Тест SQL

1. **Какая разница между CHAR, VARCHAR**

CHAR используется для хранения символьных строк фиксированной длины, в то время как VARCHAR используется для хранения строковых значений переменной длины.

1. **Какая разница между UNION и UNION ALL**

UNION - это оператор, который объединяет результаты двух или более запросов в одну таблицу. Он объединяет строки на основе совпадения значений в определенных столбцах.

UNION ALL - это то же самое, что и UNION, но он также объединяет дубликаты строк, если они есть в результирующей таблице.

1. **Что такое Primary Key / Foreign key?**

Primary key (первичный ключ) - это уникальный идентификатор, который используется для определения уникальности записи в таблице. Он может быть представлен в виде одного или нескольких полей, и его значение не может повторяться для разных записей в таблице.

Foreign key (внешний ключ) - это поле или комбинация полей в одной таблице, которое ссылается на уникальный идентификатор другой таблицы. Значение внешнего ключа должно быть уникальным и соответствовать значению первичного ключа в связанной таблице. Если значение внешнего ключа не соответствует значению первичного ключа, то возникает ошибка при попытке сохранения записи в базе данных.

1. **Чем таблица (Table) отличается от представления (View)?**

Таблица и представление имеют разные функции и используются в разных ситуациях. Таблица содержит конкретные данные, которые можно изменять и использовать для запросов, в то время как представление представляет собой абстракцию, которая создается на основе других объектов базы данных и используется для быстрого доступа к данным.

1. **Есть две таблицы «Офисы» (Office) и «Города» (City)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Office** | | |  | **City** | |
| **Office\_Id** | **Name** | **City\_id** |  | **City\_id** | **Name** |
| 1 | Office\_M1 | 1 |  | 1 | Moscow |
| 2 | Office\_L1 | 2 |  | 2 | London |
| 3 | Office\_M2 | 1 |  | 3 | Paris |
| 4 | Office\_R1 | 4 |  |  |  |
| 5 | Office\_R2 | 4 |  |  |  |

**Сколько строк вернут запросы:**

|  |  |
| --- | --- |
| SELECT \* FROM Office o LEFT JOIN City c ON o.City\_id=c.City\_id | 5 |
| SELECT \* FROM Office o INNER JOIN City c ON o.City\_id=c.City\_id | 3 |
| SELECT \* FROM Office o OUTER JOIN City c ON o.City\_id=c.City\_id | 6 |
| SELECT \* FROM Office o CROSS JOIN City c ON o.City\_id=c.City\_id | Именно такой запрос выдаст error, а обычный запрос выдаст 15 |
| SELECT \* FROM Office o FULL JOIN City c ON o.City\_id=c.City\_id | 6 |
| SELECT \* FROM Office o RIGHT JOIN City c ON o.City\_id=c.City\_id | 4 |
| SELECT \* FROM Office o, City c | 15 |
| SELECT \* FROM Office o, City c WHERE o.City\_id=c.City\_id | 3 |
| SELECT \* FROM Office o  UNION ALL SELECT \* FROM City c | error |
| SELECT Office\_Id FROM Office o  UNION All SELECT City\_id FROM City c | 8 |

|  |  |
| --- | --- |
| **Id** | **A** |
| -1 | Mary |
| 2 | 3 |
| 3 | 3 |
| -4 | <NULL> |
| 5 | -2 |
| 6 | <NULL> |

**Какой результат вернут запросы:**

|  |  |
| --- | --- |
| Select count(\*) from T | 6 |
| Select count(A) from T | 6 |
| Select count(distinct A) from T | 4 |
| Select count(distinct \*) from T | error |
| Select SUM(1) from T | 6 |

**Сколько строк вернут запросы:**

|  |  |
| --- | --- |
| SELECT \* FROM T WHERE A = NULL | 0 |
| SELECT \* FROM T WHERE id >= A | error |
| SELECT COUNT(\*), A FROM T GROUP BY A | 4 |

1. **Получить максимальное значение поля ID в таблице T без использования функции max()**

SELECT Id

from T

order by id desc

limit 1

1. **Выбрать записи с четными ID**

SELECT \* FROM T

WHERE Id % 2 != 0

1. **Посчитать количество положительных и отрицательных значений столбца ID в таблице T одним запросом**

SELECT

COUNT(CASE WHEN id > 0 THEN 1 END) AS positive\_count,

COUNT(CASE WHEN id < 0 THEN 1 END) AS negative\_count

\*FROM T