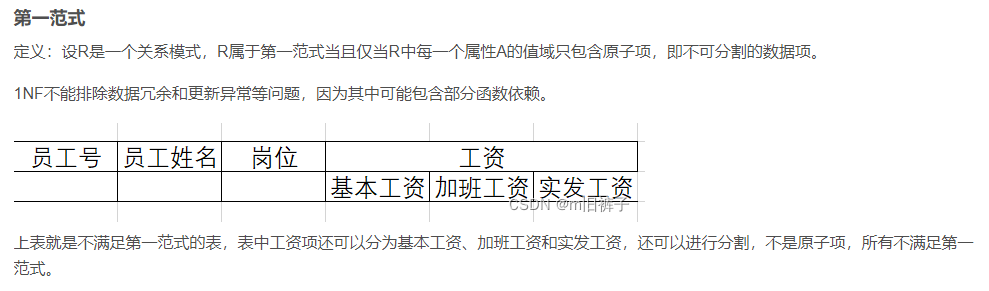
1. **数据库三大范式：（第一，第二，第三，巴斯科德范式，第五，第六范式）**

第一范式：

数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项



第二范式：

在第一范式的基础上，且每个非主属性都完全依赖于候选码。消除部份依赖



分析第二范式之前，先来了解几个概念。

元组：表中的一行即为一个元组，对应存储文件中的一个记录值。

属性：表中的列称为属性，每一列有一个属性名。属性名相当于记录中的数据项或字段值。

候选码：属性或属性组合，其值能够唯一标识一个元组。

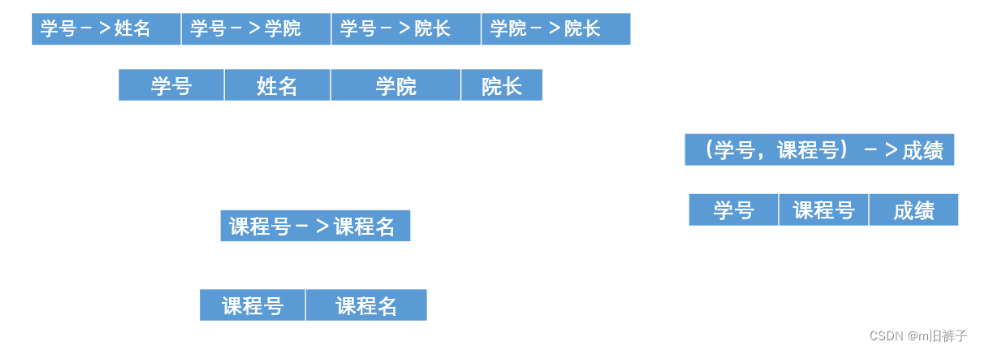
主码（主键）：在一个关系中可能有多个候选码，从中选择一个作为主码。

部分函数依赖：如果X－＞Ｙ，但Ｙ不完全函数依赖于Ｘ，则称Y对Ｘ部分函数依赖。部分函数依赖就是一个组合里任意真子集能够决定依赖关系，例如（学号,课程号）---->姓名这个组合关系的函数依赖中，单一个学号也能够决定姓名了，所以这就是部分函数依赖。

传递函数依赖：在Ｒ（Ｕ，Ｆ）中，如果Ｘ－＞Ｙ，Ｙ－＞Ｚ，则称Z对Ｘ传递依赖。

从上图第二行可以看出姓名、学院、院长都部分函数依赖于（学号，课程号）中的学号，课程名部分函数依赖于（学号、课程号）中的课程名，非主属性不都完全函数依赖于候选码，所以不满足数据库第二范式。

而想要上表满足第二范式，我们就需要对表进行拆解。把表拆解成以上三个表之后，可以看出，三个表中非主属性都完全函数依赖于候选码，且具有无损连接性和保持函数依赖性，满足了数据库第二范式。



第三范式：

在第二范式的基础上。且每个非主属性都非传递依赖于候选码。

在2NF基础上，任何非主属性不依赖于其它非主属性（在2NF基础上消除传递依赖）



1. **数据库事务**

数据库事务( transaction)是访问并可能操作各种数据项的一个数据库操作序列，要么全部执行，要么都不执行

特性：

原子性：

事务的全部操作是不可分割的，要么全部完成，要么都不完成

一致性：

一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态

隔离性：

每个事务的操作应该与其他事务的操作相互独立，互不干扰。即使多个事务同时并发执行，每个事务也应该感觉不到其他事务的存在。

持久性：

事务一旦提交(或者说已经成功执行)，其对数据库的修改应该是永久性的，即使系统发生故障，数据也不会丢失

1. **存储过程**

存储过程其实就是能完成一定操作的一组SQL语句

那为什么要用存储过程呢？

1.存储过程只在创造时进行编译，以后每次执行存储过程都不需再重新编译，而一般SQL语句每执行一次就编译一次,所以使用存储过程可提高数据库执行速度。

2.当对数据库进行复杂操作时，可将此复杂操作用存储过程封装起来与数据库提供的事务处理结合一起使用。

3.存储过程可以重复使用,可减少数据库开发人员的工作量。

4.安全性高,可设定只有某些用户才具有对指定存储过程的使用权

创建存储过程：

DELIMITER $$

CREATE PROCEDURE testa()

BEGIN

SELECT \* FROM t\_department;

END $$;

DELIMITER

执行存储过程 CALL testa();

1. **乐观锁和悲观锁**

乐观锁：乐观锁在操作数据时非常乐观，认为别人不会同时修改数据。因此乐观锁不会上锁，只是在执行更新的时候判断一下在此期间别人是否修改了数据：如果别人修改了数据则放弃操作，否则执行操作。

悲观锁：悲观锁在操作数据时比较悲观，认为别人会同时修改数据。因此操作数据时直接把数据锁住，直到操作完成后才会释放锁；上锁期间其他人不能修改数据。

1. **视图**

视图，虚拟表，从一个表或多个表中查询出来的表，作用和真实表一样，包含一系列带有行和列的数据。视图中，用户可以使用SELECT语句查询数据，也可以使用INSERT，UPDATE，DELETE修改记录，视图可以使用户操作方便，并保障数据库系统安全。

优点：

简单化，数据所见即所得。

安全性，用户只能查询或修改他们所能见到得到的数据。

逻辑独立性，可以屏蔽真实表结构变化带来的影响。

缺点：

性能相对较差，简单的查询也会变得稍显复杂。

修改不方便，特变是复杂的聚合视图基本无法修改。

1. **事务的隔离级别**

read uncommited：读取未提交内容

read committed：读取提交内容

repeatable read：可重读

serializable：可串行化

1. MVCC（Multiversion concurrency control）多版本并发控制

它是一种数据库事务管理技术，用于解决并发访问数据库的问题。MVCC 通过创建多个版本的同一数据，每个版本与一个事务关联，来实现并发控制。

1. 触发器

触发器是基于事件的，主要的事件也就是MySQL的增删改操作，即insert,delete,update。

触发器在单表的命名空间内，所以同一个表的触发器名称需要不同。不同表可以有相同的触发器名称。

CREATE TRIGGER 触发器名 BEFORE|AFTER 触发事件

ON 表名 FOR EACH ROW

BEGIN

执行语句列表

END

FOR EACH ROW表示任何一条记录上的操作满足触发事件都会触发该触发器