Build a Data Mart in SQL (DLBDSPBDM01)

AIRBNB DATABASE DESIGN

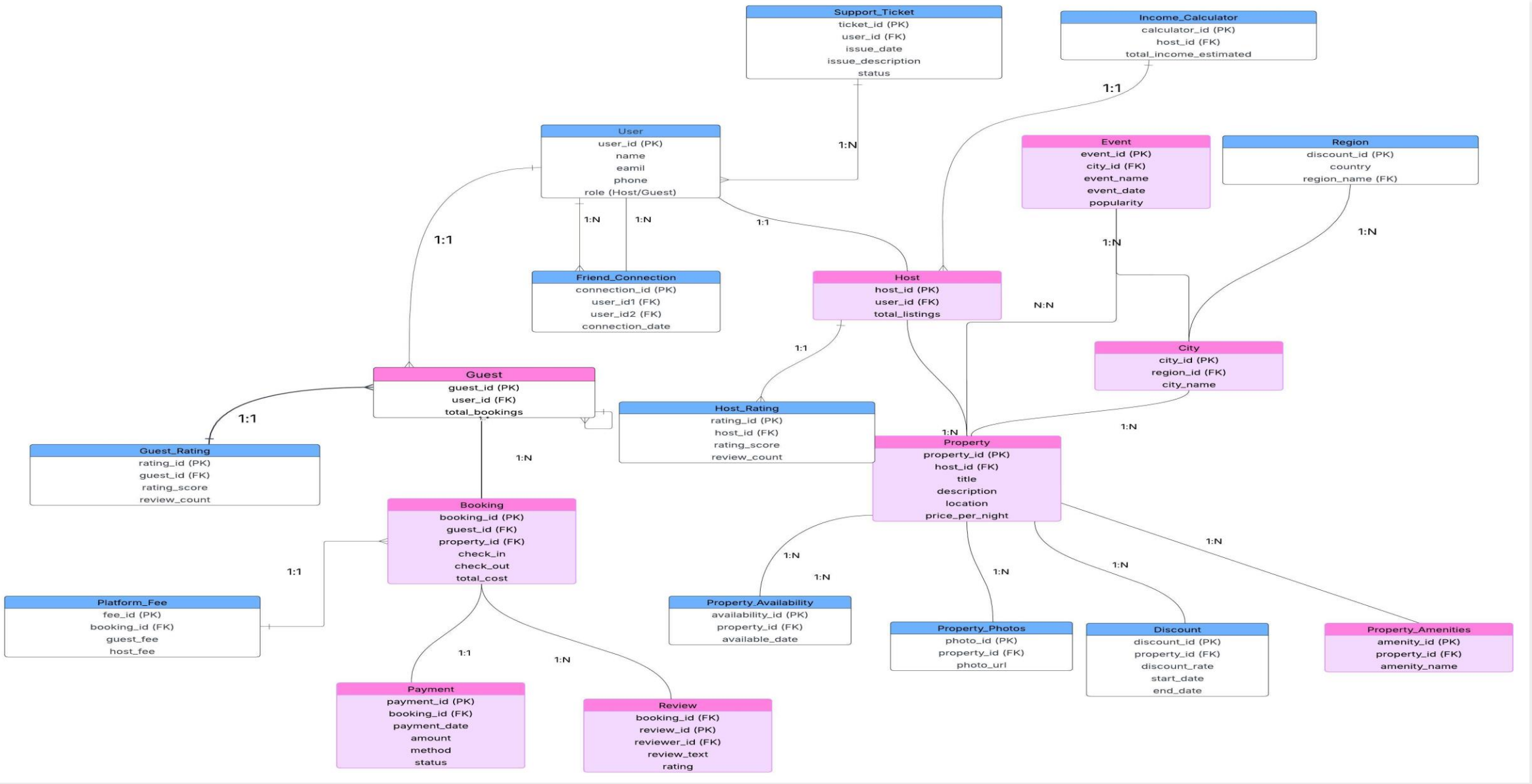
By Tawakalt Akolade

Introduction

This project designs a relational database for Airbnb to manage users, properties, bookings, payments, and reviews efficiently. The database supports secure transactions, smooth booking processes, and organized property listings.

Using Chen’s Notation, the ER diagram maps out 20 entities, ensuring database integrity and clear relationships. The database is implemented in MySQL, ensuring efficient queries, data consistency, and platform reliability.

ER Model: Graphical Representation of the Airbnb Database



DATA BASE DESIGN PROCESS

The Airbnb database was designed to manage Airbnb operations, ensuring data integrity and security using the following steps:

1.Creation of Entities: 20 essential entities were created (e.g., User, Property, Booking, Payment, etc.).

2. Defining Relationships: Chen notation was used to map out 1:M, M:N relationships.

3. Normalization was applied to eliminate redundancy.

4. Appropriate data types and constraints were selected to ensure data integrity (e.g., **INT** for PK, **VARCHAR** for names/emails, **DECIMAL** for prices and, **Dates** for booking period).

SQL Statements
Database Structure

1. USER TABLE

```
CREATE TABLE user (user_id INT  PRIMARY KEY, name VARCHAR(100) NOT NULL, email
                   VARCHAR(100), phone VARCHAR(100) NOT NULL, role ENUM('HOST’,
                   'Guest') NOT NULL);
```

TEST QUERY

```
SELECT * FROM users;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

user_id	name	email	phone	role
1	Tawakalt AKolade	beekayzdata@gmail.com	09072649151	HOST
2	Folaranmi Rodiat	folaranmi@gmail.com	090765432192	Guest
3	Michael Johnson	michael@email.com	1122334455	HOST
4	Emily Brown	emily@email.com	6677889900	Guest
5	Daniel Wilson	daniel@email.com	5566778899	Guest
6	Sophia Taylor	sophia@email.com	4433221100	Guest
7	David Anderson	david@email.com	7788990011	HOST
8	Olivia Martinez	olivia@email.com	9988776655	Guest
9	James Thomas	james@email.com	3344556677	HOST
10	Isabella Hernandez	isabella@email.com	1122446688	Guest
11	Ethan White	ethan@email.com	2233445566	HOST
12	Mia Robinson	mia@email.com	4455667788	Guest

5566778899

2. HOST TABLE

```
CREATE TABLE host   (host_id INT PRIMARY KEY, user_id INT UNIQUE, total_listings
                    INT DEFAULT 0, FOREIGN KEY (user_id) REFERENCES
                    user(user_id));
```

TEST QUERY

```
SELECT * FROM Host ORDER BY host_id;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

guest_id	user_id	total_bookings
4	8	4
5	10	1
6	12	6
7	14	3
8	16	2
9	18	5
10	20	4
11	21	2
12	22	5
13	23	3
14	24	4
15	25	1
16	26	6
17	27	2
18	28	3

3. GUEST TABLE

```
CREATE TABLE guest (guest_id INT PRIMARY KEY, user_id INT UNIQUE,
                    total_bookings INT DEFAULT 0, FOREIGN KEY (user_id)
                    REFERENCES User(user_id));
```

TEST QUERY

```
SELECT * FROM Guest ORDER BY guest_id;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

guest_id	user_id	total_bookings
4	8	4
5	10	1
6	12	6
7	14	3
8	16	2
9	18	5
10	20	4
11	21	2
12	[illegible]	5
13	23	3
14	24	4
15	25	1
16	26	6
17	27	2
18	28	3
19	29	5
20	30	4
NULL	NULL	NULL

4. PROPERTY TABLE

```
CREATE TABLE property   (property_id INT PRIMARY KEY, host_id INT, title
                         VARCHAR(200), description TEXT, location VARCHAR(255),
                         price_per_night DECIMAL(10,2), FOREIGN KEY (host_id)
                         REFERENCES HOST(host_id));
```

TEST QUERY

```
SELECT * FROM Property WHERE host_id =3;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

property_id	host_id	title	description	location	price_per_night
3	3	Cozy Cabin	A warm, wood-finished cabin in the mountains.	Denver, CO	150.00
13	3	Desert Retreat	A peaceful stay in the heart of the desert.	Phoenix, AZ	140.00

5. PROPERTY PHOTOS TABLE

```
CREATE TABLE property_photos (photo_id INT NOT NULL, property_id INT PRIMARY KEY
                              AUTO_INCREMENT, photo_url VARCHAR(255) NOT NULL,
                              FOREIGN KEY (property_id) REFERENCES
                              PROPERTY(property_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Property_Photos ORDER BY photo_id;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

photo_id	property_id	photo_url
1	1	https://example.com/photos/property1_1.jpg
2	2	https://example.com/photos/property1_2.jpg
3	3	https://example.com/photos/property2_1.jpg
4	4	https://example.com/photos/property2_2.jpg
5	5	https://example.com/photos/property3_1.jpg
6	6	https://example.com/photos/property3_2.jpg
7	7	https://example.com/photos/property4_1.jpg
8	8	https://example.com/photos/property4_2.jpg
9	9	https://example.com/photos/property5_1.jpg
10	10	https://example.com/photos/property5_2.jpg
11	11	https://example.com/photos/property6_1.jpg
12	12	https://example.com/photos/property6_2.jpg
13	13	https://example.com/photos/property7_1.jpg
14	14	https://example.com/photos/property7_2.jpg

6. PROPERTY AVAILABILITY TABLE

```
CREATE TABLE property_availabilty    (availability_id INT PRIMARY KEY
                                      AUTO_INCREMENT,  property_id INT NOT NULL,
                                      available_date DATE NOT NULL,  FOREIGN KEY
                                      (property_id) REFERENCES
                                      Property(property_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Property_Availability ORDER BY property_id;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

availability_id	property_id	available_date
1	1	2024-07-01
2	1	2024-07-10
3	2	2024-07-02
4	2	2024-07-12
5	3	2024-07-03
6	3	2024-07-14
7	4	2024-07-04
8	4	2024-07-16
9	5	2024-07-05
10	5	2024-07-18
11	6	2024-07-06
12	6	2024-07-20
13	7	2024-07-07
14	7	2024-07-22
15	8	2024-07-08

7. BOOKING TABLE

```
CREATE TABLE BOOKING (booking_id INT PRIMARY KEY AUTO_INCREMENT,guest_id INT NOT
                      NULL,property_id INT NOT NULL,check_in DATE NOT
                      NULL,check_out DATE NOT NULL, total_cost DECIMAL(10,2) NOT
                      NULL,FOREIGN KEY (guest_id) REFERENCES Guest(guest_id) ON
                      DELETE CASCADE,FOREIGN KEY (property_id) REFERENCES
                      Property(property_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT COUNT(*) AS total_bookings FROM Booking;
```

QUERY RESULT

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

total_bookings
20

8. PAYMENT TABLE

```
CREATE TABLE Payment (payment_ID INT PRIMARY KEY AUTO_INCREMENT,booking_id INT
                      NOT NULL,payment_date DATE NOT NULL,amount DECIMAL(10,2)
                      NOT NULL,method ENUM( 'Credit Card', 'Paypal', 'Cash’,
                      'Bank Transfer') NOT NULL, status ENUM('failed', 'pending’,
                      'successful') NOT NULL,FOREIGN KEY (booking_id) REFERENCES
                      Booking(booking_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Payment WHERE method = 'Credit Card' AND status = 'Successful'
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

payment_ID	booking_id	payment_date	amount	method	status
1	1	2024-07-02	520.00	Credit Card	successful
6	6	2024-07-12	480.00	Credit Card	successful
8	8	2024-07-16	890.00	Credit Card	successful
11	11	2024-07-22	780.00	Credit Card	successful
13	13	2024-07-26	500.00	Credit Card	successful
17	17	2024-08-04	720.00	Credit Card	successful
20	20	2024-08-10	990.00	Credit Card	successful
NULL	NULL	NULL	NULL	NULL	NULL

9. REVIEW TABLE

```
CREATE TABLE Review (review_id INT PRIMARY KEY AUTO_INCREMENT, booking_id INT
                     NOT NULL, reviewer_id INT NOT NULL, review_text TEXT NOT
                     NULL, rating ENUM('1', '2', '3', '4', '5’), FOREIGN KEY
                     (booking_id) REFERENCES Booking(booking_id) ON DELETE
                     CASCADE, FOREIGN KEY (reviewer_id) REFERENCES User(user_id)
                     ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Review WHERE reviewer_id IN  (2, 4, 6);
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

review_id	booking_id	reviewer_id	review_text	rating
1	1	2	Amazing place! Very clean and cozy.	5
2	2	4	Great stay, but the Wi-Fi was slow.	4
3	3	6	Good value for money. Will visit again.	5
NULL	NULL	NULL	NULL	NULL

10. PLATFORM FEE TABLE

```
CREATE TABLE platform_fee (fee_id INT PRIMARY KEY AUTO_INCREMENT, booking_id INT
                           NOT NULL, host_fee DECIMAL(10,2) NOT NULL, guest_fee
                           DECIMAL(10,2) NOT NULL, FOREIGN KEY (booking_id)
                           REFERENCES Booking(booking_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT SUM(host_fee) AS total_host_fees, SUM(guest_fee) AS total_guest_fees FROM
Platform_Fee;
```

QUERY RESULT

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

	total_host_fees	total_guest_fees
▶	1450.00	725.00

11. SUPPORT TICKET TABLE

```
CREATE TABLE support_ticket (ticket_id INT PRIMARY KEY AUTO_INCREMENT, user_id
                             INT NOT NULL,issue_date DATE NOT NULL,
                             issue_description TEXT NOT NULL, status
                             ENUM('Open', 'In Progress', 'Resolved', 'Closed’)
                             NOT NULL,FOREIGN KEY (user_id) REFERENCES
                             user(user_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT status, COUNT(*) AS ticket_count FROM Support_Ticket GROUP BY status;
```

QUERY RESULT

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

status	ticket_count
Resolved	6
In Progress	4
Open	6
Closed	4

12. HOST RATING TABLE

```
CREATE TABLE host_rating (rating_id INT PRIMARY KEY AUTO_INCREMENT, host_id INT
                          NOT NULL, rating_score ENUM ('1', '2', '3', '4', '5’)
                          NOT NULL, review_count INT NOT NULL, FOREIGN KEY
                          (host_id) REFERENCES Host(host_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Host_Rating WHERE host_id IN (1, 3, 5);
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

rating_id	host_id	rating_score	review_count
72	1	5	12
74	3	5	15
76	5	4	11
NULL	NULL	NULL	NULL

13. GUEST RATING TABLE

```
CREATE TABLE guest_rating (rating_id INT PRIMARY KEY AUTO_INCREMENT, guest_id
                           INT NOT NULL, rating_score ENUM ('1', '2', '3', '4’,
                           '5’) NOT NULL, review_count INT NOT NULL,FOREIGN KEY
                           (guest_id) REFERENCES Guest(guest_id) ON DELETE
                           CASCADE);
```

TEST QUERY

```
SELECT * FROM Guest_Rating WHERE review_count >= 10;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

rating_id	guest_id	rating_score	review_count
1	1	5	12
3	3	5	15
5	5	4	11
7	7	5	20
9	9	4	10
11	11	5	14
14	14	5	17
15	15	4	12
17	17	5	19
19	19	4	11

14. REGION TABLE

```
create table region (region_id INT PRIMARY KEY AUTO_INCREMENT,region_name
                     VARCHAR(50) NOT NULL,country VARCHAR(50) NOT NULL);
```

TEST QUERY

```
SELECT * FROM Region ORDER BY region_name ASC;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

region_id	region_name	country
14	Abuja	Nigeria
7	Bavaria	Germany
8	Berlin	Germany
6	British Columbia	Canada
1	California	USA
16	Gauteng	South Africa
9	Hamburg	Germany
10	Ile-de-France	France
15	Kano	Nigeria
18	KwaZulu-Natal	South Africa

15. CITY TABLE

```
CREATE TABLE city (city_id INT PRIMARY KEY AUTO_INCREMENT, region_id INT NOT
                   NULL, city_name VARCHAR(50) NOT NULL, column FOREIGN KEY
                  (region_id) REFERENCES region(region_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Region WHERE region_name = 'California';
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

region_id	region_name	country
1	California	USA
NULL	NULL	NULL

16. DISCOUNT TABLE

```
CREATE TABLE Discount (discount_id INT PRIMARY KEY AUTO_INCREMENT, property_id
                       INT NOT NULL, discount_rate DECIMAL(5,2) NOT NULL,
                       start_date DATE NOT NULL, end_date DATE NOT NULL,
                       FOREIGN KEY (property_id) REFERENCES
                       Property(property_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Discount ORDER BY discount_rate DESC LIMIT 5;
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

discount_id	property_id	discount_rate	start_date	end_date
17	17	50.00	2024-09-20	2024-10-05
14	14	40.00	2024-09-05	2024-09-20
13	13	35.00	2024-09-01	2024-09-15
8	8	30.00	2024-08-05	2024-08-20
7	7	25.00	2024-08-01	2024-08-15
NULL	NULL	NULL	NULL	NULL

17. PROPERTY AMENITIES TABLE

```
CREATE TABLE Property_Amenities (amenity_id INT PRIMARY KEY AUTO_INCREMENT,
                                 property_id INT NOT NULL, amenity_name
                                 VARCHAR(100) NOT NULL, FOREIGN KEY
                                 (property_id) REFERENCES Property(property_id)
                                 ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Property_Amenities WHERE amenity_name IN ('Swimming Pool', 'Parking
Space', 'Gym Access’);
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

amenity_id	property_id	amenity_name
2	2	Swimming Pool
4	4	Parking Space
5	5	Gym Access
NULL	NULL	NULL

18. EVENT TABLE

```
CREATE TABLE Event (event_id INT PRIMARY KEY AUTO_INCREMENT, city_id INT NOT
                    NULL, event_name VARCHAR(100) NOT NULL, event_date DATE NOT
                    NULL, popularity INT NOT NULL, FOREIGN KEY (city_id)
                    REFERENCES City(city_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT * FROM Event WHERE event_date BETWEEN '2024-07-01' AND '2024-07-31';
```

QUERY RESULT

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	city_id	event_name	event_date	popularity
1	1	Music Festival	2024-07-10	85
2	2	Food & Wine Expo	2024-07-15	78
3	3	Tech Conference	2024-07-20	90
4	4	Marathon Race	2024-07-25	82
5	5	Cultural Parade	2024-07-30	88
NULL	NULL	NULL	NULL	NULL

19. FRIEND CONNECTION TABLE

```
CREATE TABLE Friend_Connection (connection_id INT PRIMARY KEY AUTO_INCREMENT,
                                user_id1 INT NOT NULL, user_id2 INT NOT NULL,
                                connection_date DATE NOT NULL, FOREIGN KEY
                                (user_id1) REFERENCES User(user_id) ON DELETE
                                CASCADE, FOREIGN KEY (user_id2) REFERENCES
                                User(user_id) ON DELETE CASCADE);
```

TEST QUERY

```
SELECT COUNT(*) AS total_connections FROM Friend_Connection;
```

QUERY RESULT

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

	total_connections
▶	20

20. INCOME CALCULATOR TABLE

```
CREATE TABLE Income_Calculator (calculator_id INT PRIMARY KEY AUTO_INCREMENT,
                                host_id INT NOT NULL, total_income_estimated
                                DECIMAL(10,2) NOT NULL, FOREIGN KEY (host_id)
                                REFERENCES Host(host_id) ON DELETE CASCADE);
```

TEST QUERY

```
DESC income_calculator;
```

QUERY RESULT

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

Field	Type	Null	Key	Default	Extra
calculator_id	int	NO	PRI	NULL	auto_increment
host_id	int	NO	MUL	NULL	
total_income_estimated	decimal(10,2)	NO		NULL	