

Managing Data and Databases

— Term Project —

Final Deliverable

Tina Dimova (260986550), Camila Loyola Rodriguez (261044780), Jeremy Gadoua (260983904), Elaine Yu (261030551), and Nemanja Gmitrovic (261055389)

Desautels Faculty of Management - McGill University

INSY437: Managing Data and Databases

Professor Dominique Welt

19th April 2023

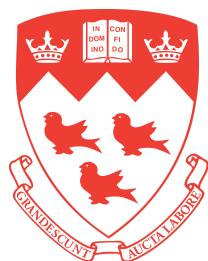


Table of Contents

Business Scenario Overview	2
Sample wireframe mockups	4
Mission Statement and Objectives	5
Mission Statement	5
Objectives	5
ERD	6
Relational Model	7
Data Dictionary	8
Description of Entities	8
Description of Attributes	9
SQL Statements	12
5 Functional SQL Statements	12
10 informational SQL Statements descriptions	14
MDDB design table	19
Star Model	20
Metadata Dictionary	21
Nodes and relationship diagram	22
Nodes and Diagram Relationship	23
Sample Graph	24
2 cypher queries description	25

Business Scenario Overview

Our business is called CapiFun. Our business model is a peer-to-peer rental service that has an application-based model that allows people to rent yards from each other. The business will earn revenue by charging a small commission on transactions. The platform will allow users to connect via a built-in chat feature and phone if necessary.

Users that wish to rent their yards will be asked to give information regarding the yard they wish to set up. People can then view this yard and rent it if it meets their requirements. The filters for renting out a yard include the allocated date and time the person wishes to rent the yard, and the location of the yard, which include information such as the country, city, municipality, street name, address, etc. Furthermore, the size of the yard will play a significant role in a person's decision-making, as well as the number of people involved with the presumed event that will be conducted. Amenities such as pets, pools, jacuzzis, barbecues, tables, alcohol, smoking, and clean-up fees are all further add-ons that will increase the price of any rented terrain. Finally, all yards/terrains will allow others to rate them depending on the satisfaction of the service provided, which will also play into fluctuating prices. Depending on their budget and needs, people can play around with their desired yard.

The renters of the yard will be able to decline or accept any offers for renting their yards depending on the rating of the person trying to rent. Any past renting experiences will be present on the user account; therefore, renters or people that rent can assess others' experiences through their ratings.

By developing a new database, we can adequately store the intensive amounts of data generated from the accounts (name, age, rating, etc.), all renting agreements, information on renting filters, etc. Furthermore, setting up a database will allow all that data to be present in one spot, meaning that any data that needs to be found or aggregated will be easily found.

The entire process will start with users entering their information on the app or the website; this information is already mentioned above. The users can then mention whether they wish to be a renter or someone who rents, which will be present on their profile. Renters will then be able to put up their postings on a homepage that could be viewed by anyone wishing to find a yard to rent. On the viewer's side, they will be able to set up filters depending on their needs, and if a renting opportunity that meets their filters is present, they can contact a renter.

People wishing to rent out a yard will be allowed to visit the yard to ensure it is up to their demands and that there are no unwanted surprises they may find on the day of the rental. This provides the person wishing to rent out the yard with a certification that the other side is a valid renter.

Both sides must accept each other before continuing with the final process of renting the yard. Both parties then meet up, the event occurs, and they finally give each other a rating after the event.

With the database system, the organization will achieve a more substantial unity of all the data instead of being dispersed over various documents, sheets, other data-holding platforms, etc. The organization will also achieve quick and efficient renting procedures for individuals that use the application or website for their bookings, as data will be readily available for the users to find their best choices.

The users of the project will be everyday individuals that are either willing to rent their yard to others or people that wish to rent a yard for any event they may have.

Some of the rules associated with the business include a strict time constraint that must be followed by the person renting out the yard. If that person goes over the time limit, there could be penalties and extra fees, or the renter can also decide to let the person continue the event but with a higher fee. Cleaning fees will also be applied if the person renting out the yard does not comply with properly cleaning up the backyard adequately and making sure any committed mess is cleaned by the end of the allocated time.

Further rules include only being able to cancel an event with a 48-hour notice to the person who owns the yard. Any other cases before 48 hours are valid and will not result in negative backlash. If anyone does end up canceling an event within 48 hours of that event, the person renting out the yard will be asked to provide a percentage of the money that they owe to the renter for any inconvenience.

Considering this is a service business, the person renting out their yard must ensure that all selected filters, chosen by the person wishing to rent the yard, are present and working the day of the actual renting. This means checking on the quality of the grass, the cleanliness of the pool, the barbecue working adequately, having a fully functional bathroom, etc. If any of these features end up not functioning the day of the rental, the person renting out the yard reserves the right to provide comments and a rating about the experience. Therefore, renters should be wary about having everything being fully functional as it has a positive effect on other people's perceptions of the yard and the quality service the renter brings.

Sample wireframe mockups

The top navigation bar features the CAPIFUN logo with a cartoon hamster icon, a search bar, and a 'Filters' button. A message 'Over 1,000 results' is displayed above the search results. The results are presented in a grid format:

- Georgetown, Maine, US**
10 minute walk to Mile Beach
April 10 - 15 \$450/hour
- York, Maine, US**
8 minute walk to Short Sands
September 20 - 24 \$200/hour
- Hampton, New Hampshire, US**
1 minute walk to Hampton Beach
March 25 - 30 \$80/hour

Each listing includes a small thumbnail image, the location name, a brief description, and the rental period and price. There are also heart icons for favoriting.

The top navigation bar features the CAPIFUN logo with a cartoon hamster icon, a search bar, and a 'MyAccount' button. The main content area shows a large thumbnail image of a garden with a white umbrella and deck chairs. To the right, there is a 'share' link and a summary of the property's details:

16+ guests | 5.5 baths
★ ★ ★ ★ 1/5 [read reviews](#)

THINGS TO KNOW:

- No dog(s) are welcome in this home. No other animals are allowed without specific approval.
- Parking notes: There is free parking for 6 vehicles.

Damage waiver: The total cost of your reservation for this Property includes a damage waiver fee which covers you for up to \$3,000 of accidental damage to the Property or its contents. More information can be found from the "Additional rules" on the checkout page.
[...more info](#)

CONTACT

Below the main content, there is a section for the York, Maine listing:

York, Maine, US
8 minute walk to Short Sands
September 20 - 24 \$200/hour

Mission Statement and Objectives

Mission Statement

The Mission Statement of CapiFun is to facilitate the distribution of acquired data between renters and people renting out yards to allow for seamless interactions and better and faster communication and searches, tracking of status, and reporting. As well as an increase in the use of the application/website.

Objectives

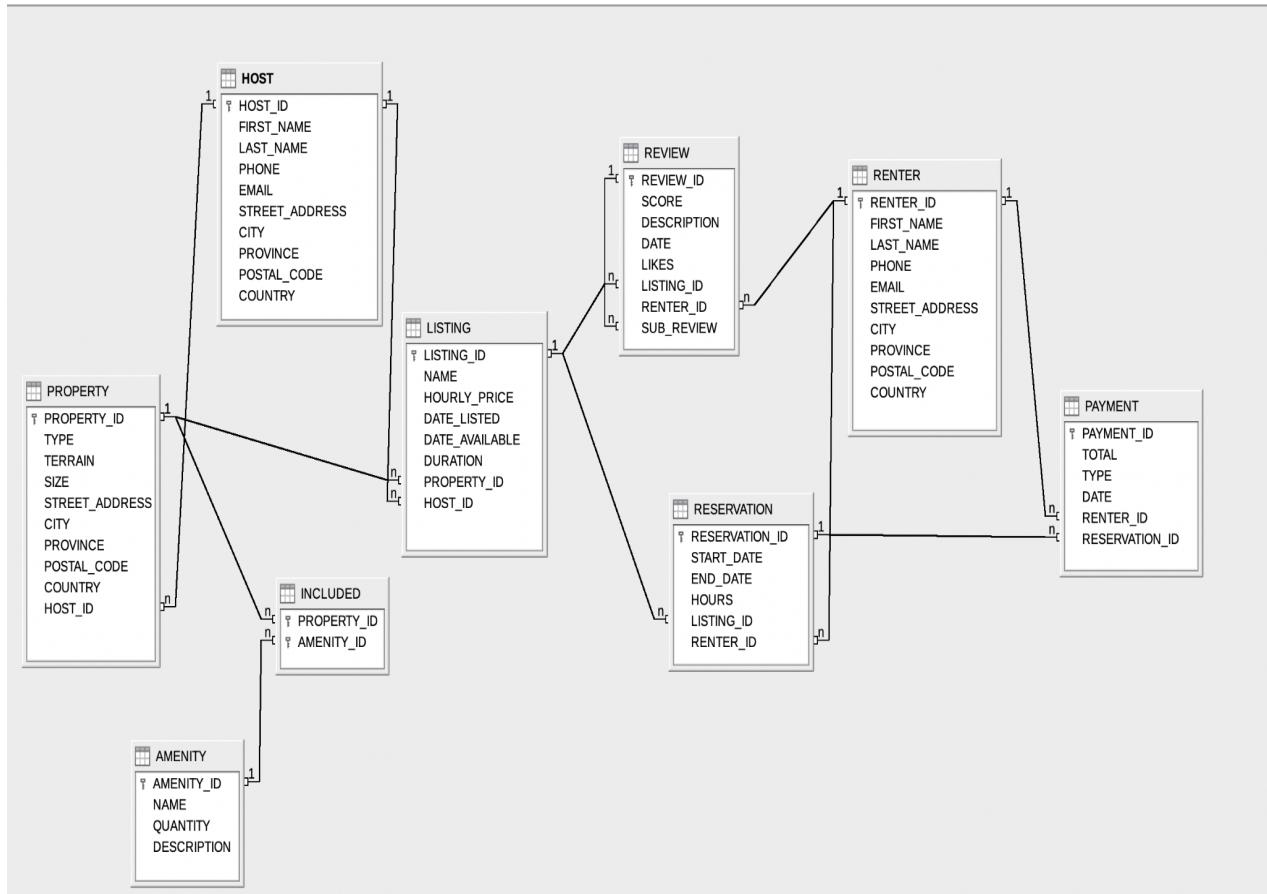
- To maintain (enter, update, delete) data on renters
- To maintain (enter, update, delete) data on people renting out yards
- To maintain (enter, update, delete) data on the renter's property/ies
- To maintain (enter, update, delete) data on properties for rent
- To maintain (enter, update, delete) data on demands for yard renting

- To perform searches on renters
- To perform searches on people renting out yards
- To perform searches on the renter's property/ies
- To perform searches on properties for rent
- To perform searches on demands for yard renting

- To track the status of the renter's property/ies
- To track the status of the properties for rent
- To track the status of the demands for yard renting

- To report on renters
- To report on people renting out yards
- To report on the renter's property/ies
- To report on properties for rent
- To report on demands for yard renting

ERD



Relational Model

Host(Host_ID, First_Name, Last_Name, Phone, Email, Street_Address, City, Province, Postal_Code, Country)

Property(Property_ID, Type, Terrain, Size, Street_Address, City, Province, Postal_Code, Country, *Host_ID*)

Amenity(Amenity_ID, Name, Quantity, Description)

Included(*Property_ID*, Amenity_ID)

Listing(Listing_ID, Name, Hourly_Price, Date_Listed, Date_Available, Duration, *Property_ID*, *Host_ID*)

Renter(Renter_ID, First_Name, Last_Name, Phone, Email, Street_Address, City, Province, Postal_Code, Country)

Reservation(Reservation_ID, Start_Date, End_Date, Hours, *Listing_ID*, *Renter_ID*)

Payment(Payment_ID, Total, Type, Date, *Renter_ID*, *Reservation_ID*)

Review(Review_ID, Score, Description, Date, Likes, *Listing_ID*, *Renter_ID*, Sub_Review).

Note: FKs are in italics instead of being circled.

Data Dictionary

Description of Entities

Entity Name	Description	Occurrence
Host	Contains the information needed to identify a particular host	Each host represents a single occurrence. <ul style="list-style-type: none"> • One host can have 0 or many listings • One host can own 0 or many properties.
Property	Contains the information about a particular property	Each property represents a single occurrence. <ul style="list-style-type: none"> • One property can have only 1 host. • One property can turn into 0 or many listings (considering the possibility that a listing gets deleted and replaced by a new one instead of being reused). • One property can provide 0 or many amenities.
Amenity	Contains the information about a particular amenity	Each amenity represents a single occurrence. <ul style="list-style-type: none"> • One amenity can be linked to 0 or many properties.
Included	Links the properties and their respective amenity/amenities	Each included pair is linked to 1 property and 1 amenity only.
Listing	Contains the information about a particular listing	Each listing represents a single occurrence. <ul style="list-style-type: none"> • One listing can have only 1 host • One listing can be about only 1 property • One listing can have 0 or many reviews
Renter	Contains the information needed to identify a particular renter	Each renter represents a single occurrence. <ul style="list-style-type: none"> • One renter can write 0 or many reviews • One renter can make 0 or many reservations • One renter can make 0 or many payments
Reservation	Contains the information about a particular reservation	Each reservation represents a single occurrence. <ul style="list-style-type: none"> • One reservation can have only 1 renter. • One reservation can be confirmed by only 1 payment • One reservation is made for only 1 listing.
Payment	Contains the information about a particular payment	Each payment represents a single occurrence. <ul style="list-style-type: none"> • One payment can only be made by 1 renter • One payment can only be made for 1 reservation
Review	Contains the information about a particular review	Each review represents a single occurrence. <ul style="list-style-type: none"> • One review can only be written by 1 renter • One review can have 0 to many sub-reviews • One sub-review can have only 1 parent review

Description of Attributes

Entity Name	Attributes	Description	Data Type	Nulls	Multi-valued	Derived	Default
HOST	<u>HOST_ID</u>	Unique ID for each host	INTEGER	None	None	None	None
	FIRST_NAME	First name of the host	VARCHAR(30)	None	None	None	None
	LAST_NAME	Last name of the host	VARCHAR(30)	None	None	None	None
	PHONE	Phone number of the host	VARCHAR(12)	None	None	None	None
	EMAIL	Email of the host	VARCHAR(30)	None	None	None	None
	STREET_ADDRESS	Street Address of the host	VARCHAR(70)	None	None	None	None
	CITY	City of the host	VARCHAR(30)	None	None	None	None
	PROVINCE	Province of the host	CHAR(2)	None	None	None	None
	POSTAL_CODE	Postal code of the host	VARCHAR(6)	None	None	None	None
	COUNTRY	Country of the host	VARCHAR(30)	None	None	None	None
PROPERTY	<u>PROPERTY_ID</u>	Unique ID for each property	INTEGER	None	None	None	None
	TYPE	Type of property (mansion, cottage, house, etc.)	VARCHAR(30)	None	None	None	None
	TERRAIN	Type of terrain (grass, sand, dirt, etc.)	VARCHAR(30)	None	None	None	None
	SIZE	Size of the property in sqft	INTEGER	None	None	None	None
	STREET_ADDRESS	Street Address of the property	VARCHAR(70)	None	None	None	None
	CITY	City of the property	VARCHAR(30)	None	None	None	None
	PROVINCE	Province of the property	CHAR(2)	None	None	None	None
	POSTAL_CODE	Postal code of the property	VARCHAR(6)	None	None	None	None
	COUNTRY	Country of the property	VARCHAR(30)	None	None	None	None
	AMENITIES_ID	Unique ID for each amenity	INTEGER	None	None	None	None
AMENITIES	NAME	Name of the amenity	VARCHAR(30)	None	None	None	None
	QUANTITY	Quantity available of the amenity	INTEGER	None	None	None	None
	DESCRIPTION	Description of the amenity	VARCHAR(100)	None	None	None	None
	PROPERTY_ID	Unique ID for each property	INTEGER	None	None	None	None
INCLUDED	AMENITIES_ID	Unique ID for each amenity	INTEGER	None	None	None	None

Entity Name	Attributes	Description	Data Type	Nulls	Multi-valued	Derived	Default
LISTING	<u>LISTING_ID</u>	Unique ID for each listing	INTEGER	None	None	None	None
	NAME	Name of the listing	VARCHAR(70)	None	None	None	None
	HOURLY_PRICE	Price/hour of renting the property in the listing	INTEGER	None	None	None	None
	DATE_LISTED	The date the listing was listed (posted) on the website.	DATETIME	None	None	None	None
	DATE_AVAILABLE	The date as of which the listing can accept visitors.	DATETIME	None	None	None	None
	DURATION	The maximum duration of stay (in hours) allowed.	INTEGER	None	None	None	None
RENTER	<u>RENTER_ID</u>	Unique ID for each renter	INTEGER	None	None	None	None
	FIRST_NAME	First name of the renter	VARCHAR(30)	None	None	None	None
	LAST_NAME	Last name of the renter	VARCHAR(30)	None	None	None	None
	PHONE	Phone number of the renter	VARCHAR(12)	None	None	None	None
	EMAIL	Email of the renter	VARCHAR(30)	None	None	None	None
	STREET_ADDRESS	Street Address of the renter	VARCHAR(70)	None	None	None	None
	CITY	City of the renter	VARCHAR(30)	None	None	None	None
	PROVINCE	Province of the renter	CHAR(2)	None	None	None	None
	POSTAL_CODE	Postal code of the renter	VARCHAR(6)	None	None	None	None
	COUNTRY	Country of the renter	VARCHAR(30)	None	None	None	None
RESERVATION	<u>RESERVATION_ID</u>	Unique ID for each reservation	INTEGER	None	None	None	None
	START_DATE	The start date and time of the reservation	DATETIME	None	None	None	None
	END_DATE	The end date and time of the reservation	DATETIME	None	None	None	None
	HOURS	The number of hours the reservation lasts	INTEGER	None	None	Yes *	None

* HOURS = END_DATE - START_DATE (all from the RESERVATION TABLE)

Entity Name	Attributes	Description	Data Type	Nulls	Multi-valued	Derived	Default
PAYMENT	PAYMENT_ID	Unique ID for each payment	INTEGER	None	None	None	None
	TOTAL	The total payment made by the renter for the reservation	INTEGER	None	None	Yes **	None
	TYPE	The type of payment (credit, debit, PayPal)	VARCHAR(30)	None	None	None	None
	DATE	The date and time the payment was made	DATETIME	None	None	None	None
REVIEW	REVIEW_ID	Unique ID for each review	INTEGER	None	None	None	None
	SCORE	The score out of 5 stars given by the reviewer	INTEGER	None	None	None	None
	DESCRIPTION	The description in their review	VARCHAR(100)	None	None	None	None
	DATE	The date and time the review was posted	DATETIME	None	None	None	None
	LIKES	The number of likes the review has received	INTEGER	None	None	None	None
	SUB REVIEW	Reference to the unique ID of a sub-review, which is a review that answers to the primary review (current REVIEW_ID)	INTEGER	Yes ***	None	None	None

** PAYMENT = HOURLY_PRICE (from table LISTING) * HOURS (from table RESERVATION)

*** Not all reviews have sub-reviews, hence why it is null.

Note: We chose to not include the foreign keys in the data dictionary to avoid redundancy. We only still included:

- the PROPERTY_ID & AMENITY_ID (PK/FKs) attributes in the INCLUDED table
- the SUB REVIEW (PK turned into FK because of a unary relationship) attribute from the REVIEW table

Because we believe they have some specificities that should be mentioned

SQL Statements

5 Functional SQL Statements

Functional statement description 1

Ref Number	01
Description	This query statement will update the hourly price of a listing
Situational context	A host wants to change the hourly price of one of his/her listings because of an increase of demand from renters
Situational Parameters	User is changing the hourly price on the listings screen

Output 1:

L00005 | Backyard with Undergrou | 250 | 2021-03-23 10:12 | 2021-04-01 10:12 | 6 | PR0005 | H00005 |

Functional statement description 2

Ref Number	02
Description	This query will delete a property from the database
Situational context	A host wants to remove one of his properties because of internal/external reasons (sold the property, not wishing to rent it anymore, etc.)
Situational Parameters	User selects a ‘property’ in the property’s table and clicks the ‘delete’ button

*** This actually takes three queries because of the relationships the property had with two other tables.

Output 2:

*Included Table

PROPERTY_ID	AMENITY_ID
PR0001	A00001
PR0001	A00002
PR0001	A00003
PR0001	A00004
PR0002	A00005
PR0003	A00005
PR0005	A00001
PR0005	A00002

*Listing Table

LISTING_ID	NAME	HOURLY_PRICE	DATE_LISTED	DATE_AVAILABLE	DURATION	PROPERTY_ID	HOST_ID
L00001	Beautiful Mansion backya	350	2022-05-21 12:45	2022-06-01 12:45	3	PR0001	H00001
L00002	Backyard Nature Cottage	215	2021-04-21 01:41	2021-05-01 01:41	6	PR0002	H00002
L00003	Seaside backyard perfect	250	2020-06-13 05:32	2020-07-01 05:32	5	PR0003	H00003
L00005	Backyard with Undergrou	250	2021-03-23 10:12	2021-04-01 10:12	6	PR0005	H00005

*Property Table

PROPERTY_ID	TYPE	TERRAIN	SIZE	STREET_ADDRESS	CITY	PROVINCE	POSTAL_CODE	COUNTRY	HOST_ID
PR0001	Mansion	Grass	18644	9106 Cherry Road	Wallaceb	ON	N8A9P9	Canada	H00001
PR0002	Cottage	Field	15067	29 Williams Avenue	Outer Nui	NU	X0A1H7	Canada	H00002
PR0003	Seaside	Sand	8033	88 Jones St.	Saint-Bru	QC	J3V4X8	Canada	H00003
PR0005	House	Grass	7002	32 Sierra St.	LAssomp	QC	J5W2A5	Canada	H00005

Functional statement description 3

Ref Number	03
Description	The query returns the total review count included on a listing
Situational context	A host wants to know the total amount of reviews on certain listings to see if their listings have positive reactions from renters
Situational Parameters	User is on the listings webpage and wishes to see the total amount of reviews on a listing

Output 3:

Output

2,

Functional statement description 4

Ref Number	04
Description	The query will sign up a renter on the website, but they will not have made a transaction yet
Situational context	A user wants to sign up to the website as a renter
Situational Parameters	The user will input all of their information in the correct sections and click on sign up, which will create their account

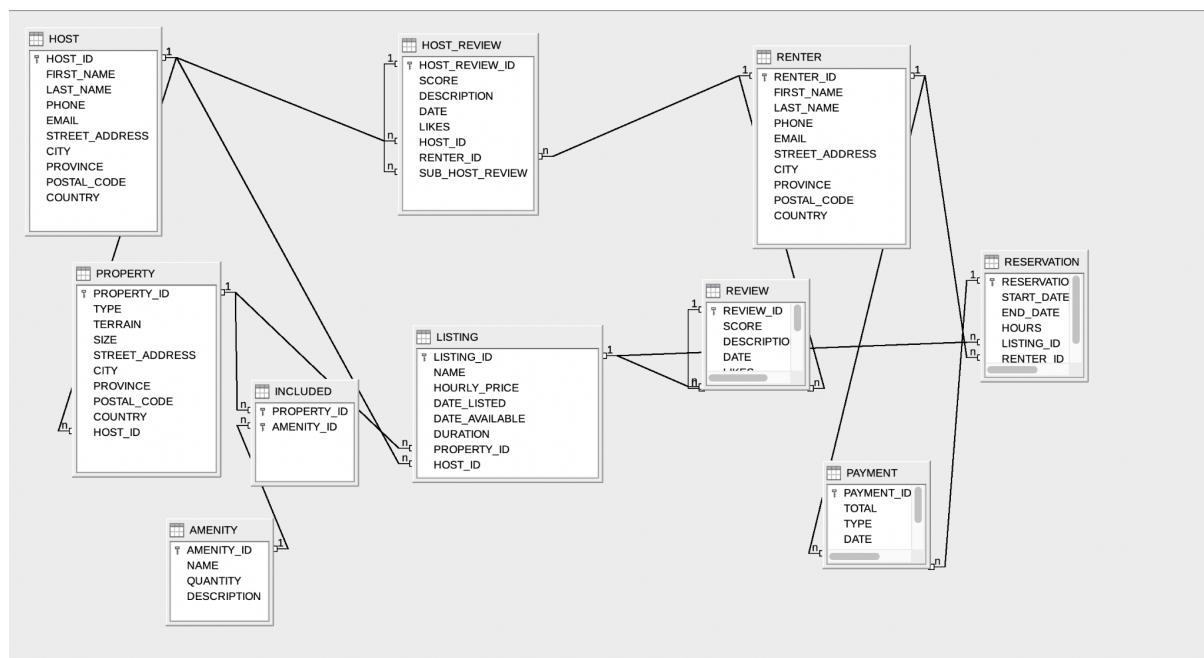
Output 4:

RENTER_ID	FIRST_NAME	LAST_NAME	PHONE	EMAIL	STREET_ADDRESS	CITY	PROVINCE	POSTAL_CODE	COUNTRY
FM0003	Felipa	Maria	4386770997	felipaaaa2@8075 Illinois Rd.	Digby	Ne	NS	B0V6M8	Canada
FR0004	Fanny	Ryer	5149325548	ryerfanfan@583 Sunset Street	Kimberley	BC		V1A5G6	Canada
FR0005	Flynn	Rider	4387201543	theygotmyr7836 E. Union Street	Beresford	NB		E8K7K7	Canada
KS0001	Konstantin	Stankovic	5146274343	pepelover@8754 Kingston St.	Aylmer	ON		N5H5K2	Canada
MJ0006	Michael	Jordan	5147289042	goat@gma40 Johnson Road	McAdam	NB		E6J5J8	Canada
YM0002	Yuliya	Mihov	4386341298	yuliyaflowe337 La Sierra St.	Pembroke	ON		K8B0L0	Canada

Functional statement description 5

Ref Number	05
Description	The query will allow the addition of a new table that will keep track of reviews made by hosts about renters that have made reservations on their property
Situational context	As our business expands, it would be pertinent to have the hosts review renters. This way, renters have a higher incentive to respect the property, and hosts can comment on how clean, respectful and compliant renters were to help future hosts decide who to rent out to
Situational Parameters	The database manager will create a new functionality that allows hosts to review renters and store the results in a separate table

Output 5:



10 informational SQL Statements descriptions

Informational statement description 1

Ref Number	01
Description	Determine which type of properties tend to fetch the highest price
Situational context	This information will allow us to market our service to individuals with highly valued properties

Output 1:

TYPE	AVERAGE_PRICE
Mansion	350
Seaside	250
Cottage	215
House	190

Informational statement description 2

Ref Number	02
Description	Most common amenities at each property in each province
Situational context	Will allow us to determine unique amenities that could make a property stand out

Output 2:

NAME	PROVINCE	NUMBEROFAMENITY
Pool	ON	1
Pool	QC	1
BBQ	ON	1
BBQ	NB	1
BBQ	QC	1
Gazebo	ON	1
Swing	ON	1
Swing	NB	1
Hammock	NU	1
Hammock	QC	1

Informational statement description 3

Ref Number	03
Description	Most common payment type
Situational context	Offering too many payment types can be expensive as operators charge fees. Thus, we can eliminate the least common payment method

Output 3:

TYPE	METHODCOUNT
Debit	2
Credit	2
Paypal	1

Informational statement description 4

Ref Number	04
Description	Top performing listings that have above average review
Situational context	Listings that outperform are critical to our service, thus we can offer incentives to keep them on Capifun.

Output 4:

NAME	SCORE
Beautiful Mansion backyard for rent	5
Backyard with Underground Pool	5

Informational statement description 5

Ref Number	05
Description	Find all new listings from the past 2 years that are less than \$1250 in total
Situational context	Identifies whether there has been growth in the number of new low cost listings in the past 2 years

Output 5:

Num_New_Listings	
2	

Informational statement description 6

Ref Number	06
Description	Most relevant reviews: find reviews with more than 5 likes that mention key words such as “bathroom”, “pool”, “amenities”, “terrain”
Situational context	This information tells Capifun which features are most important in properties and can help hosts advertise their properties

Output 6:

REVIEW_ID	DESCRIPTION	LIKES
C00003	No bathroom acc	14
C00002	Not bad, but wish	10

Informational statement description 7

Ref Number	07
Description	Total Properties by Country
Situational context	Identify markets that are potentially underserved in terms of reservations

Output 7:

COUNTRY	NUMBEROFPROPERTIES
Canada	5

Informational statement description 8

Ref Number	08
Description	Relationship between review length and likes
Situational context	Capifun will uncover whether longer reviews have more likes and can use this information to set word limits on reviews

Output 8:

NUM_WORDS	NUM_REVIEWS	AVG_LIKES
33	1	5
43	1	10
18	1	14
32	1	12
82	1	22

Informational statement description 9

Ref Number	09
Description	Host to Renter Ratio by Province
Situational context	High ratios could indicate an oversupply of properties in certain provinces and would thus influence the marketing strategy

Output 9:

PROVINCE	RATIO
ON	1
NU	1
QC	1

Informational statement description 10

Ref Number	10
Description	Undervalued properties: display listings that are charging below average rate for property of same type
Situational context	Helps hosts price match as well as ensuring higher revenue for Capifun

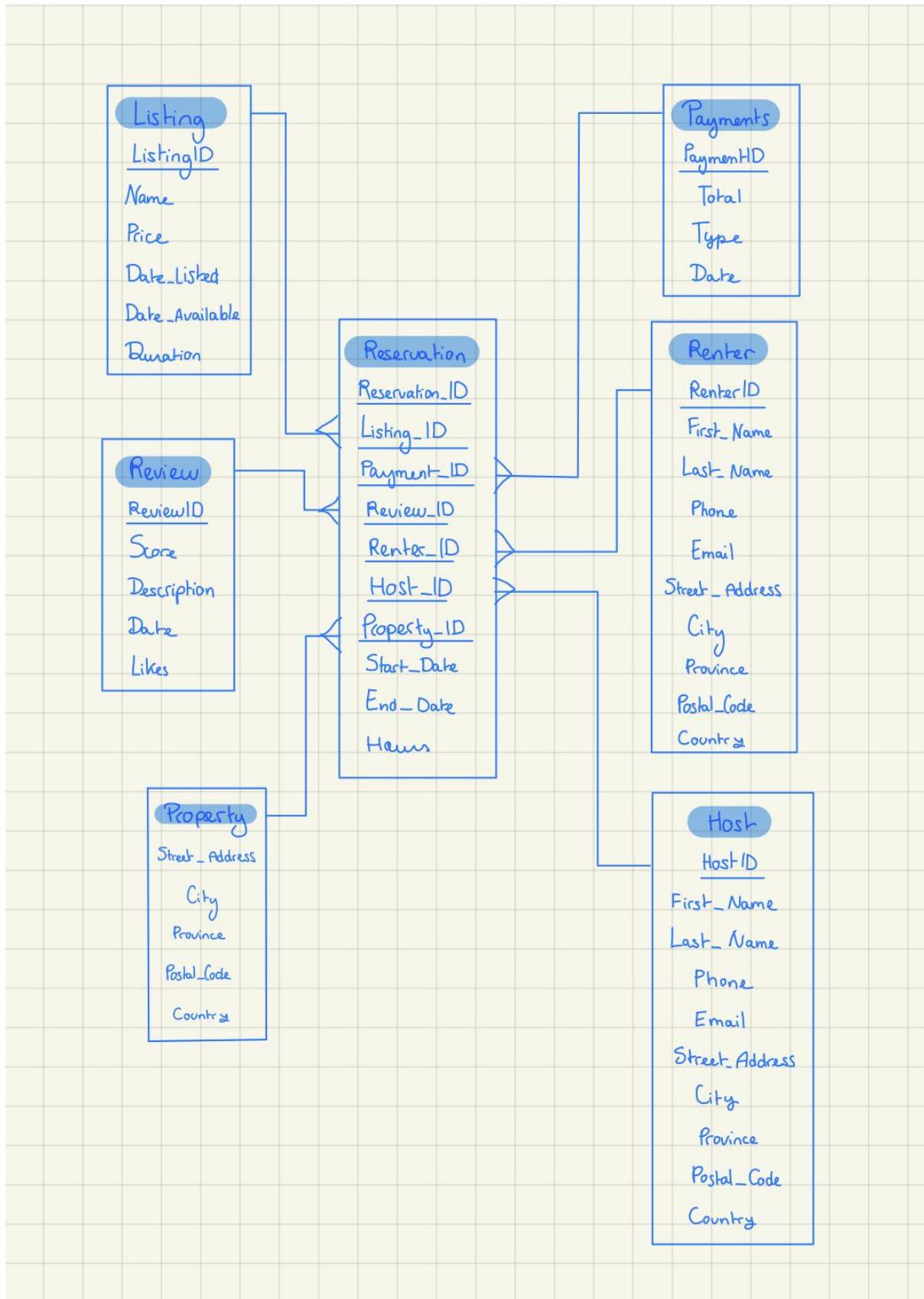
Output 10:

NAME	HOURLY_PRICE	TYPE
Best backyard for a BBQ	180	House

Mddb design table

Prompt	Atomic Example	Data sources
Outcome?	Reservation, review, and payment. A successful reservation made by a guest for a listing. Feedback left by a guest about their experience staying at a listing. A transaction between a renter and Capifun that covers the cost of a reservation.	LISTING, RESERVATION, PAYMENTS, REVIEW, HOST, and RENTER tables
When?	A time during which a listing is available for guests to book. Reservation (StartDate, EndDate)	LISTING and RESERVATION tables
Where?	The geographic location of a property from a listing.	PROPERTY table
What?	A listing that a host lists on the platform for guests to book	LISTING table
How?	Payment method	PAYMENTS table
Who?	Host and Renter	RENTER and HOST tables

Star Model



Metadata Dictionary

Name	Start_Date
Description	The date and time of the start of the reservation
Purpose	To track the start of a reservation
Formula	Start_Date
Data Sources	Reservations

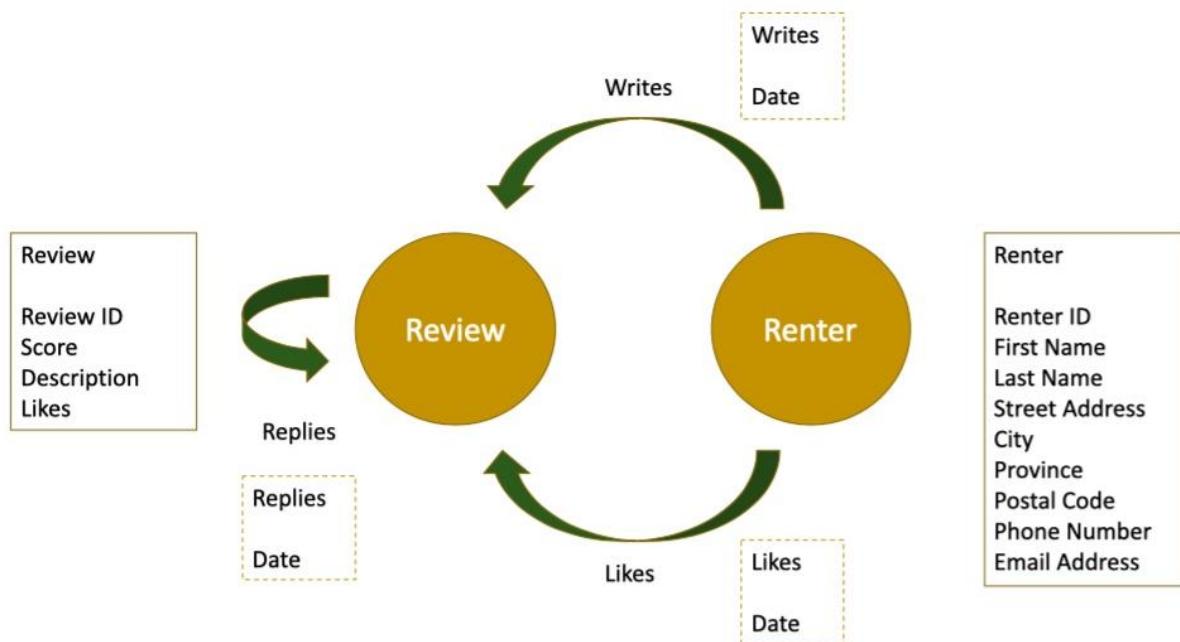
Name	End_Date
Description	The end date and time of the reservation
Purpose	To track the end of a reservation
Formula	End_Date
Data Sources	Reservations

Name	Hours
Description	The length of the reservation
Purpose	To track the length of a reservation
Formula	START_DATE - END_DATE
Data Sources	Reservations

Nodes and relationship diagram

The sample graph illustrates the unary relationship of the “Review” entity and its relationships with the “Renter” entity: a renter can write a review, a renter can like a review, and a review can be a reply to another review. All of these relationships showcase the date implemented. These relationships can be complex in a relational model. In a traditional model, it would be hard to visualize and understand the concept of sub-reviews. In addition, it can be hard to filter all of these different relationships between “renter” and “review” entities. Hence, a Graph database has a less rigid schema that permits more straightforward evolution and can help write efficient queries that do not require the JOIN functions and uncover hidden connections between concepts that would not be apparent in traditional databases. Overall, it can generate value as it is easier to analyze and conceptualize how attributes are interconnected.

Figure 1: Nodes and relationship diagram



Nodes and Diagram Relationship

Table 1: WRITES relation with sample data

WRITES_ID	RENTER_ID	REVIEW_ID	Date
WR0001	KS0001	C00001	2022-06-22
WR0002	YM0002	C00002	2021-05-22
WR0003	FM0003	C00003	2020-07-14
WR0004	FR0004	C00004	2019-06-23
WR0005	FR0005	C00005	2021-04-24

Table 2: LIKES relation with sample data

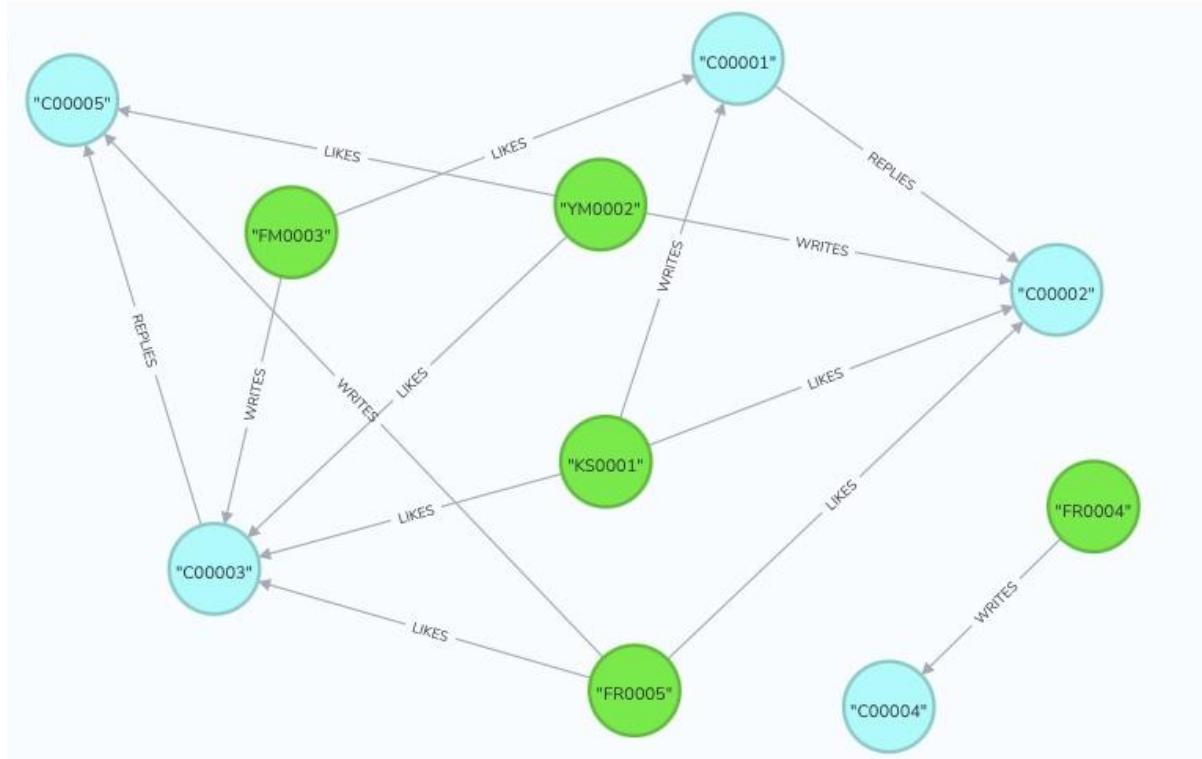
LIKES_ID	RENTER_ID	REVIEW_ID	Date
LI0001	KS0001	C00003	2021-06-29
LI0002	KS0001	C00002	2021-07-24
LI0003	YM0002	C00005	2021-08-13
LI0004	FM0003	C00001	2022-08-30
LI0005	FR0005	C00003	2020-09-11
LI0006	YM0002	C00003	2020-08-09
LI0007	FR0005	C00002	2022-07-15

Table 3: LIKES relation with sample data

REPLIES_ID	RENTER_ID	REVIEW_ID	Date
RE0001	YM0002	C00001	2022-7-12
RE0002	FR0005	C00003	2022-8-05

Sample Graph

Figure 2: Sample graph



2 cypher queries description

Table 12: Cypher query description

Ref Number	12
Description	This Query provides the number of likes of reviews with a rating that is below 3. The output shows the review ID, renter ID, first name, last name, and the number of likes, with a descending order by the latter.
Value	Low ratings are often associated with low quality. CapiFun can use the former as a benchmark to analyze aspects that make a backyard unsatisfactory and ban those that receive too many negative reviews. However, due to competition, it is common for people to write fake low reviews for their competitors in order to damage their reputation and gain a competitive advantage. Hence, the number of likes helps to determine the trustworthiness of the review.

Figure 3: Neo4j Table output

\$ MATCH (R1:RENTER) - [W:WRITES] → (C:REVIEW) ← [L:LIKES] - (R2:RENTER) WHERE C.SCORE ≤ 3 RETURN DISTINCT				
Table RAW Download				
C.REVIEW_ID	R1.RENTER_ID	R1.FIRST_NAME	R1.LAST_NAME	NUMBER_OF_LIKES
"C00002"	"YM0002"	"Yuliya"	"Mihov"	2
"C00003"	"FM0003"	"Felipa"	"Maria"	3

Showing 1-2 of 2 results Show 50

Table 13: Cypher query description

Ref Number	13
Description	This Query provides the list of reviews where the date is after 2022
Value	Some reviews might be outdated, as backyards can change overtime. Hence, it is interesting to filter reviews of the most recent year. In this example, we want the query to show reviews after year 2022.

Figure 4: Neo4j Table output

\$ MATCH (C:REVIEW) ← [W:WRITES] - (R:RENTER) WHERE W.Date ≥ dat	
Table RAW Download	
C.REVIEW_ID	C.DESCRIPTION
"C00001"	"4 acres of distinguished greenery"

Showing 1 of 1 results Show 50