|  |  |
| --- | --- |
| **customer** | * Customer account table relies on sub tables (personal\_id and address). * Separate personal\_id table allows guest table access as part of booking table without the need for a customer account. Reflects real-life hotels.com. |
| Hotel |  |
| Theme |  |
| Attraction |  |
| Room | * Need to create a way to link multiple rooms to a single booking |
| Room Amenities | * Decided to simplify both and join in one table with a column for amenity type instead, i.e. hotel or room. |
| Hotel Amenities |
| Price | * Price\_variation table should be laid out with ‘name’ and ‘rate’ column. This will feed into other tables. * ~~Prive\_variation will likely be many to many as multiple rates may be applied to one or many rooms.~~ * Actually, instead of needing to add multiple variations to price total table, e.g. weekend + Christmas, I could just create separate Christmas Weekend category. |
| Payment | * Payment option table added with various payment options, e.g. pay now vs. pay later. * Initially this was to be a one-to-many relationship with room table but real-life experience would suggest that on hotels.com site some rooms have different payment options which also link to different price rates. |
| Booking | * It seems common sense to use booking\_id in itself as a traditional booking reference as it probably more accurately reflects |
| Accessibility |  |
| Covid Policy | * Decided to simplify both and join in one table with a column for general policies and providing description and/or category for each. |
| Cancellation Policy |
| FAQ |  |
| Review | * Review table: argued that it was one hotel to many reviews. * New table added to create many-to-many relationship to work out average review score for each hotel. * Instead of customer\_id in this table, booking\_id may be more appropriate and more accurately |
| Room Availability | * Difficult: inextricably linked to guest booking dates. |
| Reward |  |