In [1]:
```python
import numpy as np
import pandas as pd
```

In this part there is the importation of the file and the creation of an array that represent the id column and a dictionary to switch from the name of a month to its number

In [2]:
```python
file = pd.read_csv("hotel_bookings.csv")
arr = [i for i in range(1, file.shape[0]+1)]; ids = pd.DataFrame(arr)
months = {"January" : "01", "February" : "02", "March" : "03", "April" : "04", "May" : "05", "June" : "06", "July" : "07",
          "August" : "08", "September" : "09", "October" : "10", "November" : "11", "December" : "12"}
```

With this piece of code i create the complete date of the arrival date with three fields

In [3]:
```python
app = [str(file['arrival_date_year'][i]) + "-" + months[file["arrival_date_month"][i]] + "-" + str(file["arrival_date_day_of_mon
file = pd.concat([file, pd.DataFrame(app)], axis = 1)
file.rename(columns = {0 : "arrival_date"}, inplace = True)
file.drop(['arrival_date_year', 'arrival_date_month', 'arrival_date_week_number', 'arrival_date_day_of_month'], axis = 1, inplac
```

In this part there is the splitting of the big dataset into three smaller parts according to the entity-relationship model

In [5]:
```python
reservation = file[['is_canceled', 'lead_time', 'arrival_date', 'stays_in_weekend_nights', 'stays_in_week_nights', 'reserved_roo
                    'assigned_room_type', 'booking_changes', 'deposit_type', 'days_in_waiting_list', 'required_car_parking_space
                    'total_of_special_requests', 'reservation_status', 'reservation_status_date', 'adr', 'meal', 'hotel']]
guests = file[['adults', 'children', 'babies', 'previous_bookings_not_canceled', 'customer_type', 'previous_cancellations',
               'is_repeated_guest', 'country']]
third_parts = file[['agent', 'company', 'market_segment', 'distribution_channel']]
```

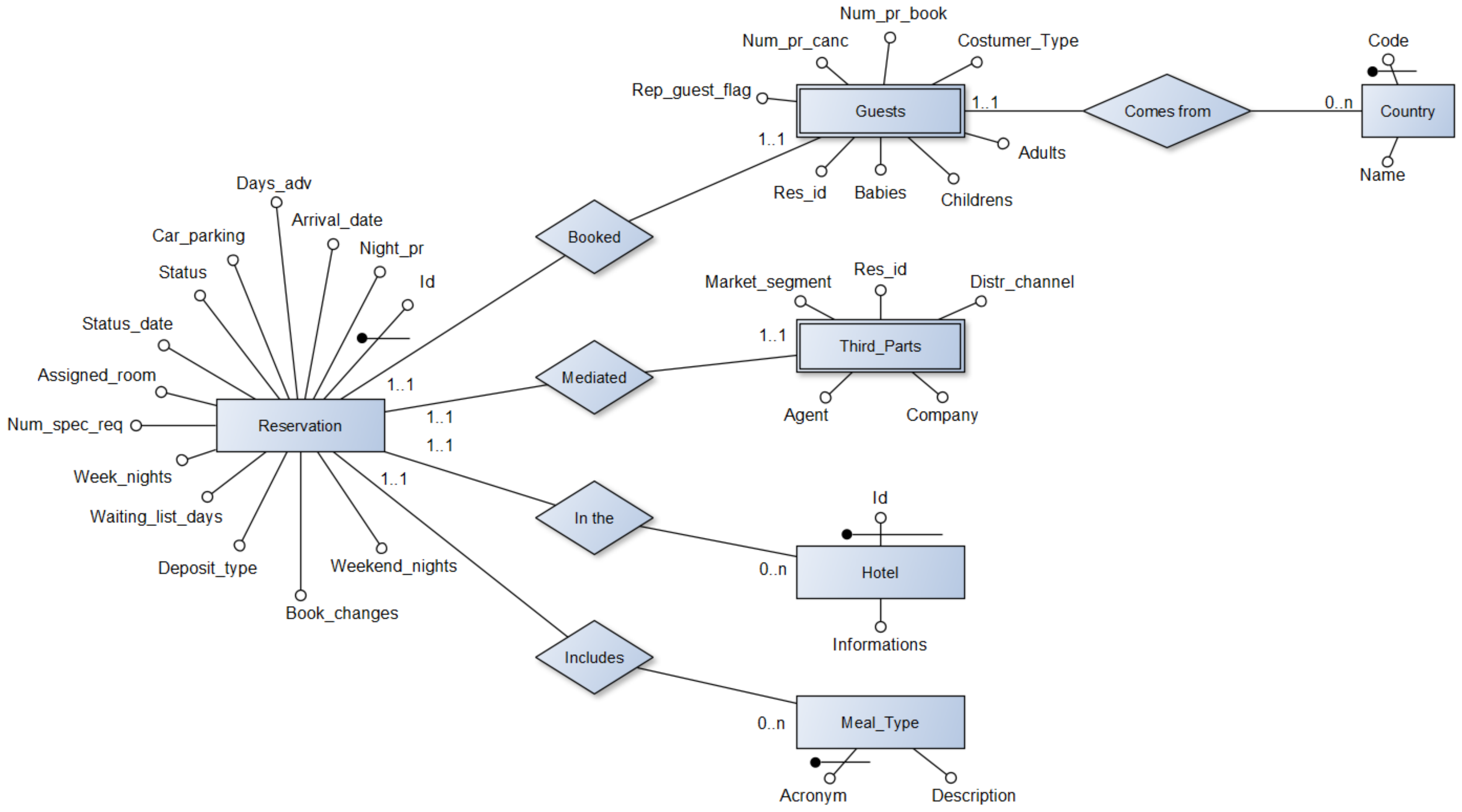
There is the concatenation between all these datasets with the id column

In [6]:
```python
reservation = pd.concat([ids, reservation], axis = 1)
reservation.rename(columns = {0 : "ID"}, inplace = True)
guests = pd.concat([ids, guests], axis = 1)
guests.rename(columns = {0 : "res_id"}, inplace = True)
third_parts = pd.concat([ids, third_parts], axis = 1)
third_parts.rename(columns = {0 : "res_id"}, inplace = True)
```

Exportation of these three datasets into csv files

In [7]:
```python
reservation.to_csv("C:\\Users\\asus\\Desktop\\Data Management\\First HM\\reservation.csv", index = False)
guests.to_csv("C:\\Users\\asus\\Desktop\\Data Management\\First HM\\guests.csv", index = False)
third_parts.to_csv("C:\\Users\\asus\\Desktop\\Data Management\\First HM\\third_parts.csv", index = False)
```

ER Model


Num_pr_book
Num_pr_canc
Costumer_Type
Rep_guest_flag
Guests
1..1
1..1
Adults
Res_id
Babies
Childrens
Comes from
0..n
Code
Country
Name
Days_adv
Arrival_date
Car_parking
Night_pr
Status
Id
Booked
Market_segment
Res_id
Distr_channel
Third_Parts
1..1
Status_date
Assigned_room
1..1
Mediated
Agent
Company
Num_spec_req
Reservation
1..1
1..1
Week_nights
1..1
Waiting_list_days
In the
Id
Hotel
0..n
Deposit_type
Weekend_nights
Book_changes
Informations
Includes
Meal_Type
0..n
Acronym
Description

Logic Project

Tables:

Reservation (ID, Night_pr, Arrival_date, Days_adv, Weekend_nights, Week_nights, Book_changes, Car_parking, Num_spec_req, Status, Status_date, Assigned_room, Deposit_type, Waiting_list_days, Hotel_id, Meal_acr)

Guests (Num_pres_book, Customer_type, Adults, Children, Babies, Rep_guest_flag, Num_pr_canc, Res_id, Country_code)

Third_Parts (Agent, Company, Market_segment, Distr_channel, Res_id)

Country (Code, Name)

Hotel (ID, informations)

Meal_Type (Acronym, Description)

Foreign key:

Reservation (Hotel_id) ----------> Hotel (ID)

Reservation (Meal_acr) ----------> Meal (Acronym)

Guests (Country_code) -----------> County (Code)

Guests (Res_id) -----------> Reservation (ID)

Third_Parts (Res_id) -----------> Reservation (ID)

The Queries:

Usefulness

Readability

Wide usage of views.
Useful for complex
Queries.

Queries aim at:

- Provide a clearer view to **hotel managers**
- Give some hints to **bookers**

Optimization:

The query optimization was done through:

- ***Temporary tables***
- ***Indexes***