使用数据库的必要性

> - 无论是集合、对象、程序一旦重启所有数据全部消失，无法做到持久化保存
> - xml 是可以保存数据的
> - 另外还可以通过IO流将数据保存到本地磁盘，但是数据缺乏结构化，无法描述复杂的业务逻辑，且读写比较慢。
> - java 里面双引号表示String 类型，单引号表示char类型，而数据库中是char（长度不尅变）和varchar(长度可变)

关于数据库

> - 一个项是数据库先行（表结构的设计、关系到项目的成败），每一个项目都有自己的数据库，项目经理将创建好的数据库放到一个服务器上，但开发的时候是个人考下来的，即为了数据安全，开发人员是链接到自己的本地数据库，以防数据被破坏。
> - 如何对数据库进行管理：
>   - slq语言是一个专门处理数据库的编程语言。
> - 数据库的组成：一个数据库是由多张表组成的，数据库中的数据保存在数据表中，数据白由两部分组成
>   - 表结构：==列信息--->字段==
>   - 表数据：==行信息--->值==

SQL入门

SQL入门导图

数据库的分类（SQL概念相关）

> - 数据库分类
>   - 关系型数据库：Oracle、DB2、SQ server、MySQL、MS Access
>   - 网状、树状、面向对象数据库
>   - 非关系型数据库有：MongoDB 、Rides、Memcached
> - 数据库操作的命令类型
>   - DDL：（数据库的定义语言，创建、删除、修改，数据库、数据表）create/alter/drop table、index; create/drop view;
>   - DML: (数据库操作语言，操作数据库中的数据)，insert、update、delete。
>   - DQL:数据查询语言，对数据进行查询（select）
>   - DCL:（数据控制语言）创建对象，控制用户权限，用来控制数据库组件的存取（事务：begin，commit，rollback）
> - 表结构
>   - 字段
>   - 记录
>   - 列
>   - 主键
>   - null值
>   - 数据类型

    - 字段长度
- 基本数据类型
    - 字符串类型（char固定长度，varchar可变长度，用多少给多少）
    - 数值类型
    - 日期和时间类型
- 规划
- 完整性约束
    - 实体完整性：又称为行完整性，要求表中不能存在完全相同的行，而且每行都要具有一个非空且又不重复的主键。可以用主键子句或者主键短语来定义
- 建表时定义主键
    - 添加主键
    - 参照完整性：又称引用完整性，指标简的规则，卓用于有关联的两张或两张以上的表，通过使用主键和外键（或为一键）之间的关系，使表中键值在相关表中保持一致。可用外键子句来定义。
- 建表时定义外键
    - 添加外键
- 用户自定义完整性：指针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。属性约束
    - 非空约束：`alter table 表名 alter column name varchar(20) not null`
    - 唯一性约束 `alter table 表名 add constraint UQ_表名_列名 unique(列)`
    - 检查约束：`alter table 表名 add constraint CK_表名_列名 check(age>5)`
    - 默认约束：`alter table 表名 add constraint DF_表名_列名 default('男') for gender`
- 删除完整性:删除约束，全局约束：包括基于元组的检查之句（check）和断言 alter table 表名 drop constraint DF*表名列*

SQL中的函数

- sql函数可以完成一些复杂的功能，但是一般不使用sql来完成复杂的查询，复杂的业务全部交给java来完成。sql只是存储功能本身是一个检索机制，否则会浪费大量资源。
- 语法与存储过程很类似，功能类似，区别是函数必须有返回值。参数只能输入参，存储过程不一定有返回值，参数可以是入参 也可以是出参。

```
      create  function   函数名（参数列表）
      returns  返回值的数据类型
      routine_body
      ----------------------------------------------------------------------------
----------------
      //实例
      create  function  name_of_user( uid  int )
      returns   varchar(11)
      begin
      return(
          select   name  from  user   where  id  = uid
      );
```

```
end;
调用函数：select  函数名（参数）
删除函数：drop  function  函数名
```

- 数学函数

  - abs（）求绝对值，会改变结果但是不会改变数据库的数据， ++select abs (score) from student;++
  - floor（num），向下取整；返回小于参数的最大整数： ++select floor(score) from student where id = 1;++
  - eil（num），向上取整 :返回大于参数的最小整数； ++select ceil (score) from student where id = 2;++
  - 字符串函数：
  - insert（s1,index,length,s2）:s1表示要替换的字段；index表示要替换的开始位置；length表示要替换的长度;s2表示要替换的内容，select insert(name, 1,1,"小红") from student；
  - upper（name），lcase（name），将字母值变成大写。

    ```
    select  upper(name)    from  studet;
    select  ucase(name)    from  student
    ```

  - lower(), lcase() 将字母变为小写

```
select  lower(name)   from  student  ;
select  lcase(name )   from stedent;
```

  - lefe (s,len) :返回s字符串的前 len个字符。++==select lefte (name,2) from student;==++
  - right (s, len) 返回s 字符串的后 len 可字符 ++==select right (name,2) from student;==++
  - substring (s, index ,len) :截取s 字符串，从index 位置开始，长度为len。++==select substring (name , 2, 2) from student id = 6;==++
  - reverse (); 反序输出; ++==select reverse (name) from student where id = 6;==++
  - 日期函数：java.util;中提供了data（）类， 而在sql中 也提供了相应的方法。
  - curdate() current_date() 获取当前日期
  - curtime() current_time() 获取当前时间
  - now() 获取当前日期+时间： select now();
  - datediff(d1,d2) d1和d2 相隔的天数，如果出现负数表示推的顺序不同； ==select datediff("2017-5-1","2018-3-14") frome student.;==
  - adddate(d,n) 返回d日期之后的n天的日期； ==select adddate("2018-1-1",33) from student ;==
  - subdate(d,n) 返回d日期之前的n天日期== select subdate("2018-1-1",33) from student ;==
  - 聚合函数（前两个常用 ）
  - count（id）根据某个字段统计出数据的条数。或者说是记录数（当前数据库保存了多少条数据） 统计函数。
  - sum（score） 计算某个字段值的总和

- avg(score) 计算某个字段的总和的平均值
- max（score）某个字段值得最大值
- min（score ） 求某个字段值的最小值。

SQL运算符

- 算术运算符
- 执行运算符：加减乘除； select score + 10 from student where id = 5;
- 比较运算符：大于 小于 等于 不等于,返回的是布鲁值0/1，1表示为true 0表示false；elect score <10 from student where id= 7;
- 逻辑运算符：与 或 非。select score >60 & age <20 from student where id = 8;
- 位运算符：按位与 按位或 按位取反
- 特殊运算符

SQL执行顺序

1、Form(left tab); **2**、On ; **3**、 join ; **4**、where；**5**、Group by; **6**、Having; **7**、select; **8**、distinct; **9**、Order by; **10**、

数据库的设计

数据库的设计是项目成功的最主要环节，（项目的重要）数据库中实现相关关系，++主键 和外键不是必须有的，但是为了安全尽量使用。++

- 主键
  - （核心）将一个字段设为主键。该字段的值是每一行记录的唯一标识。
  - 给表中某一个字段，添加主键属性，字段的值就是该条的记录唯一标识。就如同每个人的身份证号一样唯一的标识。
  - 主键不能有重复值，默认情况下每张表都有主键，++一张表只能有一个主键，所谓一张表有多个主键指的是联合主键。联合主键的特点：用多个字段作为一张表的主键++
  - 主键生成原则：代理主键，与业务无关的字段，仅仅是用来标识一行数据： ==goods: name ,price,date.额外添加一个字段 id 作为代理主键，但是它与业务无关,一般将该字段设置为int 类型，int类型占用空间小，检索更快==
  - 主键自增：在添加一条记录的时后，不需要设置主键的值，自动生成新数据的主键，每次加一。

```
create table student (
  id int primary  key  auto_increment,
  name  varchar(11)
)
```

- 外键:（由主键和外键共同组成 java和数据库的对应关系）给表中的一个字段添加一个外键属性（从表），让它由相应的主键约束（主表），与其他表的主键构成关联关系，主键约束外键。

```
    //给表中添加外键
      create table  student (
      id  int  primary  key  auto_increment, //主键设置每张表都有，并且在代码的第一行。
```

```
      name varchar(11)
      cid  int,//在创建表的字段的时候也要创建外键字段
      foreign  key( cid ) references   classes( id )   //cid为外键  , id为主键,
外键受到主键约束。
      )
    ------------------------------------------------------------------
      //修改主键或外键
       alter  table  user add  company_id  int;  //添加一个外键字段。
       alter   table  user  add   foreign   key ( company_id  )  references
company( id );
    ------------------------------------------------------------------------
      // 删除外键
      alter   table   orders  drop   foreign  key  orders_ibfk_1
    ------------------------------------------------------------------------
      表创建 在先，主外键设置在后
      //创建两张单表，没有任何关联关系
      create  table   classes  (
        //主键id
        id  int  primary  key   auto_increment,
        name  varchar(11)
        ) ;
        create  table  student  (
        id   int  primary   key  auto_increment,
        name  varchar(11)
        )
       //给从表添加 外键 cid，使用修改表关键字 alter
          alter  table  student  add   cid   int;
       //将cid 字段设置为外键，被classes 的 id约束。
         alter  table  student   add  foreign  key( cid )  references
classes( id );
```