

Preppin Data 2022-WK25: Housing Happy Hotel Guests



Table of Content

1. [Link to challenge](#)
2. [Requirements](#)
3. [Screenshot solution](#)
4. [Notebook cells](#)
5. [Final output](#)
6. [Reference materials](#)

Link to challenge

<https://preppindata.blogspot.com/2022/06/2022-week-25-housing-happy-hotel-guests.html>

Requirements

- Input the data
- Before we bring the 2 datasets together, we want to know how many Additional Requests each guest has made
 - Update *N/A* values to *null* and make sure this counts as 0 Additional Requests
- Match the guests to the rooms which have capacity for their entire party
- Filter so that double/twin bed preferences are adhered to
- Ensure guests who have accessibility requirements are only matched with accessible rooms
- Calculate the Request Satisfaction % for each room
- Filter so that guests are only left with rooms with the highest Request Satisfaction %
- Finally, for the rooms with the largest capacity, we want to ensure guests with larger parties are prioritised. Filter the data to remove parties that could fit into smaller rooms
- Output the data

Screenshot solution

The below is a screenshot of what the solution looks like in the notebook. You can view and download the final output [here](#). Check my Github repository for the compiled SQL and a copy of the notebook markdown.

all_data_hotelrooms

```
SELECT *
FROM preppindata.`2022w25_hotelrooms`;
```

Untitled

Room	Adults	Children	Features
205	2	(Empty)	Double
305	2	(Empty)	Double
203	2	1	Double
208	2	1	Double
306	2	1	Double

🕒 4 columns · 40 rows

🕒 0.7 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)**all_data_guests**

```
SELECT *
FROM preppindata.`2022w25_guests`;
```

Untitled

Party	Adults	Children	Double_Twin	Requires_Accessible_Room	Additional_Requests
Fallens	1	0	Twin	false	N/A
Marusik	1	0	Twin	false	Bath, High Floor
Lansdale	1	0	Twin	false	High Floor
Jeannel	1	0	Twin	false	High Floor, NOT Near to lift
Londer	1	0	Twin	false	High Floor, NOT Near to lift

🕒 6 columns · 30 rows

🕒 0.7 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)**testing_array****step_1**

```
-- clean guests table, replace N/A will null

SELECT
Party,
Adults,
Children,
Double_Twin,
Requires_Accessible_Room,
IF(Additional_Requests = "N/A", NULL, Additional_Requests) AS Additional_Requests
FROM preppindata.`2022w25_guests`;
```

Untitled

Party	Adults	Children	Double_Twin	Requires_Accessible_Room	Additional_Requests
Fallens	1	0	Twin	false	(Empty)
Marusik	1	0	Twin	false	Bath, High Floor
Lansdale	1	0	Twin	false	High Floor
Jeannel	1	0	Twin	false	High Floor, NOT Near to lift
Londer	1	0	Twin	false	High Floor, NOT Near to lift

🕒 6 columns · 30 rows

🕒 0.7 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)

step_2

```
-- counting the number of additional requests by guest

SELECT
Party,
Adults AS Adults_In_Party,
Children AS Children_In_Party,
Double_Twin,
Requires_Accessible_Room,
Additional_Requests,
IF(ARRAY_LENGTH(SPLIT(Additional_Requests,",")) IS NULL, 0, ARRAY_LENGTH(SPLIT(Additional_Requests,",")))
AS Nr_Requests
FROM step_1;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Fallens	1	0	Twin	false	(Empty)
Marusik	1	0	Twin	false	Bath, High Floor
Lansdale	1	0	Twin	false	High Floor
Jeannel	1	0	Twin	false	High Floor, NOT Near to lift
Londer	1	0	Twin	false	High Floor, NOT Near to lift

7 columns · 30 rows

0.7 seconds (41 minutes ago) Download CSV

Expand

step_3

```
-- cleaning hotelrooms table

SELECT
Room,
Adults,
IF (Children IS NULL, 0, Children) AS Children,
Features
FROM preppindata.`2022w25_hotelrooms`
```

Untitled

Room	Adults	Children	Features
205	2	0	Double
305	2	0	Double
203	2	1	Double
208	2	1	Double
306	2	1	Double

4 columns · 40 rows

0.8 seconds (41 minutes ago) Download CSV

Expand

step_4

```
-- bringing it all together
-- matching guests to the rooms which have capacity for their entire party

SELECT *
FROM step_2 -- guests
LEFT JOIN step_3 -- hotelrooms
ON Children_In_Party <= Children AND Adults_In_Party <= Adults
ORDER BY Party, Room ASC;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 856 rows 0.8 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)

step_5

```
-- filtering so that double/twin bed preferences are adhered to
```

```
SELECT *
FROM step_4
WHERE Features LIKE '%|| Double_Twin ||%';
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 451 rows 0.7 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)

step_6

```
-- making sure guests who have accessibility requirements are matched to accessible rooms only
```

```
SELECT *
FROM step_5
WHERE
(CASE
    WHEN Requires_Accessible_Room = TRUE
        AND CONTAINS_SUBSTR (Features,"Accessible") THEN 'Keep'
    WHEN Requires_Accessible_Room = TRUE
        AND NOT CONTAINS_SUBSTR (Features,"Accessible") THEN 'Remove'
    ELSE "Keep"
END) = 'Keep';
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 379 rows 0.9 seconds (41 minutes ago)

[Download CSV](#)[Expand](#)

step_7

```
-- calculate score for how additional requests match up to features, 1 for match, 0 for no match
```

```

SELECT *,
IF (CONTAINS_SUBSTR(Additional_Requests,"Bath") AND CONTAINS_SUBSTR(Features,"Bath"),1,0) AS Bath,
IF (CONTAINS_SUBSTR(Additional_Requests,"High Floor") AND CONTAINS_SUBSTR(Features,"High Floor"),1,0) AS Floor,
IF (NOT(CONTAINS_SUBSTR(Features,"Near to lift")) AND CONTAINS_SUBSTR(Additional_Requests,"NOT Near to lift"),1,0) AS Lift
FROM step_6;

```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Chese	2	0	Double	false	High Floor, NOT Near to lift
Chese	2	0	Double	false	High Floor, NOT Near to lift
Chese	2	0	Double	false	High Floor, NOT Near to lift
Chese	2	0	Double	false	High Floor, NOT Near to lift
Chese	2	0	Double	false	High Floor, NOT Near to lift

14 columns · 379 rows 0.9 seconds (41 minutes ago)

[Download CSV](#) [Expand](#)

step_8

```

-- calculate success request satisfaction % for each room

SELECT
*,
IF (Nr_Requests= 0, 100 , ROUND(((Bath + Floor + Lift) / Nr_Requests)*100)) AS Satisfaction_Score
FROM step_7
ORDER BY Party ASC;

```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

15 columns · 379 rows 0.8 seconds (41 minutes ago)

[Download CSV](#) [Expand](#)

step_9

```

-- calculate the max satisfaction score by party

SELECT *,
MAX(Satisfaction_Score) OVER (PARTITION BY Party) AS Suitable_Score,
FROM step_8

```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Guisler	1	0	Twin	false	Bath, High Floor, NOT Near to
Guisler	1	0	Twin	false	Bath, High Floor, NOT Near to
Guisler	1	0	Twin	false	Bath, High Floor, NOT Near to
Guisler	1	0	Twin	false	Bath, High Floor, NOT Near to
Guisler	1	0	Twin	false	Bath, High Floor, NOT Near to

16 columns · 379 rows 1.5 seconds (27 minutes ago)

[Download CSV](#) [Expand](#)

issue_to_report

Cell/table step_9 ▾

parties where none of the rooms match the requirements

Suitable_Score	is	0	—
----------------	----	---	---

+ Add filter

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests	Nr_Req
Iczokvitz	2	0	Double	true	Bath	1
Iczokvitz	2	0	Double	true	Bath	1
Iczokvitz	2	0	Double	true	Bath	1
Iczokvitz	2	0	Double	true	Bath	1

16 columns · 4 rows 1.5 seconds (18 minutes ago)

[Download CSV](#)[Expand](#)**step_10**

-- filter so that guests are only left with rooms with the highest request (MAX) satisfaction score (read: rooms that are most suitable for them)

```
SELECT *
FROM step_9
WHERE Suitable_Score = Satisfaction_Score;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Krishtopaittis	1	0	Double	false	(Empty)
Krishtopaittis	1	0	Double	false	(Empty)
Krishtopaittis	1	0	Double	false	(Empty)
Krishtopaittis	1	0	Double	false	(Empty)
Krishtopaittis	1	0	Double	false	(Empty)

16 columns · 242 rows 1.5 seconds (14 minutes ago)

[Download CSV](#)[Expand](#)**step_11**

-- for every room calculate the max of adults in the guest party (for rooms with with the largest capacity we want to make sure guests with larger parties are prioritised)

```
SELECT *,
MAX(Adults_In_Party) OVER (PARTITION BY Room) AS Most_Adults_For_Room
FROM step_10;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Aarons	2	0	Twin	false	Bath
Fallens	1	0	Twin	false	(Empty)
Cullum	2	0	Twin	false	(Empty)
Ghiardelli	2	0	Double	false	Bath
Sumers	2	0	Double	false	(Empty)

17 columns · 242 rows 1.7 seconds (14 minutes ago)

[Download CSV](#)[Expand](#)

Final output

output

```
-- filtering the data to make sure guests with larger parties are prioritised

SELECT
Party,
Adults_In_Party,
Children_In_Party,
Double_Twin,
Requires_Accessible_Room,
Additional_Requests,
Satisfaction_Score AS Request_Satisfaction_Score,
Room,
Adults,
Children,
Features
FROM step_11
WHERE Adults < 3 OR Most_Adults_For_Room = Adults_In_Party;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additional_Requests
Lansdale	1	0	Twin	false	High Floor
Londer	1	0	Twin	false	High Floor, NOT Near to lift
Jeannel	1	0	Twin	false	High Floor, NOT Near to lift
Fallens	1	0	Twin	false	(Empty)
Cullum	2	0	Twin	false	(Empty)

11 columns · 217 rows 1.6 seconds (14 minutes ago)

 Download CSV  Expand

Notebook cells

all_data_hotelrooms

```
SELECT *
FROM preppindata.`2022w25_hotelrooms`;
```

Untitled

Room	Adults	Children	Features	
205	2	(Empty)	Double	
305	2	(Empty)	Double	
203	2	1	Double	
208	2	1	Double	
306	2	1	Double	

4 columns · 40 rows 0.8 seconds (39 minutes ago)

all_data_guests

```
SELECT *
FROM preppindata.`2022w25_guests`;
```

Untitled

Party	Adults	Children	Double_Twin	Requires_Accessible_Room	Additional_Requests
Fallens	1	0	Twin	false	N/A
Marusik	1	0	Twin	false	Bath, High Floor
Lansdale	1	0	Twin	false	High Floor
Jeannel	1	0	Twin	false	High Floor, NOT Near
Londer	1	0	Twin	false	High Floor, NOT Near

6 columns · 30 rows 0.7 seconds (39 minutes ago)

testing_array

step_1

```
-- clean guests table, replace N/A will null

SELECT
Party,
Adults,
Children,
Double_Twin,
Requires_Accessible_Room,
IF(Additional_Requests = "N/A", NULL, Additional_Requests) AS Additional_Requests
FROM preppindata.`2022w25_guests`;
```

Untitled

Party	Adults	Children	Double_Twin	Requires_Accessible_Room	Additional_Requests
Fallens	1	0	Twin	false	(Empty)
Marusik	1	0	Twin	false	Bath, High Floor
Lansdale	1	0	Twin	false	High Floor
Jeannel	1	0	Twin	false	High Floor, NOT Near
Lonner	1	0	Twin	false	High Floor, NOT Near

6 columns · 30 rows 0.8 seconds (39 minutes ago)

step_2

```
-- counting the number of additional requests by guest
```

```
SELECT
Party,
Adults AS Adults_In_Party,
Children AS Children_In_Party,
Double_Twin,
Requires_Accessible_Room,
Additional_Requests,
IF(ARRAY_LENGTH(SPLIT(Additional_Requests,"")) IS NULL, 0, ARRAY_LENGTH(SPLIT(Additional_Requests,"")) ) AS Nr_Requests
FROM step_1;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Ad
Fallens	1	0	Twin	false	(Er
Marusik	1	0	Twin	false	Ba
Lansdale	1	0	Twin	false	Hig
Jeannel	1	0	Twin	false	Hig
Londer	1	0	Twin	false	Hig

7 columns · 30 rows 0.7 seconds (39 minutes ago)

step_3

```
-- cleaning hotelrooms table

SELECT
Room,
Adults,
IF (Children IS NULL, 0, Children) AS Children,
Features
FROM preppindata.`2022w25_hotelrooms`
```

Untitled

Room	Adults	Children	Features	
205	2	0	Double	
305	2	0	Double	
203	2	1	Double	
208	2	1	Double	
306	2	1	Double	

4 columns · 40 rows 0.8 seconds (39 minutes ago)

step_4

```
-- bringing it all together
-- matching guests to the rooms which have capacity for their entire party

SELECT *
FROM step_2 -- guests
LEFT JOIN step_3 -- hotelrooms
ON Children_In_Party <= Children AND Adults_In_Party <= Adults
ORDER BY Party, Room ASC;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Add
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 856 rows 0.9 seconds (39 minutes ago)

step_5

```
-- filtering so that double/twin bed preferences are adhered to
```

```
SELECT *
FROM step_4
WHERE Features LIKE '% || Double_Twin || %';
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Add
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 451 rows 0.9 seconds (39 minutes ago)

step_6

```
-- making sure guests who have accessibility requirements are matched to accessible rooms
```

```
SELECT *
FROM step_5
WHERE
(CASE
    WHEN Requires_Accessible_Room = TRUE
        AND CONTAINS_SUBSTR (Features,"Accessible") THEN 'Keep'
    WHEN Requires_Accessible_Room = TRUE
        AND NOT CONTAINS_SUBSTR (Features,"Accessible") THEN 'Remove'
    ELSE "Keep"
END) = 'Keep';
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Add
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

11 columns · 379 rows 0.8 seconds (39 minutes ago)

step_7

```
-- calculate score for how additional requests match up to features, 1 for match, 0 for no

SELECT *,
IF (CONTAINS_SUBSTR(Additional_Requests,"Bath") AND CONTAINS_SUBSTR(Features,"Bath"),1,0) AS Bath,
IF (CONTAINS_SUBSTR(Additional_Requests,"High Floor") AND CONTAINS_SUBSTR(Features,"High Floor"),1,0) AS High_Floor,
IF (NOT(CONTAINS_SUBSTR(Features,"Near to lift")) AND CONTAINS_SUBSTR(Additional_Requests,"Near to lift"),1,0) AS Lift
FROM step_6;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additio
Chese	2	0	Double	false	High Flo
Chese	2	0	Double	false	High Flo
Chese	2	0	Double	false	High Flo
Chese	2	0	Double	false	High Flo
Chese	2	0	Double	false	High Flo

14 columns · 379 rows 0.8 seconds (39 minutes ago)

step_8

```
-- calculate success request satisfaction % for each room
```

```
SELECT
*, 
IF (Nr_Requests= 0, 100 , ROUND(((Bath + Floor + Lift) / Nr_Requests)*100)) AS Satisfaction
FROM step_7
ORDER BY Party ASC;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Add
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath
Aarons	2	0	Twin	false	Bath

15 columns · 379 rows 0.8 seconds (39 minutes ago)

step_9

```
-- calculate the max satisfaction score by party

SELECT *,
MAX(Satisfaction_Score) OVER (PARTITION BY Party) AS Suitable_Score,
FROM step_8
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additio
Guisler	1	0	Twin	false	Ba
Guisler	1	0	Twin	false	Ba
Guisler	1	0	Twin	false	Ba
Guisler	1	0	Twin	false	Ba
Guisler	1	0	Twin	false	Ba

16 columns · 379 rows 0.8 seconds (39 minutes ago)

issue_to_report

Cell/table step_9

parties where none of the rooms match the requirements

Suitable_Score is 0 —

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Additio
Iczokvitz	2	0	Double	true	Bath
Iczokvitz	2	0	Double	true	Bath
Iczokvitz	2	0	Double	true	Bath
Iczokvitz	2	0	Double	true	Bath

16 columns · 4 rows 0.7 seconds (39 minutes ago)

step_10

-- filter so that guests are only left with rooms with the highest request (MAX) satisfaction (we want to make sure rooms that are most suitable for them)

```
SELECT *
FROM step_9
WHERE Suitable_Score = Satisfaction_Score;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Ad
Krishtopaittis	1	0	Double	false	(Er
Krishtopaittis	1	0	Double	false	(Er
Krishtopaittis	1	0	Double	false	(Er
Krishtopaittis	1	0	Double	false	(Er
Krishtopaittis	1	0	Double	false	(Er

16 columns · 242 rows 0.7 seconds (39 minutes ago)

step_11

-- for every room calculate the max of adults in the guest party (for rooms with the highest request we want to make sure guests with larger parties are prioritised)

```
SELECT *,
MAX(Adults_In_Party) OVER (PARTITION BY Room) AS Most_Adults_For_Room
FROM step_10;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Ad
Aarons	2	0	Twin	false	Ba
Fallens	1	0	Twin	false	(Er
Cullum	2	0	Twin	false	(Er
Ghirardelli	2	0	Double	false	Ba
Sumers	2	0	Double	false	(Er

17 columns · 242 rows 0.8 seconds (39 minutes ago)

 **Final output**

output

```
-- filtering the data to make sure guests with larger parties are prioritised

SELECT
Party,
Adults_In_Party,
Children_In_Party,
Double_Twin,
Requires_Accessible_Room,
Additional_Requests,
Satisfaction_Score AS Request_Satisfaction_Score,
Room,
Adults,
Children,
Features
FROM step_11
WHERE Adults < 3 OR Most_Adults_For_Room = Adults_In_Party;
```

Untitled

Party	Adults_In_Party	Children_In_Party	Double_Twin	Requires_Accessible_Room	Ad
Lansdale	1	0	Twin	false	Hiç
Londer	1	0	Twin	false	Hiç
Jeannel	1	0	Twin	false	Hiç
Fallens	1	0	Twin	false	(Er
Cullum	2	0	Twin	false	(Er

11 columns · 217 rows 0.8 seconds (39 minutes ago)

Reference materials

Functions & operators used

[SPLIT](#): splits the values in a string using the delimiter argument and returns an array of those values

[ARRAY_LENGTH](#): returns the size (number of elements) of an array as an integer

[CONTAINS_SUBSTR](#): performs a case insensitive search to see if a value exists, returns true if it does and false if it doesn't

[MAX\(\) OVER \(PARTITION BY\)](#): calls the MAX function over the specified window and returns the max value for every row in the window (aggregate window function)

Blogs

[Working with Arrays](#)

[An Easy Guide to Advanced SQL functions](#)