

BOOKING.COM





INTRODUCTION

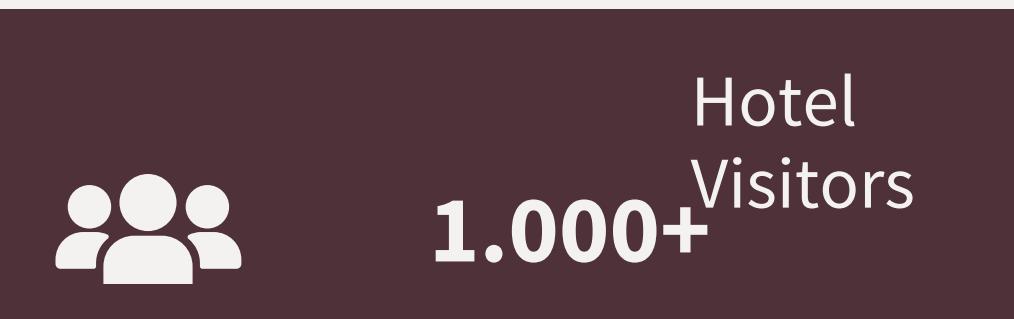
Founded in 1996 in Amsterdam, Netherlands, Booking.com started as a small Dutch startup.

Today, it is one of the world's leading digital travel platforms, owned by Booking Holdings Inc.

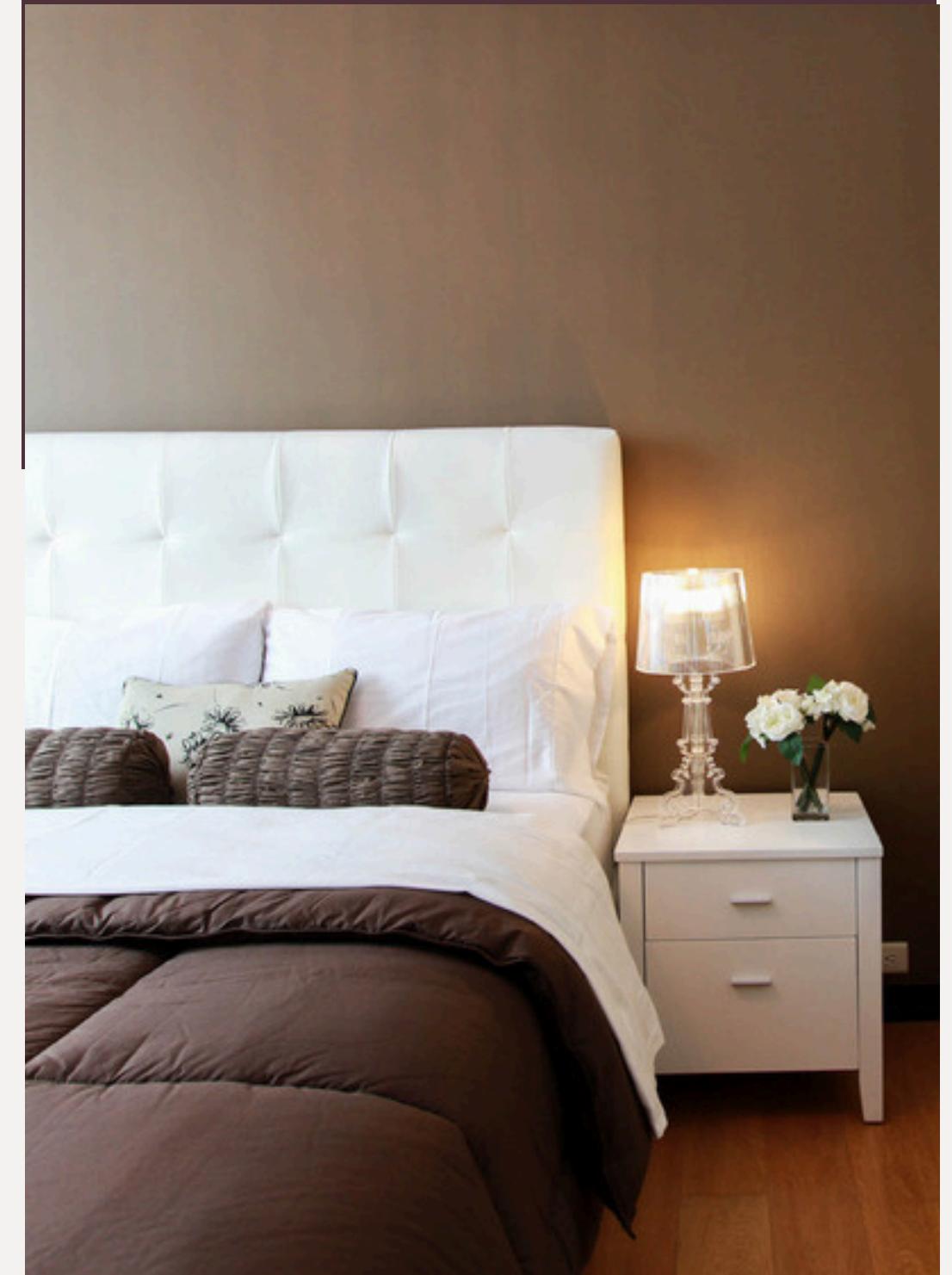
The platform allows users to search, compare, and book hotels, flights, vacation rentals, car rentals, and experiences.

Available in 40+ languages, it connects travelers with over 28 million accommodation listings worldwide.

Its mission is to make it easier for everyone to experience the world by providing affordable and accessible travel options.



DATABASE TABLES
CONTENT:
BOOKINGS
PAYMENTS
PROPERTIES
PROPERTY ADDRESSES
PROPERTY TYPES
ROLES
ROOMS
USER_PROFILES
USERROLES
USERS



ERD

D

I

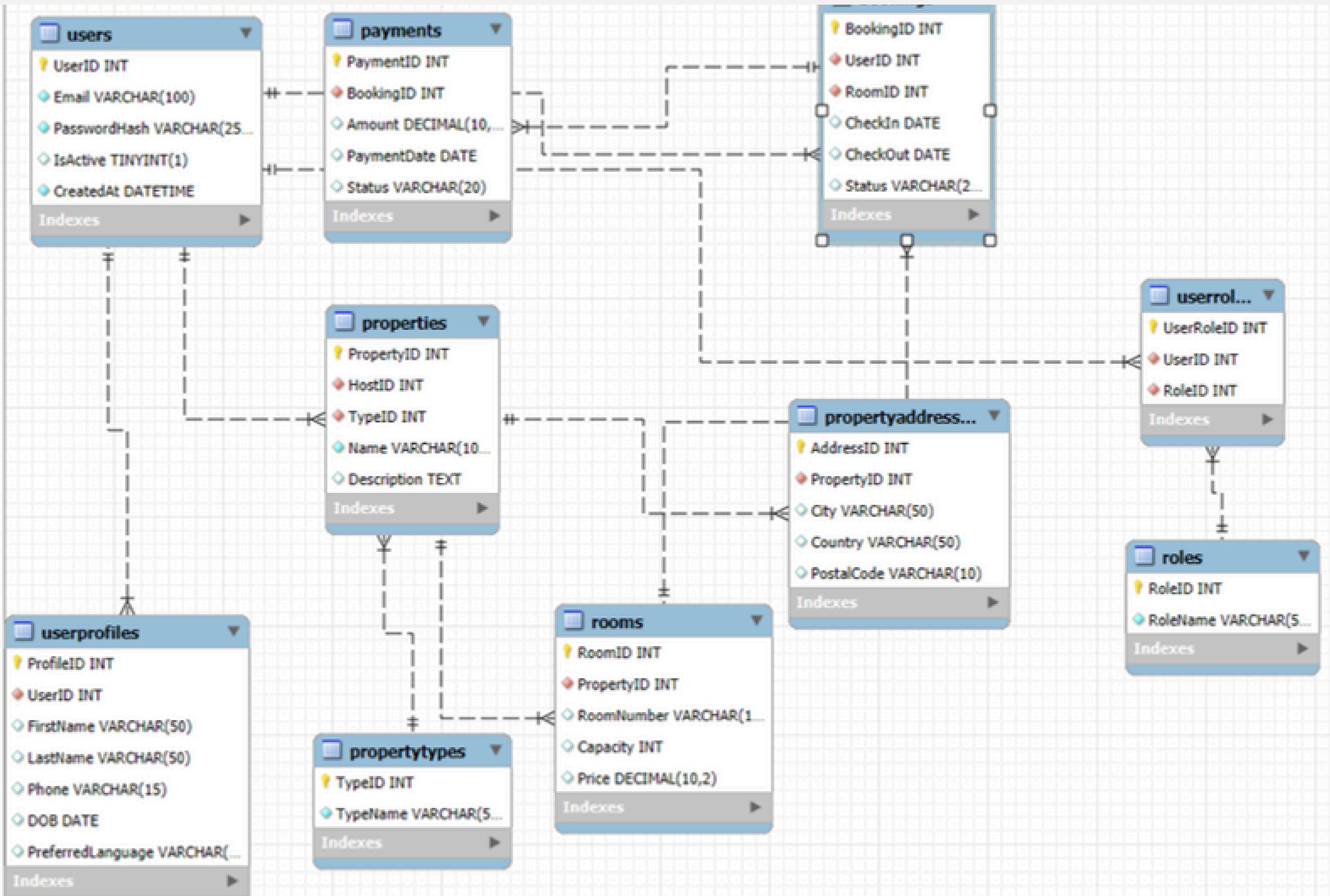
A

G

R

A

M



1. Get all active users

```
SELECT * FROM Users  
WHERE IsActive = 1;
```

	UserID	Email	PasswordHash	IsActive	CreatedAt
	8	henry@example.com	hash8	1	2024-01-17 00:00:00
	9	irene@example.com	hash9	1	2024-01-18 00:00:00
	10	jack@example.com	hash10	1	2024-01-19 00:00:00
	11	kate@example.com	hash11	1	2024-01-20 00:00:00
	12	leo@example.com	hash12	1	2024-01-21 00:00:00
	13	mona@example.com	hash13	1	2024-01-22 00:00:00
	14	nick@example.com	hash14	1	2024-01-23 00:00:00
	15	olivia@example.com	hash15	1	2024-01-24 00:00:00
	16	peter@example.com	hash16	1	2024-01-25 00:00:00
	17	queen@example.com	hash17	1	2024-01-26 00:00:00
	18	ray@example.com	hash18	1	2024-01-27 00:00:00
	19	susan@example.com	hash19	1	2024-01-28 00:00:00
	20	tom@example.com	hash20	1	2024-01-29 00:00:00
	NULL	NULL	NULL	NULL	NULL



List all properties located in "New York" (join Properties -> PropertyAddresses)

```
5  SELECT p.PropertyID, p.Name, pa.City, pa.Country  
6  FROM Properties p  
7  JOIN PropertyAddresses pa ON p.PropertyID = pa.PropertyID  
8  WHERE pa.City = 'New York';
```



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The result grid has four columns: 'PropertyID', 'Name', 'City', and 'Country'. A single row is displayed, showing values: 2, City Apartment, New York, USA.

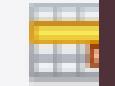
	PropertyID	Name	City	Country
→	2	City Apartment	New York	USA



Show all rooms with a price above 200

10 SELECT * FROM Rooms

11 WHERE Price > 200;

Result Grid |  Filter Rows: Edit:   

	RoomID	PropertyID	RoomNumber	Capacity	Price
▶	5	3	301	6	400.00
	8	6	601	2	300.00
	9	7	701	4	250.00
	12	10	1001	2	220.00
	15	13	1301	5	500.00
	17	15	1501	3	270.00
	20	18	1801	2	350.00
	NONE	NONE	NONE	NONE	NONE

Display all bookings with status "Confirmed"

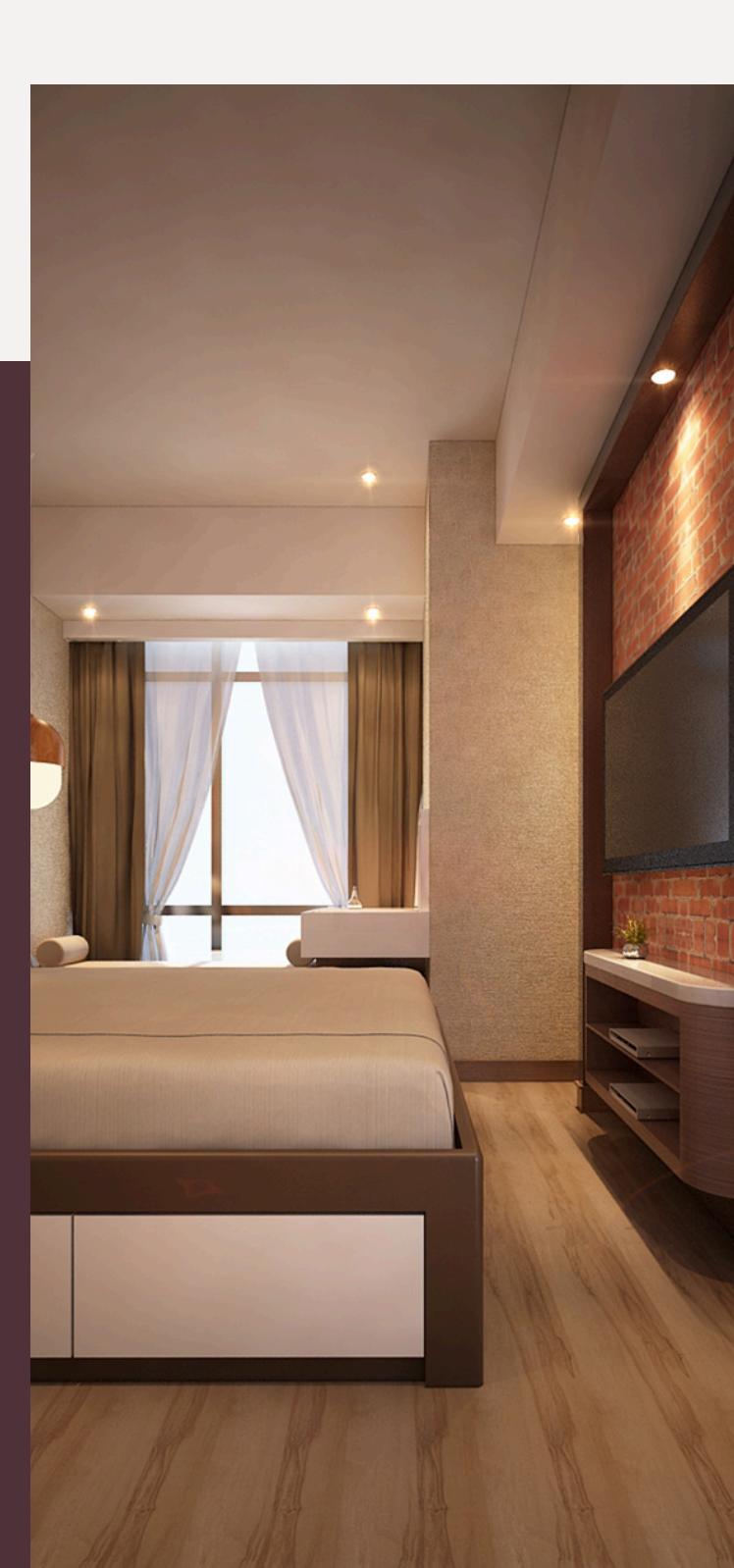
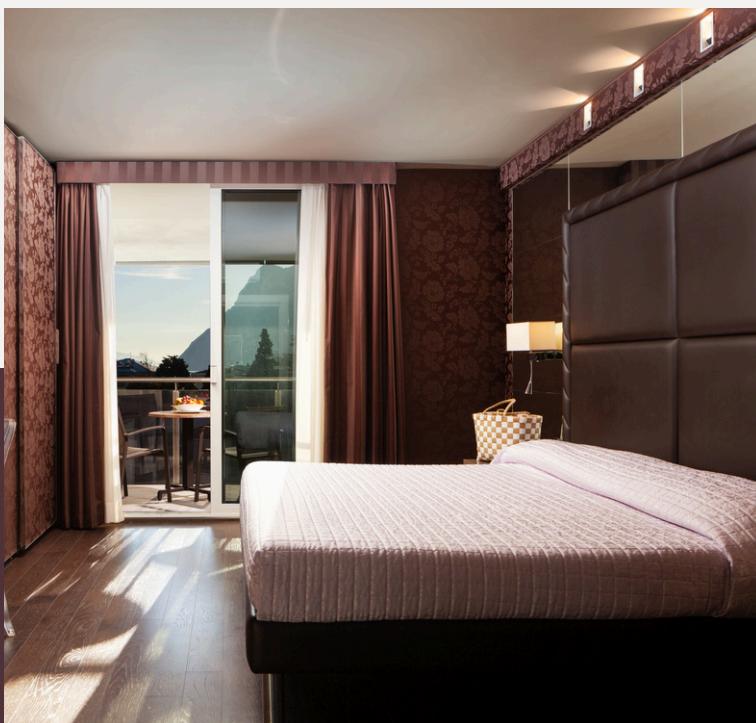
```
13  SELECT * FROM Bookings  
14  WHERE Status = 'Confirmed';
```

Result Grid | Filter Rows: Edit: Export/Import

	BookingID	UserID	RoomID	CheckIn	CheckOut	Status
▶	1	1	1	2024-05-01	2024-05-05	Confirmed
	4	4	4	2024-08-01	2024-08-08	Confirmed
	5	5	5	2024-09-10	2024-09-15	Confirmed
	7	7	7	2024-06-20	2024-06-25	Confirmed
	9	9	9	2024-08-01	2024-08-07	Confirmed
	10	10	10	2024-09-01	2024-09-05	Confirmed
	11	11	11	2024-05-01	2024-05-02	Confirmed
	13	13	13	2024-07-01	2024-07-05	Confirmed
	14	14	14	2024-08-01	2024-08-04	Confirmed
	16	16	16	2024-05-12	2024-05-15	Confirmed
	18	18	18	2024-07-10	2024-07-15	Confirmed
	19	19	19	2024-08-01	2024-08-06	Confirmed
	20	20	20	2024-09-01	2024-09-10	Confirmed
	NULL	NULL	NULL	NULL	NULL	NULL

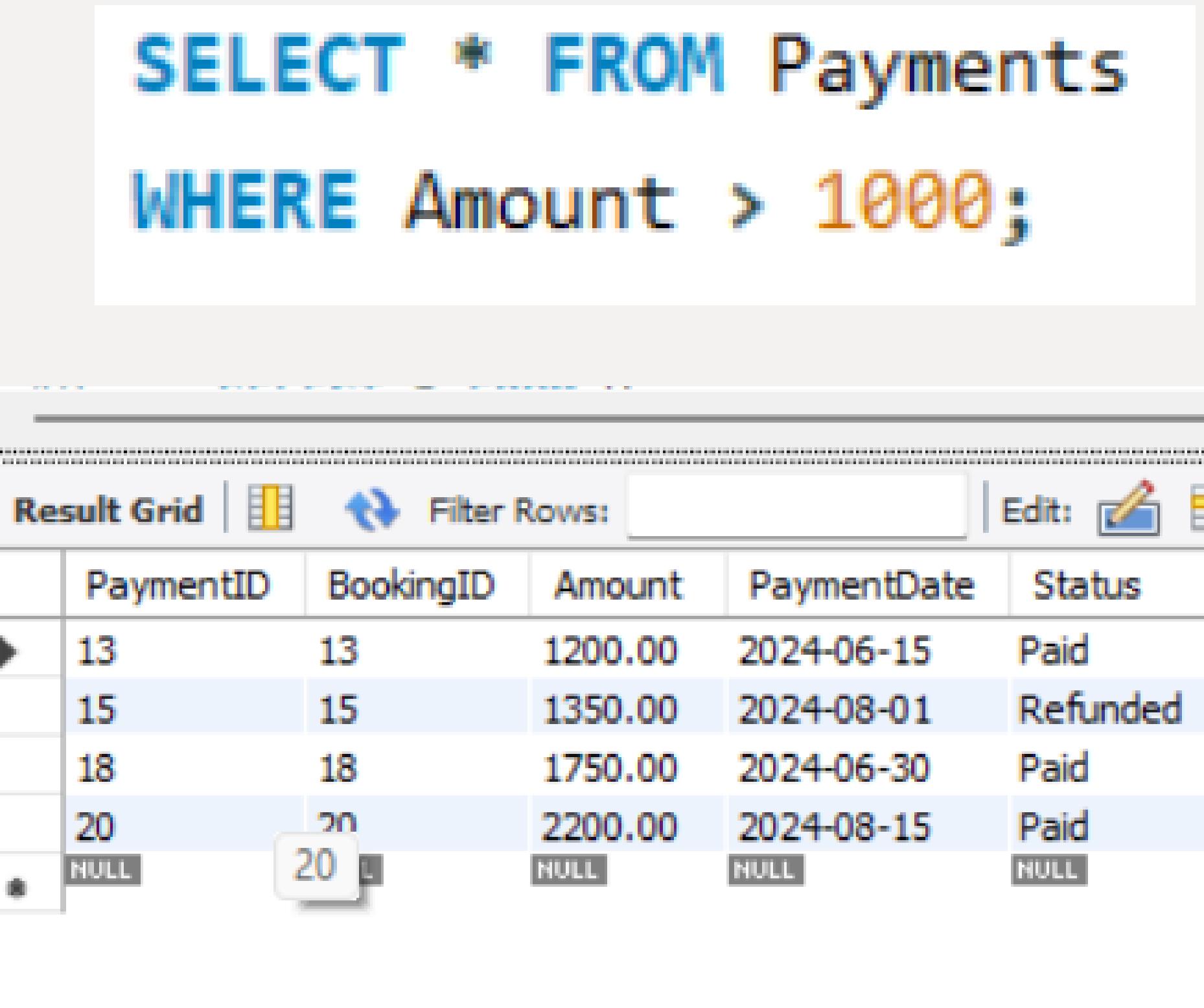


.Get all payments greater than 1000



SELECT * FROM Payments

WHERE Amount > 1000;

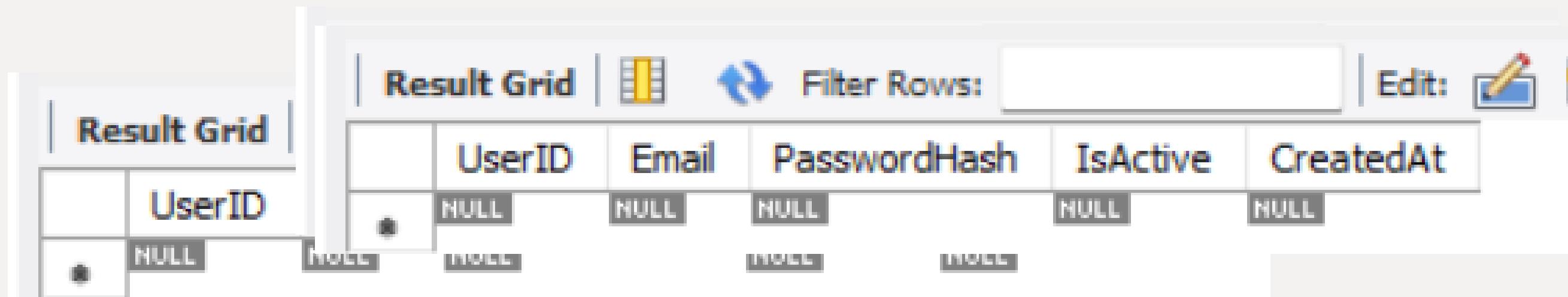


The screenshot shows a database interface with a result grid titled "Result Grid". The grid displays five columns: PaymentID, BookingID, Amount, PaymentDate, and Status. There are five rows of data:

	PaymentID	BookingID	Amount	PaymentDate	Status
▶	13	13	1200.00	2024-06-15	Paid
▶	15	15	1350.00	2024-08-01	Refunded
▶	18	18	1750.00	2024-06-30	Paid
▶	20	20	2200.00	2024-08-15	Paid
●	HULL	20	HULL	HULL	HULL

List users created after 2024-03-01

```
SELECT * FROM Users  
WHERE CreatedAt > '2024-03-01';
```



The screenshot shows a database query results grid. At the top, there is a toolbar with buttons for 'Result Grid' (highlighted), 'Filter Rows:', and 'Edit'. Below the toolbar is a header row with columns labeled: UserID, Email, PasswordHash, IsActive, and CreatedAt. The main data grid below the header is empty, displaying only the column headers and some placeholder text ('NULL') in the first few rows.

	UserID	Email	PasswordHash	IsActive	CreatedAt
	UserID				
	NONE	NONE	NONE	NONE	NONE
	NONE	NONE	NONE	NONE	NONE

Find all properties of type 'Hotel' (uses PropertyTypes lookup)

```
22  SELECT p.PropertyID, p.Name, pt.TypeName  
23  FROM Properties p  
24  JOIN PropertyTypes pt ON pTypeID = pt.TypeID;
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

	PropertyID	Name	TypeName
▶	1	Seaside Hotel	Hotel
21	Seaside Hotel	Hotel	
2	City Apartment	Apartment	
22	City Apartment	Apartment	
3	Mountain Villa	Villa	
23	Mountain Villa	Villa	
4	Youth Hostel	Hostel	
24	Youth Hostel	Hostel	
5	Sunny Guesthouse	Guesthouse	
25	Sunny Guesthouse	Guesthouse	
6	Island Resort	Resort	
26	Island Resort	Resort	
7	Jungle Bungalow	Bungalow	
27	Jungle Bungalow	Bungalow	
~	- · · · -	- · · · -	- · · · -



Rooms that have NO booking overlapping the date range 2024-06-01 .. 2024-06-30(returns rooms free for that whole window)

```
27  FROM Rooms r
28  LEFT JOIN Bookings b
29  ON r.RoomID = b.RoomID
30  AND NOT (b.CheckOut <= '2024-06-01' OR b.CheckIn >= '2024-06-30') -- overlap condition
31  WHERE b.BookingID IS NULL;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	RoomID	PropertyID	RoomNumber	Capacity	Price
1	1	1	101	2	100.00
3	2	2	201	2	150.00
4	2	2	202	4	200.00
5	3	3	301	6	400.00
6	4	4	401	1	50.00
8	6	6	601	2	300.00
9	7	7	701	4	250.00
10	8	8	801	2	180.00
11	9	9	901	3	140.00
13	11	11	1101	1	80.00
14	12	12	1201	2	110.00
15	13	13	1301	5	500.00

Show all bookings made by the user with email
'alice@example.com'

```
34    FROM Bookings b
35    JOIN Users u ON b.UserID = u.UserID
36    WHERE u.Email = 'alice@example.com';
```

The screenshot shows a database query results grid. At the top, there are navigation buttons for 'Result Grid' (with a grid icon), 'Filter Rows' (with a magnifying glass icon), 'Export' (with a floppy disk icon), and 'Wrap Cell Content' (with a text icon). Below the buttons is a table with the following data:

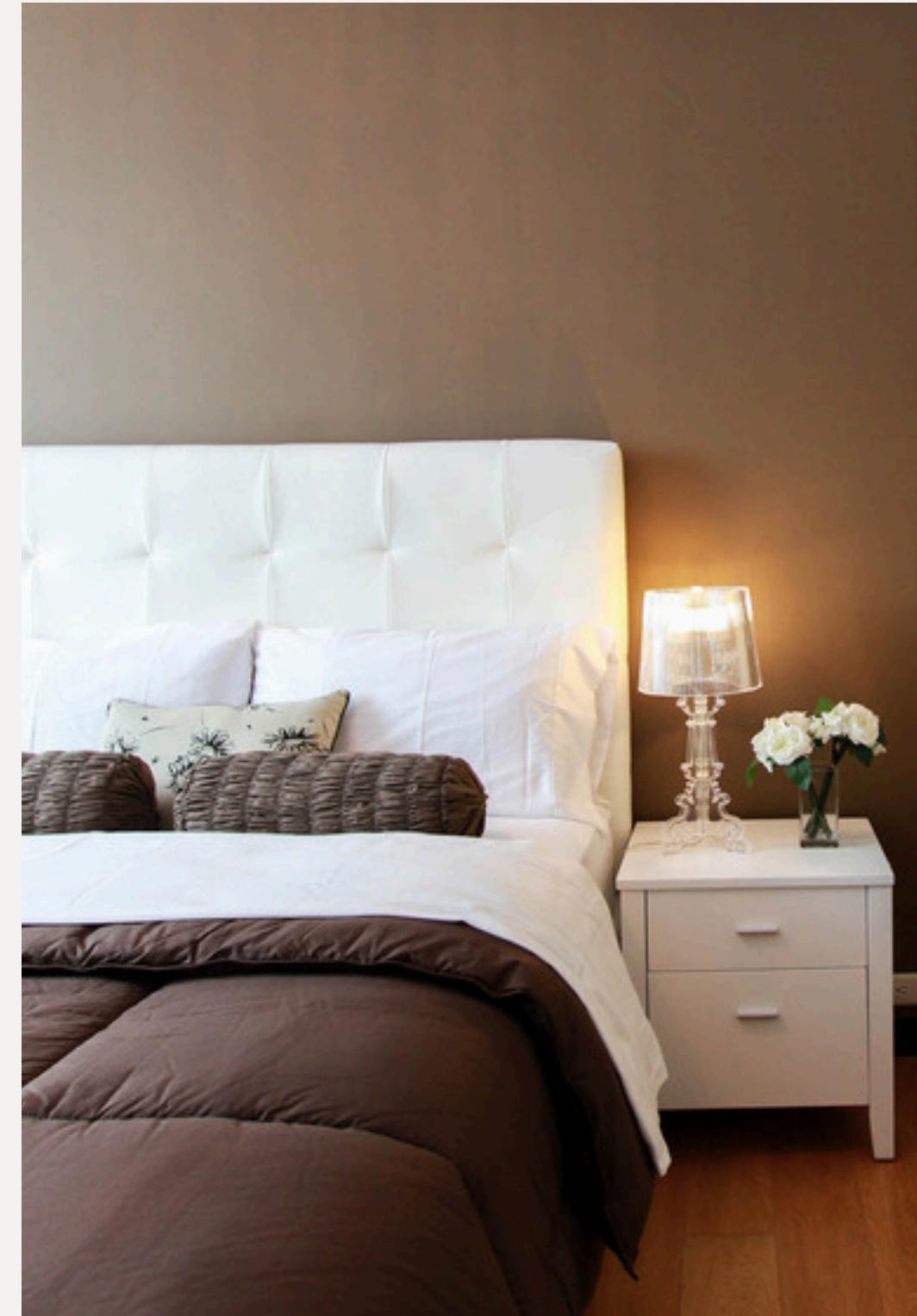
	BookingID	UserID	RoomID	CheckIn	CheckOut	Status
▶	1	1	1	2024-05-01	2024-05-05	Confirmed

Find users who have the role 'Guest' (via UserRoles -> Roles)

```
38 •   SELECT u.UserID, u.Email, r.RoleName  
39     FROM Users u  
40   JOIN UserRoles ur ON u.UserID = ur.UserID  
41   JOIN Roles r ON ur.RoleID = r.RoleID  
42 WHERE r.RoleName = 'Guest';
```

Result Grid | Filter Rows: | Export: Wrap Cell Cont

	UserID	Email	RoleName
▶	2	bob@example.com	Guest
	3	carol@example.com	Guest
	5	eva@example.com	Guest
	6	frank@example.com	Guest
	7	grace@example.com	Guest
	9	irene@example.com	Guest
	10	jack@example.com	Guest
	11	kate@example.com	Guest
	13	mona@example.com	Guest
	14	nick@example.com	Guest
	15	olivia@example.com	Guest
	16	peter@example.com	Guest
	18	ray@example.com	Guest
	19	susan@example.com	Guest
	--	--	--



List all properties with their address (city, country)

```
45 •     SELECT p.PropertyID, p.Name, pa.City, pa.Country, pa.PostalCode  
46      FROM Properties p  
47      LEFT JOIN PropertyAddresses pa ON p.PropertyID = pa.PropertyID;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	PropertyID	Name	City	Country	PostalCode
▶	1	Seaside Hotel	Miami	USA	33101
	2	City Apartment	New York	USA	10001
	3	Mountain Villa	Denver	USA	80201
	4	Youth Hostel	Berlin	Germany	10115
	5	Sunny Guesthouse	Paris	France	75001
	6	Island Resort	Maldives	Maldives	20002
	7	Jungle Bungalow	Bali	Indonesia	80361
	8	Forest Lodge	Zurich	Switzerland	8001
	9	Lake Cottage	Toronto	Canada	M5A 1A1
	10	Farmstay Bliss	Sydney	Australia	2000
	11	Capsule Inn	Tokyo	Japan	100-0001
	12	Roadside Motel	Los Ang...	USA	90001
	13	Alpine Chalet	Geneva	Switzerland	1201
	14	Houseboat Stay	Amsterdam	Netherlands	1011

.Show payments along with booking ID and payer email

```
49 •   SELECT pay.PaymentID, pay.Amount, pay.PaymentDate, pay.Status, b.BookingID, u.Email  
50      FROM Payments pay  
51      JOIN Bookings b ON pay.BookingID = b.BookingID  
52      JOIN Users u ON b.UserID = u.UserID;  
53      #14. Find all rooms located in 'London' (via PropertyAddresses) with their property names
```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

	PaymentID	Amount	PaymentDate	Status	BookingID	Email
▶	1	400.00	2024-04-25	Paid	1	alice@example.com
	2	600.00	2024-05-20	Pending	2	bob@example.com
	3	800.00	2024-06-20	Refunded	3	carol@example.com
	4	900.00	2024-07-20	Paid	4	david@example.com
	5	500.00	2024-08-01	Paid	5	eva@example.com
	6	200.00	2024-04-29	Pending	6	frank@example.com
	7	750.00	2024-06-15	Paid	7	grace@example.com
	8	300.00	2024-07-01	Refunded	8	henry@example.com
	9	650.00	2024-07-25	Paid	9	irene@example.com
	10	700.00	2024-08-20	Paid	10	jack@example.com
	11	100.00	2024-04-20	Paid	11	kate@example.com
	12	220.00	2024-05-20	Pending	12	leo@example.com
	13	1200.00	2024-06-15	Paid	13	mona@example.com
	14	320.00	2024-07-10	Paid	14	nick@example.com
	15	1350.00	2024-08-01	Refunded	15	olivia@example.com
	16	600.00	2024-05-10	Paid	16	peter@example.com

Find all rooms located in 'London' (via PropertyAddresses) with their property names

```
SELECT r.RoomID, r.RoomNumber, r.Price, p.Name AS PropertyName, pa.City  
FROM Rooms r  
JOIN Properties p ON r.PropertyID = p.PropertyID  
JOIN PropertyAddresses pa ON p.PropertyID = pa.PropertyID  
WHERE pa.City = 'London';
```

Result Grid | Filter Rows: Export: Wrap Cell Content

	RoomID	RoomNumber	Price	PropertyName	City
▶	19	1701	175.00	City Studio	London

.List payments with user first name (if profile exists), property name, and amount

```
62     JOIN Bookings b ON pay.BookingID = b.BookingID  
63     JOIN Users u ON b.UserID = u.UserID  
64     LEFT JOIN UserProfiles up ON u.UserID = up.UserID  
65     JOIN Rooms r ON b.RoomID = r.RoomID  
66     JOIN Properties p ON r.PropertyID = p.PropertyID;  
67 #16. Count total users registered
```

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	RoomID	RoomNumber	Price	PropertyName	City		
▶	19	1701	175.00	City Studio	London		

Count total users registered

```
SELECT COUNT(*) AS TotalUsers FROM Users;
```

Result Grid |

TotalUsers	
▶	20

Number of properties per city

```
70 • SELECT pa.City, COUNT(*) AS NumProperties  
71   FROM Properties p  
72   JOIN PropertyAddresses pa ON p.PropertyID = pa.PropertyID  
73   GROUP BY pa.City  
74   ORDER BY NumProperties DESC.
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	City	NumProperties
▶	Miami	1
	New York	1
	Denver	1
	Berlin	1
	Paris	1
	Maldives	1
	Bali	1
	Zurich	1
	Toronto	1
	Sydney	1
	Tokyo	1
	Los Ang...	1
	Geneva	1
	Amsterdam	1
	Nairobi	1
	Marrakech	1

Average room price by property

```
76 •   SELECT p.PropertyID, p.Name, AVG(r.Price) AS AvgRoomPrice  
77     FROM Properties p  
78     JOIN Rooms r ON p.PropertyID = r.PropertyID  
79     GROUP BY p.PropertyID, p.Name  
80     ORDER BY AvgRoomPrice DESC:
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content:

PropertyID	Name	AvgRoomPrice
13	Alpine Chalet	500.000000
3	Mountain Villa	400.000000
18	Condo Heights	350.000000
6	Island Resort	300.000000
15	Safari Camp	270.000000
7	Jungle Bungalow	250.000000
10	Farmstay Bliss	220.000000
16	Moroccan Riad	200.000000
8	Forest Lodge	180.000000
2	City Apartment	175.000000
17	City Studio	175.000000
14	Houseboat Stay	160.000000
9	Lake Cottage	140.000000
1	Seaside Hotel	110.000000
12	Roadside Motel	110.000000
5	Sunny Guestho...	90.000000



Total revenue per property (sum of payments joined through bookings -> rooms -> properties)

```
82 •   SELECT p.PropertyID, p.Name, COALESCE(SUM(pay.Amount),0) AS TotalRevenue  
83     FROM Properties p  
84     LEFT JOIN Rooms r ON p.PropertyID = r.PropertyID  
85     LEFT JOIN Bookings b ON r.RoomID = b.RoomID  
86     LEFT JOIN Payments pay ON b.BookingID = pay.BookingID  
87     GROUP BY p.PropertyID, p.Name  
88     ORDER BY TotalRevenue DESC;
```

Result Grid		
PropertyID	Name	TotalRevenue
18	Condo Heights	2200.00
16	Moroccan Riad	1750.00
2	City Apartment	1700.00
13	Alpine Chalet	1350.00
11	Capsule Inn	1200.00
1	Seaside Hotel	1000.00
5	Sunny Guesthouse	750.00
8	Forest Lodge	700.00
7	Jungle Bungalow	650.00
14	Houseboat Stay	600.00
	Grand Total	10000.00

Count bookings made in April 2024 (by booking created date not available; use CheckIn month)

```
90 •     SELECT COUNT(*) AS BookingsInApril2024  
91      FROM Bookings  
92      WHERE YEAR(CheckIn) = 2024 AND MONTH(CheckIn) = 4;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	BookingsInApril2024			
▶	0			

Number of bookings per property (grouped by property)

```
2 •  SELECT p.PropertyID, p.Name, COUNT(b.BookingID) AS NumBookings  
3      FROM Properties p  
4      LEFT JOIN Rooms r ON p.PropertyID = r.PropertyID  
5      LEFT JOIN Bookings b ON r.RoomID = b.RoomID  
6      GROUP BY p.PropertyID, p.Name  
7      ORDER BY NumBookings DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	PropertyID	Name	NumBookings
▶	1	Seaside Hotel	2
	2	City Apartment	2
	3	Mountain Villa	1
	4	Youth Hostel	1
	5	Sunny Guesthouse	1
	6	Island Resort	1
	7	Jungle Bungalow	1
	8	Forest Lodge	1
	9	Lake Cottage	1
	10	Farmstay Bliss	1
	11	Capsule Inn	1
	12	Roadside Motel	1
	13	Alpine Chalet	1
	14	Houseboat Stay	1
	15	Safari Camp	1

.Number of bookings per user

```
9 •   SELECT u.UserID, u.Email, COUNT(b.BookingID) AS BookingsCount  
10    FROM Users u  
11    LEFT JOIN Bookings b ON u.UserID = b.UserID  
12    GROUP BY u.UserID, u.Email  
13    ORDER BY BookingsCount DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	UserID	Email	BookingsCount
▶	1	alice@example.com	1
	2	bob@example.com	1
	3	carol@example.com	1
	4	david@example.com	1
	5	eva@example.com	1
	6	frank@example.com	1
	7	grace@example.com	1
	8	henry@example.com	1
	9	irene@example.com	1
	10	jack@example.com	1
	11	kate@example.com	1
	12	leo@example.com	1
	13	mona@example.com	1
	14	nick@example.com	1
	15	olivia@example.com	1
	16	meter@example.com	1

.Number of rooms available per property (rooms count; availability not tracked, so count rooms)

```
15 •   SELECT p.PropertyID, p.Name, COUNT(r.RoomID) AS TotalRooms  
16     FROM Properties p  
17     LEFT JOIN Rooms r ON p.PropertyID = r.PropertyID  
18     GROUP BY p.PropertyID, p.Name;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	PropertyID	Name	TotalRooms
▶	1	Seaside Hotel	2
	2	City Apartment	2
	3	Mountain Villa	1
	4	Youth Hostel	1
	5	Sunny Guesthouse	1
	6	Island Resort	1
	7	Jungle Bungalow	1
	8	Forest Lodge	1
	9	Lake Cottage	1
	10	Farmstay Bliss	1
	11	Capsule Inn	1
	12	Roadside Motel	1
	13	Alpine Chalet	1
	14	Houseboat Stay	1
	15	Safari Camp	1
	16	Moroccan Riad	1

Total payments grouped by month (year-month)

```
SELECT DATE_FORMAT(PaymentDate, '%Y-%m') AS YearMonth, SUM(Amount) AS TotalPayments  
FROM Payments  
GROUP BY YearMonth  
ORDER BY YearMonth;
```

Result Grid | Filter Rows:

	YearMonth	TotalPayments
▶	2024-04	700.00
	2024-05	1420.00
	2024-06	4950.00
	2024-07	2670.00
	2024-08	4750.00

How many bookings each user has (same as 22 — alternative showing only users with >0)

```
25 •   SELECT u.UserID, u.Email, COUNT(b.BookingID) AS NumBookings  
26     FROM Users u  
27     JOIN Bookings b ON u.UserID = b.UserID  
28     GROUP BY u.UserID, u.Email  
29     ORDER BY NumBookings DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	UserID	Email	NumBookings
▶	1	alice@example.com	1
	2	bob@example.com	1
	3	carol@example.com	1
	4	david@example.com	1
	5	eva@example.com	1
	6	frank@example.com	1
	7	grace@example.com	1
	8	henry@example.com	1
	9	irene@example.com	1
	10	jack@example.com	1
	11	kate@example.com	1
	12	leo@example.com	1
	13	mona@example.com	1
	14	nick@example.com	1
	15	olivia@example.com	1
	16	peter@example.com	1
	17	queen@example.com	1

Properties that have at least one booking (acts like 'popular' properties)

```
31 •   SELECT DISTINCT p.PropertyID, p.Name  
32     FROM Properties p  
33     JOIN Rooms r ON p.PropertyID = r.PropertyID  
34     JOIN Bookings b ON r.RoomID = b.RoomID;  
35
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	PropertyID	Name
▶	1	Seaside Hotel
	2	City Apartment
	3	Mountain Villa
	4	Youth Hostel
	5	Sunny Guesthouse
	6	Island Resort
	7	Jungle Bungalow
	8	Forest Lodge
	9	Lake Cottage
	10	Farmstay Bliss
	11	Capsule Inn
	12	Roadside Motel
	13	Alpine Chalet
	14	Houseboat Stay
	15	Safari Camp
	16	Moroccan Riad



Overview of Booking.com

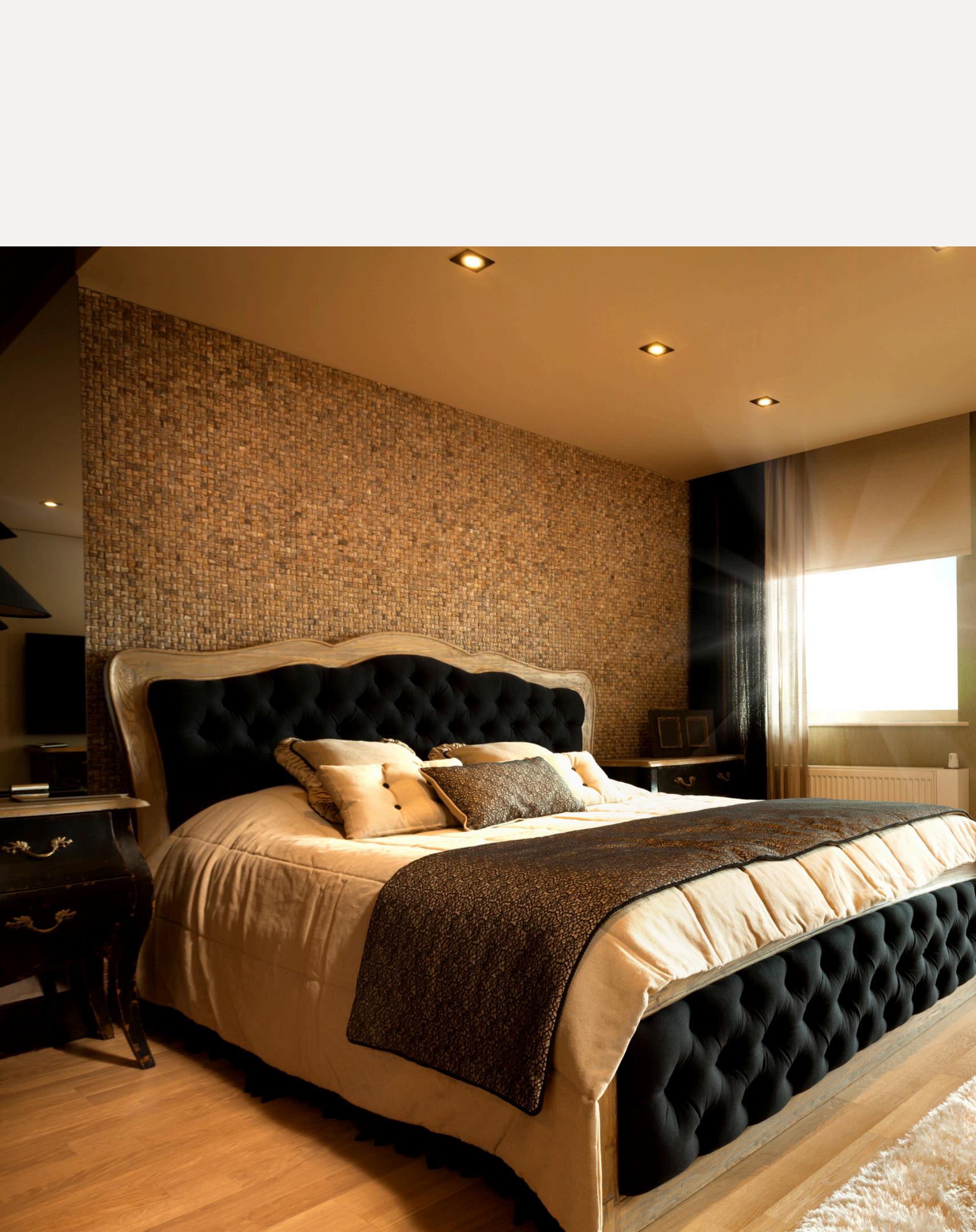
- Founded: 1996, Amsterdam, Netherlands
- Parent Company: Booking Holdings Inc.
- Core Services: Hotels, flights, car rentals, vacation homes, and attractions.
- Global Reach: Over 220+ countries and territories, 40+ languages.
- Listings: More than 28 million properties worldwide.
- Mission: "Make it easier for everyone to experience the world."





Conclusion :

- Booking.com has grown from a small startup into a global travel giant.
- Its wide network of partners, advanced technology, and customer-first approach make it one of the most reliable travel platforms.
- However, it must continue to innovate and adapt in a competitive market where consumer preferences are shifting toward personalized, sustainable, and flexible travel options.
- Overall, Booking.com remains a leader in online travel services, shaping how millions of people explore the world.



Thank You

www.booking.com

