**新建项目**

**计算机学院（软件学院）实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学生姓名** | 徐可可 | **学号** | 171530425 | **实验成绩** |  |
| **专业** | 软件工程 | **班级** | 软工四班 | **实验日期** | 19年12月17日 |
| **课程名称** | 数据库原理 | | | **任课教师** | 马丽 |
| **实验名称** | 数据库设计 | | | **实验序号** | 7 |
| **实验地点** | S411 | **实验台号** | 25 | **指导教师** | 马丽 |
| 1. **实验目的及要求**   1.熟悉数据库设计基本步骤；  2.练习数据库需求分析方法，并能给出数据字典；  3.练习ER图建模，掌握ER模型向关系模式的转化。 | | | | | |
| 1. **实验内容（或实验原理、实验拓扑）**   1.从用户需求出发，按照数据库设计步骤，分别完成如下内容：概念结构设计（ER模型），给出满足需求的最终全局ER图，要求模型尽量精简，消除不必要的冗余，并给出理由或说明；  2.逻辑结构设计（关系模式），把ER模型转换成适当的关系模式，并进行适当地规范化，设计相关完整性约束；  3.物理结构设计和实施，根据业务需求规划适当索引，并设计必要的用户视图或存储过程。 | | | | | |
| 1. **实验设备与环境**   1. 阿里云服务器：Intel(R) Xeon(R) Platinum 8269CY CPU @ 2. 50GHz、2G内存、40G高效云盘、Windows Server 2019 数据中心版、Microsoft SQL Server 2008 R2  2. 高配笔记本：Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz  、16G内存、48TB分布式硬盘、Windows 10 1909政府版、SQL Server Management Studio 18.4  3. 实验室电脑：英特尔 酷睿2 双核 T7700 @ 2.40GHz、3G内存、128 G虚拟硬盘、Windows 7 旗舰版 32位 SP1、Microsoft SQL Server 2008 R2 | | | | | |
| 1. **实验设计方案（包括实验步骤、设计思想、算法描述或开发流程等）**   第一步，数据抽象与局部E-R模型设计  A、数据抽象  在多层数据流中选择一个适当层次作为设计E-R图的出发点。  确定每个局部应用包含哪些实体，实体包含哪些属性，实体之间的联系  划分实体和属性的方法  分类：将一组具有某些共同特性和行为的对象抽象为一个实体。  聚合：将对象类型的组成成分抽象为属性。  B、局部E-R模型设计  局部E-R模型设计的原则是属性必须是不可分的数据项，不能再由放弃其他属性组成；属性不能与其他实体具有联系，联系只能发生在实体之间。  为简化E-R图，凡是能作为属性对待的，尽量作为属性。  第二步，全局E-R模型设计  集成各局部E-R模型，形成全局模型。视图集成的方法有两种：  A、多元集成法：一次性将多个局部E-R图合并为一个全局E-R图。  B、二元集成法：首先集成两个重要的局部E-R图，然后用累加的方法逐步将一个新的E-R图集成进来。  合并：  合并局部E-R图，消除冲突，初步生成E-R图。合并的关键是合理消除各局部E-R图的冲突。  优化：  消除初步E-R图中不必要的冗余，生成基本的E-R图。  冗余数据：可由基本的数据导出的数据。  冗余联系：可由基本的联系导出的联系。 | | | | | |

|  |
| --- |
| 1. **实验结果（包括设计效果、测试数据、运行结果等）** |
| 1. **实验小结（包括收获、心得体会、注意事项、存在问题及解决办法、建议等）**   应用数据库设计的范式理论对初始关系模型进行优化。数据库设计的三大范式如下：   * 第一范式 每一个分类必须是一个不可分的数据项。属性不可再分，确保每列的原子性。 * 第二范式 要求每个表只描述一件事情，每条记录有唯一标识列。 * 第三范式 数据库表中不包含已在其它表中已包含的非主关键字信息。   关系模式的规范化过程如下：   * 确定范式级别   考察关系模式的函数依赖关系，确定范式等级。   * 实施规范化处理   利用规范化方法和理论将关系模式规范化。   * 模式改进   合并：将用于关联查询的具有相同主键的各表合并可提高查询效率  分解：水平分解，将关系的元组分为若干子集，提高查询效率；垂直分解，把关系中经常一起使用的属性分解出来，形成一个子关系，提高执行效率。分解时要保持无损连接和函数依赖。 |

|  |
| --- |
| 1. **附录（包括作品、流程图、源程序及命令清单等）**  * **sc.sql** * **Newspaper DataBase Design**   **报纸数据库设计**  **Newspaper System**  **报纸系统**  **A newspaper is setting up a website where people can write, read and comment on news stories. Your job is to design a database that can record the information needed for the website to work.**  **一家报纸正在建立一个网站，人们可以在上面对新闻故事进行写作、阅读和评论。你的工作是设计一个数据库，可以记录网站工作所需的信息。**  **The website lists a number of stories, each classified within one of about 10 sections (‘Local news’, ‘World news’,‘Opinion’, ‘Sport’, ‘Technology’ etc).**  **该网站列出了许多故事，每一个故事都被归入10个栏目(“本地新闻”、“世界新闻”、“观点”、“体育”、“科技”等)中的一个栏目里。**  **Each content of each story consists of a piece of text (the manager would prefer there to be no limit on the size) as well as a headline and a short “lede” (a lede is a sentence of 10 to 20 words that summarizes the story).**  **每个故事的内容都有一段文字(经理希望没有大小限制)、一个标题和一个简短的“lede”(lede是一个10到20个单词的句子，总结了这个故事)。**  **Each story is written by one or more of our authors, who submit the story to our database on a particular date.**  **每个故事都是由我们的一个或多个作者撰写的，他们在特定日期将故事提交到我们的数据库。**  **If a story is considered worthy, it is edited by one of our editors, assigned to a section, and then published on a particular date.**  **如果一个故事被认为是有价值的，它将由我们的一位编辑编辑，分配到一个部分，然后在一个特定的日期出版。**  **Authors and editors are staff members.**  **作者和编辑都是工作人员。**  **It is not possible to be both an author and an editor.**  **既当作家又当编辑是不可能的。**  **About each staff member we store their name and when they join (and later leave) the newspaper.**  **关于每一位员工，我们会保存他们的名字，以及他们何时加入(后来离开)报纸。**  **We want to organize the website so that readers can click on a section, or an author’s name, and see a list of all the relevant stories.**  **我们希望组织网站，这样读者可以点击某个部分或作者的名字，并看到所有相关故事的列表。**  **On the main page we list each story’s title (shown as a headline) and lede.**  **在主页上，我们列出每个故事的标题(作为标题显示)和lede。**  **Then if the reader clicks on the headline, we display the entire story.**  **然后，如果读者点击标题，我们就会显示整个故事。**  **Our readers, if they wish, can choose to register themselves in our database, recording a username and password (they do not need to record their real name).**  **我们的读者，如果他们愿意，可以选择在我们的数据库中注册，记录用户名和密码(他们不需要记录他们的真实姓名)。**  **Readers who register can then comment on stories, and on other readers’ comments.**  **注册的读者可以对故事和其他读者的评论进行评论。**  **They can also click “Like” on a story – and if they change their mind later, they can “Unlike” it.**  **他们还可以点击某个故事的“喜欢”按钮——如果他们后来改变了主意，他们可以“不喜欢”这个故事。**  **When we display a story we show the number of Likes the story has received and list out the comments below it.**  **当我们显示一个故事时，我们会显示这个故事收到的赞数，并在下面列出评论。**  **With each comment we show the username of the person who commented, and the time they commented.**  **在每条评论中，我们都会显示评论人的用户名以及他们评论的时间。**  **We aim for brevity in comments, and restrict them to 1024 characters.**  **我们的目标是在评论中保持简洁，限制在1024个字符以内。** |