

Booking2

1. Abstract

Booking2.com è un sito web dedicato alla prenotazione di camere di hotel, e offre una vasta selezione di strutture agli utenti di tutto il mondo. Essi possono cercare e confrontare diverse opzioni di alloggio in base alla destinazione desiderata, le date del soggiorno e anche le preferenze personali, ad esempio come volontà di avere una suite di lusso. Il sito fornisce dettagli completi sugli alloggi disponibili, le attività praticabili, i servizi inclusi e le recensioni degli ospiti. Gli utenti possono prenotare direttamente tramite il sito, ed è possibile selezionare opzioni di pagamento e cancellazione. Booking2.com offre un'esperienza utente intuitiva e una vasta gamma di scelte di alloggio, rendendolo uno strumento prezioso per chiunque voglia organizzare viaggi e soggiorni in modo semplice ed efficiente.

2. Analisi dei requisiti

2.1 Descrizione testuale

Nella base di dati sono salvati i dati degli **Utenti** registrati all'applicazione, di ogni utente sono noti:

- Id_Utente
- Nome
- Cognome
- Email
- Password
- Telefono
- Carta_Debito

Un utente può diventare “premium” tramite un abbonamento mensile. L'utente premium ha diritto ad uno sconto sulle attività e sul pernottamento.

Le **Carte di debito** dei clienti sono memorizzate e per ognuna si sanno questi dati:

- Numero_Carta
- Intestatario
- CVC
- Data_di_Scadenza

I pagamenti ed i rimborsi non sono gestiti internamente al sito di booking2.com

Gli **Hotel** fornisce le informazioni riguardanti:

- Indirizzo: (Stato, Città, Via, numero civico, CAP)
- Nome
- Email
- Telefono

Le **Prenotazioni** degli utenti per stanze e attività hanno questi parametri:

- Id_Prenotazione
- Numero_di_Ospiti
- Rimborsabile

- DataOraInizio
- DataOraFine
- Utente
- Camera
- Prezzo

Per ogni prenotazione si ha un codice univoco che la identifica, un codice dell'utente che l'ha effettuata, il suo costo e viene indicato se essa è rimborsabile o meno.

Ogni **Camera** di hotel appartiene ad una singola struttura e memorizza i seguenti parametri:

- Id_Camera
- Costo_Giornaliero
- Num_Max_Ospiti
- Hotel

Le camere possono essere camere ordinarie oppure suite, e nel caso delle ultime possono essere offerti degli omaggi al loro interno.

Gli **Omaggi** sono una lista di beni inclusi con una prenotazione di una suite, per cui per ogni omaggio bisogna conoscere:

- Id_Omaggio
- Nome
- Suite

In questo modo possiamo associare gli omaggi alle rispettive suite.

I **Servizi** offerti dagli Hotel hanno diversi parametri tra cui:

- Id_Servizio
- Nome
- Hotel

Le **Attività** comprendono una lista di informazioni utili per identificarle:

- Id_Actività
- Nome
- Prezzo
- Voto
- Hotel

Ogni attività può essere affiliata ad un hotel o semplicemente può essere proposta da essi. L'affiliazione comporta uno sconto nella prenotazione dell'attività.

Su ogni prenotazione effettuata l'utente può lasciare una **Recensione** che viene memorizzata nel database, la recensione relativa all'hotel contiene anche un punteggio che va da 1 a 10. Se la prenotazione comprendeva anche delle attività, la recensione potrà includere anche un punteggio da 1 a 10 relativo ad esse. L'utente può decidere se scrivere anche un commento di lunghezza massima di 200 caratteri. I parametri memorizzati dalla recensione saranno quindi:

- Commento
- Prenotazione
- Attività
- Hotel

2.2 Glossario dei termini

Termine	Descrizione	Collegamento
Utente	Un utente è colui che effettua una prenotazione di una camera di un hotel, ed è libero di scegliere di prenotare o meno una o più attività.	Prenotazione, Attività, CartadiDebito
Premium	Un utente è Premium se è sottoscritto ad un abbonamento a booking2.com	Entità figlia di Utente
Carta di debito	Metodo di pagamento utilizzato per pagare le prenotazioni. Ne fanno parte anche carte di credito, carte prepagate e Google Pay	Utente, Prenotazione, Attività
Attività	Le attività possono essere intrattenimento o svago che un utente può pagare e svolgere durante il suo soggiorno	CartadiDebito, Utente, Hotel
Affiliata	Attività affiliata all'hotel in cui si ha la prenotazione e che quindi permette uno sconto su di essa	Entità figlia di Attività
Esterna	Attività non legata all'hotel in cui si ha la prenotazione e che quindi non ha vantaggi	Entità figlia di Attività
Prenotazione	La prenotazione è il primo atto di stipula del contratto di albergo e impegna al suo rispetto sia l'azienda che il cliente	CartadiDebito, Camera, Recensione
Recensione	Essa comprende un voto da 1 a 10 e la possibilità di lasciare un commento	Prenotazione, Hotel
Recensione Attività	Essa indica una recensione relativa ad un'attività, invece che ad una stanza	Attività, Entità figlia di Recensione
Hotel	Un hotel è una struttura in cui si può pernottare che comprende camere, servizi e attività	Servizi, Recensione, Camera, Attività
Servizi	Lista di benefici compresi all'interno della camera e della struttura	Hotel
Camera	Oggetto della prenotazione, ha un costo e un numero di persone massimo	Hotel, Prenotazione

Ordinaria	Camera con una minore quantità di servizi e omaggi, ma a costo minore	Entità figlia di Camera
Suite	Camera con una maggiore quantità di servizi e omaggi, ma a costo maggiore	Omaggi, Entità figlia di Camera
Omaggi	Prodotti di vario tipo in regalo all'interno della Suite	Suite

2.3 Operazioni più frequenti

In totale, ci sono più di 28 milioni di stanze disponibili sul sito di Booking2.com

Operazione	Tipo	Frequenza
Controllo di camere disponibili	L	50000 al giorno
Inserimento di un nuovo utente	S	1000 al giorno
Inserimento di una nuova prenotazione	S	10000 al giorno
Annullamento di una prenotazione esistente	S	100 al giorno
Controllo di una lista di attività affiliate ad un hotel	L	500 al giorno
Controllo di servizi di un hotel	L	5000 al giorno
Aggiornamento dei servizi di un hotel	S	12 all'anno
Controllo omaggi di una suite	L	50 alla settimana
Inserimento di una camera	S	250 al mese
Aggiornamento valutazione media delle attività	S	12 al giorno
Inserimento di una recensione relativa ad un'attività	S	200 al giorno

3. Progettazione concettuale

3.1 Lista entità

Se non specificato, l'attributo è NOT NULL

- Utente
 - **Id_Utente** int PRIMARY KEY,
 - Nome varchar(20),
 - Cognome varchar(20),
 - Email varchar(30),
 - Password varchar(50),
 - Telefono int,
 - Premium boolean,
 - Sconto float

L'entità Utente si specializza in una sottocategoria:
→ Premium
- Carta di debito
 - **Numero_carta** bigint PRIMARY KEY,
 - CVC smallint,
 - Intestatario varchar(40),
 - Data_di_scadenza date
- Prenotazione
 - **Id_prenotazione** int PRIMARY KEY,
 - Numero_di_ospiti int,
 - Prezzo int
- Camera
 - **Id_Camera** int PRIMARY KEY,
 - Costo_giornaliero int,
 - Num_Max_Ospiti int)

L'entità camera si specializza in due sottocategorie:
→ Ordinaria
→ Suite
- Omaggi:
 - **Id_Omaggio** int PRIMARY KEY,
 - Nome varchar(50)
- Hotel
 - Nome varchar(30),
 - Email varchar(30),
 - Telefono int,

La chiave primaria di Hotel è l'attributo composto:

- **Indirizzo:**
 - CAP: char (5) PRIMARY KEY,
 - Via: varchar (50) PRIMARY KEY,
 - N_civico: varchar (10) PRIMARY KEY,
 - Stato: varchar (20),
 - Città: varchar (50)
- Servizi
 - **Id_Servizio** int PRIMARY KEY,
 - Nome varchar(40)
- Attività
 - **Id_Actività** int PRIMARY KEY
 - Nome varchar(30),
 - Prezzo int,
 - Voto int

L'entità Attività si specializza in due sottocategorie:

- Affiliata
- Esterna

- Recensione
 - Commento varchar (200)

L'entità Recensione si specializza in una sottocategoria:

- Recensione Attività

3.2 Tabella delle relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Metodo di pagamento	Utente(1,1), Carta di debito (1,1)	Un utente deve registrare un metodo di pagamento, ogni metodo di pagamento deve essere associato ad un singolo utente	Nessuno
Effettua	Utente (1,N), Prenotazione(1,1)	Un utente può effettuare più prenotazioni, ogni prenotazione può essere effettuata ad un singolo utente	Nessuno
Transazione	Carta Di Debito (0,N), Prenotazione(1,1)	Una carta di debito può essere usata per zero o più prenotazioni, ogni prenotazione deve presentare una carta di debito	Nessuno
Acquistare	Carta Di Debito(0,N), Attività(0,N)	Una carta di debito può essere usata per acquistare zero o più attività, le attività possono essere o non essere acquistate da più carte di debito	Nessuno

Scelta	Utente (0,N), Attività (0,N)	Gli utenti possono scegliere da zero o più attività, le attività possono essere scelte da zero o più utenti	Nessuno
Relativa	Prenotazione(1,1), Camera(1,1)	Ogni prenotazione è associata a una e una sola camera, e ogni camera può essere associata a una e una sola prenotazione	Nessuno
Valutazione	Prenotazione (0,N), Recensione(1,1)	Le prenotazioni essere valutate da zero o più recensioni, ogni recensione valuta una singola prenotazione	Nessuno
Voto	Recensione(1,1), Hotel (0,N)	Una recensione contiene il voto da assegnare ad un hotel, a un hotel sono assegnati voti di zero o più recensioni	Punteggio: int ≥ 1 , ≤ 10
VotoAttività	Attività (0,N), RecensioneAttività(1,1)	Un'attività può avere zero o più recensioni, ogni recensione deve riguardare un'attività	Punteggio: int ≥ 1 , ≤ 10
Proporre	Attività (N,M), Hotel (0,N)	Un hotel può proporre zero o più attività, ogni attività può essere proposta da hotel diversi	Nessuno
Fornisce	Hotel (1,N), Servizi (N,M)	Un hotel può fornire uno o più servizi, mentre più servizi possono essere offerti da hotel diversi	Nessuno
Contiene	Hotel (1,N), Camera (1,1)	Un hotel contiene una o più camere, ogni camera appartiene ad un singolo hotel	Nessuno
Offerta	Suite(1,N), Omaggi(N,M)	Ogni suite offre almeno un omaggio, diversi omaggi possono essere offerti da più suite	Nessuno

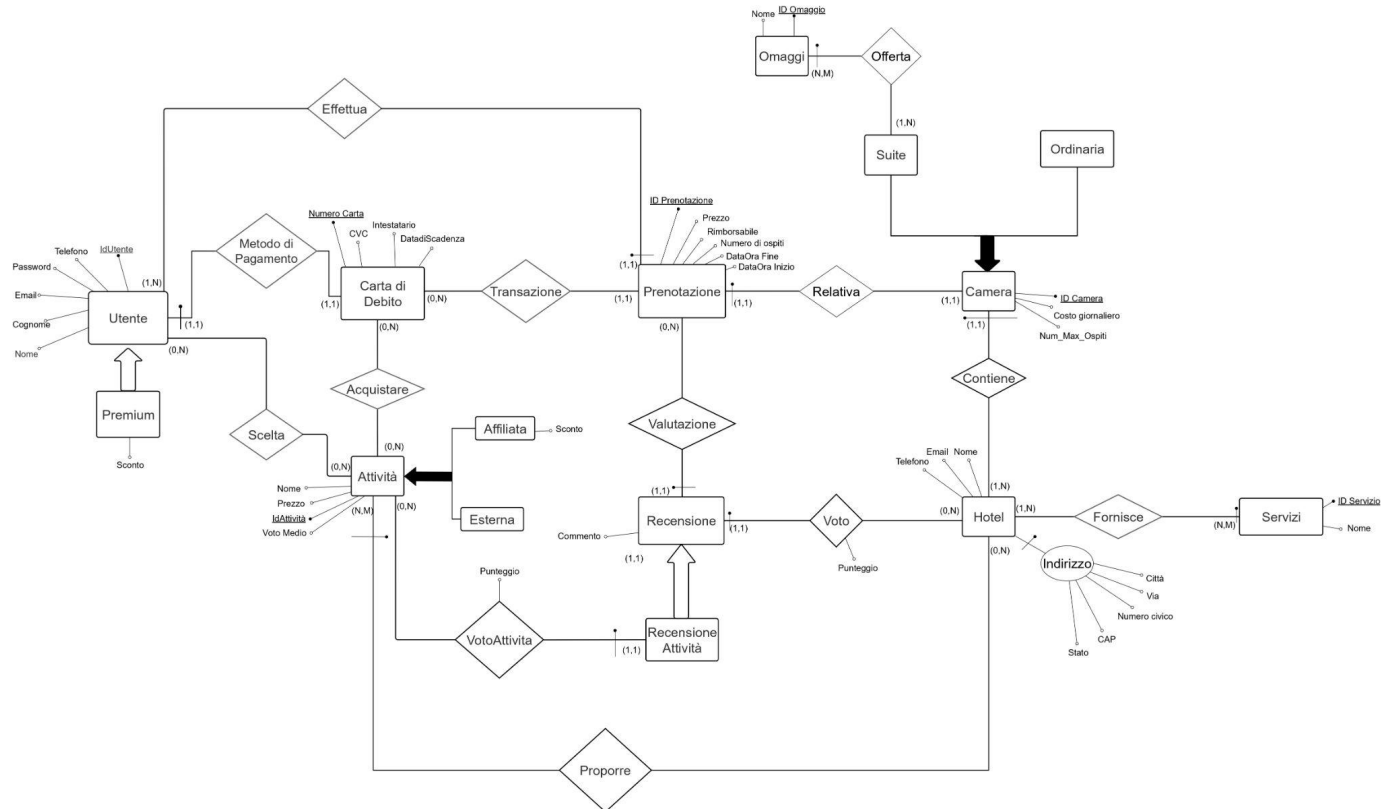
Vincoli non rappresentabili tramite schema E-R:

- Una camera prenotata non può essere prenotata da altri utenti;
- Una recensione non può essere lasciata da persone che non sono utenti del sito.
- L'attributo DataOra_Fine non può essere minore dell'attributo DataOra_Inizio:
- Il numero di ospiti per effettuare una prenotazione non può essere maggiore del numero massimo di ospiti della stanza.

Vincoli di derivazione:

- L'attributo Voto dell'entità Attività è dato dalla media dei punteggi delle recensioni relative ad un'attività.
- L'attributo prezzo dell'entità Prenotazione è uguale al costo di una stanza prenotata moltiplicato per il numero di giorni della prenotazione.

3.3 Schema ER concettuale



4. Progettazione Logica

4.1.1 Analisi delle Ridondanze

L'attributo “**voto**” dell'entità Attività è ridondante in quanto può essere calcolato facendo la media dei suoi punteggi relativi. Considerando che l'attributo viene spesso visualizzato da uno o più utenti, ricalcolarlo ogni volta tramite la media di tutti i punteggi delle recensioni sarebbe meno efficiente rispetto a lasciarlo nel database.

L'attributo “**prezzo**” dell'entità Prenotazione è ridondante in quanto può essere calcolato moltiplicando l'attributo costo giornaliero della stanza per la quantità di giorni per cui è stata prenotata. Esiste la possibilità di creare inconsistenze causate da un cambio di prezzo della camera; ad esempio: l'importo di una prenotazione avvenuta in passato, riguardante quella stessa camera e calcolato con il metodo appena spiegato, sarà diverso dal prezzo attuale per la prenotazione della stessa. Il dato del prezzo pagato in passato sarà salvato dal sito che si occupa delle transazioni.

In conclusione, rimuoviamo l'attributo prezzo nell'entità Prenotazione.

4.1.2 Eliminazione delle Generalizzazioni

La prima generalizzazione che prendiamo in esame è:

- Padre: Camera (**generalizzazione totale**);
Figlie: Ordinaria, Suite.

Risoluzione: Le entità figlie vengono accorpate nel padre. La scelta di questa soluzione è stata contribuita dal fatto che facendo così l'accesso sarà contestuale agli attributi del padre e delle figlie, e siccome le entità figlie non hanno attributi propri non ci sarà spreco di memoria per i valori nulli.

Viene aggiunto un attributo booleano "Suite" all'entità padre Camera se "Suite" == true allora si riferisce a una camera Suite e altrimenti si riferisce ad una camera ordinaria. Ora la relazione "Offerta" coinvolge le entità Omaggi e Camera.

La seconda generalizzazione che consideriamo è:

- Padre: Attività(**generalizzazione totale**);
Figlie: Affiliata, Esterna.

Risoluzione: Le entità figlie vengono accorpate nel padre. È stata fatta questa scelta perché le figlie rappresentano lo stesso concetto. L'unico contro sarà lo spreco di memoria per gli attributi con valore nullo, ma in confronto alla situazione generale questa soluzione è ideale.

Viene aggiunto un attributo booleano "Affiliata" all'entità padre Attività tale che con "Affiliata" == true allora si riferisce ad un'attività che è affiliata all'hotel in cui si ha effettuato la prenotazione, altrimenti si riferisce ad un'attività non convenzionata all'hotel. Verrà aggiunto in seguito l'attributo "sconto" che potrà essere nullo se l'attività non è affiliata.

La terza generalizzazione che consideriamo è:

- Padre: Recensione (**generalizzazione parziale**);
Figlie: Recensione Attività.

Risoluzione: l'entità RecensioneAttività viene accorpata in Recensione in quanto gli accessi alle due entità sono contestuali, la relazione Voto che prima collegava RecensioneAttività ad Attività ora collega le entità Recensione a Attività mantenendo sempre l'attributo punteggio.

La quarta generalizzazione che consideriamo è:

- Padre: Utente (**generalizzazione parziale**);
Figlie: Premium.

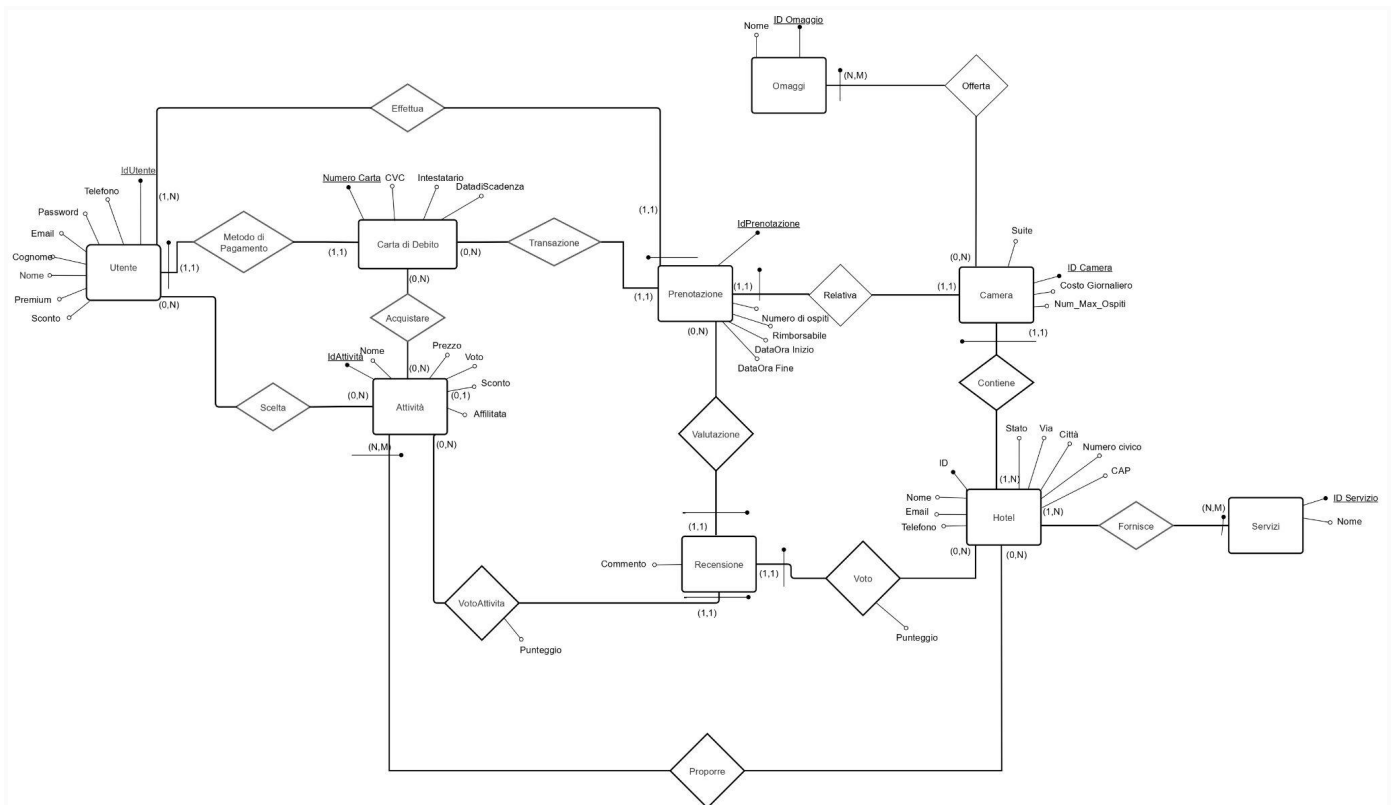
Risoluzione: L'entità figlia viene accorpata nel padre. Essa rappresenta infatti un particolare tipo di utente. Un fattore negativo sarà lo spreco di memoria per gli attributi con valore nullo, ma in confronto alla situazione generale questa soluzione è ideale.

Viene aggiunto un attributo booleano "Premium" all'entità padre Utente tale che con "Premium" == true allora si riferisce ad un utente il quale è sottoscritto ad un abbonamento premium, altrimenti si riferisce ad un utente non abbonato. Verrà aggiunto in seguito l'attributo "sconto" che potrà essere nullo se l'utente non è premium.

4.1.3 Scelta identificatori primari

Per l'entità Hotel era stato scelto di utilizzare l'attributo composto indirizzo come identificatore primario, in quanto non possono essere presenti diversi hotel in uno stesso luogo fisico. In questo però, modo le tabelle Camera, Servizi, Attività, e Recensione avrebbero dovuto memorizzare i cinque campi dell'indirizzo come chiave esterna, e per la quantità possibile di stanze questo vorrebbe dire un enorme utilizzo di memoria; quindi abbiamo deciso di introdurre l'attributo ID come chiave primaria per risolvere questo problema.

4.1.4 Schema ER ristrutturato



4.2 Creazione delle tabelle

(A→B indica che B è chiave esterna di A)

Utente: (Id_Utente, Email, Password, Nome, Cognome, Telefono, Carta_Debito→ Carta_di_Debito.Numero_Carta, Premium, Sconto)

Carta di Debito: (Numero_Carta, CVC, Intestatario, Data di Scadenza)

Hotel: (Id_Hotel, Nome, Email, Telefono, Stato, Via, Città, Cap, N_Civico)

Attività: (Id_Actività, Nome, Voto, Sconto, Affiliata, Prezzo, Hotel→Hotel.Id_Hotel)

Camera: (Id_Camera, Costo_Giornaliero, Num_Max_Ospiti, Suite, Hotel→ Hotel.Id_Hotel)

Prenotazione: (Id_Prenotazione, Numero_di_ospiti, Rimborsabile, DataOraInizio, DataOraFine, Utente→Utente.Id_Utente, Camera→Camera.Id_Camera)

Recensione: (Commento, Attività→Attività.Id_Actività, Prenotazione→Prenotazione.Id_Prenotazione, Hotel→Hotel.Id_Hotel)

Servizi: (Nome, Id_Servizio)

HotelServizi: (Hotel→Hotel.Id_Hotel, Servizio→Servizi.Id_Servizio)

Omaggi: (Nome, Id_Omaggio)

OmaggiSuite: (Omaggio→Omaggi.Id_Omaggio, Id_Suite→Camera.Id_Camera)

5. Query e indici

5.1 Lista query:

1. Mostrare il numero di hotel presenti in uno stesso stato;

```
SELECT Stato, COUNT(*) AS NumeroHotel
FROM Hotel
GROUP BY Stato
ORDER BY Stato;
```

stato character varying (20) 🔒	numerohotel bigint 🔒
Australia	1
Austria	1
Francia	2
Indonesia	1
Italia	2
Switzerland	1
UK	1
USA	4

2. Mostrare quante suite ha ciascun hotel e il numero minimo di omaggi che vengono offerti da ciascun hotel;

```

SELECT H. Id_Hotel, H. Nome AS Nome_Hotel, COUNT (C. Id_Camera) AS Numero_Suite, MIN (O. Numero_Omaggi) AS Numero_Minimo_Omaggi
FROM Hotel H
LEFT JOIN Camera C ON H.Id_Hotel = C. Hotel AND C. Suite = true
LEFT JOIN (SELECT OS. Id_Suite, COUNT (OS.Id_Omaggio) AS Numero_Omaggi
FROM OmaggiSuite OS
GROUP BY OS.Id_Suite
) O ON C.Id_Camera = O.Id_Suite
GROUP BY H.Id_Hotel, H.Nome
ORDER BY H.Id_Hotel;

```

	id_hotel [PK] integer	nome_hotel character varying (30)	numero_suite bigint	numero_minimo_omaggi bigint
2	2	Hotel Roma	1	3
3	3	Hotel Paris	1	3
4	4	Grand Hotel	1	4
5	5	Hotel de Ville	1	4
6	6	Beach Resort	1	4
7	11	Tropical Paradise Resort	1	5
8	13	Seaview Hotel	1	5
9	8	Seaside Inn	1	5
10	9	Mountain Retreat	1	5
11	10	City Center Hotel	1	5
12	7	Alpine Lodge	0	[null]
13	12	Historic Mansion	0	[null]

3. Mostrare gli hotel che hanno almeno due attività affiliate se il loro costo è maggiore di un valore x, che nell'esempio è 15;

```

SELECT h.Id_Hotel, h.Nome AS NomeHotel, COUNT(a.Id_Attivita) AS NumeroAttivitaAffiliate
FROM Hotel h
JOIN Attivita a ON h.Id_Hotel = a.Hotel
WHERE a.Affiliata = true AND a.Prezzo > 15
GROUP BY h.Id_Hotel, h.Nome
HAVING COUNT(a.Id_Attivita) >= 2;

```

id_hotel [PK] integer	nomehotel character varying (30)	numeroattivitaaffiliate bigint
7	Alpine Lodge	2
5	Hotel de Ville	2

4. Mostrare l'utente con più prenotazioni di camere e l'importo totale pagato;

```

SELECT u.Id_Utente, u.Nome, u.Cognome,
COUNT(p.Id_Prenotazione) AS NumeroPrenotazioni,
SUM(c.Costo_giornaliero * DATE_PART('day', p.DataOraFine - p.DataOraInizio)) AS ImportoTotalePagato
FROM Utente u
JOIN Prenotazione p ON u.Id_Utente = p.Utente
JOIN Camera c ON p.Camera = c.Id_Camera
GROUP BY u.Id_Utente, u.Nome, u.Cognome
ORDER BY NumeroPrenotazioni DESC
LIMIT 1;

```

	id_utente [PK] integer	nome character varying (20)	cognome character varying (20)	numeroprenotazioni bigint	importototalepagato double precision
1	15	Giorgia	Gialli	3	1600

5. Mostrare le 3 prenotazioni di durata più lunga dell'anno x, nell'esempio l'anno 2022:

```
SELECT P.Id_Prenotazione, U.Nome AS Nome_Utente, U.Cognome AS Cognome_Utente, P.DataOraInizio, P.DataOraFine,
       ROUND(EXTRACT(EPOCH FROM (P.DataOraFine - P.DataOraInizio)) / (60 * 60 * 24)) AS Durata_Giorni
FROM Prenotazione P
JOIN Utente U ON P.Utente = U.Id_Utente
WHERE P.DataOraInizio >= '2022-01-01' AND P.DataOraFine <= '2022-12-31'
ORDER BY Durata_Giorni DESC
LIMIT 3;
```

id_prenotazione	nome_utente	cognome_utente	dataorainizio	dataorafine	durata_giorni
integer	character varying (20)	character varying (20)	timestamp without time zone	timestamp without time zone	numeric
25	Linda	Marrone	2022-11-28 14:00:00	2022-12-03 12:00:00	5
32	Linda	Marrone	2022-11-28 14:00:00	2022-12-03 12:00:00	5
35	Giorgia	Gialli	2022-08-25 15:00:00	2022-08-30 11:00:00	5

6. Mostrare l'hotel con meno servizi e le sue recensioni:

```
SELECT h.Id_Hotel, h.Nome, COUNT(hs.Id_Servizio) AS NumeroServizi, r.Punteggio_Hotel, r.Commento
FROM Hotel h
LEFT JOIN HotelServizi hs ON h.Id_Hotel = hs.Id_Hotel
LEFT JOIN Recensione r ON h.Id_Hotel = r.Hotel
GROUP BY h.Id_Hotel, h.Nome, r.Punteggio_Hotel, r.Commento
HAVING COUNT(hs.Id_Servizio) = (
    SELECT MIN(ServiziCount)
    FROM (
        SELECT hs.Id_Hotel, COUNT(hs.Id_Servizio) AS ServiziCount
        FROM HotelServizi hs
        GROUP BY hs.Id_Hotel
    ) AS MinServizi
)
ORDER BY NumeroServizi ASC;
```

id_hotel	nome	numeroservizi	punteggio_hotel	commento
integer	character varying (30)	bigint	smallint	character varying (200)
1	Hotel Milano	2	1	Hotel poco accogliente e personale non molto gentile
1	Hotel Milano	2	9	Una piacevole esperienza culinaria

7. Mostrare le 3 attività affiliate con punteggio più alto e il loro prezzo tenendo in considerazione lo sconto;

```
SELECT A.Id_Attivita, A.Nome AS Nome_Attivita,
       A.Prezzo * (1 - A.Sconto) AS Prezzo_Scontato, A.Voto, H.Nome AS Nome_Hotel
FROM Attivita A
JOIN Hotel H ON A.Hotel = H.Id_Hotel
WHERE A.Affiliata = TRUE
ORDER BY A.Voto DESC
LIMIT 3;
```

id_attivita integer	nome_attivita character varying (30)	prezzo_scontato double precision	voto smallint	nome_hotel character varying (30)
10	Tour Enogastronomico	54.45	10	Grand Hotel
2	Tour delle Città	28.5	8	Hotel Milano
15	Giardino Botanico	20	8	Hotel de Ville

Indici

```
CREATE INDEX Camere on Camera(Id_Camera);
CREATE INDEX Hotel_index on Hotel(Id_Hotel);
```

Vista la gran quantità di camere ed hotel prevista dal sito di Booking2, è sembrato necessario velocizzare le operazioni che riguardano queste due entità per una maggiore efficienza nel mostrare i risultati.

6. Codice C++

Il codice cpp presenta il main e due funzioni EseguiQuery e StampaQuery, che servono rispettivamente all'esecuzione delle query e alla stampa a video dei loro risultati.

Il codice C++ per l'esecuzione delle query consiste in un unico file chiamato CodiceCpp.cpp, che va compilato attraverso il comando:

```
g++ CodiceCpp.cpp -Ldependencies/lib -lpq -o Codice
```

e per poter interagire con il codice si può utilizzare: .\Codice.exe

Questo programma mostrerà a schermo un menù che presenta le 7 query che abbiamo realizzato, le quali sono selezionabili inserendo valori numerici da 1 a 7.

Per i casi 3 e 5, le query parametriche, verrà stampata a video un'ulteriore frase che chiederà di inserire il valore da usare parametro, che per entrambi i casi dovrà essere un intero.