

## 数据库及SQL

笔记本：default

创建时间：2016/9/23 1:41

更新时间：2016/12/29 17:40

作者：【千锋互联】JS高级讲师--尹涛

---

### 数据库是什么？

存储数据的仓库，是一个很广义的概念。任何能存储数据的容器，都可以被称之为数据库。一个MP3，一个U盘，一个文件夹，甚至一个txt文档，都可以叫做数据仓库。一般来情况下我们所说的数据库文件，指的是被特定软件结构化的数据文件。

### 什么是结构化的数据？



我是一个热爱生活的人，很乐观。 我今年刚从清华大学毕业。我学的是专业是前端开发，我目前还是单身，并没有女朋友，欢迎大家跟我交朋友，谢谢。

仔细阅读上面这段话，如果让你提取这段文字的中心意思，你能做到吗？  
我相信对一个正常人来说都不是难事。可是，这段话如果交给计算机去处理呢？  
**计算机没有人类的思维，这件事对它来讲恐怕就太难了**

因此，有时候我们才需要将数据结构化。结构化的数据是什么样子呢？  
以上面那段话为例，我们将它的信息提取出来，以表格的形式展现：

姓名	年龄	性别	毕业院校	所在地	性格	专业
张三	23	男	清华	北京	开朗	前端

这样一来，我们把一段复杂的信息，进行了结构化处理，瞬间显得清爽多了。计算机处理起来也相当方便。因此，这样的数据，我们就称之为结构化数据。

### 数据库管理软件是什么？

一种对数据库文件进行管理的软件，通常简称数据库。

### 常见的数据库管理软件有哪些？

Oracle、MySQL、SQLServer、DB2.....

这些都是我们所谓的关系型数据库，还有非关系型的数据库。

例如：

Redis , Tokyo Cabinet , Cassandra , Voldemort , MongoDB , Dynamite ,  
HBase , CouchDB , Hypertable , Riak , Tin , Flare , Lightcloud ,  
KiokuDB , Scalaris , Kai , ThruDB.....

### 那么关系型数据库和非关系型数据库又是指什么呢？

我们仍然以结构化的数据为例来说明：

姓名	身份证	银行卡号	余额	消费纪律

张三	45466166464646794616156461886240.0	2016.06.01 14:20:56 -163200RMB 买大象 2016.06.03 19:10:11 -200RMB 看电影 .....
----	------------------------------------	--

这样的数据看起来是不是很别扭呢？因为它查询起来极不方便。

因此我们把它改造一下：

编号	姓名	身份证	银行卡号	余额
1	张三	4546616646464646	7946161564611616	886240.0
2	李四	4621649518444484	6260614649616149	40600.0

编号	时间	消费金额	备注	用户编号
1	2016.06.01 14:20:56	163200	买大象	1
2	2016.06.08 19:10:11	260	看电影	2
3	2016.07.15 11:45:20	16000	首饰	1

我们把用户账号的信息，和消费记录信息分离开来，通过用户编号进行了 关联  
既把不同的数据进行了分离，使得查询数据更加方便。同时又保证了它们关系的正确性。  
**这就是传统的关系型数据库。其中 用户编号 我们称之为外键**

**非关系型数据库是一种泛指。**

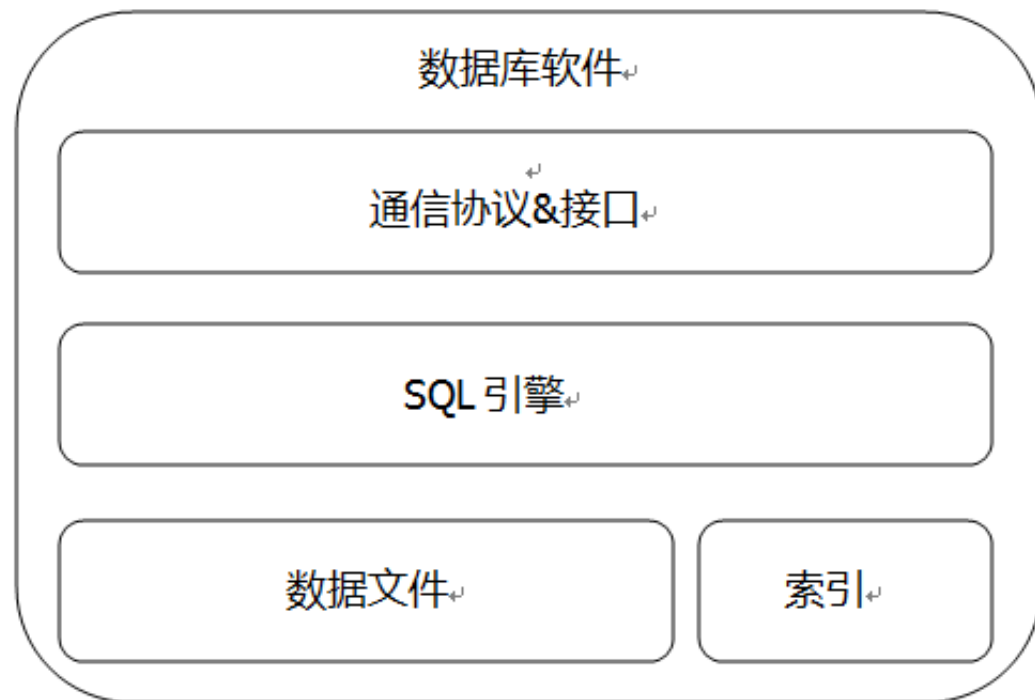
KV(键值)数据库， 例如缓存  
列存储数据库，例如分布式文件系统  
等等

为什么数据库软件需要端口号？？为什么说它是一种服务？

首先，你要理解数据库软件最大的作用是什么。  
它最大的功能在于，将数据结构化以后，可以对大量的数据提供非常便利的查询方式。  
**也就是我们接下来要讲的这门 查询语言： Struct Query Language**

是的，为了更方便的查询，它单独发明了一种查询语言，就是我们常说的SQL语言。  
目前也是W3C的规范之一。  
不过这种语言及其的简单，号称最接近自然语言的编程，因此不会带来很大的学习成本。

一个关系型数据库软件的结构大致如下：



因为实际当中，数据库都是通过网络远程访问的，它必须以提供服务的方式，让我们可以本地或远程连接

数据库管理软件如何安装？

### MySQL介绍

一种开源的数据库软件，市场占有率很高，现在属于Oracle公司

mysql默认端口 3306

mysql数据库如何存储数据？

以库为单位，在库中，以表格为单位，在表中分为行和列

如何使用Mysql呢？

安装完成后，打开MYSQL的命令行工具窗口。 输入你安装时设定的密码

常见命令：查看目前为止，有多少库存在？

```
show databases;
```

了解数据类型

创建表格的时候，每一列都有数据类型的要求，数据库中的类型跟编程中叫法不太一样但意思是一样的。

int ----> 整数

varchar -----> 字符串

blob -----> 二进制数据

date -----> 日期

SQL语句 (struct query language) 结构化查询语言

这种语言的发明，给我们查询数据提供了极大的便利，它也是一种标准，从1992到1999版因此，所有的数据库提供的SQL语言，基本都是一致的。

为什么说基本呢？ 因为SQL标准当中提供的功能实在太少，因此对于非标准之外的其他功能，数据库之间就各不相同了。

建库

```
create database yintao;
```

查看所有数据库

```
show databases;
```

打开某一个库

```
use yintao;
```

创建一张表

```
create table students(  
  id int unsigned not null auto_increment primary key,  
  name char(8) not null,  
  sex char(4) not null,  
  age tinyint unsigned not null,  
  tel char(13) null default "-"  
);
```

查看所有表

```
show tables;
```

查看某张表的结构

```
desc students;
```

为了简化建库建表的操作，有时我们会安装一个图形界面工具来连接数据库，例如Navicat



插入一条记录 insert

insert [into] 表名 [(列名1, 列名2, 列名3, ...)] values (值1, 值2, 值3, ...);

```
insert into students values(NULL, "王刚", "男", 20, "13811371377");  
insert into students (name, sex, age) values("孙丽华", "女", 21);
```

删除 delete

**delete from 表名称 where 删除条件;**

```
delete from students;  
delete from students where id=2;  
delete from students where age<20;
```

修改 update

**update 表名称 set 列名称=新值 where 更新条件;**

```
update students set tel=default where id=5;  
update students set age=age+1;  
update students set name="张伟鹏", age=19 where tel="13288097888";
```

查询 select

**select 列名称 from 表名称 [查询条件];**

```
select name, age from students;  
select * from students;  
  
select * from students where age > 21;  
select * from students where name like "%王%";  
select * from students where id<5 and age>20;
```

## 数据库的事务处理

所谓事务，就是以某件事为单位，将多条SQL语句进行捆绑  
这样做的目的是确保多条相关的SQL的完整性，也就是要么全部执行成功，要么全都不执行

例如：**账户A转账给账户B 1000元**

sql语句1 账户A减去1000元

sql语句2 账户B增加1000元

这两条SQL语句就是必须放在一个事务中的，这样才能确保转账的正确性

## 练习

```
create table EMP
(
  EMPNO numeric(4) PRIMARY KEY,
  ENAME VARCHAR(10),
  JOB VARCHAR(9),
  MGR numeric(4),
  HIREDATE DATE,
  SAL numeric(7,2),
  COMM numeric(7,2),
  DEPNO numeric(4)
);
```

```
CREATE TABLE Dept(  
  DEPTNO numeric(4),  
  DNAME VARCHAR(14),  
  LOC VARCHAR(13)  
);
```

```
CREATE TABLE Salgrade  
(  
  GRADE numeric,  
  LOSAL numeric,  
  HISAL numeric  
);
```

```
INSERT INTO Dept VALUES (10,'ACCOUNTING','NEW YORK');
```

```
INSERT INTO Dept VALUES (20,'RESEARCH','DALLAS');
```

```
INSERT INTO Dept VALUES (30,'SALES','CHICAGO');
```

```
INSERT INTO Dept VALUES (40,'OPERATIONS','BOSTON');
```

```
INSERT INTO EMP VALUES
```

```
(7369,'SMITH','CLERK',7902,'1980-12-17',800,null,20);
```

```
INSERT INTO EMP VALUES
```

```
(7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);
```

```
INSERT INTO EMP VALUES
```

```
(7521,'WARD','SALESMAN',7698,'1981-02-22',1250,500,30);
```

```
INSERT INTO EMP VALUES
```

```
(7566,'JONES','MANAGER',7839,'1981-04-02',2975,NULL,20);
INSERT INTO EMP VALUES
(7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30);
INSERT INTO EMP VALUES
(7698,'BLAKE','MANAGER',7839,'1981-05-01',2850,NULL,30);
INSERT INTO EMP VALUES
(7782,'CLARK','MANAGER',7839,'1981-06-09',2450,NULL,10);
INSERT INTO EMP VALUES
(7839,'KING','PRESIDENT',NULL,'1981-11-17',5000,NULL,10);
INSERT INTO EMP VALUES
(7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);
INSERT INTO EMP VALUES
(7900,'JAMES','CLERK',7698,'1981-12-03',950,NULL,30);
INSERT INTO EMP VALUES
(7902,'FORD','ANALYST',7566,'1981-12-03',3000,NULL,20);
INSERT INTO EMP VALUES
(7934,'MILLER','CLERK',7782,'1982-01-23',1300,NULL,10);

INSERT INTO SALGRADE VALUES (1,700,1200);
INSERT INTO SALGRADE VALUES (2,1201,1400);
INSERT INTO SALGRADE VALUES (3,1401,2000);
INSERT INTO SALGRADE VALUES (4,2001,3000);
INSERT INTO SALGRADE VALUES (5,3001,9999);
```

#要求列出每个雇员的姓名及年薪

SELECT ename,sal\*12 FROM emp;

#查看每月可以得到奖金的雇员信息

SQL> SELECT \* FROM emp WHERE comm is NOT NULL;

#要求基本工资大于1500，同时可以领取奖金的雇员信息

SQL> SELECT \* FROM emp WHERE sal>1500 AND comm is NOT NULL;

#查询在1981年雇佣的全部雇员信息，BETWEEN .. AND 包含等于的情况

SQL> SELECT \* FROM emp WHERE hiredate BETWEEN '01-JAN-81' AND '31-DEC-81';

#要求查询出雇员编号不是 7369、7499的雇员信息

SQL> SELECT \* FROM emp WHERE empno NOT IN(7369,7499);

#SQL中LIKE语句要注意通配符 % 和 \_

SQL> SELECT \* FROM emp WHERE hiredate LIKE '%81%';

#要求对雇员的工资由低到高进行排序，升序为默认(ASC)，降序(DESC)

SQL> SELECT \* FROM emp ORDER BY sal;

找出佣金高于薪金的60%的员工

SQL> SELECT \* FROM emp WHERE comm>sal\*0.6

找出部门10中所有经理(MANAGER)和部门20中所有办事员(CLERK)的详细资料

SQL> SELECT \* FROM emp WHERE (deptno=20 AND job='MANAGER') OR (deptno=10 AND job='CLERK');

找出既不是经理又不是办事员但其薪金大于或等于2000的所有员工的资料

SQL> SELECT \* FROM emp WHERE job NOT IN('MANAGER','CLERK') AND sal >= 2000;

显示不带有"R"的员工的姓名

SQL> SELECT ename FROM emp WHERE ename NOT LIKE '%R%';

后台增删改查练习

第一步： 将JDBC文件夹拷贝到 自己的tomcat服务器的 webapps目录下

第二步：在mysql中创建一个叫做 gp1602的库

第三步：在库中创建一个叫做student 的表 需要五个字段，(ID int,NAME varchar,SEX varchar,AGE int,PASSWORD varchar)

第四步：字段ID需要设为自动递增

第五步：启动服务器，测试接口

添加接口：<localhost:8080/JDBC/jdbc?type=insert&name=张三&age=52&sex=男&password=123465>

查询接口：<localhost:8080/JDBC/jdbc?type=query&pageno=2>

修改接口：<localhost:8080/JDBC/jdbc?type=update&id=5&name=张三&sex=女&password=123>

注：在进行修改数据时，ID是必须的，

删除接口：<localhost:8080/JDBC/jdbc?type=delete&id=4>

