MySQL

Занятие 2

План занятия

- in, between, like, is null.
- Арифметические операции
- логические операции.
- Аггрегирование и групповые функции.
- Пустые значения NULL

IN – NOT IN

- IN (равен любому из списка)
- NOT IN (не равен ни одному из списка)

IN – NOT IN

Получить из таблицы EXAM_MARKS сведения о студентах, имеющих экзаменационные оценки только 4 и 5.

```
SELECT *
FROM EXAM_MARKS
WHERE MARK IN (4, 5);
```

Получить сведения о студентах, не имеющих ни одной экзаменационной оценки, равной 4 и 5.

```
SELECT *
FROM EXAM_MARKS
WHERE MARK NOT IN (4, 5);
```

BETWEEN

Вывести записи о предметах обучения, количество часов , отводимых на которые, лежит в пределах между 30 и 40:

```
SELECT *
FROM SUBJECT
WHERE HOUR BETWEEN 70 AND 140;
```

Задание 1

- Напишите запрос, выполняющий вывод находящихся в та блице EXAM_MARKS номеров предметов обучения, экзамены по которым сдавались между 10 и 20 января 1999 года.

LIKE

- Применим только к символьным полям типа CHAR или VARCHAR
- Проверяет, входит ли подстрока в символьную строку
- Символ “_” определяет возможность наличия одного любого символа
- Символ “%” допускает присутствие последовательности любых символов произвольной длины

LIKE

Написать запрос, выбирающий из таблицы STUDENT сведения о студентах, у которых фамилии начинаются на букву “Р”.

```
SELECT *
FROM STUDENT
WHERE SURNAME LIKE 'Р%';
```

LIKE

- В случае необходимости использовать в образце “_” и “%” их нужно «защитить»

LIKE ‘__P ’ESCAPE ‘\’ ← Указываем символ, которым защищаем

RLIKE

- Аналогичен LIKE, но использует регулярные выражения

```
SELECT *
FROM lecturer
WHERE SURNAME RLIKE '^A.*';
```

Задание 2

- Напишите запрос, выбирающий сведения о студентах, у которых имена начинаются на буквы ‘И’ или ‘С’.

IS NULL и IS NOT NULL

~~=NULL~~

- IS NULL (является пустым)
- IS NOT NULL (является не пустым).

IS NULL и IS NOT NULL

Выведем информацию об экзаменационных оценках студентов, где сама оценка неизвестна

```
SELECT *
FROM exam_marks
WHERE MARK IS NULL;
```

- ?сколько таких строк?

Арифметические операции

- Унарный **«-»**
- Бинарные **«+»**, **«–»**, **«*»** и **«/»**

```
SELECT SURNAME, NAME, STIPEND, -STIPEND*3/2
FROM student
WHERE KURS=4 AND STIPEND>0;
```

Задание 3

- Вывести фамилии, имена студентов и величину получаемых ими стипендий, при этом значения стипендий должны быть увеличены в 100 раз.

Конкатинация строк

- CONCAT(str1, str2, …)

```
SELECT CONCAT(SURNAME,'_',NAME), STIPEND
FROM STUDENT
WHERE KURS = 4 AND STIPEND > 0;
```

Символьные функции преобразования букв различных слов в строке

- LOWER – перевод в строчные символы (нижний регистр)
- UPPER – перевод в прописные символы (верхний регистр)
- INITCAP – перевод первой буквы каждого слова строки в заглавную

```
SELECT LOWER(SURNAME), UPPER(NAME)
FROM STUDENT
WHERE KURS = 4 AND STIPEND > 0;
```

Символьные функции преобразования букв

- LPAD – дополнение строки слева

LPAD (*<строка >, <длина> , <подстрока>)*

- *<строка> дополняется слева указанной в <подстроке>* последовательностью символов до указанной *<длины> (возможно, с* повторением последовательности);
- если *<длина> меньше длины <строки>, то исходная <строка>* усекается справа до заданной *<длины >.*
- *Аналогично RPAD – дополнение справа*

```
SELECT LPAD(SURNAME, 7, '?')
FROM lecturer
WHERE SURNAME RLIKE '^A.*';
```

LTRIM, RTRIM и TRIM

- LTRIM – удаление пробелов слева
- RTRIM – удаление пробелов справа
- TRIM – с обеих сторон

```
SELECT LTRIM('        barbar');
```

Подстрока

- SUBSTR – выделение подстроки

SUBSTR (*<строка>, <начало> [,<количество >])*

- из *<строки> выбирается заданное <количество> символов , начиная с* указанной позиции в строке *<начало>;*

SELECT SUBSTR('barbarabarbarabarabar',8,7);

Функции для работы с числами

- ABS – абсолютное значение
- FLOOR – отбрасывает дробную часть
- CEIL – самое малое целое, которое равно или больше заданного числа

```
SELECT CEIL(5.2434243);
```

Функции работы с числами

- Функция округления– ROUND

ROUND(*<значимое числовое выражение>,<точность>)*

- Функция усечения– TRUNCATE

TRUNCATE(*<значимое числовое выражение>,<точность>)*

SELECT TRUNCATE(5.2474243, 2), ROUND(5.2474243, 2);

Пример

```
SELECT UNIV_NAME, RATING, ROUND(RATING, -1),
    TRUNCATE(RATING, -1)
FROM UNIVERSITY;
```

UNIV_NAME	RATING		
МГУ	606	610	600
ВГУ	296	300	290
НГУ	345	350	340
РГУ	416	420	410
БГУ	326	330	320
ТГУ	368	370	360
ВГМА	327	330	320
.....	.....	.....	.....

Задание 4

- Для студентов выведите первые 3 буквы фамилии и значение стипендии, округленное до целых

Функции работы с числами

- Тригонометрические функции – COS, SIN, TAN
- Экспоненциальная функция– (EXP)
- Логарифмические функции – (LN, LOG)
- Функция возведения в степень– POWER

POWER(*<значимое числовое выражение>,<экспонента>)*

- Определение знака числа – SIGN
- Вычисление квадратного корня– SQRT

Функции преобразования значений

CONVERT(value, type)

```
SELECT CONVERT('2014-02-28', DATE);
SELECT CONVERT(125, CHAR);
SELECT CONVERT('-4', SIGNED);
```

CAST(value AS type)

```
SELECT CAST(125 AS CHAR);
```

Преобразование дат

	MySQL DATE_FORMAT
4-digit year	%Y
2-digit year	%y
2 or 4-digit year, 20th century for 00-49	%Y
2-digit year, 20th century for 00-49	%y
Abbreviated month (Jan - Dec)	%b
Month name (January - December)	%M
Month (1 - 12)	%m
Abbreviated day (Sun - Sat)	%a
Day (1 - 31)	%d
Hour (0 - 23)	%H
Hour (1 - 12)	%h
Minutes (0 - 59)	%i
Seconds (0 - 59)	%s

Пример

```
SELECT SURNAME, NAME, BIRTHDAY, DATE_FORMAT(BIRTHDAY,
    '%d-%M-%y'), DATE_FORMAT(BIRTHDAY, '%d.%m.%y')
FROM STUDENT;
```

SURNAME	NAME	BIRTHDAY		
Иванов	Иван	3/12/1982	3-дек-1982	3.12.82
Петров	Петр	1/12/1980	1-дек-1980	1.12.80
Сидоров	Вадим	7/06/1979	7-июн-1979	7.06.79
Кузнецов	Борис	8/12/1981	8-дек-1981	8.12.81
Зайцева	Ольга	1/05/1981	1-май-1981	1.05.81
Павлов	Андрей	5/11/1979	5-ноя-1979	5.11.79
Котов	Павел	**NULL**	**NULL**	**NULL**
Лукин	Артем	1/12/1981	1-дек1981	1.12.81
Петров	Антон	5/08/1981	5-авг-1981	5.08.81
Белкин	Вадим	7/01/1980	7-янв-1980	7.01.80
.....	.....	.....	.....	.....

Задание 5

- Составьте запрос для таблицы STUDENT таким образом, чтобы выходная таблица содержала всего один столбец в следующем виде:

Б.КУЗНЕЦОВ; место жительства - БРЯНСК; родился - 8.12.81.

Изменение даты и времени

```
SELECT DATE_ADD("2017-06-15", INTERVAL -2 MONTH);

SELECT DATE_ADD("2017-06-15 09:34:21",
    INTERVAL 15 MINUTE);
```

Агрегирование и групповые функции. COUNT

- COUNT определяет количеств о строк или значений поля, выбранных посредством запроса, и не являющихся NULL-значениями;

```
SELECT COUNT(*)
FROM EXAM_MARKS;

SELECT COUNT(DISTINCT SUBJ_ID)
FROM SUBJECT;
```

Задание 6

- Напишите запрос, который позволяет подсчитать в таблице EXAM_MARKS количество различных предметов обучения.

Агрегирование и групповые функции. SUM и AVG

- SUM – вычисляет арифметическую сумму всех выбранных значений данного поля;
- AVG вычисляет среднее значение для всех выбранных значений данного поля;

```
SELECT AVG(MARK)
FROM EXAM_MARKS;
```

Агрегирование и групповые функции. MIN MAX

- MAX вычисляет наибольшее из всех выбранных значений данного поля;
- MIN вычисляет наименьшее из всех выбранных значений данного поля

Вычисление функций по группам GROUP BY

```
SELECT STUDENT_ID, MAX(MARK)
FROM EXAM_MARKS
GROUP BY STUDENT_ID;

SELECT STUDENT_ID, SUBJ_ID, MAX(MARK)
FROM EXAM_MARKS
GROUP BY STUDENT_ID, SUBJ_ID;
```

Выбор групп

- Предложение HAVING определяет критерий, по которому группы следует включать в выходные данные

```
SELECT SUBJ_NAME, MAX(HOUR)
FROM SUBJECT
GROUP BY SUBJ_NAME
HAVING MAX(HOUR) >= 72;
```

Задание 7

- Напишите запрос, который выполняет выборку для каждого студента значения его идентификатора и минимальной из полученных им оценок.

Пустые значения NULL

- COUNT вернет число строк, не содержащих пустые значения
- AVG вычисляет среднее значение всех **известных значений** множеств а элементов

Пустые значения NULL

- Условные операторы при отсутствии пустых значений возвращают либо **1** (истина), либо **0** (ложь). Если же в столбце присутствуют пустые значения, то может быть возвращено **NULL**

```
SELECT SURNAME, STIPEND>300
FROM student;
```