

# 数据库工程师常见面试题

**问题 1:** 为什么 group by 和 order by 会使查询变慢?

答: group by 和 order by 操作通常需要创建一个临时表来处理查询的结果, 所以如果查询结果很多的话会严重影响性能。

**问题 2:** delete、truncate 和 drop 的区别?

答: Delete 命令用来删除表的全部或者一部分数据行, 执行 delete 之后, 用户需要提交(commit)或者回滚(rollback) transaction 来执行删除或者撤销删除, delete 命令会触发这个表上所有的 delete 触发器。

Truncate 删除表中的所有数据, 这个操作不能回滚, 也不会触发这个表上的触发器, truncate 比 delete 更快, 占用的空间更小。

Drop 命令从数据库中删除表, 所有的数据行, 索引和权限也会被删除, 所有的 DML 触发器也不会被触发, 这个命令也不能回滚。

**问题 3:** 简述数据库的设计过程。

答: 数据库设计分为以下五个阶段:

需求分析: 主要是准确收集用户信息需求和处理需求, 并对收集的结果进行整理和分析, 形成需求说明。

概念结构设计: 对用户需求进行综合、归纳、抽象, 形成一个与具体的 DBMS 无关概念模型(一般为 ER 模型)。

逻辑结构设计: 将概念结构设计概念模型转化为某个特定的 DBMS 所支持的数据模型, 建立数据库逻辑模式, 并对其进行优化, 同时为各种用户和应用设计外模式。

物理结构设计: 为设计好的逻辑模型选择物理结构, 包括存储结构和存取方法等, 建立数据库物理模式。

实施和维护: 实施就是使用 DDL 语言建立数据库模式, 将实际数据载入数据库, 建立真正的数据库。维护阶段是对运行中的数据库进行评价、调整和修改。

**问题 4:** 插入记录时可以不指定字段名称吗?

答: 不管使用哪种 INSERT 语法, 都必须给出 VALUES 的正确数目。如果不提供字段名, 则必须给每个字段提供一个值, 否则将产生一条错误消息。如果要在 INSERT 操作中省略某些字段, 这些字段需要满足一定条件: 该列定义为允许空值; 或者表定义时给出默认值, 如果不给出值, 将使用默认值。

**问题 5:** 本地索引与全局索引的差别与适用情况。

答: 对于本地索引, 每一个表分区对应一个索引分区, 当表的分区发生变化时, 索引的维护由数据库自动进行。对于全局索引, 可以选择是否分区, 而且索引的分区可以不与表分区相对应。当对分区进行维护操作时, 通常会导致全局索引的 INVALIDED, 必须在执行完操作后 REBUILD。

**问题 6:** 事务和锁有什么关系?

答: 可以使用多种机制来确保数据的完整性, 例如约束、触发器以及本章介绍的事务和锁等。事务和锁的关系非常紧密。事务包含一系列的操作, 这些操作要么全部成功, 要么全部失败, 通过事务机制管理多个事务, 保证事务的一致性, 事务中使用锁保护指定的资源, 防止其他用户修改另外一个还没有完成的事务中的数据。

**问题 7:** 谈谈你对索引的理解?

答: 索引是若干数据行的关键字的列表, 查询数据时, 通过索引中的关键字可以快速定位到要访问的记录所在的数据块, 从而大大减少读取数据块的 I/O 次数, 因此可以显著提高性能。

**问题 8:** 谈谈你对事务的理解?

答: 事务从 COMMIT、ROLLBACK、连接到数据库或开始第一条可执行的 SQL 语句时开始, 到一条 COMMIT、ROLLBACK 语句或退出数据库时结束。如果在一个事务中包含 DDL 语句, 则在 DDL 语句的

前后都会隐含地执行 COMMIT 语句，从而开始或结束一个事务。如果一个事务由于某些故障或者由于用户改变主意而必须在提交前取消它，则数据库被恢复到这些语句和过程执行之前的状态。利用 ROLLBACK 语句可以在 COMMIT 命令前随时撤消或回退一个事务。可以回退整个事务，也可以回退部分事务，但是不能回退一个已经被提交的事务。回退部分事务的 ROLLBACK 命令为：ROLLBACK to savepoint 存储点名,存储点是用户放入事务中的标记，用来表示一个可被回退的位置。存储点通过在事务中放入一个 SAVEPOINT 命令而被插入。该命令的语法是：SAVEPOINT 存储点名,如果在 ROLLBACK 语句中没有给出存储点名，则整个事务被回退。

**问题 9：**如何设置网络数据包的大小？

答：一般情况下，设置网络数据包的大小可参考以下方法。如果应用程序常执行大容量复制操作或者是发送、接收大量的 text 和 image 数据的话，可以将此值设大一点。如果应用程序接收和发送的信息量都很小，那么可以将其设为 512 字节。

**问题 10：**你对游标的理解,游标的分类？

答：游标是结果集数据中的指针，作用是为遍历结果集时，存储每条记录的结果。游标分为显式游标和隐式游标。

**问题 11：**触发器中能否用 COMMIT，为什么？

答：在触发器中不能使用 COMMIT 等事务控制语句。因为触发器是事务触发的如果有事务控制语句就会影响到触发它的事务。即连带触发它的语句之前的已经完成的没有提交的语句都要受到影响。这是会影响到数据的一致性的。

**问题 12：**归档模式与非归档模式?为什么归档模式可以恢复到任意时刻？

答：归档是在重做日志文件被覆盖之前，将该重做日志文件通过复制操作系统文件的方式，保存到指定的位置。保存下来的重做日志文件的集合被称为归档重做日志文件，复制的过程被称为归档。

在归档日志（ARCHIVELOG）模式下，数据库对重做日志文件进行归档操作。非归档日志（NOARCHIVELOG）模式下，不会对重做日志文件进行归档操作。由于非归档模式不会在覆盖之前保存，这样就造成了数据库在一定时间之前的重做日志文件丢失，也就不能恢复到被覆盖之前。而归档模式在任意时间上都有重做日志文件，故可以恢复到任意时刻。

**问题 13：**如何提高日志的切换频率？

答：通过参数 ARCHIVE\_LAG\_TARGET 可以控制日志切换的时间间隔，以秒为单位。通过减少时间间隔，从而实现提高日志的切换频率。例如以下代码：

```
SQL> ALTER SYSTEM SET ARCHIVE_LAG_TARGET=50 SCOPE=both;
```

通过上面的命令，可以实现日志每 50 秒切换一次。

**问题 14：**表中有 A B C 三列，用 SQL 语句实现：当 A 列大于 B 列时选择 A 列否则选择 B 列，当 B 列大于 C 列时选择 B 列否则选择 C 列。

答：可以用以下 SQL 语句实现：

```
select (case when a>b then a else b end ), (case when b>c then b else c end)
from table_name
```

**问题 15：**一个日期判断的 sql 语句，请取出 tb\_send 表中日期(SendTime 字段)为当天的所有记录?(SendTime 字段为 datetime 型，包含日期与时间)？

答：select \* from tb where datediff(dd,SendTime,getdate())=0

**问题 16：**有一张表，里面有 3 个字段：语文，数学，英语。其中有 3 条记录分别表示语文 70 分，数学 80 分，英语 58 分，请用一条 sql 语句查询出这三条记录并按以下条件显示出来（并写出您的思路）：大于或等于 80 表示优秀，大于或等于 60 表示及格，小于 60 分表示不及格。

显示格式：

语文	数学	英语
及格	优秀	不及格

答：可以用以下 SQL 语句实现：

```
select
(case when 语文>=80 then '优秀'
```

```

when 语文>=60 then '及格'
else '不及格') as 语文,
(case when 数学>=80 then '优秀'
when 数学>=60 then '及格'
else '不及格') as 数学,
(case when 英语>=80 then '优秀'
when 英语>=60 then '及格'
else '不及格') as 英语,
from table

```

**问题 17:** 一个表中的 Id 有多个记录, 把所有这个 id 的记录查出来, 并显示共有多少条记录数。

答: 可以用以下 SQL 语句实现:

```

select id, Count (*) from tb group by id having count(*)>1
select * from(select count(ID) as count from table group by ID)T where T.count>1

```

**问题 18:** 存储过程和自定义函数的区别?

答: 存储过程, 功能强大, 可以执行包括修改表等一系列数据库操作, 也可以创建为 SQL Server 启动时自动运行的存储过程。而自定义函数, 用户定义函数不能用于执行一组修改全局数据库状态的操作。

存储过程, 可以使用非确定函数。自定义函数, 不允许在用户定义函数主体中内置非确定函数。

存储过程, 可返回记录集。自定义函数, 可以返回表变量, 也可以有任意个输出参数。

存储过程, 其返回值不能被直接引用, 必须单独调用。自定义函数, 其返回值可以被直接引用, 也就是可以直接 select \* from 函数。

**问题 19:** 视图的作用是什么?

答: 数据库视图的作用主要有:

1. 数据库视图隐藏了数据的复杂性。
2. 数据库视图有利于控制用户对表中某些列的访问。
3. 数据库视图使用户查询变得简单。

视图是一个虚拟表, 其内容由查询定义。同真实的表一样, 视图包含一系列带有名称的列和行数据。但是, 视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表, 并且在引用视图时动态生成。对其中所引用的基础表来说, 视图的作用类似于筛选。定义视图的筛选可以来自当前或其它数据库的一个或多个表, 或者其它视图。分布式查询也可用于定义使用多个异类源数据的视图。如果有几台不同的服务器分别存储组织中不同地区的数据, 而您需要将这些服务器上相似结构的数据组合起来, 这种方式就很有用。通过视图进行查询没有任何限制, 通过它们进行数据修改时的限制也很少。

**问题 20:** 是不是索引建立得越多越好?

答: 合理的索引可以提高查询的速度, 但不是索引越多越好。在执行插入语句的时候, 数据库要为新插入的记录建立索引。所以过多的索引会导致插入操作变慢。原则上是只有查询用的字段才建立索引。

**问题 21:** STR 函数在遇到小数时如何处理?

答: 在使用 STR 函数时, 如果数字为小数, 则在转换为字符串数据类型时, 只返回其整数部分, 如果小数点后的数字大于等于 5, 则四舍五入返回其整数部分。

**问题 22:** 自定义函数支持输出参数吗?

答: 自定义函数可以接受零个或多个输入参数, 其返回值可以是一个数值, 也可以是一个表, 但是自定义函数不支持输出参数。

**问题 23:** 为什么要及时删除不用的触发器?

答: 触发器定义之后, 每次执行触发事件, 都会激活触发器并执行触发器中的语句。如果需求发生变化, 而触发器没有进行相应的改变或者删除, 则触发器仍然会执行旧的语句, 从而会影响新的数据的完整性。因此, 要将不再使用的触发器及时删除。

**问题 24:** 什么是唯一索引?

答:唯一索引可以确保索引列不包含重复的值。在多列唯一索引的情况下,该索引可以确保索引列中每个值组合都是唯一的。例如,如果在 `last_name`、`first_name` 和 `middle_initial` 列的组合上创建了唯一索引 `full_name`,则该表中任何两个人都不可以具有相同的全名。

聚集索引和非聚集索引都可以是唯一的。因此,只要列中的数据是唯一的,就可以在同一个表上创建一个唯一的聚集索引和多个唯一的非聚集索引。

只有当唯一性是数据本身的特征时,指定唯一索引才有意义。如果必须实施唯一性以确保数据的完整性,则应在列上创建 `UNIQUE` 或 `PRIMARY KEY` 约束,而不要创建唯一索引。

**问题 25:** 为什么 `UNION ALL` 比 `UNION` 快?

答:因为在使用 `UNION` 的时候,需要删除重复的记录,但是使用 `UNION ALL` 的时候不用删除。所以如果在知道需要 `UNION` 的查询不可能出现重复数据的时候,一定要使用 `UNION ALL`。