Тест по SQL

1. Для обновления данных в строках таблиц используется выражение
   1. UPDATE
   2. INSERT IN
   3. MODIFY
   4. SAVE
2. Для удаления строк таблицы используется выражение
   1. REMOVE
   2. DELETE
   3. DESTROY
   4. DEL
3. Для создания строк таблицы используется выражение
   1. INSERT INTO
   2. ADD RECORD
   3. ADD NEW
   4. INSERT NEW
4. Каким запросом можно получить все значения колонки `login` таблицы `users`?
   1. EXTRACT login FROM users
   2. SELECT ALL login FROM users
   3. SELECT login FROM users
   4. SELECT users.login
5. Каким запросом можно получить все значения всех колонок таблицы `users`?
   1. SELECT \* FROM users
   2. SELECT ALL FROM users
   3. EXTRACT ALL FROM users
   4. SELECT users
6. Каким запросом можно получить все значения всех колонок таблицы `users` для строк где значение колонки `login` равно `admin`?
   1. SELECT \* FROM users WHERE login=='admin'
   2. SELECT \* FROM users WHERE login='admin'
   3. SELECT ALL FROM users WHERE login LIKE 'admin'
   4. SELECT ALL FROM users WHERE login IS 'admin'
7. Для получения только уникальных значений используется выражение
   1. DISTINCT
   2. UNIQUE
   3. DIFFERENT
   4. UNIQ
8. Для сортировки результатов используется выражение
   1. SORT
   2. SORT BY
   3. ORDER
   4. ORDER BY
9. Как получить количество строк в таблице users
   1. SELECT LENGTH(\*) FROM users
   2. SELECT COUNT(\*) FROM users
   3. COUNT(\*) FROM users
   4. SHOW COUNT(\*) FROM users

Имеются таблицы:

*users*

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **login** | **team\_name** | **score** |
| 1 | vasya | Марсианские суслики | 150 |
| 2 | olya | Марсианские суслики | 175 |
| 3 | petya | Суровые тефтели | 50 |
| 4 | nastya | Суровые тефтели | 350 |

*avatars*

|  |  |  |
| --- | --- | --- |
| **id** | **user\_id** | **file\_path** |
| 1 | 1 | /opt/files/avatar1.png |
| 2 | 3 | /opt/files/avatar2.png |
| 3 | 3 | /opt/files/avatar3.png |
| 4 | NULL | /opt/files/avatar4.png |
| 5 | NULL | /opt/files/avatar5.png |
| 6 | NULL | /opt/files/avatar6.png |

1. Какое количество строк выдаст запрос к таблице `users` (без дополнительных условий) при использовании  
   `INNER JOIN avatars ON users.id=avatars.user\_id`
   1. 2
   2. 3
   3. 4
   4. 5
   5. 6
2. Какое количество строк выдаст запрос к таблице `users` (без дополнительных условий) при использовании  
   `LEFT JOIN avatars ON users.id=avatars.user\_id`
   1. 2
   2. 3
   3. 4
   4. 5
   5. 6
3. Какое количество строк выдаст запрос к таблице `avatars` (без дополнительных условий) при использовании  
   `INNER JOIN users ON users.id=avatars.user\_id`
   1. 2
   2. 3
   3. 4
   4. 5
   5. 6
4. Какое количество строк выдаст запрос к таблице `avatars` (без дополнительных условий) при использовании  
   `LEFT JOIN users ON users.id=avatars.user\_id`
   1. 2
   2. 3
   3. 4
   4. 5
   5. 6
5. Запрос который выведет сумму значений колонки `score` для всех уникальных значений колонки `team\_name` в таблице `users`
   1. SELECT DISTINCT team\_name, SUM(score)  
      FROM users
   2. SELECT UNIQ team\_name, SUM(score)  
      FROM users
   3. SELECT UNIQUE team\_name, SUM(score)  
      FROM users
   4. SELECT team\_name, SUM(score)  
      FROM users   
      GROUP BY team\_name
6. Запрос который выведет все уникальные значения `team\_name` таблицы `users`, для которых сумма значений `score` больше или равно 350
   1. SELECT team\_name  
      FROM users  
      GROUP BY team\_name  
      HAVING SUM(score) >= 350
   2. SELECT team\_name  
      FROM users  
      WHERE SUM(score) >= 350
   3. SELECT UNIQUE team\_name  
      FROM users  
      WHERE SUM(score) >= 350
   4. SELECT DISTINCT team\_name  
      FROM users  
      WHERE SUM(score) >= 350