CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

DALHOUSIE UNIVERSITY

LAB ASSIGNMENT - 2

Submitted By: Kenil Shaileshkumar Patel (kenil.patel@dal.ca)
Banner ID: B00954251
Submitted On: September 30, 2023

Gitlab Repository Link

https://git.cs.dal.ca/kenil/csci5408_f23_b00954251_kenil_patel/-/tree/main/Lab2

Table of Contents

ERD/ EERD for Airbnb hotel system

Identifying the entities and their attributes for the Airbnb hotel system and creating the ERD diagram:

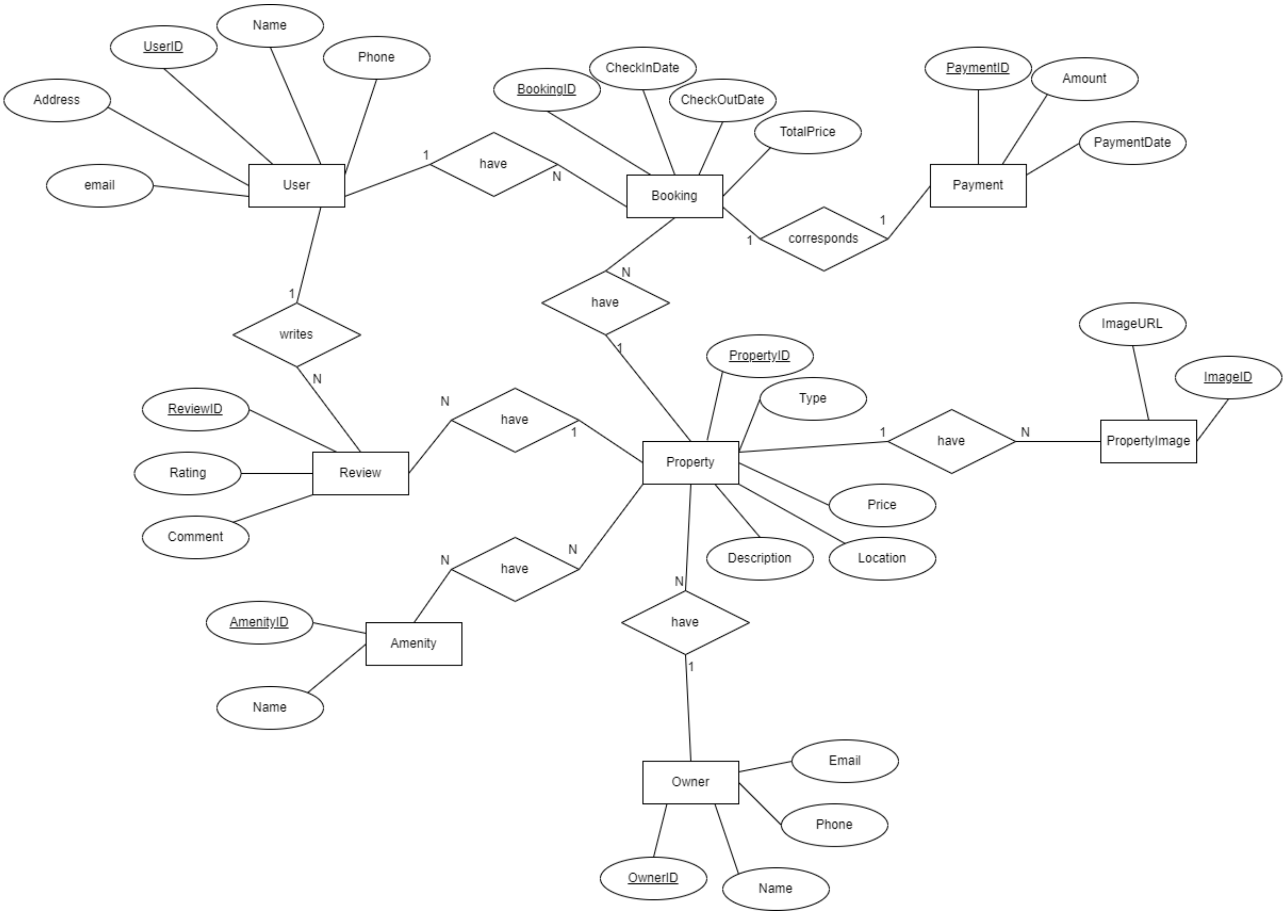


Figure 1 ERD of Airbnb hotel System

Describing the relationships between the entities:

1. User
   - One User can have many bookings i.e. One to many
   - One User can write many reviews i.e. One to many
2. Review
   - Many reviews can be for one property i.e. Many to one
   - Many reviews can be written by one user i.e. Many to one
3. Owner
   - One owner can have many properties i.e. One to many
4. Payment
   - Single payment corresponds to each booking i.e. one to one
5. Property Image
   - Many images can be of one property i.e. many to one
6. Booking
   - Many bookings can be made by one user i.e. many to one

   - Single booking has only one payment i.e. one to one
   - Many bookings can be made for one property i.e. many to one

7. Property
   - One property can have many bookings i.e. one to many
   - Many properties can be owned by a single owner i.e. many to one
   - Single property can have many reviews i.e. one to many
   - One property can have many amenities and one amenity belongs to many properties i.e. many to many
8. Amenity
   - One amenity belongs to many properties and one property can have many amenities i.e. many to many

Logical Model

Using the ERD, I have created the logical model in MySQL workbench. Moreover, I have removed the partial and transitive dependency. Below is the diagram that demonstrates the relations and cardinality between the entities. Further, it also shows the primary key and foreign keys for each table.

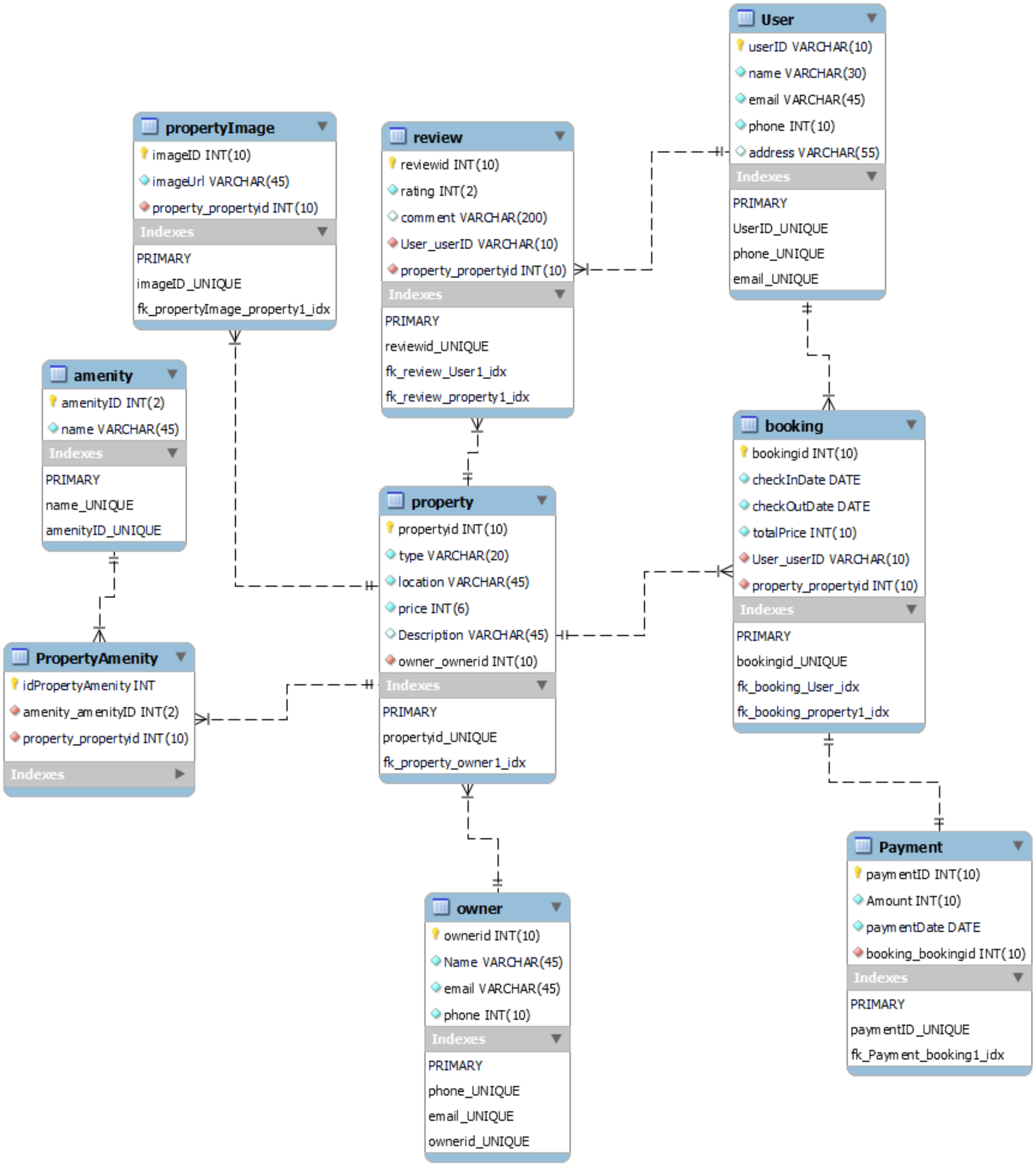
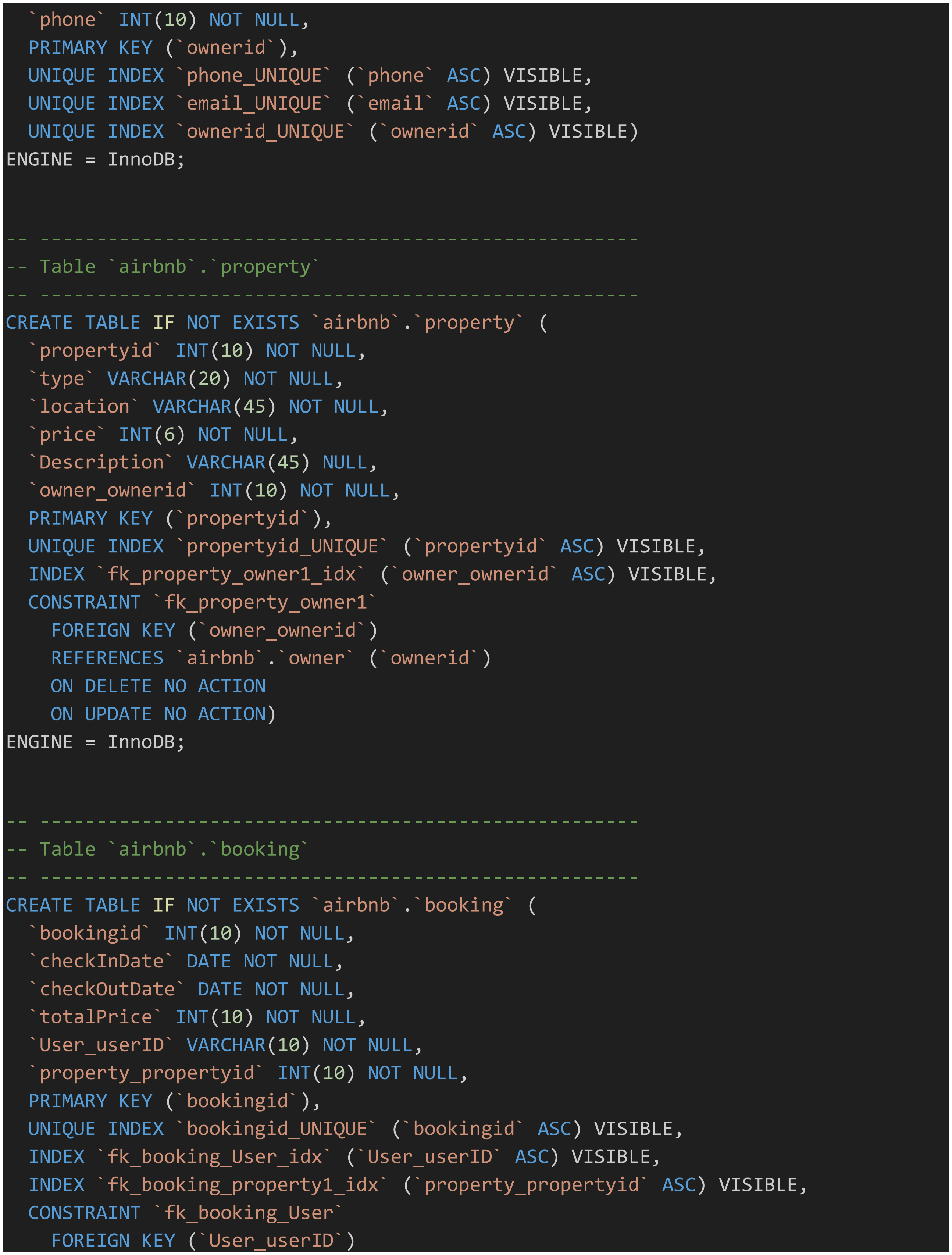


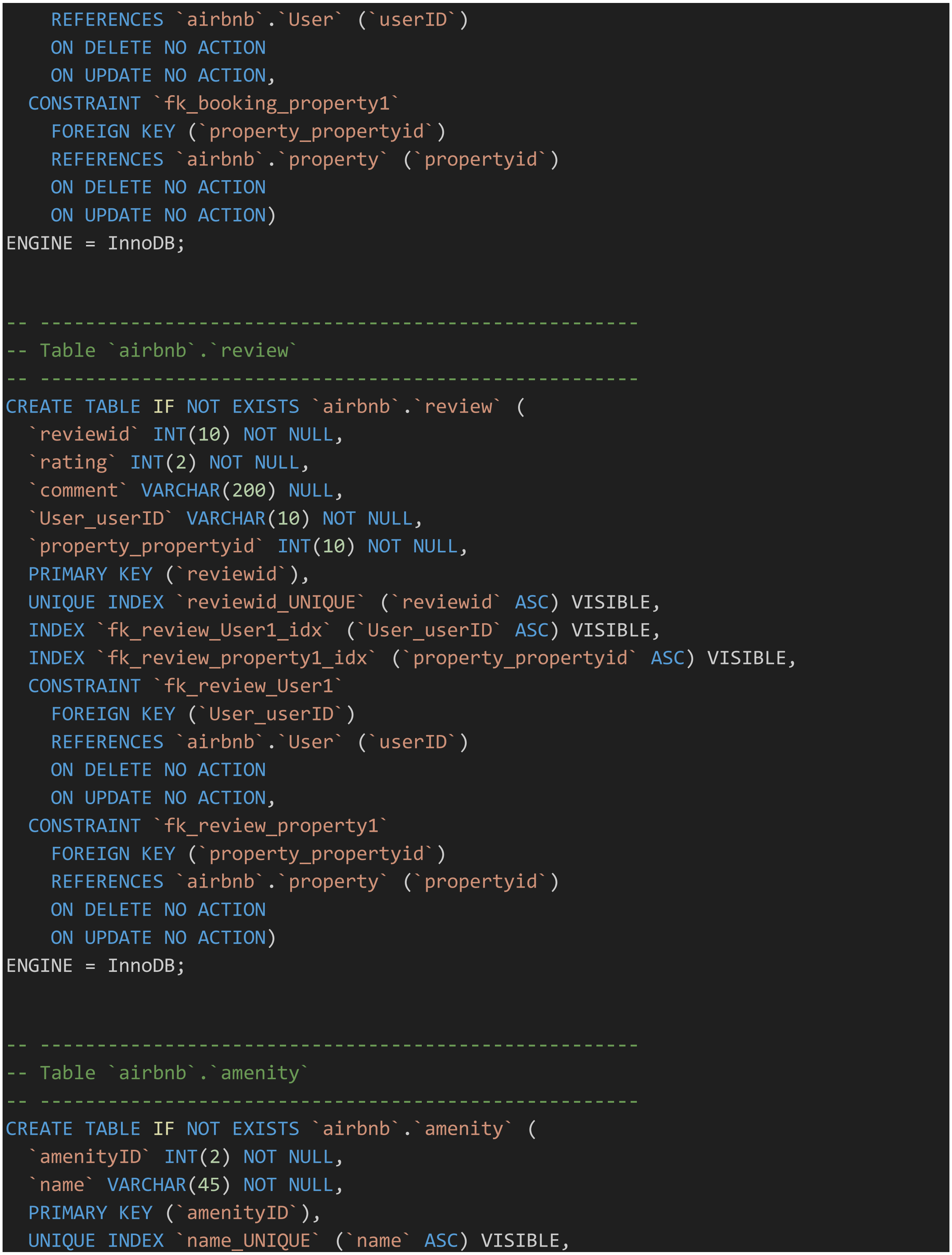
Figure 2 Logical model of Airbnb Database

SQL script generated for the table creations using forward engineering

```
-- MySQL Script generated by MySQL Workbench
-- Sat Sep 30 00:04:17 2023
-- 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='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR
_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema airbnb
-- -----------------------------------------------------

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

-- -----------------------------------------------------
-- Table `airbnb`.`User`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`User` (
  `userID` VARCHAR(10) NOT NULL,
  `name` VARCHAR(30) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone` INT(10) NOT NULL,
  `address` VARCHAR(55) NULL,
  PRIMARY KEY (`userID`),
  UNIQUE INDEX `UserID_UNIQUE` (`userID` ASC) VISIBLE,
  UNIQUE INDEX `phone_UNIQUE` (`phone` ASC) VISIBLE,
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`owner`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`owner` (
  `ownerid` INT(10) NOT NULL,
  `Name` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
```



```
  `phone` INT(10) NOT NULL,
  PRIMARY KEY (`ownerid`),
  UNIQUE INDEX `phone_UNIQUE` (`phone` ASC) VISIBLE,
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
  UNIQUE INDEX `ownerid_UNIQUE` (`ownerid` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`property`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`property` (
  `propertyid` INT(10) NOT NULL,
  `type` VARCHAR(20) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `price` INT(6) NOT NULL,
  `Description` VARCHAR(45) NULL,
  `owner_ownerid` INT(10) NOT NULL,
  PRIMARY KEY (`propertyid`),
  UNIQUE INDEX `propertyid_UNIQUE` (`propertyid` ASC) VISIBLE,
  INDEX `fk_property_owner1_idx` (`owner_ownerid` ASC) VISIBLE,
  CONSTRAINT `fk_property_owner1`
    FOREIGN KEY (`owner_ownerid`)
    REFERENCES `airbnb`.`owner` (`ownerid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`booking`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`booking` (
  `bookingid` INT(10) NOT NULL,
  `checkInDate` DATE NOT NULL,
  `checkOutDate` DATE NOT NULL,
  `totalPrice` INT(10) NOT NULL,
  `User_userID` VARCHAR(10) NOT NULL,
  `property_propertyid` INT(10) NOT NULL,
  PRIMARY KEY (`bookingid`),
  UNIQUE INDEX `bookingid_UNIQUE` (`bookingid` ASC) VISIBLE,
  INDEX `fk_booking_User_idx` (`User_userID` ASC) VISIBLE,
  INDEX `fk_booking_property1_idx` (`property_propertyid` ASC) VISIBLE,
  CONSTRAINT `fk_booking_User`
    FOREIGN KEY (`User_userID`)
```



```
    REFERENCES `airbnb`.`User` (`userID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_booking_property1`
    FOREIGN KEY (`property_propertyid`)
    REFERENCES `airbnb`.`property` (`propertyid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`review`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`review` (
  `reviewid` INT(10) NOT NULL,
  `rating` INT(2) NOT NULL,
  `comment` VARCHAR(200) NULL,
  `User_userID` VARCHAR(10) NOT NULL,
  `property_propertyid` INT(10) NOT NULL,
  PRIMARY KEY (`reviewid`),
  UNIQUE INDEX `reviewid_UNIQUE` (`reviewid` ASC) VISIBLE,
  INDEX `fk_review_User1_idx` (`User_userID` ASC) VISIBLE,
  INDEX `fk_review_property1_idx` (`property_propertyid` ASC) VISIBLE,
  CONSTRAINT `fk_review_User1`
    FOREIGN KEY (`User_userID`)
    REFERENCES `airbnb`.`User` (`userID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_review_property1`
    FOREIGN KEY (`property_propertyid`)
    REFERENCES `airbnb`.`property` (`propertyid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`amenity`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`amenity` (
  `amenityID` INT(2) NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`amenityID`),
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
  UNIQUE INDEX `amenityID_UNIQUE` (`amenityID` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`propertyImage`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`propertyImage` (
  `imageID` INT(10) NOT NULL,
  `imageUrl` VARCHAR(45) NOT NULL,
  `property_propertyid` INT(10) NOT NULL,
  PRIMARY KEY (`imageID`),
  UNIQUE INDEX `imageID_UNIQUE` (`imageID` ASC) VISIBLE,
  INDEX `fk_propertyImage_property1_idx` (`property_propertyid` ASC) VISIBLE,
  CONSTRAINT `fk_propertyImage_property1`
    FOREIGN KEY (`property_propertyid`)
    REFERENCES `airbnb`.`property` (`propertyid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`Payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`Payment` (
  `paymentID` INT(10) NOT NULL,
  `Amount` INT(10) NOT NULL,
  `paymentDate` DATE NOT NULL,
  `booking_bookingid` INT(10) NOT NULL,
  PRIMARY KEY (`paymentID`),
  UNIQUE INDEX `paymentID_UNIQUE` (`paymentID` ASC) VISIBLE,
  INDEX `fk_Payment_booking1_idx` (`booking_bookingid` ASC) VISIBLE,
  CONSTRAINT `fk_Payment_booking1`
    FOREIGN KEY (`booking_bookingid`)
    REFERENCES `airbnb`.`booking` (`bookingid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `airbnb`.`PropertyAmenity`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `airbnb`.`PropertyAmenity` (
  `idPropertyAmenity` INT NOT NULL AUTO_INCREMENT,
  `amenity_amenityID` INT(2) NOT NULL,
  `property_propertyid` INT(10) NOT NULL,
  PRIMARY KEY (`idPropertyAmenity`),
  UNIQUE INDEX `idPropertyAmenity_UNIQUE` (`idPropertyAmenity` ASC) VISIBLE,
  INDEX `fk_PropertyAmenity_amenity1_idx` (`amenity_amenityID` ASC) VISIBLE,
  INDEX `fk_PropertyAmenity_property1_idx` (`property_propertyid` ASC) VISIBLE,
  CONSTRAINT `fk_PropertyAmenity_amenity1`
    FOREIGN KEY (`amenity_amenityID`)
    REFERENCES `airbnb`.`amenity` (`amenityID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_PropertyAmenity_property1`
    FOREIGN KEY (`property_propertyid`)
    REFERENCES `airbnb`.`property` (`propertyid`)
    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;
```

References

[1] Figure 1 ERD of Airbnb Hotel System. https://app.diagrams.net/ [Accessed: 30-Sep-2023].

[2] Figure 2 Logical model of Airbnb Database. MySQL Workbench [Accessed: 30-Sep-2023].