

---

## 数据库的三大范式

### 1、第一范式（1NF）

在任何一个关系数据库中，第一范式（1NF）是对关系模式的基本要求，不满足第一范式（1NF）的数据库就不是关系数据库。

所谓第一范式（1NF）是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多值，即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性，就可能需要定义一个新的实体，新的实体由重复的属性构成，新实体与原实体之间为一对多关系。

在第一范式（1NF）中表的每一行只包含一个实例的信息。简而言之，第一范式要求数据表中的每一列（每个字段）必须是不可拆分的最小单元。

### 2、第二范式（2NF）

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或行必须可以被惟一地区分。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。

第二范式（2NF）要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。简而言之，第二范式要求表中的所有列，都必须依赖于主键，而不能有任何一列与主键没有关系。

### 3、第三范式（3NF）

满足第三范式（3NF）必须先满足第二范式（2NF）。第三范式（3NF）要求一个数据库表中不包含其它表中已包含的非主关键字信息。简而言之，第三范式要求表中的每一列只与主键直接相关而不是间接相关，表中的每一列只能依赖于主键。

## TCP 和 UDP 的区别及其适用场景

### 首先说一下什么是 TCP 和 UDP：

TCP 是传输控制协议，提供的是面向连接、可靠的字节流服务。

UDP 是用户数据报协议，是一个简单的面向数据报的运输层协议。

## TCP 和 UDP 的区别：

- TCP 面向连接的传输层协议，UDP 无连接
- TCP 是可靠交付，UDP 是尽最大努力交付
- TCP 面向字节流，UDP 面向报文
- TCP 是点对点连接的，UDP 一对一，一对多，多对多都可以
- TCP 适合用于网页，邮件等，UDP 适合用于视频，语音广播等

## TCP 和 UDP 的适用场景：

整个数据要准确无误的传递给对方，这往往用于一些要求可靠的应用，比如 HTTP、HTTPS、FTP 等传输文件的协议，POP、SMTP 等邮件传输的协议。

当对网络通讯质量要求不高的时候，要求网络通讯速度能尽量的快，比如视频、广播等，这时就可以使用 UDP。

## 说一下 Spring 的核心模块

- Spring Core【核心容器】：核心容器提供了 Spring 的基本功能。核心容器的核心功能是用 IOC 容器来管理类的依赖关系。
- Spring AOP【面向切面】：Spring 的 AOP 模块提供了面向切面编程的支持。SpringAOP 采用的是纯 Java 实现，采用基于代理的 AOP 实现方案，AOP 代理由 IOC 容器负责生成、管理，依赖关系也一并由 IOC 容器管理。
- Spring ORM【对象实体映射】：提供了与多个第三方持久层框架的良好整合。
- Spring DAO【持久层模块】：Spring 进一步简化 DAO 开发步骤，能以一致的方式使用数据库访问技术，用统一的方式调用事务管理，避免具体的实现侵入业务逻辑层的代码中。
- Spring Context【应用上下文】：它是一个配置文件，为 Spring 提供上下文信息，提供了框架式的对象访问方法。
- Spring Web【Web 模块】：提供了基础的针对 Web 开发的集成特性。
- Spring MVC【MVC 模块】：提供了 Web 应用的 MVC 实现。Spring 的 MVC 框架并不是仅提供一种传统的实现，它提供了一种清晰的分离模型。

## (转发) forward 与 (重定向) redirect 的区别

- forward 是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器。浏览器根本不知道服务器发送的内容从哪里来的，所以它的地址栏还是原来的地址。
- redirect 是服务端根据逻辑，发送一个状态码，告诉浏览器重新去请求那个地址，所以地址栏显示的是新的 URL。
- forward 转发页面和转发到的页面可以共享 request 里面的数据。
- redirect 不能共享数据。

- 
- `redirect` 不仅可以重定向到当前应用程序的其他资源，还可以重定向到同一个站点上的其他应用程序中的资源，甚至是使用绝对 URL 重定向到其他站点的资源。
  - `forward` 只能在同一个 Web 应用程序内的资源之间转发请求。
  - `forward` 是服务器内部的一种操作。
  - `redirect` 是服务器通知客户端，让客户端重新发起请求。
  - `forward` 一般用于用户登陆的时候根据角色转发到相应的模块。
  - `redirect` 一般用于用户注销登陆时返回主页面和跳转到其它的网站等。
  - `forward` 效率高。
  - `redirect` 效率低。

## redis 常用的五种数据类型

### 1.String ( 字符串 )

String 是简单的 key-value 键值对，value 不仅可以是 String，也可以是数字。它是 Redis 最基本的数据类型，一个 redis 中字符串 value 最多可以是 512M。

### 2.Hash ( 哈希 )

Redis hash 是一个键值对集合，对应 Value 内部实际就是一个 HashMap，Hash 特别适合用于存储对象。

### 3.List ( 列表 )

Redis 列表是简单的字符串列表，按照插入顺序排序。你可以添加一个元素到列表的头部（左边）或者尾部（右边）。

底层实现为一个双向链表，即可以支持反向查找和遍历，更方便操作，不过带来了部分额外的内存开销，Redis 内部的很多实现，包括发送缓冲队列等也都是用的这个数据结构。

### 4.Set ( 集合 )

Redis 的 Set 是 String 类型的无序集合，它的内部实现是一个 value 永远为 null 的 HashMap，实际就是通过计算 hash 的方式来快速排重的，这也是 set 能提供判断一个成员是否在集合内的原因。

### 5.zset(有序集合)

Redis zset 和 set 一样也是 String 类型元素的集合，且不允许重复的成员，不同的是每个元素都会关联一个 double 类型的分数，用来排序。