

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение
высшего образования

**«Московский физико-технический институт
(национальный исследовательский университет)»
(МФТИ, Физтех)**

УТВЕРЖДАЮ

Ректор МФТИ

д-р физ.-мат. наук, профессор

Д. В. Ливанов

2023 г.



**Дополнительная общеобразовательная общеразвивающая программа
«SQL для начинающих»**

Москва 2023

1. Общая характеристика программы

1.1 В обществе всё большее значение приобретает умение человека использовать компьютер не на пользовательском уровне, а на уровне начинающего программиста. В обязательном школьном курсе информатики программирование представлено на уровне, достаточном для прохождения экзамена, но не предполагает овладение практическими навыками применения языка. Следствием этого - формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

Программа «SQL для начинающих» имеет техническую направленность, в её основу заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей предполагает детальное изучение алгоритмизации, реализацию межпредметных связей, организацию проектной и исследовательской деятельности обучающихся.

Данный курс по изучению SQL предназначен для тех, кто только начинает знакомиться с языком структурированных запросов. Он ориентирован на учащихся 8-11 классов общеобразовательных учреждений. В рамках курса слушатели узнают, как создавать и управлять базами данных, извлекать нужную информацию из таблиц, использовать функции агрегации и группировки, работать с транзакциями и блокировками, а также использовать оконные функции для более продвинутого анализа данных. Курс представлен в виде уроков с пошаговым объяснением материала и практическими заданиями для закрепления полученных знаний. По окончании курса слушатели смогут самостоятельно создавать запросы на SQL и работать с базами данных.

Цель программы - формирование познавательной активности обучающихся в области программирования, формирование навыков работы с основными операциями с базами данных, запросами, функциями и циклами, получение навыков самостоятельной разработки базы данных с использованием языка программирования реляционных баз данных SQL.

1.2. Категории слушателей, на обучение которых рассчитана программа дополнительного образования (далее – программа): учащиеся 8-11 классов.

1.3. Нормативный срок освоения программы – 144 академических часа.

1.4. Форма обучения: очная с применением дистанционных образовательных технологий.

Основной вид занятий – комбинированный, сочетающий в себе элементы теории и практики. Единицей учебного процесса является блок уроков (модуль). Каждый модуль охватывает отдельную информационную технологию или её часть. Внутри модулей разбивка по времени изучения производится педагогом самостоятельно, но с учётом рекомендованного

календарно-тематического плана. Темп изучения отдельных разделов блока определяется субъективными и объективными факторами и необходимостью повторения.

2. Планируемые результаты обучения

Личностные результаты

- формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам.

Метапредметные результаты

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить

логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;

- формирование и развитие компетентности в области использования информационно коммуникационных технологий.

Предметные результаты

Слушатель должен

знать:

- основы работы с реляционными базами данных;

- понятие SQL, для чего он используется, и какие типы данных и операторы доступны в SQL;

- ключевые понятия баз данных, таких как отношения, схемы, индексы, транзакции и права доступа к данным;

- основы реляционной алгебры и реляционных исчислений;

- этапы и процесс проектирования баз данных.

уметь:

- создавать таблицы, добавлять и изменять данные в них, а также как удалять таблицы и данные из них;

- использовать оператор SELECT для выборки данных из таблицы, понимать, как работает WHERE для фильтрации данных, и как использовать операторы ORDER BY, GROUP BY и HAVING для сортировки и агрегации данных;

- обновлять и удалять данные в таблице, используя операторы UPDATE и DELETE;

- создавать индексы для ускорения работы с таблицами и использовать их в запросах;

- использовать подзапросы для выборки данных из нескольких таблиц;

- объединять таблицы с помощью операторов JOIN, LEFT JOIN и RIGHT JOIN, и понимать разницу между ними;

- создавать хранимые процедуры и функции в SQL, для повторного использования кода и упрощения работы с базой данных;

- создавать представления, для упрощения доступа к данным в таблицах;

- использовать операторы COMMIT и ROLLBACK для управления транзакциями.

3. Структура программы Учебный план

№	Тема (модуль)	Кол-во часов	В том числе		
			Аудит. занятия	Самост. работа	Промежуточная аттестация
1	Модуль 1. Введение в SQL	36	18	14	4
1.1	Понятие SQL и реляционной модели баз данных. Основы работы с данными	3	2	1	
1.2	Жизненный цикл базы данных	3	2	1	
1.3	Реляционная алгебра и реляционное исчисление	3	2	1	
1.4	Проектирование баз данных.	4	2	2	
1.5	Установка MySQL и клиента управления базами данных	4	2	2	
1.6	Типы данных	3	2	1	
1.7	Создание базы данных в MySQL. Создание таблицы. Модификация и удаление таблицы. Работа с индексами	6	3	3	
1.8	CRUD операции	6	3	3	
	Промежуточная аттестация	4	-	-	4
2	Модуль 2. Запросы в SQL	36	18	14	4
2.1	Основные операторы SQL. SELECT и запросы на выборку данных	7	4	3	
2.2	Вспомогательные функции в SQL	6	4	3	
2.3	Группировка данных	4	2	2	
2.4	Соединение таблиц и виды связей	9	5	4	
2.5	Подзапросы	5	3	2	
	Промежуточная аттестация	4	-	-	4
3	Модуль 3. Процедуры, функции и циклы	36	18	14	4
3.1	Поиск текста	4	2	2	
3.2	Процедуры, функции и триггеры	13	7	6	
3.3	Циклы	4	2	2	
3.4	ViewиCase вSQL	6	4	2	
3.5	Импорт данных	5	3	2	
	Промежуточная аттестация	4	-	-	4
4	Модуль 4. Создание базы данных (индивидуальное задание для каждого слушателя)	36	18	14	8
4.1	Использование оконных функций	4	2	2	
4.2	Работа с транзакциями и блокировками	5	3	2	
4.3	Проектирование базы данных	3	2	1	
4.4	Создание базы данных и таблиц. Заполнение данными. Создание связей между таблицами	9	6	3	
4.5	Создание View-запросов и представление базы данных	7	5	2	
	Промежуточная аттестация	4	-	-	4
	Итоговый курсовой проект	8	-	4	4
ИТОГО		144	72	52	20

4. Содержание программы

4.1. Содержание программы по модулям

Модуль 1. Введение в SQL

Тема 1.1. Понятие SQL и реляционной модели баз данных. Основы работы с данными

Основные термины. Понятие баз данных и сферы использования баз данных. Модели данных: иерархическая, сетевая, реляционная. Понятие и типы систем управления базами данных (СУБД). Понятие языка реляционных баз данных SQL. История развития SQL. Роль SQL в современном мире – актуальность языка. Преимущества и недостатки языка.

Тема 1.2. Жизненный цикл базы данных

Понятие жизненного цикла базы данных. Этапы проектирования базы данных: концептуальное, логическое и физическое проектирование. Стадии жизненного цикла базы данных. Действия, выполняемые на каждом этапе жизненного цикла базы данных. Модели жизненного цикла.

Тема 1.3. Реляционная алгебра и реляционное исчисление

Терминология: отношение, атрибут, домен, кортеж, кардинальность, степень отношения. Свойства отношений. Понятие реляционной алгебры. Основные операции реляционной алгебры: выборка (selection), проекция (projection), декартово произведение (Cartesian product), объединение (union), разность множеств (set difference). Дополнительные операции реляционной алгебры: соединение (join), пересечение (intersection), деление (division). Понятие реляционного исчисления. Реляционного исчисления кортежей. Реляционного исчисления доменов.

Тема 1.4. Проектирование баз данных

Логические модели баз данных. Типы взаимосвязей в моделях баз данных: связь один к одному, связь один ко многим, связь многие ко многим. Виды реляционных ключей: первичный и вторичный ключ. Нормализация таблиц при проектировании баз данных: первая нормальная форма, вторая нормальная форма, третья нормальная форма. Этапы проектирования баз данных: системный анализ предметной области, инфологическое проектирование, выбор целевой СУБД, логическое проектирование, физическое проектирование.

Тема 1.5. Установка MySQL и клиента управления базами данных

Обзор реляционных систем управления базами данных (2022 г.). Обзор отечественных СУБД. Обоснование выбора MySQL. Редакции MySQL 8. Установка MySQL на Windows. Скачивание дистрибутива. Установка MySQL. Настройка конфигурации MySQL Server.

Тема 1.6. Типы данных

Типы данных: числовые типы, типы даты и времени, строка (символьные и байтовые) типы, пространственные типы и тип данных JSON. Значения типов данных по умолчанию. Требования к хранилищу типов данных.

Тема 1.7. Создание базы данных в MySQL. Создание таблицы. Модификация и удаление таблицы. Работа с индексами

Создание базы данных в MySQL. Оператор для создания таблиц CREATE TABLE. Оператор для удаления таблиц DROP TABLE. Изменение структуры таблицы ALTER TABLE. Операции, выполняемые над таблицами: ADD, ALTER, CHANGE, DROP, MODIFY, RENAME. Добавление данных в таблицу: LOAD DATA, INSERT.

Понятие индексирования. Преимущества и недостатки. Типы индексов. Создание индекса. Удаление индекса.

Выборка данных – SELECT.

Тема 1.8. CRUD операции

Понятие операций CRUD: CREATE, READ, UPDATE и DELETE. Операторы Insert (создание записей), Select (чтение записей), Update (редактирование записей), Delete (удаление записей).

Модуль 2. Запросы в SQL

Тема 2.1. Основные операторы SQL. SELECT и запросы на выборку данных

Арифметические операторы. Операторы сравнения. Полная выборка. Выбор конкретных столбцов. Логические операторы. Дополнительные элементы запроса: DISTINCT, COUNT, WHERE, (предикаты IN, OR, BETWEEN, LIKE) ORDERBY, MIN, MAX, AVG, NULL.

Тема 2.2. Вспомогательные функции в SQL

Функция для подсчета символов LENGTH. Функция вырезания и возвращения символов LEFT, RIGHT, SUBSTRING, MID, SUBSTRING_INDEX. Функция поиска и замены частей строки REPLACE. Поиск по строке POSITION, LOCATE и INSTR. Функции дополнения строки до нужного размера LPAD, RPAD. Функции REVERSE, REPEAT, TRIM, LTRIM, RTRIM, SPACE, LCASE (LOWER), UCASE (UPPER).

Тема 2.3. Группировка данных

Ограничение выборки LIMIT. Группировка результатов GROUP BY. Фильтрация результатов группировки HAVING. Сложение строк при выборке CONCAT, CONCAT_WS, GROUP_CONCAT. Объединение данных из нескольких таблиц в одну UNION. INTERSECT, EXCEPT.

Тема 2.4. Соединение таблиц и виды связей

Понятие соединения таблиц. Операция JOIN и ее виды: INNERJOIN, LEFTJOIN и RIGHTJOIN, FULLJOIN, CROSSJOIN. Соединение более двух таблиц. Многотабличные запросы.

Тема 2.5. Подзапросы

Понятие подзапроса. Операция EXISTS.

Подзапросы с несколькими строками и одним столбцом - ANY, SOME, ALL.

Модуль 3. Процедуры, функции и циклы

Тема 3.1. Поиск текста

Описание и лексика операторов LIKE, NOTLIKE. Расширенные регулярные выражения REGEXP и NOT REGEXP, RLIKE и NOT RLIKE. Использование логических операторов ИЛИ (OR), И (AND). Применение рассмотренных операторов для удаления данных таблицы.

Тема 3.2. Процедуры, функции и триггеры

Понятие процедур и функций. Синтаксис хранимых функции и процедур. Команды для работы с хранимыми подпрограммами. Комментарии к синтаксису хранимых функции и процедур. Инструкция IF NOT EXISTS, [NOT] DETERMINISTIC, COMMENT. Операторы BEGIN и END. Аргументы IN, OUT, INOUT, RETURNS. Характеристика SQLSECURITY. Инструкция DEFINER. Оператор RETURN. Вызов функций и процедур. Ограничения на хранимые функции. Понятие триггера. Синтаксис создания триггера. Удаление триггера

Тема 3.3. Циклы

Ознакомиться с основами циклов в SQL, включая их синтаксис, основные типы циклов и их применение в запросах базы данных. Изучить оператор LOOP и понять, как он работает.

Изучить оператор REPEAT и научиться создавать и использовать циклы типа REPEAT в SQL. Изучить оператор WHILE и понять, как он работает.

Тема 3.4. View и Case в SQL

Понятие представления (VIEW). Преимущества использования представлений. Ограничения представлений в MySQL. Создание представлений. Алгоритмы представлений. Обновляемые представления. Функции CASE, IF, IFNULL, COALESCE

Тема 3.5. Импорт данных

Основы импорта данных: понятие, причины и необходимость импорта данных в SQL. Форматы файлов данных: описание основных форматов файлов данных, которые можно импортировать в SQL, таких как CSV, TXT, Excel и другие. Импорт данных из текстовых файлов. Импорт данных из Excel. Импорт данных с помощью ETL-инструментов. Импорт данных из веб-страниц. Импорт данных из файлов XML и JSON. Импорт данных с помощью SQL-запросов.

Модуль 4. Создание базы данных

Тема 4.1. Использование оконных функций

Введение в оконные функции: обзор понятий, определений и основных принципов работы с оконными функциями. Функции агрегации и оконные функции: разница между ними и примеры использования. Работа с разделительными строками. Работа с пустыми значениями. Использование оконных функций для группировки данных по различным критериям. Использование оконных функций для ранжирования данных: сортировка, выборка и фильтрация данных с помощью оконных функций.

Тема 4.2. Работа с транзакциями и блокировками

Транзакции: определение и основные принципы работы.

Управление транзакциями в MySQL: команды START TRANSACTION, COMMIT и ROLLBACK. Уровни изоляции транзакций: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ и SERIALIZABLE. Блокировки в MySQL: определение и основные принципы работы. Уровни блокировок: SHARE, EXCLUSIVE, READ и WRITE. Различные типы блокировок в MySQL: блокировка таблицы, блокировка строки, блокировка чтения и блокировка записи. Конфликты блокировок: детектирование, предотвращение и разрешение.

Тема 4.3. Проектирование базы данных

Проектирование базы данных по одной из тем, предложенной преподавателем.

Тема 4.4. Создание базы данных и таблиц. Заполнение данными. Создание связей между таблицами

Создание баз данных слушателями в MySQL. Создание таблиц. Заполнение таблиц данными. Создание связей между таблицами

Тема 4.5. Создание View-запросов и представление базы данных

Создание View-запросов. Определение полей и фильтров. Создание представлений базы данных. Настройка прав доступа. Тестирование и оптимизация.

4.2. Примеры заданий для организации самостоятельной работы слушателей

Модуль 1. Введение в SQL

1. Создать таблицу «stock», которая имеет следующую структуру:

Поле	Тип
stock_id	INT PRIMARY KEY AUTO_INCREMENT
product	VARCHAR(50)
amount	DECIMAL(6,2)
price	INT

2. Добавить в базу следующую информацию:

Наименование	Кол-во	Цена, д.е.
Стеллаж	18	8960
Полка	34	2890
Зеркало	12	4082
Кровать	9	12470
Подушка	51	2415
Комод	11	8225

3. Вывести всю информацию о товарах на складе, для каждой позиции посчитать ее стоимость (произведение цены на количество). Вычисляемому столбцу дать имя total_product

4. Заменить информацию о позиции «кровать» на:

тахта	14	21460
-------	----	-------

5. Удалить информацию о товарах, стоимость которых меньше 100000 д.е.

6. Вывести на экран столбцы «наименование» и «стоимость».

Результат:

product	total_product
Стеллаж	161280
Кровать	112230
Подушка	123165

Тест:

1. Какие команды SQL используются для создания базы данных в MySQL?

- a) CREATE DATABASE <название базы данных>;
- b) USE <название базы данных>;
- c) CREATE TABLE <название таблицы>;

Ответ: a) CREATE DATABASE <название базы данных>;

2. Какие команды SQL используются для удаления таблицы?

- a) DELETE FROM <название таблицы>;
- b) TRUNCATE TABLE <название таблицы>;
- c) DROP TABLE <название таблицы>;

Ответ: c) DROP TABLE <название таблицы>;

3. Что такое индексы в базах данных?

- a) Индексы - это специальные структуры данных, которые позволяют быстро находить нужные записи в таблице;
- b) Индексы - это отдельные таблицы, которые хранят информацию о записях в главной таблице;
- c) Индексы - это особые алгоритмы, которые используются для сортировки данных в таблице.

Ответ: a) Индексы - это специальные структуры данных, которые позволяют быстро находить нужные записи в таблице;

4. Какие команды SQL используются для создания индексов?

- a) CREATE INDEX <название индекса> ON <название таблицы> (<название столбца>);

b) ALTER TABLE <название таблицы> ADD INDEX <название индекса> (<название столбца>);

c) ADD INDEX <название индекса> ON <название таблицы> (<название столбца>);

Ответ: а) CREATE INDEX <название индекса> ON <название таблицы> (<название столбца>);

5. Что такое SQL?

a) Система управления базами данных;

b) Язык программирования для создания веб-приложений;

c) Язык запросов для работы с реляционными базами данных.

Ответ: c) Язык запросов для работы с реляционными базами данных.

6. Что такое реляционная база данных?

a) База данных, которая использует реляционную модель данных для хранения данных;

b) База данных, которая хранит данные в формате таблиц;

c) База данных, которая позволяет использовать SQL для работы с данными.

Ответ: a) База данных, которая использует реляционную модель данных для хранения данных.

7. Что такое первичный ключ в реляционной базе данных?

a) Уникальный идентификатор для каждой записи в таблице;

b) Столбец, который ссылается на другую таблицу;

c) Столбец, который может содержать только уникальные значения.

Ответ: a) Уникальный идентификатор для каждой записи в таблице.

8. Что означает аббревиатура SQL?

a) Structured Query Language;

b) Structured Query Library;

c) Simple Query Language;

d) Simple Query Library.

Ответ: a) Structured Query Language.

9. Что такое ER-модель?

a) Модель реляционной базы данных;

b) Модель объектно-ориентированной базы данных;

c) Модель сущность-связь.

Ответ: c) Модель сущность-связь.

10. Какие операторы являются частью языка DDL?

a) CREATE, ALTER, DROP;

b) SELECT, FROM, WHERE;

c) INSERT, UPDATE, DELETE.

Ответ: а) CREATE, ALTER, DROP.

Модуль 2. Запросы в SQL

Допустим, у вас есть база данных о продажах товаров. Схема базы данных состоит из таблицы "Товары" (Goods) и таблицы "Продажи" (Sales).

Таблица "Товары" содержит следующие поля:

- id (уникальный идентификатор товара)
- name (название товара)
- price (цена товара)

Таблица "Продажи" содержит следующие поля:

- id (уникальный идентификатор продажи)
- goods_id (идентификатор товара)
- date (дата продажи)
- quantity (количество проданного товара)

Напишите SQL-запросы для выполнения следующих задач:

1. Выведите список всех товаров и их цены.

```
SELECT name, price  
FROM Goods;
```

2. Выведите список всех продаж с указанием названия товара, его цены, даты продажи и количества проданного товара.

```
SELECT Goods.name, Goods.price, Sales.date, Sales.quantity  
FROM Goods
```

```
JOIN Sales ON Goods.id = Sales.goods_id;
```

3. Выведите список всех продаж за последнюю неделю.

```
SELECT Goods.name, Goods.price, Sales.date, Sales.quantity  
FROM Goods
```

```
JOIN Sales ON Goods.id = Sales.goods_id
```

```
WHERE Sales.date >= DATE_SUB(NOW(), INTERVAL 1 WEEK);
```

4. Выведите суммарное количество проданных товаров по каждому товару.

```
SELECT Goods.name, SUM(Sales.quantity) as total_quantity  
FROM Goods
```

```
JOIN Sales ON Goods.id = Sales.goods_id
```

```
GROUP BY Goods.name;
```

5. Выведите суммарную выручку за каждый день.

```
SELECT Sales.date, SUM(Goods.price * Sales.quantity) as total_revenue  
FROM Goods
```

```
JOIN Sales ON Goods.id = Sales.goods_id
```

```
GROUP BY Sales.date;
```

6. Выведите список всех товаров, цена которых выше средней цены всех товаров.

```
SELECT name, price  
FROM Goods
```

```
WHERE price > (SELECT AVG(price) FROM Goods);
```


7. Выведите список всех товаров, которые не были проданы.

```
SELECT name, price  
FROM Goods  
LEFT JOIN Sales ON Goods.id = Sales.goods_id  
WHERE Sales.id IS NULL;
```

8. Выведите список всех продаж, где количество проданного товара было больше 5 единиц.

```
SELECT Goods.name, Goods.price, Sales.date, Sales.quantity  
FROM Goods  
JOIN Sales ON Goods.id = Sales.goods_id  
WHERE Sales.quantity > 5;
```

9. Выведите список всех продаж, где выручка была больше 1000 рублей.

```
SELECT Goods.name, Goods.price, Sales.date, Sales.quantity, (Goods.price  
* Sales.quantity) as revenue  
FROM Goods  
JOIN Sales ON Goods.id = Sales.goods_id  
WHERE (Goods.price * Sales.quantity) > 1000;
```

Модуль 3. Процедуры, функции и циклы

1. Создайте функцию на SQL, которая будет принимать на вход число и возвращать факториал этого числа. Используйте цикл WHILE.

```
CREATE FUNCTION factorial(n INT) RETURNS INT  
BEGIN  
    DECLARE result INT DEFAULT 1;  
    WHILE n > 1 DO  
        SET result = result * n;  
        SET n = n - 1;  
    END WHILE;  
    RETURN result;  
END;
```

2. Создайте хранимую процедуру на SQL, которая будет запрашивать у пользователя количество чисел, которые необходимо сгенерировать в последовательности Фибоначчи, а затем выводить эту последовательность на экран. Для генерации последовательности используйте цикл REPEAT. (Числа Фибоначчи (строка Фибоначчи) — числовая последовательность, первые два числа которой являются 0 и 1, а каждое последующее за ними число является суммой двух предыдущих.)

```
CREATE PROCEDURE generate_fibonacci_sequence(n INT)  
BEGIN  
    DECLARE i INT DEFAULT 1;  
    DECLARE prev INT DEFAULT 0;  
    DECLARE curr INT DEFAULT 1;  
    DECLARE next INT;
```



```

REPEAT
  SELECT curr;
  SET next = prev + curr;
  SET prev = curr;
  SET curr = next;
  SET i = i + 1;
UNTIL i > n END REPEAT;
END;

```

3. Создайте функцию на SQL, которая будет принимать на вход название города и возвращать количество клиентов из этого города в таблице "Клиенты". Для этого используйте цикл WHILE.

```

CREATE FUNCTION get_client_count(city_name VARCHAR(50))
RETURNS INT
BEGIN
  DECLARE client_count INT DEFAULT 0;
  DECLARE i INT DEFAULT 1;
  DECLARE total_rows INT;
  SELECT COUNT(*) INTO total_rows FROM Клиенты;
  WHILE i <= total_rows DO
    IF (SELECT Город FROM Клиенты LIMIT i-1,1) = city_name THEN
      SET client_count = client_count + 1;
    END IF;
    SET i = i + 1;
  END WHILE;
  RETURN client_count;
END;

```

Модуль 4. Создание базы данных

1. Дана таблица "sales" с полями: "date" (дата продажи), "product" (название продукта), "price" (цена продукта) и "region" (регион продажи). Необходимо найти общую сумму продаж по регионам за каждый месяц. Используйте оконную функцию.

Решение:

```

SELECT region,
  DATE_TRUNC('month', date) AS month,
  SUM(price) OVER (PARTITION BY region, DATE_TRUNC('month',
date)) AS sales_sum
FROM sales

```

2. Дана таблица "orders" с полями: "order_id" (идентификатор заказа), "product" (название продукта), "quantity" (количество продукта), "price" (цена продукта) и "customer_id" (идентификатор заказчика). Необходимо написать транзакцию, которая уменьшит количество продукта на 1 для заказа с

заданным идентификатором заказа, и увеличит общую цену заказа на цену продукта.

Решение:

START TRANSACTION;

UPDATE orders SET quantity = quantity - 1,

total_price = total_price + price

WHERE order_id = [ID];

COMMIT;

4.3. Список рекомендуемой литературы

Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К.Дж. Дейт. - М.: Символ-плюс, 2017. - 480 с.

Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли. - М.: Вильямс И.Д., 2017. - 1440 с.

Мартишин, С. А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2022. — 160 с. — (Высшее образование). - ISBN 978-5-8199-0517-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1815962>. - Режим доступа: по подписке.

Дюбуа П., MySQL. Сборник рецептов. – Пер. с англ. – СПб: Символ-Плюс, 2006. – 1056 с.

Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж.

Г89 SQL: полное руководство, 3-е изд. : Пер. с англ. - М.: ООО "И.Д. Вильямс", 2015. - 960 с. : ил. - Парал. тит. англ

5. Материально-технические условия реализации программы

Требования к помещению:

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

Информационное обеспечение:

- операционная система (желательно Windows);
- браузеры: Yandex Browser, Chrome, Chrome Mobile, Firefox, Opera, Safari, Mobile Safari, Edge, Python;
- VKЗвонки / Яндекс телемост;
- Браузер Chrome
- Microsoft Visual Studio 2010 или выше.

Требования к оборудованию

Таблица 4

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Аудитория с доступом в Интернет	Аудиторные занятия	Компьютер, Visual Studio, проектор, видеокамера, доступ в Интернет
Аудитория с доступом в Интернет	Самостоятельная работа	Персональные компьютеры/ноутбуки, Visual Studio, доступ в Интернет

6. Оценка качества освоения программ

Оценка качества освоения программы осуществляется в процессе промежуточной аттестации.

Формы и методы промежуточного контроля представлены в таблице 5.

Таблица 5

Наименование модулей	Основные показатели оценки	Формы и методы контроля и оценки
Модуль 1. Введение в SQL	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 2. Запросы в SQL	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 3. Процедуры, функции и циклы	Установленное количество выполненных заданий	Самостоятельное выполнение заданий
Модуль 4. Создание базы данных	Установленное количество выполненных заданий	Самостоятельное выполнение заданий

Примерные задания для проведения промежуточной аттестации по модулям

Модуль 1. Введение в SQL

Задание 1

1. Создайте базу данных с названием "my_database".
2. Создайте таблицу с названием "my_table", содержащую следующие столбцы:
 - id: целочисленный тип данных, первичный ключ, автоинкрементируемый;
 - name: строковый тип данных, максимальная длина 50 символов;
 - age: целочисленный тип данных;

– email: строковый тип данных, максимальная длина 100 символов, уникальный индекс;

3. Измените структуру таблицы "my_table", добавив столбец "phone" типа VARCHAR(20) после столбца "email".
4. Добавьте 10 записей в таблицу в каждый столбец.
5. Удалите столбец "age" из таблицы "my_table".
6. Удалите таблицу "my_table".
7. Создайте индекс для столбца "email".

Задание 2

1. Создайте базу данных с названием "my_database".
 2. Создайте таблицу "users" с полями "id", "name", "email", "age".
 3. Вставьте в таблицу "users" следующие записи:
 - (1, "Alice", "alice@example.com", 25)
 - (2, "Bob", "bob@example.com", 30)
 - (3, "Charlie", "charlie@example.com", 35)
 4. Обновите запись с id=2, установив значение поля "age" равным 32.
 5. Удалите запись с id=3.
 6. Выберите все записи из таблицы "users".
 7. Выберите записи из таблицы "users", у которых значение поля "age" больше или равно 30.
 8. Выберите записи из таблицы "users", у которых значение поля "name" содержит букву "a".
- Подсчитайте количество записей в таблице "users".

Модуль 2. Запросы в SQL

Задание 1

Для таблицы "employees" (с полями "id", "name", "department", "salary") напишите запросы SQL для:

1. Выборки уникальных значений из поля "department".
2. Подсчета количества сотрудников в каждом отделе.
3. Выборки записей сотрудников, у которых зарплата выше 50000.
4. Выборки записей сотрудников, работающих в отделе "IT" или "Sales".
5. Выборки записей сотрудников, работающих в отделе "IT" и получающих зарплату более 50000.
6. Упорядочивания записей по полю "salary" в порядке возрастания.
7. Подсчета минимальной, максимальной и средней зарплаты сотрудников.
8. Выборки записей сотрудников, которые не получают зарплату (т.е. значение поля "salary" равно NULL).

Задание 2

Дана таблица employees с полями id, name, email, phone, salary. Напишите запросы для выполнения следующих задач:

1. Выбрать все имена и количество символов в имени для всех сотрудников.
2. Выбрать имена и телефоны для всех сотрудников, где телефон начинается на "+1".
3. Выбрать имена и почты для всех сотрудников, где почта содержит "gmail.com".
4. Выбрать имена и первые 3 символа телефонов для всех сотрудников.
5. Выбрать имена и последние 4 символа почтовых адресов для всех сотрудников.
6. Выбрать имена и телефоны для всех сотрудников, где номер телефона содержит "555".
7. Выбрать имена и заменить все пробелы в email на символ "_".
8. Выбрать имена и номера телефонов для всех сотрудников, где номер телефона начинается на "555", и дополнить номер до 15 символов, добавив в начало нули.

Задание 3

Дана таблица "products" со следующими столбцами: id, name, category, price, quantity.

1. Напишите запрос, который выведет список всех уникальных категорий товаров.
2. Напишите запрос, который выведет список категорий товаров, количество товаров в каждой категории и среднюю цену товара в каждой категории. Отсортируйте результаты по возрастанию средней цены.
3. Напишите запрос, который выведет список товаров в категории "электроника", у которых цена выше 1000 и количество на складе больше 10.
4. Напишите запрос, который выведет список категорий товаров, в которых количество товаров больше 5.
5. Напишите запрос, который объединит таблицы "products" и "orders" по полю id, и выведет список товаров и количества заказов для каждого товара.
6. Напишите запрос, который найдет пересечение множеств товаров в категории "одежда" и товаров с ценой выше 500.
7. Напишите запрос, который найдет разность множеств товаров в категории "электроника" и товаров в категории "одежда".

Задание 4

У вас есть две таблицы в базе данных - "Клиенты" и "Заказы". Таблица "Клиенты" содержит информацию о клиентах, включая их ID, имя и контактную информацию. Таблица "Заказы" содержит информацию о заказах клиентов, включая их ID, ID клиента, дату заказа и стоимость заказа.

Используя INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, CROSS JOIN, напишите SQL-запросы для выполнения следующих задач:

1. Получить список всех заказов и соответствующих им клиентов.
2. Получить список клиентов, которые не сделали ни одного заказа.

3. Получить список клиентов и общую стоимость их заказов.
4. Получить список всех клиентов и количества сделанных ими заказов.
5. Получить список клиентов, которые сделали более одного заказа.
6. Получить список клиентов и среднюю стоимость их заказов.
7. Получить список всех клиентов и суммарную стоимость их заказов за определенный период времени.

Задание 5

В базе данных есть две таблицы - "Студенты" и "Оценки". Таблица "Студенты" содержит информацию о студентах, включая их ID, имя и фамилию. Таблица "Оценки" содержит информацию об оценках студентов по различным предметам, включая их ID, ID студента, название предмета и оценку.

Используя подзапросы, напишите SQL-запросы для выполнения следующих задач:

1. Получить список студентов, у которых есть оценка выше 4.0 по предмету "Математика".
2. Получить список студентов, которые получили оценку выше 3.0 по любому предмету.
3. Получить список студентов, которые получили оценки выше 4.0 по всем предметам.
4. Получить список студентов, у которых все оценки выше 3.0.
5. Получить список студентов, которые не получили оценок ниже 3.0.

Модуль 3. Процедуры, функции и циклы

Задание 1

Найти все фильмы в таблице "Фильмы", название которых содержит слово "драма" и год выпуска равен 2020, либо название содержит слово "комедия" и год выпуска равен 2021.

Задание 2

У нас есть база данных, которая содержит таблицу "Клиенты" с информацией о клиентах и таблицу "Заказы" с информацией о заказах клиентов. Требуется создать следующие процедуры, функции и триггеры:

1. Процедуру для добавления нового клиента в таблицу "Клиенты". Процедура должна принимать параметры: имя, фамилию, адрес, телефон, email.
2. Функцию для получения общей стоимости заказов конкретного клиента. Функция должна принимать ID клиента в качестве параметра и возвращать общую стоимость его заказов.
3. Триггер, который при добавлении нового заказа в таблицу "Заказы" автоматически увеличивает счетчик заказов для соответствующего клиента в таблице "Клиенты".

4. Процедуру для удаления клиента из таблицы "Клиенты" и всех его заказов из таблицы "Заказы".

Задание 3

Создайте хранимую процедуру на SQL, которая будет выводить на экран таблицу умножения от 1 до 10 с использованием цикла LOOP.

Также создайте функцию на SQL, которая будет принимать на вход число и с помощью цикла WHILE будет выводить все числа от 1 до введенного числа.

Создайте хранимую процедуру на SQL, которая будет запрашивать у пользователя числа до тех пор, пока он не введет отрицательное число, и выводить на экран сумму всех введенных положительных чисел. Для этого используйте цикл REPEAT.

Задание 4

Необходимо создать View на SQL, которая будет выводить информацию о продуктах из таблицы "Продукты". Выводимая информация должна содержать название продукта, его цену и категорию продукта. Однако, для удобства пользователей, необходимо переименовать некоторые категории продуктов. Категории "Овощи" и "Фрукты" должны быть объединены в одну категорию "Фрукты и овощи", а категория "Молочные продукты" должна быть переименована в "Молочка". Для этого используйте оператор CASE.

Кроме того, создайте запрос к этой View, который будет выводить информацию о продуктах с ценой выше 100 рублей и сортировать их по названию продукта в алфавитном порядке.

Напишите SQL код для создания View и запроса.

Задание 5

Напишите запрос на:

1. Импорт данных из CSV-файла
2. Импорт данных из другой таблицы в базе данных
3. Импорт данных из другой базы данных
4. Импорт данных из текстового файла
5. Импорт данных из Excel-файла

Модуль 4. Создание базы данных

Примеры тем для разработки баз данных слушателями с названиями основных таблиц указанных баз данных:

1. Студенческий портал: студенты, преподаватели, курсы, оценки, расписание занятий.
2. Музыкальная коллекция: исполнители, альбомы, треки, жанры, пользователи и их плейлисты.
3. Учет товаров на складе: товары, склады, поставщики, заказы, отчеты о продажах и остатках товаров.

4. Система управления проектами: проекты, задачи, команды, сроки, бюджеты, отчеты о выполнении работ.

5. Система управления гостиницей: номера, бронирование, гости, услуги, платежи.

6. Интернет-магазин: товары, категории, заказы, покупатели, отзывы, скидки.

7. Заявки на техническое обслуживание: оборудование, заявки, техники, отчеты о выполнении работ.

8. Система управления библиотекой: книги, читатели, заказы, выдача книг, возврат, штрафы.

9. Система учета финансов: доходы, расходы, счета, операции, отчеты о финансовом состоянии.

10. Система управления рестораном: таблицы для заказов, меню, сотрудников, поставщиков, склада.

11. Учет лекарственных препаратов: таблицы для списка лекарств, информации об их составе и противопоказаниях, поставщиках, заказов, рецептов.

12. Система учета зарплаты сотрудников: таблицы для списка сотрудников, их должностей, отделов, рабочих часов, вычетов и налогов.

13. Система учета домашних животных: таблицы для списка питомцев, их характеристик, медицинской карты, посещений ветеринара.

Система учета автомобилей и их владельцев: таблицы для списка автомобилей, владельцев, технического обслуживания, страховки, нарушений ПДД.

7. Составители программы

Малеев Алексей Викторович, директор Центра развития ИТ-образования МФТИ

Созыкин Андрей Владимирович, к. т. н., заместитель директора Центра развития ИТ-образования МФТИ

Байчорова Карина Тагировна, администратор электронных образовательных систем Института Открытого образования Северо-Кавказской государственной академии

Якупов Азат Шавкатович, старший преподаватель кафедры программной инженерии Казанского (Приволжского) федерального университета

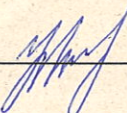
Мартемьянов Роман Юрьевич, заместитель директора Центра развития ИТ-образования МФТИ

Даревский Сергей Григорьевич, руководитель группы Центра развития ИТ-образования МФТИ

Сырцова Елена Леонидовна, PhD, к. пед. н., доцент, руководитель проектов Центра развития ИТ-образования МФТИ

Токмакова Ольга Викторовна, PhD, к. пед. н., доцент, специалист по учебно-методической работе Центра развития ИТ-образования МФТИ

Согласовано
Ведущий специалист отдела
сопровождения образовательных
программ


_____ Ж.И. Зубцова

Согласовано
Директор ЦРИТО


_____ А.В. Малеев

Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»

ВЫПИСКА ИЗ ПРОТОКОЛА № 8
заседания учебно-методического совета от 29 марта 2023 года.

ПОВЕСТКА:

Рассмотрение дополнительных общеобразовательных и профессиональных программ.

Проректор по учебной работе А. А. Воронов.

СЛУШАЛИ: заместителя директора Центра дополнительного, дополнительного профессионального и онлайн-образования «ПУСК» Рыбакову А.И. о представлении дополнительных общеобразовательных и профессиональных программ. (Центр «ПУСК» МФТИ).

ПОСТАНОВИЛИ:

Рекомендовать к утверждению в установленном порядке дополнительную общеобразовательную программу «SQL для начинающих».

Решение принято единогласно.

Форма проведения заседания: заочная.

Председатель УМС МФТИ

Ученый секретарь УМС МФТИ



А.А. Воронов

М.В. Берзникова