数据库面试常见的26个问题

1. Sql语言包括哪些类型？

数据定义ddl：create ，alter，drop

数据操作dml：select，insert，update，delete

数据控制dcl：grant授权，revoke解权

grant 权限 on 数据库对象 to 用户

grant select on testdb.\* to common\_user@'%'

1. 内连接和外链接的区别？

内连接是保证两个表中所有的行都要满足连接条件，外连接，某些不满足条件的也会列出来

1)内连接:只连接匹配的行 select A.c1,B.c2 from A join B on A.c3 = B.c3;

2)左外连接:包含左边表的全部行（不管右边的表中是否存在与它们匹配的行）以及右边表中全部匹配的行 select A.c1,B.c2 from A left join B on A.c3 = B.c3;

3)右外连接:包含右边表的全部行（不管左边的表中是否存在与它们匹配的行）以及左边表中全部匹配的行 select A.c1,B.c2 from A right join B on A.c3 = B.c3;

4)全外连接:包含左、右两个表的全部行，不管在另一边的表中是否存在与它们匹配的行 select A.c1,B.c2 from A full join B on A.c3 = B.c3;

1. 什么是存储过程？用什么来调用？

预编译的sql语句，简单说存储过程是为了完成某个数据库中的特定功能而编写的语句集，该语句集包括sql语句（增删改查）、条件语句和循环语句。优点是允许模块化的设计，就是说只需创建一次，以后再该程序中可以调用多次。如果某次操作需要执行多次sql，使用存储过程比单纯sql执行要快，可以用一个命令对象来调用存储过程（CALL）

1. 触发器的作用？

触发器是一种特殊的存储过程，主要是通过实践来触发而被执行的，可以强化约束，维护数据的完整性和一致性，可以跟踪数据库内的操作，从而不允许未经许可的更新和变化。可以级联运算，表上的触发器包含对另一个表的数据操作，而该操作又会导致该表出发器被出发。

1. 索引的作用？它的有点缺点是什么？

索引就是一种特殊的查询表，数据库的搜索引擎可以利用它加速对数据的检索。它很类似于显式生活中的书的目录，不需要查询整本书的内容就能找到想要的数据，索引可以是唯一的，创建索引允许指定单个列或者多个列。缺点是减慢了数据录入的速度，同时增加了数据库尺寸的大小

1. 什么是内存泄漏？

内存泄漏一般指堆内存的泄漏，堆内存是程序从堆中为其分配的，大小任意的，使用完需要显示释放的内存。当应用程序用关键字new 创建对象时，就是从堆中为它分配一块内存，使用完后程序调用free或者delete释放内存，否则就说内存不能被被使用，就是泄漏了，忘记回收的内存

1. 如何维护数据库的完整性和一致性

尽可能使用约束，如check，主键，外键，非空等约束，这样做效率最高，也最方便。其次是使用触发器，这种方法可以保证无论什么业务系统访问数据库都可以保证数据的完整性与一致性。最后考虑自写业务逻辑，但这样做麻烦，效率低下

1. 什么是事务？什么是锁？

事务就是被绑定在一起作为一个逻辑工作单元的sql语句分组，如果任何一个语句操作失败那么这个操作就失败了，以后操作就会回滚到操作前状态，或者是上个节点。为了确保要么执行，要么不执行，就可以使用事务。将一组语句作为事务考虑，就需要通过ACID测试，即原子性、一致性、隔离性、永久性。

锁：在DBMS中，锁是实现事务的关键，锁可以保证事务的完整性和并发性。与现实生活中锁一样，它可以使某些数据的拥有者，在某段时间内不能使用某些数据或数据结构。当然还分级别：共享锁（只读不写），排它锁（可读可写）

1. 事务隔离级别有哪些？

隔离级别：原子性，不可再分性，事务要么全部执行，要么不被执行；一致性或可串性，事务的执行使得数据从一种正确状态转换成另一种正确状态；隔离性，在事务正确提交之前，不允许把事务对数据的任何改变提供给任何其他事务；持久性：事务正确提交后，其结果将永久保存在数据库中，即使在事务提交后有了其他故障，事务的结果也会得到保存

1. 什么是视图？游标是什么？

视图是一种虚拟的表，具有和物理表相同的功能，可以对视图进行增删改查操作，视图通常是一个表或者多个表的行或列的子集。对视图的修改不影响基本表，它使我们获取数据更容易，相对于多表查询

游标：对查询结果集作为一个单元来有效的处理。游标可以定在该单元中的特定行，从结果集的当前行检索一行或多行，可以对结果集当前行进行修改。一般不使用游标，但是需要逐条处理数据的时候，游标显得十分重要。游标用于定位结果集的行，通过判断全局变量@@FETCH\_STATUS可以判断是否到了最后，通常此变量不等于0表示出错或列到了最后。

1. 什么是主键，什么是外键？

主键里的值总是唯一的，外键是一个用来建立两个表格之间关系的约束。

1. 过多索引有什么样的性能考虑？

表格的索引越多，数据库引擎用来更新、插入或者删除数据所需要的时间就越多，因为在数据操纵时，索引也需要维护。

1. 可以用什么方式确保表格里的字段只接受特定范围的值？

Check限制，他在数据库中，限制输入该列的值。

1. 什么是相关子查询？如何使用这些子查询？

相关子查询是一种包含子查询的特殊类型的查询。查询里包含的子查询会真正请求外部查询的值，从而形成一个类似循环的情况。相关子查询不能单独运行，而非相关子查询可以单独运行。

1. 使用索引查询一定能够提高性能？

索引需要空间来存储，也需要定期维护，那些不必要的索引反而会使查询的反应时间变慢，使用索引查询不一定能提高查询性能

1. 查询A(ID,Name)表中第31至40条记录，ID作为主键可能是不连续增长的列， 查询语句

select top 10 \* from A where ID >(select max(ID) from (select top 30 ID from A order by ID ) T) order by ID

1. 分页查询语句?

[mysql](https://www.2cto.com/database/MySQL/)的分页查询: select o.\* from (sql) o limit firstIndex,pageSize;

sqlserver2005的分页查询: select top pageSize o.\* from (select row\_number() over(order by orderColumn) as rownumber,\* from(sql) as o where rownumber>firstIndex;

1. 如何优化数据库，如何提高数据库性能？

①给数据库做索引，合理的索引能立即显著提高数据库系统的性能

②在适当情况下，尽量使用存储过程不是sql查询，前者已经预编译过，速度更快

③优化查询语句，通过高性能的查询语句提高数据库的性能

1. 数据库优化方面的经验

preparedStatement比statement性能高，有外键约束会影响插入和删除性能，如果程序能够保证数据的完整性，尽量去掉外键

子查询语句的效率比关联查询效率高

1. 表与表之间的关联关系

一对一，一对多，多对多

1. 三大范式

第一范式：字段属性单一，不可再分

第二范式：所有非主属性依赖于主属性，订单表，订单id，产品id，产品名称，订单id和产品id可以确定整条记录，可以将订单id和产品id作为主键，但是产品名称只需通过产品id就能确定，属于部分依赖，不符合

第三范式：不存在传递依赖；

学号 姓名 宿舍楼 宿舍楼位置；学号可以决定属于哪个宿舍楼，宿舍楼决定宿舍楼位置，存在传递依赖

# 存储过程

**概念：**是一组为了完成特定功能的sql语句集，经编译后存储在数据库中。

存储过程是数据库的一个重要对象，用户通过指定存储过程的名字并给出参数（如果存储过程带参数）来执行；

存储过程是由流控制和sql语句书写的过程，这个过程经编译和优化后存储在数据库服务器中。存储过程可由应用程序通过一个调用来执行，允许声明变量。同时，存储过程可以接受和输出参数、返回执行存储过程的状态值，可以嵌套使用

**存储过程功能**：

①变量说明

②兼容的sql命令（select，update）

③一般流程控制指令（if…else…、while）

④内部函数

**存储过程的优点**：

①大大增强了sql语言的功能和灵活性

用流控制语句编写，有很强的灵活性

②可保证数据的安全性和完整性

通过存储过程可以使没有权限的用户在控制之下间接的存取数据库，从而保证数据的安全。通过存储过程可以使相关的动作在一起发生，维护数据库的完整性

③在运行存储过程前，数据库已对其进行语法分析，并给出了优化执行方案。

这种编译好的过程极大的改善sql语句的性能。

由于执行sql语句的大部分工作已经完成，所以存储过程能以几块的速度运行。

④可以降低网络的通信量。

客户端调用存储过程只需要传存储过程名和相关参数即可，与传sql语句相比自然数据量少了很多。

⑤使体现企业规则的运算程序放入数据库服务器中，以便：

集中控制，当企业规则发生变化时，在服务器中改变存储过程即可，无修改任何应用程序

**优点概括：**

①存储过程只在创造时进行编译，以后每次执行存储过程都不需要重新编译，提高效率

②复杂操作，可以将其用存储过程封装起来与数据库提供的事务处理结合一起用

③存储过程可以重复使用，可减少开发人员工作量

④安全性高，可设定只有某些用户才能执行存储过程

缺点：

①代码是运行前编译的，所以当引用对象或者代码发生改变时，需要重新进行编译

②如果程序中大量使用存储过程，随着用户需求的增加，导致数据结构的变化，维护需要很大代价。

|  |  |
| --- | --- |
| --创建带output参数的存储过程  CREATE PROCEDURE PR\_Sum  @a int,  @b int,  @sum int output  AS  BEGIN  set @sum=@a+@b  END    --创建Return返回值存储过程  CREATE PROCEDURE PR\_Sum2  @a int,  @b int  AS  BEGIN  Return @a+@b  END    --执行存储过程获取output型返回值  declare @mysum int  execute PR\_Sum **1**,**2**,@mysum output  print @mysum    --执行存储过程获取Return型返回值  declare @mysum2 int  execute @mysum2= PR\_Sum2 **1**,**2**  print @mysum2 |  |