数据库系统DBS由5部分组成：硬件系统、数据库集合、数据库管理系统

及相关软件、数据库管理员和用户

DB(数据库）：Database；

DBMS（数据库管理系统）：Database Management System；

DBS（数据库系统）：Data Base System。

DBS = DB + DBMS+ 软件组成。

概念模型是现实世界的抽象反映，它表示实体类型及实体间的联系，是独立于计算机系统的模型，是现实世界到机器世界的一个中间层次。

数据库设计包括六个主要步骤：  
1、需求分析：了解用户的数据需求、处理需求、安全性及完整性要求；  
2、概念设计：通过数据抽象，设计系统概念模型，一般为E-R模型；  
3、逻辑结构设计：设计系统的模式和外模式，对于关系模型主要是基本表和视图；  
4、物理结构设计：设计数据的存储结构和存取方法，如索引的设计；  
5、系统实施：组织数据入库、编制应用程序、试运行；  
6、运行维护：系统投入运行，长期的维护工作。  
实体-联系方法（Entity-Relationship Approach），也叫E-R模型

1、模式（Schema）

定义：也称逻辑模式（概念模式），是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。

理解：

① 一个数据库只有一个模式；

② 是数据库数据在逻辑级上的视图；

③ 数据库模式以某一种数据模型为基础；

④ 定义模式时不仅要定义数据的逻辑结构（如数据记录由哪些数据项构成，数据项的名字、类型、取值范围等），而且要定义与数据有关的安全性、完整性要求，定义这些数据之间的联系。

2、外模式（External Schema）

定义：也称子模式（Subschema）或用户模式，是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

理解：

① 一个数据库可以有多个外模式；

② 外模式就是用户视图；

③ 外模式是保证数据安全性的一个有力措施。

3、内模式（Internal Schema）

定义：也称存储模式（Storage Schema），它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式（例如，记录的存储方式是顺序存储、按照B树结构存储还是按hash方法存储；索引按照什么方式组织；数据是否压缩存储，是否加密；数据的存储记录结构有何规定）。

理解：

① 一个数据库只有一个内模式；

② 一个表可能由多个文件组成，如：数据文件、索引文件。

它是数据库管理系统(DBMS)对数据库中数据进行有效组织和管理的方法

其目的有：

② 为了减少数据冗余，实现数据共享；

② 为了提高存取效率，改善性能。

I、关系数据库设计范式介绍  
1.1 第一范式（1NF）无重复的列  
   
      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，同一列中不能有多个值，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，第一范式就是无重复的列。

说明：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。

1.2 第二范式（2NF）属性完全依赖于主键[消除部分子函数依赖]  
   
      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。例如员工信息表中加上了员工编号（emp\_id）列，因为每个员工的员工编号是惟一的，因此每个员工可以被惟一区分。这个惟一属性列被称为主关键字或主键、主码。  
        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，第二范式就是属性完全依赖于主键。

1.3 第三范式（3NF）属性不依赖于其它非主属性[消除传递依赖]

            满足第三范式（3NF）必须先满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。例如，存在一个部门信息表，其中每个部门有部门编号（dept\_id）、部门名称、部门简介等信息。那么在的员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表，则根据第三范式（3NF）也应该构建它，否则就会有大量的数据冗余。简而言之，第三范式就是属性不依赖于其它非主属性。

1.having只能用在group by之后，对分组后的结果进行筛选，筛选行(即使用having的前提条件是分组)。

2.where肯定在group by 之前

3.where后的条件表达式里不允许使用聚合函数，而having可以。

事务注解 @Transactional一般的话都加在service层，问了一些刚入行的朋友为什么，他们给的答案很模糊，所以自己去搜了下，下面是我的理解。

什么是事务？

在数据库中,所谓事务是指一组逻辑操作单元即一组sql语句。当这个单元中的一部分操作失败,整个事务回滚，只有全部正确才完成提交。判断事务是否配置成功的关键点在于出现异常时事务是否会回滚

事务四大特性

1. 原子性（Atomicity）   
原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。

2. 一致性（Consistency）   
事务必须使数据库从一个一致性状态变换到另外一个一致性状态。(数据不被破坏

3.隔离性（Isolation）   
事务的隔离性是指一个事务的执行不能被其他事务干扰.

4.持久性（Durability）   
持久性是指一个事务一旦被提交，它对数据库中数据的改变就是永久性的.即使系统重启也不会丢失.

结合事务的特点，为什么加在service层就很好解释了。如果我们的事务注解@Transactional加在dao层，那么只要与数据库做增删改，就要提交一次事务，如此做事务的特性就发挥不出来，尤其是事务的一致性，当出现并发问题是，用户从数据库查到的数据都会有所偏差。   
一般的时候，我们的service层可以调用多个dao层，我们只需要在service层加一个事务注解@Transactional，这样我们就可以一个事务处理多个请求，事务的特性也会充分的发挥出来。

数据库操作：

DML（data manipulation language）：  
       它们是SELECT、UPDATE、INSERT、DELETE，就象它的名字一样，这4条命令是用来对 [数据库](http://lib.csdn.net/base/mysql" \t "https://www.nowcoder.com/test/question/_blank) 里的数据进行操作的语言   
DDL（data definition language）：   
       DDL比DML要多，主要的命令有CREATE、ALTER、DROP等，DDL主要是用在定义或改变表（TABLE）的结构，数据类型，表之间的链接和约束等初始化工作上，他们大多在建立表时使用   
DCL（Data Control Language）：

       是数据库控制功能。是用来设置或更改数据库用户或角色权限的语句，包括（grant,deny,revoke等）语句。在默认状态下，只有sysadmin,dbcreator,db\_owner或db\_securityadmin等人员才有权力执行DCL

层次型、网状型和关系型数据库划分原则是：数据之间的联系

MySql中的权限分为五个级别：Global Level、Database Level、Table Level、Column Level、Routine Level

1、Global Level：

Global Level 的权限控制又称为全局权限控制，所有权限信息都保存在mysql.user 表中。

Global Level 的所有权限都是针对整个mysqld 的，对所有的数据库下的所有表及所有字段都有效。

                       GRANT SELECT,UPDATE,DELETE,INSERT ON \*.\* TO 'def'@'localhost';

2、Database Level

Database Level 其作用域即为所指定整个数据库中的所有对象。

                             GRANT ALTER ON test.\* TO 'def'@'localhost';

                              GRANT DROP ON \* TO 'def'@'localhost';

3、Table Level

Table Level 的权限作用范围是授权语句中所指定数据库的指定表。

                      GRANT INDEX ON test.t1 TO 'abc'@'%.jianzhaoyang.com';

Table Level 的权限由于其作用域仅限于某个特定的表，所以权限种类也比较少，仅有  
                      ALTER，CREATE，DELETE，DROP，INDEX，INSERT，SELECT UPDATE 这八种权限。

4、Column Level

Column Level 的权限作用范围就更小了，仅仅是某个表的指定的某个（活某些）列。

                      GRANT SELECT(id,value) ON test.t2 TO 'abc'@'%.XXX';

Column Level 级别的权限仅有 INSERT ， SELECT 和 UPDATE 这三种。

5、Routine Level

Routine Level 主要针对的对象是procedure 和function 这两种对象，在授予Routine Level 权限的时候，需要指定数据库和相关对象

                        GRANT EXECUTE ON test.p1 to 'abc'@'localhost';

Routine Level 的权限主要只有EXECUTE 和ALTER ROUTINE 两种