|  |  |  |  |
| --- | --- | --- | --- |
| **学生学号** | 0121503490220 | **实验课成绩** |  |

**学 生 实 验 报 告 书**

|  |  |
| --- | --- |
| **实验课程名称** | 企业数据库应用 |
| **开 课 学 院** | 管理学院 |
| **指导教师姓名** | 肖红彦 |
| **学 生 姓 名** | 何睿 |
| **学生专业班级** | 信管1502 |

**2017 — 2018 学年 第 一 学期**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

**实验报告填写说明**

1. 综合性、设计性实验必须填写实验报告，验证、演示性实验可不写实验报告。
2. 实验报告书**必须**按统一格式制作（实验中心网站有下载）。
3. 老师在指导学生实验时，必须按实验大纲的要求，逐项完成各项实验；实验报告书中的实验课程名称和实验项目**必须**与实验指导书一致。
4. 每项实验依据其实验内容的多少，可安排在一个或多个时间段内完成，但每项实验只须填写一份实验报告。
5. 每份实验报告教师都应该有签名、评分表及实验报告成绩。
6. 教师应及时评阅学生的实验报告并给出各实验项目成绩，完整保存实验报告。在完成所有实验项目后，教师应按学生姓名将批改好的各实验项目实验报告装订成册，构成该实验课程总报告，按班级交到实验中心，每个班级实验报告袋中附带一份实验指导书及班级实验课程成绩表。
7. 实验报告封面信息需填写完整，并给出实验环节的成绩，实验环节成绩按其类型采取百分制或优、良、中、及格和不及格五级评定（与课程总成绩一致），并记入课程总成绩中。

实验课程名称：\_企业数据库应用\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | **SQL Server的安全性管理** | | | **实验成绩** |  |
| **实 验 者** | **何睿** | **专业班级** | **信管1502** | **组 别** |  |
| **同 组 者** |  | | | **实验日期** | **2017.11.8—12.6** |
| 一部分：实验预习报告（包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等）   1. 实验目的、意义   可以根据系统对安全性的不同需求，采用合适的方式来完成数据库系统安全体系的设计，独立制定具体应用环境的安全策略，要求写出实验报告，详细描述安全策略的部署与实现方法   1. 实验基本原理与方法   遵循计算机操作的基本原理和方法   1. 实验内容及要求   1）熟练掌握SQL SERVER的完整性控制内容与方法  2）可以根据系统对安全性的不同需求，采用合适的方式来完成数据库系统安全体系的设计，独立制定具体应用环境的安全策略。   1. 实验方案或技术路线（只针对综合型和设计型实验）   本实验为综合型实验。  从最基本的安全性管理方法入手，了解完整性控制内容与方法，包括数据完整性约束、存储过程、触发器、事务、封锁协议（乐观锁）、账户管理、角色管理、用户身份认证、权限控制等诸方面的安全性管理方法，独立制定具体应用环境的安全策略。  制定数据库安全策略的内容：   1. 数据完整性约束策略    1. 主键约束    2. 外键约束    3. 非空约束    4. 唯一约束    5. CHECK约束 2. 制定索引策略，保障数据完整性，提高数据查询检索效率    1. 唯一聚集索引    2. 非聚集索引    3. 包含性列索引 3. 业务级安全管理策略，运用SQL编程实现领域问题应用级安全性    1. 根据不同查询与数据更新需求，定义视图约束用户对数据库的访问模式，屏蔽数据库模式与内模式    2. 运用游标实现行级数据访问控制    3. 定义函数描述业务规则与约束    4. 定义存储过程描述业务规则与约束    5. 定义触发器描述业务规则与约束    6. 定义事务保障业务处理的一致性与完整性    7. 运用乐观锁和并发控制保障多任务环境的数据安全性 4. 身份与权限管理策略    1. 账户管理    2. 角色管理    3. 访问许可权限管理    4. 身份认证 5. 实验结论与总结 | | | | | |
| 第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）  此次实验基于在ERP课程中所创立的公司，公司的主要业务是生产服装，通过客户在线选购公司已经提供的服装模型，或者客户提供服装设计意见由公司进行辅助设计，中长期规划管理所用到的数据库表，关系式规范后如下。  **1.数据库表格**  **客户**（客户ID号、姓名、性别、身份证件号、联系方式、订制情况、设计师评价情况、资格认证、产品编号，客户喜好类型）  **产品**（产品ID号、产品名称、产品设计模型、产品设计师ID号、产品特征说明、销售价格、所含布料分类、产品模型等级、产品长度、产品使用注意事项）  **需求预测单**（预测单号，物料编码，物料名称，物料规格，计量单位，起始日期，结束日期，预测数量，状态）  **销售单（**销售单号，物料编码，物料名称，物料规格，计量单位，起始日期，结束日期，销售数量，均化类型，均化取整,状态，备注）  **生产能力计划单**（MPS报表单号，订单号码，物料编码，物料名称，物料规格，计量单位，来源，锁定人，原始数量 建议计划量 未计划量 int NOT NULL,锁定日期，开工日期，完工日期）。  **基础物料**（物料编码，物料名称，物料规格，物料类型，物料价格，物料单位，单位标准工时，累计提前期，使用状态，所属部门，所属库房，供应商编号，供应商名称）  **供应商**（供应商ID号，供应商名称，所属地，地址，负责职工，联系方式）  **1.1数据库完整性约束测策略**  首先根据上面的逻辑分析得到的表的关系，使用T—SQL语言得到数据库和数据表。   1. **首先创建数据库**   create database DB  on  (name="DB01",  filename="C:\HeRui\Sql\DB01.mdf",  size=5MB,  maxsize=20MB,  filegrowth=10%  )  log on  (name="DB02",  filename=" C:\HeRui\Sql \DB02.mdf",  size=5MB,  maxsize=20MB,  filegrowth=10%  )  **1.2 在创建数据库表（同时建立主健约束，在实验的过程将char类型更换为nchar类型，因为char类型出现了字符显示错误）**  （1）**主键约束**  创建需求预测表，以预测单号为主健约束，物料名称为非空约束  USE DB  GO  create table 预测单  (预测单号 int PRIMARY KEY, --主健约束  物料编码 int FOREIGN KEY REFERENCES 基础物料,  物料名称 varchar(18),  物料规格 varchar(18),  计量单位 varchar(18),  起始日期 datetime DEFAULT getdate(),  结束日期 datetime DEFAULT getdate(),  预测数量 int,  状态 varchar(18),  go  **（2）外键约束**  创建销售单表，对销售单号做主健约束，物料规格和计量单位，均化类型，均化取整做Check约束，状态做数字1-4约束，其余属性做非空约束。  create table 销售单  (销售单号 int PRIMARY KEY,  物料编码 int FOREIGN KEY REFERENCES NOT NULL 基础物料,  物料名称 varchar(18) NOT NULL,  物料规格 varchar(18) NOT NULL CHECK（“物料规格”）,  计量单位 varchar(18) NOT NULL CHECK（“计量单位”）,  起始日期 datetime DEFAULT getdate() NOT NULL,  结束日期 datetime DEFAULT getdate()NOT NULL,  销售数量 int NOT NULL,  均化类型 char(6) check (均化类型='不均化' or 均化类型='周均化' or 均化类型='日均化'),  均化取整 char(6) check (均化取整='不取整' or 均化取整='取上整' or 均化取整='取下整'),  时格代号 int NOT NULL,  时格说明 varchar(20) NOT NULL,  状态 varchar(18) NOT NULL Check（“状态”‘[1-4]’）NOT NULL,  备注 varchar(18) NOT NULL，  )  **（3）非空约束**  创建MPS报表，以报表单号作为主健约束，其余的属性做非空约束。  go  create table MPS报表  (MPS报表单号 int PRIMARY KEY,  销售单号 int FOREIGN KEY REFERENCES 销售单,  物料编码 int NOT NULL,  物料规格 varchar(18) NOT NULL,  计量单位 varchar(18) NOT NULL，  产品模型 varchar(18) NOT NULL，  来源 varchar(18) NOT NULL，  锁定人 varchar(18) NOT NULL，  原始数量 int NOT NULL,  建议计划量 int NOT NULL,  未计划量 int NOT NULL,  开工日期 datetime DEFAULT getdate(),  完工日期 datetime DEFAULT getdate(),  锁定日期 datetime DEFAULT getdate(),  计划员名称 varchar(18))  （4）**唯一约束**  创建基础物料文档表，对物料编码做主健约束，物料类型做外键约束，物料规格做唯一约束  go  create table 基础物料  (物料编码 int PRIMARY KEY,  物料名称 varchar(18),  物料规格 varchar(18) NOT NULL UNIQUE,  物料类型 varchar(18) FOREIGN KEY REFERENCES MPS报表(“物料属性”)  物料价格 money NOT NULL,  物料单位 varchar(18),  单位标准工时 int NOT NULL,  累计提前期 int NOT NULL,  使用状态 varchar(18),  所属部门 varchar(18),  所属库房 varchar(18),  供应商编号 int FOREIGN KEY REFERENCES 供应商,  供应商名称 varchar(18))  （5）**Check 约束**  创建职工表，职工号做主健约束，职工名称和职工类型做非空约束，职工性别做Check约束，其余属性做非空约束。  go  USE DB  GO  create table 职工  (职工号 int PRIMARY KEY,  职工名称 nvarchar(20) NOT NULL,  职工类型 nvarchar(20) NOT NULL,  职工性别 char(2) check (职工性别 = '男' OR 职工性别 = '女'),  联系方式 nvarchar(24)  )  go  create table 供应商  (供应商号 int PRIMARY KEY,  供应商名称 nvarchar(20)NOT NULL,  所属地 nvarchar(20)NOT NULL,  地址 nvarchar(20)NOT NULL,  负责职工 nvarchar(20)NOT NULL,  联系方式 nvarchar(24)  )  Go  **2．1 索引策略，制定索引策略，保障数据完整性，提高数据的检索效率**  索引是某个表中一列或多个列值得组合和相应的指向表中物料标识这些值得数据页得逻辑指针得清单。他就像书中的目录，使得在数据库中，程序能够快速得通过所以对数据进行扫描。Sql中分为以下索引。   1. 唯一索引 2. 聚集索引 3. 非聚集索引   为公司的需求预测单创建唯一索引  CREATE UNIQUE INDEX custmor\_demand ON预测单(预测单号)  为公司的产品模型创造唯一索引  CREATE UNIQUE INDEX procduct\_model 产品模型(MPS报表)  为预测单创造索引  USE DB  GO  create unique index ind\_001  on 预测单(预测单号)  go  create clustered index ind\_002  on 预测单(起始日期,结束日期,预测数量,时格代号)  go  create nonclustered index ind\_003  on 预测单(物料编码,物料名称,预测展开)  Go  为销售单创造索引  USE DB  GO  create unique index ind\_004  on 销售单(销售单号)  为MPS报表创造索引  USE DB  GO  create unique index ind\_007  on MPS报表(MPS报表单号)  go  create clustered index ind\_008  on MPS报表(来源,锁定人,开工日期,完工日期,锁定日期)  go  create nonclustered index ind\_009  on MPS报表(销售单号,物料编码,需求跟踪号,建议计划量,)  Go  为基础物料创造索引  USE DB  GO  create unique index ind\_010  on 基础物料(物料编码)  go  create clustered index ind\_011  on 基础物料(累计提前期,所属部门,所属库房,供应商编号)  go  create nonclustered index ind\_012  on 基础物料(物料名称,物料价格)  Go  为不同的供应商创建索引  USE DB  GO  create unique index ind\_013  on 供应商(供应商号)  go  create nonclustered index ind\_015  on 供应商(供应商名称,所属地,联系方式)  Go  **2.2 包含性索引列**    **3业务安全管理策略，运用SQL编程实现中长期规划管理领域问题应用级安全性**  3.1**用户视图的设定**  USE [DB]  GO  CREATE VIEW user\_view  (计划日期,  物料编码,  起始日期,  结束日期,  销售数量,  预测数量,  原始数量,  建议计划量,  未计划量,  计划完工日)  AS  SELECT  MPS报表.开工日期,  基础物料.物料编码,  预测单.起始日期,  预测单.结束日期,  销售单.销售数量,  预测单.预测数量,  MPS报表.原始数量,  MPS报表.建议计划量,  MPS报表.未计划量,  MPS报表.完工日期  FROM MPS报表,销售单,预测单,基础物料  WHERE MPS报表. 销售单号=销售单. 销售单号  AND 预测单.物料编码=基础物料.物料编码  AND 销售单.物料编码=基础物料.物料编码    **3.2运用游标实现数据访问控制**  （1）利用游标对查询的MPS报表数据集进行遍历  DECLARE mycursor CURSOR  FOR SELECT MPS报表单号, 销售单号  FROM MPS报表  OPEN mycursor  DECLARE @MPS报表单号 INT,@销售单号 INT  FETCH NEXT FROM mycursor INTO @MPS报表单号 INT,@销售单号  WHILE(@@FETCH\_STATUS=0)  BEGIN  PRINT N‘MPS报表单号’+CAST(@MPS报表单号 AS CHAR(10))+ N’ 销售单号’ +CAST(@销售单号 AS CHAR(10))  FETCH NEXT FROM mycursor INTO @MPS报表单号 INT,@销售单号  END  CLOSE mycursor  DELLOCATE mysursor   1. 利用游标实现行级数据访问控制     利用游标实现行级别控制  (3) 使用游标读取当前元素    (4) 定义scroll游标  DECLARE scroll\_custmor SCROLL CURSOR  FOR  SELECT c\_no,c\_name  FROM custmor  (5)打开滚动游标  OPEN scroll\_custmor  （6）分别使用游标读取第一行、下一行和上一行        （7）定义函数与业务规则     1. 定义业务存储过程描述业务规则于约束     （9）调用存储过程。       1. 关闭游标     （12）删除刚才定义错误的游标scroll\_product    **3.3 定义函数描述业务规则与约束**  （1） 求通过资格认证的所有客户的数量函数  CREATE FUNCTION all\_t\_custmor(@c\_authority AS INT)  RETURNS INT  BEGIN  DECLARE @all int  SET @all=''  SELECT @all='The Quantity of T Custmors are'+SUM(@c\_authority)  FROM custmor  WHERE c\_authority='T'  RETURN @all  END  此函数用于查看有效客户数量  （2） 提取客户订制情况中的数字的函数  IF OBJECT\_ID('c\_last') IS NOT NULL  DROP FUNCTION c\_last\_num  GO  CREATE FUNCTION c\_last\_num(@c\_last CHAR(10))  RETURNS char(10)  BEGIN  WHILE PATINDEX('%[^0-9]%',@c\_last) > 0  BEGIN  set @c\_last=stuff(@c\_last,patindex('%[^0-9]%',@c\_last),1,'')  END  RETURN @c\_last  END  --测试  GO  PRINT c\_last\_num ('26次')  GO  --26  此函数用于提取客户表订制情况中的数字，来统计客户的历史交易次数。  （5） 求所有产品设计师的平均设计权限等级的函数  CREATE FUNCTION avg\_d\_level(@d\_level AS INT)  RETURNS INT  BEGIN  DECLARE @level int  SET @level=''  SELECT @level='The average level of all designers is'+AVG(@d\_level)  FROM designer  RETURN @level  END  此函数用于帮助产品经理管理公司不同等级设计师数量，对新设计师的招聘决策提供帮助。  **3.4 定义存储过程业务描述规则与约束**  （1）定义和客户相关的存储过程  CREATE PROCEDURE custmorTF  @c\_no CHAR(6),  @c\_name VARCHAR(15),  @c\_authority char(1)  AS  IF EXISTS(  SELECT \* FROM custmor  WHERE c\_no = @c\_no AND c\_name = @c\_name  AND c\_authority='T'  )  SELECT @c\_authority = 1  ELSE  SELECT @c\_authority = 0  （2）调用存储过程      **3.5 定义触发器描述业务规则与约束**   1. 新增客户信息时触发，检查是否有历史交易记录，没有的话注明该客户尚无交易记录。   CREATE TRIGGER auto\_custmor  ON custmor  AFTER INSERT  AS  BEGIN  UPDATE custmor  set c\_assess= '该客户尚无交易记录'  WHERE c\_assess IS NULL  END   1. 当修改工作状态为F的设计师信息时触发，发生回滚并提示现在无法修改。   CREATE TRIGGER roll\_F\_designer  ON designer  FOR UPDATE  AS  BEGIN  DECLARE @d\_state char  SET @d\_state=(SELECT d\_state FROM designer)  IF @d\_state='F'  ROLLBACK  END     1. 当产品信息新增，更新和删除时触发提示   新增：  CREATE TRIGGER product\_insert  ON product  AFTER INSERT  AS  BEGIN  print'您新增了一条产品信息'  END  更新：  CREATE TRIGGER product\_update  ON product  AFTER UPDATE  AS  BEGIN  print'您更新了一条产品信息'  END  删除：  CREATE TRIGGER product\_delete  ON product  AFTER DELETE  AS  BEGIN  print'您删除了一条产品信息'  END  **3.6定义事务保障业务处理的一致性与完整性。**    **3.7 运用乐观锁和并发控制保障多任务环境的数据安全性**  服装ERP系统的最大问题是大量客户导致的大量并发操作，导致的产品设计师对产品信息更新操作，在下面的环节中对并行进行处理。  当多个设计师和产品经理同时对产品表作更新或者新增删除操作时会带来以问题:  （1）丢失更新，两个设计师读同一数据并进行修改,其中一个设计师的修改结果破坏了另一个修改的结果  （2）脏读，A设计师修改了数据,随后B设计师又读出该数据,但A设计师因为某些原因取消了对数据的修改,数据恢复原值,此时B设计师得到的数据就与数据库内的数据产生了不一致  （3）不可重复读，A设计师读取数据,随后B设计师读出该数据并修改,此时A设计师再读取数据时发现前后两次的值不一致  并发处理控制的主要方法是封锁,即一个设计师进行操作的一段时间内禁止其它设计师做某些操作，以避免产生数据不一致。  这里我使用更新锁U和排它锁X。  新建两个连接  在第一个连接中执行以下语句  BEGIN TRAN  UPDATE product  SET p\_price='3666'  WHERE p\_no='5'  WAITFOR DELAY '00:01:00'  COMMIT TRAN  在第二个连接中执行以下语句  BEGIN TRAN  SELECT \* FROM product  WHERE p\_no='2'  COMMIT TRAN  如果同时执行这两条语句，则查询必须等更新执行完才能执行，或者说要等待1分钟后才可以执行。  **4身份与权限管理策略**  本ERP系统中有五类用户：产品经理，中长期规划管理经理，主生产计划员，销售员，客户，创建用户如下。  CREATE LOGIN r\_manager  WITH PASSWORD='ilovemanage'  GO  USE Reserve  CREATE USER r\_manager  GO  CREATE LOGIN r\_plan\_manager  WITH PASSWORD='ilovedesign'  GO  USE Reserve  CREATE USER r\_maser\_planer  GO  CREATE LOGIN r\_seller  WITH PASSWORD='ilovebuy'  GO  USE Reserve  CREATE USER r\_custmor  GO  CREATE LOGIN r\_crm  WITH PASSWORD='ilovecrn'  GO  USE Reserve  CREATE USER r\_crm  GO  USE Reserve  CREATE USER DB\_Reserve  GO  可视化界面  Snipaste_2017-12-13_16-04-27  在下面服务器角色列表框中，列出了系统的固定服务器角色。在这些固定服务器角色的左端有相应的复选框，打勾的复选框表示该登录帐号是相应的服务器角色成员。  Snipaste_2017-12-13_16-05-45  用户映射-Snipaste_2017-12-13_16-06-32  状态  状态Snipaste_2017-12-13_16-07-58  下面我通过SQL语句分别添加产品经理，中长期规划管理经理，主生产计划员，销售员，客户等五个角色  EXEC sp\_addrole 'manager',' r\_manager'  GO  EXEC sp\_addrole ' plan\_manager ',' r\_plan\_manager '  GO  EXEC sp\_addrole ' maser\_planer ',' r\_maser\_planer '  GO  EXEC sp\_addrole ' seller ',' r\_seller '  GO  EXEC sp\_addrole ' custmor',' r\_custmor '  GO  登陆用户Snipaste_2017-12-13_16-23-54  **4.3 访问许可权限管理**  服装产品管理系统数据库对五类用户分别给予如下的权限。   1. 对产品经理，授予查看产品经理信息的权限，查看、新增、修改、删除产品设计师信息的权限，查看、新增、修改、删除产品信息的权限，查看客户信息的权限、查看和删除议价单信息的权限，查看需求文档的权限。   --授权产品经理  GRANT SELECT  ON manager  TO r\_manager  GO  GRANT SELECT,INSERT,UPDATE,DELETE  ON designer  TO r\_manager  GO  GRANT SELECT,INSERT,UPDATE,DELETE  ON product  TO r\_manager  GO  GRANT SELECT  ON custmor  TO r\_manager  GO  GRANT SELECT,DELETE  ON Price  TO r\_manager  GO  GRANT SELECT  ON doc  TO r\_manager  GO   1. 对中长期规划管理经理产品经理，授予查看销售订单，主生产计划单，需求文档的权限。   --授权中长期归规划管理产品经理  GRANT SELECT  ON plan\_manager  TO r\_plan\_manager  GO  GRANT SELECT  ON product  TO r\_plan\_manager  GO  GRANT SELECT  ON custmor  TO r\_plan\_manager  GO   1. 对主生产计划员，授予查看销售订单，查看需求文档，查看、新增、修改、删除主生产计划单，   GRANT SELECT  ON maser\_planer  TO r\_maser\_planer  GO  GRANT SELECT,INSERT,UPDATE,DELETE  ON product  TO r\_maser\_planer  GO  GRANT SELECT  ON doc  TO r\_maser\_planer  GO  GRANT SELECT,UPDATE  ON Price  TO r\_maser\_planer  GO   1. 对销售员，授予查看、新增、修改，删除销售订单的权限   GRANT SELECT,INSERT,UPDATE,DELETE  ON product  TO r\_seller  GO   1. 对客户，授予新增、修改、删除需求文档的权限，修改议价单的权限，查询产品信息的权限和新增客户信息的权限。   USE Reserve  GO  GRANT SELECT,UPDATE,DELETE  ON doc  TO r\_custmor  GO  GRANT UPDATE  ON Price  TO r\_custmor  GO  GRANT SELECT  ON product  TO r\_custmor  GO  GRANT INSERT  ON custmor  TO r\_custmor  GO | | | | | |
| 第三部分 结果与讨论（可加页）  一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）  1.此实验在物理层面综合运用了主健约束，外键约束，非空约束，唯一约束和CHECK约束，较好的实现了大量数据的存储，实现了主生产计划单的制定。并且通过制定三大类索引，十分有效的提高了数据查询的效率。  2.在数据访问方面，通过视图的设定以及游标的运用，实现了行级约束。  3.业务逻辑层面，定义了不同的函数实现了封装以及一些常用功能。通过定义触发器减少了错误操作导致的安全性问题，并通过事务和锁技术对并发处理实现有效控制。  4.在安全性层面，通过身份认证，用户管理，角色管理四个方面有效实现了数据的安全事务。  二、小结、建议及体会  此次的实验将数据库的理论知识同具体实践有机的结合了起来，一方面加深了我数据库理论方面的知识，SQL语法的理解，帮助我系统的梳理，理解数据库的相关知识点，又提高了我的实战操作能力，拓展了我的视角，从原来的理论编程视角转换为具有实战操作层面的开发视角。  数据库的学习在最开始的时候我是感到很力不从心的，知识点一懂半懂，不知道知识点有什么用。同时数据库语言和C编程语言 有很大的不同，那些在语言上都可以自行上机实验。同时这学期的数据库课程和ERP课程是在一起的，我在做ERP作业的时候，由于对数据库的不熟悉，总是在做作业的时候把数据库相关的作业冷落到一边，也因此导致在做大作业的时候显得有点力不从心。但是通过不断的练习，使用加深了我的印象，不断地思考加深了我的理解，并最终形成了体系，这对我的学习来说，是有非常大的直接帮助。  作为一个男生学习与信管有关地知识，在学习地过程中，总会不自觉的偏向写代码方向。总是认为只有写出了代码，运行出了成果才算真正地学到了东西，因为代码是可以摸得到看得到的，而思想方法，理念则是很抽象，虚无飘渺的东西。但是通过这学期的上机实验，我开始逐渐转变自己地视角，不再是原来地狭隘的编程视角，而是更为开阔的开发视角。不再拘泥于实现某一个具体的功能，而是着眼于为客户实现更加全局的功能。面向需求来考虑，极大的开阔了我的视野。  当然，逆水行舟，不进则退，我知道在很多方面都存在着很多的不足，对四五范式的理解都还不是很到位，还需要不断的向老师请教。  最后，感谢任课老师辛勤的付出和无私的帮助！ | | | | | |
| 第四部分 评分标准（教师可自行设计）及成绩   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 观测点 | 考核目标 | 权重 | 得分 | | 实验预习 | 1. 预习报告 2. 提问 3. 对于设计型实验，着重考查设计方案的科学性、可行性和创新性 | 对实验目的和基本原理的认识程度，对实验方案的设计能力 | 20% |  | | 实验过程 | 1. 是否按时参加实验 2. 对实验过程的熟悉程度 3. 对基本操作的规范程度 4. 对突发事件的应急处理能力 5. 实验原始记录的完整程度 6. 同学之间的团结协作精神 | 着重考查学生的实验态度、基本操作技能；严谨的治学态度、团结协作精神 | 30% |  | | 结果分析 | 1. 所分析结果是否用原始记录数据 2. 计算结果是否正确 3. 实验结果分析是否合理 4. 对于综合实验，各项内容之间是否有分析、比较与判断等 | 考查学生对实验数据处理和现象分析的能力；对专业知识的综合应用能力；事实求实的精神 | 50% |  | | 该项实验报告最终得分 | | |  | |   教师签名：　　　　　　　　。 | | | | | |