表设计软件：一般都用 powerdesign

**1、数据库设计的实用原则是：**

* 在数据冗余和处理速度之间找到合适的平衡点。原则是相对的，不是绝对的。
* 做表设计，读懂需求就对了。先不要管性能，先实现需求。表设计好了，写SQL的时候再考虑该合并，合并，该拆分，拆分。另外最关键的就是搞清楚一对一还是一对多。

**2、表设计范式：**

* 保证每列的原子性，不可分解，意思表达要清楚，不能含糊，高度概括字段的含义，能用一个字段表达清楚的绝不使用第二个字段，可以用两个字段表达清楚的绝不使用一个字段
* 表及其字段之间的关系, 应尽量满足第三范式。但是，满足第三范式的数据库设计，往往不是最好的设计。为了提高数据库的运行效率，常常需要降低范式标准：适当增加冗余，达到以空间换时间的目的。  
  例如：表内有商品单价和数量字段，我们设计的时候加上了一个金额的字段，这虽然违背了数据库设计的第三范式，但以查询统计，这能大大提高查询的速度，这就是空间换时间的做法。

**3、表关系设计：**

* 最好做好静态表和动态表的分离。这里解释一下静态表和动态表的含义，静态表：存储着一些固定不变的资源，比如城市/地区名/国家。动态表：一些频繁修改的表
* 不要有null值，有null值的话，数据库在进行索引的时候查询的时间更久，从而浪费更多的时间！建议可以为null的值转换成not null default ‘’
* 2张表的多对多的表关系，最好设计成3张表，即增加一张中间表，之前的两张表和中间表的关系是一对多的关系。
* 建表的时候，字段长度尽量要比实际业务的字段大3-5个字段左右（考虑到合理性和伸缩性），最好是2的n次方幂值。不能建比实际业务太大的字段长度，这是因为如果字段长度过大，在进行查询的时候索引在B-Tree树上遍历会越耗费时间，从而查询的时间会越久；但是绝对不能建小，否则mysql数据会报错，程序会抛出异常；
* 对于频繁修改的字段（一般是指状态类字段）最好用独立的数字或者单个字母去表示，不用使用汉字或者英文
* 数据库不要存储任何资源文件，比如照片/视频/网站等，可以用文件路径/外链用来代替，这样可以在程序中通过路径，链接等来进行索引
* 关系映射：多对一或者一对多的关系,关联一张表最好通过id去建立关系，而不是去做重复数据，这样做最大的好处就是中间的关系表比较清楚明白。
* 通过单一字段表示该行记录是否可用，通过一个单一字段去控制表是否可用，比如通常起名为isVaild，预制的含义为0为有效，1为无效，这样便于以后我们去剔除数据或者重整数据，使其成为boolean性质的数据 更加便于我们去操控。
* 预留备用字段：在设计一张表的时候应该预制2到3个空白字段，用于以后的扩展，因为你也不是确定这张表以后不会扩展。

**4、主键的使用：**

* 主键不要与业务逻辑有所关联，最好是毫无意义的一串独立不重复的数字，常见的比如UUID或者将主键设置为Auto\_increment;
* 主键：主键可以是一无物理意义的数字串, 由程序自动加1来实现。也可以是有物理意义的字段名或字段名的组合。不过前者比后者好。当PK是字段名的组合时，建议字段的个数不要太多，多了不但索引占用空间大，而且速度也慢。
* 一个表中组合主键的字段个数越少越好。因为主键的作用，一是建主键索引，二是做为子表的外键，所以组合主键的字段个数少了，不仅节省了运行时间，而且节省了索引存储空间；

**5、提高数据库运行效率的办法**

**在系统硬件和系统软件条件确定的情况下，提高数据库系统的运行效率的办法是：**

* 在数据库物理设计时，降低范式，增加冗余，少用触发器, 多用存储过程。
* 当计算非常复杂、而且记录条数非常巨大时（例如一千万条），复杂计算要先在数据库外面，以文件系统方式用C++语言计算处理完成之后，最后才入库追加到表中去。这是电信计费系统设计的经验。
* 发现某个表的记录太多，例如超过一千万条，则要对该表进行水平分割。水平分割的做法是，以该表主键PK的某个值为界线，将该表的记录水平分割为两个表。若发现某个表的字段太多，例如超过八十个，则垂直分割该表，将原来的一个表分解为两个表。
* 对数据库管理系统DBMS进行系统优化，即优化各种系统参数，如缓冲区个数。
* 在使用面向数据的SQL语言进行程序设计时，尽量采取优化算法。
* 如果进行更新表的数据量较大，例如，更新的字段的值，需要重新从子表查询，且是全表查询，  
   如：UPDATE TABLE\_A A SET COLUMN\_A = (SELECT COLUMN\_A FROM TABLE\_A\_B B WHERE A.COLUMN\_B = B.COLUMN\_B);  
   其中每更新一行的字段，都会把括号内的sql执行一次，对于数据量大的表，其IO流会非常大，更新时间会较长，建议使用plsql过程语言，或使用merge into 语句进行更新。  
   merge into 语句使用的是多块读，优点就是可以批量提交，缺点就是不能并行更新。

总之，要提高数据库的运行效率，必须从数据库系统级优化、数据库设计级优化、程序实现级优化，这三个层次上同时下功夫。

以上的数据库设计经验，读者不要生搬硬套，要活学活用，适时使用。

作者：JustBull  
链接：https://www.jianshu.com/p/581859fb88a8  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。