University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5th, 2023

Group Number: 64

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Henry Kim	32523722	q1y6v	henryshkim@hotmail.com
Noel Illing	65046468	b1o2w	illing.noel@gmail.com
Babak Hadady	77085647	y9m6h	babakhad@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

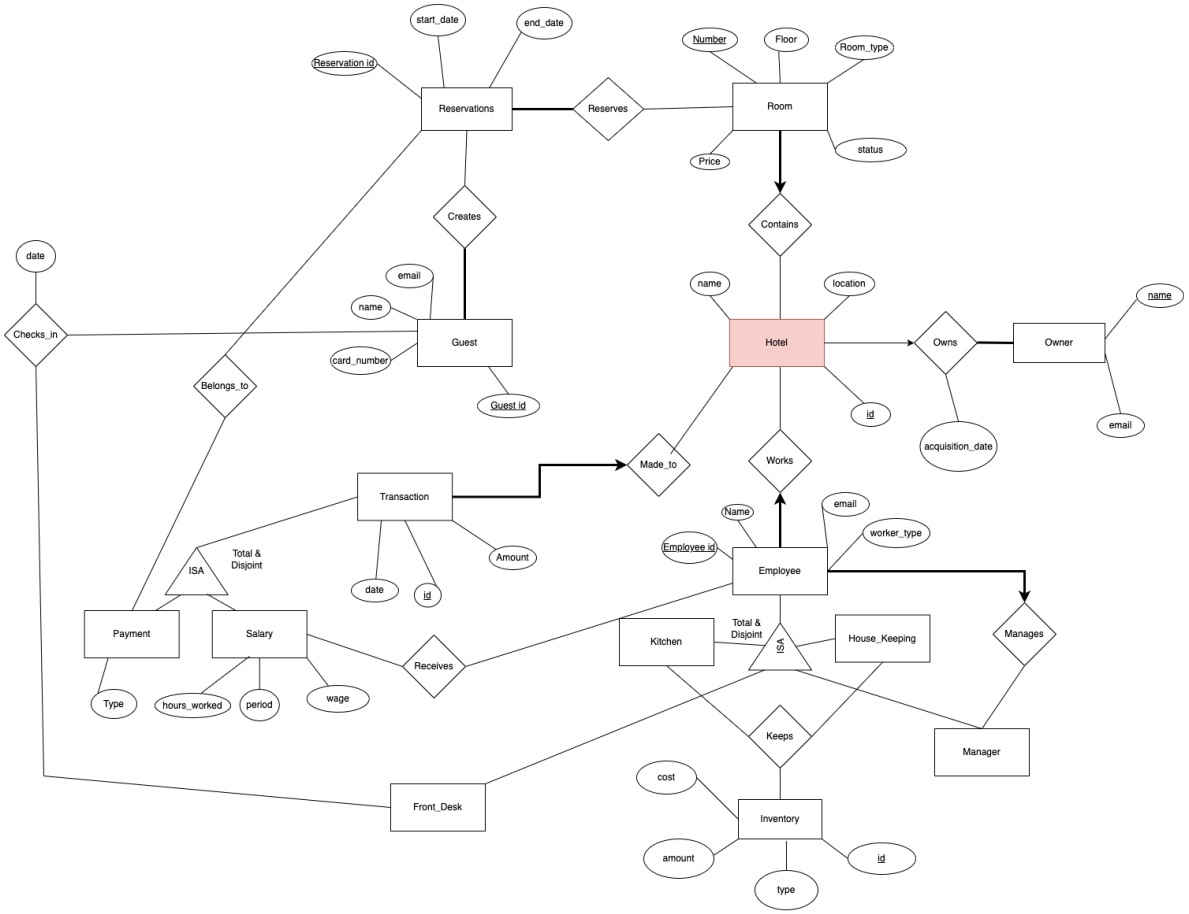
The domain of this application will be a hospitality management service. It will be helping manage room states/reservations for multiple hotels. Users of the application will be able to insert, update, and delete reservations made at the hotel by going to the update database page. Users will also be able to go to the query page where they can submit many different types of queries about reservations and rooms. Users can also view all reservations, select certain tuples from reservations, or select either reservations, reserves, or rooms tables and specific attributes to view.

Schema Changes

We changed our schema to become a subset of the one we created in milestone 2 because we wanted users to have more control over the queries that they're submitting. Since it takes time to fully implement features we'd rather have users be able to do lots of queries on fewer tables than a few queries on many tables. We also didn't normalize our roomContains table because we wanted to have our queries on rooms to have all the related attributes included, and didn't want to split the table up into two.

Schema

Queries are only able to be done on Reservations, Room, and Reserves relations.



SQL script

Guest:

```
  GUEST_ID CARD_NUMBER GUEST_NAME          EMAIL
---------- ----------- -------------------- ------------------------------
    123456    24429988 Henry Kim            walkingbuddies2002@gmail.com
    222222    48603847 Benry Bim            zedandshen@gmail.com
    123457    66739853 Jenry Jim            yuumicarry@gmail.com
    444444    12546434 Tenry Tim            thisisnotanemail@gmail.com
    666666    89745676 Lenry Lim            impostersussy@gmail.com
```

EmployeeManagesWorks:

```
EMPLOYEE_ID NAME                 EMAIL
----------- -------------------- ------------------------------
WORKER_TYPE          MANAGER_ID         ID
-------------------- ---------- ----------
     123455 Alan                 alan@email.com
Kitchen                  123455       1234

     123456 Bob                  bob@email.com
Front_Desk               123455       2342

     123457 Cole                 cole@email.com
House_Keeping            123455       3243
```

```
EMPLOYEE_ID NAME                 EMAIL
----------- -------------------- ------------------------------
WORKER_TYPE          MANAGER_ID         ID
-------------------- ---------- ----------
     123458 David                david@email.com
Manager                  123455       4532

     123459 Elliot               elliot@email.com
Kitchen                  123455       8978
```

RoomContains:

```
ROOM_NUMBER      FLOOR ROOM_TYPE           STATUS                    PRICE
----------- ---------- -------------------- -------------------- ----------
        200          2 single               vacant                      150
        201          2 single               vacant                      150
        202          2 double               vacant                      200
        401          4 double               vacant                      200
        501          5 double               vacant                      200
        600          6 double               vacant                      200
        601          6 double               vacant                      200
        602          6 double               vacant                      200
        701          7 double               vacant                      200
        801          8 double               vacant                      200
```

Reservation:

START_DATE	END_DATE	RESERVATION_ID
jan 4	jan 7	101234
jan 4	jan 7	101235
jan 4	jan 7	101236
jan 4	jan 7	101237
jan 4	jan 7	101238

Creates:

GUEST_ID	RESERVATION_ID
123456	101234
222222	101235
222222	101236
444444	101237
666666	101238

BelongsTo:

BELONGSTO_ID	RESERVATION_ID
123455	101234
123456	101235
123457	101236
123458	101237
123459	101238

Owner:

OWNER_NAME	EMAIL
Henry Kim	henry@gmail.com
Noel Illing	noel@gmail.com
Babak Bob	babak@gmail.com
Henry Joe	henryJoe@gmail.com
Henry Cam	henryCam@gmail.com

TransactionMadeTo1:

AMOUNT	HOURS_WORKED	WAGE
10	1	15
11	2	16
12	3	17
13	4	18
14	5	19

TransactionMadeTo2:

ID	TRANSACTION_DATE	TRANSACTION_TYPE	HOTEL_ID
123459	February 24, 2002	Cancellation Fee	3243

PERIOD	HOURS_WORKED	AMOUNT
October, 10, 2002 - February, 10, 2002	5	14

ID	TRANSACTION_DATE	TRANSACTION_TYPE	HOTEL_ID
123457	February 22, 2002	Cancellation Fee	3243
123458	February 23, 2002	Cancellation Fee	1234

PERIOD	HOURS_WORKED	AMOUNT
October, 10, 2002 - February, 10, 2002	3	12

ID	TRANSACTION_DATE	TRANSACTION_TYPE	HOTEL_ID
1234	February 20, 2002	Room Payment	1234
123456	February 21, 2002	Room Payment	3243

PERIOD	HOURS_WORKED	AMOUNT
October, 10, 2002 - February, 10, 2002	1	10
October, 10, 2002 - February, 10, 2002	2	11

Reserves:

RESERVATION_ID	ROOM_NUMBER
101234	200
101234	201
101234	202
101234	401
101235	501
101236	600
101236	601
101236	602
101237	701
101238	801

HotelOwns1:

LOCATION	NAME
900 W Georgia St, Vancouver	Fairmont
5959 Student Union Blvd, Vancouver	Gage
783 Imagination Rd, Vancouver	Imagine Hotel
696 Lover Drive, Vancouver	Love Hotel
3204 Database St, China	Data Hotel

HotelOwns2:

LOCATION	OWNER_NAME	HOTEL_ID	ACQUISITION_DATE
696 Lover Drive, Vancouver	Heimerdinger Smith	4532	July 23, 2005
3204 Database St, China	Garen Darius	8978	February 11, 1987

LOCATION	OWNER_NAME	HOTEL_ID	ACQUISITION_DATE
900 W Georgia St, Vancouver	Elon Musk	1234	October 10, 2004
5959 Student Union Blvd, Vancouver	Tarzan Man	3243	November 17, 1999
783 Imagination Rd, Vancouver	John Smith	2342	January 1, 2002

Inventory1:

AMOUNT	COST	INVENTORY_TYPE
100	1000	chairs
50	100	spoons
100	80	forks
100	100	knives
50	700	blankets

Inventory2:

AMOUNT	ID	INVENTORY_TYPE
100	12345678	chairs
50	58349544	spoons
100	54982365	forks
100	45217893	knives
50	99875757	blankets

Keeps:

ID	EMPLOYEE_ID
12345678	123455
45217893	123458
54982365	123457
58349544	123456
99875757	123459

Receives:

ID	EMPLOYEE_ID
123456	123455
123456	123456
123457	123457
123458	123458
123459	123459

Checks_in1:

CHECK_IN_DATE	EMPLOYEE_ID
October 5, 2021	123455
October 6, 2021	123456
October 7, 2021	123456
October 8, 2021	123457
October 9, 2021	123457

Checks_in2:

CHECK_IN_DATE	GUEST_ID
October 5, 2021	123456
October 6, 2021	222222
October 7, 2021	123457
October 8, 2021	444444
October 9, 2021	666666

SQL Queries:

All query code found in php/db-requests.php

Insert **File:** db-requests.php **Start:** 237 **End**: 276	An insert into Reservation should automatically make one for Reserves AND update the roomsContains room for that number (vacant to occupied) Reservations: INSERT INTO Reservation VALUES (user inputs) Reserves: INSERT INTO Reserves VALUES (user inputs) RoomContains: UPDATE RoomContains RC SET RC.status = “occupied” WHERE RC.number = (user input number)

Retrieved data from table Reservations:

Reservation ID	Start Date	End Date
1234	jan 1	jan 2
101234	jan 4	jan 7
101235	jan 4	jan 7
101236	jan 4	jan 7
101237	jan 4	jan 7
101238	jan 4	jan 7
101	jan 3	jan 4

Successful View Operation

Insert a Reservation

Reservation ID:

222

Start Date:

jan 1

End Date:

jan 2

Room Number:

201

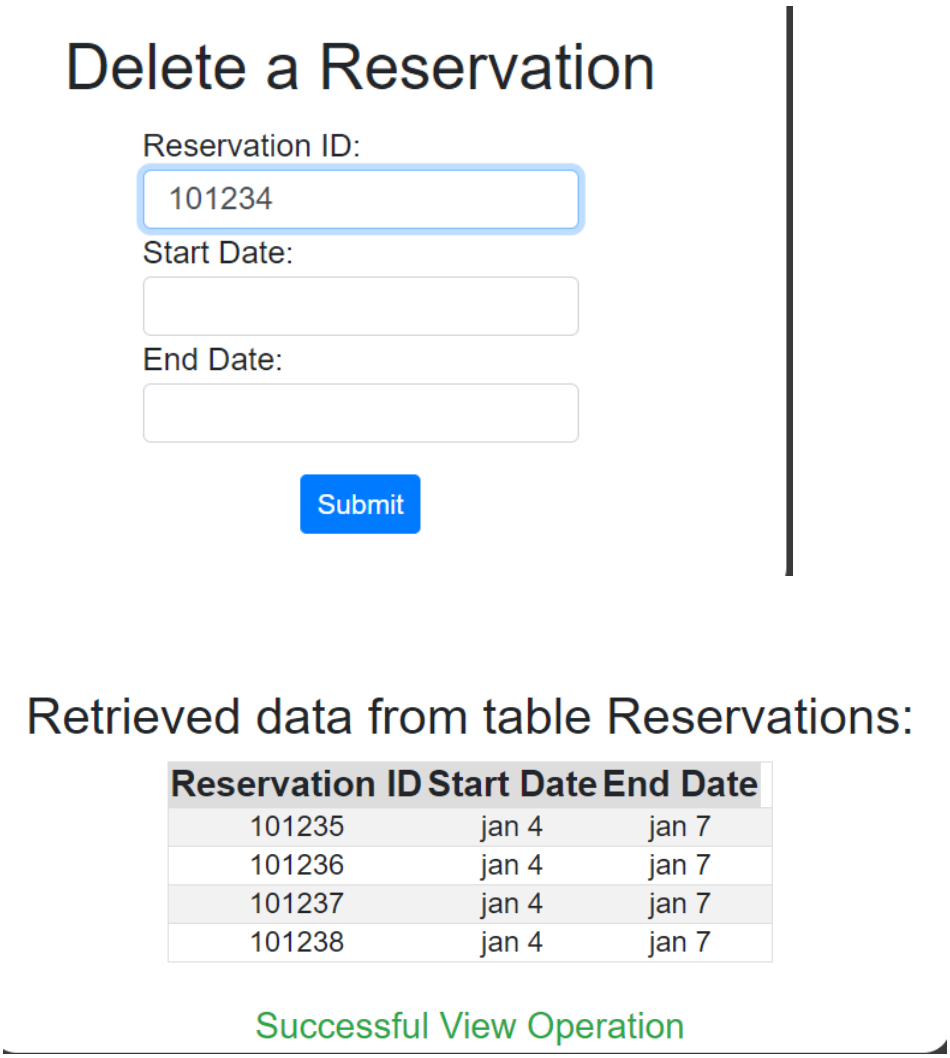
Submit

<table>
<tr>
<td></td>
<td>
Retrieved data from table Reservations:
<table>
<tr><th>Reservation ID</th><th>Start Date</th><th>End Date</th></tr>
<tr><td>1234</td><td>jan 1</td><td>jan 2</td></tr>
<tr><td>101234</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101235</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101236</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101237</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101238</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101</td><td>jan 3</td><td>jan 4</td></tr>
<tr><td>222</td><td>jan 1</td><td>jan 2</td></tr>
</table>
Successful View Operation
</td>
</tr>
<tr>
<td><u>Delete</u>

File:
db-requests.php
Start: 292
End: 340</td>
<td>
Should be cascade-on-delete, so row corresponding to Reserve should also be deleted AND roomsContains room should go to vacant

DELETE FROM Reservation
WHERE (user input - some attribute) = (user input)

Retrieved data from table Reservations:
<table>
<tr><th>Reservation ID</th><th>Start Date</th><th>End Date</th></tr>
<tr><td>101234</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101235</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101236</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101237</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101238</td><td>jan 4</td><td>jan 7</td></tr>
</table>
Successful View Operation
</td>
</tr>
</table>

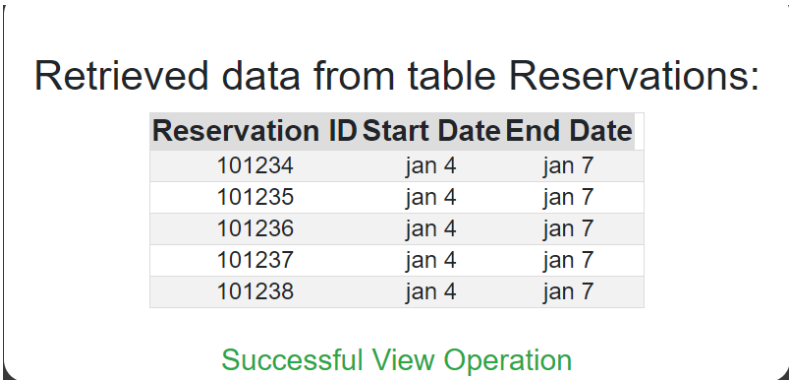


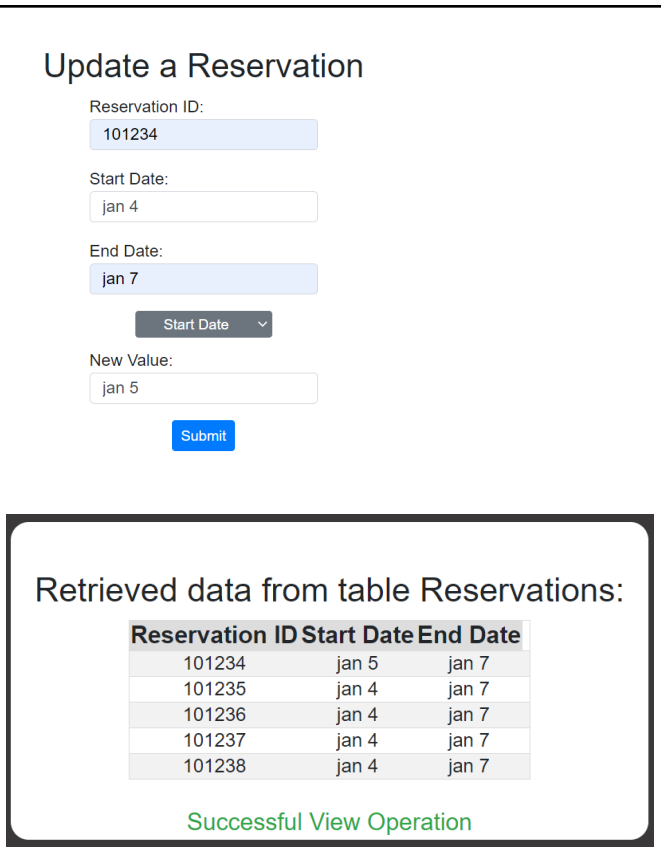
Update

File:
db-requests.php
Start: 19
End: 85

User can update multiple values

UPDATE Reservation
SET (user input - some attribute) = (user input)

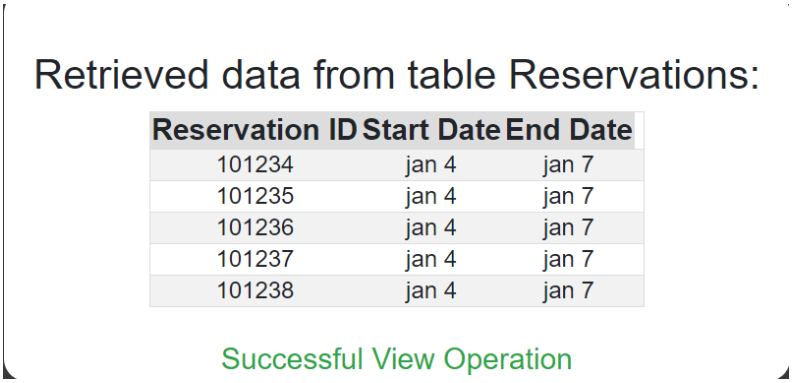




Selection

File:
db-requests.php
Start: 88
End: 128

SELECT *
FROM Reservations R
WHERE (user attribute input) = (user input)



<table>
<tr>
<td></td>
<td>
<h2>Select Certain Tuples From Reservation</h2>
Select a Column
Reservation ID
Value
101234
Submit
<h2>Retrieved data from table Reservation:</h2>
<table>
<tr><th>Reservation ID</th><th>Start Date</th><th>End Date</th></tr>
<tr><td>jan 4</td><td>jan 7</td><td>101234</td></tr>
</table>
Successful Selection Operation
</td>
</tr>
<tr>
<td><u>Projection</u>

File:
db-requests.php
Start: 131
End: 235</td>
<td>
SELECT (user input(s))
FROM (user input)
<h2>Retrieved data from table Reservations:</h2>
<table>
<tr><th>Reservation ID</th><th>Start Date</th><th>End Date</th></tr>
<tr><td>101234</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101235</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101236</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101237</td><td>jan 4</td><td>jan 7</td></tr>
<tr><td>101238</td><td>jan 4</td><td>jan 7</td></tr>
</table>
Successful View Operation
<h2>Select a Table and Attributes to View</h2>
Select a Table: Reservations Submit
Select Attributes
Reservation ID ☑
Start Date ☑
End Date ☐
Submit
</td>
</tr>
</table>

	Retrieved data from table Reservations: **Reservation ID** **Start Date** 101234 jan 4 101235 jan 4 101236 jan 4 101237 jan 4 101238 jan 4 Successful Projection Operation

Join

File: db-requests.php
Start: 459
End: 518

SELECT R.reservation_id, RC.number
FROM Reservations R, RoomContains RC
WHERE (user input - some attribute) = (user input)

Retrieved data from table Reservations:

Reservation ID	Start Date	End Date
101234	jan 4	jan 7
101235	jan 4	jan 7
101236	jan 4	jan 7
101237	jan 4	jan 7
101238	jan 4	jan 7

Successful View Operation

Retrieved data from table Room:

Room Number	Room Type	Floor	Status	Price
200	single	2	vacant	150
201	queen	2	vacant	250
202	double	2	vacant	200
401	single	4	vacant	150
501	double	5	vacant	200
600	queen	6	vacant	250
601	double	6	vacant	200
602	king	6	vacant	300
701	double	7	vacant	200
801	master	8	vacant	350

Join Reservations and Room on a Specific Condition

Select a Column

Reservation ID

Value

101234

Submit

Retrieved data from table Reservation and Room:

Reservation ID	Room Number
101234	200
101234	201
101234	202
101234	401
101234	501
101234	600
101234	601
101234	602
101234	701
101234	801

Successful Selection Operation

Aggregation with Group By

File: db-requests.php
Start: 342
End: 363

Grab number of vacant/occupied rooms for each room type

```
SELECT RC.room_type, COUNT(*)
FROM RoomContains RC
WHERE RC.status = (user input)
GROUP BY RC.room_type
```

Retrieved data from table Room:

Room Number	Room Type	Floor	Status	Price
200	single	2	vacant	150
201	queen	2	vacant	250
202	double	2	vacant	200
401	single	4	vacant	150
501	double	5	vacant	200
600	queen	6	vacant	250
601	double	6	vacant	200
602	king	6	vacant	300
701	double	7	vacant	200
801	master	8	vacant	350

Find number of vacant or occupied rooms for each room type

vacant Submit

Retrieved data from table Room:

Room Type	Count
king	1
single	2
queen	2
double	4
master	1

Successful Operation

Aggregation with Having

File: db-requests.php
Start: 365
End: 387

For each floor that has more than X available rooms, grab the cheapest room available

```
SELECT RC.floor, MIN(RC.price), RC.room_number
FROM RoomContains RC
WHERE RC.status = “vacant”
GROUP BY RC.floor
HAVING COUNT(*) > (user input)
```

Retrieved data from table Room:

Room Number	Room Type	Floor	Status	Price
200	single	2	vacant	150
201	queen	2	vacant	250
202	double	2	vacant	200
401	single	4	vacant	150
501	double	5	vacant	200
600	queen	6	vacant	250
601	double	6	vacant	200
602	king	6	vacant	300
701	double	7	vacant	200
801	master	8	vacant	350

For each floor that has more than X amount of available rooms, grab the cheapest available room

1 Submit

Retrieved data from table Room:

Floor	Min Price
6	200
2	150

Successful Operation

Nested Aggregation with Group By

File: db-requests.php
Start: 389
End: 413

Find the cheapest / most expensive available room

Most expensive:

```
SELECT RC.number
FROM RoomContains RC
WHERE RC.status = “vacant” and RC.price >= ALL (
        SELECT MAX(RC2.price)
        FROM RoomContains RC2
        WHERE RC2.status = “vacant”
        GROUP BY RC2.floor)
```

Cheapest:

```
SELECT RC.number
FROM RoomContains RC
WHERE RC.status = “vacant” and RC.price <= ALL (
        SELECT MIN(RC2.price)
        FROM RoomContains RC2
        WHERE RC2.status = “vacant”
        GROUP BY RC2.floor)
```

Retrieved data from table Room:

Room Number	Room Type	Floor	Status	Price
200	single	2	vacant	150
201	queen	2	vacant	250
202	double	2	vacant	200
401	single	4	vacant	150
501	double	5	vacant	200
600	queen	6	vacant	250
601	double	6	vacant	200
602	king	6	vacant	300
701	double	7	vacant	200
801	master	8	vacant	350

Find the most cheap/expensive available room

cheapest Submit

Retrieved data from table Room:

Number
200
401

Division

File: db-requests.php
Start: 415
End: 432

Find reservation id of reservation that reserved all rooms on floor X

```
SELECT R.reservation_id
FROM Reservations R
WHERE NOT EXISTS (
        (SELECT RC.number
        FROM RoomContains RC
        WHERE RC.floor = (user input))
        EXCEPT
        (SELECT Res.number
        FROM Reserves Res
        WHERE R.reservation_id = Res.reservation_id))
```

Retrieved data from table Reservations:

Reservation ID	Start Date	End Date
101234	jan 4	jan 7
101235	jan 4	jan 7
101236	jan 4	jan 7
101237	jan 4	jan 7
101238	jan 4	jan 7

Successful View Operation

Retrieved data from table Reserves:

Reservation ID	Room Number
101234	200
101234	201
101234	202
101234	401
101235	501
101236	600
101236	601
101236	602
101237	701
101238	801

Find reservation ID that reserved all rooms on selected floor

2 Submit

Retrieved data from table Reservations:

Reservation ID
101234

Successful Division Operation