**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет»**

**(ГГТУ)**

**Промышленно-экономический колледж**

**О Т Ч Ё Т**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

(код, наименование ПМ)

**ПМ.11 Разработка, администрирование и защита баз данных**

(код, наименование ПМ)

| Студента (ки) | Жагрин Александр Денисович | | |
| --- | --- | --- | --- |
| Курс | 3 | группа | ИСП.21.1А |
|  |  |  |  |
| Специальность | 09.02.07 Информационные системы и программирование | | |
| Место практики | АО «Генериум» | | |
| Период практики | 24.11.2023 г. – 07.12.2023 г. | | |
|  | 08.12.2023 г. – 21.12.2023 г. | | |
| Руководители практики | | | |
| от колледжа | Шалягина Наталья Сергеевна, Сафонова Марина Юрьевна | | |
|  | ФИО  Смирнова Анастасия Николаевна, Фиськова Елена Анатольевна | | |
| от организации | ФИО  Кураев Андрей Вячеславович | | |
|  | ФИО | | |

г. Орехово-Зуево, 2023

# СОДЕРЖАНИЕ

1. Ознакомление с целями и задачами практики.
2. Знакомство с рабочим местом и руководителями практики.
3. Анализ программного и аппаратного обеспечения организации.
4. Разработка программного продукта в соответствии с нуждами организации.
5. Создание базы данных в СУБД.
6. Выводы по практике.

# 1. Ознакомление с целями и задачами практики

Учебная практика проходила в АО «Генериум». Продолжительность учебной практики – 4 недели, с 24 ноября по 21 декабря 2023 года.

Целями практики являются:

* приобретение первоначального практического опыта и первичных профессиональных умений по ПМ.01 и ПМ.11;
* закрепление, углубление знаний и приобретение навыков работы;
* улучшение качества профессиональной подготовки будущего специалиста.

Основными задачами учебной практики являлись:

* формирование у студентов знаний, умений и навыков, профессиональных компетенций, профессионально значимых личностных качеств;
* развитие профессионального интереса, формирование мотивационно-целостного отношения к профессиональной деятельности, готовности к выполнению профессиональных задач в соответствии с нормами морали, профессиональной этики и служебного этикета;
* адаптация студентов к самостоятельной профессиональной деятельности.

В ходе практики были освоены общие и профессиональные компетенции, а именно:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4 Выполнять тестирование программных модулей.

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.

ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных.

ПК 11.2 Проектировать базу данных на основе анализа предметной области.

ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.

ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных.

ПК 11.5. Администрировать базы данных.

ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации.

ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04 Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.

В период учебной практики выполнены следующие работы:

1. Сделано приложение по требованиям организации
2. Разработана База данных для приложения

# 2. Знакомство с рабочим местом и руководителями практики

АО “Генериум” является российской биотехнологической компанией, специализирующейся на разработке, производстве и коммерциализации генно-инженерных препаратов для лечения тяжелых и социально значимых заболеваний. Компания была основана в 2001 году. Оно занимает заметное место на российском рынке биотехнологий. Компания активно сотрудничает с ведущими мировыми производителями генно-терапевтических средств и участвует в международных проектах по разработке и внедрению новых методов лечения.

На предприятии АО “Генериум”, в рамках учебной практики, был обеспечен доступ к рабочему месту, оборудованному современной компьютерной техникой и программным обеспечением, необходимым для выполнения задач практики.

Руководителем практики был назначен специалист отдела информационных технологий – Кураев Андрей Вячеславович. В ходе практики он консультировал по всем возникающим вопросам и оказывал помощь в выполнении задач.

# 3. Анализ программного и аппаратного обеспечения организации

Предприятие использует современную компьютерную технику, соответствующую требованиям рабочих мест. В частности, используются компьютеры на базе процессоров Intel Core i5 и i7 с оперативной памятью от 8 до 16 ГБ (смотрите Рис. 1.), а также мониторы с разрешением от 1920x1080 пикселей. Всего рабочих мест 50

В организации используется широкий спектр программного обеспечения, в том числе операционная система Windows, офисные пакеты Microsoft Office, система управления базами данных 1С:Предприятие, а также специализированное программное обеспечение для решения задач предприятия.



Рисунок 1 – Аппаратное обеспечение.

# 4. Разработка программного продукта в соответствии с нуждами организации

Во время практики была разработана программа для упрощения учёта номеров сотрудников. Программа была разработана на языке программирования C# с использованием WPF для создания графического интерфейса.

Программа позволяет упростить процесс учёта данных о сотрудниках, а также предоставляет возможность создания отчетов в различных форматах.

Функционал программы состоит из: Добавление, редактирование и удаление информации о сотрудниках. Поиск по номеру сотрудника: функция быстрого поиска по номеру сотрудника для быстрого доступа к его данным и контактной информации.

Приложение по учету номеров сотрудников принесет следующую пользу:

– Улучшение коммуникации между сотрудниками: благодаря приложению, сотрудники могут легко найти контактную информацию друг друга и связаться друг с другом, что способствует улучшению коммуникации и координации работы.

Было создано окно MainWindow на котором будут отображаться страницы (смотрите Рис. 2.)

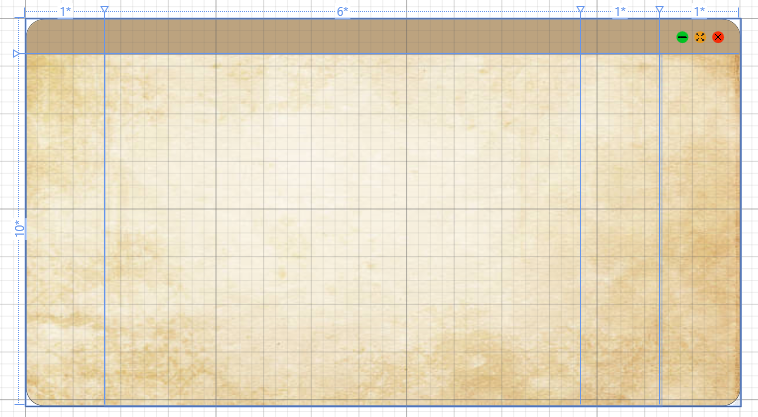


Рисунок 2 – Окно MainWindow.

Создано Окно WindowAdd для добавления новых сотрудников (смотрите Рис. 3.)

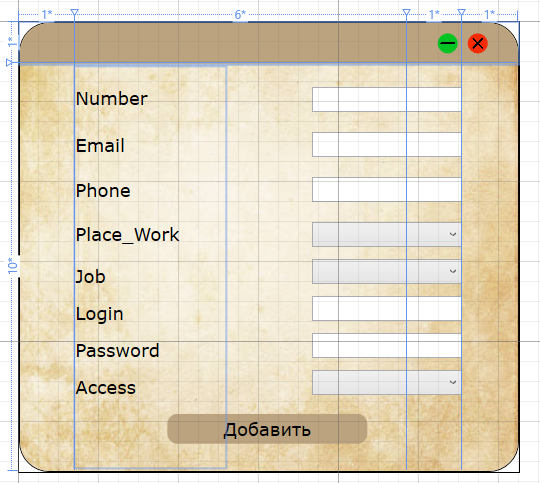


Рисунок 3 – Окно WindowAdd.

Создана страница PageMenu с навигацией по приложению.(смотрите Рис. 4.)

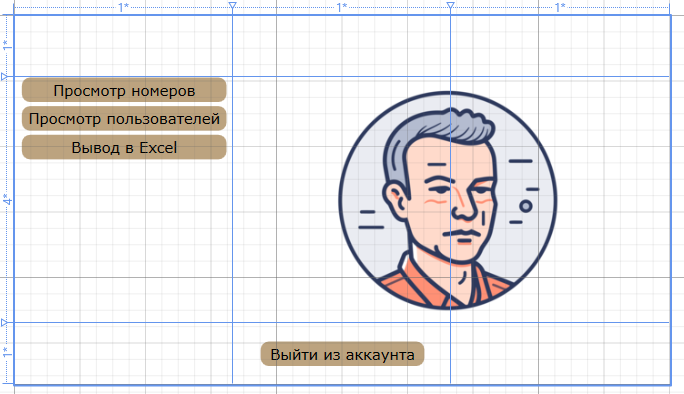


Рисунок 4 – Страница PageMenu.

Была создана страница PageInfo для просмотра сотрудников (смотрите Рис. 5.)

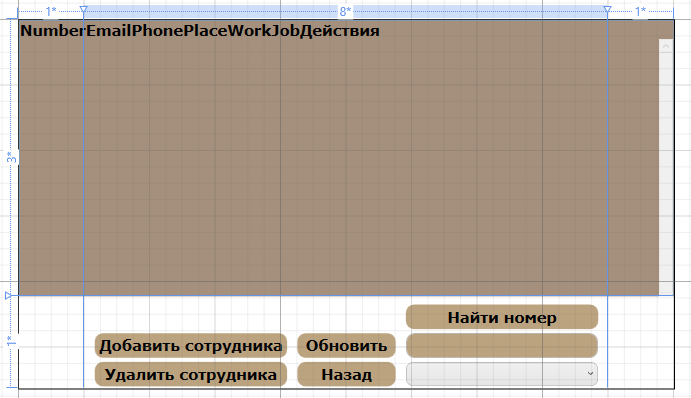


Рисунок 5 – Страница PageInfo.

Была создана страница PageUsers для просмотра пользователей (смотрите Рис. 6.)

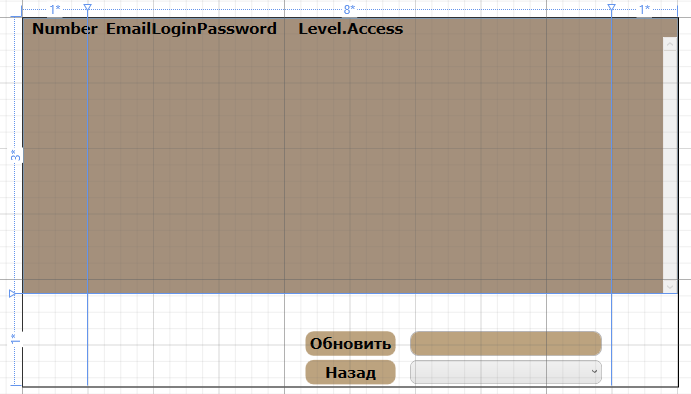


Рисунок 6 – Страница PageUsers.

# 5. Создание базы данных в СУБД

Для создания базы данных приложения была выбрана Microsoft SQL Server Management Studio. Потому что она идеально подходит для выполнения наших задач.

Была создана логическая схема Базы данных (смотрите Рис. 7.)

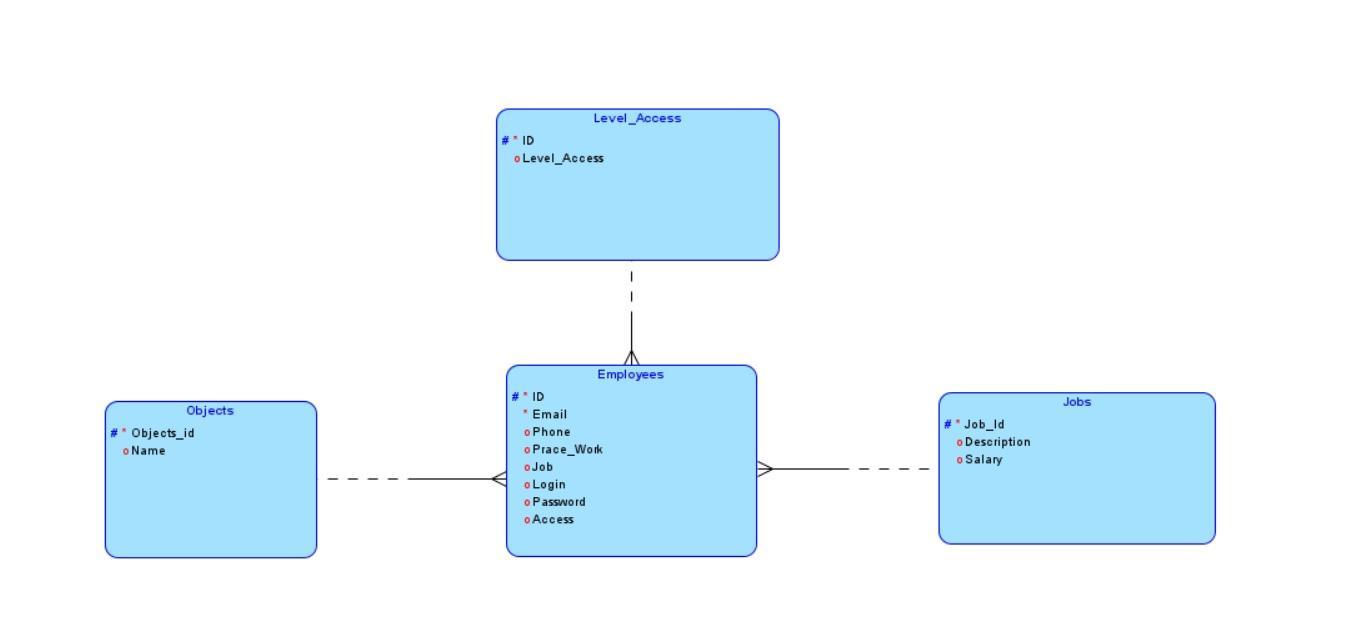


Рисунок 7 – Логическая схема Базы данных

SQL запрос для создания таблиц:

USE [lesuser11]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Employees] Script Date: 12.12.2023 23:16:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Employees](

[ID] [int] NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[Phone] [nvarchar](50) NOT NULL,

[Place\_Work] [int] NOT NULL,

[Job] [int] NOT NULL,

[Login] [nvarchar](50) NOT NULL,

[Password] [nvarchar](50) NOT NULL,

[Access] [int] NOT NULL,

CONSTRAINT [PK\_Employees] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Jobs] Script Date: 12.12.2023 23:16:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Jobs](

[Job\_Id] [int] IDENTITY(1,1) NOT NULL,

[Description] [nvarchar](255) NOT NULL,

[Salary] [int] NOT NULL,

CONSTRAINT [PK\_Jobs] PRIMARY KEY CLUSTERED

(

[Job\_Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Level\_Access] Script Date: 12.12.2023 23:16:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Level\_Access](

[ID] [int] IDENTITY(1,1) NOT NULL,

[Level\_Access] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_Users] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Objects] Script Date: 12.12.2023 23:16:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Objects](

[Objects\_Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](100) NOT NULL,

CONSTRAINT [PK\_objects] PRIMARY KEY CLUSTERED

(

[Objects\_Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK\_Employees\_Jobs] FOREIGN KEY([Job])

REFERENCES [dbo].[Jobs] ([Job\_Id])

GO

ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK\_Employees\_Jobs]

GO

ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK\_Employees\_Level\_Access] FOREIGN KEY([Access])

REFERENCES [dbo].[Level\_Access] ([ID])

GO

ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK\_Employees\_Level\_Access]

GO

ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK\_Employees\_Objects] FOREIGN KEY([Place\_Work])

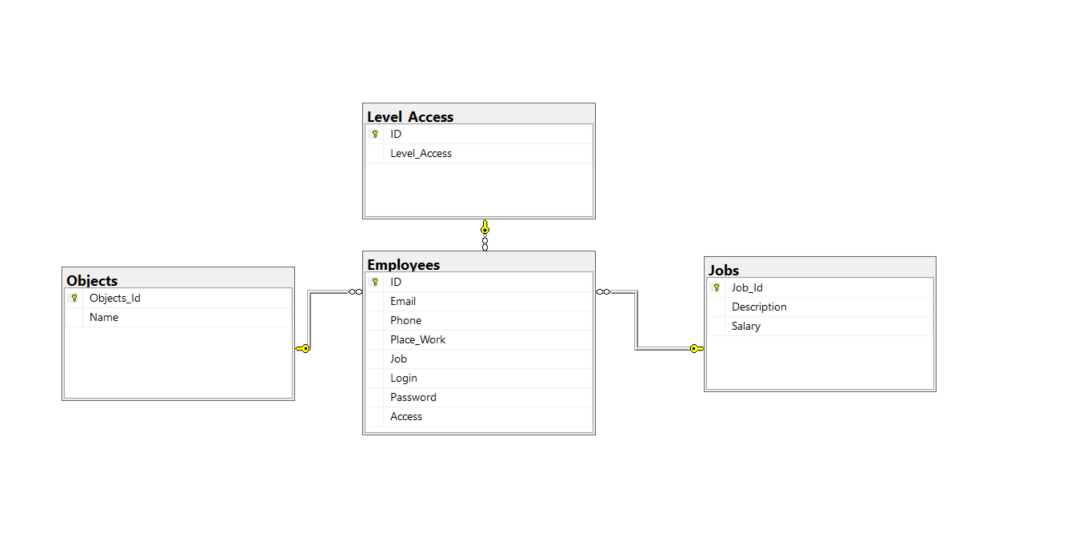
REFERENCES [dbo].[Objects] ([Objects\_Id])

GO

ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK\_Employees\_Objects]

GO

ER Диаграмма в Microsoft SQL Server Management Studio (смотрите Рис. 8.):

Рисунок 8 – ER-Диаграмма Базы данных

UML Диаграмма приложения (смотрите Рис. 9.):

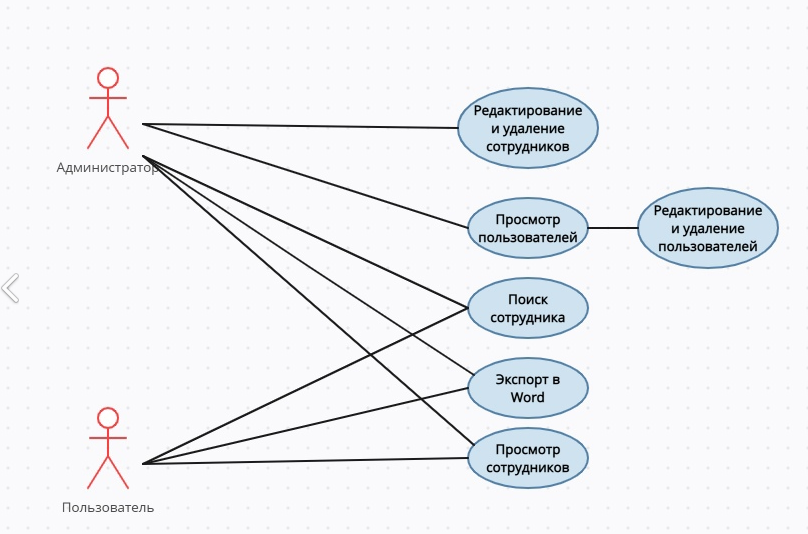


Рисунок 9 – UML-Диаграмма

# 6. Выводы по практике

В ходе практики были исследованы цели и задачи, а также ознакомление с рабочим местом и руководством. Был проведен анализ программного и аппаратного обеспечения организации.

Проанализировав предметную область был создан программный продукт для учета номеров сотрудников, который включает в себя базу данных, созданную в выбранной системе управления базами данных.

Цель практики, поставленная в начале, была достигнута. Все задачи были успешно выполнены, а полученные навыки и знания будут полезны в дальнейшей профессиональной деятельности.

**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет»**

**(ГГТУ)**

**Промышленно-экономический колледж**

**ПРИЛОЖЕНИЯ К ОТЧЁТУ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

(код, наименование ПМ)

**ПМ.11 Разработка, администрирование и защита баз данных**

(код, наименование ПМ)

| Студента (ки) | Жагрин Александр Денисович | | |
| --- | --- | --- | --- |
| Курс | 3 | группа | ИСП.21.1А |
|  |  |  |  |
| Специальность | 09.02.07 Информационные системы и программирование | | |
| Место практики | АО «Генериум» | | |
| Период практики | 24.11.2023 г. – 07.12.2023 г. | | |
|  | 08.12.2023 г. – 21.12.2023 г. | | |
| Руководители практики | | | |
| от колледжа | Шалягина Наталья Сергеевна, Сафонова Марина Юрьевна | | |
|  | ФИО  Смирнова Анастасия Николаевна, Фиськова Елена Анатольевна | | |
| от организации | ФИО  Кураев Андрей Вячеславович | | |
|  | ФИО | | |

г. Орехово-Зуево, 2023

Ответы на вопросы к экзамену

1. База данных - совместно используемый набор логически связанных данных,

предназначенный для удовлетворения информационных потребностей организации.

Банк данных — это система, которая хранит и обрабатывает информацию. Он может включать в себя базы данных, системы управления базами данных, инструменты для анализа данных и другие компоненты.

Сходства между БД и Банком данных:

Оба представляют собой структуры данных, предназначенные для хранения

информации.

Они могут быть использованы для различных целей, таких как хранение

данных, управление информацией и анализ данных.

Они оба могут быть структурированы и организованы для облегчения

поиска и извлечения информации.

Различия между БД и Банком данных:

Банк данных обычно состоит из нескольких баз данных, в то время как база

данных может состоять из одной или нескольких таблиц.

Банк данных может иметь централизованное управление, в то время как

базы данных могут управляться разными организациями.

Банк данных может хранить различные типы данных, такие как текст,

изображения, аудио и видео, в то время как база данных обычно хранит один тип данных.

1. Система управления базами данных (СУБД) - программное обеспечение, которое позволяет создавать, изменять и управлять базами данных. СУБД выполняют следующие функции:

– Создание и изменение структуры баз данных;

– Управление доступом к данным;

– Обеспечение целостности данных;

– Оптимизация запросов к базе данных;

Требования к СУБД:

Надежность: СУБД должна обеспечивать сохранность данных в случае сбоев в

системе;

Масштабируемость: возможность масштабирования системы для работы с

большими объемами данных;

Производительность: СУБД должна обеспечивать высокую скорость обработки

запросов;

Безопасность: защита данных от несанкционированного доступа;

Гибкость: возможность адаптации СУБД к различным условиям работы;

Классификация СУБД:

Реляционные СУБД (MS SQL Server, Oracle, MySQL, PostgreSQL);

Иерархические СУБД (IBM DB2, Microsoft SQL Server, Sybase);

Сетевые СУБД (Oracle, Microsoft SQL Server);

Объектно-ориентированные СУБД (ObjectDB, Versant Object Database);

В NoSQL СУБД (MongoDB, Cassandra, Redis).

1. Архитектура СУБД состоит из следующих компонентов:

Ядро СУБД - отвечает за управление данными, выполнение запросов и обеспечение

безопасности данных;

Менеджер транзакций - обеспечивает атомарность, согласованность и изоляцию

транзакций;

Процессор запросов - обрабатывает запросы пользователей и генерирует

необходимый SQL-код для выполнения запросов;

Подсистема хранения данных - управляет физической структурой данных на диске;

Администратор баз данных - предоставляет инструменты для создания, изменения

и удаления баз данных.

1. Настольные СУБД предназначены для работы на отдельных рабочих станциях. Они обычно имеют простой интерфейс и предназначены для небольших объемов данных. Примеры настольных СУБД включают Microsoft Access, SQLite и Oracle Database Lite.

В архитектуре файл/сервер обработка запросов происходит следующим образом:

клиент отправляет запрос на сервер, который затем передает его базе данных. База данных

обрабатывает запрос и возвращает результаты клиенту.

Недостатком этой архитектуры является то, что она может быть неэффективной при большом количестве клиентов, так как каждый клиент должен отправлять запрос на сервер и ждать ответа. Кроме того, безопасность в этой архитектуре может быть проблемой, так как все данные хранятся на одном сервере, который может быть атакован.

1. Система данных — это набор данных, организованных таким образом, чтобы их можно было легко найти, извлечь и использовать. Системы данных могут включать в себя базы данных, файлы, таблицы и другие структуры, которые используются для хранения и управления данными.
2. Многомерная СУБД — это тип СУБД, который используется для хранения и

анализа многомерных данных. Многомерные данные представляют собой набор

измерений, каждое из которых имеет свои атрибуты.

Измерение — это одно из направлений многомерных данных, которое представляет

собой набор атрибутов, определяющих характеристики объекта в данном измерении.

Ячейка — это значение в многомерной модели данных, которое получается путем

пересечения значений по всем измерениям.

Срез — это подмножество многомерной модели данных, полученное путем

фиксации значений по некоторым измерениям.

1. Предметная область — это совокупность объектов и явлений, которые

описываются в базе данных.

Объект — это элемент предметной области, который имеет определенные свойства и характеристики.

Атрибут (элемент данных) — это характеристика или свойство объекта, которое

может быть измерено или описано.

Запись — это экземпляр объекта, то есть конкретный объект с определенными

значениями атрибутов.

Кортеж — это упорядоченный набор значений атрибутов, который представляет

собой описание объекта.

1. Проектирование базы данных — это процесс создания структуры базы данных, которая будет удовлетворять требованиям пользователей и обеспечивать эффективное хранение и обработку данных. Проектирование БД включает в себя определение структуры таблиц, установление связей между таблицами, определение необходимых индексов и ограничений.

К базам данных предъявляется ряд требований, таких как:

Нормализация данных позволяет избежать дублирования информации и

обеспечивает целостность данных.

Производительность базы данных должна быть высокой, чтобы обеспечить

быстрое выполнение запросов пользователей.

Безопасность данных обеспечивается путем шифрования данных, использования

механизмов аутентификации и авторизации пользователей, а также обеспечения физической безопасности серверов и баз данных.

1. Этапы жизненного цикла базы данных включают в себя следующие этапы:

– Анализ требований

– Проектирование базы данных

– Разработка структуры таблиц

– Создание схемы базы данных

– Загрузка данных

– Тестирование и отладка

– Развертывание базы данных

– Поддержка и обновление базы данных

1. Модель “сущность-связь” (ER-модель) предназначена для представления и

анализа структуры данных в базе данных. Она позволяет определить объекты (сущности) и связи между ними, а также свойства каждого объекта. На ER-диаграмме сущности изображаются в виде прямоугольников, а связи - в виде линий между ними.

Типы связей могут быть разными: один-к-одному, один-ко-многим, многие-ко-многим. Класс принадлежности сущности определяет, является ли данная сущность обязательной или необязательной.

1. Связь 1:1 означает, что каждой записи в одной таблице соответствует ровно

одна запись в другой таблице. Для преобразования такой связи в реляционные таблицу, необходимо создать две таблицы, и связать их через общий атрибут или набор атрибутов.

Связь 1:М означает, что одной записи в первой таблице соответствует несколько

записей во второй таблице. В этом случае, для преобразования в реляционную таблицу,

нужно создать таблицу с атрибутами первой связи и таблицу с атрибутами второй связи, а затем установить связь между ними через внешний ключ.

М:N означает много-ко-многим связь, когда одной записи в одной таблице может

соответствовать несколько записей в другой таблице, и наоборот. Для преобразования в реляционные таблицы, необходимо создать три таблицы: две с атрибутами связей и третью таблицу, которая будет связывать первые две через два внешних ключа.

1. Существует три основных типа связей между таблицами:

Один к одному (1:1) — это связь между двумя таблицами, при которой каждой

записи в одной таблице соответствует только одна запись в другой. Например, таблица “Сотрудники” и таблица “Контакты”, где каждый сотрудник имеет только один контакт.

Один ко многим (1:М) — это связь, при которой одной записи в одной таблице

соответствует много записей в другой. Например, в таблице “Сотрудники” может быть много записей о проектах, над которыми они работают.

Многие ко многим (М:N) — это связь, при которой много записей из одной таблицы связаны с многими записями из другой таблицы. Например, если у нас есть таблица “Проекты” и таблица “Сотрудники”, то каждый проект может иметь несколько сотрудников, и каждый сотрудник может работать над несколькими проектами. В этом случае необходимо создать третью таблицу для связи между ними.

1. Computer-Aided Design (CAD) - автоматизированная система, предназначенная для решения задач проектирования. Информационная согласованность в САПР — это точность и надежность данных, используемых в процессе проектирования. Она гарантирует, что вся информация является точной, полной и непротиворечивой на протяжении всего процесса проектирования, от первоначальной концепции до конечного продукта.
2. Независимость данных — это принцип, согласно которому данные должны быть отделены от процессов их обработки. Это означает, что данные должны храниться и обрабатываться таким образом, чтобы они могли быть использованы различными приложениями и системами без необходимости изменения их структуры или формата.
3. Концептуальный уровень — это уровень, на котором определяются основные понятия и объекты предметной области, а также связи между ними. На этом уровне определяются сущности, их атрибуты и отношения между ними, которые затем используются для построения логической модели данных.
4. Логический уровень - на этом уровне происходит преобразование концепций и объектов предметной области в структуры данных, которые могут быть использованы в конкретной СУБД. На логическом уровне определяются типы данных, ограничения целостности, триггеры и процедуры для работы с данными.
5. Физический уровень — это уровень, на котором происходит отображение логических структур данных в структуры, поддерживаемые конкретной СУБД. На этом уровне определяется размещение данных на носителях, методы доступа к данным, оптимизации запросов и т.д.
6. На уровне пользователя информация представляется в виде таблиц, форм, отчетов, графиков и других элементов интерфейса, которые позволяют пользователю работать с данными.
7. Иерархическая модель данных представляет собой структуру, в которой данные организованы в виде древовидной структуры. Пример: каждый сотрудник находится на определенном уровне иерархии и имеет подчиненных.
8. Сетевая модель данных позволяет представить данные в виде сети, где каждый узел может иметь несколько связей с другими узлами. Пример: где пользователи связаны друг с другом через отношения дружбы, подписки.
9. Реляционная модель данных состоит из отношений, которые представляют собой множества объектов. Каждый объект имеет набор атрибутов, которые определяют его свойства. Домен — это множество возможных значений атрибута. Атрибут — это

свойство объекта, которое может быть определено на домене.

1. Иерархическая и сетевая модели данных отличаются от реляционной модели тем, что они используют структуры данных, отличные от отношений. В реляционной модели данные связаны между собой через внешние ключи, которые указывают на соответствующие записи в других отношениях.
2. Кортеж — это строка в таблице, которая состоит из значений атрибутов. Кортеж представляет собой набор значений, соответствующих одной строке в таблице.

Отношение — это таблица в реляционной базе данных. Отношение состоит из набора атрибутов и множества кортежей, которые соответствуют этим атрибутам.

1. Отношения в реляционной модели обладают рядом свойств и ограничений:

—Атомарность: каждый атрибут должен иметь только одно значение для каждого кортежа.

—Отсутствие дубликатов: кортежи не должны содержать повторяющихся групп значений.

—Упорядоченность: кортежи должны быть упорядочены, но порядок атрибутов не имеет значения.

—Кортежам соответствуют значения атрибутов, но не наоборот.

—Значения атрибутов должны быть из одного домена.

—Отношения должны быть нормализованы, то есть разбиты на несколько отношений для устранения избыточности данных.

—Отношения не должны иметь циклических зависимостей.

–Каждая пара атрибутов должна быть связана первичным ключом.

–Внешние ключи должны быть уникальными и не должны дублироваться.

1. Поле — это столбец в таблице, который содержит значения атрибутов.

Запись — это строка в таблице, содержащая значения атрибутов для конкретного объекта.

Ключ — это атрибут или группа атрибутов, которые однозначно идентифицируют запись в таблице. Различают первичный и внешние ключи.

Первичный ключ — это атрибут или набор атрибутов, который однозначно определяет запись в таблице и не содержит нулевых значений.

Внешний ключ — это атрибут или набор атрибутов в одной таблице, которые соответствуют первичному ключу в другой таблице и используются для установления связи между таблицами.

1. Реляционная целостность — это набор правил и ограничений, которые гарантируют корректность и согласованность данных в реляционной базе данных.

Она включает в себя два основных аспекта:

Целостность таблиц: это правило гарантирует, что данные, хранящиеся в таблицах, соответствуют определенным правилам. Например, в таблице не может быть пустых значений для первичных ключей или нарушение ограничений уникальности.

Ссылочная целостность — это правило обеспечивает согласованность между связанными таблицами, гарантируя, что значения внешних ключей всегда соответствуют значениям первичных ключей в других таблицах. Это предотвращает появление несогласованных данных, таких как наличие записи о товаре, который не существует в таблице товаров.

1. Постреляционная модель расширяет возможности реляционной модели, позволяя работать с нетипичными данными, такими как массивы, повторяющиеся группы и сложные типы данных.

Она также снимает некоторые ограничения, такие как обязательное наличие первичного ключа и обязательное соблюдение ссылочной целостности. Однако постреляционные системы могут быть менее производительными и менее стандартизированными, чем реляционные системы.

1. Объектно-ориентированный подход к моделированию данных основывается на понятиях объектов, классов, методов, наследования, инкапсуляции и полиморфизма.

Объект — это основной элемент объектно-ориентированной модели, представляющий собой экземпляр класса и обладающий свойствами (атрибутами) и методами.

Класс — это шаблон, по которому создаются объекты. Он определяет структуру и поведение объектов.

Метод — это процедура или функция, связанная с классом и определяющая поведение объектов этого класса.

Наследование — это механизм, позволяющий создавать новые классы на основе существующих. Новый класс наследует свойства и методы родительского класса, и может добавлять свои собственные.

Инкапсуляция — это принцип, согласно которому данные и методы, связанные с объектом, скрываются от внешнего мира. Это обеспечивает защиту данных и упрощает изменение реализации.

Полиморфизм — это возможность использовать один и тот же метод для операций над объектами разных классов. Это позволяет писать более универсальный и гибкий код.

1. Объектно-реляционные модели данных представляют собой компромисс между реляционными и объектно-ориентированными моделями. Они позволяют хранить объекты в реляционных таблицах, но при этом сохраняют некоторые объектно-ориентированные свойства, такие как наследование и полиморфизм.
2. Многомерная модель данных предназначена для анализа и представления данных в виде многомерных кубов.

Измерение — это одна из сторон многомерного куба, представляющая собой категорию данных. Например, измерение “время” может иметь значения “год”, “месяц”, “день”.

Ячейка — это пересечение нескольких измерений, которое содержит значение факта. Например, ячейка “продажи за год” будет содержать количество проданных товаров за определенный год.

Факт — это числовое значение, которое хранится в ячейке. Факты обычно представляют собой количественные показатели, такие как суммы, количества, проценты и т.д.

1. Ключ отношения — это атрибут или набор атрибутов, которые однозначно идентифицируют кортеж в таблице.

Первичный ключ отношения — это особый тип ключа, который уникально идентифицирует каждый кортеж в отношении. Это означает, что для каждого кортежа существует только один набор значений первичного ключа. В реляционной модели данных первичный ключ обязателен для каждой таблицы.

1. Альтернативный ключ отношения — это ключ, который, как и первичный ключ, уникально идентифицирует кортеж, но не является первичным ключом. Альтернативные ключи могут быть полезны, если первичный ключ не подходит для конкретной задачи или если требуется дополнительная степень защиты от дублирования данных.
2. В реляционной модели данных отношения связаны между собой с помощью внешних ключей. Это гарантирует, что данные в связанных таблицах остаются согласованными и не содержат противоречий.
3. Основные принципы реляционной алгебры включают в себя:

— Атомарность: операции выполняются над одним отношением за раз.

— Совместимость типов: аргументы и результаты операций должны иметь совместимые типы.

— Ассоциативность: порядок выполнения операций не имеет значения, если аргументы совместимы.

— Коммутативность: порядок аргументов не имеет значения.

— Дистрибутивность: операция может быть распределена между другими операциями.

—Базовые теоретико-множественные операции включают в себя объединение, пересечение, разность и декартово произведение.

—Специальные теоретико-множественные операции включают выбор, проекцию, соединение и деление.

1. Операция проекции — это процесс выбора определенных атрибутов из отношения для создания нового отношения.

Операция селекции — это процесс выбора кортежей из отношения, удовлетворяющих определенным условиям.

Операция объединения — это процесс объединения двух или более отношений, имеющих одинаковые атрибуты, в одно отношение.

1. Операция пересечения: результатом операции пересечения двух множеств является множество, состоящее из всех элементов, которые присутствуют в обоих множествах.

Операция разности: результатом операции разности двух множеств является множество, состоящее из элементов первого множества, которые не входят во второе множество.

Операция соединения: результатом операции соединения двух или более множеств является единое множество, включающее все элементы исходных множеств.

1. Операция декартово произведение: результатом декартового произведения двух множеств есть множество всех возможных сочетаний этих множеств.

Основные объекты базы данных:

Таблица — основная структурная единица базы данных, состоящая из строк и столбцов.

Строка (запись) — набор данных, описывающих конкретный объект или ситуацию.

Столбец — атрибут, определяющий свойство объекта или ситуации.

Индекс — структура данных, которая ускоряет поиск и сортировку данных в таблице.

Первичный ключ — столбец или группа столбцов, однозначно идентифицирующих каждую строку таблицы.

Внешний ключ — столбец в одной таблице, который ссылается на первичный ключ другой таблицы.

Транзакция — группа операций, которые либо все успешно выполняются, либо все откатываются в случае ошибки.

Триггер — это программа, которая автоматически выполняется при возникновении определенного события в базе данных.

1. Создание базы данных включает в себя несколько этапов:

— Определение целей и задач базы данных;

— Определение структуры данных, необходимых для решения поставленных задач;

— Разработка схемы базы данных и моделей отношений между таблицами;

— Проектирование таблиц базы данных с учетом ограничений целостности и бизнес-правил;

— Создание таблиц базы данных с использованием соответствующего программного обеспечения

Правила создания таблиц баз данных:

— Каждая таблица должна иметь первичный ключ;

— Каждый столбец должен иметь уникальное имя;

— Значения в столбце не должны дублироваться;

— Столбцы, которые используются для связи между таблицами, должны иметь уникальные имена.

1. Строковые данные (String) - представляют собой последовательности символов, которые могут быть различной длины.

Числовые данные (Numeric) - включают в себя различные типы чисел, такие как целые числа (Integer), десятичные числа (Decimal) и числа с плавающей запятой (Float).

Дата и время (Date and Time) - используются для хранения дат и времени. Включают в себя типы данных Date, Time и DateTime.

Логические данные (Boolean) - могут принимать только два значения: True или False.

1. – Имя атрибута

— Тип данных

— Обязательность

— Значение по умолчанию

— Длина и точность

Взаимосвязь между таблицами базы данных может быть определена через внешние

ключи. Внешние ключи — это столбцы в одной таблице, которые соответствуют первичным ключам в другой таблице. Это позволяет установить связь между двумя таблицами и обеспечивает целостность данных.

1. Сущность — это объект реального мира, информацию о котором мы хотим хранить в базе данных.

Атрибут — это свойство сущности, которое имеет значение.

Идентификатор сущности — это атрибут или комбинация атрибутов, которые уникально идентифицируют каждую сущность в таблице.

1. “Один к одному” (1:1) - одна запись в одной таблице связана с одной записью в другой таблице;

– “Один ко многим” (1:N) - одна запись в первой таблице связана со многими записями во второй таблице, но каждая запись во второй таблице связана только с одной записью в первой таблице;

– “Многие ко многим” (N:M) - одна запись в каждой таблице связана с множеством

записей в другой таблице, и наоборот. В этом случае необходимо создать дополнительную таблицу для связывания двух исходных таблиц.

1. Нормализация таблиц — это процесс организации данных в таблицах таким образом, чтобы исключить дублирование, избыточность и обеспечить целостность данных. Цель нормализации - уменьшение избыточности данных, упрощение их обновления и обеспечение целостности.

Первая нормальная форма (1NF):

– Все атрибуты атомарные, то есть они не могут быть разделены на более мелкие части.

– Все не ключевые атрибуты зависят от первичного ключа.

Вторая нормальная форма (2NF):

– Таблица находится в 1NF.

– Ни один не ключевой атрибут не зависит от другого не ключевого атрибута, а зависит только от первичного ключа.

Третья нормальная форма (3NF):

-Таблица находится во 2NF.

-Ни один не ключевой атрибут не зависит функционально от любого другого не ключевого атрибута.

1. Преобразование логической модели в физическую включает в себя создание таблиц, индексов, триггеров и других объектов, которые необходимы для работы с данными на физическом уровне.
2. Функциональная зависимость — это отношение между атрибутами таблицы, при котором значение одного атрибута однозначно определяется значением другого атрибута.

Вторая нормальная форма (2НФ) требует, чтобы все атрибуты, не входящие в состав первичного ключа, были функционально зависимыми от ключа. То есть, для каждого атрибута должно быть выполнено условие: атрибут должен зависеть от ключа, но не от других атрибутов. Если это условие не выполняется, то таблица не находится во второй нормальной форме и требуется ее нормализация - разделение на две или более таблиц.