**Технологическая карта № 3**

**Наименование:** Структурированный язык запросов SQL. Команда SELECT

**1. Цель работы:**

1.1. Изучить структуру оператора SQL.

1.2. Выработать навыки построения запросов на выборку.

**2. Подготовка к работе:**

3.1. Изучить предложенную литературу.

3.2. Подготовить бланк отчёта.

**3. Литература:**

2.1. Фуфаев Э.В., Фуфаев Д.Э**.** Разработка и эксплуатация удаленных баз данных: учебник для студ. сред. проф. образования / под ред. Э.В. Фуфаев, Д.Э. Фуфаев. – Москва: Издательский центр «Академия», 2008. – 256 с.

2.2. Кунтаева Г.Х. Конспект лекций по дисциплине: «Разработка и эксплуатация удаленных баз данных».

**4. Перечень необходимого оборудования:**

4.1. Персональный IBM PC.

**5. Задание на работу:**

5.1. Вывести наименования предметов, с количеством часов равным 40 или 50.

5.2. Вывести наименования предметов, у которых количество часов не равно сумме лекционных и практических.

5.3. Вывести номера студентов и номера предметов, по которым получали оценки 3 или 4.

5.4. Выведите из таблицы predmet все наименования предметов с количеством часов, принадлежащих интервалу 100 - 140

5.5. Из таблицы Успеваемость выведите код предмета, табельный номер преподавателя, номер студента, который получал двойку.

5.6. Выведите на экран количество студентов в ‘101’ группе.

5.7. Создайте запрос, который выводит на экран код кафедры ‘математика’.

5.8. Из таблицы Преподаватель выведите на экран фамилии и табельные номера преподавателей с кафедры ‘02’.

5.9. Из таблицы Изучение выведите на экран номер группы и табельный номер преподавателя, который ведёт предмет с количеством часов (chas) более 50.

**6. Порядок проведения лабораторной работы:**

6.1. Получить допуск к работе.

6.2. Выполнить задания в соответствие с приложением.

6.3. Ответить на контрольные вопросы.

6.4. Оформление отчета.

**7. Содержание отчета:**

7.1. Наименование и цель работы.

7.2. Запись результата выполненных заданий в отчёт.

7.3. Ответы контрольные вопросы.

7.4. Выводы о проделанной работе.

**8. Контрольные вопросы**:

8.1. С помощью какого оператора прописывается условие?

* 1. Как связываются две таблицы?
  2. Что такое DISTINCT?
  3. Какой оператор позволяет вывести сразу две оценки?

**Приложение**

SQL символизирует собой Структурированный Язык Запросов. Запросы - вероятно наиболее часто используемый аспект SQL. Запрос - команда которую вы даете вашей программе базы данных, и которая сообщает ей чтобы она вывела определенную информацию из таблиц в память. Эта информация обычно посылается непосредственно на экран компьютера.

*1. Формат команды SELECT*

Общий вид команды:

SELECT \* | { [ DISTINCT | ALL] < value expression >.,..}

FROM { < table name > [ < alias > ] }.,..

[ WHERE ]

[ GROUP BY { | }.,..]

[ HAVING ]

[ ORDERBY { | }.,..]

В самой простой форме, команда SELECT просто инструктирует базу данных чтобы извлечь информацию из таблицы. Например, вы могли бы вывести таблицу ‘Студент’ напечатав следующее:

SELECT ns,fio FROM student

Эта команда просто выводит все данные из таблицы.

***SELECT***  Ключевое слово, которое сообщает базе данных что эта

команда - запрос. Все запросы начинаются этим словом

***ns, fio*** Это - список столбцов из таблицы которые выбираются

запросом. Любые столбцы, не перечисленные здесь, не

будут включены в вывод команды. Это, конечно, не значит

что они будут удалены или их информация будет стерта

из таблиц, потому что запрос не воздействует на

информацию в таблицах; он только показывает данные/

**FROM** - ключевое слово, подобно SELECT, которое должно

быть представлено в каждом запросе. Оно сопровожда-

ется пробелом и затем именем таблицы используемой

в качестве источника информации.

В данном случае - это таблица student

Если вы хотите видеть каждый столбец таблицы, имеется необязательное сокращение которое вы можете использовать. Звездочка (\*) может применяться для вывода полного списка столбцов следующим образом:

SELECT \* FROM student;

Это приведет к тому же результату что и наша предыдущая команда.

***DISTINCT (ОТЛИЧИЕ***) - аргумент который обеспечивает Вас способом устранять двойные значения из вашего предложения SELECT. Предположим, что вы хотите знать какие студенты получали оценки.

SELECT DISTINCT ns

FROM uspevaemost

Другими словами, DISTINCT следит за тем, какие значения были ранее, так что бы они не были продублированы в списке. Это - полезный способ избежать избыточности данных. DISTINCT опускает строки где все выбранные поля идентичны. Строки, в которых некоторые значения одинаковы, а некоторые различны - будут сохранены.

***WHERE*** - предложение команды SELECT, которое позволяет вам устанавливать предикаты, условие которых может быть или верным, или неверным для любой строки таблицы. Команда извлекает только те строки из таблицы для которой такое утверждение верно. Например, предположим вы хотите видеть номера групп с количеством студентов более 20 человек.

SELECT ng

FROM gruppa

WHERE kol > 20

Когда предложение WHERE представлено, программа базы данных просматривает всю таблицу по одной строке и исследует каждую строку чтобы определить, верно ли утверждение.

*2. Реляционные операторы*

Реляционный оператор - математический символ, который указывает на определенный тип сравнения между двумя значениями. Реляционные операторы которыми располагает SQL:

= Равный к

**>** Больше чем

< Меньше чем

>= Больше чем или равно

<= Меньше чем или равно

<> Не равно

Основные Булевы операторы также распознаются в SQL. Стандартные Булевы операторы распознаваемыми в SQL являются:

**AND, OR,** и **NOT**

*3. Использование специальных операторов в условиях*.

SQL использует специальные операторы IN, BETWEEN, LIKE, и IS NULL. Мы рассмотрим как их использовать и как реляционные операторы позволяют создавать более сложные и мощные предикаты. Обсуждение оператора IS NULL будет включать отсутствие данных и значение NULL, которое указывает на то: что данные отсутствуют. Вы также узнаете о разновидностях использования оператора NOT применяющегося с этими операторами.

*Оператор IN*

Оператор IN определяет набор значений в которое данное значение может или не может быть включено.

*Оператор BETWEEN*

Оператор BETWEEN похож на оператор IN. В отличии от определения по номерам из набора, как это делает IN, BETWEEN определяет диапазон, значения которого должны уменьшаться что делает предикат верным. Вы должны ввести ключевое слово BETWEEN с начальным значением, ключевое AND и конечное значение. В отличие от IN, BETWEEN чувствителен к по-

рядку, и первое значение в предложении должно быть первым по алфавитному или числовому порядку.