



724,96

Рейтинг

OTUS. Онлайн-образование

Цифровые навыки от ведущих экспертов



vlstrochkov 23 июля 2019 в 18:21

## Топ-65 вопросов по SQL с собеседований, к которым вы должны подготовиться в 2019 году. Часть I

Автор оригинала: Aayushi Johari

Блог компании OTUS. Онлайн-образование, SQL, Microsoft SQL Server, Интервью

Перевод

*Перевод статьи подготовлен для студентов курса «MS SQL Server разработчик»*

Реляционные базы данных являются одними из наиболее часто используемых баз данных по сей день, и поэтому навыки работы с SQL для большинства должностей являются обязательными. В этой статье с вопросами по SQL с собеседований я познакомлю вас с наиболее часто задаваемыми вопросами по SQL (Structured Query Language — язык структурированных запросов). Эта статья является идеальным руководством для изучения всех концепций, связанных с SQL, Oracle, MS SQL Server и базой данных MySQL.

Наша статья с вопросами по SQL — универсальный ресурс, с помощью которого вы можете ускорить подготовку к собеседованию. Она состоит из набора из **65 самых распространенных вопросов**, которые интервьюер может задать во время собеседования. Оно обычно начинается с базовых вопросов по SQL, а затем переходит к более сложным на основе

обсуждения и ваших ответов. Эти вопросы по SQL с собеседований помогут вам извлечь максимальную выгоду на различных уровнях понимания.

Давайте начнем!

## Вопросы по SQL с собеседований

### Вопрос 1. В чем разница между операторами DELETE и TRUNCATE?

DELETE	TRUNCATE
Используется для удаления строки в таблице	Используется для удаления всех строк из таблицы
Вы можете восстановить данные после удаления	Вы не можете восстановить данные (прим. перевод.: операции логируются по разному, но в SQL Server есть возможность сделать откат) транзакции)
DML-команда	DDL-команда
Медленнее, чем оператор TRUNCATE	Быстрее

### № Вопрос 2. Из каких подмножеств состоит SQL?

- *DDL (Data Definition Language, язык описания данных)* — позволяет выполнять различные операции с базой данных, такие как CREATE (создание), ALTER (изменение) и DROP (удаление объектов).
- *DML (Data Manipulation Language, язык управления данными)* — позволяет получать доступ к данным и манипулировать ими, например, вставлять, обновлять, удалять и извлекать данные из базы данных.
- *DCL (Data Control Language, язык контролирования данных)* — позволяет контролировать доступ к базе данных. Пример — GRANT (предоставить права), REVOKE (отозвать права).

### Вопрос 3. Что подразумевается под СУБД? Какие существуют типы СУБД?

База данных — структурированная коллекция данных. Система управления базами данных (СУБД) — программное обеспечение, которое взаимодействует с пользователем, приложениями и самой базой данных для сбора и анализа данных. СУБД позволяет пользователю взаимодействовать с базой данных. Данные, хранящиеся в базе данных, могут быть изменены, извлечены и удалены. Они могут быть любых типов, таких как строки, числа, изображения и т. д.

#### Существует два типа СУБД:

- Реляционная система управления базами данных: данные хранятся в отношениях (таблицах). Пример — MySQL.
- Нереляционная система управления базами данных: не существует понятия отношений, кортежей и атрибутов. Пример — Mongo.

### Вопрос 4. Что подразумевается под таблицей и полем в SQL?

Таблица — организованный набор данных в виде строк и столбцов. Поле — это столбцы в таблице. Например:

**Таблица:** Student\_Information

**Поле:** Stu\_Id, Stu\_Name, Stu\_Marks

### Вопрос 5. Что такое соединения в SQL?

Для соединения строк из двух или более таблиц на основе связанного между ними столбца используется оператор JOIN. Он используется для объединения двух таблиц или получения данных оттуда. В SQL есть 4 типа соединения, а именно:

- Inner Join (Внутреннее соединение)
- Right Join (Правое соединение)

- Left Join (Левое соединение)
- Full Join (Полное соединение)

### Вопрос 6. В чем разница между типом данных CHAR и VARCHAR в SQL?

И Char, и Varchar служат символьными типами данных, но varchar используется для строк символов переменной длины, тогда как Char используется для строк фиксированной длины. Например, char(10) может хранить только 10 символов и не сможет хранить строку любой другой длины, тогда как varchar(10) может хранить строку любой длины до 10, т.е. например 6, 8 или 2.

### Вопрос 7. Что такое первичный ключ (Primary key)?

Student Table	
Stu_ID	Stu_Name
1	John
2	Jack
3	Tyler
4	Sofia

- Первичный ключ — столбец или набор столбцов, которые однозначно идентифицируют каждую строку в таблице.
- Однозначно идентифицирует одну строку в таблице
- Нулевые (Null) значения не допускаются

Пример: в таблице Student StuID является первичным ключом.

### Вопрос 8. Что такое ограничения (Constraints)?

Ограничения (constraints) используются для указания ограничения на тип данных таблицы. Они могут быть указаны при создании или изменении таблицы. Пример ограничений:

- NOT NULL
- CHECK
- DEFAULT
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY

### Вопрос 9. В чем разница между SQL и MySQL?

SQL — стандартный язык структурированных запросов (Structured Query Language) на основе английского языка, тогда как MySQL — система управления базами данных. SQL — язык реляционной базы данных, который используется для доступа и управления данными, MySQL — реляционная СУБД (система управления базами данных), также как и SQL Server, Informix и т. д.

### Вопрос 10. Что такое уникальный ключ (Unique key)?

- Однозначно идентифицирует одну строку в таблице.
- Допустимо множество уникальных ключей в одной таблице.

### Вопрос 11. Что такое внешний ключ (Foreign key)?

- Внешний ключ поддерживает ссылочную целостность, обеспечивая связь между данными в двух таблицах.
- Внешний ключ в дочерней таблице ссылается на первичный ключ в родительской таблице.
- Ограничение внешнего ключа предотвращает действия, которые разрушают связи между дочерней и родительской таблицами.

### Вопрос 12. Что подразумевается под целостностью данных?

Целостность данных определяет точность, а также согласованность данных, хранящихся в базе данных. Она также определяет ограничения целостности для обеспечения соблюдения бизнес-правил для данных, когда они вводятся в приложение или базу данных.

### Вопрос 13. В чем разница между кластеризованным и некластеризованным индексами в SQL?

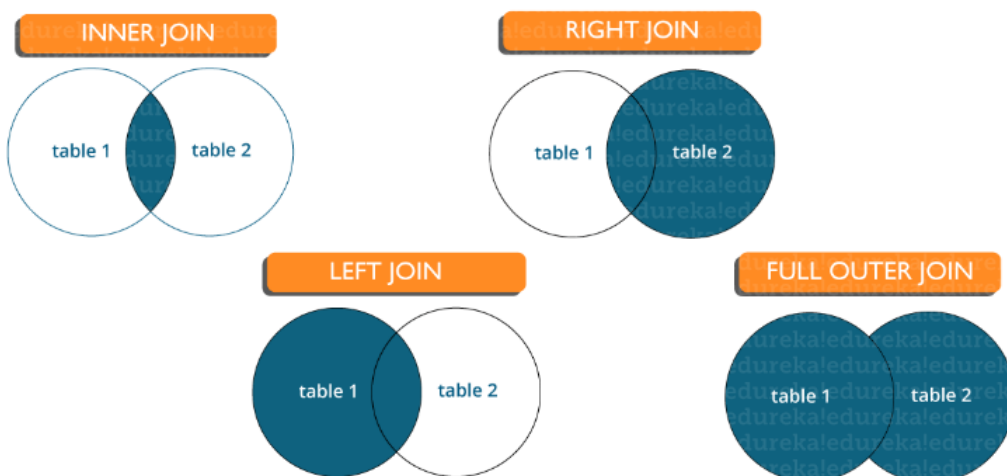
1. Различия между кластеризованным и некластеризованным индексами в SQL:  
Кластерный индекс используется для простого и быстрого извлечения данных из базы данных, тогда как чтение из некластеризованного индекса происходит относительно медленнее.
2. Кластеризованный индекс изменяет способ хранения записей в базе данных — он сортирует строки по столбцу, который установлен как кластеризованный индекс, тогда как в некластеризованном индексе он не меняет способ хранения, но создает отдельный объект внутри таблицы, который указывает на исходные строки таблицы при поиске.
3. Одна таблица может иметь только один кластеризованный индекс, тогда как некластеризованных у нее может быть много.

### Вопрос 14. Напишите SQL-запрос для отображения текущей даты.

В SQL есть встроенная функция `GetDate ()`, которая помогает возвращать текущий timestamp/дату.

### Вопрос 15. Перечислите типы соединений

Существуют различные типы соединений, которые используются для извлечения данных между таблицами. Принципиально они делятся на четыре типа, а именно:



**Inner join** (Внутреннее соединение): в MySQL является наиболее распространенным типом. Оно используется для возврата всех строк из нескольких таблиц, для которых выполняется условие соединения.

**Left Join** (Левое соединение): в MySQL используется для возврата всех строк из левой (первой) таблицы и только совпадающих строк из правой (второй) таблицы, для которых выполняется условие соединения.

**Right Join** (Правое соединение): в MySQL используется для возврата всех строк из правой (второй) таблицы и только совпадающих строк из левой (первой) таблицы, для которых выполняется условие соединения.

**Full Join** (Полное соединение): возвращает все записи, для которых есть совпадение в любой из таблиц. Следовательно, он возвращает все строки из левой таблицы и все строки из правой таблицы.

## Вопрос 16. Что вы подразумеваете под денормализацией?

Денормализация — техника, которая используется для преобразования из высших к низшим нормальным формам. Она помогает разработчикам баз данных повысить производительность всей инфраструктуры, поскольку вносит избыточность в таблицу. Она добавляет избыточные данные в таблицу, учитывая частые запросы к базе данных, которые объединяют данные из разных таблиц в одну таблицу.

## Вопрос 17. Что такое сущности и отношения?

**Сущности:** человек, место или объект в реальном мире, данные о которых могут храниться в базе данных. В таблицах хранятся данные, которые представляют один тип сущности. Например — база данных банка имеет таблицу клиентов для хранения информации о клиентах. Таблица клиентов хранит эту информацию в виде набора атрибутов (столбцы в таблице) для каждого клиента.

**Отношения:** отношения или связи между сущностями, которые имеют какое-то отношение друг к другу. Например — имя клиента связано с номером учетной записи клиента и контактной информацией, которая может быть в той же таблице. Также могут быть отношения между отдельными таблицами (например, клиент к счетам).

## Вопрос 18. Что такое индекс?

Индексы относятся к методу настройки производительности, позволяющему быстрее извлекать записи из таблицы. Индекс создает отдельную структуру для индексируемого поля и, следовательно, позволяет быстрее получать данные.

## Вопрос 19. Опишите различные типы индексов.

Есть три типа индексов, а именно:

1. **Уникальный индекс (Unique Index):** этот индекс не позволяет полю иметь повторяющиеся значения, если столбец индексируется уникально. Если первичный ключ определен, уникальный индекс может быть применен автоматически.
2. **Кластеризованный индекс (Clustered Index):** этот индекс меняет физический порядок таблицы и выполняет поиск на основе значений ключа. Каждая таблица может иметь только один кластеризованный индекс.
3. **Некластеризованный индекс (Non-Clustered Index):** не изменяет физический порядок таблицы и поддерживает логический порядок данных. Каждая таблица может иметь много некластеризованных индексов.

## Вопрос 20. Что такое нормализация и каковы ее преимущества?

Нормализация — процесс организации данных, цель которого избежать дублирования и избыточности. Некоторые из преимуществ:

- Лучшая организация базы данных
- Больше таблиц с небольшими строками
- Эффективный доступ к данным
- Большая гибкость для запросов
- Быстрый поиск информации
- Проще реализовать безопасность данных
- Позволяет легко модифицировать
- Сокращение избыточных и дублирующихся данных
- Более компактная база данных
- Обеспечивает согласованность данных после внесения изменений

## Вопрос 21. В чем разница между командами DROP и TRUNCATE?

Команда DROP удаляет саму таблицу, и нельзя сделать Rollback команды, тогда как команда TRUNCATE удаляет все строки из таблицы (*прим. перевод.: в SQL Server Rollback нормально отработает и откатит DROP*).

## Вопрос 22. Объясните различные типы нормализации.

Существует много последовательных уровней нормализации. Это так называемые нормальные формы. Каждая последующая нормальная форма включает предыдущую. Первых трех нормальных форм обычно достаточно.

- *Первая нормальная форма (1NF)* — нет повторяющихся групп в строках
- *Вторая нормальная форма (2NF)* — каждое неключевое (поддерживающее) значение столбца зависит от всего первичного ключа
- *Третья нормальная форма (3NF)* — каждое неключевое значение зависит только от первичного ключа и не имеет зависимости от другого неключевого значения столбца

## Вопрос 23. Что такое свойство ACID в базе данных?

ACID означает атомарность (Atomicity), согласованность (Consistency), изолированность (Isolation), долговечность (Durability). Он используется для обеспечения надежной обработки транзакций данных в системе базы данных.

**Атомарность.** Гарантирует, что транзакция будет полностью выполнена или потерпит неудачу, где транзакция представляет одну логическую операцию данных. Это означает, что при сбое одной части любой транзакции происходит сбой всей транзакции и состояние базы данных остается неизменным.

**Согласованность.** Гарантирует, что данные должны соответствовать всем правилам валидации. Проще говоря, вы можете сказать, что ваша транзакция никогда не оставит вашу базу данных в недопустимом состоянии.

**Изолированность.** Основной целью изолированности является контроль механизма параллельного изменения данных.

**Долговечность.** Долговечность подразумевает, что если транзакция была подтверждена (COMMIT), произошедшие в рамках транзакции изменения сохранятся независимо от того, что может встать у них на пути (например, потеря питания, сбой или ошибки любого рода).

## Вопрос 24. Что вы подразумеваете под «триггером» в SQL?

Триггер в SQL — особый тип хранимых процедур, которые предназначены для автоматического выполнения в момент или после изменения данных. Это позволяет вам выполнить пакет кода, когда вставка, обновление или любой другой запрос выполняется к определенной таблице.

## Вопрос 25. Какие операторы доступны в SQL?

В SQL доступно три типа оператора, а именно:

1. Арифметические Операторы
2. Логические Операторы
3. Операторы сравнения

## Вопрос 26. Совпадают ли значения NULL со значениями нуля или пробела?

Значение NULL вовсе не равно нулю или пробелу. Значение NULL представляет значение, которое недоступно, неизвестно, присвоено или неприменимо, тогда как ноль — это число, а пробел — символ.

## Вопрос 27. В чем разница между перекрестным (cross join) и естественным (natural join) соединением?

Перекрестное соединение создает перекрестное или декартово произведение двух таблиц, тогда как естественное соединение основано на всех столбцах, имеющих одинаковое имя и типы данных в обеих таблицах.

## Вопрос 28. Что такое подзапрос в SQL?

Подзапрос — это запрос внутри другого запроса, в котором определен запрос для извлечения данных или информации из базы данных. В подзапросе внешний запрос называется основным запросом, тогда как внутренний запрос называется подзапросом. Подзапросы всегда выполняются первыми, а результат подзапроса передается в основной запрос. Он может быть вложен в SELECT, UPDATE или любой другой запрос. Подзапрос также может использовать любые операторы сравнения, такие как >, < или =.

## Вопрос 29. Какие бывают типы подзапросов?

Существует два типа подзапросов, а именно: коррелированные и некоррелированные.

- **Коррелированный подзапрос:** это запрос, который выбирает данные из таблицы со ссылкой на внешний запрос. Он не считается независимым запросом, поскольку ссылается на другую таблицу или столбец в таблице.
- **Некоррелированный подзапрос:** этот запрос является независимым запросом, в котором выходные данные подзапроса подставляются в основной запрос.

## Вопрос 30. Перечислите способы получить количество записей в таблице?

Для подсчета количества записей в таблице вы можете использовать следующие команды:

```
SELECT * FROM table1
```

```
SELECT COUNT(*) FROM table1
```

```
SELECT rows FROM sysindexes WHERE id = OBJECT_ID(table1) AND indid < 2
```

Ещё 35 вопросов с ответами опубликуем в следующей части... Следите за новостями!

**Теги:** SQL, Oracle, MS SQL Server, MySQL, DELETE и TRUNCATE, подмножества SQL, DDL, DML, DCL, СУБД, объединения в SQL, Inner Join, Right Join, Left Join, Full Join, CHAR и VARCHAR2, Primary key, Constraints, Unique key, Foreign key, Денормализация, Unique Index, Clustered Index, Non-Clustered Index, DROP и TRUNCATE, свойство ACID в БД

**Хэбы:** Блог компании OTUS. Онлайн-образование, SQL, Microsoft SQL Server, Интервью

↑ +4 ↓ 501 150k 87 Поделиться



OTUS. Онлайн-образование

Цифровые навыки от ведущих экспертов



19,0

Карма

0,0

Рейтинг

**Владислав Строчков** @vlstorchkov

Email-маркетолог, PR-менеджер

Сайт Facebook ВКонтакте Telegram

### ПОХОЖИЕ ПУБЛИКАЦИИ

5 октября 2020 в 14:39

**Шифрование в MySQL: хранилище ключей**

↑ +9 1,7k 33 1



21 сентября 2020 в 12:19

### Производительность вычисляемых столбцов в SQL Server

+7 2,8k 24 1

10 сентября 2020 в 13:38

### Введение в графовые базы данных SQL Server 2017

+10 4,2k 61 1

## Комментарии 87

**usharik** 23 июля 2019 в 18:27 

0

Вопросы простые, но нужные. Жду следующей части!

**gleb\_l** 23 июля 2019 в 18:33 

+9

Кошмар. Мастерство автомеханика не в том, что он может назвать свойства каждого инструмента в своём наборе, но в том, что он умеет решить задачу, построив в уме оптимальную последовательность шагов, на каждом из которых используется тот или иной инструмент

**Maksclub** 24 июля 2019 в 09:08 

0

Все верно, в данном посте инструмент SQL, знающий его человек сможет построить такую механику, не знающий — нет. Как определить знающего — задать некоторые вопросы выше.

Вы можете попытаться объяснить, что и под задпчу можнл учить, но практика подсказывает, что задача отдается тем, кто знает, а не тем, кто готов знать (и тратить чужие деньги).

Ну и выше перечисленные знания можно уже причислить к фундаменту, более того — считаю эти вопросы очень поверхностными и знать нужно куда глубже.

**NoRegrets** 24 июля 2019 в 10:50 

+10

Это не вопросы на собеседовании. Это вопросы, которые, по мнению джуниора прочитавшего теорию по SQL, могут задать на собеседовании.

Откуда вообще эти вопросы набрали?

**gleb\_l** 24 июля 2019 в 12:39 

0

Все правильно. Не владея инструментами, человек априори будет не способен решить задачу. А владея — с определенным качеством, зависящим от степени владения. Именно это и нужно проверять — если целью является не проскочить через собес, а найти действительно хорошего специалиста

НЛО прилетело и опубликовало эту надпись здесь

**Rambalac** 23 июля 2019 в 18:55 

+8

В SQL не существует VARCHAR2.

**alexxpgrg** 23 июля 2019 в 19:55 

0

Оракл накладывает определённый отпечаток на мышление )

НЛО прилетело и опубликовало эту надпись здесь

**Rusty\_Fox** 24 июля 2019 в 21:23 

+3

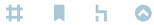


Oracle — sysdate, в PostgreSQL — now(), в MySQL — now(), curdate(), MsSQL — getdate.

А вот ANSI SQL знает вот такое CURRENT\_TIMESTAMP. По идее, должна работать везде (проверил на Oracle и PostgreSQL ).



**vdem** 25 июля 2019 в 12:50



↑ 0 ↓

В MySQL/MariaDB тоже CURRENT\_TIMESTAMP есть.



**uaggster** 31 июля 2019 в 20:25



↑ 0 ↓

Я в MSSQLSERVER только CURRENT\_TIMESTAMP и пользуюсь.

Т.к. постоянно забываю, как эта гребаная getdate пишется.



**osipov\_dv** 23 июля 2019 в 19:00



↑ 0 ↓

Ну вот, даже про транзакцию не спросили... Не говоря уже про WAL и как работает движок базы данных.

Какой смысл этих вопросов, их можно посмотреть в справочнике, а понимание этих механизмов объяснит работу почти любой БД.



**eefadeev** 24 июля 2019 в 14:22



↑ +1 ↓

Строго говоря WAL никак не связан с SQL. Если покопаться можно найти нежурналируемые реализации РСУБД.

НЛО прилетело и опубликовало эту надпись здесь



**Daemonis** 23 июля 2019 в 19:07



↑ 0 ↓

«Inner join в MySQL»

А в остальных СУБД?



**vdem** 23 июля 2019 в 19:14



↑ +3 ↓

Невычитано от слова совсем, даже лень список багов составлять.

P.S. 1) «SQL — ядро реляционной базы данных» — что это было? 2) «Допустимы нулевые значения» (это про unique key) — нулевые это которые? 0 или таки NULL? 3) «Значение NULL представляет значение, которое недоступно, неизвестно, *присвоено* или неприменимо» — может таки *неприсвоено*? И не указано, что в т.ч. NULL не равен самому себе.



**KristinaMyLife** 25 июля 2019 в 11:37



↑ +1 ↓

Спасибо большое, немного не синхронизировались, сейчас постарались подправить все комментарии, чтобы было корректнее и понятнее.



**CrushBy** 23 июля 2019 в 19:51



↑ +1 ↓

А можно уточнить, а по какому SQL стандарту это вопросы? SQL-92 или самому свежему SQL-2016 (который кстати, полностью далеко не все СУБД поддерживают)?

Например, в SQL-92 не нахожу функции GetDate.



**ncix** 23 июля 2019 в 19:55



↑ +3 ↓

16 Денормализация — это всего лишь сознательное нарушение одной из нормальных форм.

Вообще некоторые ответы довольно спорные.



**McKinseyBA** 23 июля 2019 в 20:33



↑ +3 ↓

я бы сказал, что больше половины спорные.

Чего стоит одно объяснение ACID — в русской википедии намного корректнее.

**minamoto** 24 июля 2019 в 18:04

↑ +2 ↓

Вообще я не увидел ни одного корректного вопроса/ответа. Косяк либо в постановке вопроса, либо в формулировке ответа, либо в нужности самого вопроса (типа NATURAL JOIN, который я, за много лет работы с базой, ни разу не видел и знать не знаю).

**McKinseyBA** 23 июля 2019 в 20:21

↑ +1 ↓

А где же ключевой вопрос «икспертов» — чем LEFT JOIN отличается от RIGHT JOIN?

**ncix** 23 июля 2019 в 21:16

↑ 0 ↓

И что будет, если написать просто JOIN

**alexhott** 23 июля 2019 в 21:27

↑ 0 ↓

а если написать left hash join или left loop join

**ncix** 23 июля 2019 в 21:31

↑ +3 ↓

попахивает майкрософтщиной

**taliban** 23 июля 2019 в 21:04

↑ 0 ↓

Вы можете восстановить данные после удаления

Как это восстановить данные после удаления?

**ncix** 23 июля 2019 в 21:32

↑ 0 ↓

REVERT TRANSACTION?

**taliban** 23 июля 2019 в 22:31

↑ 0 ↓

1. точно ли имеется ввиду незакомиченные операции?
- 1.1 точно ли с truncate так же нельзя (в статье указывается truncate как опозит)?
2. вы меня спрашиваете или это ответ на мой вопрос?
3. Я не думаю что автор это имел ввиду )

**vdem** 23 июля 2019 в 22:34

↑ 0 ↓

С TRUNCATE точно нельзя. Ну т.е. если эта команда и была вызвана в транзакции, то после отката таблица все равно останется пустой.

UPD:

После операции TRUNCATE для некоторых СУБД (например, Oracle) следует неявная операция COMMIT. Поэтому удаленные в таблице записи нельзя восстановить операцией ROLLBACK. Но существуют и СУБД, в которых операция TRUNCATE может участвовать в транзакциях, например, Microsoft SQL Server.

Короче, не следует полагаться на то, что после отката на некой СУБД данные, удаленные TRUNCATE, вернутся.

**taliban** 23 июля 2019 в 22:43

↑ 0 ↓

Вот да, mssql может откатить truncate. Но в любом случае, это странно просто сказать что данные можно восстановить, я ведь могу подумать что и транзакции можно не использовать, мне то всего одну запись удалить надо, а потом восстановить вдруг

**vdem** 23 июля 2019 в 23:35 # 1 2 3 4

↑ +2 ↓

Я тоже очень удивился, прочитав слово «восстановление», потом уже дошло, что имелся в виду откат транзакции :) В общем или сам источник кривой, или криво переведено да еще и не вычитано, а скорее всего и то и другое.

**Tyusha** 23 июля 2019 в 21:10 # 1

↑ +4 ↓

Наверное следует поправить следы машинного перевода в вопросе 4, но само по себе это здорово:

| SВопросL

**harlong** 24 июля 2019 в 04:15 # 1 2 3 4

↑ +2 ↓

Энциклопедии приветствуют Вас. Подозреваю, что в оригинале в заголовках вопросов было Q1, Q2, ..., Qx

**alexhott** 23 июля 2019 в 21:26 # 1

↑ 0 ↓

триггеры в MS SQL выполняются либо вместо либо после выполнения инструкции.  
и чем отличается count(\*) от count(column\_name)?

**InChaos** 24 июля 2019 в 12:07 # 1 2 3 4

↑ 0 ↓

А Вы сравните сами, а если вдруг одно поле окажется binary с файлами по несколько мегабайт то сюрприз.  
Я вообще всегда использую Count(1)

**alexhott** 24 июля 2019 в 16:47 # 1 2 3 4

↑ 0 ↓

на самом деле есть еще сюрприз  
если делать count по конкретному полю то результатом может быть не полное количество строк, а только те где данное поле не null  
как тут отработает Count(1) не знаю

**eefadeev** 24 июля 2019 в 17:38 # 1 2 3 4

↑ +1 ↓

Сюрприз?! Серьёзно?

А если подумать — как же может отработать COUNT(1) если нам доподлинно известно что 1 IS NOT NULL? Интрига...

**InChaos** 1 августа 2019 в 16:58 # 1 2 3 4

↑ 0 ↓

На моей тестовой базе в таблице на 10 млн строк, всего 3 числовых поля COUNT(\*) — 1,5 сек, COUNT(1) — 0,5 сек.  
COUNT(\*) именно читает полностью всю строку, а COUNT(1) только заголовки, т.к. не указано поле и ему нечего читать.

**eefadeev** 1 августа 2019 в 19:32 # 1 2 3 4

↑ 0 ↓

Уточняющий вопрос: сколько раз вы выполняли эти запросы (по какому количеству исполнений получена статистика)?

**epee** 24 июля 2019 в 18:57 # 1 2 3 4

↑ 0 ↓

| как тут отработает Count(1) не знаю

в MS SQL Server отработает без всяких сюрпризов — посчитает все строки. тоже использую COUNT именно так

**nikolayv81** 28 июля 2019 в 02:01 # 1 2 3 4

↑ 0 ↓

Ну это как бы в стандарте записано, тут скорее возник бы вопрос почему count(\*) считает все строки, даже те в которых все значения во всех полях null :)

**minamoto** 24 июля 2019 в 17:54

↑ +1 ↓

Если вы работаете с MS SQL, то можете выполнить `count(1)` и `count(*)` и посмотреть планы. Сюрприз — они одинаковые, т.е. ваша попытка оптимизации лишняя. ЕМНИП, то ли с 2000-й, то ли с 2005-й версии SQL Server.

**epee** 24 июля 2019 в 19:04

↑ 0 ↓

именно, поэтому если видишь как кто-то использует `COUNT(1)` можно быть почти уверенным что он пользуется SQL Server с довольно древних версий :)))

**nikolayv81** 28 июля 2019 в 02:08

↑ -1 ↓

Всегда пишу `count(1)` но никогда не работал с mssql, причина более удобно, плюс нахождение `*` в запросе, для меня, признак не полной проработки и незавершённости.

**unfilled** 24 июля 2019 в 12:28

↑ +1 ↓

`count(*)` считает все строки, `count(column_name)` — строки, в которых `column_name` is not null

**AlexToArt** 25 июля 2019 в 10:01

↑ 0 ↓

`count(column_name)` посчитает количество NOT NULL значений в `column_name`

**x\_shader** 23 июля 2019 в 22:32

↑ +5 ↓

Вопросы никак не для собеседования на SQL разработчика.  
Для фронтенда, чтоб проверить кругозор. А скорее, умение повторить справочник перед интервью.

**gudvinr** 24 июля 2019 в 00:14

↑ +1 ↓

То есть 65 — только *самые* распространенные и это не предел? А чтоб к остальным подготовиться — надо знать назубок ANSI SQL и держать шпаргалку с документацией по постгресу/ораклу/мускулу?

**saboteur\_kiev** 24 июля 2019 в 04:19

↑ 0 ↓

Нереляционная система управления базами данных: не существует понятия отношений, кортежей и атрибутов. Пример — Mongo

Причем тут кортежи и атрибуты?

Реляционные базы — это базы, с отношениями между таблицами. Все остальное — может быть реализовано, а может не быть.

Вопросы дублируются (1 и 21)

Язык слишком академический, и неестественный — автор явно сам не слишком хорошо в SQL.

Часто путает сам SQL и его конкретную реализацию в Оракле или MySQL, те же триггеры.

Если планируется еще одна статья — настоячиво рекомендую пригласить нормального ДБА-шника и потратить хотя бы 1-2 вечера на вычитывание. Пусть это будет не 65 а топ-20 вопросов по SQL, зато реально полезных и описанных по-человечески.

**Magikan** 24 июля 2019 в 04:46


↑ +3 ↓

пологаю DBA, прочитавшие эту статью, уже пишут свой топ в ответку с со всеми вытекающими )


**Pydeg** 24 июля 2019 в 05:33

↑ +1 ↓

Реляционные базы прежде всего реализуют реляционную алгебру, в которой множества, кортежи, атрибуты и их отношения. Таблицы это уже абстракция реляционной базы данных и вместо них может быть что-то другое.

 **Anton\_Zh** 24 июля 2019 в 07:25 # 0

По-моему определение 1NF совсем неправильно, ну либо переводчик так криво перевел, что смысл потерялся.

 **unfilled** 24 июля 2019 в 07:33 # 5

q1 TRUNCATE «Вы не можете восстановить данные»

Если речь идёт об откате транзакции — то это зависит от СУБД. В MS SQL Server можете.

q5 JOIN во всех русскоязычных книгах/материалах (встречавшихся мне) переводится как СОЕДИНЕНИЕ, а не ОБЪЕДИНЕНИЕ.

q10 «Допустимы нулевые значения.» в Unique key — «NULL»евые

q13 «чтение из некластеризованного индекса происходит относительно медленнее» да неужели? А если он покрывающий?

q15 на хабре совсем недавно был материал, почему «диаграммы Венна» плохо подходят для джойнов. Можно было бы и ссылку дать.

q18 «Индексы относятся к методу настройки производительности, позволяющему быстрее извлекать записи из таблицы. Индекс создается для каждого значения и, следовательно, позволяет быстрее получать данные.» если кому-то стало понятно «Что такое индекс» после такого ответа — я ему очень завидую.

q21 «DROP удаляет саму таблицу, и ее нельзя откатить из базы данных» что тут написано? Что нельзя откатить? Таблицу? Дроп? Опять-таки зависит от СУБД. В MS SQL Server можно откатить транзакцию с DROP TABLE.

q23 «Изолированность. Основной целью изолированности является контроль параллелизма.» Классный ответ. Параллелизма чего? Чем традиционное определение не устраивает?

«Долговечность подразумевает, что если транзакция запущена, она будет происходить независимо от того, что может встать у нее на пути, например, потеря питания, сбой или ошибки любого рода.» о, а можно мне хоть в одной СУБД такую «долговечность»? Я такую очень хочу.

Сначала думал, что мои вопросы к первоисточнику, но оказалось, примерно 50/50. Те же свойства ACID в источнике описаны корректно. Самые пуканоразрывающие вопросы (для меня) на совести переводчиков. Какой надмозг перевёл «concurrency control» как «параллелизм», «has been committed» как «была запущена», «NULL values» как «нулевые значения» и «You cannot rollback» как «вы не можете восстановить»?

 **gudvinr** 24 июля 2019 в 21:57 # 0

Опять-таки зависит от СУБД. В MS SQL Server можно откатить транзакцию с DROP TABLE.

От СУБД и уровня изоляции. Многие СУБД, которые позволяют DDL откатывать, не позволяют это делать на уровнях изоляции менее строгих, чем Serializable, который используется не сказать чтобы часто.

 **saboteur\_kiev** 25 июля 2019 в 13:32 # 0

Бэкап — это тоже вариант отката транзакций.

Или GRP в оракле — еще одна помесь бэкапа и транзакций


 **eefadeev** 25 июля 2019 в 13:43 # 1

Бэкап — это тоже вариант отката транзакций.

Нет. Восстановление из бэкапа — это возврат к состоянию БД в некоторый момент времени. В некоторых частных случаях (когда в БД нет и не будет никаких других транзакций кроме вашей) — его можно считать вариантом «отката». Но это именно частный случай.

 **KristinaMyLife** 25 июля 2019 в 11:35 # 0

спасибо большое, постарались подправить косяки, так же добавила комменты кое-где по SQL Server.

 **winmasta** 24 июля 2019 в 11:05 # 1

В вопросе №1 DELETE — DML команда, но в вопросе №2 она уже DDL, как так?

 **tas** 24 июля 2019 в 11:27  

 0 

Вопрос немного посложнее:






tablename1 имеет 2 поля fieldname1 и fieldname2. Также tablename1 имеет индекс по полю fieldname2.

При выполнении запроса вида:

```
SELECT fieldname1 FROM tablename1 WHERE fieldname2=1
```

... имеем Index Scan в плане запроса.

Назовите как MIN 3 причины, почему вместо Index Seek выполняется Index Scan?

 **uaggster** 24 июля 2019 в 21:32    

 0 

Эээ...






1. fieldname2 имеет тип varchar, например, и, следовательно, каждое значение в столбце должно быть преобразовано в тип int для сравнения.
2. Индекс по полю имеет низкую селективность, например, 80% значений = 1, и оптимизатор верно оценивает статистику.
3. Значение лежит за пределами статистики по полю, например, вследствие ее устаревания, и оптимизатор неверно оценивает кардинальность.

И потом, индекс скан какого индекса? По полю fieldname2, или скан какого-то другого индекса, кластерного например, или покрывающего индекса, например вида fieldname1 include (fieldname2)?

Т.е. «почему оптимизатор игнорирует индекс по полю, а использует другой».

Ответ будет: потому, что оптимизатор оценивает издержки на Index seek + key lookup выше, чем скан стороннего индекса, с «добычей» искомого значения из include поля, ну, или из кластерного индекса.

Так пойдет?

 **eefadeev** 25 июля 2019 в 05:04    

 0 

Я бы добавил

4. Потому что размер индекса меньше некоторого порога (например в две страницы, могут быть различия в реализациях), но индекс при этом покрывающий. Оптимизатор вполне разумно решил что проще будет его просканировать :)




 **tas** 25 июля 2019 в 10:13    

 0 

Да, Вы приняты :)

Вы назвали 3 основных причины: преобразование типов, низкая селективность и проблема со статистикой. Ниже добавили 4-ю...

На счет «индекс скан какого индекса?» — то в примере четко написано, что «tablename1 имеет индекс по полю fieldname2». Все! Не нужно усложнять себе задачу додумывая возможное наличие других индексов или условий...

 **uaggster** 25 июля 2019 в 13:16    

 0 






Ненене!

"... имеем Index Scan в плане запроса. "

Не сказано, Scan какого индекса. Т.е. возможен вариант скана в плане, но другого индекса :-)

Так что не «додумывать», а предусматривать варианты.

:-)

 **DrunkBear** 26 июля 2019 в 10:46    

 0 

Хм, когда отключили автообновление индекса и не сделали ребилд после заливки большого куска данных — вроде тоже будет печальный скан?

К тому же, может быть всё уже хорошо, но старый план выполнения вместо поиска всё равно предложит скан.

 **eefadeev** 26 июля 2019 в 13:12    

 0 

То есть индекс невалидный? Какой смысл, тогда, его сканировать? Его перестраивать надо — в нём же половины данных просто нет!



**tas** 26 июля 2019 в 17:14



↑ 0 ↓

Этот вариант входит в «проблема со статистикой», конкретных причин там может быть много...



**просто** 25 июля 2019 в 08:57



↑ 0 ↓

Кмк, надо отделять мух от котлет.

Т.е. вопросы по SQL для прикладных программистов и вопросы по тюнингу производительности для DBA (в том числе особенности работы индексов).

Ответы на первую часть более менее общие для разных СУБД.

Ответы на вторую сильно завязаны на особенности движка конкретной СУБД. Более того, могут отличаться, в зависимости от версии движка.



**DrunkBear** 24 июля 2019 в 11:39



↑ +1 ↓

Забыли про cross join в списке join, хотя про него многие забывают и если видят — делают огромные глаза и спрашивают «а шо это??»  
Типов СУБД гораздо больше — придёт человек и расскажет про k-v хранилища, графо-ориентированную СУБД или объектное хранилище, неудобно же будет.

В том же MSSQL с 2012 появились колоночные индексы, с 2014 — in-memory таблицы.



**uaggster** 24 июля 2019 в 21:48



↑ 0 ↓

А с 2017 — графы.



**просто** 24 июля 2019 в 14:15



↑ +2 ↓

Правильное название для пункта 15 в его текущей редакции:

"Вопрос 15. Перечислите самые распространенные мифы о типах соединений SQL."



**Rusty\_Fox** 24 июля 2019 в 23:23



↑ +1 ↓

Вы об этом?



**просто** 25 июля 2019 в 08:43



↑ +1 ↓

Спасибо за ссылку. Там все изложено верно. Здесь тоже.

К сожалению, подавляющее большинство программистов об этом не знает и свято верит в мифы изложенные в ответе на Вопрос 15.



**Rusty\_Fox** 24 июля 2019 в 21:12



↑ 0 ↓

27 вопрос с подвохом. Вот зачем кому-то спрашивать, в чем разница между cross и natural? Почему именно между ними? Суть в том, что если в двух таблицах нет одинаковых столбцов, natural начинает работать как cross. Очевидно, автор видел такой вопрос, знает, как работают обе конструкции, но не понимает, откуда вопрос возник.



**uaggster** 26 июля 2019 в 16:29



↑ 0 ↓

Natural join — это, разве, не типичная оракуловская фишка?



**nikolayv81** 28 июля 2019 в 02:15



↑ 0 ↓

ИМХО natural join это плохая практика, ещё хуже чем select \*



**Rusty\_Fox** 31 июля 2019 в 15:18

↑ 0 ↓

Вроде как нет. На практике ни разу не встречал, если честно.

**Victor\_D** 25 июля 2019 в 08:30

↑ 0 ↓

На практике, на 95% собеседованиях по SQL, в основном, задают практические задачи с различными изощренными применениями joinov.

**eugeneyp** 25 июля 2019 в 08:30

↑ 0 ↓

Прямо таки набор заблуждений видимый сквозь призму одной БД.

- 1) Два вида ББ, как мнение мое и не правильное. А как-же иерархические, сетевые и т.д.?
- 2) Как писали многие RDBMS умеют откатывать DDL по rollback. Т.е. утверждение не верно.
- 3) Нету вопроса про DOMAIN
- 4) Где в foreign key модификаторы on delete set null, on update cascade? Как раз set null разрушение связи.
- 5) Нормализация БД это приведение к нормальной форме. Всё остальное это следствие нормализации.
- 6) Первая нормальная форма описана не правильно, смотрите описание в Wiki. В двух словах каждое поле должно содержать одно значение. Т.е. в современных реалиях РСУБД не позволит создать БД которая не удовлетворяет 1НФ.
- 7) Про индексы написан бред. Насколько упадет производительность БД если в таблице будет много индексов на булево поле, или на поле константы? Про индексы на несколько полей автор наверное не знает? Что сужествуюн hash, bitmap и т.д. виды индексов?
- 8) про null надо спрашивать is [not] distinct from и в чер разнца с равенством и неравенством

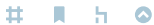
В итоге эта статья является списком вопросов и вариантами ответов которые показывают вашу некомпетентность.

**funca** 25 июля 2019 в 22:55

↑ 0 ↓

Т.е. в современных реалиях РСУБД не позволит создать БД которая не удовлетворяет 1НФ.

Нормализация относится к схеме, а не данным. Понятие атомарности в терминах НФ зависит от предметной области. Ни кто не помешает разработчику создать текстовое поле, чтобы хранить там имя и фамилию человека. Если с ФИО будут работать как с неделимым значением, тогда выбор корректный для 1НФ. Если предполагаются связи по отдельным словам — тогда нет.

**eugeneyp** 26 июля 2019 в 10:41

↑ 0 ↓

По моему это теологический спор.

Посмотрел wiki. Все ссылки на книги изданные после того как мне преподавали нормальные формы. Но они сводятся к тому что один атрибут к одной строке не может содержать два и более значения. К сожалению сейчас не используют термин DOMAIN, с ним некоторые вещи реализуются и объясняются проще.

Если считать что поле содержит два значения разделенных запятой или переводом строки то структуру и записи в БД которые не удовлетворяют первой нормальной формы создать можно.

Если считать что в поле можно запихнуть два INTEGER в виде SET или ARRAY то можно только в некоторых РСУБД.

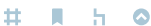
А если брать БД когда в полях храниться JSON/BSON/XML и т.д. то про какие нормальные формы вообще можно говорить? Если

**eefadeev** 26 июля 2019 в 15:14

↑ 0 ↓

если брать БД когда в полях храниться JSON/BSON/XML и т.д. то про какие нормальные формы вообще можно говорить?

Ни про какие. Это уже не реляционные данные (точнее таблицы с такими полями не находятся даже в 1НФ если вы хоть как-то на стороне БД используете знание о наличии у них внутренней структуры).

**funca** 26 июля 2019 в 23:59

↑ 0 ↓

Тут лучше начинать вот с этой Википедии [en.m.wikipedia.org/wiki/Relational\\_model](https://en.m.wikipedia.org/wiki/Relational_model).

The relational model's central idea is to describe a database as a collection of **predicates** over a finite set of predicate variables, describing constraints on the possible values and combinations of values.

В реляционной модели данные представляют собой множества предикатов — логических утверждений касательно фактов

предметной области, которые можно свести к true или false. Отсюда следует требование атомарности: один факт — одно значение. Если его не выполнить, то в модели проявятся логические противоречия — аномалии. Собственно нормализация это метод последовательной трансформации реляционной модели с целью исключения из нее этих вот самых аномалий. Домены накладывают дополнительные ограничения на допустимые значения конкретного атрибута. Это могут быть хоть простые типы, например целые числа, хоть даты (из которых за пределами модели можно извлекать компоненты: день, месяц, год и т.п.), хоть массивы, JSON, XML или даже отдельные отношения. Пока в вашей модели такое значение отражает единичный факт и для связи с другими фактами не требует углубляться во внутреннюю структуру, оно соответствует 1НФ.



**Mikluho** 25 июля 2019 в 20:19



+2



Плохой, вредный оригинал, не надо было его переводить, и уж тем более использовать!

В крайнем случае — как список того, как не надо отвечать на вопросы!

Такое ощущение, что автор где-то нагуглил списки вопросов на собеседованиях, а потом нагуглил первые попавшиеся ответы...

**funca** 25 июля 2019 в 23:02



+1



Сейчас такой вид постов в англоязычном интернете достаточно распространен. «10 удивительных фактов», «12 топовых вопросов», «5 основных функций». Подавляющее большинство из них откровенный мусор. Наверняка какая-то часть собирается автоматически тупо по набору ключевых слов.



**Mikluho** 25 июля 2019 в 23:17



0



Видимо они у себя там так за посещаемость борются. Но здесь-то это зачем?

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

- Сутки
- Неделя
- Месяц

Почему Джеф Безос – самый опасный политикан на планете

+59 57,9k 149 478

Украденные данные CD Projekt Red выставили на аукционе

+21 37k 13 60

Добраться до Марса: новые семь минут ужаса всего через семь дней

+70 21,5k 34 74

Дуров рассказал, как будет монетизироваться Telegram

+25 19,7k 7 44

Чего бы почитать про Data Science? Коллекция классных постов

Мегапост

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	Мерч