31. printer_ids

13. Таблица office_info

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
office_id	Smallint	Primary Key	Not Null
city	Varchar	-	Not Null
Post_index	Varchar	-	Not Null
adress	Varchar	-	Not Null
Amount_of_workers	Varchar	-	Not Null

Таблица 32. office_info

14. Таблица mouse_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null

Supplier_id	Smallint	Foreign_key	Not Null
-------------	----------	-------------	----------

Таблица 33. mouse_models

15. Таблица iMac_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
model	Smallint	Primary Key	Not Null
CPU	Varchar	-	Not Null
GPU	Varchar	-	Not null
SSD	Varchar	-	Not Null
RAM	Varchar	-	Not Null
Display	Varchar	-	Not null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 34. iMac_models

16. Таблица server_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 35. server_models

17. Таблица switch_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 36. switch_models

18. Таблица TV_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 37. TV_models

19. Таблица UPS_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 38. UPS_models

20. Таблица keyboard_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 39. keyboard_models

21. Таблица printer_models

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Model	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Price	Integer	-	Not Null
Supplier_id	Smallint	Foreign_key	Not Null

Таблица 40. printer_models

22. Таблица tech_suppliers

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Supplier_id	Smallint	Primary Key	Not Null
Tech_type	Varchar	-	Not Null
Company	Varchar	-	Not Null
Website	Varchar	-	Not Null
phone	Varchar	-	Null

Таблица 41. tech_suppliers

23. Таблица programm_suppliers

Наименование	Тип данных	Первичный/внешний ключ	Not null/null
Supplier_id	Smallint	Primary Key	Not Null
Company	Varchar	-	Not Null
Website	Varchar	-	Not Null
phone	Varchar	-	Null

Таблица 42. programm_suppliers

Глава 2. Реализация БД

2.1 DDL-Сценарий создания БД

Скрипт создание Базы данных + DDL-скрипт.

```
-- Generated by Oracle SQL Developer Data Modeler 21.4.2.059.0838
-- at:      2022-06-09 23:41:43 MSK
-- site:    Oracle Database 11g
-- type:    Oracle Database 11g
```

```
Create Database Kursovaya on (
  Name = Kursovaya,
  Filename = 'd:\SQL\Kursovaya.mdf',
  Size = 10,
  Filegrowth = 5)
```

```
Log on (
  Name = 'Torg_log',
  Filename = 'd:\SQL\Kursovaya.ldf',
  Size = 5MB,
  Filegrowth = 5MB)
```

```
CREATE TABLE cabinets_workplaces_info (
  workplace_id SMALLINT NOT NULL,
  room_number CHAR(3),
  emp_id INTEGER NOT NULL,
  mouse_id SMALLINT NOT NULL,
  imac_id SMALLINT NOT NULL,
  tv_id SMALLINT NOT NULL,
  keyboard_id SMALLINT NOT NULL,
  ups_id SMALLINT NOT NULL,
  printer_id SMALLINT NOT NULL
);
```

```
CREATE UNIQUE INDEX cabinets_info_idx ON
  cabinets_workplaces_info (
    emp_id
  ASC );
```

```
CREATE UNIQUE INDEX cabinets_info_idxv1 ON
  cabinets_workplaces_info (
    imac_id
  ASC );
```

```
CREATE UNIQUE INDEX cabinets_info_idxv2 ON
  cabinets_workplaces_info (
    ups_id
  ASC );
```

```
CREATE UNIQUE INDEX cabinets_info_idxv3 ON
```

```

cabinets_workplaces_info (
    keyboard_id
ASC );

CREATE UNIQUE INDEX cabinets_info_idxv4 ON
cabinets_workplaces_info (
    mouse_id
ASC );

CREATE UNIQUE INDEX cabinets_info_idxv5 ON
cabinets_workplaces_info (
    printer_id
ASC );

CREATE UNIQUE INDEX cabinets_info_idxv6 ON
cabinets_workplaces_info (
    tv_id
ASC );

ALTER TABLE cabinets_workplaces_info ADD CONSTRAINT workplace_id_pkv1
PRIMARY KEY ( workplace_id );

CREATE TABLE clusters_workplaces_info (
    workplace_id SMALLINT NOT NULL,
    cluster_name CHAR(20),
    "row" CHAR(1),
    table_number CHAR(1),
    mouse_id SMALLINT NOT NULL,
    imac_id SMALLINT NOT NULL,
    ups_id SMALLINT NOT NULL,
    keyboard_id SMALLINT NOT NULL,
    emp_id INTEGER NOT NULL
);

CREATE UNIQUE INDEX clusters_info_idx ON
clusters_workplaces_info (
    mouse_id
ASC );

CREATE UNIQUE INDEX clusters_info_idxv1 ON
clusters_workplaces_info (
    imac_id
ASC );

CREATE UNIQUE INDEX clusters_info_idxv2 ON
clusters_workplaces_info (
    ups_id
ASC );

CREATE UNIQUE INDEX clusters_info_idxv3 ON
clusters_workplaces_info (
    keyboard_id

```

ASC);

```
CREATE UNIQUE INDEX clusters_info_idxv4 ON
clusters_workplaces_info (
    emp_id
ASC );
```

```
ALTER TABLE clusters_workplaces_info ADD CONSTRAINT workplace_id_pk PRIMARY
KEY ( workplace_id );
```

```
CREATE TABLE employee_info (
    emp_id          INTEGER NOT NULL,
    name            CHAR(20),
    sirname         CHAR(20),
    fathurname      CHAR(20),
    sex             CHAR(1),
    birth_date      DATE,
    passport_id     CHAR(8),
    living_address  CHAR(100),
    personal_phone_number CHAR(14),
    work_phone_number CHAR(14),
    position        CHAR(20),
    salary          INTEGER,
    office_id       SMALLINT NOT NULL
);
```

```
ALTER TABLE employee_info ADD CONSTRAINT emp_id PRIMARY KEY ( emp_id );
```

```
CREATE TABLE imac_ids (
    imac_id SMALLINT NOT NULL,
    status CHAR(10),
    problem CHAR(512),
    model CHAR(20) NOT NULL,
    office_id SMALLINT NOT NULL
);
```

```
ALTER TABLE imac_ids ADD CONSTRAINT список_пк_pk PRIMARY KEY ( imac_id );
```

```
CREATE TABLE imac_models (
    model CHAR(20) NOT NULL,
    cpu CHAR(30),
    gpu CHAR(30),
    ssd CHAR(20),
    ram CHAR(20),
    display CHAR(40),
    price SMALLINT,
    supplier_id SMALLINT NOT NULL
);
```

```
ALTER TABLE imac_models ADD CONSTRAINT model_pkv6 PRIMARY KEY ( model );
```

```
CREATE TABLE keyboard_ids (
```



```

keyboard_id SMALLINT NOT NULL,
status CHAR(10),
problem CHAR(512),
model CHAR(20) NOT NULL,
office_id SMALLINT NOT NULL
);

```

```

ALTER TABLE keyboard_ids ADD CONSTRAINT список_клавиатур_pk PRIMARY KEY (
keyboard_id );

```

```

CREATE TABLE "keyboard models" (
model CHAR(20) NOT NULL,
company CHAR(20),
price INTEGER,
supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE "keyboard models" ADD CONSTRAINT model_pkv1 PRIMARY KEY (
model );

```

```

CREATE TABLE mouse_ids (
mouse_id SMALLINT NOT NULL,
status CHAR(10),
problem CHAR(512),
model CHAR(20) NOT NULL,
office_id SMALLINT NOT NULL
);

```

```

ALTER TABLE mouse_ids ADD CONSTRAINT список_мышек_pk PRIMARY KEY (
mouse_id );

```

```

CREATE TABLE "mouse models" (
model CHAR(20) NOT NULL,
company CHAR(20),
price INTEGER,
supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE "mouse models" ADD CONSTRAINT model_pkv7 PRIMARY KEY ( model
);

```

```

CREATE TABLE office_info (
office_id SMALLINT NOT NULL,
city CHAR(20) NOT NULL,
post_index CHAR(6) NOT NULL,
adress CHAR(100) NOT NULL,
amount_of_workers SMALLINT NOT NULL
);

```

```

ALTER TABLE office_info ADD CONSTRAINT office_id_pk PRIMARY KEY ( office_id );

```

```

CREATE TABLE printer_ids (

```

```

printer_id SMALLINT NOT NULL,
status CHAR(10),
problem CHAR(512),
model CHAR(20) NOT NULL,
office_id SMALLINT NOT NULL
);

```

```

ALTER TABLE printer_ids ADD CONSTRAINT список_принтеров_pk PRIMARY KEY (
printer_id );

```

```

CREATE TABLE "printer models" (
model CHAR(20) NOT NULL,
company CHAR(20),
price INTEGER,
supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE "printer models" ADD CONSTRAINT model_pkv2 PRIMARY KEY ( model
);

```

```

CREATE TABLE programm_suppliers (
supplier_id CHAR(2) NOT NULL,
company CHAR(20),
website CHAR(512),
phone CHAR(15)
);

```

```

ALTER TABLE programm_suppliers ADD CONSTRAINT supplier_id_pkv1 PRIMARY KEY
( supplier_id );

```

```

CREATE TABLE server_ids (
server_id SMALLINT NOT NULL,
status CHAR(10),
problem CHAR(512),
office_id SMALLINT NOT NULL,
model CHAR(20) NOT NULL
);

```

```

ALTER TABLE server_ids ADD CONSTRAINT список_серверов_pk PRIMARY KEY (
server_id );

```

```

CREATE TABLE server_models (
model CHAR(20) NOT NULL,
company CHAR(20),
price INTEGER,
supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE server_models ADD CONSTRAINT model_pkv5 PRIMARY KEY ( model );

```

```

CREATE TABLE server_tables_info (
place_id SMALLINT NOT NULL,

```

```
server_id SMALLINT NOT NULL,  
switch_id SMALLINT NOT NULL,  
office_id SMALLINT NOT NULL  
);
```

```
ALTER TABLE server_tables_info ADD CONSTRAINT place_id_pk PRIMARY KEY (  
place_id );
```

```
CREATE TABLE switch_ids (  
switch_id SMALLINT NOT NULL,  
status CHAR(10),  
problem CHAR(512),  
office_id SMALLINT NOT NULL,  
model CHAR(20) NOT NULL  
);
```

```
ALTER TABLE switch_ids ADD CONSTRAINT список_коммутаторов_pk PRIMARY KEY  
( switch_id );
```

```
CREATE TABLE switch_models (  
model CHAR(20) NOT NULL,  
company CHAR(20),  
price INTEGER,  
supplier_id SMALLINT NOT NULL  
);
```

```
ALTER TABLE switch_models ADD CONSTRAINT model_pkv4 PRIMARY KEY ( model );
```

```
CREATE TABLE tech_suppliers (  
supplier_id SMALLINT NOT NULL,  
tech_type CHAR(15),  
company CHAR(15),  
website CHAR(512),  
phone CHAR(15)  
);
```

```
ALTER TABLE tech_suppliers ADD CONSTRAINT supplier_id_pk PRIMARY KEY (  
supplier_id );
```

```
CREATE TABLE tv_ids (  
tv_id SMALLINT NOT NULL,  
status CHAR(10),  
problem CHAR(512),  
model CHAR(20) NOT NULL,  
office_id SMALLINT NOT NULL  
);
```

```
ALTER TABLE tv_ids ADD CONSTRAINT список_телевизоров_pk PRIMARY KEY ( tv_id  
);
```

```
CREATE TABLE "TV models" (  
model CHAR(20) NOT NULL,
```

```

    company    CHAR(20),
    price      INTEGER,
    supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE "TV models" ADD CONSTRAINT model_pkv3 PRIMARY KEY ( model );

```

```

CREATE TABLE ups_ids (
    ups_id     SMALLINT NOT NULL,
    status     CHAR(10),
    problem    CHAR(512),
    model      CHAR(20) NOT NULL,
    office_id  SMALLINT NOT NULL
);

```

```

ALTER TABLE ups_ids ADD CONSTRAINT список_бесперебойников_pk PRIMARY KEY
( ups_id );

```

```

CREATE TABLE "UPS models" (
    model      CHAR(20) NOT NULL,
    company    CHAR(20),
    price      INTEGER,
    supplier_id SMALLINT NOT NULL
);

```

```

ALTER TABLE "UPS models" ADD CONSTRAINT model_pk PRIMARY KEY ( model );

```

```

ALTER TABLE clusters_workplaces_info
    ADD CONSTRAINT emp_id_fk FOREIGN KEY ( emp_id )
        REFERENCES employee_info ( emp_id );

```

```

ALTER TABLE cabinets_workplaces_info
    ADD CONSTRAINT emp_id_fkv2 FOREIGN KEY ( emp_id )
        REFERENCES employee_info ( emp_id );

```

```

ALTER TABLE clusters_workplaces_info
    ADD CONSTRAINT imac_id_fk FOREIGN KEY ( imac_id )
        REFERENCES imac_ids ( imac_id );

```

```

ALTER TABLE cabinets_workplaces_info
    ADD CONSTRAINT imac_id_fkv1 FOREIGN KEY ( imac_id )
        REFERENCES imac_ids ( imac_id );

```

```

ALTER TABLE clusters_workplaces_info
    ADD CONSTRAINT keyboard_id_fk FOREIGN KEY ( keyboard_id )
        REFERENCES keyboard_ids ( keyboard_id );

```

```

ALTER TABLE cabinets_workplaces_info
    ADD CONSTRAINT keyboard_id_fkv1 FOREIGN KEY ( keyboard_id )
        REFERENCES keyboard_ids ( keyboard_id );

```

```

ALTER TABLE mouse_ids

```

```

ADD CONSTRAINT model_fk FOREIGN KEY ( model )
REFERENCES "mouse models" ( model );

ALTER TABLE imac_ids
ADD CONSTRAINT model_fkv2 FOREIGN KEY ( model )
REFERENCES imac_models ( model );

ALTER TABLE server_ids
ADD CONSTRAINT model_fkv3 FOREIGN KEY ( model )
REFERENCES server_models ( model );

ALTER TABLE switch_ids
ADD CONSTRAINT model_fkv4 FOREIGN KEY ( model )
REFERENCES switch_models ( model );

ALTER TABLE tv_ids
ADD CONSTRAINT model_fkv5 FOREIGN KEY ( model )
REFERENCES "TV models" ( model );

ALTER TABLE ups_ids
ADD CONSTRAINT model_fkv6 FOREIGN KEY ( model )
REFERENCES "UPS models" ( model );

ALTER TABLE keyboard_ids
ADD CONSTRAINT model_fkv7 FOREIGN KEY ( model )
REFERENCES "keyboard models" ( model );

ALTER TABLE printer_ids
ADD CONSTRAINT model_fkv8 FOREIGN KEY ( model )
REFERENCES "printer models" ( model );

ALTER TABLE clusters_workplaces_info
ADD CONSTRAINT mouse_id_fk FOREIGN KEY ( mouse_id )
REFERENCES mouse_ids ( mouse_id );

ALTER TABLE cabinets_workplaces_info
ADD CONSTRAINT mouse_id_fkv2 FOREIGN KEY ( mouse_id )
REFERENCES mouse_ids ( mouse_id );

ALTER TABLE mouse_ids
ADD CONSTRAINT office_id_fk FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE imac_ids
ADD CONSTRAINT office_id_fkv1 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE server_tables_info
ADD CONSTRAINT office_id_fkv2 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE server_ids

```

```

ADD CONSTRAINT office_id_fkv3 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE switch_ids
ADD CONSTRAINT office_id_fkv4 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE tv_ids
ADD CONSTRAINT office_id_fkv5 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE ups_ids
ADD CONSTRAINT office_id_fkv6 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE keyboard_ids
ADD CONSTRAINT office_id_fkv7 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE printer_ids
ADD CONSTRAINT office_id_fkv8 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE employee_info
ADD CONSTRAINT office_id_fkv9 FOREIGN KEY ( office_id )
REFERENCES office_info ( office_id );

ALTER TABLE cabinets_workplaces_info
ADD CONSTRAINT printer_id_fk FOREIGN KEY ( printer_id )
REFERENCES printer_ids ( printer_id );

ALTER TABLE server_tables_info
ADD CONSTRAINT server_id_fk FOREIGN KEY ( server_id )
REFERENCES server_ids ( server_id );

ALTER TABLE "printer models"
ADD CONSTRAINT supplider_id_fk FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE "keyboard models"
ADD CONSTRAINT supplider_id_fkv2 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE "UPS models"
ADD CONSTRAINT supplider_id_fkv3 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE "TV models"
ADD CONSTRAINT supplider_id_fkv4 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE switch_models

```

```

ADD CONSTRAINT supplider_id_fkv5 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE server_models
ADD CONSTRAINT supplider_id_fkv6 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE imac_models
ADD CONSTRAINT supplider_id_fkv7 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE "mouse models"
ADD CONSTRAINT supplider_id_fkv8 FOREIGN KEY ( supplier_id )
REFERENCES tech_suppliers ( supplier_id );

ALTER TABLE server_tables_info
ADD CONSTRAINT switch_id_fk FOREIGN KEY ( switch_id )
REFERENCES switch_ids ( switch_id );

ALTER TABLE cabinets_workplaces_info
ADD CONSTRAINT tv_id_fk FOREIGN KEY ( tv_id )
REFERENCES tv_ids ( tv_id );

ALTER TABLE clusters_workplaces_info
ADD CONSTRAINT ups_id_fk FOREIGN KEY ( ups_id )
REFERENCES ups_ids ( ups_id );

ALTER TABLE cabinets_workplaces_info
ADD CONSTRAINT ups_id_fkv2 FOREIGN KEY ( ups_id )
REFERENCES ups_ids ( ups_id );

```

-- Oracle SQL Developer Data Modeler Summary Report:

```

--
-- CREATE TABLE                23
-- CREATE INDEX                 12
-- ALTER TABLE                 63
-- CREATE VIEW                   0
-- ALTER VIEW                    0
-- CREATE PACKAGE                0
-- CREATE PACKAGE BODY           0
-- CREATE PROCEDURE              0
-- CREATE FUNCTION               0
-- CREATE TRIGGER                0
-- ALTER TRIGGER                 0
-- CREATE COLLECTION TYPE        0
-- CREATE STRUCTURED TYPE        0
-- CREATE STRUCTURED TYPE BODY   0
-- CREATE CLUSTER                0
-- CREATE CONTEXT                0
-- CREATE DATABASE               0

```

```

-- CREATE DIMENSION                0
-- CREATE DIRECTORY                  0
-- CREATE DISK GROUP                  0
-- CREATE ROLE                        0
-- CREATE ROLLBACK SEGMENT            0
-- CREATE SEQUENCE                    0
-- CREATE MATERIALIZED VIEW            0
-- CREATE MATERIALIZED VIEW LOG        0
-- CREATE SYNONYM                     0
-- CREATE TABLESPACE                 0
-- CREATE USER                        0
--
-- DROP TABLESPACE                   0
-- DROP DATABASE                      0
--
-- REDACTION POLICY                   0
--
-- ORDS DROP SCHEMA                   0
-- ORDS ENABLE SCHEMA                  0
-- ORDS ENABLE OBJECT                  0
--
-- ERRORS                             0
-- WARNINGS                           0

```

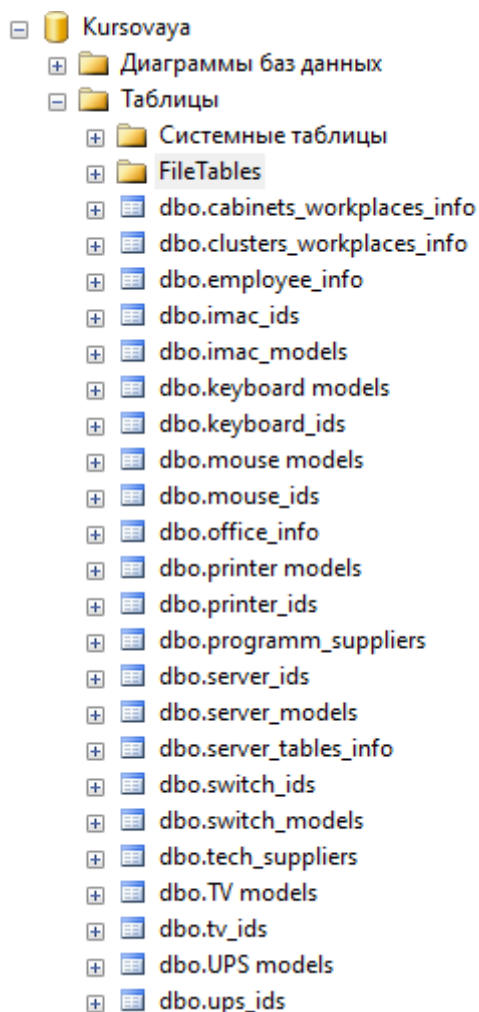


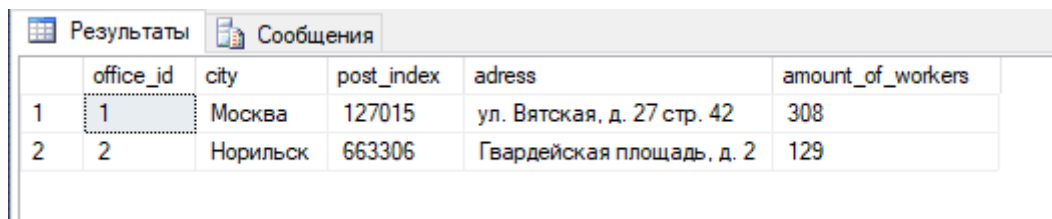
Рисунок 6. Успешно созданная БД и таблицы к ней

2.2 Заполнение таблиц через SQL-скрипт

1. Заполнение таблицы office_info:

```
INSERT INTO office_info  
VALUES('1', 'Москва', '127015', 'ул. Вятская, д. 27 стр. 42', '308')
```

```
INSERT INTO office_info  
VALUES('2', 'Норильск', '663306', 'Гвардейская площадь, д. 2', 129)
```



	office_id	city	post_index	adress	amount_of_workers
1	1	Москва	127015	ул. Вятская, д. 27 стр. 42	308
2	2	Норильск	663306	Гвардейская площадь, д. 2	129

Рисунок 7. Успешно заполненная таблица office_info

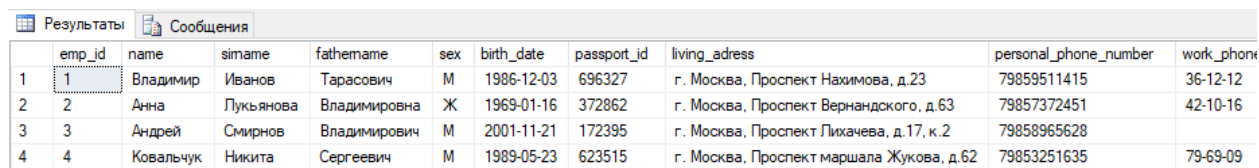
2. Заполнение таблицы employee_info:

```
INSERT INTO employee_info  
VALUES('1', 'Владимир', 'Иванов', 'Тарасович', 'М', '03-12-1986', '696327', 'г.  
Москва, Проспект Нахимова, д.23', '79859511415', '36-12-12', 'Middle  
разработчик', '140000', '1')
```

```
INSERT INTO employee_info  
VALUES('2', 'Анна', 'Лукьянова', 'Владимировна', 'Ж', '16-01-1969', '372862', 'г.  
Москва, Проспект Вернадского, д.63', '79857372451', '42-10-16',  
'руководитель HR', '210000', '1')
```

```
INSERT INTO employee_info  
VALUES('3', 'Андрей', 'Смирнов', 'Владимирович', 'М', '21-11-2001', '172395',  
'г. Москва, Проспект Лихачева, д.17, к.2', '79858965628', '', 'Стажер MIOps',  
'62000', '1')
```

```
INSERT INTO employee_info  
VALUES('4', 'Ковальчук', 'Никита', 'Сергеевич', 'М', '23-05-1989', '623515', 'г.  
Москва, Проспект маршала Жукова, д.62', '79853251635', '79-69-09',  
'TeamLead аналитики', '246000', '1')
```



	emp_id	name	sname	fathename	sex	birth_date	passport_id	living_adress	personal_phone_number	work_phone
1	1	Владимир	Иванов	Тарасович	М	1986-12-03	696327	г. Москва, Проспект Нахимова, д.23	79859511415	36-12-12
2	2	Анна	Лукьянова	Владимировна	Ж	1969-01-16	372862	г. Москва, Проспект Вернадского, д.63	79857372451	42-10-16
3	3	Андрей	Смирнов	Владимирович	М	2001-11-21	172395	г. Москва, Проспект Лихачева, д.17, к.2	79858965628	
4	4	Ковальчук	Никита	Сергеевич	М	1989-05-23	623515	г. Москва, Проспект маршала Жукова, д.62	79853251635	79-69-09

Рисунок 8. Успешно заполненная таблица employee_info

3. Заполнение таблицы tech_suppliers:

```
insert into tech_suppliers  
values('0', 'PC', 'Apple', 'https://www.apple.com/', '8-800-555-67-34')
```

```
insert into tech_suppliers  
values('1', 'mouse', 'Logitech', 'https://www.logitech.com/', '7-347-225-61-47')
```

```
insert into tech_suppliers  
values('2', 'keyboard', 'Apple', 'https://www.apple.com/', '8-800-555-67-34')
```

```
insert into tech_suppliers  
values('3', 'UPS', 'APC', 'https://www.apc.com/', '0-800-279-92-54')
```

```
insert into tech_suppliers  
values('4', 'Switch', 'Cisco', 'https://www.cisco.com/', '')
```

```
insert into tech_suppliers  
values('5', 'TV', 'Samsung', 'https://www.samsung.com/us/', '')
```

```
insert into tech_suppliers  
values('6', 'Printer', 'Xerox', 'https://www.xerox.com/en-us', '1-800-821-2797')
```

```
insert into tech_suppliers  
values('7', 'server', 'Cisco', 'https://www.cisco.com/', '')
```



	supplier_id	tech_type	company	website	phone
1	0	PC	Apple	https://www.apple.com/	8-800-555-67-34
2	1	mouse	Logitech	https://www.logitech.com/	7-347-225-61-47
3	2	keyboard	Apple	https://www.apple.com/	8-800-555-67-34
4	3	UPS	APC	https://www.apc.com/	0-800-279-92-54
5	4	Switch	Cisco	https://www.cisco.com/	
6	5	TV	Samsung	https://www.samsung.com/us/	
7	6	Printer	Xerox	https://www.xerox.com/en-us	1-800-821-2797
8	7	server	Cisco	https://www.cisco.com/	

Рисунок 9. Успешно заполненная таблица tech_suppliers

4. Заполнение таблицы programm_suppliers:

```
insert into programm_suppliers  
values('0', 'JetBrains', 'https://www.jetbrains.com/', '')
```

```
insert into programm_suppliers  
values('1', 'Microsoft', 'https://www.microsoft.com/ru-ru/', '')
```

```
insert into programm_suppliers
values('2', 'Jira', 'https://www.atlassian.com/software/jira', '')
```

```
insert into programm_suppliers
values('3', 'Git', 'https://github.com/', '')
```

	supplier_id	company	website	phone
1	0	JetBrains	https://www.jetbrains.com/	
2	1	Microsoft	https://www.microsoft.com/ru-ru/	
3	2	Jira	https://www.atlassian.com/software/jira	
4	3	Git	https://github.com/	

Рисунок 10. Успешно заполненная таблица *programm_suppliers*

5. Заполнение таблицы *mouse_models*:

```
insert into [mouse models]
values('MX anywhere 3', 'Logitech', '63', '1')
```

	model	company	price	supplier_id
1	MX anywhere 3	Logitech	63	1

Рисунок 11. Успешно заполненная таблица *mouse_models*

6. Заполнение таблицы *iMac_models*:

```
insert into imac_models
values('iMac_24_V1', 'M1 8cores', 'M1 8cores', '512 GB', '8GB', '24-inch 4.5K Retina Display', '1699', '0')
```

```
insert into imac_models
values('iMac_24_V2', 'M1 8cores', 'M1 8cores', '1 TB', '16GB', '24-inch 4.5K Retina Display', '2099', '0')
```

```
insert into imac_models
values('iMac_27_V1', 'Intel Core i7 8cores 3.8GHz', 'Radeon Pro 5500 XT', '1 TB', '16GB', '27-inch 5K Retina Display', '2799', '0')
```

```
insert into imac_models
values('iMac_27_V2', 'Intel Core i9 10cores 3.6Ghz', 'Radeon Pro 5500 XT', '1 TB', '16GB', '27-inch 5K Retina Display', '3199', '0')
```

```
insert into imac_models
values('iMac_27_V3', 'Intel Core i9 10cores 3.6Ghz', 'Radeon Pro 5700', '1 TB', '32GB', '27-inch 5K Retina Display', '3499', '0')
```

	model	cpu	gpu	ssd	ram	display	price	supplier_id
1	iMac_24_V1	M1 8cores	M1 8cores	512 GB	8GB	24-inch 4.5K Retina Display	1699	0
2	iMac_24_V2	M1 8cores	M1 8cores	1 TB	16GB	24-inch 4.5K Retina Display	2099	0
3	iMac_27_V1	Intel Core i7 8cores 3.8GHz	Radeon Pro 5500 XT	1 TB	16GB	27-inch 5K Retina Display	2799	0
4	iMac_27_V2	Intel Core i9 10cores 3.6Ghz	Radeon Pro 5500 XT	1 TB	16GB	27-inch 5K Retina Display	3199	0
5	iMac_27_V3	Intel Core i9 10cores 3.6Ghz	Radeon Pro 5700	1 TB	32GB	27-inch 5K Retina Display	3499	0

Рисунок 12. Успешно заполненная таблица iMac_models

7. Заполнение таблицы server_models

insert into server_models

values ('HPE ProLiant DL380', 'Cisco', '1099', '7')

	model	company	price	supplier_id
1	HPE ProLiant DL380	Cisco	1099	7

Рисунок 13. Успешно заполненная таблица server_models

8. Заполнение таблицы switch_models

insert into switch_models

values ('N77-C7710-B36S3E', 'Cisco', '1099', '4')

	model	company	price	supplier_id
1	N77-C7710-B36S3E	Cisco	1099	4

Рисунок 14. Успешно заполненная таблица switch_models

9. Заполнение таблицы TV_models

insert into [TV models]

values('QE65Q60ABUXRU', 'Samsung', '2000', '5')

	model	company	price	supplier_id
1	QE65Q60ABUXRU	Samsung	2000	5

Рисунок 15. Успешно заполненная таблица TV_models

10. Заполнение таблицы UPS_models

insert into [UPS models]

values('BX 1400UI', 'APC', '44', '3')

Результаты		Сообщения		
	model	company	price	supplier_id
1	BX1400UI	APC	44	3

Рисунок 16. Успешно заполненная таблица *UPS_models*

11. Заполнение таблицы **keyboard_models**

```
insert into [keyboard models]
values('MagicKeyboard NumPad', 'Apple', '60', '2')
```

Результаты		Сообщения		
	model	company	price	supplier_id
1	MagicKeyboard NumPad	Apple	60	2

Рисунок 17. Успешно заполненная таблица *keyboard_models*

12. Заполнение таблицы **printer_models**

```
insert into [printer models]
values('Xerox B215', 'Xerox', '400', '6')
```

Результаты		Сообщения		
	model	company	price	supplier_id
1	Xerox B215	Xerox	400	6

Рисунок 18. Успешно заполненная таблица *printer_models*

13. Заполнение таблицы **mouse_ids**

```
insert into mouse_ids
values('0', 'active', ", 'MX anywhere 3', '1')
```

```
insert into mouse_ids
values('1', 'active', ", 'MX anywhere 3', '1')
```

```
insert into mouse_ids
values('2', 'active', ", 'MX anywhere 3', '1')
```

```
insert into mouse_ids
values('3', 'active', ", 'MX anywhere 3', '1')
```

Результаты		Сообщения			
	mouse_id	status	problem	model	office_id
1	0	active		MX anywhere 3	1
2	1	active		MX anywhere 3	1
3	2	active		MX anywhere 3	1
4	3	active		MX anywhere 3	1

Рисунок 19. Успешно заполненная таблица *mouse_ids*

14. Заполнение таблицы *iMac_ids*

```
insert into imac_ids
values('0', 'active', '', 'iMac_27_V2', '1')
```

```
insert into imac_ids
values('1', 'active', '', 'iMac_27_V1', '1')
```

```
insert into imac_ids
values('2', 'active', '', 'iMac_24_V1', '1')
```

```
insert into imac_ids
values('3', 'active', '', 'iMac_27_V3', '1')
```

Результаты		Сообщения			
	imac_id	status	problem	model	office_id
1	0	active		iMac_27_V2	1
2	1	active		iMac_27_V1	1
3	2	active		iMac_24_V1	1
4	3	active		iMac_27_V3	1

Рисунок 20. Успешно заполненная таблица *iMac_ids*

15. Заполнение таблицы *server_ids*

```
insert into server_ids
values('0', 'active', '', '1', 'HPE ProLiant DL380')
```

```
insert into imac_ids
values('1', 'active', '', '1', 'HPE ProLiant DL380')
```

```
insert into imac_ids
values('2', 'active', '', '1', 'HPE ProLiant DL380')
```

```
insert into imac_ids
values('3', 'active', '', '1', 'HPE ProLiant DL380')
```

Результаты		Сообщения			
	server_id	status	problem	office_id	model
1	0	active		1	HPE ProLiant DL380
2	1	active		1	HPE ProLiant DL380
3	2	active		1	HPE ProLiant DL380
4	3	active		1	HPE ProLiant DL380

Рисунок 21. Успешно заполненная таблица server_ids

16. Заполнение таблицы switch_ids

insert into switch_ids

values('0', 'active', '', '1', 'N77-C7710-B36S3E')

insert into switch_ids

values('1', 'active', '', '1', 'N77-C7710-B36S3E')

insert into switch_ids

values('2', 'active', '', '1', 'N77-C7710-B36S3E')

insert into switch_ids

values('3', 'active', '', '1', 'N77-C7710-B36S3E')

Результаты		Сообщения			
	switch_id	status	problem	office_id	model
1	0	active		1	N77-C7710-B36S3E
2	1	active		1	N77-C7710-B36S3E
3	2	active		1	N77-C7710-B36S3E
4	3	active		1	N77-C7710-B36S3E

Рисунок 22. Успешно заполненная таблица switch_ids

17. Заполнение таблицы TV_ids

insert into tv_ids

values('0', 'active', '', 'QE65Q60ABUXRU', '1')

insert into tv_ids

values('1', 'active', '', 'QE65Q60ABUXRU', '1')

Результаты		Сообщения			
	tv_id	status	problem	model	office_id
1	0	active		QE65Q60ABUXRU	1
2	1	active		QE65Q60ABUXRU	1

Рисунок 23. Успешно заполненная таблица TV_ids

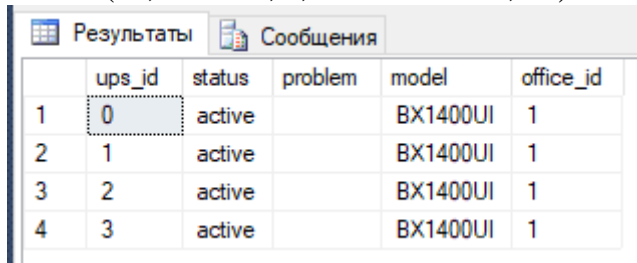
18. Заполнение таблицы UPS_ids

```
insert into ups_ids
values('0', 'active', '', 'BX 1400UI', '1')
```

```
insert into ups_ids
values('1', 'active', '', 'BX 1400UI', '1')
```

```
insert into ups_ids
values('2', 'active', '', 'BX 1400UI', '1')
```

```
insert into ups_ids
values('3', 'active', '', 'BX 1400UI', '1')
```



	ups_id	status	problem	model	office_id
1	0	active		BX1400UI	1
2	1	active		BX1400UI	1
3	2	active		BX1400UI	1
4	3	active		BX1400UI	1

Рисунок 24. Успешно заполненная таблица UPS_ids

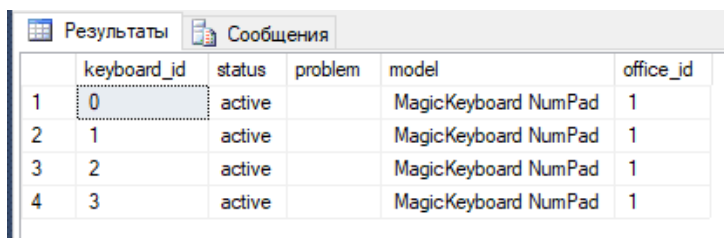
19. Заполнение таблицы keyboards_ids

```
insert into keyboard_ids
values('0', 'active', '', 'MagicKeyboard NumPad', '1')
```

```
insert into keyboard_ids
values('1', 'active', '', 'MagicKeyboard NumPad', '1')
```

```
insert into keyboard_ids
values('2', 'active', '', 'MagicKeyboard NumPad', '1')
```

```
insert into keyboard_ids
values('3', 'active', '', 'MagicKeyboard NumPad', '1')
```



	keyboard_id	status	problem	model	office_id
1	0	active		MagicKeyboard NumPad	1
2	1	active		MagicKeyboard NumPad	1
3	2	active		MagicKeyboard NumPad	1
4	3	active		MagicKeyboard NumPad	1

Рисунок 25. Успешно заполненная таблица keyboards_ids

20. Заполнение таблицы printer_ids

```
insert into printer_ids
values('0', 'active', '', 'Xerox B215', '1')
```

```
insert into printer_ids
```



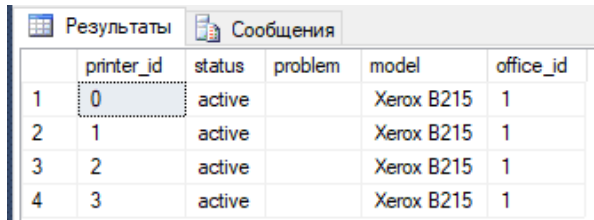
```
values('1', 'active', '', 'Xerox B215', '1')
```

```
insert into printer_ids
```

```
values('2', 'active', '', 'Xerox B215', '1')
```

```
insert into printer_ids
```

```
values('3', 'active', '', 'Xerox B215', '1')
```



	printer_id	status	problem	model	office_id
1	0	active		Xerox B215	1
2	1	active		Xerox B215	1
3	2	active		Xerox B215	1
4	3	active		Xerox B215	1

Рисунок 26. Успешно заполненная таблица *printer_ids*

21. Заполнение таблицы *server_tables_info*

```
insert into server_tables_info
```

```
values('0', '0', '0', '1')
```

```
insert into server_tables_info
```

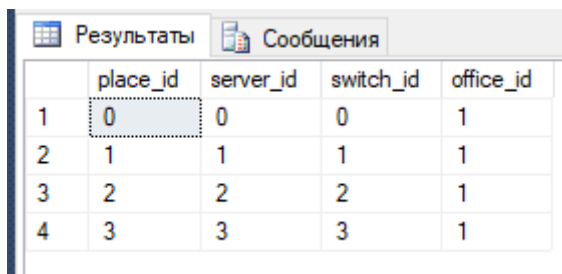
```
values('1', '1', '1', '1')
```

```
insert into server_tables_info
```

```
values('2', '2', '2', '1')
```

```
insert into server_tables_info
```

```
values('3', '3', '3', '1')
```



	place_id	server_id	switch_id	office_id
1	0	0	0	1
2	1	1	1	1
3	2	2	2	1
4	3	3	3	1

Рисунок 27. Успешно заполненная таблица *server_tables_info*

22. Заполнение таблицы *clusters_workplaces_info*

```
insert into clusters_workplaces_info
```

```
values('0', 'mirage', 'c', '3', '0', '0', '0', '0', '1')
```

```
insert into clusters_workplaces_info
```

```
values('1', 'supernova', 'h', '1', '1', '1', '1', '1', '3')
```

```
insert into clusters_workplaces_info
```

values('2', 'atlantis', 'm', '2', '2', '2', '2', '2', '4')

	workplace_id	cluster_name	row	table_number	mouse_id	imac_id	ups_id	keyboard_id	emp_id
1	0	mirage	c	3	0	0	0	0	1
2	1	supernova	h	1	1	1	1	1	3
3	2	atlantis	m	2	2	2	2	2	4

Рисунок 28. Успешно заполненная таблица clusters_workplaces_info

23. Заполнение таблицы cabinets_workplaces_info

insert into cabinets_workplaces_info
values ('0', '302', '2', '3', '3', '0', '3', '3', '')

	workplace_id	room_number	emp_id	mouse_id	imac_id	tv_id	keyboard_id	ups_id	printer_id
1	0	302	2	3	3	0	3	3	0

Рисунок 29. Успешно заполненная таблица cabinets_workplaces_info

2.3 SQL-запросы, Select

A. Select на извлечение данных из связанных таблиц

select sirname, name from employee_info
where (office_id='1' AND salary>'150000')

Запрос на извлечение данных из таблицы employee_info. Запрос извлекает ФИО сотрудников, работающих в офисе с office_id = 1 (в нашем случае это московский офис) и зарплатой выше 150000

	sirname	name	fathename
1	Лукьянова	Анна	Владимировна
2	Никита	Ковальчук	Сергеевич

Рисунок 30. Успешно выполненный запрос select

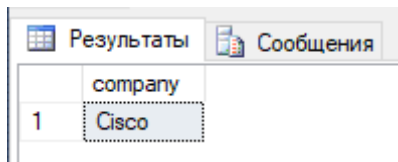
B. Inner Join

select server_models.company from server_models
inner join switch_models on server_models.company = switch_models.company

Запрос на извлечение данных, которые повторяются в 2 или более таблицах.

Запрос извлекает название фирмы-производителя, которые предоставляет

оборудование для разных устройств (в нашем случае этот производитель – cisco, который производит и коммутаторы, и сервера)



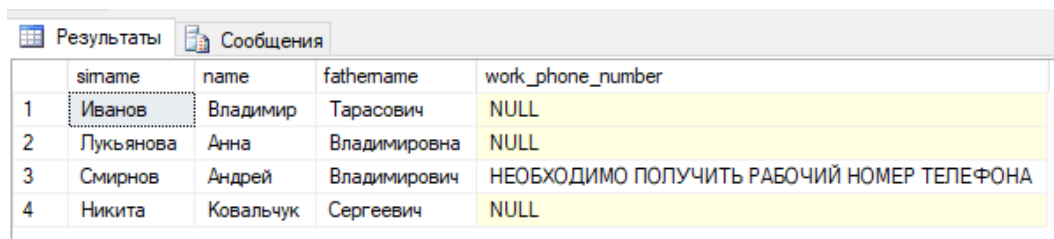
	company
1	Cisco

Рисунок 31. Успешно выполненный запрос Inner Join

C. CASE

```
select employee_info.surname, employee_info.name, employee_info.fathername,
case
    when (work_phone_number = '') then ('НЕОБХОДИМО ПОЛУЧИТЬ
РАБОЧИЙ НОМЕР ТЕЛЕФОНА')
    else (NULL)
end as work_phone_number from employee_info;
```

Запрос на извлечение данных: ФИО сотрудников, у которых отсутствует рабочий номер телефона. В случае отсутствия, в столбце work_phone_number у соответствующего сотрудника появляется запись о том, что ему необходимо получить рабочий номер телефона

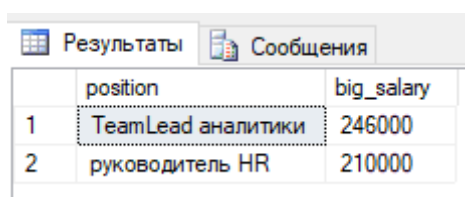


	surname	name	fathername	work_phone_number
1	Иванов	Владимир	Тарасович	NULL
2	Лукьянова	Анна	Владимировна	NULL
3	Смирнов	Андрей	Владимирович	НЕОБХОДИМО ПОЛУЧИТЬ РАБОЧИЙ НОМЕР ТЕЛЕФОНА
4	Никита	Ковальчук	Сергеевич	NULL

Рисунок 32. Успешно выполненный запрос с Case

D. HAVING + GROUP BY

```
select position, max(salary) as big_salary from employee_info
group by position
having max(salary) > 200000
```

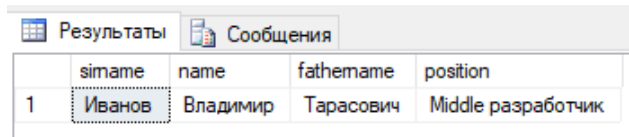


	position	big_salary
1	TeamLead аналитики	246000
2	руководитель HR	210000

Рисунок 33. Успешно выполненный запрос с having+group by

Е. С ВЛОЖЕННЫМ SELECT

```
select sirname, name, fathename, position from employee_info  
where emp_id = (select  
    emp_id from clusters_workplaces_info  
    where cluster_name = 'mirage' and row = 'c' and table_number = '3')
```



	sirname	name	fathename	position
1	Иванов	Владимир	Тарасович	Middle разработчик

Рисунок 34. Успешно выполненный запрос с вложенным Select

Запрос на извлечение данных с внутренним подзапросом, выводящий информацию о сотруднике (ФИО и должность), который закреплен за местом mirage-c-3

Ф. VIEW

```
create view Female_workers as  
select sirname, name, fathename from employee_info  
where sex = 'Ж'
```

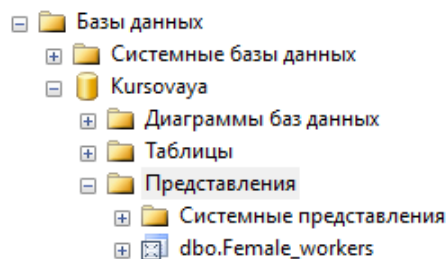
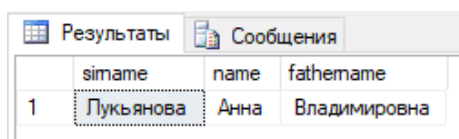


Рисунок 35. Успешное создание представления



	sirname	name	fathename
1	Лукьянова	Анна	Владимировна

Рисунок 36. Содержание представления

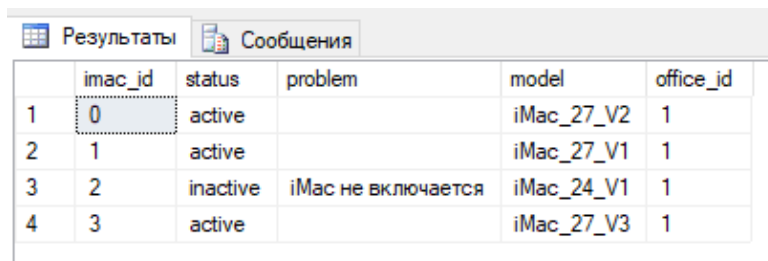
Запрос на создание представления, содержащего ФИО сотрудников женского пола

2.4 Триггеры и их реализация

```
CREATE TRIGGER Problem_INSERT_or_UPDATE
ON iMac_ids AFTER INSERT, UPDATE
AS
    if (UPDATE(problem))
    begin
        update iMac_ids
        set status = 'inactive'
        WHERE iMac_id in (SELECT iMac_id FROM inserted)
    end;

update iMac_ids
set problem = 'iMac не включается'
where iMac_id = 2
```

Выше создали и спровоцировали запуск триггера, который срабатывает на вставку или обновление поля Problem. Если поле изменило свое значение, то у соответствующего iMac меняется статус с active на inactive.



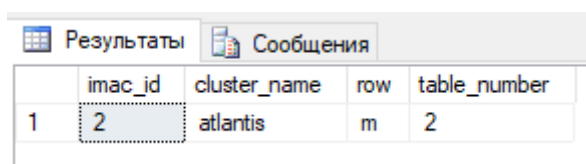
	iMac_id	status	problem	model	office_id
1	0	active		iMac_27_V2	1
2	1	active		iMac_27_V1	1
3	2	inactive	iMac не включается	iMac_24_V1	1
4	3	active		iMac_27_V3	1

Рисунок 37. Успешное выполнение триггера

2.5 Хранимые процедуры и их реализация.

```
CREATE PROCEDURE inactive_iMacs
AS
Select iMac_id=(select iMac_id from iMac_ids where status = 'inactive'),
cluster_name, row, table_number
from clusters_workplaces_info
where iMac_id=(select iMac_id from iMac_ids where status = 'inactive')

exec inactive_iMacs
```



	iMac_id	cluster_name	row	table_number
1	2	atlantis	m	2

Рисунок 38. Успешное выполнение процедуры

Выше создали и запустили процедуру, которая смотрит статус всех iMac. Если какой-то из ПК неактивен, то процедура выведет id этого iMac, а также его местонахождение в кластере

```
CREATE PROCEDURE get_location_by_id
AS
DECLARE @worker_id smallint=2
if (EXISTS(Select emp_id from clusters_workplaces_info where
emp_id=@worker_id))
    select sirname=(select sirname from employee_info where emp_id =
@worker_id),
    name=(select name from employee_info where emp_id =
@worker_id),
    fathename=(select fathename from employee_info where emp_id =
@worker_id),
    position=(select position from employee_info where emp_id =
@worker_id),
    cluster_name, row, table_number from clusters_workplaces_info
where emp_id = @worker_id
else if (EXISTS(Select emp_id from cabinets_workplaces_info where
emp_id=@worker_id))
    select sirname=(select sirname from employee_info where emp_id =
@worker_id),
    name=(select name from employee_info where emp_id =
@worker_id),
    fathename=(select fathename from employee_info where emp_id =
@worker_id),
    position=(select position from employee_info where emp_id =
@worker_id),
    room_number from cabinets_workplaces_info where emp_id =
@worker_id;

exec get_location_by_id
```

Результаты		Сообщения			
	sirname	name	fathename	position	room_number
1	Лукьянова	Анна	Владимировна	руководитель HR	302

Рисунок 39. Успешное выполнение процедуры

Выше создали процедуру, которая по заданному id работника (он записывается в переменную @worker_id) находит его местоположение. Сначала выводится его ФИО, должность, далее – в зависимости от того, где он работает, выводится номер кабинета или место в кластере.

Заключение

Цель курсовой работы – разработка базы данных компании. Успешного выполнения цели первым делом был проведен анализ предметной области, который помог определить бизнес-требования к разрабатываемой базе данных. Так же на практике были отработаны и изучены инструменты построения БД – Oracle Data Modeler и Microsoft SQL Server Management Studio. С помощью Oracle Data Modeler были построены логическая ER-модель и реляционная модель, изучена разница между разными уровнями абстракции представления базы данных и способами переноса информации и структур реального мира в цифровой.

Все поставленные задачи и цели курсовой работы по разработке базы данных выполнены и представлены в виде реляционной, концептуальной модели, подробным описанием их структур и наполнением, последующей реализацией базы данных в Microsoft SQL Server Management Studio.

Список литературы

1. SQL Stored Procedures. — Текст : электронный // w3schools : [сайт]. — URL: https://www.w3schools.com/sql/sql_stored_procedures.asp (дата обращения: 10.06.2022).
2. SQL CREATE VIEW, REPLACE VIEW. — Текст : электронный // w3schools : [сайт]. — URL: https://www.w3schools.com/sql/sql_view.asp (дата обращения: 10.06.2022).
3. Microsoft SQL: <https://docs.microsoft.com/ru-ru/sql/relational-databases/sql-server-guides?view=sql-server-2017> (дата обращения 10.06.2022).

4. SQL CASE Statement. — Текст : электронный // w3shools : [сайт]. — URL: https://www.w3schools.com/sql/sql_case.asp (дата обращения: 10.06.2022).
5. Triggers in SQL Server. — Текст : электронный // C#Corner : [сайт]. — URL: <https://www.c-sharpcorner.com/UploadFile/63f5c2/triggers-in-sql-server/#:~:text=A%20SQL%20trigger%20is%20a%20database%20object%20which%20fires%20when,insert%20in%20a%20database%20table>. (дата обращения: 10.06.2022).
6. SQL HAVING Clause. — Текст : электронный // w3shools : [сайт]. — URL: https://www.w3schools.com/sql/sql_having.asp (дата обращения: 10.06.2022).
7. SQL INNER JOIN Keyword. — Текст : электронный // w3shools : [сайт]. — URL: https://www.w3schools.com/sql/sql_join_inner.asp (дата обращения: 10.06.2022).