# select a.skyDragonTimes from player a inner join skydragon b on a.userId=b.userId where a.account="mr1";

# SHOW CREATE DATABASE/TABLES 查看建库或者建表语句

# insert playerprop(userid,help,behelp) values (1,1,1); 插入数据

# alter table 原表名 rename to 新表名

# alter table 表名 change 要修改的字段名 新字段名 新字段的数据类型

# alter table playerprop character set utf8mb4 COLLATE utf8mb4\_general\_ci; 修改表的编码格式

# show columns from 查看所有表头信息

# select distinct name数据去重

# SELECT COUNT(\*) FROM table\_name 数据数量

# truncate table wp\_comments;

# delete \* from wp\_comments;

# drop table table\_name;

# offset：偏移量，和limit一起用做到分页的效果。select \* from test1 limit 5 offset 1;

# union：俩表数据去重。union all输出所有。select \* from test1 union select \* from test2;

# group by：分组查询和去重查询（distinct）。可以跟（with rollup 总统计）

# SELECT @@tx\_isolation：查看mysql事务隔离级别。

# SELECT count(TABLE\_NAME) FROM information\_schema.TABLES WHERE TABLE\_SCHEMA='dbname';查看某个数据库有多少表

# 关系型数据库的优势：

## 保持数据的一致性（事务处理）

## 由于以标准化为前提，数据更新的开销很小（相同的字段基本上都只有一处）

## 可以进行Join等复杂查询

## 其中能够保持数据的一致性是关系型数据库的最大优势。

# 关系型数据库的不足：

## 不擅长的处理大量数据的写入处理

## 为有数据更新的表做索引或表结构（schema）变更

## 字段不固定时应用

## 对简单查询需要快速返回结果的处理。

# 索引有什么作用

## 对数据库表中一列或多列的值进行排序的一种结构

## 第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。

## 第二，可以大大加快数据的检索速度，这也是创建索引的最主要的原因。

## 第三，可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

## 第四，在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。

## 第五，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。

# MyISAM数据库引擎。

## 只支持表级锁，在操作表的时候会自动给表加锁。也可以通过lock table的命令锁表，模拟事务，但是消耗非常大。

# InnoDB数据库引擎。

## 支持事务和行级锁。

## 事务的ACID属性。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 隔离级别/副作用 | 数据一致性 | 脏读 | 不可重复读 | 幻读 | 解释 |
| read uncommitted | 不读物理损坏的数据 | 是 | 是 | 是 | 不能解决任何问题 |
| read committed | 语句级 | 否 | 是 | 是 | oracle默认 |
| repeatable read | 事务级 | 否 | 否 | 是 | mysql默认 |
| serializable | 事务级 | 否 | 否 | 否 | 锁表 |

# 事务的ACID属性：

## 原子性（Atomicity）原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。

## 一致性（Consistency）：事务必须使数据库从一个一致性状态变换到另外一个一致性状态。

## 隔离性（Isolation）：事务的隔离性是指一个事务的执行不能被其他事务干扰，即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

## 持久性（Durability）：持久性是指一个事务一旦被提交，它对数据库中数据的改变就是永久性的，接下来的其他操作和数据库故障不应该对其有任何影响。

# Alert,modify,change

Alert:

Modify:细微操作：改字段、长度等。

Change:大操作。