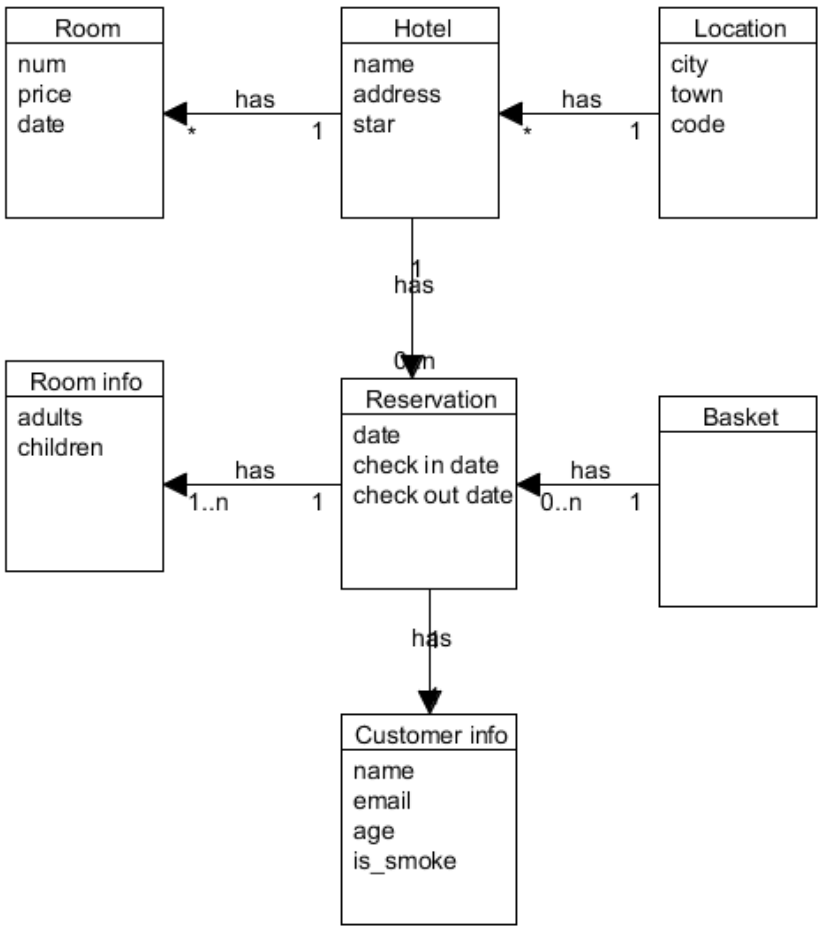
HW4

领域建模

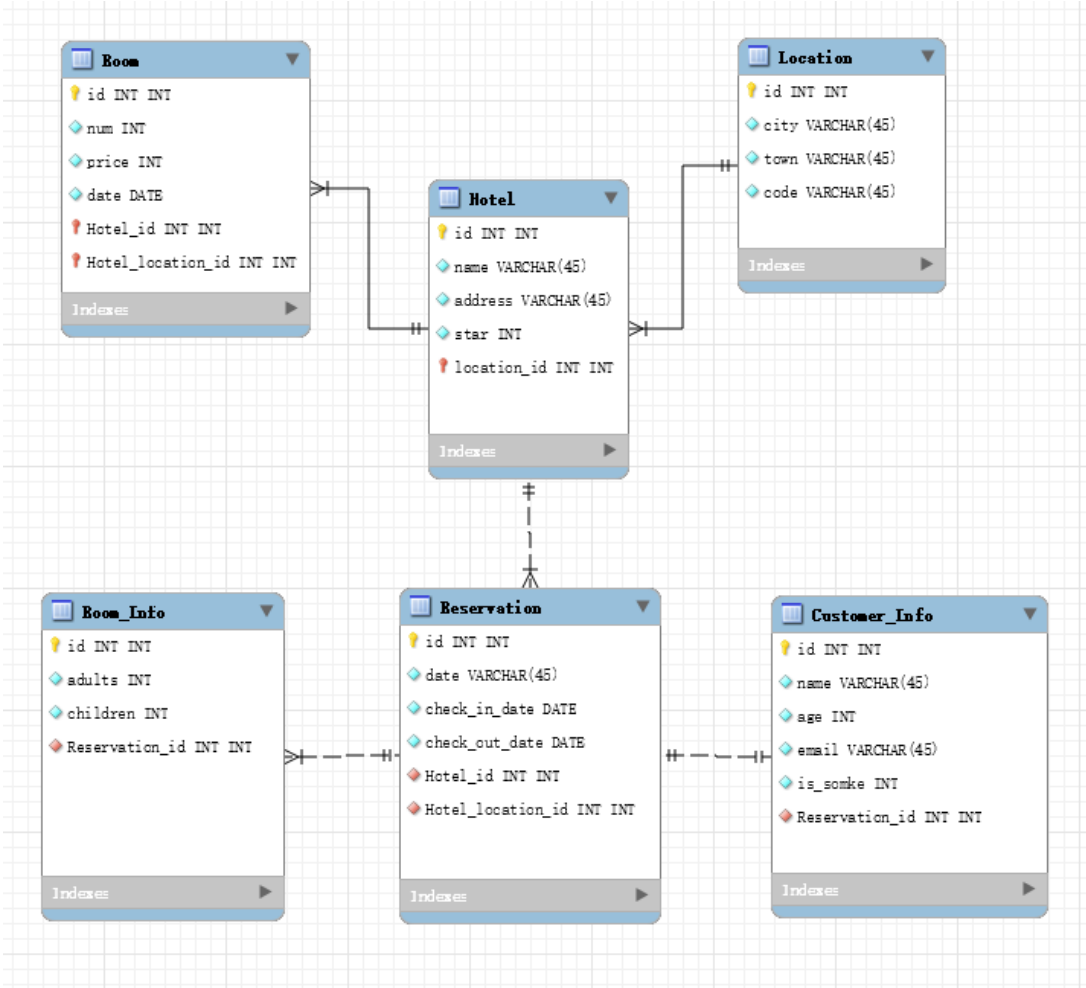
a. 阅读 Asg_RH 文档，按用例构建领域模型。

- 按 Task2 要求，请使用工具 UMLet，截图格式务必是 png 并控制尺寸
- 说明：请不要受 PCMEF 层次结构影响。你需要识别实体（E）和 中介实体（M，也称状态实体）
  - 在单页面应用（如 vue）中，E 一般与数据库构建有关， M 一般与 store 模式 有关
  - 在 java web 应用中，E 一般与数据库构建有关， M 一般与 session 有关



b. 数据库建模(E-R 模型)

- 按 Task 3 要求，给出系统的 E-R 模型（数据逻辑模型）
- 建模工具 PowerDesigner（简称PD） 或开源工具 OpenSystemArchitect
- 不负责的链接 http://www.cnblogs.com/mcgrady/archive/2013/05/25/3098588.html



– 导出 Mysql 物理数据库的脚本

```
-- MySQL Script generated by MySQL Workbench
-- Sun Apr 29 16:21:49 2018
-- Model: New Model  Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----------------------------------------------------
```

```
-- Table `mydb`.`Location`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Location` (
 `id INT` INT NOT NULL,
 `city` VARCHAR(45) NOT NULL,
 `town` VARCHAR(45) NOT NULL,
 `code` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`id INT`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`Hotel`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Hotel` (
 `id INT` INT NOT NULL,
 `name` VARCHAR(45) NOT NULL,
 `address` VARCHAR(45) NOT NULL,
 `star` INT NOT NULL,
 `location_id INT` INT NOT NULL,
 PRIMARY KEY (`id INT`, `location_id INT`),
 INDEX `location_id INT_idx` (`location_id INT` ASC),
 CONSTRAINT `location_id INT`
  FOREIGN KEY (`location_id INT`)
  REFERENCES `mydb`.`Location` (`id INT`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`Room`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Room` (
 `id INT` INT NOT NULL,
 `num` INT NOT NULL,
 `price` INT NOT NULL,
 `date` DATE NOT NULL,
 `Hotel_id INT` INT NOT NULL,
 `Hotel_location_id INT` INT NOT NULL,
 PRIMARY KEY (`id INT`, `Hotel_id INT`, `Hotel_location_id INT`),
 INDEX `fk_Room_Hotel1_idx` (`Hotel_id INT` ASC, `Hotel_location_id INT` ASC),
 CONSTRAINT `fk_Room_Hotel1`
  FOREIGN KEY (`Hotel_id INT` , `Hotel_location_id INT`)
  REFERENCES `mydb`.`Hotel` (`id INT` , `location_id INT`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `mydb`.`Reservation`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Reservation` (
 `id INT` INT NOT NULL,
 `date` VARCHAR(45) NOT NULL,
 `check_in_date` DATE NOT NULL,
 `check_out_date` DATE NOT NULL,
 `Hotel_id INT` INT NOT NULL,
 `Hotel_location_id INT` INT NOT NULL,
 PRIMARY KEY (`id INT`),
 INDEX `fk_Reservation_Hotel1_idx` (`Hotel_id INT` ASC, `Hotel_location_id INT` ASC),
 CONSTRAINT `fk_Reservation_Hotel1`
  FOREIGN KEY (`Hotel_id INT` , `Hotel_location_id INT`)
  REFERENCES `mydb`.`Hotel` (`id INT` , `location_id INT`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`Room_Info`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Room_Info` (
 `id INT` INT NOT NULL,
 `adults` INT NOT NULL,
 `children` INT NOT NULL,
 `Reservation_id INT` INT NOT NULL,
 PRIMARY KEY (`id INT`),
 INDEX `fk_Room_Info_Reservation1_idx` (`Reservation_id INT` ASC),
 CONSTRAINT `fk_Room_Info_Reservation1`
  FOREIGN KEY (`Reservation_id INT`)
  REFERENCES `mydb`.`Reservation` (`id INT`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`Customer_Info`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`Customer_Info` (
 `id INT` INT NOT NULL,
 `name` VARCHAR(45) NOT NULL,
 `age` INT NOT NULL,
 `email` VARCHAR(45) NOT NULL,
 `is_somke` INT NOT NULL,
 `Reservation_id INT` INT NOT NULL,
```

```
 PRIMARY KEY (`id INT`),
 INDEX `fk_Customer_Info_Reservation1_idx` (`Reservation_id INT` ASC),
 CONSTRAINT `fk_Customer_Info_Reservation1`
  FOREIGN KEY (`Reservation_id INT`)
  REFERENCES `mydb`.`Reservation` (`id INT`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

– 简单叙说 数据库逻辑模型与领域模型的异同

领域模型可以被看作是一个系统的概念模型，用于以可视化的形式描述系统中的各个实体及其之间的关系。数据库模型是描述数据库结构和使用的方法和技术。
领域模型是系统的结构化视图，而数据库模型是专指数据库的结构化视图。领域模型主要描述系统中各实体的关系，而数据库模型主要描述数据的表示、关系。