# 数据库设计

数据库设计指根据用户的需求，以需求说明书为基础，以系统设计说明书为指导，设计数据库的结构和建立数据库的过程，使之能够有效的存储数据。

1. 1. 数据库选取

根据系统设计的要求，选取与系统平台相适应的数据库系统及其版本。一般选取Sqlserver、Oracle或MySql。

* 1. 设计原则

1. 把具有同一个主题的数据存储在一个数据表中，“一表一用”。
2. 尽量消除冗余，提高访问数据库的速度。
3. 一般要求数据库设计达到第三范式，多对多，最大限度消除了数据冗余、修改异常、插入异常、删除异常，基本满足关系规范化的要求。
4. 关系数据库中，各个数据表之间关系只能为一对一和一对多的关系。对于多对多的关系必须转换为一对多的关系来处理。
5. 设计数据表结构时，应考虑表结构的动态适应性。
   1. 设计规范

只有当每个字段只有一个值且每个表有正确的字段集，这个数据库才是规范的。根据设计原则的要求，一般数据库设计要达到第三范式，也就是说至少满足三个范式的要求。

* + 1. 第一范式（1NF）

释义：表中所有字段只能存储单值。比如，不能在姓名字段中存储两个姓名。

* + 1. 第二范式（2NF）

释义：表中的每个非键字段由整个主键确定，且不能由主键自身的一部分确定。

比如下表，学员表：

| Id | Name | Mobile | Age | Course\_No | Course |
| --- | --- | --- | --- | --- | --- |
| 1 | 张三 | 18603712222 | 19 | 01 | 前端 |
| 2 | 李四 | 18603712255 | 22 | 02 | 服务端 |
| 3 | 王五 | 18603713232 | 21 | 03 | 数据库 |
| 1 | 张三 | 18103715678 | 19 | 02 | 服务端 |

上表中的Id和Course\_No合并作为其主键，主键应该确定所有的非键字段。作为主键一部分的Id，确定了Name、Age和Phone，但确定不了Course。同时，上表也会产生更新问题，表中张三有两个电话号码，更新时不确定应该更新哪个。

为了解决此问题，需进行分表设计，将上面的一个表分为两个表：

学生表（Student）：

| Id | Name | Mobile | Age |
| --- | --- | --- | --- |
| 1 | 张三 | 18603712222 | 19 |
| 2 | 李四 | 18603712255 | 22 |
| 3 | 王五 | 18603713232 | 21 |

课程表(Course)

| Id | Name |
| --- | --- |
| 1 | 前端 |
| 2 | 服务端 |
| 3 | 数据库 |

学生课程关系表，一对多的关系：

| Id | StudentId | Course Id |
| --- | --- | --- |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 1 | 2 |

* + 1. 第三范式（3NF）

释义：表中不存在可以确定其他非键字段的非键字段。

比如用户表（Xuser）：

| Id | Name | Email | Mobile | Password | DatetimeIn | DatetimeOut |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 张三 | 100001@163.com | 18603712222 | 123456 | 2019-01-02 09：00 | 2019-01-02 18:30 |
| 2 | 李四 | 100002@163.com | 18603712255 | 432122 | 2019-01-02 09：00 | 2019-01-02 18:15 |
| 3 | 王五 | 100003@163.com | 18603713232 | 666666 | 2019-01-02 09：00 | 2019-01-02 15:30 |
| 4 | 李四 | 100002@163.com | 18603712255 | 765422 | 2019-01-03 09：00 | 2019-01-03 17:45 |

上表中，Email作为非键字段，可以确定Name、Mobile和Password。但表中为李四存储了两个密码，无法确定那个是正确的，会产生更新问题。

同样，为了解决此问题，需进行分表设计，将上面的一个表分为两个表：

用户表（Xuser）：

| Id | Name | Email | Mobile | Password |
| --- | --- | --- | --- | --- |
| 1 | 张三 | 100001@163.com | 18603712222 | 123456 |
| 2 | 李四 | 100002@163.com | 18603712255 | 432122 |
| 3 | 王五 | 100003@163.com | 18603713232 | 666666 |

访问表（Visit）：

| Id | XuserId | DatetimeIn | DatetimeOut |
| --- | --- | --- | --- |
| 1 | 1 | 2019-01-02 09:00 | 2019-01-02 18:30 |
| 2 | 2 | 2019-01-02 09:00 | 2019-01-02 18:15 |
| 3 | 3 | 2019-01-02 09:00 | 2019-01-02 15:30 |
| 4 | 2 | 2019-01-03 09:00 | 2019-01-03 17:45 |

* 1. 实体关系

数据库被设计为一组相关的实体，实体关系可用ER图来表示，即实体关系图（entity relationship）。实体关系图使用符号表达实体、实体的属性和实体之间的关系。

1. 一对一关系

一对一是将数据表“垂直切分”，其实是不常见，或不常用的。为什么要这样的设计呢，不是增加了复杂性吗，然而并不一定，举例说明：

一个系统有 Employee（员工表）（包含字段：EmployeeId、姓名、性别、年龄、电话、地址等），每个员工都为一个用户，所以还有张User表（包含字段：UserId（关联 EmployeeId）、用户名、密码、角色等），这样你会发现，整合为一张表是否不太妥当？因为，User 的记录只会在登录时用到。

如果当一张表的字段过于太多，而很多字段可能只有在某些情况下，才会使用到，这时也考虑使用一对一设计。

1. 一对多关系

 一对多，是最常见的一种设计。就是 A 表的一条记录，对应 B 表的多条记录，且 A 的主键作为 B 表的外键。

最典型的一对多关系，比如订单表与订单明细表，订单表包含订单号、日期、金额、客户名称等信息。订单明细表则包含明细编号、订单号（外键）、商品名称、单价、数量等信息。其ER图如下：

Id

Date

Amount

CustomerId

Order

Id

OrderId

Name

Price

Quantity

OrderDetail

其他一对多关系例子，比如角色和用户，一个角色可能会有多个用户

1. 多对多关系

多对多关系，在数据库中也比较常见，可以理解为是一对多和多对一的组合。要实现多对多，一般都需要有一张中间表（也叫关联表），将两张表进行关联，形成多对多的形式。

学员和课程之间的关系就是典型的多对多关系。

Id

Name

Sex

Age

Mobile

Student

Id

StudentId

CourseId

StudentCoursesese

Id

Name

Course

* 1. 数据库设计说明书
     1. 版式要求

数据库设计说明书的版式要求同需求说明书。

* + 1. 结构要求

数据库设计说明书结构上一般包括封面、文档修订记录、目录、引言、数据库环境、数据库命名约定（规则）、数据结构设计、数据库安全设计等方面。

1. 封面

封面包括公司Logo、文档名称、编制信息表、公司名称及时间（年月）。

编制信息表为表格，一般包括编制人、审核人、项目经理、最终版本、终稿日期、总页数等信息。

1. 文档修订记录

表格形式，一般包括版本、修订日期、修订内容、参与人员、修订人等信息。

1. 目录

根据文档正文（引言及其以后内容）的标题生成的目录索引。

1. 引言

包括编写目的、文档范围、术语与缩写解释（或者叫定义）、参考文献等。

1. 数据库环境

说明选取的数据库类型（关系型/非关系型）、名称、版本号、部署环境、设计工具等。

1. 数据库命名约定

包括数据表、字段、存储过程、参数变量、关键字、语句、函数、注释等。

1. 数据结构设计

按照模块分类撰写。包括概述、静态数据、实体（表）关系图、存储过程设计等。

其中静态数据要写明表名及其中文名称、主键、字段名称、数据类型（精度范围）、是否为空、唯一性、默认值以及约束条件/说明等。属于数字字典的应列出其列表项。

比如：tblEquipment（设备表）

| 表名 | | tblEquipment | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 主键 | | Id | | | | |
| 序号 | 字段名称 | 数据类型  (精度范围) | 允许为  空Y/N | 唯一  Y/N | 默认值 | 约束条件/说明 |
|  | Id | int | N | Y | 0 | 主键 自增 |
|  | ReferenceId | uniqueidentifier | N | Y | 0 | 设备guid |
|  | MineId | int | N | N | 0 | 煤矿企业Id |
|  | ModelNumber | nvarchar(50) | N | N | “” | 设备型号 |
|  | EquipmentTypeId | int | N | N | 0 | 设备类别Id |
|  | UsingLocation | nvarchar(100) | N | N | “” | 使用地点 |
|  | Manufacturer | nvarchar(100) | N | N | “” | 厂商 |
|  | InstallDate | nvarchar(10) | N | N | “” | 安装日期 |
|  | CheckingDate | datetime | N | N | “” | 检查日期 |
|  | ValidBeginDate | datetime | N | N | “” | 检测有效期起始日期 |
|  | ValidEndDate | datetime | N | N | “” | 检测有效期截止日期 |
|  | TheYear | int | N | N | 0 | 检查年份 |
|  | EquipmentStatusId | int | N | N | 0 | 设备状态Id |
| 补充说明 | |  | | | | |

2. 数据库安全设计

包括数据库登录用户及密码设置、数据库定期自动备份等。