Kids Crew - A Child Care Service

Milestone: Project Report

Group 11

Student1 Sharayu Thosar
Student2 Pranjali Bhatt

602-849-37846 (Tel of Student 1)
908-340-2087 (Tel of Student 2)

thosar.sh@northeastern.edu
bhatt.pranj@northeastern.edu

Percentage of Effort Contributed by Student1: 50%
Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Sharayu Thosar
Signature of Student 2: Pranjali Bhatt

Submission Date: December 10th, 2022

USE CASE STUDY REPORT

Group No.: Group 11

Student Names: Sharayu Thosar and Pranjali Bhatt

Executive Summary:

Our aim is to create a database model for the company KidsCrew and gain insights into how to enhance the quality of service. The primary objective of this study was to design and implement a database solution for the company providing the childcare services.

The company provides families with children babysitting and pick up drop off services for children in the 1–10-year age group. Sitters looking for part time jobs can register themselves along with their availability. Inturn parents can find a perfect fit based on their budget and time constraints.

The database was modeled by taking requirements of every sitter, parent and child. First, the EER and UML diagrams were modeled, followed by the mapping of the conceptual model to a relational model with the required primary and foreign keys and Normalization. This database was then loaded on MySQL Workbench and various simple and complex queries were executed to gain insights. Additionally, Neo4j was used to visualize the graph databases and gain additional insights. A thorough analysis of the database was done by connecting it to Python. The data was intelligently queried and further visualized to reveal unknown patterns.

I. Introduction

While creating an application, it is important to have a system in place to record all the information during any interactions. For example, the KidsCrew app we are working on has two types of person who can register, the parents who are looking for babysitters and the babysitters who want to work part time. It is important to record the bank details, occupations, phone number and address of these people to enable smooth communications and payments during the service. To better improve the service, the experience of the babysitters that includes their rate, rating, number of hours logged etc., can be stored to monitor their performance. Lastly, creating a database to store all information can provide us actionable insights to improve the service.

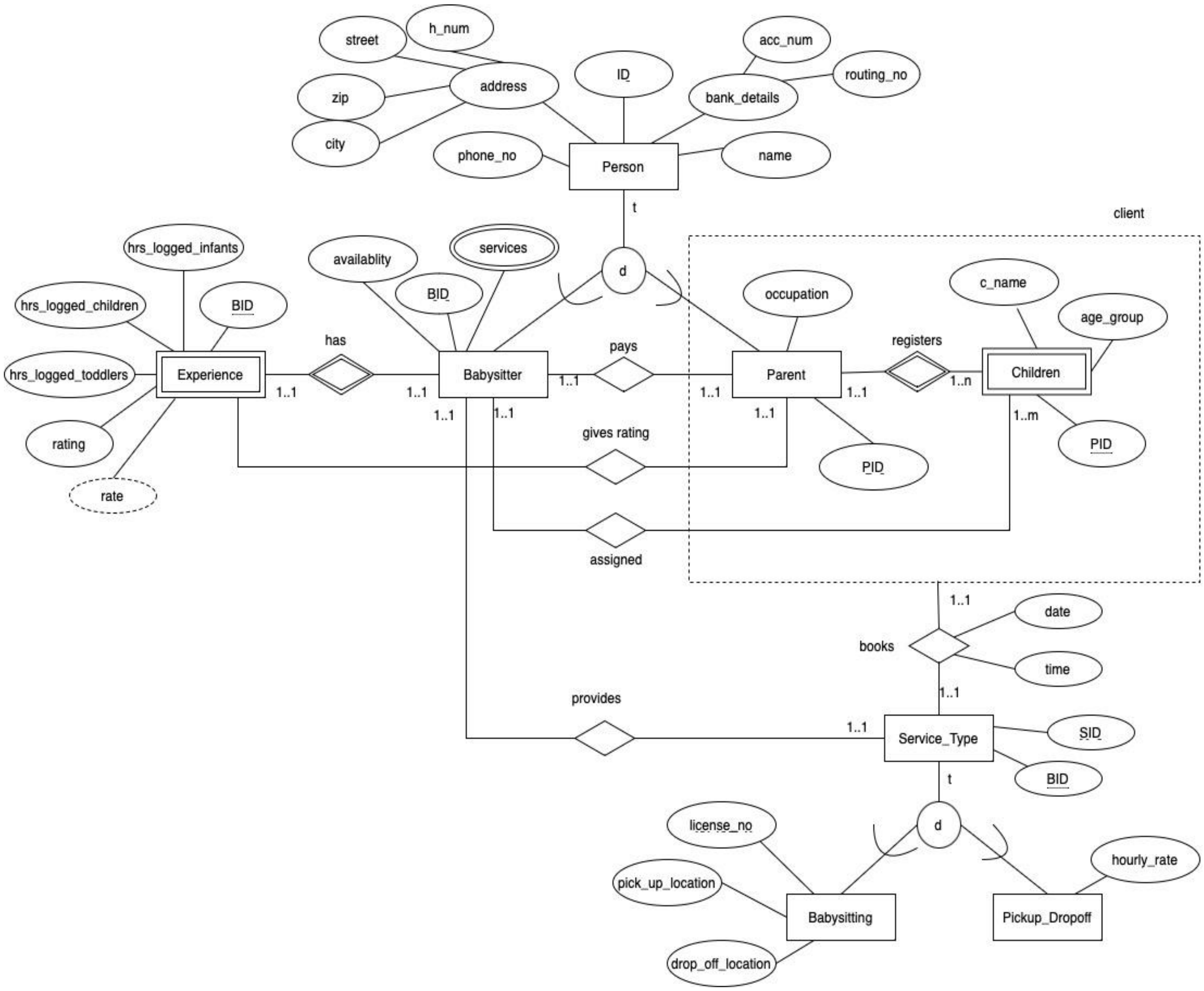
Hence, our goal here is to create a database for the company where it stores the information of all the parents who are looking for babysitters along with the information of their children such as their name and age group. The aggregation of the Parent and Children is our client and Babysitters are Service providers. Babysitters availability and service type is also recorded in the database to allocate perfect babysitters to the families. Additionally, the babysitters rate is decided based on the parents rating. All this data needs to be recorded in order to facilitate calculation of hourly rate. All of these entities combined, will provide a solid database that can be used to extract interesting insights.

Our project includes modeling the database as ER and UML model followed by relational model and then creating schema and populating data on MySQL. We have used MySQL, Neo4j and Python to analyze the data and get the insights.

II. Conceptual Data Modeling

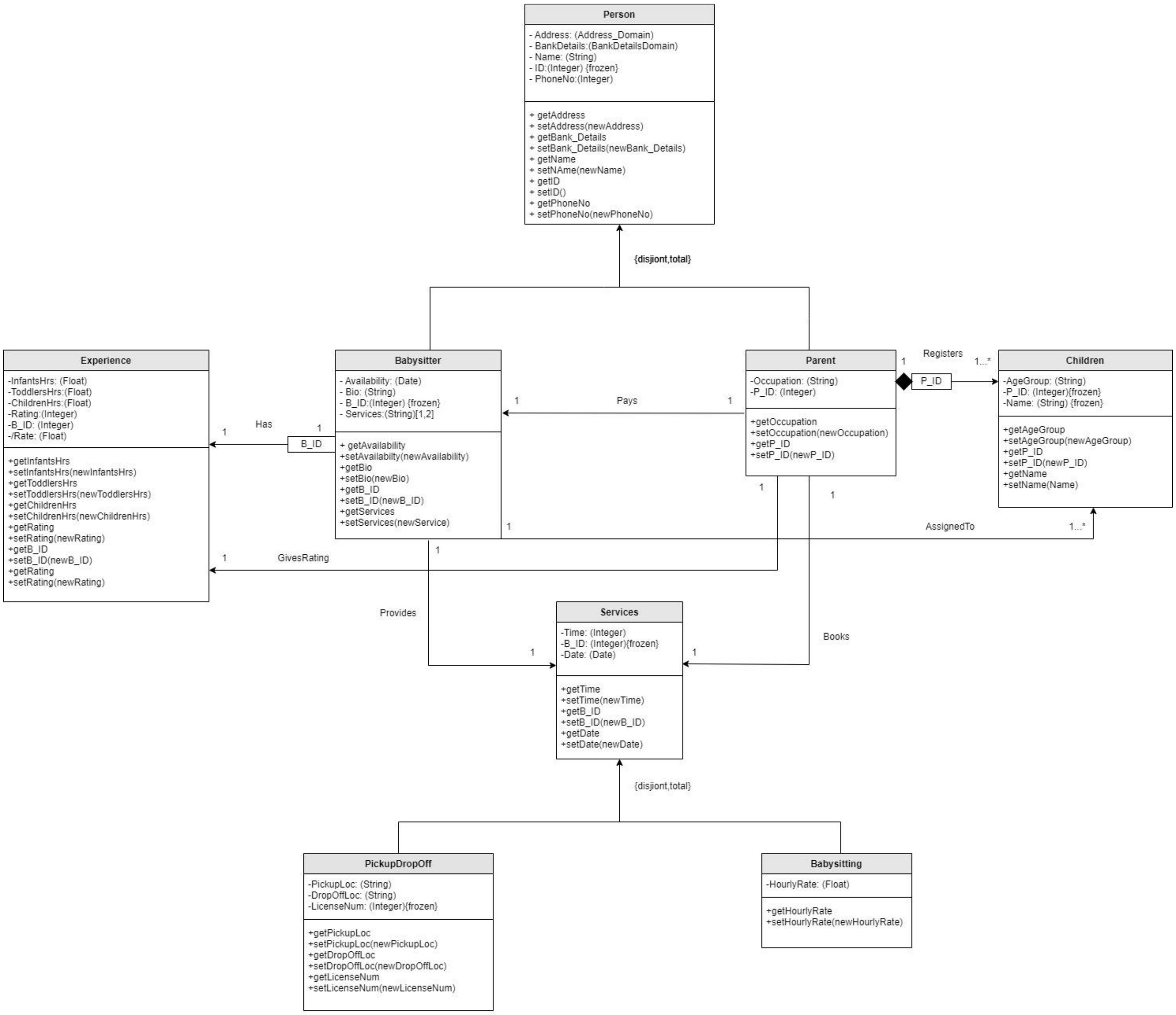
Following ER Diagram includes all the Entitites along with their attribute types. The ER model also uses total and disjoint specialization for Person and Service_Type and aggregation for the client with Parent and Children. It has also weak entity types like Children and Experience.

1. EER Model



2. UML Diagram:

UML diagram was modeled to overcome the limitations of ER model. Therefore, the OCL constraints are also included in the UML Diagram.



OCL Constraints

Context: Experience
invariant: self.rate >0

Context: Experience
invariant: self.rating >=1 AND self.rating<=5

Context: Person
invariant:self.PhoneNo->size()=10

Context: Children
invariant:self.AgeGroup = (Infants,Children,Toddlers)

III. Mapping Conceptual Model to Relational Model

Primary Key- Underlined **Foreign Key**- Italicized

Parent(PID, occupation, acc_num, p_name, phone_no, HNum, street,zip,city,routing_num, acc_num)

- PID is the primary key

Babysitter(BID,acc_num, name, phone_no, HNum, street,zip,city, availability,*PID,*routing_num, acc_num)

- BID is primary key
- Acc_num is a foreign key referring to ac_num in BankDetails;NOT NULL
- PID is foreign key referring to PID in Parent for the relation Pays;NOT NULL

ServiceType(*BID,*SID, Service)

- BID is a foreign key referring to BID in Babysitter ;NOT NULL
- SID, BID is the primary key

Experience(*BID*, hrs_logged_infants, hrs_logged_toddlers, hrs_logged_children, rating, rate, *PID*)

- PID is a foreign key referring to PID in Parent for relation gives rating;NOT NULL
- Experience is a weak entity. It refers to the BID in Babysitter. BID is the primary and foreign key

Client(PID,*BID, SID,* Name)

- PID, Name is primary key as it comes under aggregation of Parent and Children
- BID is a foreign key referring to BID in Babysitter;NOT NULL
- SID is foreign key referencing to SID in ServiceType as client books relation;NOT NULL

PickupDropOff(*BID*, license_no, pickup_loc, drop_off_loc)

- BID is the primary key
- BID is a foreign key referring to BID in Babysitter;NOT NULL
- License_no, BID are candidate keys

Book(*SID*, date, time, *BID*)

- SID is foreign key referencing to SID ;NOT NULL
- BID is a foreign key referring to BID in Babysitter.;NOT NULL
- BID,SID is primary key

Babysitting(hourly_rate,*BID*)

- BID is primary key
- BID is a foreign key referring to BID in Babysitter;NOT NULL

Parent(PID, occupation, acc_num, name, phone_no, HNum, street,zip,city)

- PID is the primary key

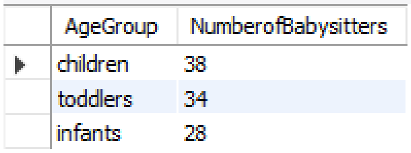
Children(Name, age_group, *PID,BID*)

- PID and Name is the primary key
- Children Entity is a weak entity. PID is a foreign key referring to Parent for registers relation;NOT NULL
- BID is foreign key referring to BID in Babysitter for assigned relation;NOT NULL
- **IV. Implementation of Relation Model via MySQL and NoSQL**

MySQL Implementation:
The database was implemented on MySQL and the following queries were performed:

Query 1: Find the number of sitters available for each age group in descending order.

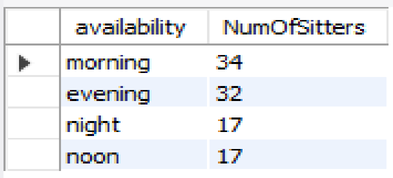
```
SELECT AGE_GROUP as AgeGroup,
count(BID) AS NumberofBabysitters
FROM kids_crew_new.children
GROUP BY age_group
ORDER BY NumberofBabysitters DESC;
```



	AgeGroup	NumberofBabysitters
▶	children	38
	toddlers	34
	infants	28

Query 2: Find the number of sitters as per availability in descending order.

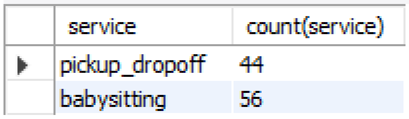
```
SELECT availability,count(availability)as
NumOfSitters
FROM babysitter
GROUP BY availability
ORDER BY NumOfSitters DESC;
```



	availability	NumOfSitters
▶	morning	34
	evening	32
	night	17
	noon	17

Query 3:Find the number of sitters for each type of service.

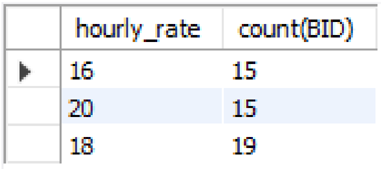
```
SELECT  service, count(service)
FROM service_type
GROUP BY service;
```



	service	count(service)
▶	pickup_dropoff	44
	babysitting	56

Query 4: Find the number of babysitters having an hourly rate greater than the average hourly rate.

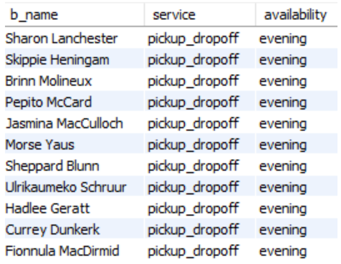
```
SELECT hourly_rate, count(BID)
FROM babysitting
GROUP BY hourly_rate
HAVING hourly_rate > (SELECT AVG(hourly_rate)
FROM babysitting);
```



	hourly_rate	count(BID)
▶	16	15
	20	15
	18	19

Query 5: Find the name of all babysitters available for pickup/dropoff in the evening.

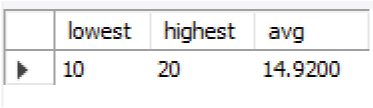
```
SELECT b1.b_name, s.service,b1.availability
FROM babysitter b1, service_type s
WHERE b1.bid = s.bid  AND service= 'pickup_dropoff'
AND b1.availability = 'evening';
```



b_name	service	availability
Sharon Lanchester	pickup_dropoff	evening
Skippie Heningam	pickup_dropoff	evening
Brinn Molineux	pickup_dropoff	evening
Pepito McCard	pickup_dropoff	evening
Jasmina MacCulloch	pickup_dropoff	evening
Morse Yaus	pickup_dropoff	evening
Sheppard Blunn	pickup_dropoff	evening
Ulrikaumeko Schruur	pickup_dropoff	evening
Hadlee Geratt	pickup_dropoff	evening
Currey Dunkerk	pickup_dropoff	evening
Fionnula MacDirmid	pickup_dropoff	evening

Query 6: What is the lowest, highest, and average hourly rate of Sitters? (Aggregation)

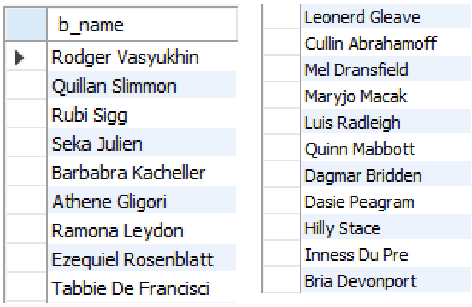
```
SELECT MIN(hourly_rate) as lowest,
MAX(hourly_rate) as highest, AVG(hourly_rate) as avg
FROM babysitting;
```



lowest	highest	avg
10	20	14.9200

Query 7: Find the name of the babysitters that have a rating greater than 3 and serviced children for over 50 hrs. (Nested Query)

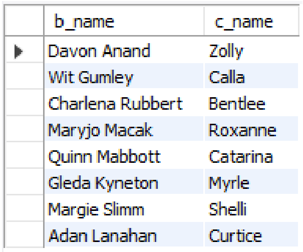
```
SELECT b_name
FROM babysitter
WHERE bid IN
(SELECT bid
 FROM experience
 WHERE rating >3
 AND hrs_logged_children >50);
```



b_name
Rodger Vasyukhin
Quillan Slimmon
Rubi Sigg
Seka Julien
Barbabra Kacheller
Athene Gligori
Ramona Leydon
Ezequiel Rosenblatt
Tabbie De Francisci
Leonerd Gleave
Cullin Abrahamoff
Mel Dransfield
Maryjo Macak
Luis Radleigh
Quinn Mabbott
Dagmar Bridden
Dasie Peagram
Hilly Stace
Inness Du Pre
Bria Devonport

Query 8: Find the name of all babysitters that are available in the morning along with the child assigned to them whose parent is a Surgeon. (Join)

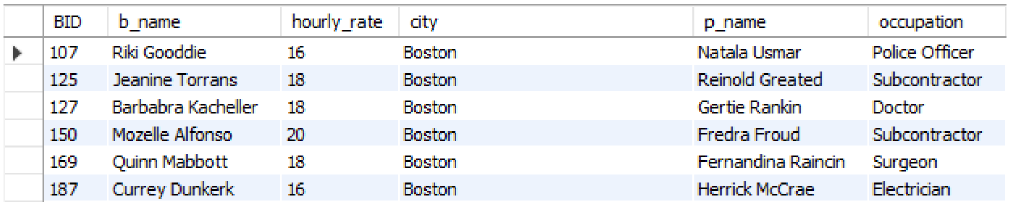
```
SELECT b.b_name, c.c_name
FROM children c, babysitter b, parent p
WHERE c.BID = b.BID AND p.PID = c.PID
AND b.availability='morning'
AND p.occupation='Surgeon';
```



b_name	c_name
Davon Anand	Zolly
Wit Gumley	Calla
Charlena Rubbert	Bentlee
Maryjo Macak	Roxanne
Quinn Mabbott	Catarina
Gleda Kyneton	Myrle
Margie Slimm	Shelli
Adan Lanahan	Curtice

Query 9: Find the name, ID of the babysitter having an hourly rate greater than average in boston along with parent name, and their occupation.

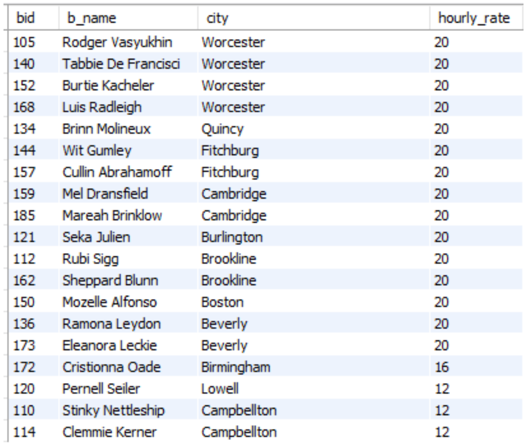
```
SELECT b.BID, b.b_name, bs1.hourly_rate, b.city, p.p_name,P.occupation
FROM babysitter b, babysitting bs1, parent p
WHERE b.BID=bs1.BID AND b.PID = p.PID AND P.occupation IN (SELECT
occupation FROM parent WHERE city='boston') AND bs1.hourly_rate > (SELECT
AVG(bs2.hourly_rate) FROM babysitting bs2) AND b.city='boston';
```



BID	b_name	hourly_rate	city	p_name	occupation
107	Riki Gooddie	16	Boston	Natala Usmar	Police Officer
125	Jeanine Torrans	18	Boston	Reinold Greated	Subcontractor
127	Barbabra Kacheller	18	Boston	Gertie Rankin	Doctor
150	Mozelle Alfonso	20	Boston	Fredra Froud	Subcontractor
169	Quinn Mabbott	18	Boston	Fernandina Raincin	Surgeon
187	Currey Dunkerk	16	Boston	Herrick McCrae	Electrician

Query 10: Find the ID,Name, city, and Hourly Rate of all babysitters who have the highest salary of all babysitters located in the same city. (View, Join, ALL Operator)

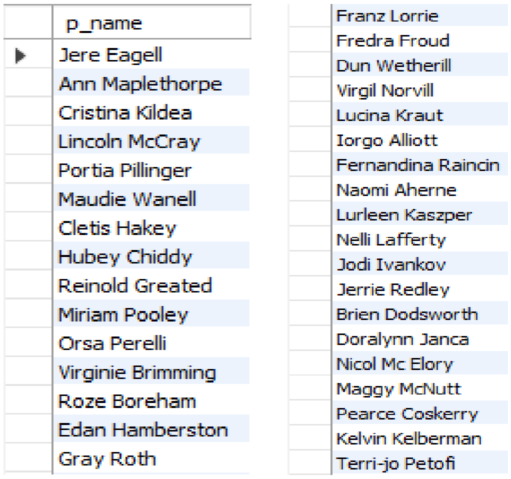
```
CREATE VIEW babysitter_info as
SELECT b1.bid, b1.b_name,
b1.city,bs.hourly_rate
FROM babysitter b1, babysitting bs
WHERE b1.bid = bs.bid;

SELECT b1.bid,b1.b_name,b1.city,
b1.hourly_rate
FROM babysitter_info b1
WHERE b1.hourly_rate >=
ALL( SELECT b2.hourly_rate
        FROM babysitter_info b2
        WHERE b2.city=b1.city)
ORDER BY b1.hourly_rate DESC, city
DESC;
```



bid	b_name	city	hourly_rate
105	Rodger Vasyukhin	Worcester	20
140	Tabbie De Francisci	Worcester	20
152	Burtie Kacheler	Worcester	20
168	Luis Radleigh	Worcester	20
134	Brinn Molineux	Quincy	20
144	Wit Gumley	Fitchburg	20
157	Cullin Abrahamoff	Fitchburg	20
159	Mel Dransfield	Cambridge	20
185	Mareah Brinklow	Cambridge	20
121	Seka Julien	Burlington	20
112	Rubi Sigg	Brookline	20
162	Sheppard Blunn	Brookline	20
150	Mozelle Alfonso	Boston	20
136	Ramona Leydon	Beverly	20
173	Eleanora Leckie	Beverly	20
172	Cristionna Oade	Birmingham	16
120	Pernell Seiler	Lowell	12
110	Stinky Nettleship	Campbellton	12
114	Clemmie Kerner	Campbellton	12

Query 11: Find the names of all parents that need babysitters in the morning.

```
SELECT p.p_name
FROM parent p
WHERE EXISTS
(SELECT *
FROM babysitter b
WHERE b.PID=p.PID
AND b.availability='morning');
```

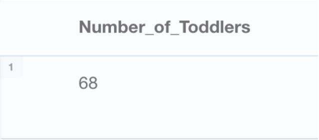


p_name
Jere Eagell
Ann Maplethorpe
Cristina Kildea
Lincoln McCray
Portia Pillinger
Maudie Wanell
Cletis Hakey
Hubey Chiddy
Reinold Greated
Miriam Pooley
Orsa Perelli
Virginie Brimming
Roze Boreham
Edan Hamberston
Gray Roth
Franz Lorrie
Fredra Froud
Dun Wetherill
Virgil Norvill
Lucina Kraut
Iorgo Alliott
Fernandina Raincin
Naomi Aherne
Lurleen Kaszper
Nelli Lafferty
Jodi Ivankov
Jerrie Redley
Brien Dodsworth
Doralynn Janca
Nicol Mc Elory
Maggy McNutt
Pearce Coskerry
Kelvin Kelberman
Terri-jo Petofi

NoSQL Implementation:
The Children and Parent were loaded in the Neo4j and following Cypher queries were executed.

Query 1: Count the number of children that belong to age group ‘toddlers’

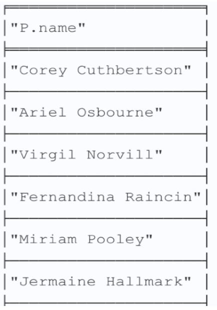
```
MATCH (n:children)-[:belongs]-(a:AgeGroup)
WHERE a.age='toddlers'
RETURN count(n) as Number_of_Toddlers
```



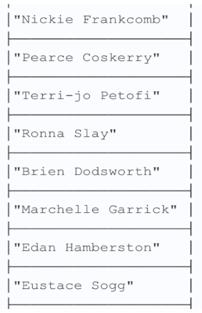
	Number_of_Toddlers
1	68

Query 2: Display the Parents whose occupation is Surgeon

```
MATCH (P:parent)-[r:IS_A]->(O:occupation)
WHERE O.occupation='Surgeon'
RETURN P.name
```



```
"P.name"
"Corey Cuthbertson"
"Ariel Osbourne"
"Virgil Norvill"
"Fernandina Raincin"
"Miriam Pooley"
"Jermaine Hallmark"
```

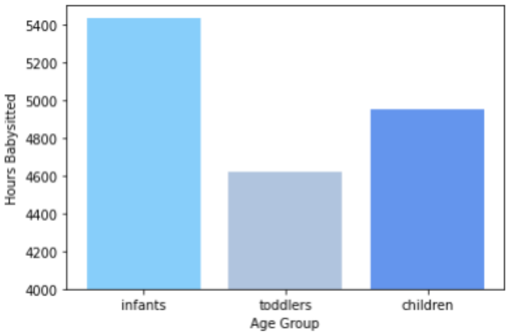


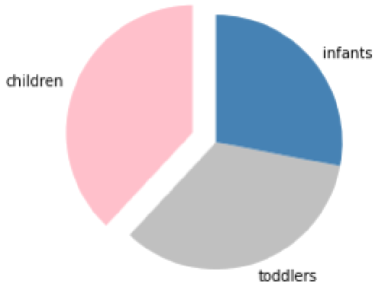
```
"Nickie Frankcomb"
"Pearce Coskerry"
"Terri-jo Petofi"
"Ronna Slay"
"Brien Dodsworth"
"Marchelle Garrick"
"Edan Hamberston"
"Eustace Sogg"
```

V. Database Access via R or Python

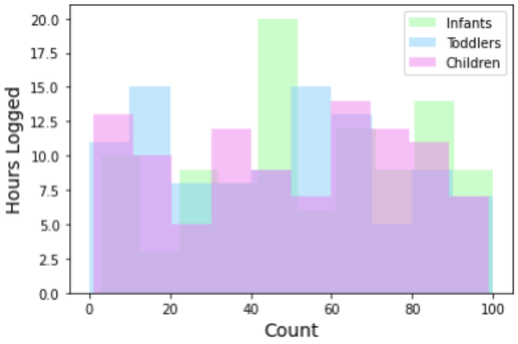
The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.excecute to run and fetchall from query, followed by converting the list into a dataframe using pandas library and using matplotlib to plot the graphs for the analytics.

Analysis - 1 : From the pie chart and bar chart it is concluded that in spite of high demand of children, the Babysitters logged most amount of hours for Infants.

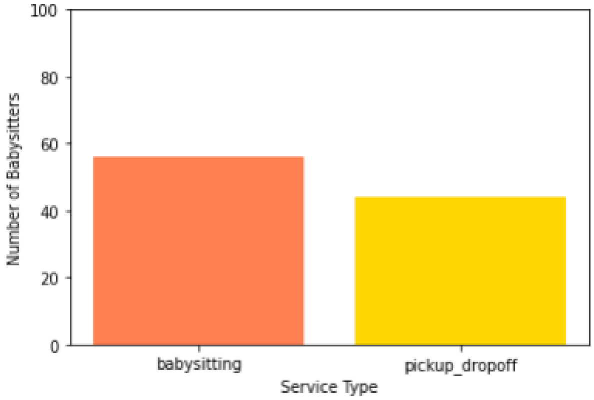




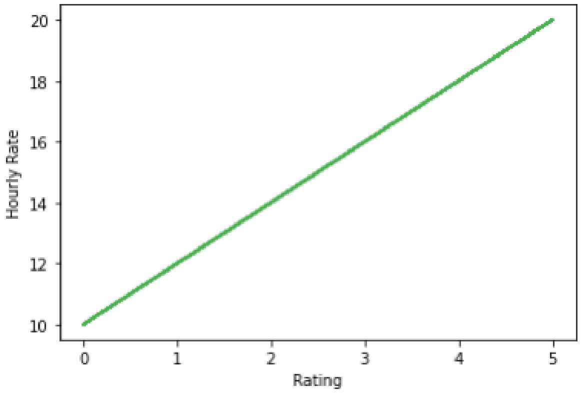
Analysis - 2 : Histogram of hours logged for Age group



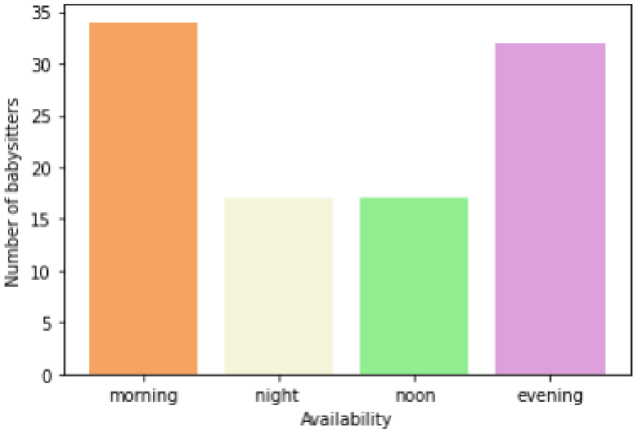
Analysis - 3: Number of sitters for pickup/drop off service are less compared to those who provide babysitting services



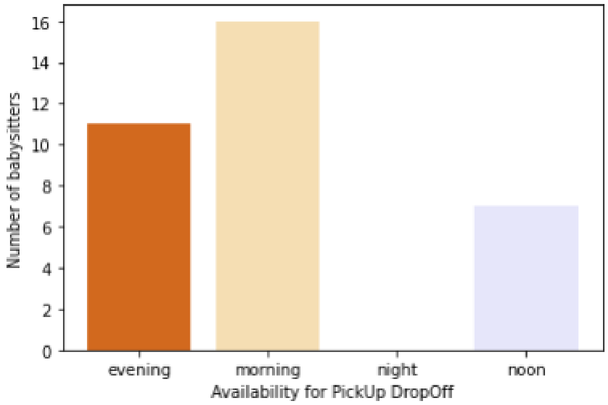
Analysis - 4 : Hourly rate of the sitter increases linearly with the rating they receive



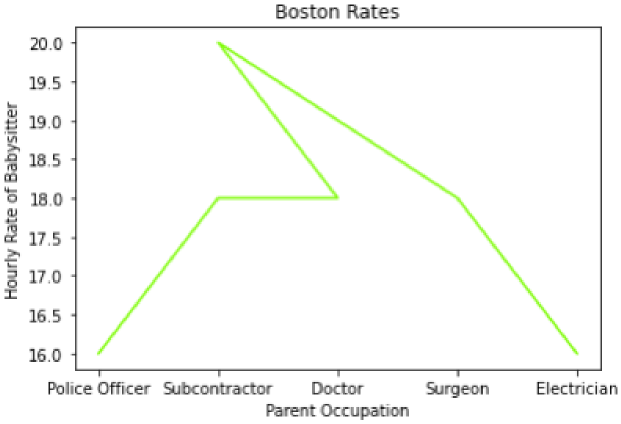
Analysis - 5 : Bar chart depicting sitters availability for babysitting service



Analysis - 6 : Bar chart depicting sitters availability for pickup/dropoff service



Analysis -7 : Hourly rate variation of babysitters with Parent occupation in Boston



VII. Summary and recommendation

The need of sitters is very high however finding the right match for a particular family and their needs is a hassle. This database solution aims to map babysitters to children in the most efficient way. The current design of the database is simple to understand and analyze characteristics of babysitters and families. It can be implemented in a real world scenario to become the next go to solution for families to find babysitters.

Improvements could be done by extracting real time data to enhance the quality of the database. Modifying the service type attribute type to allow sitters to provide more than one service.

The current database is a snapshot of a one day data. We can further increase the complexity of the database by including more data in real time. A tableau dashboard can be used to visualize the data and draw meaningful insights.