

# 数据库的设计思维

前面的话：这个模块不涉及代码演示，只谈论一些思想和实战经验，建议学习了解MySQL的基础知识再来看。

形容数据库的专业术语罢了

## 数据库概念

### 关系（关系型数据库）

两张表共有的字段去确定数据的完整性（大白话：两张表有相同的字段（包括主键外键），使得两张表在数据内容上有了联系，提高数据浏览的效率）

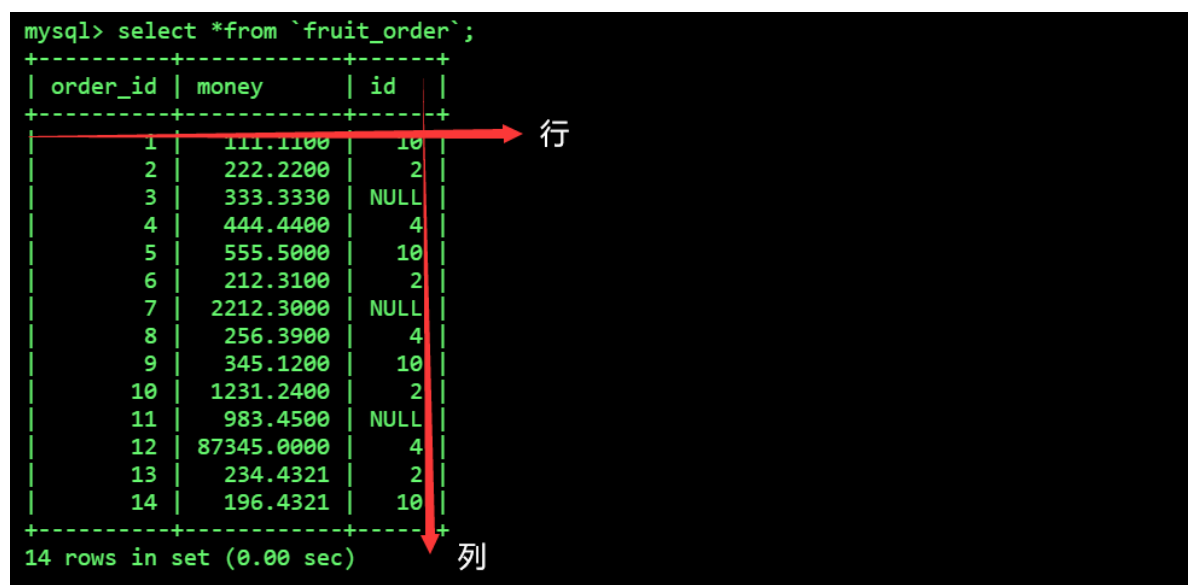
### 实体：行 属性：列

直接看图在数据表中

```
mysql> select *from `fruit_order`;
```

order_id	money	id
1	111.1100	10
2	222.2200	2
3	333.3330	NULL
4	444.4400	4
5	555.5000	10
6	212.3100	2
7	2212.3000	NULL
8	256.3900	4
9	345.1200	10
10	1231.2400	2
11	983.4500	NULL
12	87345.0000	4
13	234.4321	2
14	196.4321	10

14 rows in set (0.00 sec)

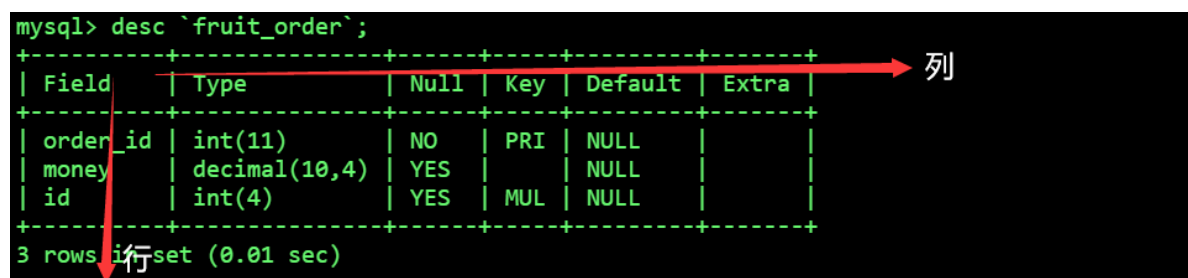


在结构表中

```
mysql> desc `fruit_order`;
```

Field	Type	Null	Key	Default	Extra
order_id	int(11)	NO	PRI	NULL	
money	decimal(10,4)	YES		NULL	
id	int(4)	YES	MUL	NULL	

3 rows in set (0.01 sec)



### 实体与实体之间的关系

大白话：多张表中一条数据记录与一条数据记录的关系，甚至多条数据记录的关系（方便理解专业术语还是要说的）

方便理解我们程序猿开发程序的例子来讲解

## 一对一

一个程序员负责一个项目

## 一对多

一个程序员负责多个项目

## 多对一

多个程序员负责一个项目

## 多对多

乱交多个程序员负责多个项目

###

---

## 数据冗余

冗 - 百度汉语

	读音: [rǒng] 
部首: 冫	五笔: PMB
释义: 1. 多余的。 2. 烦琐; 繁忙。 3. 繁忙的事。	

数据冗余发生在数据库系统中，指的是一个字段在多个表里重复出现，虽然这样会增加服务器性能损耗，增加维护成本，但是由于我们实际的查询以及使用的需要我们还是需要这种情况发生。

数据冗余的缺点：

1. 造成维护成本的提高
2. 造成资源浪费

数据冗余的优点：

1. 提高查询性能
2. 数据恢复，如建立备份文件以备正式文件被破坏时恢复
3. 数据核查，如设立数据校验位可以检查数据在存贮、传输等过程中的改变
4. 数据使用的便利，如为了查看数据的直观，使用数据的方便、高效
5. 减少数据通讯开销，如分布式数据库在不同场地重复

冗余信息被作为加快数据访问速度的手段应用最多的情况一般不是在一个表里设置冗余字段，而是在很多海量数据的数据仓库里把很多小粒度的数据计算成为以一天、一周、一个月作为更大粒度统计单位的冗余信息表或者指标信息表，而直接访问这些大粒度的冗余数据，比直接访问最小粒度的数据进行统计效率可能快上几千倍。

数据冗余或者信息冗余是生产、生活所必然存在的行为，没有好与不好的总体倾向。应用者根据系统设计的经验和自己的实际需求去判断，应选择在满足自己系统业务运转要求的前提下“成本”更低的方式。

---

# 数据规范 Codd

---

共有6条，实际中我们能用3条范式，还要结合实际情况，这里只介绍三种，感兴趣可自行查询。

## 确保每列的原子性

大白话：确保每一列不能再分，也就是字段不能在分了，详细说就是字段数据的种类要单一。比如说时间段，有时间开始也有时间结束，这个字段就可以再分，开始时间时间为一个字段，结束时间为一个字段。题外话：别抬杠啊，别说开始时间还能分小时分钟秒。

## 非键字段必须依赖于键字段

大白话：什么样表，放什么样的数据。和表设定没关系的数据不要放入表中，说白了就是问你什么答什么，中尼玛别他妈的没事找事。

## 消除传递依赖

大白话：多余的字段，如果没有实际需求的话要干掉，结合上面数据冗余的情况去理解。

**最后要说的是数据库的设计思路要侧重于实际项目实际的需要不要教条化，要灵活，最终的目标是服务好客户的需要。**

---