# MySQL常用命令

数据库安装：mysql官网->https://dev.mysql.com/downloads/installer/

命令行查询：

数据库连接：win+R 输入cmd --> cd进入到C:\Program Files\MySQL\MySQL Server 8.0\bin --> mysql -h localhost -u root -p或mysql -u root -p --> 输入mysql密码

基于上述操作可以执行以下命令(;号不能省略)

查看数据库信息命令：

查询所有数据库-->show databases;

查询当前使用的数据库-->select database();

查看数据库使用端口-->show variables like ‘port’;

查看数据库编码-->show variables like ‘character%’;

查看数据库表信息-->show tables;

列出表的列信息-->show columns from 表名;

查看表结构信息-->desc 表名;

查看数据表当时创建表的语句-->show create table 表名;

修改表结构alter add/modify/change

--修改表添加字段

---修改表结构-->alter table 表名 add 列名 类型/约束 [default 默认值];

--修改表–修改字段；不重命名版

---alter table 表名 modify 列名 类型及约束：

--修改表–修改字段：重命名版

---alter table 表名 change 原列名 新列名 类型及约束；

--修改表–删除字段

---alter table 表名 drop 字段;

删除表-->drop table 表名;

数据库增删改查：

插入：

指定列插入(值和列一一对应)-->insert into 表名 (列1,列2...) values (值1,值2...);

多行插入 批量插入-->insert into 表名（列1,...） values (值1,...),(值2,...)...;

修改：

update 表名set 列1=值1，列2=值2…[where 条件];

删除：

物理删除 –-> delete from 表名 [where 条件判断]

完整的select语句：

按顺序：

from 表名

where 条件

group by //分组

select distinct \* //去重

having ...

order by ...

limit start,count

查询语句大全：

-简单查询：

直接查询-->select 字段 from 表名;

查询所有字段 select \* from 表名;

条件查询-->select 字段 from 表名 where 条件;

模糊查询-->select 字段 from表名 where 字段 like ‘%数据%’;

条件运算符：> < = != <> >= <=

逻辑运算符：and or not

in和not in 运算符：in-->在一个条件列表中 not in -->不在一个条件列表中

排序查询-->select 字段 from 表名 order by 字段 排序方式(asc升序，desc降序);

-高级查询：

--范围运算-->select 字段 from 表名 where 字段 between 范围1 and 范围2;

--限制查询(分页查询)-->select 字段 from 表名 limit n,m;(n开始记录行(0表示第一行)，m表示显示行，从n开始，共显示几行记录，分页指的是从n开始，以m为页的记录数进行查询，需要一个current当前页开始start = (current-1)\*m，记录数m)

--嵌套查询(分为查询语句{子查询语句},使用嵌套查询，两个表必须要有一个关联字段)-->select 字段1,字段2 from 表1 where 字段1=(select 字段1 from 表2 where 字段2 = 值);

--多表查询(多表连接查询，多表之间需要有一个共同字段连接多个表)，常见连接有:内连接,外连接

---内连接：根据两个表的共有字段进行匹配，将符合条件的合集进行拼接，关键字inner join ...on...，通常位于表名后面

---语法：select 字段 from 表1 inner join 表2 on 表1.字段=表2.字段;

---左连接：左连接是以左表为主，指定字段都显示，右表为从表，没内容显示null。关键字：left join ... on，通常位于表名后面

---语法：select 字段 from 表1 left join 表2 on 连接条件;

---右连接：右连接是以右表为主，指定字段都显示，坐标为从表，没内容显示null

--聚合函数：

---最小值(查询字段的最小数值)-->select min(字段) from 表名;

---最大值(查询字段的最大数值)-->select max(字段) from 表名;

---求和sum()-->select sum(字段) from 表名;

---平均值avg()-->select avg(字段) from 表名;

---统计记录count(统计记录的个数)-->select count(字段) from 表名

---as聚合别名，常用于聚合查询之后-->select 运算函数(字段) as 别名 from 表名;

---大小写转换upper()，常位于select之后-->select upper/lower(字段) from 表名;

通过txt文件导入数据：

1、查看local\_infile变量值是否为on:show global variables like ‘local\_infile’;

2、如果是off，需改为on：set global local\_infile=1

3、如果还有错误，登录时执行一下语句：

mysql -u root -p --local-infile=1

txt文件数据格式：

字段与字段数据之间以一个制表符隔开，如果数据为空则使用\N，每一行对应表中的一条数据。

导入命令：

load data local infile ‘path路径名，文件夹之间以”\\”隔开\\.txt’ into table 表名;

数据库语言分类：

1. DDL数据定义语言

数据库及表的创建、删除、修改，CREATE、DROP、ALTER

1. DQL数据查询语言

数据库的数据查询，SELECT

1. DML数据操纵语言

用于数据库表中数据的增加、删除、修改，INSERT INTO、DELETE、UPDATE

1. 数据控制功能DCL

数据库访问权限的控制，如给用户授予访问权限：GRANT语句，取消授权：REVOKE

1. 事务控制语言TCL

BEGIN，SAVEPOINT xxx，ROLLBACK，ROLLBACK TO xxx，COMMIT

用于提交事务和回滚事务，维护数据的一致性

事务：数据库中的事务是指对数据库执行一批操作，在同一个事务当中，这些操作最终要么全部执行成功，要么全部失败，不会存在部分成功的情况。

-事务是一个原子操作。是一个最小执行单元，可以由一个或多个SQL语句组成

-在同一个事务当中，所有的SQL语句都成功执行时，整个事务成功，有一个SQL语句执行失败，整个事务都执行失败。

事务的特点：原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability)

原子性：事务的整个过程如原子操作一样，最终要么全部成功，要么全部失败，整个结果不能分割。

一致性：一个事务必须使数据库从一个一致性状态变换到另一个一致性状态。

隔离性：一个事务的执行不能被其他事务干扰。一个事务内部的操作及使用的数据对并发的其他事务时隔离的，并发执行的各个事务之间不能相互干扰。

隔离级别：读未提交(read uncommitted)、读已提交(read committed)、可重复读(repeatable read)、串行化(serializable)

持久性：一个事务一旦提交，它对数据库中数据的改变就应该时永久性的。当事务提交之后，数据会持久化到硬盘，修改时永久性的。

MySQL中事务操作：

Mysql中事务默认是隐式事务，执行insert、update、delete操作的时候，数据库自动开启事务、提交和回滚事务

是否开启隐式事务由变量autocommit控制。

事务分类：显示事务和隐式事务

隐式事务：事务自动开启、提交或回滚，比如insert、update、delete语句，事务的开启、提交或回滚由mysql内部自动控制的。

查看是否开启autocommit：show variables like ‘autocommit’;

显示事务：事务需要手动开启、提交或回滚，由开发者自己控制。

2种方式手动控制事务：

方式1：

set autocommit=0;//设置不自动提交事务

commit|rollback//执行事务操作

方式2：

start transaction;//开启事务

commit|rollback;执行事务操作

savepoint和rollback to：配合使用，回滚部分数据

savepoint part1;

。。。。

rollback to part1;//回滚了part到rollback之间的事务

只读事务：表示在事务中执行的是一些只读操作，如查询，但是不会做insert、update、delete操作，数据库内部对只读事务可能会有一些性能上的优化。

start transaction read only;

事务中存在的问题：

1、更新丢失

丢失更新就是两个不同的事务在某一时刻对同一数据进行读取后，先后进行修改。导致第一次操作数据丢失。有2中情况：1、a,b事务同时操作同一数据，a先对数据进行了更改，b再次更改时失败然后回滚，把a更新的数据也回滚了。2、a先对数据进行修改，b再次更改并提交，把a提交的数据给覆盖了。

2、脏读

一个事务在执行的过程中读取到了其他事务还没有提交的数据。两个事务同时操作同一数据，a事务对该数据进行了修改还没提交的时候，b事务访问了该条事务，并且使用了该条数据，此时a事务回滚，那么b事务读到的就是脏数据。

3、读已提交：事务中的每次读取操作，读取到的都是数据库中其他事务已提交的最新的数据

4、不可重复读：在同一事务中，多次读取同一数据返回的结果有所不同，后续读取可以读到另一事务已提交的更新数据。（多次读取的数据不同）

5、可重复读：一个事务操作中对于一个读取操作不管多少次，读取到的结果都是一样的。后续读取不能读到另一事务已提交的更新数据。

6、幻读：新增操作-->事务中后面的操作需要上面的读取操作提供支持，但是读取操作却不能支持下面的操作时产生的错误，就行发生了幻觉一样。修改删除操作-->事务a在操作一堆数据的时候，事务b插入了一条数据，a事务再次查询，发现多了一条数据，像是幻觉。

为了解决并发控制可能产生的异常问题，数据库定义了四种事务的隔离级别

读未提交：READ-UNCOMMITTED

读已提交：READ-COMMITTED

可重复读：REPEATABLE-READ

串行：SERIALIZABLE

查看隔离级别：show variables like ‘transaction\_isolation’;

隔离级别的设置：修改mysql中的my.ini文件

方式1设置事务隔离级别：

设置全局隔离级别set global transaction isolation level REPEATABLE READ;

设置会话隔离级别 set session transaction isolation level REPEATABLE READ;

方式2隔离级别设置,READ-UNCOMMITTED读未提交,READ-COMMITTED读已提交,REPEATABLE-READ可重复读,SERIALIZABLE串行

重启mysql：





读未提交：当前事务读取了其他事务还未提交的数据。可以读取到其他事务未提交的数据

读已提交：事务a内部的相同查询语句在不同时刻读出的结果不一致。也就是在同一个事务内，能读取到其他事务已经提交的数据。只能读取到其他事务已经提交的数据，不能读取到其他事务未提交的数据。

可重复读：事务a内部的相同查询语句在不同时刻读出的结果一致。读取不到其他数据已提交和未提交的数据。

串行都：使得事务串行化执行，与并发矛盾。读写互斥：事务A中先读取操作，事务B发起写入操作，事务A中的读取会导致事务B中的写入处于等待状态，直到A事务完成为止。

保证用户并发访问统一资源时，数据的一致性、安全性、有效性等问题，此时就会用上锁的概念。

分类：

粒度：表锁、行锁

操作：读锁（共享锁）、写锁（排他锁）

性能：乐观锁、悲观锁

1、表锁

优点

每次操作都是锁住表，锁的开销小，加锁快，直接锁住表就完事，并且不会出现死锁。

缺点

当然缺点也是足够明显的，锁冲突率很高（直接锁的表），并发度很低

表锁中读锁会阻塞写，但是不会阻塞读。而写锁则会把读和写都阻塞

2、行锁

缺点

加锁开销大、加锁慢、会出现死锁。

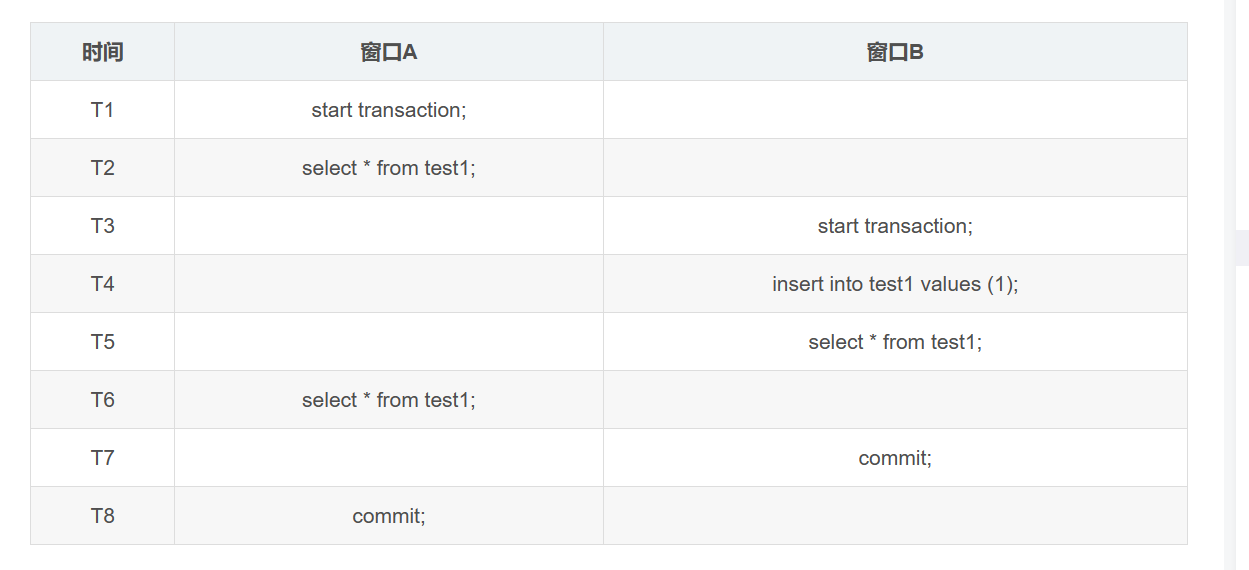
优点

锁的粒度小、发生锁的冲突概率比较小、并发高。

案例：

1、读未提交：设置读未提交隔离级别

结论：读未提交情况下，可以读取到其他事务还未提交的数据，多次读取结果不一样，出现了脏读、不可重复读、幻读



T2-A：无数据，T6-A：有数据，T6时刻B还未提交，此时A已经看到了B插入的数据，说明出现了脏读。

T2-A：无数据，T6-A：有数据，查询到的结果不一样，说明不可重复读。

2、读已提交：设置读已提交隔离级别



T5-B：有数据，T6-A窗口：无数据，A看不到B的数据，说明没有脏读。

T6-A窗口：无数据，T8-A：看到了B插入的数据，此时B已经提交了，A看到了B已提交的数据，说明可以读取到已提交的数据。

结论：读已提交情况下，无法读取到其他事务还未提交的数据，可以读取到其他事务已经提交的数据，多次读取结果不一样，未出现脏读，出现了读已提交、不可重复读、幻读。

3、可重复读：设置可重复读隔离级别

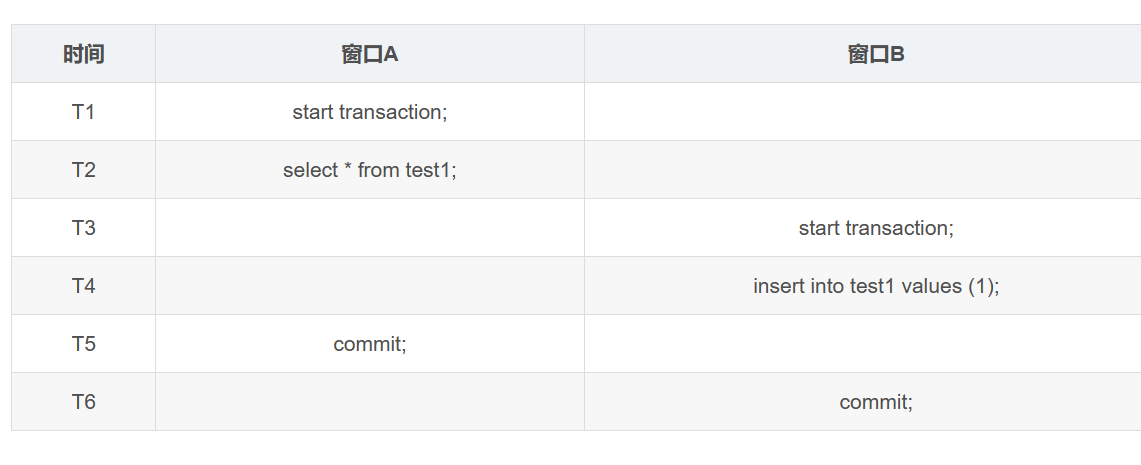
结论：可重复读情况下，未出现脏读，未读取到其他事务已提交的数据，多次读取结果一致，即可重复读。



T2-A、T6-A窗口：无数据，T5-B：有数据，A看不到B的数据，说明没有脏读。

T8-A：无数据，此时B已经提交了，A看不到B已提交的数据，A中3次读的结果一样都是没有数据的，说明可重复读。

4、串行：设置串行隔离级别



事务只能串行执行了。串行情况下不存在脏读、不可重复读、幻读的问题了。

默认隔离级别：如果没有指定隔离级别，数据库就会使用默认的隔离级别。在MySQL中，如果使用 InnoDB，默认的隔离级别是Repeatable Read。

总结：

理解事务的4个特性：原子性、一致性、隔离性、持久性

掌握事务操作常见命令的介绍

set autocommit可以设置是否开启自动提交事务

start transaction：开启事务

start transaction read only：开启只读事物

commit：提交事务

rollback：回滚事务

savepoint：设置保存点

rollback to 保存点：可以回滚到某个保存点

掌握4种隔离级别及了解其特点

脏读、不可重复读、幻读

幻读案例：唯一约束



案例数据准备：

create table test1 (a int); delete from test1;

select \* from test1;

作为读未提交、读已提交、可重复读、串行

幻读：create table t\_user(id int primary key,name varchar(16) unique key);

