



链滴

mysql 初级面试题

作者: [xiaokedamowang](#)

原文链接: <https://ld246.com/article/1628942856264>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

mysql 初级面试题

一. 行转列

```
CREATE TABLE tb_score(  
  id INT(11) NOT NULL auto_increment,  
  userid VARCHAR(20) NOT NULL COMMENT '用户id',  
  subject VARCHAR(20) COMMENT '科目',  
  score DOUBLE COMMENT '成绩',  
  PRIMARY KEY(id)  
)ENGINE = INNODB DEFAULT CHARSET = utf8;  
INSERT INTO tb_score(userid,subject,score) VALUES ('001','语文',90);  
INSERT INTO tb_score(userid,subject,score) VALUES ('001','数学',92);  
INSERT INTO tb_score(userid,subject,score) VALUES ('001','英语',80);  
INSERT INTO tb_score(userid,subject,score) VALUES ('002','语文',88);  
INSERT INTO tb_score(userid,subject,score) VALUES ('002','数学',90);  
INSERT INTO tb_score(userid,subject,score) VALUES ('002','英语',75.5);  
INSERT INTO tb_score(userid,subject,score) VALUES ('003','语文',70);  
INSERT INTO tb_score(userid,subject,score) VALUES ('003','数学',85);  
INSERT INTO tb_score(userid,subject,score) VALUES ('003','英语',90);  
INSERT INTO tb_score(userid,subject,score) VALUES ('003','政治',82);
```

1. 使用IF 配合聚合函数 SUM或者MAX

```
SELECT userid,  
SUM(IF(`subject`='语文',score,0)) as '语文',  
SUM(IF(`subject`='数学',score,0)) as '数学',  
SUM(IF(`subject`='英语',score,0)) as '英语',  
SUM(IF(`subject`='政治',score,0)) as '政治'  
FROM tb_score  
GROUP BY userid
```

2. 使用case...when....then 配合聚合函数 SUM或者MAX

```
SELECT userid,  
SUM(CASE `subject` WHEN '语文' THEN score ELSE 0 END) as '语文',  
SUM(CASE `subject` WHEN '数学' THEN score ELSE 0 END) as '数学',  
SUM(CASE `subject` WHEN '英语' THEN score ELSE 0 END) as '英语',  
SUM(CASE `subject` WHEN '政治' THEN score ELSE 0 END) as '政治'  
FROM tb_score  
GROUP BY userid
```

3. 网上还有很多 没必要记 应付1下面试就行

二. in和exists

in适合大表驱动小表, exists适合小表驱动大表

A表100W条数据, B表1000条, 使用select * from A where xx in (select xx from B where XXX)

反之: select * from A where xx exists (select 1 from B where A.xx = B.xx)

三. 有哪些存储引擎

顺便回答1下区别

1. MyISAM

不支持事务, 没有聚簇索引, 文件分成3分, 表结构, 索引, 数据

不支持外键

不支持行锁

通过变量保存行的总数

2. InnoDB

支持事务, 支持聚簇索引

支持外键

3. Memory

不支持事务, 在内存中

支持hash索引

4. 还有几个 没意义 记住3个就行

四. 有哪些隔离级别, 分别解决了什么问题

事务隔离级别	脏读	不可重复读	幻读
读未提交 (read-uncommitted)	是	是	是
读已提交 (read-committed)	否	是	是
可重复读 (repeatable-read)	否	否	是
串行化 (serializable)	否	否	否

五. 事务的ACID

- (A) 原子性: 一个事务必须被视为一个不可分割的最小单位, 要不全部提交成功, 要么全部失败回。

由undolog保证原子性

- (C) 一致性: 数据库的完整性约束没有被破坏, 事务执行的前后都是合法的数据状态, 不会部分数据态改变了部分状态没有改变 (网上的其他答案都不像人话, 这是我东拼西凑出来的)

由其他3个特性一起保证

- (I) 隔离性: 通常来说, 一个事务所做的修改在最终提交之前, 对其他事务是不可见的。这个和数据的隔离级别有关, 所以只能通常来说。

MVCC + undolog + 锁

- (D) 持久性: 一旦事务提交, 则其所做的修改会被永久保存到数据库中

redolog

六. char和varchar的区别

char长度固定, char如果插入的长度小于定义长度时, 用空格填充

varchar可变长度, 小于定义长度时, 还是按实际长度存储

七. 说说MVCC

具体可以看这篇博客: [mysql MVCC多版本并发控制](#)

八. 索引及优化

具体可以看这两篇博客: [mysql 索引](#), [mysql explain执行计划](#)

九. 数据库设计三范式

具体可以看这篇博客: [数据库设计三范式](#)

十. undolog,redolog,binlog是什么

1. undolog是用来回滚的, 保证了事务的原子性, 配合MVCC还保证了隔离性
2. redolog是用来保证持久性
3. binlog是用来主从复制, 或者恢复数据用

十一. 数据库有哪些锁

具体可以看这篇博客: [Mysql 各种锁](#)