数据库常识

2019年8月1日

15:56

一、软件开发流程

1、需求分析

主要是了解清楚用户想要的是什么东西，做什么用，把软件的功能搞清楚

主要途径：与用户沟通，或者通过既有的系统来了解需求。

2、概要设计

根据用户的负载要求、软硬件要求、网络环境，来决定开发软件时所使用的语言、数据库等。

3、详细设计

把软件功能拆分开来进行详细设计，也包括数据库设计

4、编码实现

5、测试

测试会贯穿整个流程

6、安装维护

二、设计数据库的步骤

属于概要设计和详细设计阶段的作业

1、分析软件功能，与用户沟通他想要的内容

2、找出功能当中的实体类netity

比如：学生、课程、成绩、老师

3、针对每个实体，了解用户关心的内容，确定实体的属性

4、标识实体之间的关系，主要是主从关系。

5、绘制ER图entity-relationship，实体关系图

实体：矩形来标识，一般是名词

属性：椭圆来标识，一般是名词

关系：菱形来标识，一般是动词

6、将ER图转换为数据库中的表

实体对应的是表

属性对应的是表中的字段

关系一般由主外键来建立

三、主键

可以唯一标识一条数据的一个或多个字段

作用：唯一标识一条记录，不能重复，不能为空，用来保证行的完整性，作为一个可以被外键引用的对象和其他表建立联系。

四、外键

A表中的一个普通字段是B表中的主键，那么它可以作为A表的外键

作用：和另一张表建立联系，用来保证引用完整性

五、三大范式

1、第一范式

列不可再分，即保证每列的原子性

比如：学生信息表

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 联系方式 |
| 1001 | 王浩 | 山东济南438214 |

修改：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 联系电话 | 家庭住址 |

2、第二范式

在第一范式的基础上，确保表中的每列都和主键相关

比如：学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 课程号 | 课程名 | 成绩 |
| 1001 | 王浩 | 男 | 1 | java基础 | 90 |

修改：

学生表：

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 性别 |

成绩表：

|  |  |  |
| --- | --- | --- |
| 课程号 | 课程名 | 成绩 |

注：第二范式要求一张表只描述一种信息

3、第三范式

在第一范式和第二范式的基础上，要求任何字段不能由其他字段派生出来,要求字段没有冗余.

比如：学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 学院编号 | 学院住址 | 学院电话 |
| 1001 | 王浩 | 男 | 2 |  |  |
| 1002 | 王楠 | 女 | 2 |  |  |

修改：

学生信息表：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 学院编号 |

学院信息表：

|  |  |  |
| --- | --- | --- |
| 学院编号 | 学院地址 | 学院电话 |

六、数据库的备份与还原

导出：exp hvgu/hvgu@orcl file=d:\t1.dmp tables=(t1);

exp hvgu/hvgu@orcl file=d:\t1.dmp full=y;

注意：full-y:代表导入该用户下的所有表

导入：imp hvgu/hvgu@orcl file=d:\t1.dmp full=y;