一、(重点)事务(Transaction)

1. 什么是事务：事务是作为单个逻辑工作单元执行的一系列操作整体向系统提交，要么都执行，要么都不执行.

2. 事务的特性:

* **原子性（Atomicity）**

**事务是一个完整的操作，事务的各步操作是不可分的(原子的)，要么都执行，要么都不执行**

* **一致性（Consistency）**

**当事务完成时，数据必须处于一致状态**

* **隔离性（Isolation）**

**并发事务之间彼此隔离、独立，它不应以任何方式依赖于或影响其他事务**

* **持久性（Durability）**

**事务完成后,它对数据库的修改被永久保持**

3. 显示事务语句

开始事务:begin 或者 start transaction

提交事务:commit

回滚事务:rollback

|  |
| --- |
| 示例一:基于事务实现转账(模拟都成功)  START TRANSACTION ;-- 开始事务  -- 第一步:张三减100  UPDATE bank\_custom SET money=money-100 WHERE NAME='张三'; -- 700  -- 第二步:李四加100  UPDATE bank\_custom SET money=money+100 WHERE NAME='李四'; -- 300  -- 提交  COMMIT; |

|  |
| --- |
| 示例二:基于事务实现转账(模拟不成功)  START TRANSACTION ;-- 开始事务  -- 第一步:张三减100  UPDATE bank\_custom SET money=money-100 WHERE NAME='张三'; -- 600  -- 第二步:李四加100  UPDATE bank\_custom1 SET money=money+100 WHERE NAME='李四'; -- 300  -- 回滚  ROLLBACK; |

4.扩展:隐式事务::每个sql语句都是基于事务执行，成功提交，失败回滚

-- 隐式事务默认是开启的.会自动提交回滚.

SELECT \* FROM students

-- 关闭隐式事务 需手动提交或者回滚

SET autocommit=0;

DELETE FROM students WHERE xh=109;

ROLLBACK;

SET autocommit=1; -- 开启隐式事务

二、视图(VIEW)

1. 什么是视图

视图是一张虚拟表，表示一张表的部分数据或多张表的综合数据，基结构和数所是建立在对表的查询基础.

注意:视图不保存数据(数据在原始表中)，保存的是查询基础.

2. 视图的作用:

为不同的数据创建不同的视图，防止访问敏感数据.

降低数据库的复杂程度

3. 创建视图的语法

Create view 视图名称

As

查询语句

4. 查询视图

Select \* from 视图名 [where 条件]

5. 删除视图

DROP VIEW 视图名称;

DROP VIEW `my\_view`;

6. 创建视图的注意事项

* **视图中可以使用多个表**
* **一个视图可以嵌套另一个视图**
* **对视图数据进行添加、更新和删除操作直接影响所引用表中的数据**
* **当视图数据来自多个表时，不允许添加和删除数据**

**注意:**

**使用视图修改数据会有许多限制，一般在实际开发中视图仅用作查询**

示例:

|  |
| --- |
| -- 示例三:创建视图  -- 班主任:关注姓名，性别，地址，状态  CREATE VIEW my\_view  AS  SELECT NAME,sex,address,state FROM students ;  -- 使用视图(查询视图)  SELECT \* FROM my\_view  SELECT NAME,sex FROM my\_view WHERE state=1 |

|  |
| --- |
| -- 示例四:创建视图  -- 教员:关注姓名，性别，地址，年级名称  CREATE VIEW my\_view2  AS  SELECT NAME,sex,address,gname FROM students LEFT JOIN grade ON students.gid=grade.gid;  -- 使用视图(查询视图)  SELECT \* FROM my\_view2; |

三、(理解)索引

1. 什么是索引:

索引是一种有效组合数据的方式，为快速查找到指定记录。

2.作用:

提高查询的效率，改善数据库性能.

3.常见索引类型(了解)



4.创建索引的语法

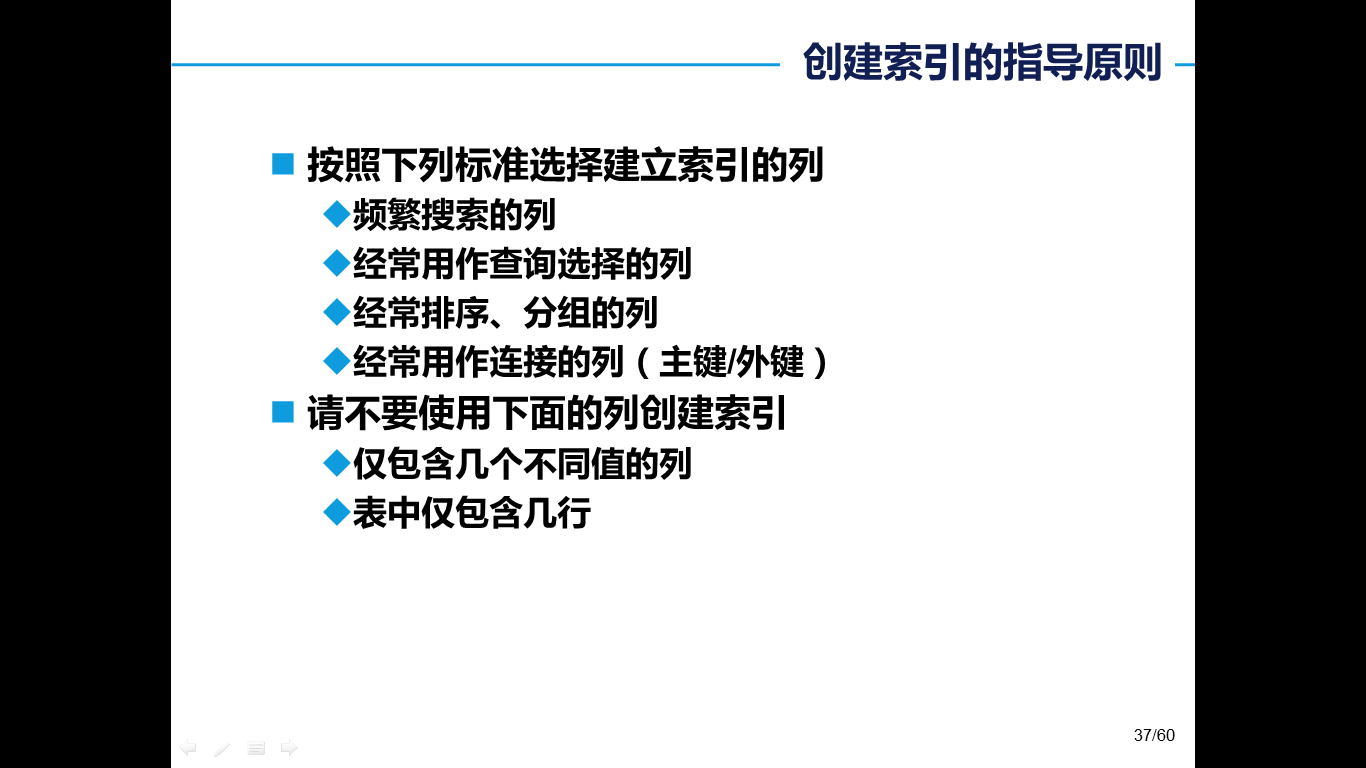
Create 索引类型 index 索引名称 on 表名（列名）

索引类型有:unique 唯一索引、fulltext 全文索引、spatial 空间索引

5. 删除索引

Drop index 索引名称 on 表名;

5.创建索引的原则:



示例:

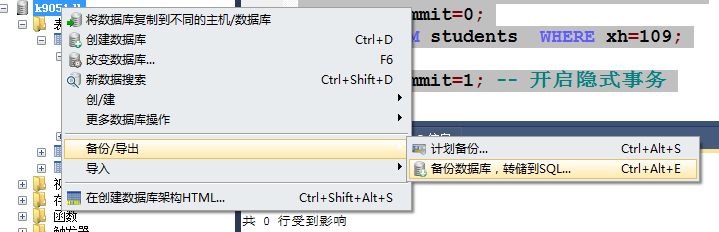
DROP INDEX my\_index ON students;

示例:

|  |
| --- |
| 创建索引  -- 示例:在学生表的姓名列上添加唯一索引  CREATE UNIQUE INDEX my\_index ON students(NAME); |

四、备份数据和还原数据库

1. 备份数据库:导出成sql文件



2.基于sql文件还原数据库

运行sql即可

了解:面试题:

如何提高查询的效率

答:分库分表、索引、缓存、sql优化(高质量的sql)