

原创

数据库面试题(开发者必看)

2018-03-05 09:19:01

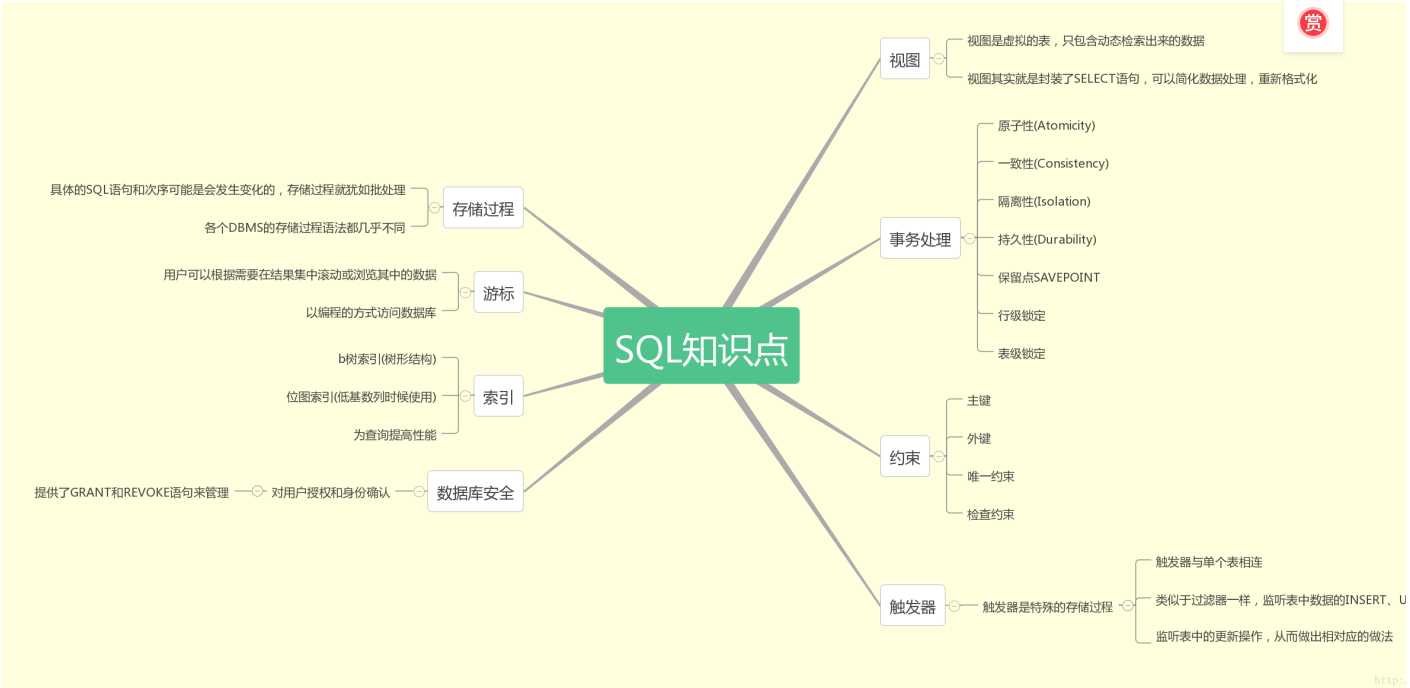
Java3y

阅读数 10418

更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/Java_3y/article/details/79441845

数据库常见面试题(开发者篇)



什么是存储过程？有哪些优缺点？

什么是存储过程？有哪些优缺点？

存储过程就像我们编程语言中的函数一样，封装了我们的代码(PL/SQL、T-SQL)。

存储过程的优点：

- 能够将代码封装起来
- 保存在数据库之中
- 让编程语言进行调用
- 存储过程是一个预编译的代码块，执行效率比较高
- 一个存储过程替代大量T_SQL语句，可以降低网络通信量，提高通信速率

存储过程的缺点：

- 每个数据库的存储过程语法几乎都不一样，十分难以维护（不通用）
- 业务逻辑放在数据库上，难以迭代

三个范式是什么

三个范式是什么

第一范式（1NF）：数据库表中的字段都是单一属性的，不可再分。这个单一属性由基本类型构成，包括整型、实数、字符型、逻辑型、日期型等。

第二范式（2NF）：数据库表中不存在非关键字段对任一候选关键字段的部分函数依赖（部分函数依赖指的是存在组合关键字中的某些字段决定非关键字段都完全依赖于任意一组候选关键字。

第三范式（3NF）：在第二范式的基础上，数据表中如果不存在非关键字段对任一候选关键字段的传递函数依赖则符合第三范式。所谓传递函数依赖，指“C”的决定关系，则C传递函数依赖于A。因此，满足第三范式的数据库表应该不存在如下依赖关系：关键字段 → 非关键字段x → 非关键字段y

上面的文字我们肯定是看不懂的，也不愿意看下去的。接下来我就总结一下：

- 首先要明确的是：**满足着第三范式，那么就一定满足第二范式、满足着第二范式就一定满足第一范式**
- 第一范式：**字段是最小的单元不可再分**
 - 学生信息组成学生信息表，有年龄、性别、学号等信息组成。这些字段都不可再分，所以它是满足第一范式的
- 第二范式：满足第一范式，**表中的字段必须完全依赖于全部主键而非部分主键。**
 - **其他字段组成的这行记录和主键表示的是同一个东西，而主键是唯一的，它们只需要依赖于主键，也就成了唯一的**
 - 学号为1024的同学，姓名为Java3y，年龄是22岁。姓名和年龄字段都依赖着学号主键。
- 第三范式：满足第二范式，**非主键外的所有字段必须互不依赖**
 - **就是数据只在一个地方存储，不重复出现在多张表中，可以认为就是消除传递依赖**
 - 比如，我们大学分了很多系（中文系、英语系、计算机系.....），这个系别管理表信息有以下字段组成：系编号，系主任，系简介，系架构。在学生信息表添加系编号，系主任，系简介，系架构字段呢？不行的，因为这样就冗余了，非主键外的字段形成了依赖关系(依赖到学生信息表做法是：学生表就只能增加一个系编号字段。

参考链接：

- <https://www.zhihu.com/question/24696366>
- <http://www.cnblogs.com/CareySon/archive/2010/02/16/1668803.html>

什么是视图？以及视图的使用场景有哪些？

什么是视图？以及视图的使用场景有哪些？

视图是一种基于数据表的一种虚表

- （1）视图是一种虚表
- （2）视图建立在已有表的基础上, 视图赖以建立的这些表称为基表
- （3）**向视图提供数据内容的语句为 SELECT 语句,可以将视图理解为存储起来的 SELECT 语句**
- （4）视图向用户提供基表数据的另一种表现形式
- （5）视图没有存储真正的数据，真正的数据还是存储在基表中
- （6）程序员虽然操作的是视图，但最终视图还会转成操作基表
- （7）一个基表可以有0个或多个视图

有的时候，我们可能只关系一张数据表中的某些字段，而另外的一些人只关系同一张数据表的某些字段...

那么把全部的字段都都显示给他们看，这是不合理的。

我们应该做到：**他们想看到什么样的数据，我们就给他们什么样的数据...一方面就能够让他们只关注自己的数据，另一方面，我们也保证数据不被泄露出来...**

我们在查询数据的时候，常常需要编写非常长的SQL语句，几乎每次都要写很长很长....上面已经说了，**视图就是基于查询的一种虚表，也就是说，视图来的数据进行封装。。。那么我们在使用的时候就会变得非常方便...**

值得注意的是：**使用视图可以让我们专注与逻辑，但不提高查询效率**

drop、delete与truncate分别在什么场景之下使用？

drop、delete与truncate分别在什么场景之下使用？

我们来对比一下他们的区别：

drop table

- 1)属于DDL
- 2)不可回滚
- 3)不可带where
- 4)表内容和结构删除
- 5)删除速度快

truncate table

- 1)属于DDL
- 2)不可回滚
- 3)不可带where
- 4)表内容删除
- 5)删除速度快

delete from

- 1)属于DML
- 2)可回滚
- 3)可带where
- 4)表结构在，表内容要看where执行的情况
- 5)删除速度慢,需要逐行删除
- **不再需要一张表的时候，用drop**
- **想删除部分数据行时候，用delete，并且带上where子句**
- **保留表而删除所有数据的时候用truncate**

索引是什么？有什么作用以及优缺点？

索引是什么？有什么作用以及优缺点？

什么是索引【Index】

- （1）是一种快速查询表中内容的机制，类似于新华字典的目录
- （2）运用在表中某些字段上，但存储时，独立于表之外



40



4

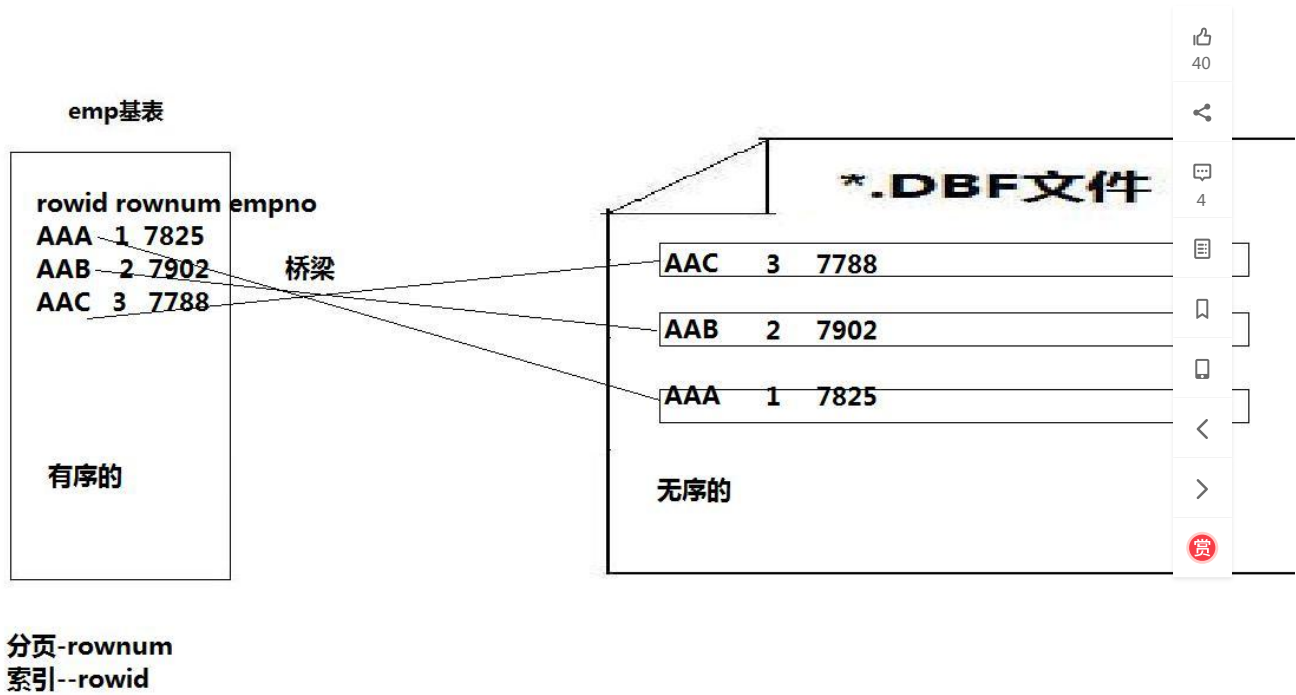


编程语言大PK，你选谁？

关闭

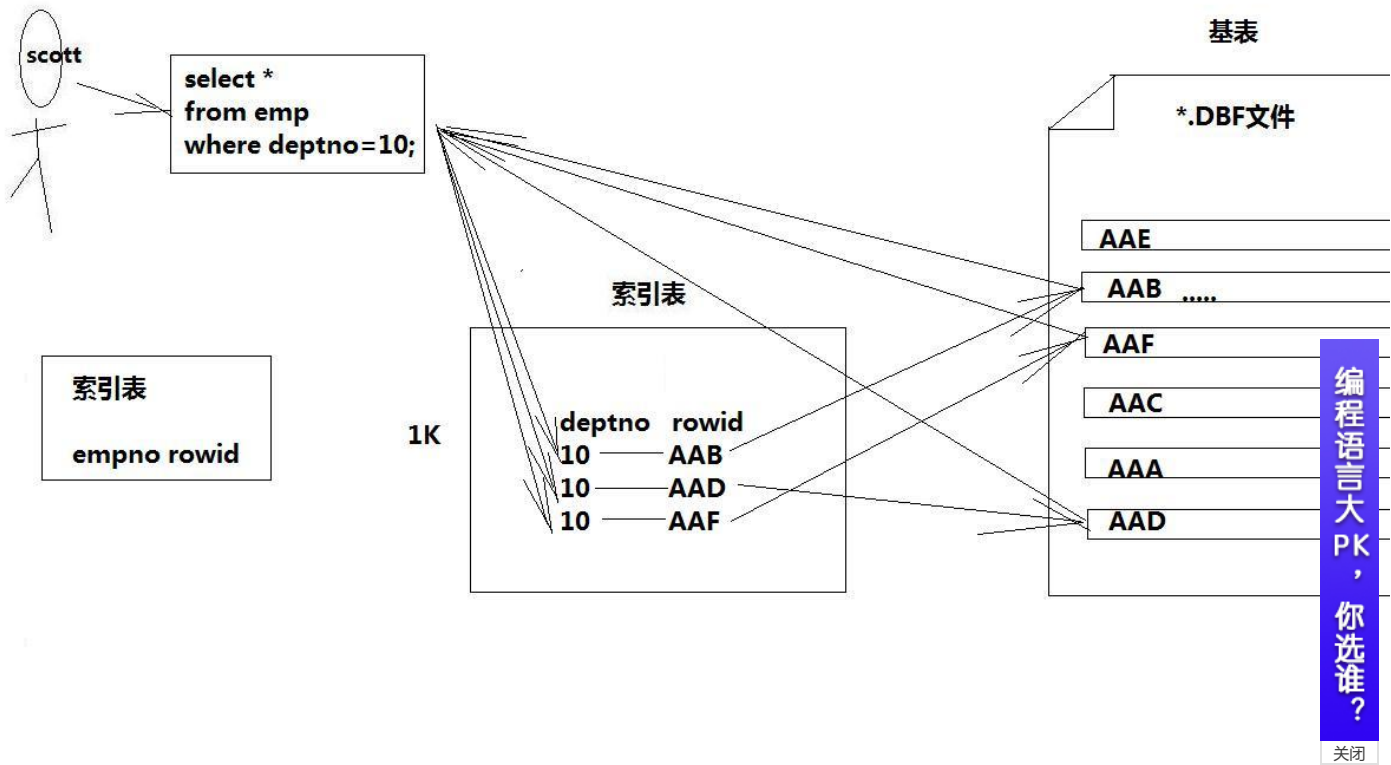


索引表把数据变成是有序的....



<http://blog.csdn.net/3y>

快速定位到硬盘中的数据文件...



<http://blog.csdn.net/3y>

rowid特点

rowid的特点

- (1) 位于每个表中, 但表面上看不见, 例如: desc emp是看不见的
- (2) 只有在select中, 显示写出rowid, 方可看见
- (3) 它与每个表绑定在一起, 表亡, 该表的rowid亡, 二张表rownum可以相同, 但rowid必须是唯一的
- (4) rowid是18位大小写加数字混杂体, 唯一表代该条记录在DBF文件中的位置
- (5) rowid可以参与=/like比较时, 用" 单引号将rowid的值包起来, 且区分大小写
- (6) rowid是联系表与DBF文件的桥梁

索引特点

索引的特点

- (1) 索引一旦建立, * Oracle管理系统会对其进行自动维护*, 而且由Oracle管理系统决定何时使用索引
- (2) 用户不用在查询语句中指定使用哪个索引
- (3) 在定义primary key或unique约束后系统自动在相应的列上创建索引
- (4) 用户也能按自己的需求, 对指定单个字段或多个字段, 添加索引

需要注意的是: Oracle是自动帮我们管理索引的, 并且如果我们指定了primary key或者unique约束, 系统会自动在对应的列上创建索引..

什么时候【要】创建索引

- (1) 表经常进行 SELECT 操作
- (2) 表很大(记录超多), 记录内容分布范围很广
- (3) 列名经常在 WHERE 子句或连接条件中出现

什么时候【不要】创建索引

- (1) 表经常进行 INSERT/UPDATE/DELETE 操作
- (2) 表很小(记录超少)
- (3) 列名不经常作为连接条件或出现在 WHERE 子句中

索引优缺点:

- 索引加快数据库的检索速度
- 索引降低了插入、删除、修改等维护任务的速度(虽然索引可以提高查询速度, 但是它们也会导致数据库系统更新数据的性能下降, 因为大部分数据更新索引)
- 唯一索引可以确保每一行数据的唯一性, 通过使用索引, 可以在查询的过程中使用优化隐藏器, 提高系统的性能
- 索引需要占物理和数据空间

索引分类:

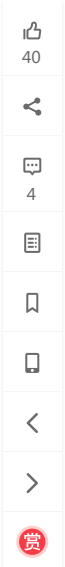
- **唯一索引**: 唯一索引不允许两行具有相同的索引值
- **主键索引**: 为表定义一个主键将自动创建主键索引, 主键索引是唯一索引的特殊类型。主键索引要求主键中的每个值是唯一的, 并且不能为NULL
- **聚集索引(Clustered)**: 表中各行的物理顺序与键值的逻辑(索引)顺序相同, 每个表只能有一个
- **非聚集索引(Non-clustered)**: 非聚集索引指定表的逻辑顺序。数据存储在另一个位置, 索引存储在另一个位置, 索引中包含指向数据存储位置的指针, 小于249个

深入理解索引可参考:

- <https://kb.cnblogs.com/page/45712/>
- <https://www.cnblogs.com/drizzlewithwind/p/5707058.html>

什么是事务?

什么是事务?



事务简单来说：一个Session中所进行所有的操作，要么同时成功，要么同时失败

ACID — 数据库事务正确执行的四个基本要素

- 包含：原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation)、持久性 (Durability)。

一个支持事务 (Transaction) 中的数据库系统，必需要具有这四种特性，否则在事务过程 (Transaction processing) 当中无法保证数据的一致性，数据可能达不到交易。

举个例子:A向B转账，转账这个流程中如果出现问题，事务可以让数据恢复成原来一样【A账户的钱没变，B账户的钱也没变】。

事例说明：

```
1
2
3
4      /*
5      * 我们来模拟A向B账号转账的场景
6      *  A和B账户都有1000块，现在我让A账户向B账号转500块钱
7      *
8      */
9
10     //JDBC默认的情况下是关闭事务的，下面我们看看关闭事务去操作转账操作有什么问题
11
12     //A账户减去500块
13     String sql = "UPDATE a SET money=money-500 ";
14     preparedStatement = connection.prepareStatement(sql);
15     preparedStatement.executeUpdate();
16
17     //B账户多了500块
18     String sql2 = "UPDATE b SET money=money+500";
19     preparedStatement = connection.prepareStatement(sql2);
20     preparedStatement.executeUpdate();
```

从上面看，我们的确可以发现A向B转账，成功了。可是如果A向B转账的过程中出现了问题呢？下面模拟一下

```
1
2     //A账户减去500块
3     String sql = "UPDATE a SET money=money-500 ";
4     preparedStatement = connection.prepareStatement(sql);
5     preparedStatement.executeUpdate();
6
7     //这里模拟出现问题
8     int a = 3 / 0;
9
10
11     String sql2 = "UPDATE b SET money=money+500";
12     preparedStatement = connection.prepareStatement(sql2);
13     preparedStatement.executeUpdate();
14
```

显然，上面代码是会抛出异常的，我们再来查询一下数据。A账户少了500块钱，B账户的钱没有增加。这明显是不合理的。

我们可以通过事务来解决上面出现的问题

```
1
2     //开启事务,对数据的操作就不会立即生效。
3     connection.setAutoCommit(false);
4
5     //A账户减去500块
6     String sql = "UPDATE a SET money=money-500 ";
7     preparedStatement = connection.prepareStatement(sql);
8     preparedStatement.executeUpdate();
9
10    //在转账过程中出现问题
11    int a = 3 / 0;
12
13    //B账户多500块
14    String sql2 = "UPDATE b SET money=money+500";
15    preparedStatement = connection.prepareStatement(sql2);
```



40



4



数据一致性

编程语言大PK，你选谁？

关闭



```
16         preparedStatement.executeUpdate();
17
18         //如果程序能执行到这里，没有抛出异常，我们就提交数据
19         connection.commit();
20
21         //关闭事务【自动提交】
22         connection.setAutoCommit(true);
23
24
25     } catch (SQLException e) {
26         try {
27             //如果出现了异常，就会进到这里来，我们就把事务回滚【将数据变成原来那样】
28             connection.rollback();
29
30             //关闭事务【自动提交】
31             connection.setAutoCommit(true);
32         } catch (SQLException e1) {
33             e1.printStackTrace();
34         }
35     }
```

♡40

🔗

💬4

📖

🔖

📱

⏪

⏩

👍

上面的程序也一样抛出了异常，A账户钱没有减少，B账户的钱也没有增加。

注意：当Connection遇到一个未处理的SQLException时，系统会非正常退出，事务也会自动回滚，但如果程序捕获到了异常，是需要在catch中显式

事务隔离级别

数据库定义了4个隔离级别：

- 1. Serializable【可避免脏读，不可重复读，虚读】
- 2. Repeatable read【可避免脏读，不可重复读】
- 3. Read committed【可避免脏读】
- 4. Read uncommitted【级别最低，什么都避免不了】

分别对应Connection类中的4个常量

- 1. TRANSACTION_READ_UNCOMMITTED
- 2. TRANSACTION_READ_COMMITTED
- 3. TRANSACTION_REPEATABLE_READ
- 4. TRANSACTION_SERIALIZABLE

脏读：一个事务读取到另外一个事务未提交的数据

例子：A向B转账，A执行了转账语句，但A还没有提交事务，B读取数据，发现自己账户钱变多了！B跟A说，我已经收到钱了。A回滚事务【rollback】账户的钱时，发现钱并没有多。

不可重复读：一个事务读取到另外一个事务已经提交的数据，也就是说一个事务可以看到其他事务所做的修改

注：A查询数据库得到数据，B去修改数据库的数据，导致A多次查询数据库的结果都不一样【危害：A每次查询的结果都是受B的影响的，那么就没有意思了】

虚读(幻读)：是指在一个事务内读取到了别的事务插入的数据，导致前后读取不一致。

注：和不可重复读类似，但虚读(幻读)会读到其他事务的插入的数据，导致前后读取不一致

简单总结：脏读是不可容忍的，不可重复读和虚读在一定的情况下是可以的【做统计的肯定就不行】。

数据库的乐观锁和悲观锁是什么？

数据库的乐观锁和悲观锁是什么？

编程语言大PK，你选谁？

关闭

🔊

🔔

确保在多个事务同时存取数据库中同一数据时不破坏事务的隔离性和统一性以及数据库的统一性，**乐观锁和悲观锁是并发控制主要采用的技术手段。**

- 悲观锁：假定会发生并发冲突，屏蔽一切可能违反数据完整性的操作
 - **在查询完数据的时候就把事务锁起来，直到提交事务**
 - 实现方式：使用数据库中的锁机制
- 乐观锁：假设不会发生并发冲突，只在提交操作时检查是否违反数据完整性。
 - **在修改数据的时候把事务锁起来，通过version的方式来进行锁定**
 - 实现方式：使用version版本或者时间戳

悲观锁：

1.悲观锁

```
select * from eb_sku where sku_id = 1001 for udapte

update eb_sku set stock = 100 - #{quantity} where sku_id = 1001

缺点：性能低
```

乐观锁：

1.乐观锁

	sku_id	stock	version
a(查询的数据)	1001	100	1
	1001	98	2
b(查询的数据)	1001	100	1

```
select * from eb_sku where sku_id = 1001

update eb_sku set stock = 100 - #{quantity}, version = #{version} +1 where sku_id = #{skuld} and version = #{version} and stock >= #{quantity}
```

参考资料：

- <http://www.open-open.com/lib/view/open1452046967245.html>

超键、候选键、主键、外键分别是什么？

超键、候选键、主键、外键分别是什么？

- | | | |
|--|----------------------------|-----|
| <ul style="list-style-type: none">超键：在关系中能唯一标识元组的属性集称为关系模式的超键。一个属性可以为作为一个超键，多个属性组合在一起也可以作为一个键。候选键(候选码)：是最小超键，即没有冗余元素的超键。主键(主码)：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。一个数据列只能有一个主键，且主键的取值不能取空值（Null）。外键：在一个表中存在的另一个表的主键称此表的外键。 | <div>40</div> <div>4</div> | 超键包 |
|--|----------------------------|-----|

候选码和主码：

例子：邮寄地址（城市名，街道名，邮政编码，单位名，收件人）

- 它有两个候选键:{城市名, 街道名} 和 {街道名, 邮政编码}
- 如果我选取{城市名, 街道名}作为唯一标识实体的属性, 那么{城市名, 街道名} 就是主码(主键)

SQL 约束有哪几种？

SQL 约束有哪几种？

- NOT NULL: 用于控制字段的内容一定不能为空 (NULL)。
- UNIQUE: 控件字段内容不能重复，一个表允许有多个 Unique 约束。
- PRIMARY KEY: 也是用于控件字段内容不能重复，但它在表只允许出现一个。
- FOREIGN KEY: 用于预防破坏表之间连接的动作，也能防止非法数据插入外键列，因为它必须是它指向的那个表中的值之一。
- CHECK: 用于控制字段的值范围。

数据库运行于哪种状态下可以防止数据的丢失？

数据库运行于哪种状态下可以防止数据的丢失？

在archivelog mode(归档模式)只要其归档日志文件不丢失，就可以有效地防止数据丢失。

Mysql存储引擎

Mysql的存储引擎有以下几种：

我的是5.7.15版本，默认使用的是InnoDB版本！

常用的存储引擎有以下：

- **Innodb引擎**, Innodb引擎提供了对数据库ACID事务的支持。并且还提供了行级锁和外键的约束。它的设计的目标就是处理大数据容量的
- **MyIASM引擎**(原本MySQL的默认引擎),不提供事务的支持,也不支持行级锁和外键。
- **MEMORY引擎**:所有的数据都在内存中,数据的处理速度快,但是安全性不高。

同一个数据库也可以使用多种存储引擎的表。如果一个表修改要求比较高的事务处理，可以选择InnoDB。这个数据库中可以将查询要求比较高的表使用MyISAM引擎存储。如果该数据库需要一个用于查询的临时表，可以选择MEMORY存储引擎。

参考资料：

- <https://www.cnblogs.com/xiaohaillong/p/6079551.html>
- <http://blog.csdn.net/l5718/article/details/52248040>
- <http://blog.csdn.net/t146lla128xx0x/article/details/78737290>

MyIASM和Innodb两种引擎所使用的索引的数据结构是什么？

MyIASM和Innodb两种引擎所使用的索引的数据结构是什么？

40

答案:都是B+树!

MyIASM引擎，B+树的数据结构中存储的内容实际上是实际数据的地址值。也就是说它的索引和实际数据是分开的，**只不过使用索引指向实际数据。模式被称为非聚集索引。**

Innodb引擎的索引的数据结构也是B+树，**只不过数据结构中存储的都是实际的数据，这种索引有被称为聚集索引。**

varchar和char的区别

varchar和char的区别

Char是一种固定长度的类型，varchar是一种可变长度的类型

mysql有关权限的表都有哪几个

mysql有关权限的表都有哪几个

MySQL服务器通过权限表来控制用户对数据库的访问，权限表存放在mysql数据库里，由mysql_install_db脚本初始化。这些权限表分别user，db，tables_priv和host。下面分别介绍一下这些表的结构和内容：

- user权限表：记录允许连接到服务器的用户帐号信息，里面的权限是全局级的。
- db权限表：记录各个帐号在各个数据库上的操作权限。
- tables_priv权限表：记录数据表级的操作权限。
- columns_priv权限表：记录数据列级的操作权限。
- host权限表：配合db权限表对给定主机上数据库级操作权限作更细致的控制。这个权限表不受GRANT和REVOKE语句的影响。

数据表损坏的修复方式有哪些？

数据表损坏的修复方式有哪些？

使用 myisamchk 来修复，具体步骤：

- 1) 修复前将mysql服务停止。
- 2) 打开命令行方式，然后进入到mysql的/bin目录。
- 3) 执行myisamchk -recover 数据库所在路径/*.MYI

使用repair table 或者 OPTIMIZE table命令来修复，REPAIR TABLE table_name 修复表 OPTIMIZE TABLE table_name 优化表 REPAIR TABLE table_name 修复并优化表。

OPTIMIZE TABLE 用于回收闲置的数据库空间，当表上的数据行被删除时，所占据的磁盘空间并没有立即被回收，使用了OPTIMIZE TABLE命令，则会对表进行碎片整理，回收空间，并且对磁盘上的数据进行重排（注意：是磁盘上，而非数据库）

MySQL中InnoDB引擎的行锁是通过加在什么上完成

MySQL中InnoDB引擎的行锁是通过加在什么上完成

InnoDB是基于索引来完成行锁

例: `select * from tab_with_index where id = 1 for update ;`

`for update` 可以根据条件来完成行锁锁定,并且 `id` 是有索引键的列,

如果 `id` 不是索引键那么InnoDB将完成表锁,,并发将无从谈起

数据库优化的思路

SQL优化

在我们书写SQL语句的时候，其实书写的顺序、策略会影响到SQL的性能，虽然实现的功能是一样的，但是它们的性能会有些许差别。

因此，下面就讲解在书写SQL的时候，怎么写比较好。

①选择最有效率的表名顺序

数据库的解析器按照从右到左的顺序处理FROM子句中的表名，FROM子句中写在最后的表将被最先处理

在FROM子句中包含多个表的情况下：

- 如果三个表是完全无关系的话，将记录和列名最少的表，写在最后，然后依次类推
- 也就是说：选择记录条数最少的表放在最后

如果有3个以上的表连接查询：

- 如果三个表是有关联的话，将引用最多的表，放在最后，然后依次类推。
- 也就是说：被其他表所引用的表放在最后

例如：查询员工的编号，姓名，工资，工资等级，部门名

emp表被引用得最多，记录数也是最多，因此放在form字句的最后面

```
1
2 select emp.empno,emp.ename,emp.sal,salgrade.grade,dept.dname
3 from salgrade,dept,emp
4 where (emp.deptno = dept.deptno) and (emp.sal between salgrade.losal and salgrade.hisal)
5
6
```

②WHERE子句中的连接顺序

数据库采用自右而左的顺序解析WHERE子句,根据这个原理,表之间的连接必须写在其他WHERE条件之左,那些可以过滤掉最大数量记录的条件必须写在最右。

emp.sal可以过滤多条记录，写在WHERE字句的最右边

```
1
2 select emp.empno,emp.ename,emp.sal,dept.dname
3 from dept,emp
4 where (emp.deptno = dept.deptno) and (emp.sal > 1500)
```

③SELECT子句中避免使用*号

我们当时学习的时候，“*”号是可以获取表中全部的字段数据的。

- 但是它要通过查询数据字典完成的，这意味着将耗费更多的时间
- 使用*号写出来的SQL语句也不够直观。

④用TRUNCATE替代DELETE

这里仅仅是：删除表的全部记录，除了表结构才这样做。

DELETE是一条一条记录的删除，而Truncate是将整个表删除，保留表结构，这样比DELETE快

⑤多使用内部函数提高SQL效率

例如使用mysql的concat()函数会比使用||来进行拼接快，因为concat()函数已经被mysql优化过了。

⑥使用表或列的别名

如果表或列的名称太长了，使用一些简短的别名也能稍微提高一些SQL的性能。毕竟要扫描的字符长度就变少了。。。

⑦多使用commit

comiit会释放回滚点...

40

4

4

4

4

4

4

4

4

编程语言大PK，你选谁？

关闭

4

4

- https://mp.weixin.qq.com/s?__biz=MzI4Njc5NjM1NQ==&mid=2247483696&idx=1&sn=5f472ce7720aede89e2e15ea64bed1bc&chksm=ebd63e1cdca1b70ad18db2cbe11f9ce7b0633980c78a28bd5b1b57c4efbe7a3411&scene=21#wechat_redirect

基本表结构：

```
1 student(sno,sname,sage,ssex) 学生表
2 course(cno,cname,tno) 课程表
3 sc(sno,cno,score) 成绩表
4
5 teacher(tno,tname) 教师表
6
```

题目：

```
1
2
3 101, 查询课程1的成绩比课程2的成绩高的所有学生的学号
4 select a.sno from
5 (select sno,score from sc where cno=1) a,
6 (select sno,score from sc where cno=2) b
7 where a.score>b.score and a.sno=b.sno
8
9
10
11 102, 查询平均成绩大于60分的同学的学号和平均成绩
12 select a.sno as "学号", avg(a.score) as "平均成绩"
13 from
14 (select sno,score from sc) a
15 group by sno having avg(a.score)>60
16
17
18
19
20 103, 查询所有同学的学号、姓名、选课数、总成绩
21 select a.sno as 学号, b.sname as 姓名,
22 count(a.cno) as 选课数, sum(a.score) as 总成绩
23 from sc a, student b
24 where a.sno = b.sno
25 group by a.sno, b.sname
26
27 或者：
28
29 select student.sno as 学号, student.sname as 姓名,
30 count(sc.cno) as 选课数, sum(score) as 总成绩
31 from student left Outer join sc on student.sno = sc.sno
32 group by student.sno, sname
33
34 104, 查询姓“张”的老师的人数
35
36 select count(distinct(tname)) from teacher where tname like '张%'
37 或者：
38 select tname as "姓名", count(distinct(tname)) as "人数"
39 from teacher
40 where tname like '张%'
41 group by tname
42
43
44
45
46 105, 查询没学过“张三”老师课的同学的学号、姓名
47 select student.sno, student.sname from student
48 where sno not in (select distinct(sc.sno) from sc, course, teacher
49 where sc.cno=course.cno and teacher.tno=course.tno and teacher.tname='张三')
50
51
52
53 106, 查询同时学过课程1和课程2的同学的学号、姓名
54 select sno, sname from student
55 where sno in (select sno from sc where sc.cno = 1)
```



编程语言大PK，你选谁？

关闭



```
56 and sno in (select sno from sc where sc.cno = 2)
57 或者 :
58
59 selectc.sno, c.sname from
60 (select sno from sc where sc.cno = 1) a,
61 (select sno from sc where sc.cno = 2) b,
62 student c
63 where a.sno = b.sno and a.sno = c.sno
64 或者 :
65
66 select student.sno,student.sname from student,sc where student.sno=sc.sno and sc.cno=1
67 and exists( select * from sc as sc_2 where sc_2.sno=sc.sno and sc_2.cno=2)
68
69
70
71
72 107, 查询学过“李四”老师所教所有课程的所有同学的学号、姓名
73 select a.sno, a.sname from student a, sc b
74 where a.sno = b.sno and b.cno in
75 (select c.cno from course c, teacher d where c.tno = d.tno and d.tname = '李四')
76
77 或者 :
78
79 select a.sno, a.sname from student a, sc b,
80 (select c.cno from course c, teacher d where c.tno = d.tno and d.tname = '李四') e
81 where a.sno = b.sno and b.cno = e.cno
82
83
84
85 108, 查询课程编号1的成绩比课程编号2的成绩高的所有同学的学号、姓名
86 select a.sno, a.sname from student a,
87 (select sno, score from sc where cno = 1) b,
88 (select sno, score from sc where cno = 2) c
89 where b.score > c.score and b.sno = c.sno and a.sno = b.sno
90
91
92
93
94 109, 查询所有课程成绩小于60分的同学的学号、姓名
95 select sno,sname from student
96 where sno not in (select distinct sno from sc where score > 60)
97
98
99
100 110, 查询至少有一门课程与学号为1的同学所学课程相同的同学的学号和姓名
101 select distinct a.sno, a.sname
102 from student a, sc b
103 where a.sno <> 1 and a.sno=b.sno and
104 b.cno in (select cno from sc where sno = 1)
105
106 或者 :
107
108 select s.sno,s.sname
109 from student s,
110 (select sc.sno
111 from sc
112 where sc.cno in (select sc1.cno from sc sc1 where sc1.sno=1)and sc.sno<>1
113 group by sc.sno)r1
114 where r1.sno=s.sno
115
116
117 111、把“sc”表中“王五”所教课的成绩都更改为此课程的平均成绩
118 update sc set score = (select avg(sc_2.score) from sc sc_2 wheresc_2.cno=sc.cno)
119 from course,teacher where course.cno=sc.cno and course.tno=teacher.tno andteacher.tname='王五'
120
121
122 112、查询和编号为2的同学学习的课程完全相同的其他同学学号和姓名
123 这一题分两步查 :
124
125 1 ,
126
```



40



4



赏

编程语言大PK，你选谁？

关闭



```

127 select sno
128 from sc
129 where sno <> 2
130 group by sno
131 having sum(cno) = (select sum(cno) from sc where sno = 2)
132
133 2 ,
134 select b.sno, b.sname
135 from sc a, student b
136 where b.sno <> 2 and a.sno = b.sno
137 group by b.sno, b.sname
138 having sum(cno) = (select sum(cno) from sc where sno = 2)
139
140 113、删除学习“王五”老师课的sc表记录
141 delete sc from course, teacher
142 where course.cno = sc.cno and course.tno = teacher.tno and tname = '王五'
143
144 114、向sc表中插入一些记录，这些记录要求符合以下条件：
145 将没有课程3成绩同学的该成绩补齐，其成绩取所有学生的课程2的平均成绩
146 insert sc select sno, 3, (select avg(score) from sc where cno = 2)
147 from student
148 where sno not in (select sno from sc where cno = 3)
149
150 115、按平均分从高到低显示所有学生的如下统计报表：
151 -- 学号,企业管理,马克思,UML,数据库,物理,课程数,平均分
152 select sno as 学号
153 ,max(case when cno = 1 then score end) AS 企业管理
154 ,max(case when cno = 2 then score end) AS 马克思
155 ,max(case when cno = 3 then score end) AS UML
156 ,max(case when cno = 4 then score end) AS 数据库
157 ,max(case when cno = 5 then score end) AS 物理
158 ,count(cno) AS 课程数
159 ,avg(score) AS 平均分
160 FROM sc
161 GROUP by sno
162 ORDER by avg(score) DESC
163
164 116、查询各科成绩最高分和最低分：
165
166 以如下形式显示：课程号，最高分，最低分
167 select cno as 课程号, max(score) as 最高分, min(score) 最低分
168 from sc group by cno
169
170 select course.cno as '课程号'
171 ,MAX(score) as '最高分'
172 ,MIN(score) as '最低分'
173 from sc,course
174 where sc.cno=course.cno
175 group by course.cno
176
177 117、按各科平均成绩从低到高和及格率的百分数从高到低顺序
178 SELECT t.cno AS 课程号,
179 max(course.cname)AS 课程名,
180 isnull(AVG(score),0) AS 平均成绩,
181 100 * SUM(CASE WHEN isnull(score,0)>=60 THEN 1 ELSE 0 END)/count(1) AS 及格率
182 FROM sc t, course
183 where t.cno = course.cno
184 GROUP BY t.cno
185 ORDER BY 及格率 desc
186
187 118、查询如下课程平均成绩和及格率的百分数(用"1行"显示):
188
189 企业管理 ( 001 ) , 马克思 ( 002 ) , UML ( 003 ) , 数据库 ( 004 )
190 select
191 avg(case when cno = 1 then score end) as 平均分1,
192 avg(case when cno = 2 then score end) as 平均分2,
193 avg(case when cno = 3 then score end) as 平均分3,
194 avg(case when cno = 4 then score end) as 平均分4,

```



40



4



编程语言大PK，你选谁？

关闭



```

198 100 * sum(case when cno = 1 and score > 60 then 1 else 0 end) / sum(case when cno = 1 then 1 else 0 end) as 及格率1,
199 100 * sum(case when cno = 2 and score > 60 then 1 else 0 end) / sum(case when cno = 2 then 1 else 0 end) as 及格率2,
200 100 * sum(case when cno = 3 and score > 60 then 1 else 0 end) / sum(case when cno = 3 then 1 else 0 end) as 及格率3,
201 100 * sum(case when cno = 4 and score > 60 then 1 else 0 end) / sum(case when cno = 4 then 1 else 0 end) as 及格率4
202 from sc
203
204 119、查询不同老师所教不同课程平均分, 从高到低显示
205 select max(c.tname) as 教师, max(b.cname) 课程, avg(a.score) 平均分
206 from sc a, course b, teacher c
207 where a.cno = b.cno and b.tno = c.tno
208 group by a.cno
209 order by 平均分 desc
210 或者 :
211 select r.tname as '教师', r.rname as '课程' , AVG(score) as '平均分'
212 from sc,
213 (select t.tname, c.cno as rcso, c.cname as rname
214 from teacher t, course c
215 where t.tno=c.tno)r
216 where sc.cno=r.rcso
217 group by sc.cno, r.tname, r.rname
218 order by AVG(score) desc
219
220
221 120、查询如下课程成绩均在第3名到第6名之间的学生的成绩 :
222 -- [学生ID],[学生姓名], 企业管理, 马克思, UML, 数据库, 平均成绩
223 select top 6 max(a.sno) 学号, max(b.sname) 姓名,
224 max(case when cno = 1 then score end) as 企业管理,
225 max(case when cno = 2 then score end) as 马克思,
226 max(case when cno = 3 then score end) as UML,
227 max(case when cno = 4 then score end) as 数据库,
228 avg(score) as 平均分
229 from sc a, student b
230 where a.sno not in

```

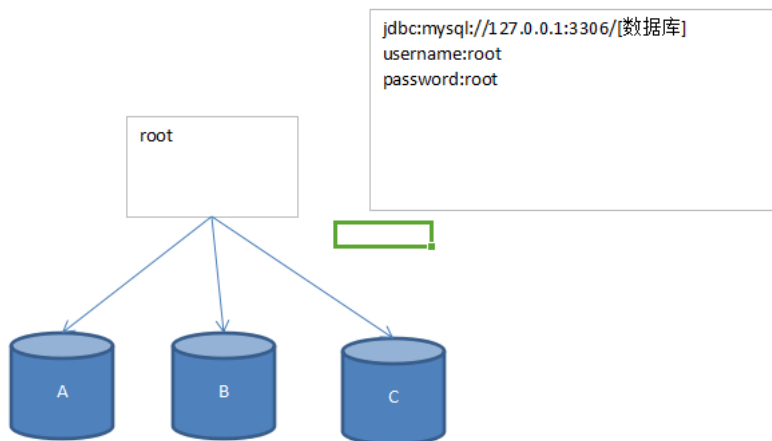
```

(select top 2 sno from sc where cno = 1 order by score desc)
and a.sno not in (select top 2 sno from sc where cno = 2 order by scoredsc)
and a.sno not in (select top 2 sno from sc where cno = 3 order by scoredsc)
and a.sno not in (select top 2 sno from sc where cno = 4 order by scoredsc)
and a.sno = b.sno
group by a.sno

```

Oracle和Mysql的区别

在Mysql中，一个用户下可以创建多个库：



http://blog.csdn.net/hon_3y

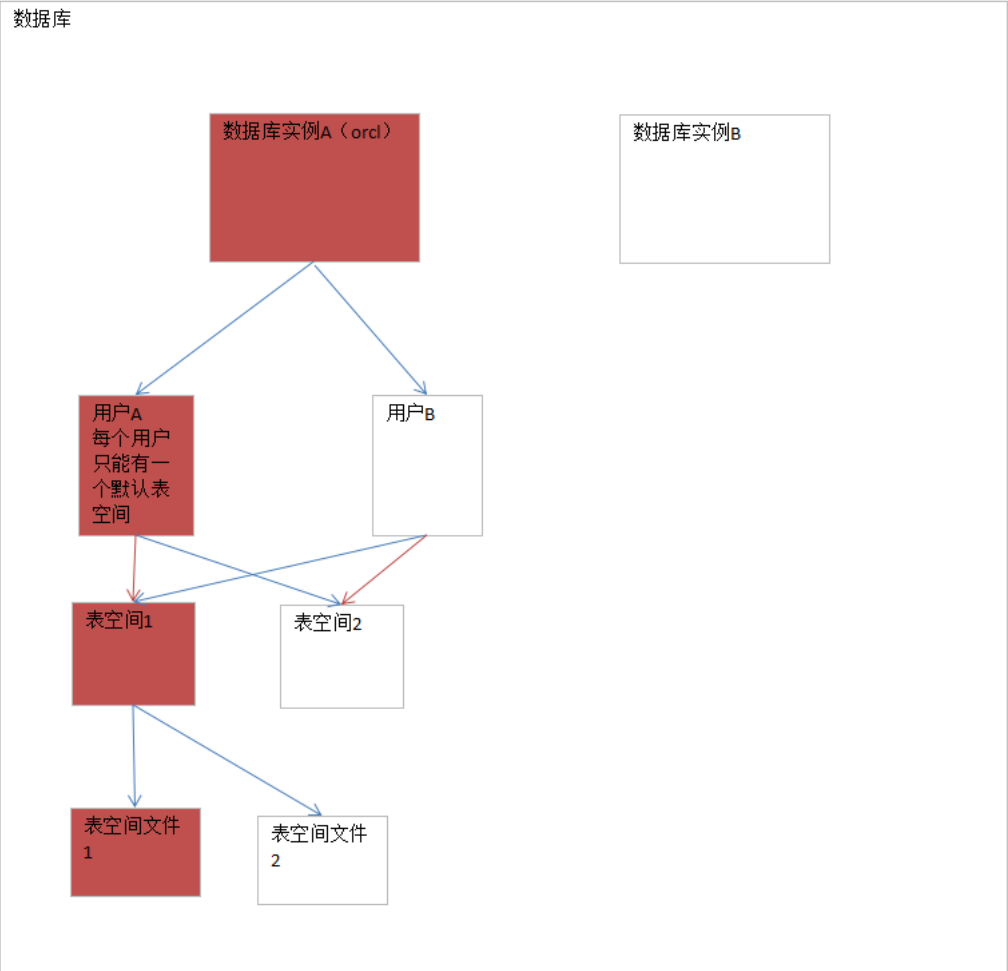
而在Oracle中，Oracle服务器是由两部分组成

- 数据库实例【理解为对象，看不见的】

- 数据库【理解为类，看得见的】

一个数据库实例可拥有多个用户，一个用户默认拥有一个表空间。

表空间是存储我们数据库表的地方，表空间内可以有多个文件。



```
jdbc:oracle:thin@127.0.0.1:1521:orcl
username: scott
password:tiger
```

当我们使用Oracle作为我们数据库时，我们需要指定用户、表空间来存储我们所需要的数据！

最后

参考资料：

- <http://blog.csdn.net/xlgen157387/article/details/46899031>
- http://blog.csdn.net/beauty_1991/article/details/51209107
- <https://zhuanlan.zhihu.com/p/23713529>
- <http://blog.csdn.net/wickedvalley/article/details/51527551>
- <http://blog.csdn.net/zhugewendu/article/details/73550414>

如果文章有错的地方欢迎指正，大家互相交流。习惯在微信看技术文章，想要获取更多的Java资源的同学，可以关注微信公众号:Java3y

有 0 个人打赏

文章最后发布于: 2019-10-10 11:11:11

20道数据库面试题

阅读数 1824

1.什么是封锁？封锁的基本类型有哪几种？含义是什么？所谓封锁就是当一个事务在对某个数据对象（... 博文 来自： 拥之则安的博客



想对作者说点什么

wananya 1年前 #1楼

最后一个mysql个Oracle区别，对象应该是看得见的吧？类应该是看不见的，是不是写反了？

摘要： 本文对面试/笔试过程中经常会被问到的一些关于数据库(MySQL)的问题进行了梳理和总结... 博文 来自： [清新哥哥的博客](#)



数据库面试题(开发者必看) - weixin_33842328的博客 - CSDN博客

巨景·顶新

1、事务槽updatedelete争用事务槽每个块都有数据槽2、100万分区 一亿分区区别...一亿数据量可以考... [博文](#) 来自: [双子星塔](#)












清新哥

114篇

关注

排名:千里之外



	40	
		
	4	查
		
		
		
		
		
		

11-24
下载

编程语言大PK，你选谁？

Hibernate最全面试题 - 3y - CSDN博客

数据库常见面试题(附答案) - CSDN博客

数据库常见的面试题大全

阅读数 7192

点击上方“程序员大咖”，选择“置顶公众号”关键时刻，第一时间送达！来源：java学习<https://mp...> 博文 来自：sfsfsdfsdfds...

数据库常见面试题

阅读数 457

一.数据库的基础知识1.数据库有哪几种范式数据库有三种范式第一范式（1NF）：数据库表中的字段都... 博文 来自：w450093854...

常见面试题整理--数据库篇(每位开发者必备) - weinierz..._CSDN博客

数据库常见面试题(一) - xiongluo0628的博客 - CSDN博客

JAVA面试基础-数据库的存储过程小结

阅读数 1245

1.存储过程是什么？简单来说就是为以后的使用而保存的一条或者多条MySQL语句的集合，可将其视... 博文 来自：weixin_37197...



多商户商城系统v3.0开发

数据库面试3 - ggxxkkll的专栏 - CSDN博客

SQL数据库面试题以及答案（50例题）

阅读数 15万+

SQL数据库面试题以及答案（50例题） 博文 来自：魂-淡

34个数据库常见面试题讲解

阅读数 1万+

1、触发器的作用？触发器是一中特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约... 博文 来自：咖啡的博客

数据库面试题---MySQL

阅读数 1136

1、SQL的select语句完整的执行过程SQLSelect语句完整的执行顺序：1、from子句组装来自不同数据... 博文 来自：彷徨的博客

常见面试题整理--数据库篇（每位开发者必备）

阅读数 8万+

常见面试题整理--数据库篇（每位开发者必备）转载自：<https://zhuanlan.zhihu.com/p/23713529> ... 博文 来自：weinierzui的专...

数据库开发面试问题汇总

阅读数 172

1.索引有哪些类型？就不同的算法结构来说，索引有B+树、有序数组、哈希等；就DDL来说，可为字段... 博文 来自：weixin_42813...

[推动全社会公益氛围形成，使公益与空气和阳光一样触手可及。](#)

公益缺你不可，众多公益项目等你PICK——百度公益 让公益像「空气和阳光」一样触手可及！
gongyi.baidu.com

数据库常见面试题（一）

阅读数 273

本博客内容：一、超键、候选键、主键、外键的区别？一、超键、候选键、主键、外键的区别？参考：... 博文 来自：xiongluo0628...

常见数据库知识总结 + 30个数据库常见面试题讲解

阅读数 5904

常见数据库知识总结MYAQL：事务：事务是并发控制的基本单元，事务是一个操作序列，要么都执行... 博文 来自：Mr.ACO的专栏

数据库开发面试题

阅读数 284

博文 来自：IT小石头

数据库常见笔试题面试题

阅读数 58万+

数据库基础(面试常见问题)一、数据库基础1.数据抽象：物理抽象、概念抽象、视图级抽象,内模式、模式... 博文 来自：Hectorhua的...

数据库面试题(开发者必看)

阅读数 645

数据库常见面试题(开发者篇)什么是存储过程？有哪些优缺点？什么是存储过程？有哪些优缺点？存储... 博文 来自：weixin_34242...



40



4



编程语言大PK，你选谁？

关闭





车险报价电话 网上车险报价

车险网上报价

一个SQL存储过程面试题(比较简单)

阅读数 1万+

三个月前刚毕业的时候，听到存储过程就头疼。写一个SQL存储过程，建立一个表USER字段是姓名，年... 博文 来自： JAVA之迷

经典数据库笔试题面试题

阅读数 8579

经典数据库笔试题面试题一.创建一个学生档案表(表名为xsda)，其表结构如下：列名类型长度允许空备注... 博文 来自： ZhangALin_Bl...

java面试之数据库面试知识点

阅读数 1万+

1.主键超键候选键外键主键：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。... 博文 来自： 嘟哒

今天面试问了数据库设计问题

其他的就不说了 1：有个无线级别的树，功能是快速查询任——一个节点，及其节点任一父树及父以上任一节... 论坛

【面试必过】MySQL数据库高级工程师-面试题-MySQL DBA面试题（共3套）

【面试必过】MySQL数据库高级工程师-面试题-MySQL DBA面试题（共3套） 1.1 1、MySQL 的复制原理... 论坛



冰箱品牌排行榜前十名

冰箱排行榜

数据库优化

阅读数 31

1.主键超键候选键外键主键：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。... 博文 来自： chizhuang688...

数据库常见面试题20道

阅读数 1356

1.MySQL查询时，只有满足联接条件的记录才包含在查询结果，这种联接是（内联接）。内联接：典型... 博文 来自： m569_y的博客

索引的作用？和它的优点缺点是什么？

阅读数 1371

索引的作用？和它的优点缺点是什么？索引就一种特殊的查询表，数据库的搜索引擎可以利用它加速对... 博文 来自： weixin_37657...

c#读蓝牙数据 c# exe反编译成源码 c#流程控制语句 c#调用静态类里的变量 c# csv库 c# int 小端 c# 文件在线管理 c# 精确小数点以为 c#只保留字符串的汉字 .net c# 程序部署

没有更多推荐了，[返回首页](#)

Java3y

私信 关注

TA的个人主页 >

原创

398

粉丝

707

获赞

637

评论

164

等级

博客 5

积分

4373

勋章

访问

18万+

排名

9976

热门文章

外行人都能看懂的SpringCloud，错过了血亏！

阅读数 19339

二叉树就是这么简单

阅读数 18439

编程语言大PK，你选谁？
关闭

数据库面试题(开发者必看)

阅读数 10400

冒泡排序就这么简单

阅读数 9356

购物车【JavaWeb小项目简单版】

阅读数 7657

1 商城系统

2 IT培训机构排名

3 舆情监测平台

4 it外包公司

5 电商系统

6 电商分销系统

7 开源商城系统

8 外卖源码

9 网上商城开发

10 笔记本电脑租赁

11 网站建设商城

12 网上商城

13 多用户商城系统

14 商城源码

15 天猫店转让费

16 程序员月薪

17 雅思封闭式培训

18 小程序开发

19 雅思一对一在线

20 网站建设设计

21 商城系统开发

22 oa办公系统

23 小程序制作

24 雅思英语培训



CSDN学院



CSDN企业招聘

QQ客服

客服论坛

kefu@csdn.net

400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

40

4

编程语言大PK，你选谁？

关闭

https://blog.csdn.net/Java_3y/article/details/79441845

21/21