第二章：sql语句

1.mysql -u root -p

2.create database db_name;

3.use db_name;

4.create table table_name (列名 类型 约束) ;

5.drop table table_name;

6.alter table tabel_name add column 列名 类型等 ;

7.alter table table_name change 列名 new列名 类型;

8.alter table table drop 列名;

9.alter table table_name rename table_name2;

10.insert into table_name (列名，列名 ...) values (列值，列值...);

11.update table_name set 列名=列值 where 列名=列值;

12.delete from table_name where 列名=列值;

13.select 列名1，列名2... from table_name where 列名=列值;

14.select distinct 列名 from table_name; //distinct去重

15.order by 列名 [desc] limit start,length ; //指定行数的排序

16.group by //分类聚合 对某一字段进行分类聚合操作

17.select 列名1，列名2 from table_name1 left join table_name2 on 条件;//左连接

获取所有左边表的所有字段的值，右表中不满足条件的用null填充

右查询同理

18.select * from table_name where 列名 in (select 列名2 from table_name2);

子查询

19.select name from table_name1 union select name from table_name2;

联合查询 union 会去除重复的值，而union all 不去除

第三章：数据类型

1.数值：int 4字节 ,integer 4 , bigint 8 , float 4， double 8 ...

2.日期：date 4 如：2018-02-02 ； datetime 8 如：2012-02-02 02：02：02

timestamp 4 时间戳

3.字符串类型：char(m) m∈（0，255）； varchar(m) m∈（0，65535）；text 同上

第四章：常用函数

1.concat(s1,s2 ...) 连接s1,s2 ...为一个字符串

2.now() 返回当前时间

3.count(*) 行数

.....

第十章：索引的设计与使用

使用索引的原则：①出现在where 条件字句中的列 ②使用唯一索引时，字段值的分布返回绝大越好 ③使用短索引查询字符串，而不是全部字符串进行查询, ④不要过度使用索引

hash索引：自能用 = 或 <=>操作符

btree索引：使用> ,< , <= ,>= , between等

第十二章：存储过程和函数

1.区别：①函数必须有返回值，而存储过程并不必须有. ②存储过程参数有IN OUT INOUT 三种类型，而函数只有in类型

2.创建存储过程 ：

```
delimiter $$
create procedure pro_name(参数... 如： in param_name int )
begin
    执行操作.....
end$$
delimiter //
```

调用： call pao_name(1);

删除：drop procedure pro_name;

查看创建语句：show create procedure pro_name;

3.存储过程的变量声明，赋值，流程控制语句，等（在 begin 和 end 之间）

①声明变量 ： declare name char(10);

②变量赋值：set name = "kk";

流程控制：

③if condition then dosomething

elseif condition then .. else .... endif;

④case condition ... enndcase;

⑤loop循环： lable:loop ..... end loop; leave lable；可以跳出循环

⑥while ... do ... end while;

第十三章：触发器trigger

只能创建在永久表上，临时表不行.

指的是，在对表中的某一操作之前或者之后，做相应的处理.

1.用法：

create trigger t_name trigger_time trigger_envent on table_name for each row;

t_name : 触发器名字

trigger_time : 有before和after

trigger_event : 事件，三种，insert ， update ， delete

例：

```
delimiter $$
create trigger t_name after insert on table_name for each row
begin
.....
end $$
delimiter //
```

删除触发器：drop trigger trigget_name;

注意：触发器中不要使用事务.

第18章：sql优化

1.通过show status 命令了解各种sql的执行频率.

show status like 'innodb_rows_read';

innodb_rows_read ; select查询返回的行数

innodb_rows_inserted；执行insert操作的行数

innodb_rows_updated；执行update操作的行数

innodb_rows_deleted；执行delete操作的行数

2.通过explain关键字分析sql语句的执行计划，分析其效率

```
explain select * from mk_article;
```

结果字段解释：

select_type：显示本行是简单或复杂select；

type ： 数据访问/读取操作类型（ALL、index、range、ref、eq_ref、const/system、NULL）从左到右，性能由最差到最好.

all:全表查

index：索引全扫描

range：索引范围扫描

ref:使用非唯一索引，等值查询

...

key: 显示mysql决定采用哪个索引来优化查询

rows:为了找到所需的行而需要读取的行数，估算值，不精确。通过把所有rows列值相乘，可粗略估算整个查询会检查的行数

3.索引问题

3.1mysql中索引的使用场景

①匹配全值：对所有的索引的列等值匹配.如组合索引中.

②匹配值范围查找：对索引值进行范围查找.

③最左前缀：组合索引中，必需使用最左边的索引，如：row1 + row2 + row3的一个组合索引，能够使用组合索引的条件查询只有：（row1）（row1+row2）（row1+row2 + row3）这三种情况.

④只查询索引

....

3.2存在索引，但是不会使用的情况

①以%开头的like查询，不会使用btree索引.

②数据出现隐式的转换的时候，int 转成 bigint

③组合索引即复合索引，查询条件中，必需满足最右原则.

④如果mysql估计使用索引比全表查更慢，则不用.

⑤用or分开的条件中，条件中的列必需全部有索引，否则不会使用索引.

3.3查看索引被使用的情况

show status like 'handle_read';

4.常用sql优化

4.1大批量插入数据时：

set uinque_checks=0;设置关闭表的非唯一性索引的更新

set autocommit=0 ; 关闭自动提交

使用文件插入比insert语句快奖金20倍.

4.2 排序的优化：减少额外的排序，通过索引直接返回原有序数据。

4.3 优化分页查询：在分页查询的类中，记录一下读取的第一行的主键id，查询下一页时，可以直接用： id > 查询的id limit 10; 这样会更快，利用的唯一索引.