NUI Galway
OÉ Gaillimh

Group Assignment Submission Form

Assignment Details

Module Code: **MS218** Assignment Title: **Database Design & Development**

Group Members: (please use BLOCK CAPITALS)

Student ID	Programme (e.g. MBS, M.Sc.ISM)	Student Name	Contact Details (Email, Telephone)
16403592	4BC8	Odhran Griffin	o.griffin3@nuigalway.ie 087-671-3936
16300036	4BC8	Edward Cloete	e.cloete1@nuigalway.ie 0834158015
16450796	4BC8	Brian Mooney	B.MOONEY4@nuigalway.ie

I/We hereby declare that this assignment submission is my/our own original work. I/We have read the University *Code of Practice for Dealing with Plagiarism*[*] and are aware that the possible penalties for plagiarism include expulsion from the University. ***I/We attach a list of all sources that were consulted in the preparation of this assignment e.g. books, journals, Web sites etc.***

Signature: [illegible] **Date: 17/11/2019**

Brian Mooney

[illegible]

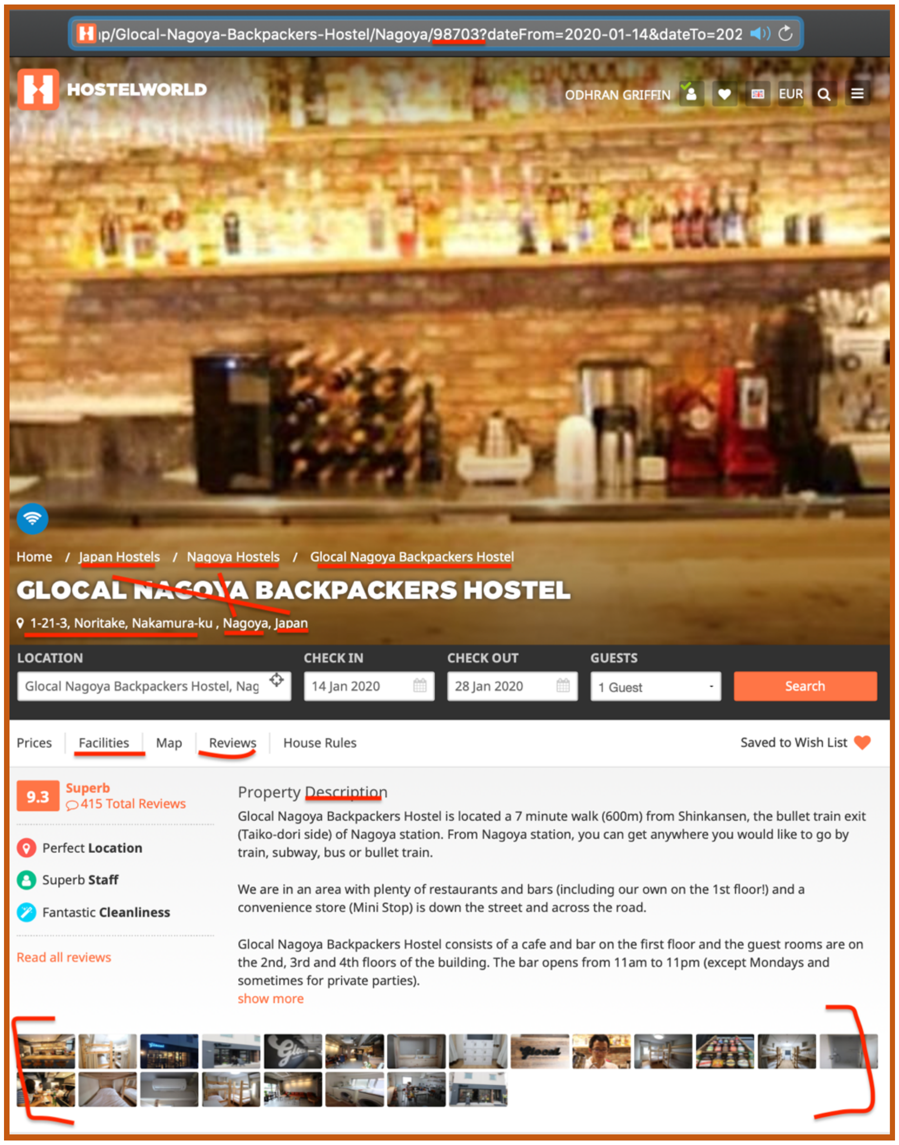
[*] http://www.nuigalway.ie/celt/teaching_and_learning/academic_honesty.html

Table of Contents

Reverse Engineering of the database of hostelworld.com

Introduction

Hostel World is a global Hostel-based online booking platform. The site allows both Hostel owners and hostel guests streamline the booking experience. Hostels can create a page which offers all necessary information to potential guests and the infrastructure for them to make a booking. These potential guests can create user profiles. In order to make a booking, one must create a user profile. When creating a profile, they will be prompted to enter personal information. This user information will then be saved in the database and will be assigned to the user from that moment onwards. When a booking is made, it includes a collection of data from the Hostel Page, the user profile and the specific information selected by the user when they made the booking. After their stay, a user can create a review of their experience. They will be prompted to enter a score for 7 different attributes. They will also be asked to leave a description. The scores they have selected will be subject to AVERAGE/SUM/COUNT operations within the database and displayed at different points within the site.



p/Glocal-Nagoya-Backpackers-Hostel/Nagoya/98703?dateFrom=2020-01-14&dateTo=202
HOSTELWORLD
ODHRAN GRIFFIN
EUR
Home / Japan Hostels / Nagoya Hostels / Glocal Nagoya Backpackers Hostel
GLOCAL NAGOYA BACKPACKERS HOSTEL
1-21-3, Noritake, Nakamura-ku , Nagoya, Japan
LOCATION
Glocal Nagoya Backpackers Hostel, Nag
CHECK IN
14 Jan 2020
CHECK OUT
28 Jan 2020
GUESTS
1 Guest
Search
Prices
Facilities
Map
Reviews
House Rules
Saved to Wish List
9.3
Superb
415 Total Reviews
Perfect Location
Superb Staff
Fantastic Cleanliness
Read all reviews

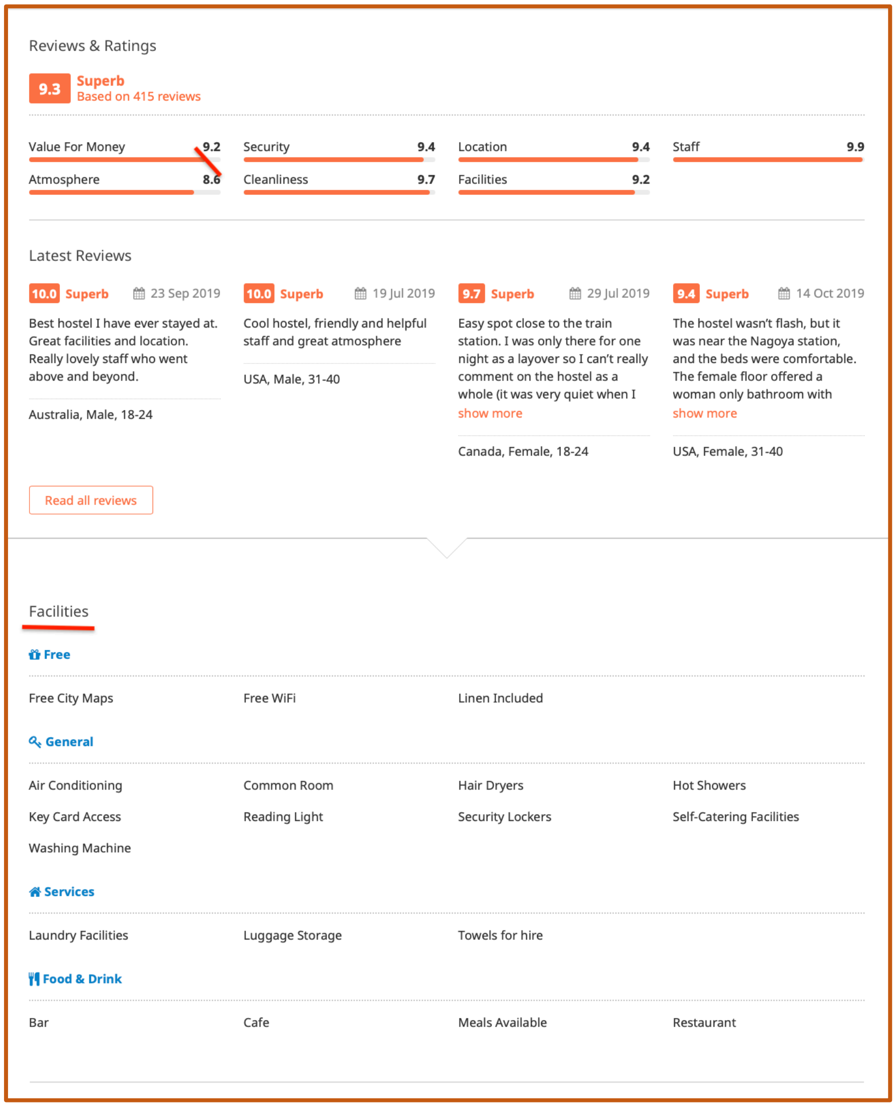
Property Description

Glocal Nagoya Backpackers Hostel is located a 7 minute walk (600m) from Shinkansen, the bullet train exit (Taiko-dori side) of Nagoya station. From Nagoya station, you can get anywhere you would like to go by train, subway, bus or bullet train.

We are in an area with plenty of restaurants and bars (including our own on the 1st floor!) and a convenience store (Mini Stop) is down the street and across the road.

Glocal Nagoya Backpackers Hostel consists of a cafe and bar on the first floor and the guest rooms are on the 2nd, 3rd and 4th floors of the building. The bar opens from 11am to 11pm (except Mondays and sometimes for private parties).

show more



Reviews & Ratings

9.3 Superb
Based on 415 reviews

Value For Money	9.2	Security	9.4	Location	9.4	Staff	9.9
Atmosphere	8.6	Cleanliness	9.7	Facilities	9.2		

Latest Reviews

10.0 Superb 23 Sep 2019
Best hostel I have ever stayed at. Great facilities and location. Really lovely staff who went above and beyond.
Australia, Male, 18-24

10.0 Superb 19 Jul 2019
Cool hostel, friendly and helpful staff and great atmosphere
USA, Male, 31-40

9.7 Superb 29 Jul 2019
Easy spot close to the train station. I was only there for one night as a layover so I can't really comment on the hostel as a whole (it was very quiet when I
show more
Canada, Female, 18-24

9.4 Superb 14 Oct 2019
The hostel wasn't flash, but it was near the Nagoya station, and the beds were comfortable. The female floor offered a woman only bathroom with
show more
USA, Female, 31-40

Read all reviews

Facilities

Free

Free City Maps	Free WiFi	Linen Included	

General

Air Conditioning	Common Room	Hair Dryers	Hot Showers
Key Card Access	Reading Light	Security Lockers	Self-Catering Facilities
Washing Machine			

Services

Laundry Facilities	Luggage Storage	Towels for hire	

Food & Drink

Bar	Cafe	Meals Available	Restaurant

Normalization

Assumptions:

- Each hostel has a hostel code. It is not displayed on the screen but can be seen within the URL of the page. The number ‘98703’ can be seen within the URL of the Glocal Backpackers Hostels page displayed above. This is the unique hostel code given to this particular hostel.
- Each user has a username. Users must register an account in order to leave a review or make a booking. They must select a unique username when creating an account.
- Each booking has a booking reference number. This number will be displayed on booking pages.

In the following notation, underlining signifies the primary key (aka “identifier” or “determinant”) and curly brackets { } signify multi-valued attributes or repeating groups.

We will ignore the fields containing an average rating, average review score and total reviews. These are computed fields based on AVERAGE/SUM/COUNT operations done on review data within the database.

Pricing of hostel beds will also be excluded as they will be generated by algorithms and change in response to demands. They will not be static values that can be represented in this database.

The normalization process will begin at the Hostel page. From this page users can receive information about a particular hostel.

ONF

HOSTEL = <u>HOSTEL_ID (PK)</u>
+HOSTEL_NAME
+HOSTEL_DESCRIPTION
+EMAIL
+PHONE
+HOSTEL_LONG/LAT
+HOSTEL_REGION
+HOSTEL_COUNTRY
+HOSTEL_CITY
+HOSTEL_ADDRESS
+HOSTEL_REVIEWS {USER_NAME, USER_NATIONALITY, DATE, FACILITIES_SCORE…}
+HOSTEL_PHOTO {IMG_ID, IMG_LINK}
+HOSTEL_FACILITIES {FAC_ID, FAC_NAME, FAC_CATEGORY}

HOSTEL_ID is the primary key of this table and it is a unique value. It is minimal as it consists of just a single attribute. It is stable as there are few reasons to change the ID once it has been applied. There is also no reason to believe it will ever be null.

HOSTEL_NAME is not necessarily unique; it is a non-key attribute. In this example, ‘Glocal Backpackers Hostels’ is an instance of HOSTEL_NAME. HOSTEL_DESCRIPTION is another non-key attribute.

EMAIL and PHONE are not listed on the hostel page. However, they are produced on the booking page and so we have assumed that each hostel has these attributes assigned. Although they are unique, they should not be used as key attributes as in some cases they may be NULL. They are non-key attributes.

HOSTEL_LONG/LAT, HOSTEL_REGION, HOSTEL_COUNTRY, HOSTEL_CITY and HOSTEL_ADDRESS can be place under a subordinate entity called HOSTEL_LOCATION.

HOSTEL_LOCATION {HOSTEL_LONG/LAT, HOSTEL_REGION, HOSTEL_COUNTRY, HOSTEL_CITY, HOSTEL_ADDRESS}

A HOSTEL can have, and often does have, multiple HOSTEL_REVIEWS. Therefore, this a repeating group in 0NF.

HOSTEL_PHOTO: Each hotel can have many photos, but each photo will only relate to one hostel. This means HOSTEL_PHOTO's is a multi-valued attribute. A separate table will be created for photos with a 1:M relationship.

Each HOSTEL can have many FACILITIES and each FACILITY can be connected to many HOSTELS. This will be considered as we progress.

1NF

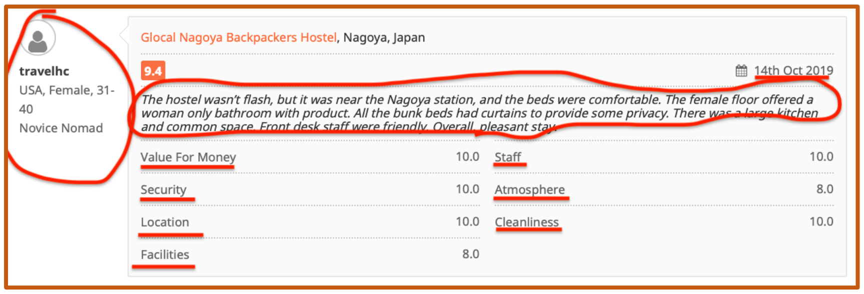
HOSTEL =	HOSTEL_ID (PK)
	+HOSTEL_NAME
	+HOSTEL_DESCRIPTION
	+EMAIL
	+PHONE

LOCATION =	LONG/LAT (PK)
	+HOSTEL_ID (PK,FK)
	+REGION
	+COUNTRY
	+CITY
	+ADDRESS

Hostel Location is made up of 4 attributes. Each REGION, COUNTRY and CITY all have pages on hostelworld.com. The total address appears as a single hyperlink. Clicking this will display a map. This leads us to believe unique co-ordinates are assigned to each hostel (LONG/LAT). These co-ordinates along with HOSTEL_ID will be used as the primary key for the table LOCATION. This composite key is necessary as there may possibly be more than one HOSTEL at a single LONG/LAT. Two hostels within one building. Note: *We will assume addresses not to be unique to a hostel (eg. door number) in order to more easily demonstrate our understanding of partial dependencies.*

REVIEWS =	<u>REVIEW_ID (PK)</u>
	+USERNAME (FK)
	+HOSTEL_ID (FK)
	+VALUE_FOR_MONEY_SCORE
	+STAFF_SCORE
	+SECURITY_SCORE
	+ATMOSPHERE_SCORE
	+LOCATION_SCORE
	+CLEANLINESS_SCORE
	+FACILITIES_SCORE
	+DESCRIPTION
	+DATE

Creating a composite key out of USERNAME and HOSTEL_ID is not an option. It would not be unique as one USER can make may reviews on the same HOSTEL. There would also be a deletion anomaly. If a user deleted their profile, and USERNAME was a PK, the review would seize to exist. The username field should change to 'anonymous' in the case of a user deleting their profile.



travelhc
USA, Female, 31-40
Novice Nomad

Glocal Nagoya Backpackers Hostel, Nagoya, Japan

9.4 14th Oct 2019

The hostel wasn't flash, but it was near the Nagoya station, and the beds were comfortable. The female floor offered a woman only bathroom with product. All the bunk beds had curtains to provide some privacy. There was a large kitchen and common space. Front desk staff were friendly. Overall, pleasant stay.

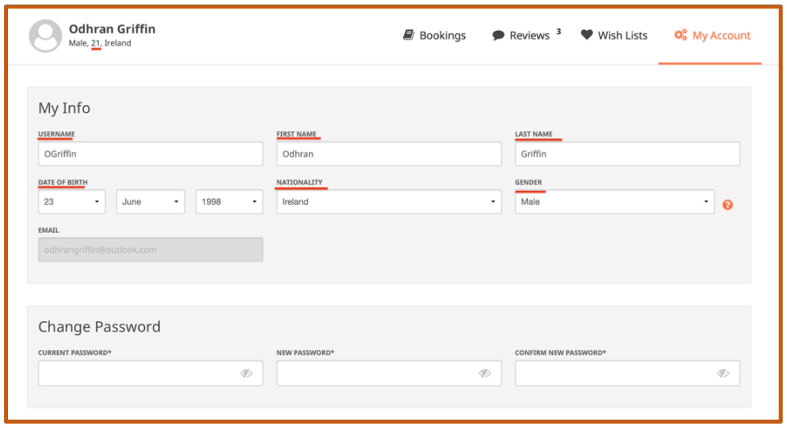
Value For Money	10.0	Staff	10.0
Security	10.0	Atmosphere	8.0
Location	10.0	Cleanliness	10.0
Facilities	8.0		

Each Hostel can have many reviews and each user can leave many reviews. These are many to many relationships. The review table may be seen as an intersection table. There is a 1:M relationship between Hostel and Review and a 1:M relationship between User and review. Each review can only link to 1 user and 1 hostel.

As seen in the screenshot above, ratings are divided into 7 categories. SUM and AVG formulas would be used top generated the overall scores and so we will not include them in our database design.

USER =	<u>USERNAME (PK)</u>
	+USER_FIRSTNAME
	+USER_SURNAME
	+USER_NATIONALITY
	+USER_DOB
	+USER_AGE
	+USER_PHONE
	+USER_EMAIL
	+USER_PHOTO

In the case of a user not uploading a photo a stock image will be used. This prevents a deletion anomaly.


Odhran Griffin
Male, 21, Ireland

My Info

USERNAME	FIRST NAME	LAST NAME
OGriffin	Odhran	Griffin

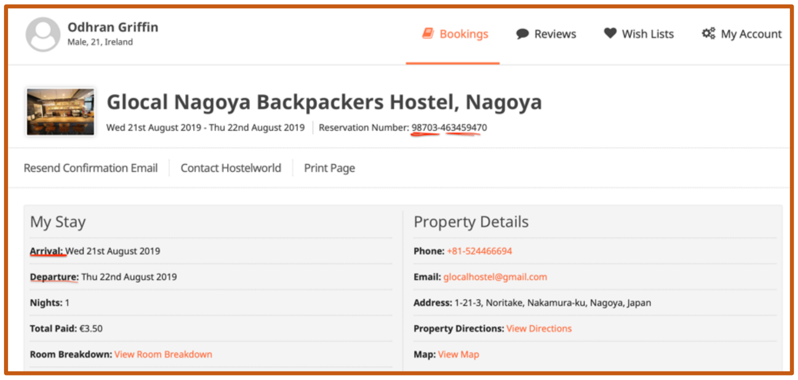
DATE OF BIRTH: 23 June 1998
NATIONALITY: Ireland
GENDER: Male
EMAIL: odhrangriffin@outlook.com

Change Password

CURRENT PASSWORD* | NEW PASSWORD* | CONFIRM NEW PASSWORD*

BOOKING = <u>BOOKING_REF (PK)</u>
+HOSTEL_ID (FK)
+USER_ID (FK)
+DATE_OF_BOOKING
+ARRIVAL_DATE
+DEPARTURE_DATE
+NUM_OF_NIGHTS

'BOOKING' is a Weak Entity as it can only exist if there is a related instance of 'HOSTEL' and 'USER'. 'HOSTEL' and 'USER' are Strong Entities as they can exist independently.
BOOKING_REF is a unique primary key. The primary key of this table cannot be a composite key made up of USERNAME and HOSTEL_ID as any one user can make many bookings in the same hostel. This would also create a deletion anomaly. If a USER or HOSTEL was to seize to exist, the bookings they were once related to must continue to exist.


Odhran Griffin
Male, 21, Ireland

Glocal Nagoya Backpackers Hostel, Nagoya
Wed 21st August 2019 - Thu 22nd August 2019 | Reservation Number: 98703-463459470

My Stay	Property Details
Arrival: Wed 21st August 2019	**Phone:** +81-524466694
Departure: Thu 22nd August 2019	**Email:** glocalhostel@gmail.com
Nights: 1	**Address:** 1-21-3, Noritake, Nakamura-ku, Nagoya, Japan
Total Paid: €3.50	**Property Directions:** View Directions
Room Breakdown: View Room Breakdown	**Map:** View Map

BOOKING, similar to REVIEW can be seen as an intersection entity. It forms a link between the M:M relationship between USER and HOSTEL to make two 1:M relationships.

HOSTEL_PHOTOS = IMG_LINK (PK)
+HOSTEL_ID (FK)

IMG_LINK can be the unique primary key as each image should only be assigned to a single hostel.

HOSTEL_FACILITIES= FAC_ID (PK, FK)
+HOSTEL_ID (PK, FK)

FACILITIES = FAC_ID (PK)
+FACILITY_NAME
+FAC_CATEGORY

On hostelworld.com, hostels choose which facilities their hostel has using check boxes. Each facility has its own unique key. Each facility can have many hostels. Each hostel can have many facilities.

2NF

HOSTEL = HOSTEL_ID (PK)
+HOSTEL_NAME
+HOSTEL_DESCRIPTION
+EMAIL
+PHONE

HOSTEL_LOCATION= LONG/LAT (PK, FK)
+HOSTEL_ID (PK, FK)

Here existed a partial dependency. REGION, COUNTRY, CITY and ADDRESS relied only on the LONG/LAT part of the composite primary key. This is resolved in 2NF by taking REGION, COUNTRY, CITY and ADDRESS out of HOSTEL_LOCATION and into a new separate table called LOCATION. This will also allow for more than one hostel to be connected to the same LONG/LAT. This is necessary as the same building could contain two hostels in some cases.

ADDRESS is listed as a single link on hostelworld.com. It is not divided into Street name, door number ect. The Hostel can write all of these in a single entry to the database. It will be connected to a link containing the long/lat and will display a map with those co-ordinates.

LOCATION = LONG/LAT (PK)
+REGION
+COUNTRY
+CITY
+ADDRESS

REVIEW =	REVIEW_ID (PK)
	+HOSTEL_ID (FK)
	+USERNAME (FK)
	+VALUE_FOR_MONEY_SCORE
	+STAFF_SCORE
	+SECURITY_SCORE
	+ATMOSPHERE_SCORE
	+LOCATION_SCORE
	+CLEANLINESS_SCORE
	+FACILITIES_SCORE
	+DESCRIPTION
	+DATE
USER =	USERNAME (PK)
	+USER_FIRSTNAME
	+USER_SURNAME
	+USER_NATIONALITY
	+USER_DOB
	+USER_AGE
	+USER_PHONE
	+USER_EMAIL
	+USER_PHOTO
BOOKING =	BOOKING_REF (PK)
	+HOSTEL_ID (FK)
	+USER_ID (FK)
	+DATE_OF_BOOKING
	+ARRIVAL_DATE
	+DEPARTURE_DATE
	+NUM_OF_NIGHTS
HOSTEL_PHOTOS =	HOSTEL_ID (PK, FK)
	+IMG_LINK
HOSTEL_FACILITIES=	FAC_ID (PK, FK)
	+HOSTEL_ID (PK, FK)
FACILITIES =	FAC_ID (PK)
	+FACILITY_NAME
	+FAC_CATEGORY

3NF

HOSTEL =	HOSTEL_ID (PK)
	+HOSTEL_NAME
	+HOSTEL_DESCRIPTION
	+EMAIL
	+PHONE
HOSTEL_LOCATION=	LONG_LAT (PK, FK)
	+HOSTEL_ID (PK, FK)
LOCATION =	LONG/LAT (PK)
	+REGION
	+COUNTRY
	+CITY
	+ADDRESS
REVIEW =	REVIEW_ID (PK)
	+HOSTEL_ID (FK)
	+USERNAME (FK)
	+VALUE_FOR_MONEY_SCORE
	+STAFF_SCORE
	+SECURITY_SCORE
	+ATMOSPHERE_SCORE
	+LOCATION_SCORE
	+CLEANLINESS_SCORE
	+FACILITIES_SCORE
	+REVIEW_DESCRIPTION
	+DATE_REVIEW

The numeric data type 'bit' will be used for the 'score' attributes. Instances of these cannot be >10. They will be stored as values between 0 and 1 and multiplied by 10 using an algorithm when displayed. (E.g. 0.71 = 7.1.) They cannot be int as they are not whole numbers. They must be used for calculations so cannot be varchar.

USER =	USERNAME (PK)
	+USER_FIRSTNAME
	+USER_SURNAME
	+USER_NATIONALITY
	+USER_DOB
	+USER_PHONE
	+USER_EMAIL
	+USER_PHOTO

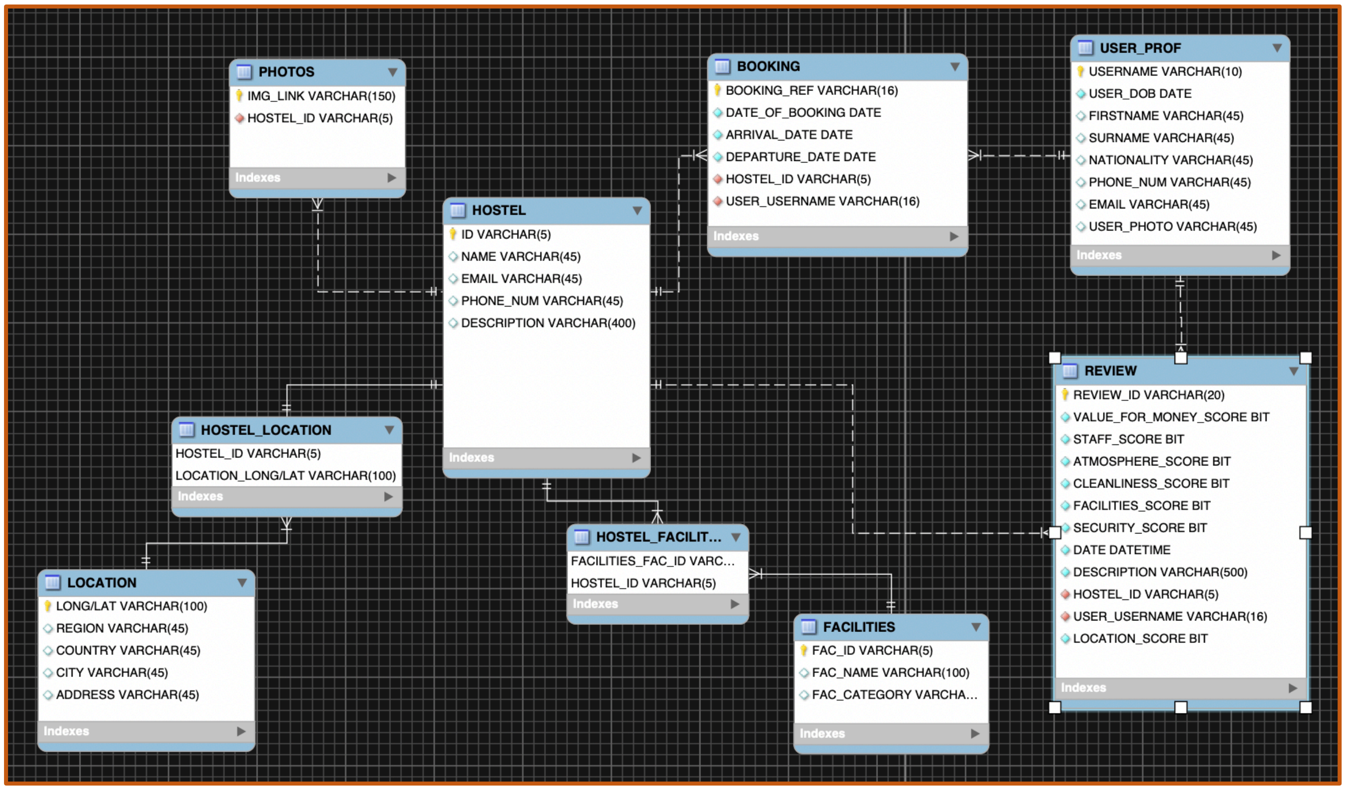
USER_AGE, a non-key attribute, is derived from USER_DOB which is a non-key attribute. This will be mitigated as we progress to third normal form. The age of the user can be generated by an algorithm outside of our database but by using information from the database and therefor there is no need to hold USER_AGE data on the database.

BOOKING =	BOOKING_REF (PK)
	+HOSTEL_ID (FK)
	+USERNAME (FK)
	+DATE_OF_BOOKING
	+DATE_OF_ARRIVAL
	+DATE_OF_DEPARTURE

There is another derived field in the BOOKING_REF table. NUM_OF_NIGHTS is the difference between ARRIVAL_DATE and DEPARTURE_DATE. There is no need to store this as it is a simple computation which can be easily generated during use.

HOSTEL_PHOTOS =	IMG_LINK (PK)
	+HOSTEL_ID (FK)
HOSTEL_FACILITIES=	FAC_ID (PK, FK)
	+HOSTEL_ID (PK, FK)
FACILITIES =	FAC_ID (PK)
	+FACILITY_NAME
	+FAC_CATEGORY

Entity Relationship diagram



<u>Created using MySQLWorkbench</u>

SQL Code

```
# Database of hostelworld.com.

DROP TABLE IF EXISTS HOSTEL;
DROP TABLE IF EXISTS BOOKING;
DROP TABLE IF EXISTS USER_PROF;
DROP TABLE IF EXISTS REVIEW;
DROP TABLE IF EXISTS FACILITIES;
DROP TABLE IF EXISTS LOCATION;
DROP TABLE IF EXISTS PHOTOS;
DROP TABLE IF EXISTS HOSTEL_LOCATION;
DROP TABLE IF EXISTS HOSTEL_FACILITIES;


CREATE TABLE HOSTEL (
   HOSTEL_ID       VARCHAR(5) UNIQUE,
   HOSTEL_NAME     VARCHAR(45) CHARACTER SET UTF8MB4 NOT NULL,
   EMAIL           VARCHAR(45) CHARACTER SET UTF8MB4 UNIQUE,
   PHONE_NUM       VARCHAR(45) CHARACTER SET UTF8MB4,
   DESCRIPTION     VARCHAR(400) CHARACTER SET UTF8MB4,
   PRIMARY KEY(HOSTEL_ID) );


CREATE TABLE USER_PROF (
   USERNAME        VARCHAR(45) UNIQUE,
   USER_FIRSTNAME  VARCHAR(45) CHARACTER SET UTF8MB4,
   USER_SURNAME    VARCHAR(45) CHARACTER SET UTF8MB4,
   USER_DOB        DATE NOT NULL,
   USER_NATIONALITY    VARCHAR(45),
   USER_NUMBER         VARCHAR(45),
   USER_EMAIL          VARCHAR(45) CHARACTER SET UTF8MB4 UNIQUE,
   USER_PHOTO          VARCHAR(45),
   PRIMARY KEY(USERNAME) );


CREATE TABLE BOOKING (
   BOOKING_REF VARCHAR(16) UNIQUE,
   HOSTEL_ID   VARCHAR(5),
   USERNAME    VARCHAR(16) CHARACTER SET UTF8MB4,
   DATE_OF_BOOKING     DATE,
   DATE_OF_ARRVIAL     DATE,
   DATE_OF_DEPARTURE   DATE,
   PRIMARY KEY (BOOKING_REF),
   FOREIGN KEY (HOSTEL_ID) REFERENCES HOSTEL (HOSTEL_ID),
   FOREIGN KEY (USERNAME) REFERENCES USER_PROF (USERNAME) );
   CREATE TABLE REVIEW  (
   REVIEW_ID   VARCHAR(20) UNIQUE,
   HOSTEL_ID   VARCHAR(5),
   USERNAME    VARCHAR(16) CHARACTER SET UTF8MB4,
   VALUE_FOR_MONEY_SCORE   bit,
   STAFF_SCORE            bit,
   SECURITY_SCORE         bit,
   ATMOSPHERE_SCORE       bit,
   LOCATION_SCORE         bit,
   CLEANLINESS_SCORE      bit,
   FACILITIES_SCORE       bit,
   DATE_REVIEW            DATE,
   REVIEW_DESCRIPTION     VARCHAR(500),
   PRIMARY KEY (REVIEW_ID),
   FOREIGN KEY (HOSTEL_ID) REFERENCES HOSTEL (HOSTEL_ID),
   FOREIGN KEY (USERNAME) REFERENCES USER_PROF (USERNAME) );


CREATE TABLE PHOTOS (
   IMG_LINK    VARCHAR(150),
   HOSTEL_ID   VARCHAR(5),
   PRIMARY KEY (IMG_LINK),
   FOREIGN KEY (HOSTEL_ID) REFERENCES HOSTEL (HOSTEL_ID) );

CREATE TABLE LOCATION (
   LONG_LAT    VARCHAR(100) CHARACTER SET UTF8MB4,
   REGION      VARCHAR(45),
   COUNTRY     VARCHAR(45),
   CITY        VARCHAR(45) CHARACTER SET UTF8MB4,
   ADDRESS     VARCHAR(100) CHARACTER SET UTF8MB4,
   PRIMARY KEY (LONG_LAT) );

CREATE TABLE HOSTEL_LOCATION (
   HOSTEL_ID   VARCHAR(5),
   LONG_LAT    VARCHAR(100) CHARACTER SET UTF8MB4,
   PRIMARY KEY (LONG_LAT, HOSTEL_ID),
   FOREIGN KEY (HOSTEL_ID) REFERENCES HOSTEL (HOSTEL_ID),
   FOREIGN KEY (LONG_LAT) REFERENCES LOCATION (LONG_LAT) );


CREATE TABLE FACILITIES (
   FAC_ID         VARCHAR(5) UNIQUE,
   FAC_NAME       VARCHAR(100),
   FAC_CATEGORY   VARCHAR(20),
   PRIMARY KEY (FAC_ID) );

CREATE TABLE HOSTEL_FACILITIES (
   FAC_ID      VARCHAR(5),
   HOSTEL_ID   VARCHAR(5),
   PRIMARY KEY (FAC_ID, HOSTEL_ID),
   FOREIGN KEY (FAC_ID) REFERENCES FACILITIES (FAC_ID),
   FOREIGN KEY (HOSTEL_ID) REFERENCES HOSTEL (HOSTEL_ID) );
```