

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните задания лабораторной работы.

Результаты работы, отчет, предоставить преподавателю на e-mail: xvsviv@rambler.ru

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- название работы, постановку цели, вывод;
- скриншоты поэтапного выполнения лабораторной работы.

При возникновении вопросов по приведенному материалу обращаться по номеру телефона: 0721389311 (WhatsApp).

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лабораторная работа. «Многотабличные запросы. Подзапросы и выражения с запросами. Выполнение операций над данными с использованием операторов языка SQL»

Цель: вырабатывать навыки создания различных запросов с помощью MySQL, научиться создавать схему данных, получить представление о подзапросах и представлениях.

Краткие теоретические сведения

Запросы являются фундаментом SQL. Многие разработчики используют SQL исключительно в качестве инструмента для создания запросов. Поэтому важнейшей инструкцией является инструкция SELECT, которая используется для построения SQL-запросов.

Синтаксис оператора *SELECT* имеет следующий вид:

SELECT [ALL | DISTINCT] «писок полей»/)*

FROM <Список таблиц>

[*WHERE* <Предикат-условие выборки или соединения>]

[*GROUP BY* <Список полей результата>]

[*HAVING* <Предикат-условие для группы>]

[*ORDER BY* <Список полей, по которым упорядочить вывод>]

Здесь ключевое слово *ALL* означает, что в результирующий набор строк включаются все строки, удовлетворяющие условиям запроса. Значит, в результирующий набор могут попасть одинаковые строки. Ключевое слово *DISTINCT* означает, что в результирующий набор включаются только различные строки, то есть дубликаты строк результата не включаются в набор.

Символ *. (звездочка) означает, что в результирующий набор включаются все столбцы из исходных таблиц запроса.

В разделе *FROM* задается перечень исходных отношений (таблиц) запроса.

В разделе *WHERE* задаются условия отбора строк результата или условия соединения кортежей исходных таблиц, подобно операции условного соединения в реляционной алгебре.

В разделе *GROUP BY* задается список полей группировки и используется когда в предложении *SELECT*, использованы статистические функции (*sum()*, *Avg()*, *count()*, *min()*, *max()*).

В разделе *HAVING* задаются предикаты-условия, накладываемые на каждую группу.

В части *ORDER BY* задается список полей упорядочения результата, то есть список полей, который определяет порядок сортировки в результирующем отношении. Например, если первым полем списка будет указана Фамилия, а вторым Номер группы, то в результирующем отношении сначала будут собраны в алфавитном порядке студенты, и если найдутся однофамильцы, то они будут расположены в порядке возрастания номеров групп.

В выражении условий раздела *WHERE* могут быть использованы следующие предикаты:

Предикаты сравнения { =, <>, >, <, >=, <= }, которые имеют традиционный смысл.

Предикат *Between A and B* — принимает значения между A и B. Предикат истинен, когда сравниваемое значение попадает в заданный диапазон, включая границы диапазона. Одновременно в стандарте задан и противоположный предикат *Not Between A and B*, который истинен тогда, когда сравниваемое значение не попадает в заданный интервал, включая его границы.

Предикат вхождения в множество *IN* (множество) истинен тогда, когда сравниваемое значение входит в множество заданных значений. При этом множество значений может быть задано простым перечислением или встроенным подзапросом. Одновременно существует противоположный предикат *NOT IN* (множество), который истинен тогда, когда сравниваемое значение не входит в заданное множество.

Предикаты сравнения с образцом *LIKE* и *NOT LIKE*. Предикат *LIKE* требует задания шаблона, с которым сравнивается заданное значение, предикат истинен, если сравниваемое значение соответствует шаблону, и ложен в противном случае. Предикат *NOT LIKE* имеет противоположный смысл.

По стандарту в шаблон могут быть включены специальные символы:

- символ подчеркивания () - для обозначения любого одиночного символа;
- символ процента (%) - для обозначения любой произвольной последовательности символов;
- остальные символы, заданные в шаблоне, обозначают самих себя.

Предикат сравнения с неопределенным значением *IS NULL*. Понятие неопределенного значения было внесено в концепции баз данных позднее. Неопределенное значение интерпретируется в реляционной модели как

значение, неизвестное на данный момент времени. Это значение при появлении дополнительной информации в любой момент времени может быть заменено на некоторое конкретное значение. При сравнении неопределенных значений не действуют стандартные правила сравнения: одно неопределенное значение никогда не считается равным другому неопределенному значению. Для выявления равенства значения некоторого атрибута неопределенному применяют специальные стандартные предикаты:

<имя атрибута> IS NULL и *<имя атрибута> IS NOT NULL*.

Если в данном кортеже (в данной строке) указанный атрибут имеет неопределенное значение, то предикат *IS NULL* принимает значение «Истина» (*TRUE*), а предикат *IS NOT NULL* — «Ложь» (*FALSE*), в противном случае предикат *IS NULL* принимает значение «Ложь», а предикат *IS NOT NULL* принимает значение «Истина».

Задание 1.

Контрольные вопросы:

1. Запишите инструкцию создания базы данных, создания таблицы, добавления или удаления столбца таблицы.
2. Какие еще действия с таблицей относятся к модификации таблицы. Запишите их инструкции.