

湖南工业大学

课程设计（实训）

资 料

学 院 _____ 计算机学院 _____ 2018-2019 学年第 2 学期

课程名称 _____ 数据库系统概论 _____ 指导教师 _____ 肖满生 _____

学生姓名 _____ 周昭仰 _____ 专业班级 _____ 软件工程 1701 _____ 学号 _____ 17408001158 _____

题 目 _____ 企业项目管理系统 _____

起止日期 _____ 2019 年 6 月 17 日 ~ _____ 2019 年 6 月 21 日

目 录 清 单

序号	材 料 名 称	资料数量	备 注
1	课程设计任务书	1	
2	课程设计（实训）报告	1	
3	设计文档和源程序（电子文档）	1	
4	答辩 ppt（电子文档）	1	

课程设计（实训）任务书

学 院： 计算机学院 学生姓名： 周昭仰 专业班级： 软件工程 1701
 课程名称： 数据库系统概论 设计题目： 企业项目管理系统
 完成期限：自 2019 年 6 月 17 日至 2019 年 6 月 21 日共 1 周

内容及任务	<div>一、设计的任务及主要技术参数</div> <div>综合运用 MySQL 的开发技术，设计企业项目管理系统，实现通过组内分工共同完成此系统的设计和实现，并完成设计过程中答辩和实训报告文档。</div> <div>二、设计任务</div> <div>1) 系统分析(包括可行性分析、用户需求分析等)及功能分析；</div> <div>2) 系统总体数据库设计；</div> <div>3) 数据库调测；设计说明书的书写。</div> <div>三、设计工作量</div> <div>完成企业项目管理系统，用 1 周的时间来完成该系统的分析与设计，撰写出各个阶段需要的文档，由个人独立完成系统。</div>			
进度安排	起止日期		工作内容	
	6 月 17 日		选题、需求分析	
	6 月 18 日-6 月 19 日		完成具体内容里的设计	
	6 月 20 日		测试、完成课设报告	
	6 月 21 日		答辩、提交材料	
主要参考资料	<div>[1] 钱雪忠. 数据库原理及应用[M]. 北京：邮电大学出版社. 2007. 8</div> <div>[2] 陈晓勇. MySQL DBA 修炼之道[M]. 北京：机械工业出版社. 2017. 1</div> <div>[3] 戴小平. 数据库系统及应用[M]. 北京：中国科学技术出版社</div>			
同组信息	学号	姓名	分工负责的内容	备注
	17408001158	周昭仰	数据库系统设计实现	组长

指导教师（签字）：_____

年 月 日

系（教研室）主任（签字）：_____

年 月 日



课程设计（实训）报告

课程名称：_____数据库系统概论_____

题 目：_____企业项目管理系统_____

起止日期： 2019 年 6 月 17 日 至 2019 年 6 月 21 日

学 院 _____计算机学院_____

学 生 姓 名 _____周昭仰_____

班 级 _____1701_____

学 号 _____17408001158_____

成 绩 _____

指 导 教 师 （ 签 字 ） _____

2019 年 6 月

目 录

1	概述.....	1
1.1	背景和意义.....	1
1.2	任务概述.....	1
1.3	设计环境.....	1
2	具体内容.....	2
2.1	数据流程图.....	2
2.2	数据字典.....	4
2.3	概念结构设计.....	5
2.4	逻辑结构设计.....	6
2.5	物理结构设计.....	8
2.5.1	数据库命名规范.....	8
2.5.2	数据库存储引擎.....	8
2.5.3	数据类型.....	9
2.6	数据库实施阶段.....	9
2.6.1	表.....	9
2.6.2	查询.....	12
2.6.3	视图.....	15
2.7	数据库运行与维护.....	17
2.8	课程设计心得.....	18

1 概述

1.1 背景和意义

对于企业来说，一套项目管理系统主要包含项目的参与者和工作任务，对工作任务的监督是项目经理日常工作的一部分。对于项目参与者，则需要随时报告自己的任务进展情况，并及时告知项目经理。

在工作任务上，则还有不同的任务优先级控制，每个工作任务的轻重缓急都有所不同，对于特别紧迫的任务来说，则要求负责的员工必须尽快完成。

由于不同项目形成的项目小组，因此，项目任务管理过程中还要有员工任务分配的过程，这样才能随时指导每个项目中的任务到底是谁来完成的，以及完成的情况。

通常一个项目可以有多个员工完成，一个项目包括多个任务，一个员工可以同时完成多项任务，每个任务具有相应的任务状态（例如等待、执行、完成等）和优先级，每项任务由一名员工负责并且至少由一名员工承担。

1.2 任务概述

分析实体、建立实体表；分析关系，建立关系表；根据实际情况，设置主外键等约束条件；插入数据模拟；测试调整修改。

1.3 设计环境

使用 Mysql 作为数据库。





























使用 Navicat 工具管理数据库。

Windows 10 操作系统。

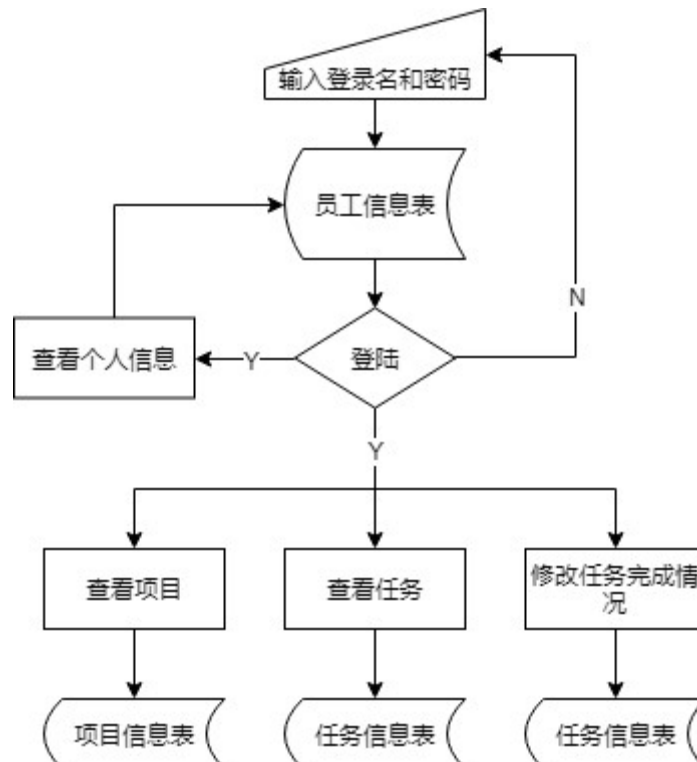
2 具体内容

2.1 数据流程图

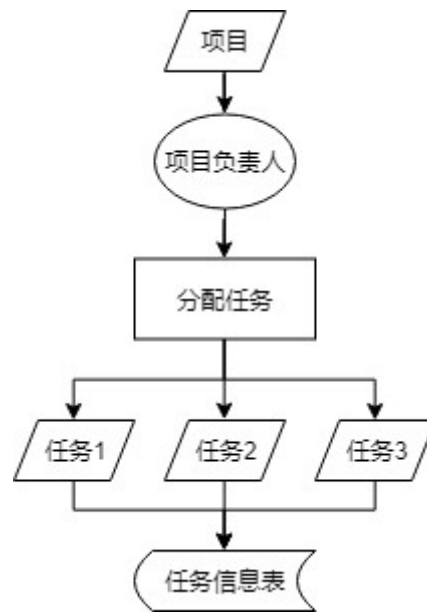
流程图符号定义

	过程		内部储存		手动输入		资料带
	可选过程		文档		手动操作		汇总连接
	决策		多文档		联系		或者
	数据		终止		离页连接符		对照
	预定义过程		准备		卡片		排序
	摘录		合并		库存数据		延期
	顺序访问存储器		磁盘		直接访问存储器		显示

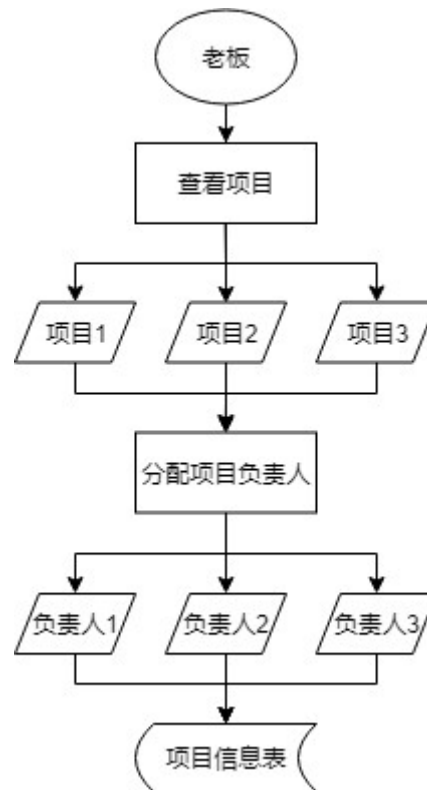
员工登录、查询个人信息、查看工作项目、查看工作任务、修改工作完成情况流程：



项目负责人分配任务流程：



老板分配项目负责人流程：



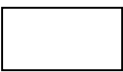
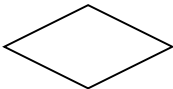
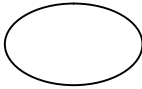
任务负责人修改任务优先级流程：



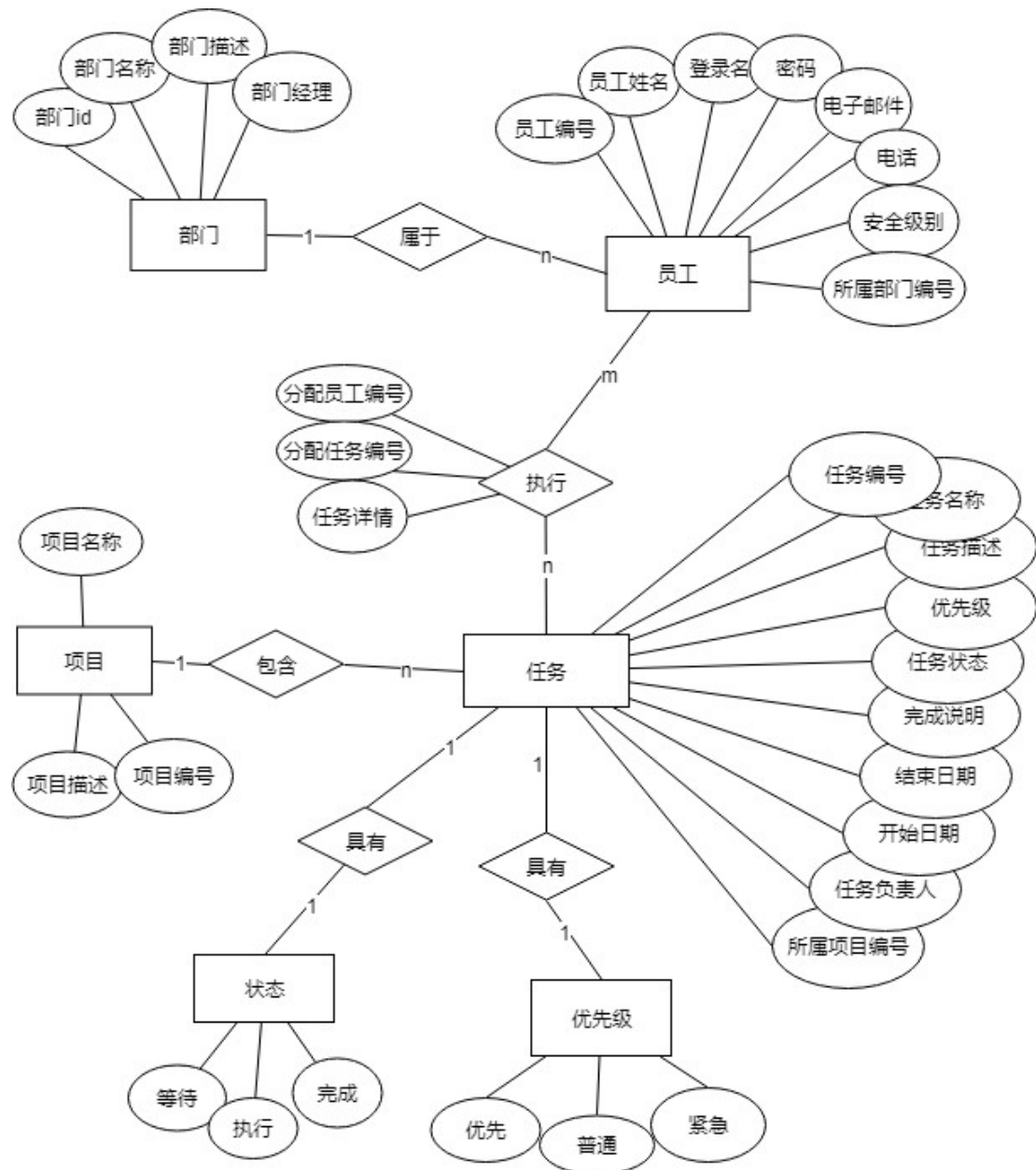
2.2 数据字典

数据库名	所属表	字段	类型长度	注释
epms	department_info	dpt_id	int(11)	部门编号
epms	department_info	dpt_name	varchar(255)	部门名称
epms	department_info	dpt_desc	varchar(255)	部门描述
epms	department_info	dpt_manager_id	int(11)	部门经理
epms	employee_info	ee_id	int(11)	员工编号
epms	employee_info	ee_name	varchar(255)	员工姓名
epms	employee_info	ee_loginname	varchar(255)	登录名
epms	employee_info	ee_password	varchar(255)	密码
epms	employee_info	ee_email	varchar(255)	电子邮件
epms	employee_info	ee_phone	varchar(11)	电话
epms	employee_info	ee_security_level	varchar(255)	安全级别
epms	employee_info	ee_dpt_id	int(11)	所属部门
epms	priority_info	priority_id	int(11)	优先级id
epms	priority_info	priority_name	varchar(255)	优先级名称
epms	priority_info	priority_desc	varchar(255)	优先级描述
epms	project_info	proj_id	int(11)	项目编号
epms	project_info	proj_name	varchar(255)	项目名称
epms	project_info	proj_desc	varchar(255)	描述
epms	rel_employee2task	rel_ee_id	int(11)	员工id
epms	rel_employee2task	rel_task_id	int(11)	任务id
epms	rel_employee2task	rel_details	varchar(255)	具体工作内容
epms	status_info	status_id	int(11)	状态id
epms	status_info	status_name	varchar(255)	状态名称
epms	status_info	status_desc	varchar(255)	状态描述
epms	task_info	task_id	int(11)	任务编号
epms	task_info	task_name	varchar(255)	任务名称
epms	task_info	task_desc	varchar(255)	任务描述
epms	task_info	task_priority	int(11)	优先级
epms	task_info	task_status	int(11)	任务状态
epms	task_info	task_finish_ins	varchar(255)	完成说明
epms	task_info	task_start_date	datetime	开始日期
epms	task_info	task_end_date	datetime	结束日期
epms	task_info	task_proj_id	int(11)	任务所属项目
epms	task_info	task_manager_id	int(11)	

2.3 概念结构设计

图例：  表示实体  表示联系  表示属性

具体 E-R 图：



2.4 逻辑结构设计

依据通常一个项目可以有多个员工完成，一个项目包括多个任务，一个员工可以同时完成多项任务，每个任务具有相应的任务状态（例如等待、执行、完成等）和优先级，每项任务由一名员工负责并且至少由一名员工承担等条件，得出员工对任务是执行关系、项目与任务是所属关系、状态对任务是所属关系、优先级对任务是所属关系、员工对项目是负责关系。

将状态信息、优先级信息、项目信息和员工信息的相应编号合并到任务信息表，在此表内可以根据每条任务查询到相应的编号，进而查询到编号对应的信息。

将每条任务信息内包含的状态信息和优先级信息分别分解到单独的表，可以节约存储资源消耗，避免重复数据造成冗余。

创建了总的任务视图，可以在任务视图内看到每一条任务对应的执行情况；创建了员工视图，可以看到所有员工的身份信息，方便企业管理员工；创建了项目视图，可以看到所有的项目信息，方便领导查看项目和管理项目经理。

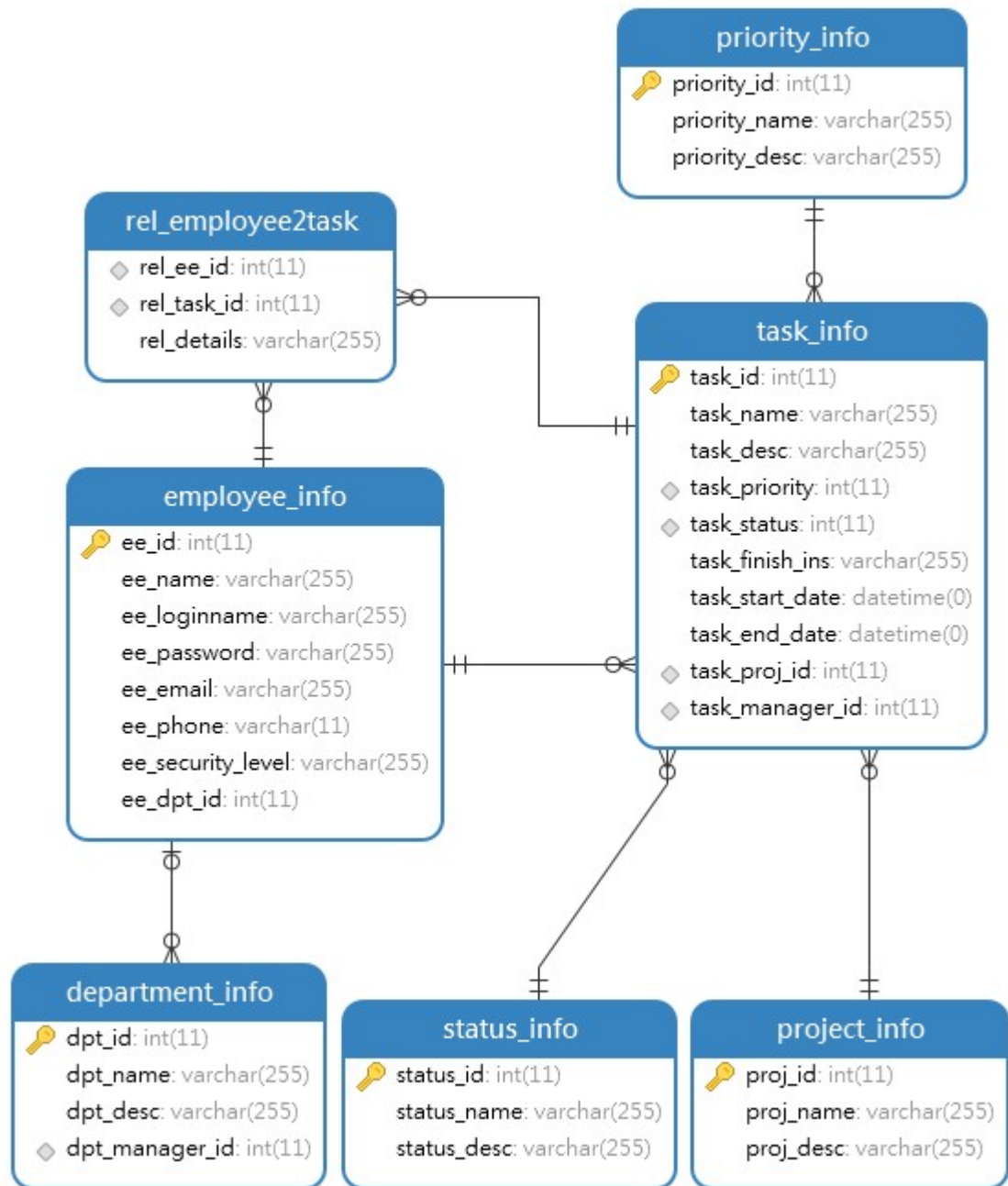
其中员工信息表中员工编号为主键，状态信息表中状态编号为主键，项目信息表中项目编号为主键，优先级信息表中优先级编号为主键，任务信息表中任务编号为主键，在任务信息表中员工编号、状态编号、项目编号、优先级编号联合作为此表的外键。

员工信息表、项目信息表、状态信息表、任务信息表、员工信息表、部门表中都包含了主键约束、非空约束、长度约束，在任务信息表中还包括了外键约束。每个表的主键 id 都设为主键自增。

系统中创建的各表所包含的数据项：

- （1）员工信息（员工编号、员工姓名、登录名、密码、电子邮件、电话、安全级别等）
- （2）项目信息（项目编号、项目名称、项目负责人、描述等）
- （3）任务信息（任务编号、任务名称、任务描述、优先级、任务状态、完成说明、开始日期、结束日期等）
- （4）状态信息（状态名称、状态描述等）
- （5）优先级信息（优先级名称、优先级描述等）
- （6）部门信息（部门编号、部门名称、部门描述、部门经理等）

项目的 UML 图：



数据库是关系型数据库，状态信息表、优先级信息表、员工信息表、部门表均符合第一范式，表的列的具有原子性,不可再分解。项目信息表符合第三范式，一张表最多存两层同类型信息，其中项目信息表可分为项目和负责员工两个表。

2.5 物理结构设计

2.5.1 数据库命名规范

- 数据库命名规范

采用 26 个英文字母(区分大小写)和 0-9 的自然数(经常不需要)加上下划线'_'组成;命名简洁明确(长度不能超过 30 个字符)。

- 数据库表名命名规范

采用 26 个英文字母(区分大小写)和 0-9 的自然数(经常不需要)加上下划线'_'组成;命名简洁明确,多个单词用下划线'_'分隔。

- 数据库表字段名命名规范

采用 26 个英文字母(区分大小写)和 0-9 的自然数(经常不需要)加上下划线'_'组成;命名简洁明确,多个单词用下划线'_'分隔。每个表中必须有自增主键。表与表之间的相关联字段名称要求尽可能的相同。

- 数据库表索引规范

命名简洁明确,为每个表创建一个主键索引;为每个表创建合理的索引;慎重建立复合索引。

2.5.2 数据库存储引擎

InnoDB 表类型,它提供了事务、行级锁机制和外键约束的功能。InnoDB 的表需要更多的内存和存储,它会在主内存中建立其专用的缓冲池用于高速缓冲数据和索引。

使用 InnoDB 具有如下的优点:

- 更新密集的表: InnoDB 存储引擎特别适合处理多重并发的更新请求。
- 事务: InnoDB 存储引擎是支持事务的标准 MySQL 存储引擎。
- 自动灾难恢复: 与其它存储引擎不同, InnoDB 表能够自动从灾难中恢复。
- 外键约束: MySQL 支持外键的存储引擎只有 InnoDB。
- 支持自动增加列 AUTO_INCREMENT 属性。

因此使用 InnoDB 是最理想的选择。

2.5.3 数据类型

MySQL 支持所有标准 SQL 数值数据类型。在该系统中，针对系统内的数据，制定了数据库表字段类型规范，针对不同的数据合理选择数据类型。

用尽量少的存储空间来存数一个字段的的数据；能使用 int 就不要使用 varchar、char；固定长度的类型最好使用 char；能使用 tinyint 就不要使用 smallint,int；最好给每个字段一个默认值,最好不能为 null。

2.6 数据库实施阶段

2.6.1 表

2.6.1.1 员工信息表

用于记录员工的基本信息，包括员工编号、员工姓名、登录名、密码、电子邮件、电话、安全级别等信息，具体如下：

```
CREATE TABLE `employee_info` (  
  `ee_id` int(11) NOT NULL COMMENT '员工编号',  
  `ee_name` varchar(255) DEFAULT NULL COMMENT '员工姓名',  
  `ee_loginname` varchar(255) DEFAULT NULL COMMENT '登录名',  
  `ee_password` varchar(255) DEFAULT NULL COMMENT '密码',  
  `ee_email` varchar(255) DEFAULT NULL COMMENT '电子邮件',  
  `ee_phone` varchar(11) DEFAULT NULL COMMENT '电话',  
  `ee_security_level` varchar(255) DEFAULT NULL COMMENT '安全级别',  
  PRIMARY KEY (`ee_id`) USING BTREE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ ee_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	员工编号
ee_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		员工姓名
ee_loginname	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		登录名
ee_password	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		密码
ee_email	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		电子邮件
ee_phone	varchar	11	0	<input type="checkbox"/>	<input type="checkbox"/>		电话
ee_security_level	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		安全级别

2.6.1.2 任务优先级表

用于记录任务优先级的信息，包括优先级编号、优先级名称和优先级描述信息，其具体显现如下：

```
CREATE TABLE `priority_info` (  
  `priority_id` int(11) NOT NULL COMMENT '优先级 id',  
  `priority_name` varchar(255) DEFAULT NULL COMMENT '优先级名称',  
  `priority_desc` varchar(255) DEFAULT NULL COMMENT '优先级描述',  
  PRIMARY KEY (`priority_id`),  
  KEY `priority_id` (`priority_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
priority_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	优先级id
priority_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		优先级名称
priority_desc	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		优先级描述

2.6.1.3 项目信息表

该表用于记录项目的信息，包括项目编号、项目名称、项目负责人和项目描述信息，其具体实现如下：

```
CREATE TABLE `project_info` (  
  `proj_id` int(11) NOT NULL COMMENT '项目编号',  
  `proj_name` varchar(255) DEFAULT NULL COMMENT '项目名称',  
  `proj_manager` varchar(255) DEFAULT NULL COMMENT '项目负责人',  
  `proj_desc` varchar(255) DEFAULT NULL COMMENT '描述',  
  PRIMARY KEY (`proj_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
proj_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	项目编号
proj_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		项目名称
proj_manager	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		项目负责人
proj_desc	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		描述

2.6.1.4 任务状态表

该表用来记录任务完成的状态的信息，包括状态编号、状态描述和状态名称信息，表设计如下：

```
CREATE TABLE `status_info` (
  `status_id` int(11) NOT NULL COMMENT '状态 id',
  `status_desc` varchar(255) DEFAULT NULL COMMENT '状态描述',
  `status_name` varchar(255) DEFAULT NULL COMMENT '状态名称',
  PRIMARY KEY (`status_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
status_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	状态id
status_desc	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		状态描述
status_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		状态名称

2.6.1.5 任务信息表

该表用来记录任务的各种信息，包括任务名称、优先级、任务状态、开始日期等信息，表设计如下：

```
CREATE TABLE `task_info` (
  `task_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '任务编号',
  `task_name` varchar(255) NOT NULL COMMENT '任务名称',
  `task_desc` varchar(255) DEFAULT NULL COMMENT '任务描述',
  `task_priority` int(11) DEFAULT NULL COMMENT '优先级',
  `task_status` int(11) DEFAULT NULL COMMENT '任务状态',
  `task_finish_ins` varchar(255) DEFAULT NULL COMMENT '完成说明',
  `task_start_date` datetime DEFAULT NULL COMMENT '开始日期',
  `task_end_date` datetime DEFAULT NULL COMMENT '结束日期',
  `ee_id` int(11) DEFAULT NULL COMMENT '任务分配的员工',
  `proj_id` int(11) DEFAULT NULL COMMENT '所属项目编号',
  `task_manager` int(11) DEFAULT NULL COMMENT '任务负责人',
  PRIMARY KEY (`task_id`),
  KEY `ee_id` (`ee_id`),
  KEY `proj_id` (`proj_id`),
  KEY `priority_id` (`task_priority`),
  KEY `status_id` (`task_status`),
  CONSTRAINT `ee_id` FOREIGN KEY (`ee_id`) REFERENCES
`employee_info` (`ee_id`),
  CONSTRAINT `priority_id` FOREIGN KEY (`task_priority`)
REFERENCES `priority_info` (`priority_id`),
  CONSTRAINT `proj_id` FOREIGN KEY (`proj_id`) REFERENCES
```

```
`project_info` (`proj_id`),
  CONSTRAINT `status_id` FOREIGN KEY (`task_status`) REFERENCES
`status_info` (`status_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
task_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	任务编号
task_name	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		任务名称
task_desc	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		任务描述
task_priority	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		优先级
task_status	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		任务状态
task_finish_ins	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		完成说明
task_start_date	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>		开始日期
task_end_date	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>		结束日期
ee_id	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		任务分配的员工
proj_id	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		所属项目编号
task_manager	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		任务负责人

2.6.1.6 部门信息表

该表用来记录部门的各种信息，包括部门编号、部门名称、部门描述、部门经理，表设计如下：

```
CREATE TABLE `department_info` (
  `dpt_id` int(11) NOT NULL COMMENT '部门编号',
  `dpt_name` varchar(255) NOT NULL COMMENT '部门名称',
  `dpt_desc` varchar(255) DEFAULT NULL COMMENT '部门描述',
  `dpt_manager_id` int(11) NOT NULL COMMENT '部门经理',
  PRIMARY KEY (`dpt_id`) USING BTREE,
  KEY `fk_dpt_manager_id` (`dpt_manager_id`) USING BTREE,
  CONSTRAINT `fk_dpt_manager_id` FOREIGN KEY (`dpt_manager_id`)
REFERENCES `employee_info` (`ee_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
ROW_FORMAT=DYNAMIC;
```

名	类型	长度	小数点	不是 null	虚拟	键	注释
dpt_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	部门编号
dpt_name	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		部门名称
dpt_desc	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		部门描述
dpt_manager_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		部门经理

2.6.2 查询

数据查询对几个表进行了全表查询，各表的结果如下：

2.6.2.1 员工信息查询

```
SELECT
employee_info. ee_id,
employee_info. ee_name,
employee_info. ee_loginname,
employee_info. ee_password,
employee_info. ee_email,
employee_info. ee_phone,
employee_info. ee_security_level,
employee_info. ee_dpt_id
FROM
employee_info
```

ee_id	ee_name	ee_loginname	ee_password	ee_email	ee_phone	ee_security_level	ee_dpt_id
1	周昭仰	admin	123123	q@qq.com	12345678900	1	1
2	小周	aaa	aaa	aaa@qq.com	12345678900	1	2
3	小昭	bbb	bbb	bbb@qq.com	12345678900	1	3

2.6.2.2 优先级信息查询

```
SELECT
priority_info. priority_id,
priority_info. priority_name,
priority_info. priority_desc
FROM
priority_info
```

信息	Result 1	概况	状态
priority_id	priority_name	priority_desc	
1	紧急	最先做	
2	重要	次于紧急任务	
3	普通	次于重要任务	

2.6.2.3 项目信息查询

```
SELECT
project_info. proj_id,
project_info. proj_name,
project_info. proj_desc
```

```
FROM
project_info
```

proj_id	proj_name	proj_desc
1	游戏	game

2.6.2.4 状态信息查询

```
SELECT
status_info.status_desc,
status_info.status_name
FROM
status_info
```

信息	Result 1	概况	状态
status_desc	status_name		
等待	To Do		
执行	Doing		
完成	Finished		

2.6.2.5 任务信息查询

```
SELECT
task_info.task_id,
task_info.task_name,
task_info.task_desc,
task_info.task_priority,
task_info.task_status,
task_info.task_finish_ins,
task_info.task_start_date,
task_info.task_end_date,
task_info.task_proj_id,
task_info.task_manager_id
FROM
task_info
```

task_id	task_name	task_desc	task_priority	task_status	task_finish_ins	task_start_date	task_end_date	task_proj_id	task_manager_id
1	登录注册	登录注册模块	1	1	实现登陆注册功能	2019-06-01 21:38:48	2019-06-14 21:38:56	1	1
2	UI设计	UI/UX设计	1	1	上交设计稿	2019-06-19 19:33:09	2019-06-21 19:33:12	1	1
3	剧情策划	剧本	2	2	上交剧本	2019-06-19 20:00:49	2019-06-22 20:00:51	1	2

2.6.2.6 工作任务分配查询

```
SELECT
rel_employee2task.rel_ee_id,
rel_employee2task.rel_task_id,
rel_employee2task.rel_details
FROM
rel_employee2task
```

rel_ee_id	rel_task_id	rel_details
1	1	
1	2	
2	1	
2	2	
3	2	
3	3	

2.6.3 视图

2.6.3.1 所有任务详细信息视图

```
SELECT
`task_info`.`task_id` AS `task_id`,
`task_info`.`task_name` AS `task_name`,
`task_info`.`task_desc` AS `task_desc`,
`task_info`.`task_finish_ins` AS `task_finish_ins`,
`task_info`.`task_start_date` AS `task_start_date`,
`task_info`.`task_end_date` AS `task_end_date`,
`employee_info`.`ee_name` AS `ee_name`,
`status_info`.`status_name` AS `status_name`,
`priority_info`.`priority_name` AS `priority_name`,
`project_info`.`proj_name` AS `proj_name`
FROM
(
(
(
```

```

        ( `task_info` JOIN `employee_info` ON
        ( ( `task_info`.`task_manager_id` = `employee_info`.`ee_id` ) ) )
        JOIN `status_info` ON ( ( `task_info`.`task_status` =
        `status_info`.`status_id` ) )
    )
    JOIN `priority_info` ON ( ( `task_info`.`task_priority` =
    `priority_info`.`priority_id` ) )
    )
    JOIN `project_info` ON ( ( `task_info`.`task_proj_id` =
    `project_info`.`proj_id` ) )
    )

```

task_id	task_name	task_desc	task_finish_ins	task_start_date	task_end_date	ee_name	status_name	priority_name	proj_name
1	登录注册	登录注册模块	实现登陆注册功能	2019-06-01 21:38:48	2019-06-14 21:38:56	周昭仰	To Do	紧急	游戏
2	UI设计	UI/UX设计	上交设计稿	2019-06-19 19:33:09	2019-06-21 19:33:12	周昭仰	To Do	紧急	游戏
3	剧情策划	剧本	上交剧本	2019-06-19 20:00:49	2019-06-22 20:00:51	小周	Doing	重要	游戏

2.6.3.2 员工所属部门视图

SELECT

```

`employee_info`.`ee_name` AS `ee_name`,
`employee_info`.`ee_loginname` AS `ee_loginname`,
`employee_info`.`ee_password` AS `ee_password`,
`employee_info`.`ee_email` AS `ee_email`,
`employee_info`.`ee_phone` AS `ee_phone`,
`employee_info`.`ee_security_level` AS `ee_security_level`,
`employee_info`.`ee_dpt_id` AS `ee_dpt_id`,
`employee_info`.`ee_id` AS `ee_id`,
`department_info`.`dpt_name` AS `dpt_name`

```

FROM

```

( `employee_info` JOIN `department_info` ON
( ( `department_info`.`dpt_manager_id` =
`employee_info`.`ee_id` ) ) )

```

ee_name	ee_loginname	ee_password	ee_email	ee_phone	ee_security_level	ee_dpt_id	ee_id	dpt_name
周昭仰	admin	123123	q@qq.com	12345678900	1	1	1	技术部
小周	aaa	aaa	aaa@qq.com	12345678900	1	2	2	财务部
小昭	bbb	bbb	bbb@qq.com	12345678900	1	3	3	业务部

2.6.3.3 所有项目包含的任务信息：

SELECT

```

project_info.proj_name,

```

```

project_info.proj_desc,
task_info.task_name,
task_info.task_desc
FROM
project_info
INNER JOIN task_info ON task_info.task_proj_id =
project_info.proj_id

```

proj_name	proj_desc	task_name	task_desc
▶ 游戏	game	登录注册	登录注册模块
游戏	game	UI设计	UI/UX设计
游戏	game	剧情策划	剧本

2.6.3.4 某员工所属部门:

```

SELECT
employee_info.ee_name,
department_info.dpt_name,
department_info.dpt_desc
FROM
employee_info
INNER JOIN department_info ON department_info.dpt_manager_id =
employee_info.ee_id

```

ee_name	dpt_name	dpt_desc
▶ 周昭仰	技术部	技术人员
小周	财务部	管理帐务
小昭	业务部	业务人员

2.7 数据库运行与维护

软件维护是软件生命周期的最后一个阶段，它处于系统投入生产性运行以后的日期中，为了纠正错误或满足新的需要而进行修改软件的过程，即是软件维护也是为软件的升级提供详细的资料。系统维护的目的是保证管理信息系统正常而可靠的运行，并能使系统不断得到改善和提高，以充分发挥作用。

系统维护的具体实践有以下

- 表管理

包括数据库内所有的表，对其信息进行修改、查询，删除。

- 数据备份

包括数据库的备份和日志文件的备份，可以随时将数据备份到硬盘或优盘保存，以免以后系统出现故障，可以借助这些备份文件进行恢复。

- 数据恢复

当数据丢失或出现其他故障后，可以从备份文件恢复数据。

- 稳定性、兼容性测试

反复更改代码，不断调试，测试种种可能性。

2.8 课程设计心得

经过数据库课程的学习，我们掌握了创建数据库以及对数据库的操作的基础知识。在这此次课程设计实践中，将所学运用到实际，我收获了很多，比如说我对做一个完整的系统的流程有了更新的掌握，也让自己学到了更多简单又基础的东西。在这个过程中，我熟悉了 SQL Server2008 的操作环境，熟练了 Word 里面的操作。

整个开发过程中我深深体会到，要严格按照软件工程的开发思路，不能急于求成，一来就马上编程，应该多做分析和研究，理清思路。前一阶段的工作的好坏将直接影响后一阶段工作的开展。多学习多请教，更多新的技术不断的涌现，可以通过请教别人或者多上网学习别人的编程方法、参考编程案例，可以转变成你的知识，才能做的更好。这些经验是相当宝贵的，为我们以后能够编出更好的程序是一个良好的启发。

限于我的技术水平有限这次的课程设计论文和编程软件的错误和不当之处在所难免，还得请老师多多指教！在这次的课程设计中虽然时间紧迫但我学会了很多，也感到自身知识的贫乏，希望在日后的努力学习中把它做成更完善的系统，并能做其他完善的系统。通过这个课程设计，受益匪浅，我发现只有自己真正动手去做，才能体会其中的乐趣，只要自己愿意去做，什么事都有可能做好。