

Group 3 Final Project Report

Create Tables and Insert Data

```
USE airbnb;
/*
1. Guest
*/
DROP TABLE IF EXISTS Guest;

CREATE TABLE Guest (
    guestID CHAR(5) NOT NULL,
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,
    email VARCHAR(60) NOT NULL,
    joinYear INT NOT NULL,
    joinMonth INT NOT NULL,
    joinDay INT NOT NULL,
    currentAvgRating DECIMAL(2,1) NOT NULL,
    PRIMARY KEY (guestID));
```

```
INSERT INTO Guest
(guestID, firstName, lastName, email, joinYear, joinMonth, joinDay, currentAvgRating)
VALUES
('G0001','Han','Lu','luhan777@qq.com',2022,1,1,4.9),
('G0002','Leon','Kennedy','leon.kennedy@rpdmail.com',2022,1,12,4.5),
('G0003','Ada','Wong','ada.wong@umbrella-inc.com',2022,2,12,4.9),
('G0004','Claire','Redfield','claire.redfield@brightmail.com',2022,3,5,4.8),
('G0005','Chris','Redfield','chris.redfield@bsaa.net',2022,4,21,4.4),
('G0006','Xinghui','Shen','shenxinghui1016@qq.com',2022,5,18,4.7),
('G0007','Shen','Li','lishen0905@qq.com',2022,6,18,4.7),
('G0008','Yu','Qi','qiyu0306@qq.com',2022,7,18,4.7),
('G0009','Che','Qin','qinche0418@qq.com',2022,10,16,4.7),
('G0010','Yizhou','Xia','xiayizhou0613@qq.com',2022,12,5,4.7);
```

```
SELECT * FROM Guest;
```

```
/*
2. GuestPhone
*/
DROP TABLE IF EXISTS GuestPhone;
```

```
CREATE TABLE GuestPhone (
    guestID CHAR(5) NOT NULL,
    phoneNumber VARCHAR(15) NOT NULL,
    PRIMARY KEY (guestID, phoneNumber),
    FOREIGN KEY (guestID) REFERENCES Guest(guestID));
```

```
INSERT INTO GuestPhone (guestID, phoneNumber) VALUES
('G0001', '8613864237810'),
('G0002', '12025549713'),
('G0003', '13106584329'),
('G0003', '19174368205'),
('G0004', '14087321658'),
('G0005', '17148450962'),
('G0006', '8615690854421'),
('G0007', '8613177269534'),
('G0007', '8613984726501'),
('G0008', '8618950483667'),
('G0008', '8615079362845'),
('G0008', '8618672594138'),
('G0009', '8614723915802'),
('G0010', '8617835640975');
```

```
SELECT * FROM GuestPhone;
```

```
/*
3. Host
*/
DROP TABLE IF EXISTS Host;
```

```
CREATE TABLE Host (
    hostID CHAR(5) NOT NULL,
    firstName VARCHAR(30) NOT NULL,
    lastName VARCHAR(30) NOT NULL,
    introduction VARCHAR(400) NOT NULL,
    email VARCHAR(100) NOT NULL,
    joinYear INT NOT NULL,
    joinMonth INT NOT NULL,
    joinDay INT NOT NULL,
    PRIMARY KEY (hostID));
```

```
INSERT INTO Host
(hostID, firstName, lastName, introduction, email, joinYear, joinMonth, joinDay)
```

VALUES

('H0001','Duffy','Bearington',
'I'm Duffy, a warmhearted traveler bear who loves sharing cozy seaside homes with friends.',
'duffy.bearington@disneyresort.com',2021,3,14),
(('H0002','ShellieMay','Honeywell',
'Hi, I'm ShellieMay — I host charming pastel cottages filled with love and handmade touches.',
'shelliemay.home@yahoo.com',2021,8,7),
(('H0003','StellaLou','Starbelle',
'Hello! I'm StellaLou, a dancer bunny hosting graceful studios with soft lights and mirrors.',
'stellalou.ballet.fr@gmail.com',2022,1,29),
(('H0004','LinaBell','Foxworth',
'I'm LinaBell, a clever fox who loves hosting curious guests in nature-inspired cabins.',
'linabell.discovery.cn@163.com',2021,9,2),
(('H0005','Cinnamoroll','Skypuff',
'I'm Cinnamoroll, your cloud-loving puppy host — come relax above the city lights.',
'cinnamoroll.sky@gmail.com',2022,1,25),
(('H0006','Pompom','Purinton',
'Welcome! I'm Pompompurin, a laid-back golden retriever who hosts sunny yellow rooms.',
'pompompurin.smile@gmail.com',2021,6,12),
(('H0007','Kuromi','Mischief',
'Hey there! I'm Kuromi — my gothic loft blends charm, chaos, and a dash of mischief.',
'kuromi.rebel@outlook.com',2021,10,30),
(('H0008','Cappy','Barrows',
'Hello! I'm Cappy Barrows, a relaxed capybara who hosts tranquil riverside cabins with natural hot baths.',
'cappy.barrows@gmail.com',2021,2,15),
(('H0009','Chinchilla','Willowfur',
'Hi, I'm Chinchilla — a soft-spoken chinchilla offering peaceful hillside stays with fluffy blankets and calm views.',
'ilovetreats@gmail.com',2017,7,11),
(('H0010','Ginny','Pigsworth',
'Welcome! I'm Ginny Pigsworth, your friendly guinea pig host with warm tea, cozy corners, and cheerful company.',
'ginny.pigsworth@outlook.com',2021,5,9);

SELECT * FROM Host;

```
/*
4. HostPhone
*/
DROP TABLE IF EXISTS HostPhone;
```

```
CREATE TABLE HostPhone (
    hostID  CHAR(5)  NOT NULL,
    phoneNumber VARCHAR(15) NOT NULL,
    PRIMARY KEY (hostID, phoneNumber),
    FOREIGN KEY (hostID) REFERENCES Host(hostID));
```

```
INSERT INTO HostPhone (hostID, phoneNumber) VALUES
('H0001', '12025548973'),
('H0001', '13107864215'),
('H0002', '14087539126'),
('H0003', '15038924761'),
('H0004', '8613894726508'),
('H0004', '8615073926481'),
('H0004', '8618685402739'),
('H0005', '819036572481'),
('H0006', '818045963217'),
('H0007', '817056428390'),
('H0008', '16028574390'),
('H0009', '17034589217'),
('H0009', '18025647318'),
('H0010','19017836425'),
('H0010','14156283907'),
('H0010','17148450692');
```

```
SELECT * FROM HostPhone;
```

```
/*
5. Listing
*/
DROP TABLE IF EXISTS Listing;
```

```
CREATE TABLE Listing (
    listingID CHAR(5) NOT NULL,
    currentAvgRating DECIMAL(3,2) NOT NULL,
    latitude DECIMAL(8,5) NOT NULL,
    longitude DECIMAL(8,5) NOT NULL,
    PRIMARY KEY (listingID));
```

```
INSERT INTO Listing
(listingID, currentAvgRating, latitude, longitude)
VALUES
('L0001', 3.56, 42.65789, -73.75370),
```

```
('L0002', 4.75, 42.65222, -73.76724),
('L0003', 4.52, 42.64615, -73.75966),
('L0004', 4.73, 42.65222, -73.76724),
('L0005', 4.79, 42.65559, -73.76506),
('L0006', 4.57, 42.64734, -73.75851),
('L0007', 4.83, 42.66078, -73.80553),
('L0008', 4.96, 42.67567, -73.82438),
('L0009', 4.81, 42.67583, -73.78297),
('L0010', 4.83, 42.65154, -73.76867);
```

```
SELECT * FROM Listing;
```

```
/*
6. Listing_location
*/
DROP TABLE IF EXISTS Listing_location;
```

```
CREATE TABLE Listing_location (
    latitude DECIMAL(8,5) NOT NULL,
    longitude DECIMAL(8,5) NOT NULL,
    zipcode CHAR(5) NOT NULL,
    city VARCHAR(40) NOT NULL,
    state VARCHAR(40) NOT NULL,
    country VARCHAR(40) NOT NULL,
    streetName VARCHAR(100) NOT NULL,
    PRIMARY KEY (latitude, longitude));
```

```
INSERT INTO Listing_location
(latitude, longitude, zipcode, city, state, country, streetName)
VALUES
(42.65789, -73.75370, '85201', 'Mesa',      'Arizona',   'USA', '45 Main St'),
(42.65328, -73.79725, '90077', 'Los Angeles', 'California', 'USA', '78 Sunset Blvd'),
(42.64615, -73.75966, '89444', 'Wellington', 'Nevada',    'USA', '101 Desert Rd'),
(42.65222, -73.76724, '79029', 'Dumas',      'Texas',     'USA', '222 Pine Ave'),
(42.65559, -73.76506, '33139', 'Miami',      'Florida',    'USA', '333 Ocean Dr'),
(42.64734, -73.75851, '81505', 'Aspen',      'Colorado',   'USA', '789 Aspen Ln'),
(42.66078, -73.80553, '85927', 'Mesa',      'Arizona',   'USA', '45 Main St'),
(42.67567, -73.82438, '90077', 'Los Angeles', 'California', 'USA', '78 Sunset Blvd'),
(42.67583, -73.78297, '89444', 'Wellington', 'Nevada',    'USA', '101 Desert Rd'),
(42.65154, -73.76867, '78361', 'Hebronville', 'Texas',     'USA', '222 Pine Ave');
```

```
SELECT * FROM Listing_location;
```

```
/*
7. AmenityCategory
*/
DROP TABLE IF EXISTS AmenityCategory;

CREATE TABLE AmenityCategory (
    AmenitycategoryID CHAR(5) NOT NULL,
    categoryName VARCHAR(50) NOT NULL,
    PRIMARY KEY (AmenitycategoryID));

INSERT INTO AmenityCategory
(AmenitycategoryID, categoryName)
VALUES
('ac001', 'Home safety'),
('ac002', 'Bedroom'),
('ac003', 'Air conditioning'),
('ac004', 'Access service'),
('ac005', 'Entertainment'),
('ac006', 'Entertainment'),
('ac007', 'Bedroom'),
('ac008', 'Workspace'),
('ac009', 'Heating & cooling'),
('ac010', 'Kitchen & dining');

SELECT * FROM AmenityCategory;

/*
8. Amenity
*/
DROP TABLE IF EXISTS Amenity;

CREATE TABLE Amenity (
    amenityID CHAR(6) NOT NULL,
    amenityName VARCHAR(80) NOT NULL,
    AmenitycategoryID CHAR(5) NOT NULL,
    PRIMARY KEY (amenityID),
    FOREIGN KEY (AmenitycategoryID) REFERENCES AmenityCategory(AmenitycategoryID));

INSERT INTO Amenity
```

```
(amenityID, amenityName, AmenitycategoryID)
VALUES
('Am0001', 'Smoke alarm', 'ac001'),
('Am0002', 'Essentials','ac002'),
('Am0003', 'Heating','ac003'),
('Am0004', 'Smart lock','ac004'),
('Am0005', 'HDTV with standard cable', 'ac005'),
('Am0006', 'Apple TV','ac006'),
('Am0007', 'Hangers', 'ac007'),
('Am0008', 'Dedicated workspace','ac008'),
('Am0009', 'Window AC unit','ac009'),
('Am0010', 'Gas stove','ac010');
```

```
SELECT * FROM Amenity;
```

```
/*
9. Attraction
*/
DROP TABLE IF EXISTS Attraction;
```

```
CREATE TABLE Attraction (
    attractionID CHAR(6) NOT NULL,
    attractionName VARCHAR(80) NOT NULL,
    streetName VARCHAR(100) NOT NULL,
    zipcode CHAR(5) NOT NULL,
    PRIMARY KEY (attractionID)
);
```

```
INSERT INTO Attraction
(attractionID, attractionName, streetName, zipcode)
VALUES
('AT0001', 'Downtown', 'Downtown Albany', '12224'),
('AT0002', 'Palace theater', '19 Clinton Ave', '12207'),
('AT0003', 'Washington Park','Washington Park Rd','12203'),
('AT0004', 'Lark Street', 'Lark St', '12208'),
('AT0005', 'Empire Plaza', '100 S Mall Arterial', '12242'),
('AT0006', 'Albany Med Offices', '43 New Scotland Ave', '12208'),
('AT0007', 'The State Capital', 'State St. and, Washington Ave','12224'),
('AT0008', 'NYS Museum', '222 Madison Ave', '12230'),
('AT0009', 'Governor"s Mansion','138 Eagle St', '12202');
```

```
SELECT * FROM Attraction;
```

```
/*
10. Booking_detail
*/
DROP TABLE IF EXISTS Booking_detail;
```

```
CREATE TABLE Booking_detail (
    bookingID CHAR(5) NOT NULL,
    totalPrice INT NOT NULL,
    checkinYear INT NOT NULL,
    checkinMonth INT NOT NULL,
    checkinDay INT NOT NULL,
    checkinHour INT NOT NULL,
    checkoutYear INT NOT NULL,
    checkoutMonth INT NOT NULL,
    checkoutDay INT NOT NULL,
    checkoutHour INT NOT NULL,
    PRIMARY KEY (bookingID));
```

```
INSERT INTO Booking_detail
(bookingID, totalPrice,
checkinYear, checkinMonth, checkinDay, checkinHour,
checkoutYear, checkoutMonth, checkoutDay, checkoutHour)
VALUES
('B0001', 120, 2022, 1, 23, 5, 2022, 1, 30, 13),
('B0002', 135, 2022, 1, 27, 18, 2022, 2, 21, 4),
('B0003', 150, 2022, 3, 1, 10, 2022, 3, 14, 18),
('B0004', 165, 2022, 3, 28, 21, 2022, 4, 15, 9),
('B0005', 185, 2022, 5, 25, 7, 2022, 5, 29, 22),
('B0006', 210, 2022, 6, 29, 14, 2022, 7, 22, 22),
('B0007', 230, 2022, 7, 30, 3, 2022, 8, 8, 15),
('B0008', 240, 2022, 8, 26, 12, 2022, 9, 22, 1),
('B0009', 160, 2022, 11, 22, 19, 2022, 12, 8, 11),
('B0010', 275, 2022, 12, 28, 9, 2023, 1, 2, 20);
```

```
SELECT * FROM Booking_detail;
```

```
/*
11. Guest_Concern_Report
*/
```

```
DROP TABLE IF EXISTS Guest_Concern_Report;
```

```
CREATE TABLE Guest_Concern_Report (
    reportID CHAR(6) NOT NULL,
    guestID CHAR(5) NOT NULL,
    reportReason VARCHAR(100) NOT NULL,
    country VARCHAR(40) NOT NULL,
    state VARCHAR(40) NOT NULL,
    streetName VARCHAR(100) NOT NULL,
    zipCode CHAR(5) NOT NULL,
    year SMALLINT NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    hour INT NOT NULL,
    minute INT NOT NULL,
    PRIMARY KEY (reportID),
    FOREIGN KEY (guestID) REFERENCES Guest(guestID));
```

```
INSERT INTO GUEST_CONCERN_REPORT
(guestID, reportID, reportReason, country, state, streetName, zipCode, year, month, day, hour,
minute)
VALUES
('G0001','GR0001','Noise complaint','USA','Arizona','45 Main St','85721',2022,1,25,8,33),
('G0002','GR0002','Cleaning issue','USA','California','78 Sunset Blvd','90001',2022,2,3,9,15),
('G0003','GR0003','Lake check-in','USA','Nevada','101 Desert Rd','89109',2022,2,15,22,48),
('G0004','GR0004','Broken air conditioner','USA','Texas','222 Pine Ave','73301',2022,4,2,14,20),
('G0005','GR0005','Wi-fi not working','USA','Florida','333 Ocean Dr','33139',2022,5,27,13,18),
('G0006','GR0006','No hot water','USA','Colorado','789 Aspen Ln','80014',2022,7,9,2,41),
('G0001','GR0007','Noise from neighbors','USA','Arizona','45 Main St','85721',2022,1,28,19,12),
('G0002','GR0008','Host not responsive','USA','California','78 Sunset Blvd','90001',2022,2,15, 22,
48),
('G0003','GR0009','Cleaning issue','USA','Nevada','101 Desert Rd','89109',2022,3,11,16,29),
('G0004','GR0010','Bad smell','USA','Texas','222 Pine Ave','73301',2022,4,11,6,57);
```

```
SELECT * FROM Guest_Concern_Report;
```

```
/*
12. Host_Concern_Report
*/
DROP TABLE IF EXISTS Host_Concern_Report;
```

```
CREATE TABLE Host_Concern_Report (
```

```

reportID CHAR(6) NOT NULL,
hostID CHAR(5) NOT NULL,
reportReason VARCHAR(100) NOT NULL,
country VARCHAR(40) NOT NULL,
state VARCHAR(40) NOT NULL,
streetName VARCHAR(100) NOT NULL,
zipCode CHAR(5) NOT NULL,
year INT NOT NULL,
month INT NOT NULL,
day INT NOT NULL,
hour INT NOT NULL,
minute INT NOT NULL,
PRIMARY KEY (reportID),
FOREIGN KEY (hostID) REFERENCES Host(hostID));

```

```

INSERT INTO Host_Concern_Report
(hostID, reportID, reportReason, country, state, streetName, zipCode,
year, month, day, hour, minute)
VALUES
('H0001','HR0001','Guest damaged property','USA','Arizona','12 Hillcrest
Ave','85721',2022,2,2,4,51),
('H0001','HR0002','Guest left late','USA','Arizona','12 Hillcrest Ave','85721', 2022,2,4,16,39),
('H0002','HR0003','Excessive noise from guests','USA','California','245 Ocean
Blvd','90001',2022,2,25,22,5),
('H0003','HR0004','Smoking in property','USA','Nevada','18 Desert Dr','89109', 2022,3,19,8,42),
('H0003','HR0005','Trash not disposed','USA','Nevada','18 Desert Dr','89109',2022,3,20,15,33),
('H0004','HR0006','Unauthorized pet','USA','Texas','451 Oak St','73301', 2022,4,17,2,28),
('H0005','HR0007','Broken furniture','USA','Florida','789 Palm Rd','33139', 2022,6,2,10,41),
('H0005','HR0008','Stolen towels','USA','Florida','789 Palm Rd','33139', 2022,6,4,5,26),
('H0002','HR0009','Key not returned','USA','California','245 Ocean Blvd','90001',
2022,2,27,11,14),
('H0004','HR0010','Water leak reported late','USA','Texas','451 Oak St','73301',
2022,4,21,23,57);

```

```

SELECT * FROM Host_Concern_Report;

```

```

/*
13. Booking_Aircover
*/
DROP TABLE IF EXISTS Booking_Aircover;

```

```

CREATE TABLE Booking_Aircover (

```

```

bookingID CHAR(5) NOT NULL,
policyNumber CHAR(5) NOT NULL,
policyStartYear INT NOT NULL,
policyStartMonth INT NOT NULL,
policyStartDay INT NOT NULL,
policyStartHour INT NOT NULL,
policyStartMinute INT NOT NULL,
policyEndYear INT NOT NULL,
policyEndMonth INT NOT NULL,
policyEndDay INT NOT NULL,
policyEndHour INT NOT NULL,
policyEndMinute INT NOT NULL,
PRIMARY KEY (bookingID, policyNumber),
FOREIGN KEY (bookingID) REFERENCES Booking_detail(bookingID));

```

```

INSERT INTO Booking_Aircover
(bookingID, policyNumber, policyStartYear, policyStartMonth, policyStartDay, policyStartHour,
policyStartMinute,
policyEndYear, policyEndMonth, policyEndDay, policyEndHour , policyEndMinute)
VALUES
('B0001','P0001',2022,1,23,5,14, 2022,1,30,13,22),
('B0002','P0002',2022,1,27,18,42, 2022,2,21,4,57),
('B0003','P0003',2022,3,1,10,3, 2022,3,14,18,11),
('B0004','P0004',2022,3,28,21,56, 2022,4,15,9,36),
('B0005','P0005',2022,5,25,7,40, 2022,5,29,22,3),
('B0006','P0006',2022,6,29,14,27, 2022,7,22,6,49),
('B0007','P0007',2022,7,30,3,51, 2022,8,8,15,27),
('B0008','P0008',2022,8,26,12,9, 2022,9,22,1,54),
('B0009','P0009',2022,11,22,19,33,2022,12,8,11,5),
('B0010','P0010',2022,12,28,9,48, 2023,1,2,20,42);

```

```
SELECT * FROM Booking_Aircover;
```

```

/*
14. Listing_Aircover
*/
DROP TABLE IF EXISTS Listing_Aircover;

CREATE TABLE Listing_Aircover (
listingID CHAR(5) NOT NULL,
policyNumber CHAR(5) NOT NULL,
macStart DATE NULL,

```

```

macEnd DATE NULL,
policyStartYear INT NOT NULL,
policyStartMonth INT NOT NULL,
policyStartDay INT NOT NULL,
policyEndYear INT NOT NULL,
policyEndMonth INT NOT NULL,
policyEndDay INT NOT NULL,
PRIMARY KEY (listingID, policyNumber),
FOREIGN KEY (listingID) REFERENCES Listing(listingID));

```

```

INSERT INTO Listing_Aircover
(listingID, policyNumber, macStart, macEnd,
policyStartYear, policyStartMonth, policyStartDay,
policyEndYear, policyEndMonth, policyEndDay)
VALUES
('L0001','P0001','2021-03-14','2031-03-14',2021,3,14,2031,3,14),
('L0002','P0002','2021-08-07','2031-08-07',2021,8,7,2031,8,7),
('L0003','P0003','2022-01-29','2032-01-29',2022,1,29,2032,1,29),
('L0004','P0004','2021-09-02','2031-09-02',2021,9,2,2031,9,2),
('L0005','P0005','2022-01-25','2032-01-25',2022,1,25,2032,1,25),
('L0006','P0006','2021-06-12','2031-06-12',2021,6,12,2031,6,12),
('L0007','P0007','2021-10-30','2031-10-30',2021,10,30,2031,10,30),
('L0008','P0008','2021-02-15','2031-02-15',2021,2,15,2031,2,15),
('L0009','P0009','2021-07-11','2031-07-11',2021,7,11,2031,7,11),
('L0010','P0010','2021-05-09','2031-05-09',2021,5,9,2031,5,9);

```

```
SELECT * FROM Listing_Aircover;
```

```

/*
15. Review
*/
DROP TABLE IF EXISTS Review;

```

```

CREATE TABLE Review (
bookingID CHAR(5) NOT NULL,
forGuestComment VARCHAR(200) NOT NULL,
forGuestRating INT NOT NULL,
forListingComment VARCHAR(200) NOT NULL,
forListingRating INT NOT NULL,
PRIMARY KEY (bookingID),
FOREIGN KEY (bookingID) REFERENCES Booking_detail(bookingID)
);

```

```
INSERT INTO Review
(bookingID, forGuestComment, forGuestRating, forListingComment, forListingRating)
VALUES
('B0001','respectful, checked in and out on time',5,'Great location, thanks!',5),
('B0002','did not respond quickly',4,'This made my birthday',5),
('B0003','checked out late',3,'Thanks for going above and beyond and helping with the garage',5),
('B0004','Great guest, even took off bed sheets and left dishes washed',5,'Bedbugs!!',1),
('B0005','Left food in fridge, but everything else was ok',4,'Great spot to stay, center of town',5),
('B0006','So happy with these guests. Please come again!',5,'This was so convenient. thanks for communicating',5),
('B0007','Let me know your eta next time',4,'Hard to find the gate!',4),
('B0008','Good',5,'Will come again!',5),
('B0009','Neighbors got upset because music was too loud',3,'Had the best time.',5),
('B0010','Come again!',5,'Wifi didnt work!',3);
```

```
SELECT * FROM Review;
```

```
/*
16. Payment
*/
DROP TABLE IF EXISTS Payment;
```

```
CREATE TABLE Payment (
paymentID CHAR(5) NOT NULL,
bookingID CHAR(5) NOT NULL,
days INT NOT NULL,
totalPaymentUSD DECIMAL(10,2) NOT NULL,
paymentMethod VARCHAR(20) NOT NULL,
macDate DATETIME NOT NULL,
year INT NOT NULL,month INT NOT NULL,
Day INT NOT NULL,
hour INT NOT NULL,
minute INT NOT NULL,
PRIMARY KEY (paymentID),
FOREIGN KEY (bookingID) REFERENCES Booking_detail(bookingID));
```

```
INSERT INTO Payment
(paymentID, bookingID, days, totalPaymentUSD, paymentMethod,macDate, year, month, day,
hour, minute)
VALUES
```

```
('C0001','B0001',7, 350.43,'visa',  '2022-01-11 14:33:00',2018,1, 9,13,45),
('C0002','B0002',25,2310.21,'master', '2022-01-19 03:22:00',2020,1,10,18, 2),
('C0003','B0003',13, 450.89,'discover','2022-02-21 16:09:00',2021,2, 4,12,34),
('C0004','B0004',18,1090.90,'visa',   '2022-03-15 09:48:00',2022,3,13,21,26),
('C0005','B0005',4, 276.67,'master', '2022-05-07 18:12:00',2023,1,10,17,49),
('C0006','B0006',22,1987.20,'master', '2022-06-05 07:26:00',2024,4, 9,20,15),
('C0007','B0007',9, 681.90,'visa',   '2022-07-07 22:41:00',2025,7,12,23,36),
('C0008','B0008',27,2790.67,'discover','2022-08-12 11:03:00',2025,5,15,13,21),
('C0009','B0009',16,1110.12,'visa',   '2022-11-05 04:57:00',2025,8,17,16, 1),
('C0010','B0010',5, 678.12,'visa',   '2022-12-17 20:29:00',2025,10,2,21,14);
```

```
SELECT * FROM Payment;
```

```
/*
```

```
17. Host_Has_Listing
```

```
*/
```

```
DROP TABLE IF EXISTS Host_Has_Listing;
```

```
CREATE TABLE Host_Has_Listing (
    hostID CHAR(5) NOT NULL,
    listingID CHAR(5) NOT NULL,
    PRIMARY KEY (hostID, listingID),
    FOREIGN KEY (hostID) REFERENCES Host(hostID),
    FOREIGN KEY (listingID) REFERENCES Listing(listingID));
```

```
INSERT INTO Host_Has_Listing (hostID, listingID) VALUES
('H0001', 'L0001'),
('H0001', 'L0002'),
('H0002', 'L0003'),
('H0003', 'L0004'),
('H0003', 'L0005'),
('H0004', 'L0006'),
('H0005', 'L0007'),
('H0005', 'L0008'),
('H0002', 'L0009'),
('H0004', 'L0010');
```

```
SELECT * FROM Host_Has_Listing;
```

```
/*
```

```
18. Listing_Has_Amenity
*/
DROP TABLE IF EXISTS Listing_Has_Amenity;

CREATE TABLE Listing_Has_Amenity (
    listingID CHAR(5) NOT NULL,
    amenityID CHAR(6) NOT NULL,
    PRIMARY KEY (listingID, amenityID),
    FOREIGN KEY (listingID) REFERENCES Listing(listingID),
    FOREIGN KEY (amenityID) REFERENCES Amenity(amenityID)
);

INSERT INTO Listing_Has_Amenity (listingID, amenityID) VALUES
('L0001','Am0001'),
('L0002','Am0002'),
('L0003','Am0003'),
('L0004','Am0001'),
('L0005','Am0004'),
('L0006','Am0002'),
('L0007','Am0005'),
('L0008','Am0001'),
('L0009','Am0006'),
('L0010','Am0005');

SELECT * FROM Listing_Has_Amenity;
```

```
/*
19. Listing_Is_Near_To_Attraction
*/
DROP TABLE IF EXISTS Listing_Is_Near_To_Attraction;

CREATE TABLE Listing_Is_Near_To_Attraction (
    listingID  CHAR(5) NOT NULL,
    attractionID CHAR(6) NOT NULL,
    PRIMARY KEY (listingID, attractionID),
    FOREIGN KEY (listingID)  REFERENCES Listing(listingID),
    FOREIGN KEY (attractionID) REFERENCES Attraction(attractionID));

INSERT INTO Listing_Is_Near_To_Attraction (listingID, attractionID) VALUES
('L0001','AT0001'),
('L0002','AT0002'),
('L0003','AT0003');
```

```
('L0004','AT0002'),
('L0005','AT0004'),
('L0006','AT0005'),
('L0007','AT0003'),
('L0008','AT0001'),
('L0009','AT0001'),
('L0010','AT0005');
```

```
SELECT * FROM Listing_Is_Near_To_Attraction;
```

```
/*
20. Booking
*/
DROP TABLE IF EXISTS Booking;
```

```
CREATE TABLE Booking (
    bookingID CHAR(5) NOT NULL,
    guestID CHAR(5) NOT NULL,
    listingID CHAR(5) NOT NULL,
    PRIMARY KEY (bookingID),
    FOREIGN KEY (bookingID) REFERENCES Booking_detail(bookingID),
    FOREIGN KEY (guestID) REFERENCES Guest(guestID),
    FOREIGN KEY (listingID) REFERENCES Listing(listingID));
```

```
INSERT INTO Booking (guestID, listingID, bookingID) VALUES
('G0001','L0001','B0001'),
('G0002','L0002','B0002'),
('G0003','L0003','B0003'),
('G0004','L0004','B0004'),
('G0005','L0005','B0005'),
('G0006','L0006','B0006'),
('G0007','L0007','B0007'),
('G0008','L0008','B0008'),
('G0009','L0009','B0009'),
('G0010','L0010','B0010');
```

```
SELECT * FROM Booking;
```

Retrieve Data using SQL Queries with Business Situations

```
/*
```

1. Retrieve the data from each table by using the SELECT * statement and order by PK column(s).

Show the output. Make sure you show the print screen of the complete set of rows and columns.

The rows must be ordered by PK column(s).

```
*/
```

```
-- 1. Guest
```

```
SELECT *
```

```
FROM Guest
```

```
ORDER BY guestID;
```

A screenshot of a database query results grid. The grid has a header row with columns: guestID, firstName, lastName, email, joinYear, joinMonth, joinDay, and currentAvgRating. Below the header are 10 data rows, each representing a guest with their ID, name, email, and joining details. The data is ordered by guestID. The right side of the interface includes a toolbar with icons for Result Grid, Form Editor, Field Types, and Query Stats.

guestID	firstName	lastName	email	joinYear	joinMonth	joinDay	currentAvgRating
G0001	Han	Lu	luhan777@qq.com	2022	1	1	4.9
G0002	Leon	Kennedy	leon.kennedy@rpdmail.com	2022	1	12	4.5
G0003	Ada	Wong	ada.wong@umbrella-inc.com	2022	2	12	4.9
G0004	Claire	Redfield	claire.redfield@brightmail.com	2022	3	5	4.8
G0005	Chris	Redfield	chris.redfield@bsaa.net	2022	4	21	4.4
G0006	Xinghui	Shen	shenxinghui016@qq.com	2022	5	18	4.7
G0007	Shen	Li	lishen0905@qq.com	2022	6	18	4.7
G0008	Yu	Qi	qiyu0306@qq.com	2022	7	18	4.7
G0009	Che	Qin	qincheng0418@qq.com	2022	10	16	4.7
G0010	Yizhou	Xia	xiaiyizhou0613@qq.com	2022	12	5	4.7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
-- 2. GuestPhone
```

```
SELECT *
```

```
FROM GuestPhone
```

```
ORDER BY guestID, phoneNumber;
```

A screenshot of a database query results grid. The grid has a header row with columns: guestID and phoneNumber. Below the header are 10 data rows, each representing a guest with their ID and their corresponding phone number. The data is ordered by guestID. The right side of the interface includes a toolbar with icons for Result Grid, Form Editor, Field Types, and Query Stats.

guestID	phoneNumber
G0001	8613864237810
G0002	12025549713
G0003	13106584329
G0004	19174368205
G0005	14087321658
G0006	17148450962
G0007	8615690854421
G0008	8613177269534
G0009	8613984726501
G0010	8615079362845
NULL	NULL

```
-- 3. Host
```

```
SELECT *
```

```
FROM Host
```

```
ORDER BY hostID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

hostID	firstName	lastName	introduction	email	joinYear	joinMonth	joinDay
H0001	Duffy	Bearington	I'm Duffy, a warmhearted traveler bear who loves sharing cozy seaside homes with f...	duffy.bearington@disneyresort.com	2021	3	14
H0002	ShellieMay	Honeywell	Hi, I'm ShellieMay — I host charming pastel cottages filled with love and handmade...	shelliemay.home@yahoo.com	2021	8	7
H0003	StellaLou	Starbelle	Hello! I'm StellaLou, a dancer bunny hosting graceful studios with soft lights and mir...	stellalou.ballet.fr@gmail.com	2022	1	29
H0004	LinaBell	Foxworth	I'm LinaBell, a clever fox who loves hosting curious guests in nature-inspired cabins.	linabell.discovery.cn@163.com	2021	9	2
H0005	Cinnamoroll	Skypuff	I'm Cinnamoroll, your cloud-loving puppy host — come relax above the city lights.	cinnamoroll.sky@gmail.com	2022	1	25
H0006	Pompom	Purinton	Welcome! I'm Pompompurin, a laid-back golden retriever who hosts sunny yellow ro...	pompompurin.smile@gmail.com	2021	6	12
H0007	Kuromi	Mischief	Hey there! I'm Kuromi — my gothic loft blends charm, chaos, and a dash of mischief.	kuromi.rebel@outlook.com	2021	10	30
H0008	Cappy	Barrows	Hello! I'm Cappy Barrows, a relaxed capybara who hosts tranquil riverside cabins w...	cappy.barrows@gmail.com	2021	2	15
H0009	Chinchilla	Willowfur	Hi, I'm Chinchilla — a soft-spoken chinchilla offering peaceful hillside stays with fluff...	ilovetreats@gmail.com	2017	7	11
H0010	Ginny	Pigsworth	Welcome! I'm Ginny Pigsworth, your friendly guinea pig host with warm tea, cozy co...	ginny.pigsworth@outlook.com	2021	5	9
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 4. HostPhone

```
SELECT *
```

```
FROM HostPhone
```

```
ORDER BY hostID, phoneNumber;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

hostID	phoneNumber
H0001	12025548973
H0001	13107864215
H0002	14087539126
H0003	15038924761
H0004	8613894726508
H0004	8615073926481
H0004	8618685402739
H0005	819036572481
H0006	818045963217
H0007	817056428390
H0008	16028574390
H0009	17034589217
H0009	18025647318
H0010	14156283907
H0010	17148450692
H0010	19017836425
HULL	HULL

-- 5. Listing

```
SELECT *
```

```
FROM Listing
```

```
ORDER BY listingID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

listingID	currentAvgRating	latitude	longitude
L0001	3.56	42.65789	-73.75370
L0002	4.75	42.65222	-73.776724
L0003	4.52	42.64615	-73.75966
L0004	4.73	42.65222	-73.76724
L0005	4.79	42.65559	-73.76506
L0006	4.57	42.64734	-73.75851
L0007	4.83	42.66078	-73.80553
L0008	4.96	42.67567	-73.82438
L0009	4.81	42.67583	-73.78297
L0010	4.83	42.65154	-73.76867
HULL	HULL	HULL	HULL

-- 6. Listing_location

```
SELECT *
```

```
FROM Listing_location
```

```
ORDER BY latitude, longitude;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

latitude	longitude	zipcode	city	state	country	streetName
42.64615	-73.75966	89444	Wellington	Nevada	USA	101 Desert Rd
42.64734	-73.75851	81505	Aspen	Colorado	USA	789 Aspen Ln
42.65154	-73.76867	78361	Hebronville	Texas	USA	222 Pine Ave
42.65222	-73.76724	79029	Dumas	Texas	USA	222 Pine Ave
42.65328	-73.79725	90077	Los Angeles	California	USA	78 Sunset Blvd
42.65559	-73.76506	33139	Miami	Florida	USA	333 Ocean Dr
42.65789	-73.75371	85201	Mesa	Arizona	USA	45 Main St
42.66078	-73.80553	85927	Mesa	Arizona	USA	45 Main St
42.67567	-73.82438	90077	Los Angeles	California	USA	78 Sunset Blvd
42.67583	-73.78297	89444	Wellington	Nevada	USA	101 Desert Rd
HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 7. AmenityCategory

```
SELECT *
FROM AmenityCategory
ORDER BY AmenitycategoryID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

AmenitycategoryID	categoryName
ac001	Home safety
ac002	Bedroom
ac003	Air conditioning
ac004	Access service
ac005	Entertainment
ac006	Entertainment
ac007	Bedroom
ac008	Workspace
ac009	Heating & cooling
ac010	Kitchen & dining
HULL	HULL

-- 8. Amenity

```
SELECT *
FROM Amenity
ORDER BY amenityID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

amenityID	amenityName	AmenitycategoryID
Am0001	Smoke alarm	ac001
Am0002	Essentials	ac002
Am0003	Heating	ac003
Am0004	Smart lock	ac004
Am0005	HDTV with standard cable	ac005
Am0006	Apple TV	ac006
Am0007	Hangers	ac007
Am0008	Dedicated workspace	ac008
Am0009	Window AC unit	ac009
Am0010	Gas stove	ac010
HULL	HULL	HULL

-- 9. Attraction

```
SELECT *
FROM Attraction
ORDER BY attractionID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

attractionID	attractionName	streetName	zipcode
AT0001	Downtown	Downtown Albany	12224
AT0002	Palace theater	19 Clinton Ave	12207
AT0003	Washington Park	Washington Park Rd	12203
AT0004	Lark Street	Lark St	12208
AT0005	Empire Plaza	100 S Mall Arterial	12242
AT0006	Albany Med Offices	43 New Scotland Ave	12208
AT0007	The State Capital	State St. and, Washington Ave	12224
AT0008	NYS Museum	222 Madison Ave	12230
AT0009	Governor's Mansion	138 Eagle St	12202
HULL	HULL	HULL	HULL

-- 10. Booking_detail

```
SELECT *
```

```
FROM Booking_detail
```

```
ORDER BY bookingID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

bookingID	totalPrice	checkinYear	checkinMonth	checkinDay	checkinHour	checkoutYear	checkoutMonth	checkoutDay	checkoutHour
B0001	120	2022	1	23	5	2022	1	30	13
B0002	135	2022	1	27	18	2022	2	21	4
B0003	150	2022	3	1	10	2022	3	14	18
B0004	165	2022	3	28	21	2022	4	15	9
B0005	185	2022	5	25	7	2022	5	29	22
B0006	210	2022	6	29	14	2022	7	22	22
B0007	230	2022	7	30	3	2022	8	8	15
B0008	240	2022	8	26	12	2022	9	22	1
B0009	160	2022	11	22	19	2022	12	8	11
B0010	275	2022	12	28	9	2023	1	2	20
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 11. Guest_Concern_Report

```
SELECT *
```

```
FROM Guest_Concern_Report
```

```
ORDER BY reportID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

reportID	guestID	reportReason	country	state	streetName	zipCode	year	month	day	hour	minute
GR0001	G0001	Noise complaint	USA	Arizona	45 Main St	85721	2022	1	25	8	33
GR0002	G0002	Cleaning issue	USA	California	78 Sunset Blvd	90001	2022	2	3	9	15
GR0003	G0003	Lake check-in	USA	Nevada	101 Desert Rd	89109	2022	2	15	22	48
GR0004	G0004	Broken air conditioner	USA	Texas	222 Pine Ave	73301	2022	4	2	14	20
GR0005	G0005	Wi-fi not working	USA	Florida	333 Ocean Dr	33139	2022	5	27	13	18
GR0006	G0006	No hot water	USA	Colorado	789 Aspen Ln	80014	2022	7	9	2	41
GR0007	G0001	Noise from neighbors	USA	Arizona	45 Main St	85721	2022	1	28	19	12
GR0008	G0002	Host not responsive	USA	California	78 Sunset Blvd	90001	2022	2	15	22	48
GR0009	G0003	Cleaning issue	USA	Nevada	101 Desert Rd	89109	2022	3	11	16	29
GR0010	G0004	Bad smell	USA	Texas	222 Pine Ave	73301	2022	4	11	6	57
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 12. Host_Concern_Report

```
SELECT *
```

```
FROM Host_Concern_Report
```

```
ORDER BY reportID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

Result Grid

reportID	hostID	reportReason	country	state	streetName	zipCode	year	month	day	hour	minute
HR0001	H0001	Guest damaged property	USA	Arizona	12 Hillcrest Ave	85721	2022	2	2	4	51
HR0002	H0001	Guest left late	USA	Arizona	12 Hillcrest Ave	85721	2022	2	4	16	39
HR0003	H0002	Excessive noise from guests	USA	California	245 Ocean Blvd	90001	2022	2	25	22	5
HR0004	H0003	Smoking in property	USA	Nevada	18 Desert Dr	89109	2022	3	19	8	42
HR0005	H0003	Trash not disposed	USA	Nevada	18 Desert Dr	89109	2022	3	20	15	33
HR0006	H0004	Unauthorized pet	USA	Texas	451 Oak St	73301	2022	4	17	2	28
HR0007	H0005	Broken furniture	USA	Florida	789 Palm Rd	33139	2022	6	2	10	41
HR0008	H0005	Stolen towels	USA	Florida	789 Palm Rd	33139	2022	6	4	5	26
HR0009	H0002	Key not returned	USA	California	245 Ocean Blvd	90001	2022	2	27	11	14
HR0010	H0004	Water leak reported late	USA	Texas	451 Oak St	73301	2022	4	21	23	57
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 13. Booking_Aircover

```
SELECT *
FROM Booking_Aircover
ORDER BY bookingID, policyNumber;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

bookingID	policyNumber	policyStartYear	policyStartMonth	policyStartDay	policyStartHour	policyStartMinute	policyEndYear	policyEndMonth	policyEndDay	policyEndHour	policyEndMinute
B0001	P0001	2022	1	23	5	14	2022	1	30	13	22
B0002	P0002	2022	1	27	18	42	2022	2	21	4	57
B0003	P0003	2022	3	1	10	3	2022	3	14	18	11
B0004	P0004	2022	3	28	21	56	2022	4	15	9	36
B0005	P0005	2022	5	25	7	40	2022	5	29	22	3
B0006	P0006	2022	6	29	14	27	2022	7	22	6	49
B0007	P0007	2022	7	30	3	51	2022	8	8	15	27
B0008	P0008	2022	8	26	12	9	2022	9	22	1	54
B0009	P0009	2022	11	22	19	33	2022	12	8	11	5
B0010	P0010	2022	12	28	9	48	2023	1	2	20	42
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 14. Listing_Aircover

```
SELECT *
FROM Listing_Aircover
ORDER BY listingID, policyNumber;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Result Grid

Form Editor

Field Types

listingID	policyNumber	macStart	macEnd	policyStartYear	policyStartMonth	policyStartDay	policyEndYear	policyEndMonth	policyEndDay
L0001	P0001	2021-03-14	2031-03-14	2021	3	14	2031	3	14
L0002	P0002	2021-08-07	2031-08-07	2021	8	7	2031	8	7
L0003	P0003	2022-01-29	2032-01-29	2022	1	29	2032	1	29
L0004	P0004	2021-09-02	2031-09-02	2021	9	2	2031	9	2
L0005	P0005	2022-01-25	2032-01-25	2022	1	25	2032	1	25
L0006	P0006	2021-06-12	2031-06-12	2021	6	12	2031	6	12
L0007	P0007	2021-10-30	2031-10-30	2021	10	30	2031	10	30
L0008	P0008	2021-02-15	2031-02-15	2021	2	15	2031	2	15
L0009	P0009	2021-07-11	2031-07-11	2021	7	11	2031	7	11
L0010	P0010	2021-05-09	2031-05-09	2021	5	9	2031	5	9
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- 15. Review

```
SELECT *
FROM Review
ORDER BY bookingID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

bookingID	forGuestComment	forGuestRating	forListingComment	forListingRating
B0001	respectful, checked in and out on time	5	Great location, thanks!	5
B0002	did not respond quickly	4	This made my birthday	5
B0003	checked out late	3	Thanks for going above and beyond and helping with the garage	5
B0004	Great guest, even took off bed sheets and left dishes washed	5	Bedbugs!!	1
B0005	Left food in fridge, but everything else was ok	4	Great spot to stay, center of town	5
B0006	So happy with these guests. Please come again!	5	This was so convenient. thanks for communicating	5
B0007	Let me know your eta next time	4	Hard to find the gate!	4
B0008	Good	5	Will come again!	5
B0009	Neighbors got upset because music was too loud	3	Had the best time.	5
B0010	Come again!	5	Wifi didnt work!	3
HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types

-- 16. Payment

```
SELECT *
FROM Payment
ORDER BY paymentID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

paymentID	bookingID	days	totalPaymentUSD	paymentMethod	macDate	year	month	Day	hour	minute
C0001	B0001	7	350.43	visa	2022-01-11 14:33:00	2018	1	9	13	45
C0002	B0002	25	2310.21	master	2022-01-19 03:22:00	2020	1	10	18	2
C0003	B0003	13	450.89	discover	2022-02-21 16:09:00	2021	2	4	12	34
C0004	B0004	18	1090.90	visa	2022-03-15 09:48:00	2022	3	13	21	26
C0005	B0005	4	276.67	master	2022-05-07 18:12:00	2023	1	10	17	49
C0006	B0006	22	1987.20	master	2022-06-05 07:26:00	2024	4	9	20	15
C0007	B0007	9	681.90	visa	2022-07-07 22:41:00	2025	7	12	23	36
C0008	B0008	27	2790.67	discover	2022-08-12 11:03:00	2025	5	15	13	21
C0009	B0009	16	1110.12	visa	2022-11-05 04:57:00	2025	8	17	16	1
C0010	B0010	5	678.12	visa	2022-12-17 20:29:00	2025	10	2	21	14
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types

-- 17. Host_Has_Listing

```
SELECT *
FROM Host_Has_Listing
ORDER BY hostID, listingID;
```

Result Grid Filter Rows: Search Edit: Export/Import:

hostID	listingID
H0001	L0001
H0001	L0002
H0002	L0003
H0002	L0009
H0003	L0004
H0003	L0005
H0004	L0006
H0004	L0010
H0005	L0007
H0005	L0008
HULL	HULL

Result Grid Form Editor Field Types

-- 18. Listing_Has_Amenity

```
SELECT *
FROM Listing_Has_Amenity
ORDER BY listingID, amenityID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

listingID	amenityID
L0001	Am0001
L0002	Am0002
L0003	Am0003
L0004	Am0001
L0005	Am0004
L0006	Am0002
L0007	Am0005
L0008	Am0001
L0009	Am0006
L0010	Am0005
HULL	HULL

-- 19. Listing_Is_Near_To_Attraction

```
SELECT *
FROM Listing_Is_Near_To_Attraction
ORDER BY listingID, attractionID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

listingID	attractionID
L0001	AT0001
L0002	AT0002
L0003	AT0003
L0004	AT0002
L0005	AT0004
L0006	AT0005
L0007	AT0003
L0008	AT0001
L0009	AT0001
L0010	AT0005
HULL	HULL

-- 20. Booking

```
SELECT *
FROM Booking
ORDER BY bookingID;
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid

bookingID	guestID	listingID
B0001	G0001	L0001
B0002	G0002	L0002
B0003	G0003	L0003
B0004	G0004	L0004
B0005	G0005	L0005
B0006	G0006	L0006
B0007	G0007	L0007
B0008	G0008	L0008
B0009	G0009	L0009
B0010	G0010	L0010
HULL	HULL	HULL

/*

2. Write an SQL involving the junction table and two other related tables.

You must use the INNER JOIN to connect with all three tables.

*/

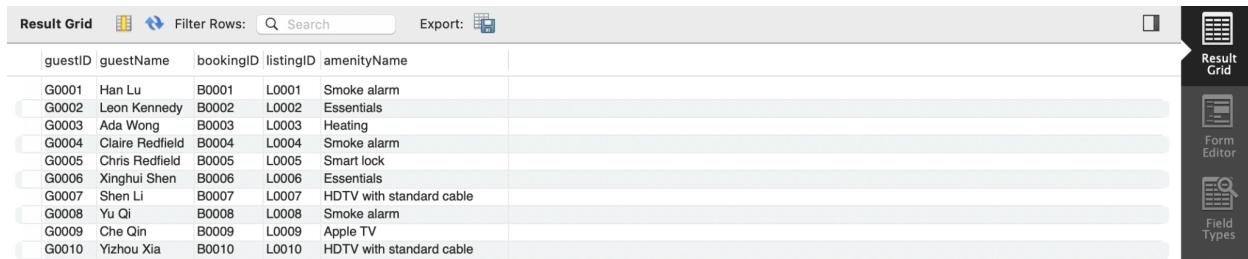
/*

Business situation:

This query identifies which amenities each guest experienced at the listings they booked. Airbnb can analyze guests' preferences regarding amenities,

so that in the future, airbnb can prioritize and recommend the listings with those amenities, providing personalized search results.

```
/*
SELECT
    G.guestID,
    CONCAT(G.firstName, ' ', G.lastName) AS guestName,
    B.bookingID,
    L.listingID,
    A.amenityName
FROM Booking AS B
INNER JOIN Guest AS G
    ON B.guestID = G.guestID
INNER JOIN Listing AS L
    ON B.listingID = L.listingID
INNER JOIN Listing_Has_Amenity AS LHA
    ON L.listingID = LHA.listingID
INNER JOIN Amenity AS A
    ON LHA.amenityID = A.amenityID
ORDER BY G.guestID, L.listingID, A.amenityID;
```



The screenshot shows a database query results grid. The columns are labeled: guestID, guestName, bookingID, listingID, and amenityName. The data consists of 10 rows, each representing a booking record with associated guest and listing details, along with the specific amenity listed. The grid has a header row and 10 data rows below it. To the right of the grid, there is a vertical toolbar with three icons: 'Result Grid' (selected), 'Form Editor', and 'Field Types'.

guestID	guestName	bookingID	listingID	amenityName
G0001	Han Lu	B0001	L0001	Smoke alarm
G0002	Leon Kennedy	B0002	L0002	Essentials
G0003	Ada Wong	B0003	L0003	Heating
G0004	Claire Redfield	B0004	L0004	Smoke alarm
G0005	Chris Redfield	B0005	L0005	Smart lock
G0006	Xinghui Shen	B0006	L0006	Essentials
G0007	Shen Li	B0007	L0007	HDTV with standard cable
G0008	Yu Qi	B0008	L0008	Smoke alarm
G0009	Che Qin	B0009	L0009	Apple TV
G0010	Yizhou Xia	B0010	L0010	HDTV with standard cable

```
/*
```

3. Write an SQL by including two or more tables and using the LEFT OUTER JOIN. Show the results and sort the results by key field(s).

```
*/
```

```
/*
```

Business situation:

This query lists all hosts and shows their listing details and aircover insurance information when available.

This helps Airbnb understand which hosts are actively hosting.

```
*/
```

```
SELECT
```

```
    H.hostID,
    CONCAT(H.firstName, ' ', H.lastName) AS hostName,
    H.introduction,
    HL.listingID,
```

```

LA.policyNumber
FROM Host AS H
LEFT OUTER JOIN Host_Has_Listing AS HL
    ON H.hostID = HL.hostID
LEFT OUTER JOIN Listing_Aircover AS LA
    ON HL.listingID = LA.listingID
ORDER BY H.hostID, HL.listingID, LA.policyNumber;

```

The screenshot shows a database query results grid. The columns are labeled: hostID, hostName, introduction, listingID, and policyNumber. The data consists of 10 rows, each representing a host and their corresponding listing. The introduction column contains a brief description of each host's personality and interests.

hostID	hostName	introduction	listingID	policyNumber
H0001	Duffy Bearington	I'm Duffy, a warmhearted traveler bear who loves sharing cozy seaside homes with...	L0001	P0001
H0002	ShellieMay Honeywell	Hi, I'm ShellieMay — I host charming pastel cottages filled with love and handmade...	L0002	P0002
H0003	StellaLou Starbelle	Hello! I'm StellaLou, a dancer bunny hosting graceful studios with soft lights and mir...	L0003	P0003
H0004	LinaBell Foxworth	Hello! I'm LinaBell, a clever fox who loves hosting curious guests in nature-inspired cabins.	L0004	P0004
H0005	Cinnamoroll Skypuff	I'm Cinnamoroll, your cloud-loving puppy host — come relax above the city lights.	L0005	P0005
H0006	Cinnamoroll Skypuff	I'm Cinnamoroll, your cloud-loving puppy host — come relax above the city lights.	L0006	P0006
H0007	Kuromi Mischief	Welcome! I'm Pomponpurin, a laid-back golden retriever who hosts sunny yellow ro...	NONE	NONE
H0008	Cappy Barrows	Hey there! I'm Kuromi — my gothic loft blends charm, chaos, and a dash of mischief.	NONE	NONE
H0009	Chinchilla Willowfur	Hello! I'm Cappy Barrows, a relaxed capybara who hosts tranquil riverside cabins wi...	NONE	NONE
H0010	Ginny Pigsworth	Hi, I'm Chinchilla — a soft-spoken chinchilla offering peaceful hillside stays with fluff...	NONE	NONE

```
/*
```

4. Write a single-row subquery. Show the results and sort the results by key field(s).

```
*/
```

```
/*
```

Business situation:

The quality team wants to identify the top-rated listings on the platform so they can feature them in marketing campaigns or “Superhost picks”. They want to see all listings whose rating equals the maximum current average rating in the system.

```
*/
```

```
SELECT
```

```
    l.listingID,
    l.currentAvgRating,
```

```
-- Host information
```

```
    h.hostID,
    h.firstName AS hostFirstName,
    h.lastName AS hostLastName,
    CONCAT(h.joinYear, '-', h.joinMonth, '-', h.joinDay) AS hostJoinDate,
```

```
-- Location information
```

```
    ll.city,
    ll.state,
    ll.country,
    ll.streetName,
```

```

a.amenityName,
atn.attractionName
FROM Listing AS l
JOIN Host_Has_Listing AS hl
ON l.listingID = hl.listingID
JOIN Host AS h
ON hl.hostID = h.hostID
JOIN Listing_location AS ll
ON l.latitude = ll.latitude
AND l.longitude = ll.longitude
LEFT JOIN Listing_Has_Amenity AS lha
ON l.listingID = lha.listingID
LEFT JOIN Amenity AS a
ON lha.amenityID = a.amenityID
LEFT JOIN Listing_Is_Near_To_Attraction AS lina
ON l.listingID = lina.listingID
LEFT JOIN Attraction AS atn
ON lina.attractionID = atn.attractionID
WHERE l.currentAvgRating =
SELECT MAX(currentAvgRating)
FROM Listing
)
ORDER BY l.listingID, a.amenityName, atn.attractionName;

```

The screenshot shows a database query results grid with the following columns and data:

listingID	currentAvgRating	hostID	hostFirstName	hostLastName	hostJoinDate	city	state	country	streetName	amenityName	attractionName
L0008	4.96	H0005	Cinnamoroll	Skypuff	2022-1-25	Los Angeles	California	USA	78 Sunset Blvd	Smoke alarm	Downtown

/*

5. Write a multiple-row subquery. Show the results and sort the results by key field(s).

*/

/*

Business situation:

The trust & safety team wants to see all listings owned by hosts who have ever submitted a concern report about guests. These hosts may be stricter or have had problematic guest experiences, so their listings might need tailored guidance or extra support.

*/

SELECT

```

h.hostID,
h.firstName AS hostFirstName,
h.lastName AS hostLastName,
hl.listingID,
```

```

hcr.reportID,
hcr.reportReason
FROM Host AS h
JOIN Host_Has_Listing AS hl
    ON h.hostID = hl.hostID
JOIN Host_Concern_Report AS hcr
    ON h.hostID = hcr.hostID
WHERE h.hostID IN (
    SELECT DISTINCT hostID
    FROM Host_Concern_Report
)
ORDER BY
    h.hostID,
    hl.listingID,
    hcr.reportID;

```

Result Grid Filter Rows: Search Export:

hostID	hostFirstName	hostLastName	listingID	reportID	reportReason
H0001	Duffy	Bearington	L0001	HR0001	Guest damaged property
H0001	Duffy	Bearington	L0001	HR0002	Guest left late
H0001	Duffy	Bearington	L0002	HR0001	Guest damaged property
H0001	Duffy	Bearington	L0002	HR0002	Guest left late
H0002	ShellieMay	Honeywell	L0003	HR0003	Excessive noise from guests
H0002	ShellieMay	Honeywell	L0003	HR0009	Key not returned
H0002	ShellieMay	Honeywell	L0009	HR0003	Excessive noise from guests
H0002	ShellieMay	Honeywell	L0009	HR0009	Key not returned
H0003	StellaLou	Starbelle	L0004	HR0004	Smoking in property
H0003	StellaLou	Starbelle	L0004	HR0005	Trash not disposed
H0003	StellaLou	Starbelle	L0005	HR0004	Smoking in property
H0003	StellaLou	Starbelle	L0005	HR0005	Trash not disposed
H0004	LinaBell	Foxworth	L0006	HR0006	Unauthorized pet
H0004	LinaBell	Foxworth	L0006	HR0010	Water leak reported late
H0004	LinaBell	Foxworth	L0010	HR0006	Unauthorized pet
H0004	LinaBell	Foxworth	L0010	HR0010	Water leak reported late
H0005	Cinnamoroll	Skypuff	L0007	HR0007	Broken furniture
H0005	Cinnamoroll	Skypuff	L0007	HR0008	Stolen towels
H0005	Cinnamoroll	Skypuff	L0008	HR0007	Broken furniture
H0005	Cinnamoroll	Skypuff	L0008	HR0008	Stolen towels

Result Grid Form Editor Field Types Query Stats Execution Plan

/*

6. Write an SQL to aggregate the results by using multiple columns in the SELECT clause.

*/

/*

Business situation:

The business team wants to know revenue by city and payment method, to see where most money comes from and which payment methods are used in each city.

*/

SELECT

```

    ll.city,
    p.paymentMethod,
    COUNT(DISTINCT p.bookingID) AS numBookings,
    SUM(p.totalPaymentUSD) AS totalRevenueUSD

```

FROM Payment AS p

JOIN Booking AS b

```

    ON p.bookingID = b.bookingID
JOIN Listing AS l
    ON b.listingID = l.listingID
JOIN Listing_location AS ll
    ON l.latitude = ll.latitude
    AND l.longitude = ll.longitude
GROUP BY
    ll.city,
    p.paymentMethod
ORDER BY
    ll.city,
    p.paymentMethod;

```

Result Grid Filter Rows: Search Export:

The screenshot shows a database query results grid. The columns are labeled: city, paymentMethod, numBookings, and totalRevenueUSD. The data rows are:

city	paymentMethod	numBookings	totalRevenueUSD
Aspen	master	1	1987.20
Dumas	master	1	2310.21
Dumas	visa	1	1090.90
Hebbronville	visa	1	678.12
Los Angeles	discover	1	2790.67
Mesa	visa	2	1032.33
Miami	master	1	276.67
Wellington	discover	1	450.89
Wellington	visa	1	1110.12

On the right side of the grid, there is a vertical toolbar with three items: "Result Grid" (selected), "Form Editor", and "Field Types".

```

/*
7. Write a subquery using the NOT IN operator. Show the results
and sort the results by key field(s).
*/
/*

```

Business situation:

This query identifies guests who have never filed any concern reports, which helps Airbnb identify low-risk, satisfied guests. The marketing or loyalty team can target these guests with special discounts or early-access offers to encourage repeat bookings.

```

*/
SELECT
    g.guestID,
    g.firstName,
    g.lastName,
    g.email
FROM Guest AS g
WHERE g.guestID NOT IN (
    SELECT DISTINCT guestID
    FROM Guest_Concern_Report
)
ORDER BY g.guestID;

```

guestID	firstName	lastName	email
G0007	Shen	Li	lishen0905@qq.com
G0008	Yu	Qi	qiyu0306@qq.com
G0009	Che	Qin	qinche0418@qq.com
G0010	Yizhou	Xia	xiayizhou0613@qq.com
HULL	HULL	HULL	HULL

/*

8. Write a query using a UNION statement.

Show the results and sort the results by key field(s).

*/

/*

Business situation:

This query creates a single combined incident log of all reports from both guests and hosts. Customer support and risk teams can use this unified view to monitor patterns, prioritize investigations, and track overall platform safety across both sides of the marketplace.

*/

SELECT

```
reportID,
'Guest' AS reporterType,
guestID AS personID,
reportReason,
state,
zipCode,
year,
month,
day,
hour,
minute
```

FROM Guest_Concern_Report

UNION

SELECT

```
reportID,
'Host' AS reporterType,
hostID AS personID,
reportReason,
state,
zipCode,
year,
month,
```

day,
hour,
minute

FROM Host_Concern_Report

ORDER BY reportID;

Result Grid

reportID	reporterType	personID	reportReason	state	zipCode	year	month	day	hour	minute
GR0001	Guest	G0001	Noise complaint	Arizona	85721	2022	1	25	8	33
GR0002	Guest	G0002	Cleaning issue	California	90001	2022	2	3	9	15
GR0003	Guest	G0003	Lake check-in	Nevada	89109	2022	2	15	22	48
GR0004	Guest	G0004	Broken air conditioner	Texas	73301	2022	4	2	14	20
GR0005	Guest	G0005	Wifi not working	Florida	33139	2022	5	27	13	18
GR0006	Guest	G0006	No hot water	Colorado	80014	2022	7	9	2	41
GR0007	Guest	G0001	Noise from neighbors	Arizona	85721	2022	1	28	19	12
GR0008	Guest	G0002	Host not responsive	California	90001	2022	2	15	22	48
GR0009	Guest	G0003	Cleaning issue	Nevada	89109	2022	3	11	16	29
GR0010	Guest	G0004	Bad smell	Texas	73301	2022	4	11	6	57
HR0001	Host	H0001	Guest damaged prop...	Arizona	85721	2022	2	2	4	51
HR0002	Host	H0001	Guest left late	Arizona	85721	2022	2	4	16	39
HR0003	Host	H0002	Excessive noise fro...	California	90001	2022	2	25	22	5
HR0004	Host	H0003	Smoking in property	Nevada	89109	2022	3	19	8	42
HR0005	Host	H0003	Trash not disposed	Nevada	89109	2022	3	20	15	33
HR0006	Host	H0004	Unauthorized pet	Texas	73301	2022	4	17	2	28
HR0007	Host	H0005	Broken furniture	Florida	33139	2022	6	2	10	41
HR0008	Host	H0005	Stolen towels	Florida	33139	2022	6	4	5	26
HR0009	Host	H0002	Key not returned	California	90001	2022	2	27	11	14
HR0010	Host	H0004	Water leak reported i...	Texas	73301	2022	4	21	23	57

Result Grid

Form Editor

Field Types

Query Stats

Execution Plan

/*

9-1. Write a query using the NOT EXISTS operator.

Show the results and sort the results by key field(s).

*/

/*

Business situation:

This query identifies listings which have not been booked.

These listings can be analyzed for undesirable qualities
and for fixing to ensure all listings are high quality.

*/

SELECT listingID FROM listing

WHERE NOT EXISTS (SELECT listingID from Booking);

Result Grid

listingID
NULL

Result Grid

/*

9-2. Write a query using the NOT EXISTS operator.

Show the results and sort the results by key field(s).

*/

/*

Business situation:

This query averages the listing ratings for bookings which do not have associated concern reports. When compared to the overall average rating (4.3) it seems that even if there are factors which detract from the rating, customers are not writing concern reports. Airbnb could institute a change in how concern reports are made to make it easier for unsatisfied customers to be identified and helped.

*/

```
SELECT AVG(forListingRating) as Average FROM review
    JOIN booking USING (bookingID)
    WHERE NOT EXISTS (SELECT 1 FROM guest_concern_report AS gcr
        WHERE gcr.guestID = booking.guestID);
```



/*

10. Write a subquery using the NOT NULL operator in the inner query.

Show the results and sort the results by key field(s).

*/

/*

Business situation:

As an Operations Manager at Airbnb, probably teams want to know which listings currently have AirCover coverage fully enabled. So in our datasets, we can check the listings that have a coverage start date and end date entered in the Listing_Aircover table.

*/

```
SELECT l.listingID, l.currentAvgRating,
la.policyNumber, la.macStart, la.macEnd
FROM Listing AS l
INNER JOIN Listing_Aircover AS la
    ON l.listingID = la.listingID
WHERE l.listingID IN (
    SELECT DISTINCT listingID FROM Listing_Aircover
    WHERE macStart IS NOT NULL
        AND macEnd IS NOT NULL)
ORDER BY
    la.listingID,
    la.policyNumber;
```

Result Grid Filter Rows: Search Export:

Result Grid

Form Editor

Field Types

listingID	currentAvgRating	policyNumber	macStart	macEnd
L0001	3.56	P0001	2021-03-14	2031-03-14
L0002	4.75	P0002	2021-08-07	2031-08-07
L0003	4.52	P0003	2022-01-29	2032-01-29
L0004	4.73	P0004	2021-09-02	2031-09-02
L0005	4.79	P0005	2022-01-25	2032-01-25
L0006	4.57	P0006	2021-06-12	2031-06-12
L0007	4.83	P0007	2021-10-30	2031-10-30
L0008	4.96	P0008	2021-02-15	2031-02-15
L0009	4.81	P0009	2021-07-11	2031-07-11
L0010	4.83	P0010	2021-05-09	2031-05-09