

Day14

java课程-李彦伯

Day14

mysql的安装(详见文档)

mysql绑定eclipse(详见文档)

SQL分类

常用sql语句

数据库操作

表中数据类型

常用类型

常用约束

数据库表中记录的操作

数据库表中记录的查询操作

条件查询

聚合函数查询

分组操作

分页操作

作业

查询总结

设计表所需数据

多表查询

数据准备

交叉连接(基本不用)

内连接查询

外连接查询

子查询

JDBC

开发步骤

mysql的安装(详见文档)

mysql绑定eclipse(详见文档)

SQL分类

- 数据定义语言：简称DDL(Data Definition Language)，用来定义数据库对象：数据库，表，列等。关键字：create，alter，drop等
- 数据操作语言：简称DML(Data Manipulation Language)，用来对数据库中表的记录进行更新。关键字：insert，delete，update等
- 数据控制语言：简称DCL(Data Control Language)，用来定义数据库的访问权限和安全级别，及创建用户。
- 数据查询语言：简称DQL(Data Query Language)，用来查询数据库中表的记录。关键字：select，from，where等

常用sql语句

数据库操作

- 创建数据库 `create database 库名 或者 create database 库名 character set 编码`
- 查看编码 `show create database 库名`
- 删除数据库 `drop database 库名`
- 使用数据库 `use 库名`
- 查看当前操作的库 `select database()`
- 查看当前所有的数据库 `show databases`

表中数据类型

- 数据库中表中的数据类型

分类	类型名称	说明
整数类型	tinyInt	很小的整数
	smallint	小的整数
	mediumint	中等大小的整数
	int(integer)	普通大小的整数
小数类型	float	单精度浮点数
	double	双精度浮点数
	decimal (m,d)	压缩严格的定点数,m表示总位数,d表示小数点几位

日期类型	year	YYYY 1901~2155
	time	HH:MM:SS -838:59:59~838:59:59
	date	YYYY-MM-DD 1000-01-01~9999-12-3
	datetime	YYYY-MM-DD HH:MM:SS 1000-01-01 00:00:00~ 9999-12-31 23:59:59
	timestamp	YYYY-MM-DD HH:MM:SS 1970~01~01 00:00:01 UTC~2038-01-19 03:14:07UTC
文本、二进制类型	CHAR(M)	M为0~255之间的整数
	VARCHAR(M)	M为0~65535之间的整数
	TINYBLOB	允许长度0~255字节
	BLOB	允许长度0~65535字节
	MEDIUMBLOB	允许长度0~167772150字节
	LOBLOB	允许长度0~4294967295字节
	TINYTEXT	允许长度0~255字节
	TEXT	允许长度0~65535字节
	MEDIUMTEXT	允许长度0~167772150字节
	LONGTEXT	允许长度0~4294967295字节
	VARBINARY(M)	允许长度0~M个字节的变长字节字符串
	BINARY(M)	允许长度0~M个字节的定长字节字符串

常用类型

类型	描述
int	整型
double	浮点型
varchar	字符型
date	日期型

常用约束

约束名称	sql表示	说明
主键约束	primary key	唯一性，非空性
唯一约束	unique	唯一性，可以空，但只能有一个
默认约束	default	该数据的默认值
自增长	auto_increment	一般用于主键自动增长
非空约束	not null	数据不能为空

- 创建表

```
create table 表名(  
    字段名 类型(长度) [约束],  
    字段名 类型(长度) [约束],  
    字段名 类型(长度) [约束]  
);
```

- 查看当前数据库中的表 `show tables`
- 查看表结构 `desc 表名`
- 删除表 `drop table 表名`
- 修改表
 - 添加一列 `alter table 表名 add 字段名 类型(长度) [约束]`
 - 修改列的类型(长度,类型) `alter table 表名 modify 要修改的字段名 类型(长度) [约束]`
 - 修改列名 `alter table 表名 change 旧列名 新列名 类型(长度) [约束]`
 - 删除表中的列 `alter table 表名 drop 列名`
 - 修改表名 `rename table 表名 to 新表名`
 - 修改表的字符集 `alter table 表名 character set 编码`
 - 查看当前表结构 `show create table 表名`

数据库表中记录的操作

- 插入记录 `insert into 表名(列名1,列名2,列名3.....) values(值1,值2,值3.....)` 或者 `insert into 表名 values(值1,值2,值3.....)`
- 插入记录中文乱码问题 `set names gbk;`
- 修改表记录 `update 表名 set 字段名=值, 字段名=值, 字段名=值.....`
- 修改表记录带条件 `update 表名 set 字段名=值, 字段名=值, 字段名=值..... where 条件`
- 删除表记录带条件 `delete from 表名 where 条件`

- 删除表记录不带条件 `delete from 表名;` 或者 `truncate table 表名;`

数据库表中记录的查询操作

- 查询所有 `select * from product;`
- 查询列 `select pname,price from product;`
- 查询所有商品信息使用表别名: `select * from product as p;`
- 查询商品使用列别名 `select pname as p from product`
- 查询去掉重复 `select distinct(price) from product;`
- 将所有的商品的价格+10进行显示 `select pname,price+10 from product;`

条件查询

运算符分类	运算符表示	说明
比较运算符	< <= >= = <>	大于、小于、大于(小于)等于、不等于
	BETWEEN ...AND...	显示在某一区间的值(含头含尾)
	IN(set)	显示在in列表中的值，例：in(100,200)
	LIKE 通配符	模糊查询，Like语句中有两个通配符: % 用来匹配多个字符；例first_name like 'a%'; _ 用来匹配一个字符。例first_name like 'a_';
	IS NULL	判断是否为空 is null; 判断为空 is not null; 判断不为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立，例：where not(salary>100);

- 查询商品名称为洗衣机的商品信息 `select * from product where pname='洗衣机';`
- 查询价格>60的商品信息 `select * from product where price > 60;`
- 查询名字中含有机机的商品信息 `select * from product where pname like '%机%';` 或者 `select * from product where pname like "%机%";`
- 查询id在(3,6,9)范围的商品信息 `select * from product where pid in (3,6,9);`
- 查询商品名称中含有机并且id为6的商品信息 `select * from product where pname like '%机%' and pid=6;`
- 查询id为2或者6的商品信息 `select * from product where pid=2 or pid=6;`
- 按照商品价格(升序/降序) `select * from product order by price asc;` , `select * from product order by price desc;`

- 查询名称含有器的并且按照价格降序排序 `select * from product where pname like '器%' order by price desc;`

聚合函数查询

- 获取所有商品的价格总和 `select sum(price) as 价格总和 from product;`
- 获取所有商品的平均价格 `select avg(price) as 平均价格 from product;`
- 获取所有商品的个数 `select count(1) as 总数量 from product;`

分组操作

- 根据pname字段分组,分组后统计商品的个数 `select pname,count(*) from product group by pname;`
- 根据pname分组,分组后统计商品的平均价格,并且平均价格>3000 `select pname,avg(price) as avgprice from product group by pname having avgprice > 3000 ;`

分页操作

`select * from product limit a,b` 其中 `a` 代表从第几条记录开始查询,如果 `a` 是 5,那么记录的条数就是从 6 开始, `b` 代表一页显示几条

- 当显示的内容小于 `b` 的时候,sql 不会发生错误,会正常显示剩余所有的条数

作业

1. 每页显示的 5 条数据,请显示出第 3 页的数据
2. 如何计算总页数

查询总结

- `select` 一般在的后面的内容都是要查询的字段
- `from` 要查询到表
- `where`
- `group by`
- `having` 分组后带有条件只能使用 `having`
- `order by` 它必须放到最后面

设计表所需数据

```
create table product (  
pid int primary key auto_increment,  
pname varchar(30) not null,  
price int not null  
);  
  
insert into product values (null,'海尔洗衣机',3000);  
insert into product values (null,'荣升电冰箱',1000);  
insert into product values (null,'美的微波炉',3000);  
insert into product values (null,'西门子洗衣机',3000);  
insert into product values (null,'可可洗脚盆',300);  
insert into product values (null,'手电筒',30);  
insert into product values (null,'乖乖洗衣机',30000);  
insert into product values (null,'格力空调',8999);  
insert into product values (null,'微波炉',300);  
insert into product values (null,'电视机',2000);  
insert into product values (null,'微波炉',1000);  
insert into product values (null,'美的烤箱',399);  
insert into product values (null,'TCL电视',4500);  
insert into product values (null,'格力空调',5000);  
insert into product values (null,'联想笔记本',3111);  
insert into product values (null,'西门子冰箱',6000);  
insert into product values (null,'格力空调',8000);  
insert into product values (null,'小米洗衣机',2500);  
insert into product values (null,'便宜洗衣机',30);  
insert into product values (null,'美的电饼铛',2500);  
insert into product values (null,'美的电饼铛',2500);  
insert into product values (null,'电冰箱',3230);  
insert into product values (null,'海尔洗衣机',4000);  
insert into product values (null,'格力电风扇',3000);  
insert into product values (null,'美的电饭锅',9000);  
insert into product values (null,'自动洗衣机',4000);  
insert into product values (null,'小天鹅洗衣机',6000);  
insert into product values (null,'格力变频空调',15000);  
insert into product values (null,'小天鹅电风扇',300);  
insert into product values (null,'大力洗衣机',20000);  
insert into product values (null,'电风扇',300);  
insert into product values (null,'美的冰箱',2000);  
insert into product values (null,'买空调到格里',9000);  
insert into product values (null,'乐视大空调',5);  
insert into product values (null,'长虹电风扇',3000);  
insert into product values (null,'九阳榨汁机',300);  
insert into product values (null,'viov x9',2999);  
insert into product values (null,'nb洗衣机',30000);  
insert into product values (null,'死全家捕鼠器',888);  
insert into product values (null,'九阳电饭煲',5200);  
insert into product values (null,'美的空调',5555);
```

```
insert into product values (null,'老板空调',19999);
insert into product values (null,'海尔液晶电视',30000);
insert into product values (null,'我的电饼铛',6666);
insert into product values (null,'大神洗衣机',500);
insert into product values (null,'魅族豆浆机',2000);
insert into product values (null,'吃饱饱电饭煲',3000);
insert into product values (null,'九阳豆浆机',1100);
insert into product values (null,'电风扇',10);
insert into product values (null,'飞科剃须刀',300);
insert into product values (null,'海尔电冰箱',2975);
insert into product values (null,'东风X-8洗衣机',11000);
insert into product values (null,'电风扇',10);
insert into product values (null,'三星电视机',9000);
insert into product values (null,'iphone10',30);
insert into product values (null,'志高空调',5000);
insert into product values (null,'所罗门冰箱',5000);
insert into product values (null,'爱疯打火机',666);
insert into product values (null,'战斗机',30000000);
insert into product values (null,'海氏（Hauswirt）多功能和面机打蛋器搅拌机',1600);
insert into product values (null,'Max电视机',5600);
insert into product values (null,'小飞鹅洗衣机',30000);
insert into product values (null,'终结者T100电冰箱',300000);
insert into product values (null,'奥斯科空调',4999);
insert into product values (null,'波波洗衣机',4999);
```

多表查询

数据准备

- 一对多

部门表	
id	
name	
1	开发部
2	教学部
3	网络部

员工表		
id		
name		
外键id		
1	李雷	1
2	张三	3

一对多
 部门表中的一个记录
 对应员工表中的多个记录

• 多对多



```

create table department (
id int primary key auto_increment,
name varchar(30) not null
);
create table staff(
id int primary key auto_increment,
name varchar(10) not null,
did int not null default 1
);
insert into department values (null,'开发部');
insert into department values (null,'教学部');
insert into department values (null,'网络部');
insert into department values (null,'运维部');
insert into department values (null,'市场部');

insert into staff values (null,'张三工程师',1);
insert into staff values (null,'李四老师',2);
insert into staff values (null,'王五装电脑',3);
insert into staff values (null,'可老师',2);
insert into staff values (null,'ss老师',2);
insert into staff values (null,'li老师',2);
insert into staff values (null,'lili网管',3);
insert into staff values (null,'lucy工程师',1);
insert into staff values (null,'pl',4);

```

交叉连接(基本不用)

```
select * from staff , department;
```

内连接查询

使用关键字inner join,inner可以省略

- 隐式内链接

```
select * from department d,staff s where d.id = did;
```

- 显式内链接

```
select * from department as d inner join staff as s on d.id = s.did;
```

外连接查询

关键字是outer join,outer可以省略,外连接要用on不能使用where

- 左外连接

```
select * from department d left join staff on d.id = did;
```

- 右外连接

```
select * from department d right join staff on d.id = did;
```

子查询

一个查询的结果作为另一个查询的条件

```
select * from staff where did = (select id from department where name = '教学部');
```

JDBC

JDBC (Java Data Base Connectivity,java数据库连接) 是一种用于执行SQL语句的Java API , 可以为多种关系数据库提供统一访问 , 它由一组用Java语言编写的类和接口组成。是Java访问数据库的标准规范

开发步骤

- 在工程中导入jar包
- 注册驱动 `Class.forName("com.mysql.jdbc.Driver");`

- 获得连接 `Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb","root","root");`, 如果是本机路径可以写为 `Connection con = DriverManager.getConnection("jdbc:mysql:///mydb","root","root");`
- 获取执行平台 `Statement stmt = con.createStatement();`
- 执行sql语句 `int num = stmt.executeUpdate(String sql);` 或者 `ResultSet rs= stmt.executeQuery(String sql);`
- 处理结果集

```
while( rs.next() ){  
    //方法参数为数据库表中的列名  
    String sid = rs.getString("pid");  
    //获取当前行的分类名称  
    String sname = rs.getString("pname");  
}
```

- 释放资源

```
rs.close();  
stmt.close();  
con.close();
```