**《数据库原理及应用》课程设计**

**设计题目 机票预订信息系统**

**系 部 信息技术工程学院**

**班 级 网络 1302**

**学 号**

**姓 名**

**2016 年 1 月**

目录

[一、 需求分析 3](#_Toc441154963)

[ 背景分析 3](#_Toc441154964)

[ 需求分析 3](#_Toc441154965)

[1) 系统功能： 3](#_Toc441154966)

[2) 系统信息： 3](#_Toc441154967)

[3) 数据流图： 3](#_Toc441154968)

[4) 数据字典： 4](#_Toc441154969)

[二、 结构设计 5](#_Toc441154970)

[ E-R图 5](#_Toc441154971)

[飞机E-R图，如图4-1所示 6](#_Toc441154972)

[航班E-R图，如图4-2所示。 6](#_Toc441154973)

[机票E-R图，如图4-3所示。 6](#_Toc441154974)

[旅客E-R图，如图4-4所示。 6](#_Toc441154975)

[账单E-R图，如图4-5所示。 7](#_Toc441154976)

[航班坐位E-R图，如图4-6所示。 7](#_Toc441154977)

[取票通知E-R图，如图4-7所示。 7](#_Toc441154978)

[总E-R图，如图4-8所示。 8](#_Toc441154979)

[ 关系模式 9](#_Toc441154980)

[基本表 9](#_Toc441154981)

[虚表（视图） 9](#_Toc441154982)

[三、 数据库实现 9](#_Toc441154983)

[ 创建表空间与赋权限 9](#_Toc441154984)

[ 飞机信息表建立 9](#_Toc441154985)

[ 航班信息表建立 10](#_Toc441154986)

[ 机票信息表建立 10](#_Toc441154987)

[ 旅客信息表建立 10](#_Toc441154988)

[ 账单信息表建立 10](#_Toc441154989)

[ 航班坐位视图 10](#_Toc441154990)

[ 取票通知视图 11](#_Toc441154991)

[ 外键语句 11](#_Toc441154992)

[四、 数据库实施 11](#_Toc441154993)

[ 添加飞机 11](#_Toc441154994)

[ 添加航班 11](#_Toc441154995)

[ 添加用户 12](#_Toc441154996)

[ 预定机票(存储过程) 12](#_Toc441154997)

[ 打印取票通知(存储函数) 13](#_Toc441154998)

[ 生成账单(触发器和存储过程) 13](#_Toc441154999)

[五、 总结 14](#_Toc441155000)

1. 需求分析

* 背景分析

以信息技术为主要标志的科技进步日新月异，高科技成果向现实生产力的转化越来越快。 纵观全球经济发展， 信息技术和信息产业已经成为经济增长的主要推动力之一， 正在改变着传统的生产和经营方式以至生活方式， 发达国家经过产业结构的升级和经济结构 的转型已进入信息经济阶段。 机票预订系统也开始影响着人们日常生活和出行，并且变得越来越重要。而原有的系统随着航空公司载客量的迅猛增长和人们对便捷性要求的提高，已经变得无法满足需求。原有的系统不仅效率比较低下，而且在安全性、准确性等方面有很多不足。为了实现航空公司以及旅游行业的现代化管理，进一步提高工作效率，方便旅客，需要开发一个机票预订系统。

* 需求分析
  + - 1. 系统功能：
* 航班信息的输入，修改查询。
* 航班座位信息的输入，修改查询。
* 根据旅客输入的基本信息，为旅客安排航班，打印取票通知和帐单。
* 旅客在飞机起飞前一天凭取票通知交款取票。
* 旅客能够退订机票。
* 能够查询每个航班的预定情况、计算航班的满座率。
  + - 1. 系统信息：
      * **航班信息**: 航班编号、飞机编号、起飞地、目的地、起飞时间、飞行时间等。
      * **飞机信息**：飞机编号、飞机型号、座位数。
      * **机票信息**：航班编号、旅客编号、座位号、票价等。
      * **旅客信息**：姓名、性别、联系电话、身份证号等。
      1. 数据流图：

工作人员

飞机表

航班表

机票表

用户表

账单表

旅客

* + - 1. 数据字典：

根据总体结构图设计机票预定系统基本表结构，其相应标的定义如下：

表4-1 飞机信息<plain>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 是否非空 | 主关键字 | 备注 |
| plain\_id | VARCHAR2 | 10 |  | Y | 飞机编号 |
| model | VARCHAR2 | 10 | N |  | 飞机型号 |
| seat | int | 2 | N |  | 总座位 |

表4-2 航班信息< flight >

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 是否非空 | 主关键字 | 备注 |
| flight\_id | VARCHAR2 | 10 |  | Y | 航班编号 |
| plain\_id | VARCHAR2 | 10 | N |  | 飞机编号 |
| start | VARCHAR2 | 10 | N |  | 起飞地 |
| end | VARCHAR2 | 10 | N |  | 目的地 |
| departure\_time | datetime |  | N |  | 起飞时间 |
| duration | int | 4 | N |  | 飞行时间 |

表4-3 机票信息<tickets>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 是否非空 | 主关键字 | 备注 |
| tickets\_id | VARCHAR2 | 10 |  | Y | 机票编号 |
| flight\_id | VARCHAR2 | 10 |  | Y | 航班编号 |
| traveler\_id | VARCHAR2 | 10 | N |  | 旅客编号 |
| seating\_list | int | 4 | N |  | 座位号 |
| cabin\_rating | VARCHAR2 | 10 | N |  | 机舱等级 |
| updata\_time | datetime |  | N |  | 更新时间 |
| status | int | 10 | N |  | 状态 |

表4-4 旅客信息<traveler>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 是否非空 | 主关键字 | 备注 |
| traveler\_id | VARCHAR2 | 10 |  | Y | 旅客编号 |
| name | VARCHAR2 | 10 | N |  | 姓名 |
| sex | int | 1 | N |  | 性别 |
| id\_value | VARCHAR2 | 20 | N |  | 身份证号 |
| phone | VARCHAR2 | 11 | N |  | 联系方式 |

表4-5 账单信息<bill>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 是否非空 | 主关键字 | 备注 |
| bill\_id | VARCHAR2 | 10 |  | Y | 账单编号 |
| traveler\_id | VARCHAR2 | 10 |  | Y | 旅客编号 |
| tickets\_id | VARCHAR2 | 10 |  | Y | 机票编号 |
| payment | float(2) | N | N |  | 价格 |
| updata\_time | datetime |  | N |  | 更新时间 |
| status | int | 4 | N |  | 付款状态 |

1. 结构设计

* E-R图

根据需求分析抽象出信息结构，可得该系统的E-R图。

飞机E-R图，如图4-1所示。

飞机

图4-1 飞机E-R图

航班E-R图，如图4-2所示。

航班

图4-2 航班E-R图

机票E-R图，如图4-3所示。

机票

图4-3 机票E-R图

旅客E-R图，如图4-4所示。

旅客

图4-4 旅客E-R图

账单E-R图，如图4-5所示。

账单

图4-5 账单E-R图

航班坐位E-R图，如图4-6所示。

航班坐位

图4-6 航班坐位E-R图

取票通知E-R图，如图4-7所示。

取票通知

图4-7 取票通知E-R图

总E-R图，如图4-8所示。

飞机

航班

有

1

1

机票

包含

1

m

购买

1

旅客

1

账单

属于

1

1

图4-8 总E-R图

* 关系模式

基本表

飞机信息表（飞机编号、飞机型号、飞机座位）

航班信息表（航班编号、飞机编号、 起飞地、目的地、起飞时间、飞行时间）

机票信息表（机票编号、航班编号、旅客编号、机舱等级、座位号、更新时间、状态）

旅客信息表（旅客编号、姓名、性别、联系电话、身份证号）

账单表 (账单编号、旅客编号、机票编号、价格、更新时间、状态)

虚表（视图）

航班坐位情况表 (机票信息表.航班编号、机票信息表.座位号、机票信息表.状态)

取票通知表 (机票信息表.航班编号、机票信息表. 旅客编号、 机票信息表.座位号、机票信息表.状态)

1. 数据库实现
   * + - * 创建表空间与赋权限

create temporary tablespace HKSP\_TEMP

tempfile'C:\oraclexe\app\oracle\oradata\XE\HKSP\_TEMP.dbf'

size 50M

autoextend on

next 50M maxsize 1048M

extent management local;

create tablespace HKSP\_DATA

atafile 'C:\oraclexe\app\oracle\oradata\XE\HKSP\_DATA.dbf'

size 50M

autoextend on

next 50M maxsize 1048M

extent management local;

create user HKSP identified by 123456

default tablespace HKSP\_DATA

temporary tablespace HKSP\_TEMP;

grant connect,resource,dba to HKSP;

* + - * + 飞机信息表建立

CREATE TABLE "HKSP"."plain" (

"plain\_id" VARCHAR2(10 BYTE) NOT NULL primary key ,

"model" VARCHAR2(10 BYTE) NOT NULL ,

"seat" NUMBER NOT NULL

)

* + - * + 航班信息表建立

CREATE TABLE "HKSP"."flight" (

"flight\_id" VARCHAR2(10 BYTE) NOT NULL primary key,

"plain\_id" VARCHAR2(10 BYTE) NOT NULL ,

"start" VARCHAR2(20 BYTE) NOT NULL ,

"end" VARCHAR2(20 BYTE) NOT NULL ,

"departure\_time" DATE NOT NULL ,

"duration" NUMBER NOT NULL

)

* + - * + 机票信息表建立

CREATE TABLE "HKSP"."tickets" (

"tickets\_id" VARCHAR2(10 BYTE) NOT NULL primary key,

"flight\_id" VARCHAR2(10 BYTE) NOT NULL ,

"traveler\_id" VARCHAR2(10 BYTE) NOT NULL ,

"seating\_list" NUMBER NOT NULL ,

"cabin\_rating" VARCHAR2(10 BYTE) NOT NULL ,

"updata\_time" DATE NOT NULL ,

"status" NUMBER NOT NULL

)

* + - * + 旅客信息表建立

CREATE TABLE "HKSP"."traveler" (

"traveler\_id" VARCHAR2(10 BYTE) NOT NULL primary key ,

"name" VARCHAR2(10 BYTE) NOT NULL ,

"sex" NUMBER NOT NULL ,

"id\_value" VARCHAR2(20 BYTE) NOT NULL ,

"phone" VARCHAR2(11 BYTE) NOT NULL

)

* + - * + 账单信息表建立

CREATE TABLE "HKSP"."bill" (

"bill\_id" VARCHAR2(10 BYTE) NOT NULL primary key,

"traveler\_id" VARCHAR2(10 BYTE) NOT NULL ,

"tickets\_id" VARCHAR2(10 BYTE) NOT NULL ,

"payment" FLOAT(10) NOT NULL ,

"updata\_time" DATE NOT NULL ,

"status" NUMBER NOT NULL

)

* + - * + 航班坐位视图

CREATE OR REPLACE FORCE VIEW "HKSP"."seating\_view" AS

SELECT

HKSP."tickets"."flight\_id",

HKSP."tickets"."seating\_list",

HKSP."tickets"."status"

FROM

HKSP."tickets";

* + - * + 取票通知视图

CREATE OR REPLACE FORCE VIEW "HKSP"."notice\_view" AS

SELECT

HKSP."tickets"."flight\_id",

HKSP."tickets"."traveler\_id",

HKSP."tickets"."status"

FROM

HKSP."tickets"

WHERE

HKSP."tickets"."status" = 2;

* + - * + 外键语句

ALTER TABLE "HKSP"."bill" ADD FOREIGN KEY ("traveler\_id")

REFERENCES "HKSP"."traveler" ("traveler\_id") ON DELETE CASCADE;

ALTER TABLE "HKSP"."bill" ADD FOREIGN KEY ("tickets\_id")

REFERENCES "HKSP"."tickets" ("tickets\_id") ON DELETE CASCADE;

ALTER TABLE "HKSP"."flight" ADD FOREIGN KEY ("plain\_id")

REFERENCES "HKSP"."plain" ("plain\_id") ON DELETE CASCADE;

ALTER TABLE "HKSP"."tickets" ADD FOREIGN KEY ("traveler\_id")

REFERENCES "HKSP"."traveler" ("traveler\_id") ON DELETE CASCADE;

1. 数据库实施

* 添加飞机

INSERT INTO "HKSP"."plain" VALUES ('F-T1235', 'T12306', '100');

INSERT INTO "HKSP"."plain" VALUES ('F-T1236', 'T12306', '100');

INSERT INTO "HKSP"."plain" VALUES ('F-T1237', 'T12306', '100');

INSERT INTO "HKSP"."plain" VALUES ('F-T1234', 'T12306', '100');

* 添加航班

INSERT INTO "HKSP"."flight" VALUES ('MH370', 'F-T1235', '北京', '马来西亚', TO\_DATE('2016-01-18 10:12:09', 'YYYY-MM-DD HH24:MI:SS'), '5');

INSERT INTO "HKSP"."flight" VALUES ('MH371', 'F-T1236', '北京', '西安', TO\_DATE('2016-01-19 09:20:13', 'YYYY-MM-DD HH24:MI:SS'), '4');

INSERT INTO "HKSP"."flight" VALUES ('MH372', 'F-T1237', '北京, '上海', TO\_DATE('2016-01-20 20:14:17', 'YYYY-MM-DD HH24:MI:SS'), '11');

INSERT INTO "HKSP"."flight" VALUES ('MH373', 'F-T1234', '北京', '杭州', TO\_DATE('2016-01-19 05:29:21', 'YYYY-MM-DD HH24:MI:SS'), '3');

* 添加用户

INSERT INTO "HKSP"."traveler" VALUES ('LK00001', '小明', '1', '140522199311111111', '123456781');

INSERT INTO "HKSP"."traveler" VALUES ('LK00002', '小红', '2', '140522199311111112', '123456782');

INSERT INTO "HKSP"."traveler" VALUES ('LK00003', '小芳', '2', '140522199311111113', '123456783');

INSERT INTO "HKSP"."traveler" VALUES ('LK00004', '小小', '1', '140522199311111114', '123456784');

INSERT INTO "HKSP"."traveler" VALUES ('LK00005', '小亮', '1', '140522199311111115', '123456785');

INSERT INTO "HKSP"."traveler" VALUES ('LK00006', '小心', '2', '140522199311111116', '123456786');

* 预定机票(存储过程)

CREATE OR REPLACE

PROCEDURE book\_tickets (flight\_id IN VARCHAR2, traveler\_id IN VARCHAR2, seating IN NUMBER)

AS

ticketsid "tickets"."tickets\_id" %TYPE;

test\_ticketsid NUMBER;

test\_cabin "tickets"."cabin\_rating" %TYPE;

temp varchar(32);

test\_value NUMBER:=0;

BEGIN

--生成机票ID

temp:=sys\_guid();

ticketsid:= substr(temp,1,6)||substr(temp,28,4);

--测试座位是否被预定

SELECT count(\*) INTO test\_ticketsid

FROM "tickets"

WHERE "seating\_list" = seating;

--查看结果

IF test\_ticketsid = 0 THEN

--座位未被预定

--判断仓位等级

test\_value := seating/10;

CASE

WHEN test\_value < 1 THEN

test\_cabin:='商务舱';

WHEN test\_value < 2 THEN

test\_cabin:='二等舱';

ELSE

BEGIN

test\_cabin:='经济舱';

END;

END CASE;

--预定机票

INSERT INTO "HKSP"."tickets" VALUES (ticketsid, flight\_id, traveler\_id, seating, test\_cabin, SYSDATE,1);

DBMS\_OUTPUT.PUT\_LINE('预定成功');

ELSE

--座位已被预定

DBMS\_OUTPUT.PUT\_LINE('座位已被预定');

END IF;

END;

* 打印取票通知(存储函数)

CREATE OR REPLACE

FUNCTION get\_voucher (flight\_id IN VARCHAR2, traveler\_id IN VARCHAR2)

RETURN VARCHAR2

AS

test\_tickets "tickets"."tickets\_id" %TYPE;

BEGIN

--获得机票ID

SELECT "tickets"."tickets\_id" INTO test\_tickets

FROM "tickets"

WHERE "flight\_id" = flight\_id AND "traveler\_id" = traveler\_id;

--更新状态

UPDATE "tickets" SET "status"=2

WHERE "tickets"."tickets\_id" = test\_tickets;

--返回机票ID

RETURN test\_tickets;

EXCEPTION

WHEN OTHERS THEN

RETURN '你没有预定机票';

END;

* 生成账单(触发器和存储过程)

――插入账单　存储过程

CREATE OR REPLACE

PROCEDURE insert\_bill (traveler\_id IN VARCHAR2, tickets\_id IN VARCHAR2,status IN NUMBER)

AS

temp varchar2(32);

uuid varchar2(10);

BEGIN

temp:=sys\_guid();

uuid:=substr(temp,1,10);

IF status = 2 THEN

INSERT INTO "HKSP"."bill" ("bill\_id","traveler\_id","tickets\_id","payment","updata\_time","status") VALUES (uuid, traveler\_id, tickets\_id, 100 , SYSDATE, 1);

END IF;

END;

――机票状态触发器

CREATE OR REPLACE TRIGGER bill\_add

AFTER UPDATE OF "status" ON "HKSP"."tickets"

FOR EACH ROW

BEGIN

INSERT\_BILL(:OLD."traveler\_id",:OLD."tickets\_id",:NEW."status");

END;

1. 总结

经过一周的努力，本次课程设计终于完成了。通过这次课程设计，我对数据库这门课程有了更深入的理解，数据库是一门实践性较强的课程，同时我明白了理论与实际应用相结合的重要性，掌握并熟练运用SQL语句，提高了我的综合运用所学知识的能力。

1. 独立创建表空间，创建用户，并赋予权限。
2. 创建表，指定主键，指定外键。
3. Select ,include,update,delect等语句的使用。
4. 存储过程，存储函数，触发器的使用。
5. 对数据库的创建过程有了一个系统的理解。

在本次课程设计和过程中，由于时间不是很长，再加上要求独立完成，系统需求分析上可能不是很全面，程序中还存在很多缺，希望老师能给予批评和指导。经过这次课程设计，我的知识得到了很大，经验也更加丰富。希望能在今后的不断的学习中和努力中，把程序做得更好。

最后，感谢指导老师给予的帮助。