Data Management Assignment 1

# SQL, Data, Importing and Views

**Student Name:** Ola Stefan Rudvin

**Student ID:** 51549217

**Email:** o.rudvin.15@aberdeen.ac.uk

# MyAberdeen PHPMyAdmin Database

# **Username:** u01osr15\_cs2015

# **Password:** cs2015

# Question 1.

SELECT R.HotelNo, COUNT(R.type) AS SingleRooms

FROM room R

WHERE R.type = "single"

GROUP BY R.HotelNo;

# Question 2.

SELECT R.HotelNo, COUNT(R.type) AS Rooms

FROM room R

WHERE R.type = "family" OR R.type = "double"

GROUP BY R.HotelNo

HAVING COUNT(R.type) >= '4'

ORDER BY Rooms ASC;

# Question 3.

SELECT C.Name, C.Surname, C.StreetName,C.City,C.Postcode, C.Country, H.Name as HotelName, H.Location as HotelLocation

FROM customer C, booking B, hotel H

WHERE C.CustNo = B.CustNo

AND B.HotelNo = H.HotelNo

GROUP BY C.Surname, H.Name;

# Question 4.

SELECT C.Name, C.Surname, C.StreetName,C.City,C.Postcode, C.Country, B.nights \* R.Rate AS TotalCost

FROM customer C, booking B, room R

WHERE C.Surname = 'Flounder'

AND C.CustNo = B.CustNo

AND B.RoomNo = R.RoomNo

AND R.HotelNo = B.HotelNo;

# 

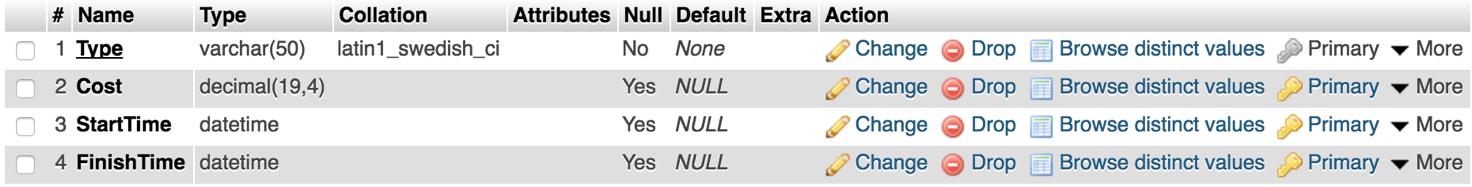
# Part 2

# Question 5.

In this report I will critique the hotel database based on my experience with it. I will first analyze each table separately for noteworthy mentions and later assess the database as a whole. While I do not know of the requirements and limitations of the hotel chain, I believe that a look at its SQL database provides a sufficient perspective on its business practises.

First of all, the booking table has some problems. The custNo ID attribute has an extremely questionable name and should be changed into something like CustomerNo to avoid confusion. It is the very first item that a new database user sees, and should therefore be named clearly to give a good impression of the whole database. The second issue is the HotelNo field. Personally I feel that the value ‘A01’ is unnecessarily complex as well as it using the alphabet, which limits values below 26 items. Perhaps a simpler combination of numbers and letters instead of just ‘01’ could help resolve this issue. Another problem is in the StartDate attribute, as it includes the hours, minutes and seconds of new bookings which is completely unnecessary. Although this timestamp is probably used for compatibility with e.g. UNIX timestamps, it creates extra hassle and confusion for the database user. However, if this data would have been utilised correctly, such as in a busy hotel where reservations are competitive up to certain hours, using a more accurate timestamp could be helpful.

The biggest problem of the table is the use of 3 primary keys, which can all change when new bookings are made. Ideally the database should use a single primary key for each booking (possibly linked to the customer table as well) that is not used as data, only as identification. One shouldn’t base the primary key value off of the data in the row. This applies to all tables, especially the function table which uses the ‘type’ attribute as it’s primary key.



The ‘Type’ attribute as the primary key in the function table

Moving on to the customer table, we have a problem of first and middle names. J.Robert is written in one table row, which should either be set to only include a customers first name or the row should be split into a first and middle name. The house number is another issue, where the letter should probably be set into a separate row from the number. The number is also limited to be at most 3 characters, which might not be enough for some addresses (especially outside the UK). On the data side the NULL values in the house numbers, postcodes and car registration plates are obviously in need of correction. The telephone numbers could also include the prefix of the country it belongs to in case the call is international. This would make it much faster to contact customers in case of emergencies.

As a side note, the rights to the customer table should only be granted to the administrator to respect the customers’ privacy. With the customer’s full address, car registration details and phone number severe harm could be achieved with social engineering. However, some more identification could be useful to track down customers should something bad happen- such as date of birth (Although most hotels require presentation of formal identification at the beginning of the stay).

As for the function table, it appears to be concise yet it is completely static data which could possibly be incorporated completely outside the database. The hotel table has issues with the manager and deputy manager attributes, where names have Mr, Ms prefixes with only initials for first names. Having this data in a proper format is crucial for expanding the business, applying price reductions for employees or comparing another ‘Managers’ database with our hotel database.

The room table has problems in duplicated room numbers in different hotels, in the category attribute which only holds a binary value of smoking vs. no smoking, in the facilities attribute which show multiple values and the services attributes which are terribly implemented. A cleaner, professional database would have separate, named columns for each service and facility for future compatibility and ease of use. Currently a database administrator must physically read what each facility contains, as the name of the column doesn’t do any work towards revealing its contents. All of the rooms have an abundance of repetition going on. Although this is useful for identifying rooms, a grouping function of e.g. “family”, “double”, “single” etc. rooms would be easier to read before getting into the individual room attributes.

Overall, the relationships between the rooms, bookings and entertainment options between hotels are very long-winded with multiple primary keys linking tables. This creates headaches for managers who must write long SQL statements and sets bottlenecks for execution time as well. Perhaps there could be views for each single hotel that show all required information. Also, one of the more disappointing factors was the quality of data. Although much of the data was available and set correctly, the abundance of NULL and ill-formatted values (such as first names) speaks otherwise.

I have been very critical of the database over the course of this report, but in conclusion it is still a working service that is useful in all aspects of hotel work. By giving different users access to different parts of the database, employees can for example see the entertainment options for the weekend and managers can see booking information to aid their work. Although there are several issues with primary keys, secondary keys and data quality, the database still provides enough information to be useable, albeit not to its fullest extent.