# 1. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА СУБД

## 1.1 Oracle Database

Oracle Database — это объектно-реляционная система управления базами данных (СУБД), разработанная корпорацией Oracle. Она является одной из самых мощных, надежных и масштабируемых СУБД в мире, широко используемой в корпоративной среде для хранения и обработки критически важных данных.

Oracle поддерживает множество функций, включая:

* многопользовательский доступ с высокой степенью безопасности;
* поддержка транзакций (ACID-совместимость);
* репликация, кластеризация и отказоустойчивость;
* расширенные возможности аналитики и обработки больших данных;
* поддержка PL/SQL (процедурное расширение SQL);
* интеграция с облачными сервисами (Oracle Cloud).

Плюсы Oracle Database:

* высокая производительность – Оптимизированные механизмы выполнения запросов, кэширование и параллельная обработка данных.
* масштабируемость – Поддержка распределенных баз данных и кластерных решений (RAC – Real Application Clusters).
* надежность и отказоустойчивость – Встроенные механизмы резервного копирования (RMAN), восстановления и репликации (Data Guard).
* безопасность – Расширенные функции шифрования, ролевой доступ (RBAC), аудит и защита от SQL-инъекций.
* поддержка больших объемов данных – Эффективная работа с Big Data, поддержка JSON, XML и in-memory технологий.
* кросс-платформенность – Работает на Windows, Linux, Unix, Solaris и в облаке.
* богатый функционал – Встроенные инструменты для аналитики, машинного обучения и IoT.

Минусы Oracle Database:

* высокая стоимость – Лицензирование Oracle очень дорогое, особенно для корпоративных решений.
* сложность администрирования – Требует высокой квалификации DBA, настройка и оптимизация могут быть нетривиальными.
* ресурсоемкость – Для работы требуется мощное серверное оборудование.
* закрытый код – В отличие от PostgreSQL или MySQL, Oracle – проприетарная СУБД, что ограничивает кастомизацию.
* жесткие условия лицензирования – Ограничения на использование в виртуализации и облаках, сложный аудит лицензий.

Oracle Database – это мощное решение для крупных предприятий, где критически важны надежность, безопасность и производительность. Однако для малого и среднего бизнеса или стартапов стоимость и сложность могут быть избыточными, и более легкие альтернативы (PostgreSQL, MySQL, Microsoft SQL Server) могут оказаться предпочтительнее.

## 1.2 Microsoft Access

Microsoft Access — это система управления базами данных (СУБД), разработанная компанией Microsoft. В отличие от серверных СУБД (таких как Oracle или SQL Server), Access представляет собой настольное решение, сочетающее в себе реляционную базу данных и графический интерфейс для разработки приложений.

Access входит в состав пакета Microsoft Office и часто используется для создания небольших локальных баз данных, отчетов и простых бизнес-приложений без необходимости программирования.

Плюсы Microsoft Access:

* простота использования – Интуитивно понятный интерфейс с конструкторами форм, запросов и отчетов (подходит для пользователей без навыков программирования).
* быстрое прототипирование – Позволяет быстро создавать небольшие базы данных и приложения.
* интеграция с Microsoft Office – Легко взаимодействует с Excel, Word и Outlook (импорт/экспорт данных, автоматизация через VBA).
* готовые шаблоны – Встроенные шаблоны баз данных (учет клиентов, склад, бухгалтерия и др.).
* поддержка SQL и VBA – Можно писать сложные запросы и добавлять логику через Visual Basic for Applications.
* низкая стоимость – Входит в пакет Microsoft Office, не требует дополнительных лицензий (в отличие от корпоративных СУБД).
* локальная работа – Не требует сервера, работает на одном компьютере.

Минусы Microsoft Access:

* ограниченная производительность – Замедляется при большом объеме данных (рекомендуется не более 1–2 ГБ).
* низкая масштабируемость – Не подходит для многопользовательских систем (более 10–15 пользователей).
* слабая безопасность – Нет встроенных механизмов шифрования, парольная защита ненадежна.
* риск повреждения данных – Файловая структура (.accdb) уязвима к сбоям при неправильном закрытии или сетевом доступе.
* ограниченная поддержка в корпоративной среде – Нет резервного копирования в реальном времени, репликации и кластеризации.
* зависимость от Windows – Не работает на Linux/macOS (только через эмуляторы).

Microsoft Access — отличный инструмент для небольших проектов, индивидуального использования и быстрого прототипирования. Однако для серьезных бизнес-приложений с большим количеством пользователей и критически важными данными лучше выбрать более надежные СУБД (SQL Server, PostgreSQL, MySQL).

## 1.3 Microsoft SQL Server

Microsoft SQL Server — это реляционная система управления базами данных (СУБД), разработанная корпорацией Microsoft. Она предназначена для хранения, обработки и защиты данных в корпоративных средах, поддерживая высокую производительность, масштабируемость и интеграцию с другими продуктами Microsoft.

SQL Server доступен в различных редакциях (Express, Standard, Enterprise, Developer) и работает на Windows и Linux (с 2016 года). Он широко используется в бизнес-аналитике, веб-приложениях и крупных корпоративных системах.

Плюсы Microsoft SQL Server:

* высокая производительность – Оптимизированный движок обработки запросов, поддержка in-memory OLTP и колоночного хранения (Columnstore).
* масштабируемость – Поддержка распределенных баз данных, шардинга и кластеризации (Always On Availability Groups).
* интеграция с экосистемой Microsoft – Глубокая совместимость с Azure, Power BI, Visual Studio и .NET.
* безопасность – Встроенное шифрование (TDE), Row-Level Security (RLS), динамическое маскирование данных.
* BI и аналитика – Инструменты SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS) и ETL-процессы (SQL Server Integration Services — SSIS).
* гибкость лицензирования – Бесплатная версия Express для небольших проектов и облачные решения (Azure SQL).
* поддержка Linux и Docker – С 2017 года работает не только на Windows, но и в Linux-средах.

Минусы Microsoft SQL Server:

* высокая стоимость коммерческих версий – Лицензии Enterprise Edition очень дорогие (особенно для крупных развертываний).
* зависимость от Windows (в некоторых компонентах) – Не все функции (например, SSIS) полностью поддерживаются в Linux.
* ограниченная кросс-платформенность – Несмотря на поддержку Linux, некоторые инструменты (SSMS) работают только в Windows.
* сложность администрирования – Требует квалифицированных администраторов (DBA) для настройки и оптимизации.
* закрытый исходный код – В отличие от PostgreSQL или MySQL, SQL Server — проприетарная СУБД.
* ограничения в бесплатной версии – SQL Server Express имеет ограничения по размеру БД (10 ГБ) и ресурсам.

Microsoft SQL Server — мощная СУБД для корпоративных решений, особенно в средах, где используется стек Microsoft (Windows Server, Azure, .NET). Хорошо показывает себя в работе с крупными бизнес-приложениями с высокой нагрузкой, системами бизнес-аналитики (BI) и отчетности, разработки под .NET. В случаях, если в проекте есть с требование open-source и высоконагруженные системы, где могут потребоваться NoSQL-решения, лучше рассмотреть другие варианты СУБД.

## 1.4 Выбор СУБД

Для реализации проекта выбор пал на Microsoft SQL Server, т.к. уже имеется опыт работы с ней, а также ее функционал удовлетворяет требованиям, предъявляемым к выполнению работы.

Плюсы Microsoft SQL Server: высокая производительность, интеграция с экосистемой Microsoft, безопасность и гибкость лицензирования – стали ключевыми при выборе этой СУБД.

В то же время минусы: высокая стоимость коммерческих версий, зависимость от Windows, ограниченная кросс-платформенность, закрытый исходный код – не оказывают серьезного влияния на процесс работы с СУБД.

# 2. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ЯП

## 2.1 C#

**C#** — это современный, объектно-ориентированный и типобезопасный язык программирования, разработанный компанией Microsoft в начале 2000-х годов в рамках ее стратегии .NET. Изначально он создавался как конкурент Java и унаследовал лучшие черты таких языков, как C++, Java и Delphi.

Ключевая особенность C# — он **компилируется** в промежуточный язык (IL), который затем выполняется в среде исполнения **CLR** (Common Language Runtime), являющейся частью платформы .NET. Это обеспечивает кроссплатформенность, безопасность памяти и управление ресурсами (сборка мусора).

С момента своего выхода C# активно развивается и добавляет новые функции с каждой версией, оставаясь одним из самых современных и востребованных языков.

Плюсы C#:

* Многофункциональность и универсальность (Unity + .NET) – идеален для создания классических приложений (WinForms, WPF), сервисов и игр под Windows.
* Статическая типизация и безопасность – компилятор проверяет типы данных на этапе компиляции, что позволяет находить множество ошибок до запуска программы. Это делает код более предсказуемым, надежным и удобным для рефакторинга.
* Высокая производительность – C# показывает производительность, сравнимую с C++ и Rust в многих сценариях, особенно с появлением .NET Core и постоянными оптимизациями.

Минусы C#:

* Историческая привязка к экосистеме Microsoft – хотя современный .NET кроссплатформенный, многие лучшие инструменты (как Visual Studio) и фреймворки (например, WPF) все еще сильнее заточены под Windows. Для полноценной работы под Linux/macOS иногда требуется больше усилий.
* Потребление памяти – из-за наличия виртуальной машины (CLR) и сборщика мусора (Garbage Collector) C# может потреблять больше памяти по сравнению с нативными языками вроде C++ или Rust. Хотя это решается тонкой настройкой, для встроенных систем с жесткими ограничениями по памяти это может быть критично.
* Меньшая гибкость по сравнению с языками с динамической типизацией – статическая типизация — это плюс для надежности, но иногда она может ощущаться как ограничение, особенно при работе с динамическими данными (например, JSON с неизвестной структурой). Хотя эта проблема смягчена с появлением типа dynamic и System.Text.Json.

C# — это зрелый, мощный и быстро развивающийся язык, который подходит для решения огромного спектра задач. Его главные преимущества — универсальность, производительность и поддержка со стороны Microsoft. Основные компромиссы — это историческая связь с Windows и некоторые ограничения, накладываемые управляемой средой исполнения.

## 2.2 Java

Java — это строго типизированный объектно-ориентированный язык программирования общего назначения, который занимает одно из первых мест в мире по популярности и востребованности.

Это достигается за счет уникальной архитектуры: код на Java компилируется не в машинные инструкции конкретного процессора, а в специальный промежуточный байт-код. Этот байт-код выполняется на Виртуальной машине Java (JVM), которая уже адаптирована под конкретную операционную систему (Windows, Linux, macOS и др.). Именно JVM обеспечивает кроссплатформенность и безопасность выполнения кода.

Плюсы Java:

* Кроссплатформенность (JVM) – это главное преимущество. Скомпилированный байт-код (.class файлы) можно запустить на любом устройстве, где установлена JVM. Это огромное преимущество для корпоративных сред, где часто используются разные ОС.
* Масштабируемость и производительность – благодаря современным JIT-компиляторам (Just-In-Time) в составе JVM, Java-приложения работают очень быстро. Горячий код оптимизируется и компилируется в машинный прямо во время выполнения, что приближает производительность к нативным языкам. Язык идеально подходит для создания больших, сложных и высоконагруженных систем (банки, крупные интернет-магазины, Big Data).
* Зрелость и стабильность – Java — проверенный временем язык. Его обратная совместимость тщательно соблюдается. Это критически важно для бизнеса, так как гарантирует, что написанный сегодня код будет работать и через 10 лет.

Минусы Java:

* Более многословный синтаксис (Verbose) – по сравнению с современными языками (Kotlin, Python, Go) код на Java требует написания большего количества строк для одних и тех же задач. Много шаблонного кода (boilerplate), например, для создания простого POJO-класса с геттерами и сеттерами. Это снижает скорость разработки.
* Платные лицензии для коммерческого использования в некоторых случаях – Хотя сам язык и большинство дистрибутивов OpenJDK бесплатны, Oracle с 2019 года ввела платные лицензии для долгосрочной поддержки (LTS) своих коммерческих сборок Oracle JDK для использования в production-среде. Это создало некоторую путаницу, однако существуют полностью бесплатные альтернативы (Amazon Corretto, Azul Zulu, Eclipse Temurin).
* Высокое потребление памяти – JVM и сама среда выполнения потребляют значительный объем оперативной памяти. Это делает Java не лучшим выбором для систем с крайне ограниченными ресурсами (например, микроконтроллеры).

Java — это беспроигрышный выбор для: крупного корпоративного ПО, высоконагруженных распределенных систем, разработки под Android: особенно для поддержки legacy-проектов или в тандеме с Kotlin, Big Data и обработки больших объемов данных, начинающих программистов, которые хотят получить глубокое понимание ООП и устроиться в крупную IT-компанию.

## 2.3 Kotlin

Kotlin — это современный, статически типизированный язык программирования, который работает на виртуальной машине Java (JVM) и полностью совместим с Java. Он был разработан компанией JetBrains в 2010 году и официально представлен как язык с открытым исходным кодом. В 2017 году Google объявил Kotlin предпочтительным языком для разработки под Android. Главная философия Kotlin — прагматичность. Он призван решать реальные практические задачи разработчиков, устраняя недостатки и многословность Java, при этом сохраняя всю мощь и совместимость с существующей экосистемой Java.

Плюсы Kotlin:

* Официальная поддержка от Google для Android – Kotlin — это язык первого выбора для Android-разработки. Новые инструменты, библиотеки и документация в первую очередь ориентированы на Kotlin.
* Лаконичность и выразительность – Kotlin позволяет достигать тех же целей, что и Java, но с значительно меньшим количеством кода. Меньше шаблонного кода (boilerplate) — выше читаемость и меньше места для ошибок.
* Мультиплатформенность (Kotlin Multiplatform - KMP)
* Позволяет делиться бизнес-логикой (кодом) между различными платформами: Android, iOS (через Native), Web (JS) и сервер (JVM). Это сокращает затраты на разработку и обеспечивает согласованность поведения на всех платформах.

Минусы Kotlin:

* Все еще развивающаяся мультиплатформенность – Kotlin Multiplatform — очень перспективная, но относительно новая технология. Она еще не достигла такой же стабильности и зрелости, как нативные инструменты от Apple или Flutter от Google. Некоторые iOS-библиотеки могут быть недоступны, а настройка окружения может быть сложной.
* Упрощенный синтаксис может быть обманчивым – лаконичность — это плюс, но иногда она может скрывать сложность операций, происходящих под капотом. Новичку может быть неочевидно, сколько всего на самом деле делает одна короткая строка кода, сгенерированная компилятором.
* Скорость компиляции – хотя команда Kotlin постоянно над этим работает, в некоторых крупных проектах компиляция Kotlin может быть немного медленнее, чем компиляция Java. Однако разрыв постоянно сокращается.

Kotlin — это превосходный выбор, если вы: Android-разработчик, бэкенд-разработчик, работаете в команде, которая хочет модернизировать старый Java-код, не отказываясь от него полностью, хотите делиться кодом между разными платформами (Android, iOS, веб) с помощью KMP, цените современный, элегантный и продуманный синтаксис, который делает программирование более приятным.

## 2.4 Выбор языка программирования

В связи с тем, что основной СУБД была выбрана MSSQL, выбор языка программирования пал на C#, т.к. он обеспечивает наилучшие совместимость и оптимизацию, уже имеется опыт работы с ним, а также существует множество библиотек, расширений и фреймворков, которые могут облегчить работу с ним. Его преимущества могут быть использованы в проекте, а недостатки никак не влияют на его использование в проекте.