宾馆客房管理系统

**需求描述：**

该酒店管理系统旨在为酒店提供包括员工管理、客房管理、客户管理、订单管理、报表维护等方面的一体化解决方案。同时，还提供用户登录和密码管理等功能。其业务需求涵盖了从客户预订、入住到退房的整个流程管理，并提供详细的报表支持，以便于酒店的日常运营和决策。

员工管理：管理员可以添加新员工的信息，如姓名、职位、联系方式等；删除现有员工的信息，确保信息的准确性和及时性；更新员工的详细信息，维护员工档案的完整性；支持通过姓名、职位等多种方式查询员工信息，方便管理。

客房管理：管理客户的预约信息。录入客户的预订信息，生成预订单；处理客户的预约取消请求，更新预订状态。处理客户的入住登记，包括个人入住和团体入住，记录单个客户的入住信息，分配房间，处理团体客户的入住需求，分配多个房间。管理客户退房手续，结算费用，更新房间状态。管理酒店的房间信息，包括添加新房间和修改现有房间信息，录入新房间的详细信息，如房间号、房型、价格等，更新现有房间的信息，保持房间资料的准确性。

客户管理：提供多种方式查询客户信息，方便服务和管理；录入新客户的信息，建立客户档案。

订单管理：查询客户的预订信息，跟踪预订状态；查询客户的订单信息，包括已完成和未完成的订单。

报表和维护：定期备份系统数据库，确保数据安全；导出数据库数据，进行本地存储或迁移；导出特定数据表，便于分析和审查。生成和展示各种管理报表，提供数据支持和决策依据；导出报表数据表格，便于记录和分享。

登录功能：用户通过用户名和密码登录系统，确保系统安全；提供找回密码功能，帮助用户重新获取访问权限；用户可以修改个人密码，确保账号的安全性。

**功能描述：**

**员工管理功能**

添加员工：录入员工的详细信息，如姓名、职位、联系方式，建立员工档案。

删除员工：删除员工信息，确保信息的准确性和及时性。

修改员工：更新员工的详细信息，维护员工档案的完整性。

查询员工信息：通过姓名、职位等多种方式查询员工信息，方便管理和调度。

**客房管理功能**

预约功能

预约：录入并管理客户的预约信息，生成预订单。

取消预约：处理预约取消请求，更新预订状态。

入住功能

个人入住：记录客户入住信息，分配房间。

团体入住：处理团体客户入住，分配多个房间。

房间管理

添加房间：录入新房间信息，如房间号、房型、价格等，更新系统房间数据。

修改房间：更新现有房间信息，保持数据准确性。

**客户管理功能**

查询客户：通过多种方式查询客户信息，方便服务和管理。

添加客户：录入新客户信息，建立客户档案。

**订单管理功能**

查询预订：查询并跟踪客户预订信息。

查询订单：查询客户的订单信息，管理订单状态。

**报表和维护功能**

数据库备份：定期备份数据库，确保数据安全。

导出数据库：导出数据库数据，便于存储和迁移。

导出数据表：导出特定数据表，方便分析和审查。

报表展示

生成报表：生成各类管理报表，支持决策。

导出报表：导出报表数据，便于记录和分享。

**登录功能**

登录：用户通过用户名和密码登录系统。

找回密码：提供找回密码功能，确保用户可以重新获取登录权限。

修改密码：用户可以修改个人密码，确保账号安全。

# 概念结构设计阶段

员工staff：{员工编号sid，员工姓名sname，员工性别ssex，登录账号susername，登录密码spassword，权限等级srole }

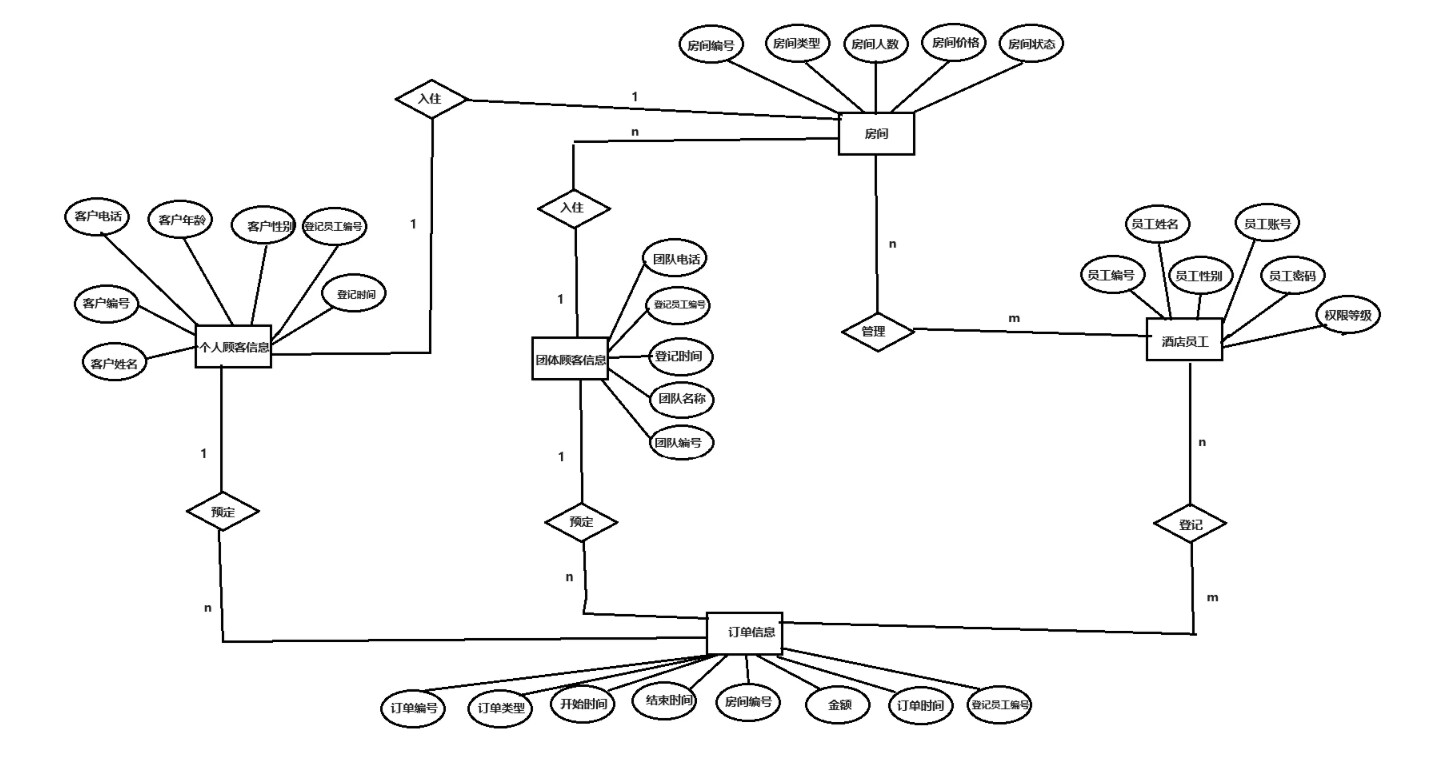
个人客户client：{客户姓名cname，客户编号cid，客户电话cphone，客户年龄cage，客户性别csex，登记员工编号register\_sid,登记时间register\_time}

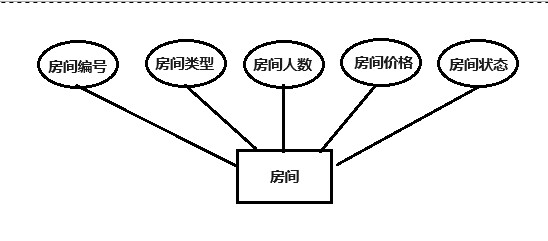
客房room：{客房编号rid，客房类型rtype，客房人数rsize，客房价格rprice，客房状态rstatus }

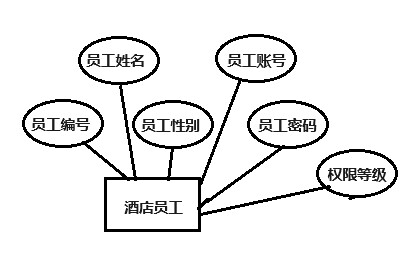
订单hotelorder：{订单编号id，订单类型ordertype，开始时间start\_time，结束时间end\_time，客房编号rid，订单金额money，订单时间order\_time,登记员工编号register\_sid}

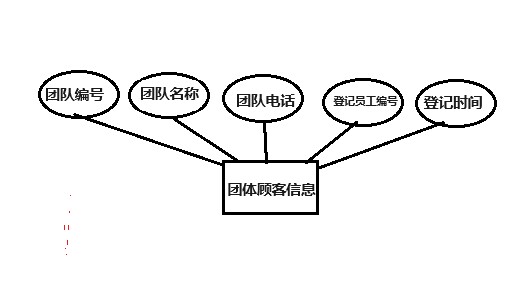
团队客户：{团队名称tname，团队编号tid，团体电话tphone, 登记员工编号check\_in\_sid,登记时间register\_time }

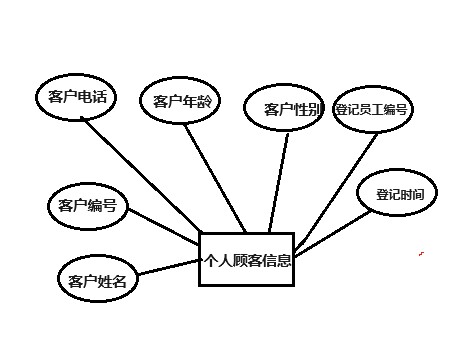
## ER 图

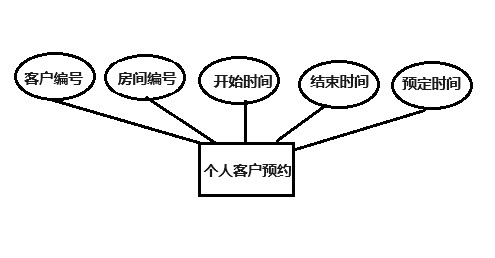


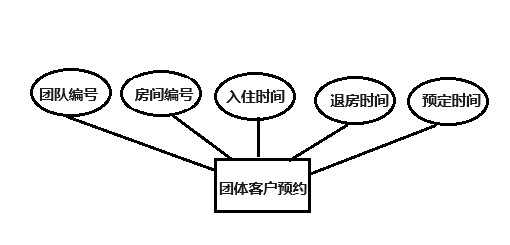


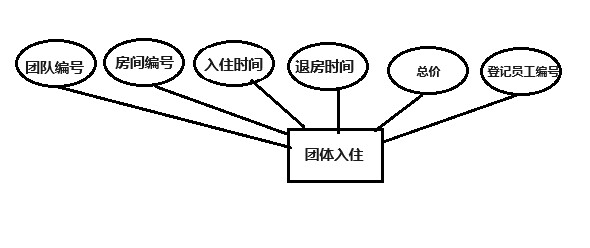


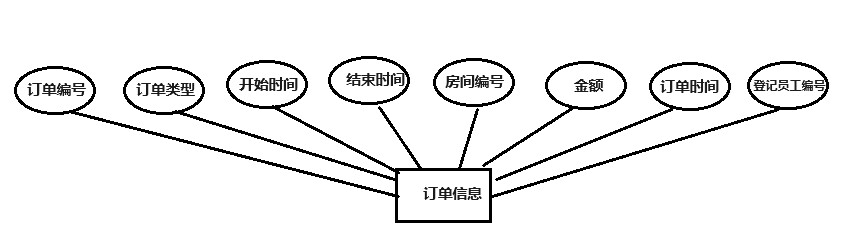












# 逻辑结构设计

团体客户和客房的入驻联系是1：n，可以将它们之间的联系与n端实体团体客户合并；客户与订单的预订关系也是1：n，同样可以将其中的联系与n端实体合并，而员工与客房之间的管理和员工与订单的登记联系方式是n：m，将其联系转化称独立的关系模式。具体的基本E-R图向关系模式的转化如下所述：

客房（客房编号，客房类型，客房容量，客房价格，客房状态）

员工（员工编号，员工姓名，员工性别，登录账号，登录密码，权限等级）

个人客户（客户编号，客户姓名，客户电话，客户年龄，客户性别，登记时间，登记员工编号）

团队客户（团队编号，团队名称，团体电话，登记时间，登记员工编号）

预订（预约编号，类型,客户编号,客房编号,入住时间，退房时间，预订时间, 登记员工编号）

前台登记（登记编号,类型,客户编号,客房编号,入住时间，退房时间，总价,登记员工编号）

订单（订单编号，订单类型，开始时间，结束时间，房间编号,订单金额，订单时间,登记员工编号）

# 物理结构设计

表1 房间信息表room

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| rid | varchar(255) | Not null (主键) | 房间编号 |
| rtype | varchar(255) | Not null | 房间类型 |
| rsize | SMALLINT | Not null | 房间人数 |
| rprice | varchar(255) | Not null | 房间价格 |
| rstatus | varchar(255) | Not null | 房间状态 |

表2 酒店员工的个人信息表staff

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| sid | varchar(255) | Not null (主键) | 员工编号 |
| sname | varchar(255) | Not null | 员工姓名 |
| ssex | varchar(255) | 可为空 | 员工性别 |
| susername | varchar(255) | Not null | 员工账号 |
| spassword | varchar(255) | Not null | 员工密码 |
| srole | varchar(255) | Not null | 权限等级 |

表3 个人顾客信息表client

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| cname | varchar(255) | Not null | 客户姓名 |
| cid | varchar(255) | Not null (主键) | 客户编号 |
| cphone | varchar(255) | 可为空 | 客户电话 |
| cage | varchar(255) | Not null | 客户年龄 |
| csex | varchar(255) | 可为空 | 客户性别 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |
| register\_time | timestamp | 可为空 | 登记时间 |

表4 团体顾客信息表team

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| tname | varchar(255) | Not null | 团队名称 |
| tid | varchar(255) | Not null (主键) | 团队编号 |
| tphone | varchar(255) | 可为空 | 团队电话 |
| check\_in\_sid | varchar(255) | 可为空 | 登记员工编号 |
| register\_time | timestamp | 可为空 | 登记时间 |

表5预约记录表booking

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| booking\_id | int | Not null(主键) | 预约编号 |
| type | enum('individual','team') | Not null | 预约类型 |
| client\_or\_team\_id | varchar(255) | Not null | 客户编号 |
| room\_id | varchar(255) | Not null | 房间编号 |
| start\_time | date | 可为空 | 入住时间 |
| end\_time | date | 可为空 | 退房时间 |
| booking\_time | timestamp | 可为空 | 预订时间 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

表6前台登记记录表checkin

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| checkin\_id | int | Not null(主键) | 登记编号 |
| type | enum('individual','team') | Not null | 登记类型 |
| client\_or\_team\_id | varchar(255) | Not null | 客户编号 |
| room\_id | varchar(255) | Not null | 房间编号 |
| start\_time | date | 可为空 | 入住时间 |
| end\_time | date | 可为空 | 退房时间 |
| total\_price | decimal(10,2) | 可为空 | 总价 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

表7 酒店所有订单信息和收入情况表hotelorder

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| id |  | Not null (主键) | 订单编号 |
| ordertype | varchar(255) | Not null | 订单类型 |
| start\_time | date | Not null | 订单开始时间 |
| end\_time | date | Not null | 订单结束时间 |
| rid | varchar(255) | Not null | 房间编号 |
| money | decimal(10,2) | 可为空 | 金额 |
| order\_time | timestamp | 可为空 | 订单时间 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

# 数据库实现

**数据库的建立:**

CREATE DATABASE `HotelSystem` DEFAULT CHARACTER SET = 'utf8mb4';

**表的建立:**

CREATE TABLE `room` (

`rid` VARCHAR(255) NOT NULL,

`rtype` VARCHAR(255) NOT NULL,

`rsize` SMALLINT NOT NULL,

`rprice` VARCHAR(255) NOT NULL,

`rstatus` VARCHAR(255) NOT NULL,

PRIMARY KEY (`rid`),

KEY `rprice` (`rprice`),

KEY `rtype` (`rtype`, `rprice`)

);

CREATE TABLE `staff` (

`sid` varchar(255) NOT NULL,

`sname` varchar(255) NOT NULL,

`ssex` varchar(255) DEFAULT NULL,

`susername` varchar(255) NOT NULL,

`spassword` varchar(255) NOT NULL,

`srole` varchar(255) NOT NULL,

PRIMARY KEY (`sid`),

UNIQUE KEY `susername` (`susername`)

);

CREATE TABLE `client` (

`cname` varchar(255) NOT NULL,

`cid` varchar(255) NOT NULL,

`cphone` varchar(255) DEFAULT NULL,

`cage` varchar(255) NOT NULL,

`csex` varchar(255) DEFAULT NULL,

`register\_sid` varchar(255) DEFAULT NULL,

`register\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`cid`),

KEY `sid` (`register\_sid`),

KEY `cid` (`cid`, `register\_sid`),

CONSTRAINT `sid` FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)

);

CREATE TABLE `team` (

`tname` varchar(255) NOT NULL,

`tid` varchar(255) NOT NULL,

`tphone` varchar(255) DEFAULT NULL,

`check\_in\_sid` varchar(255) DEFAULT NULL,

`register\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`tid`),

KEY `team\_sid` (`check\_in\_sid`),

CONSTRAINT `team\_sid` FOREIGN KEY (`check\_in\_sid`) REFERENCES `staff` (`sid`)

);

CREATE TABLE `booking` (

`booking\_id` INT AUTO\_INCREMENT PRIMARY KEY,

`type` ENUM('individual', 'team') NOT NULL,

`client\_or\_team\_id` VARCHAR(255) NOT NULL,

`room\_id` VARCHAR(255) NOT NULL,

`start\_time` DATE DEFAULT NULL,

`end\_time` DATE DEFAULT NULL,

`booking\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`register\_sid` VARCHAR(255) DEFAULT NULL,

FOREIGN KEY (`room\_id`) REFERENCES `room` (`rid`),

FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)

);

CREATE TABLE `checkin` (

`checkin\_id` INT AUTO\_INCREMENT PRIMARY KEY,

`type` ENUM('individual', 'team') NOT NULL,

`client\_or\_team\_id` VARCHAR(255) NOT NULL,

`room\_id` VARCHAR(255) NOT NULL,

`start\_time` DATE DEFAULT NULL,

`end\_time` DATE DEFAULT NULL,

`total\_price` DECIMAL(10, 2) DEFAULT NULL,

`check\_in\_sid` VARCHAR(255) DEFAULT NULL,

FOREIGN KEY (`room\_id`) REFERENCES `room` (`rid`),

FOREIGN KEY (`check\_in\_sid`) REFERENCES `staff` (`sid`)

);

CREATE TABLE IF NOT EXISTS `hotelorder` (

`id` INT AUTO\_INCREMENT PRIMARY KEY,

`ordertype` VARCHAR(255) NOT NULL,

`start\_time` DATE NOT NULL,

`end\_time` DATE NOT NULL,

`rid` VARCHAR(255) NOT NULL,

`money` DECIMAL(10, 2) DEFAULT NULL,

`order\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`register\_sid` VARCHAR(255) DEFAULT NULL,

FOREIGN KEY (`rid`) REFERENCES `room` (`rid`),

FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)

);

**视图建立:**

-- 创建Customers视图：包含客户的ID、姓名、性别和电话信息

CREATE VIEW Customers AS SELECT Cid, Cname, Csex, Cphone FROM Client;

-- 创建Rooms视图：包含房间的ID、类型、大小、价格和状态

CREATE VIEW Rooms AS

SELECT Rid, Rtype, Rsize, Rprice, rstatus

FROM Room;

-- 创建Living视图：包含入住的房间ID、客户ID、开始时间、结束时间和总价格

CREATE VIEW Living AS

SELECT

checkin\_id,

client\_or\_team\_id,

start\_time,

end\_time,

total\_price

FROM checkin;

-- 创建Administrators视图：包含管理员的ID、姓名和用户名，角色大于1的员工

CREATE VIEW Administrators AS

SELECT Sid, Sname, Susername

FROM Staff

WHERE

Srole > 1

**索引建立:**

--客户电话号码索引：加快通过电话号码查询客户信息的速度

CREATE INDEX idx\_cphone ON client (cphone);

--房间价格索索引：加快通过价格查询房间的速度

CREATE INDEX idx\_rprice ON room (rprice);

--为状态字段添加索引可能会有助于快速筛选

CREATE INDEX idx\_rstatus ON room (rstatus);

--根据入住日期、预订日期或注册时间来查询数据

CREATE INDEX idx\_booking\_start\_time ON booking (start\_time);

CREATE INDEX idx\_booking\_end\_time ON booking (end\_time);

CREATE INDEX idx\_client\_register\_time ON client (register\_time);

CREATE INDEX idx\_checkin\_start\_time ON checkin (start\_time);

CREATE INDEX idx\_checkin\_end\_time ON checkin (end\_time);

**安全性设计:**

-- 创建1级权限员工角色

CREATE ROLE Level1\_Staff;

-- 创建2级权限员工角色

CREATE ROLE Level2\_Staff;

--授予员工权限

GRANT Level1\_Staff TO 'zs123' @'%';

GRANT Level2\_Staff TO 'ranxi' @'%';

1级员工授予部分权限

2级员工授予全部权限

**切记激活角色权限,不然授予用户后无法使用权限**

SET global activate\_all\_roles\_on\_login = ON;

**完整性设计:**

**触发器建立:**

-- 创建入住以及退房时更新房间状态的触发器

DELIMITER $$

CREATE TRIGGER UpdateRoomStatusOnCheckIn

AFTER INSERT ON checkin

FOR EACH ROW

BEGIN

UPDATE room

SET rstatus = '占用'

WHERE rid = NEW.room\_id;

END$$

CREATE TRIGGER UpdateRoomStatusOnCheckOut

AFTER UPDATE ON checkin

FOR EACH ROW

BEGIN

IF NEW.end\_time < CURRENT\_DATE THEN

UPDATE room

SET rstatus = '空闲'

WHERE rid = NEW.room\_id;

END IF;

END$$

DELIMITER;

--预约订单和登记订单自动插入到全部订单表的触发器

DELIMITER $$

CREATE TRIGGER after\_insert\_checkin

AFTER INSERT ON checkin

FOR EACH ROW

BEGIN

-- 更新房间状态为占用

UPDATE room

SET rstatus = '占用'

WHERE rid = NEW.room\_id;

-- 插入订单到 hotelorder 表

INSERT INTO hotelorder (ordertype, start\_time, end\_time, rid, money, order\_time, register\_sid)

VALUES ('前台登记', NEW.start\_time, NEW.end\_time, NEW.room\_id, NEW.total\_price, NOW(), NEW.check\_in\_sid);

END$$

CREATE TRIGGER after\_update\_checkin

AFTER UPDATE ON checkin

FOR EACH ROW

BEGIN

IF NEW.end\_time < CURRENT\_DATE THEN

UPDATE room

SET rstatus = '空闲'

WHERE rid = NEW.room\_id;

END IF;

END$$

DELIMITER;