

TESINA BASI DI DATI

ANNO ACCADEMICO

2022/2023



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Mario Formato N46005548

Giovanni Falco N46004975

Salvatore Emanuele Di Cicco N46004863

Sommario

1. INTRODUZIONE	2
2. PROGETTAZIONE CONCETTUALE	3
2.1 SCHEMA E/R PORTANTE	3
2.2 SCHEMA E/R COMPLETO	3
2.3 DESCRIZIONI	4
3. PROGETTAZIONE LOGICA	5
3.1 TRADUZIONE DELLO SCHEMA E/R	5
3.2 TRASFORMAZIONE	6
3.3 TRADUZIONE	6
4 DIMENSIONAMENTO	10
4.1 TIPI DI DATO	10
4.2 TABELLE	10
5 PROGETTAZIONE FISICA	14
5.1 CREAZIONE TABELLE	14
5.2 VINCOLI	17
5.3 INDICI	18
5.4 CONCORRENZA	19
5.5 AFFIDABILITA'	19
5.6 SICUREZZA	19
5.7 POPOLAMENTO	21
5.8 QUERY	28
5.9 TRIGGERS	31
5.10 PROCEDURE	33

1. INTRODUZIONE

Dalle specifiche di progetto si vuole progettare una base di dati per un'agenzia di viaggio che consente la prenotazione di particolari pacchetti turistici e tenga conto del pagamento.

Si hanno le seguenti caratteristiche:

- 1) Pacchetto offerto, identificato da un codice e caratterizzato dal numero di persone per cui è pensato, con appositi spostamenti e soggiorni.
- 2) Spostamenti per il quale si vuole memorizzare il luogo di arrivo, luogo di partenza, le coordinate GPS per entrambi i luoghi e le date di arrivo e partenza (con appositi orari), con appositi mezzo di trasporto previsto.
- 3) Albergo dove alloggeranno gli utenti prenotati, identificati da un codice, nome, data check-in, data check-out e il costo giornaliero.
- 4) Camere in cui alloggeranno i singoli utenti per quel particolare soggiorno. Per le camere si vuole memorizzare, la tipologia, numero occupanti e servizi offerti.
- 5) Prenotazioni effettuate.
- 6) Utenti che prenotano e viaggiatori.
- 7) Metodo di pagamento (carta di credito e bonifico bancario).

Si suppone che, nell'arco del primo anno di esercizio, la base di dati dovrà gestire la seguente mole di informazioni:

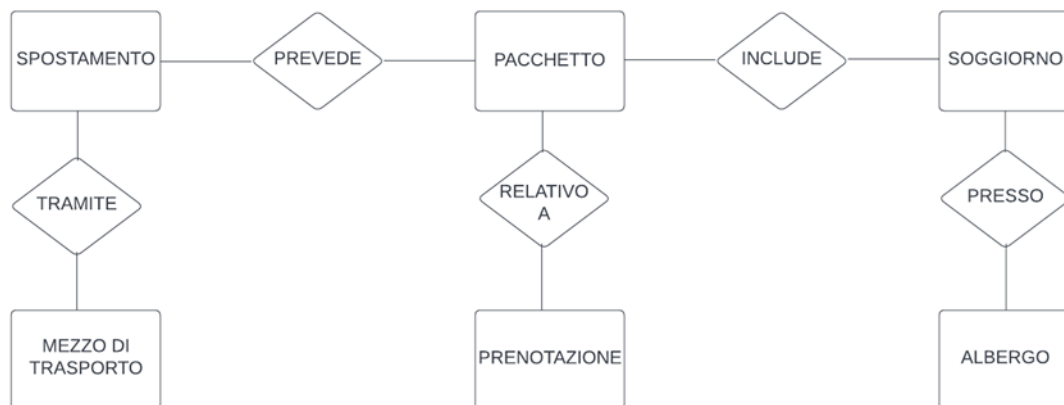
- 100 pacchetti
- 500 spostamenti
- 15 mezzi di trasporto
- 200 soggiorni
- 100 alberghi
- 200 camere
- 9 servizi offerti nelle camere degli alberghi
- 3000 prenotazioni

- 12000 utenti
- 6000 pagamenti con carta
- 6000 pagamenti con bonifico

2. PROGETTAZIONE CONCETTUALE

2.1 SCHEMA E/R PORTANTE

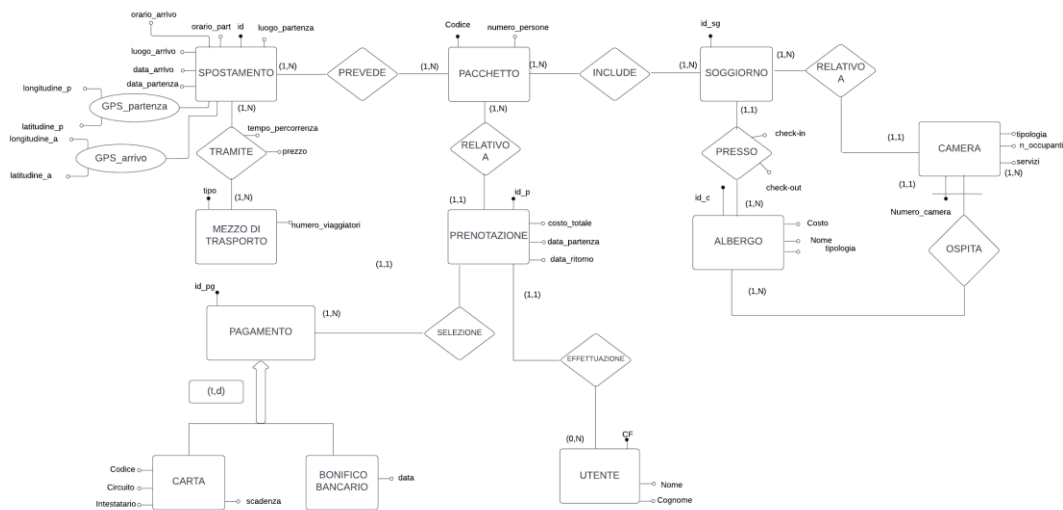
In base alle specifiche di progetto possiamo abbozzare uno schema portante:



2.2 SCHEMA E/R COMPLETO

Dopo una rilettura possiamo notare :

- Una relazione ternaria tra SOGGIORNO, CAMERA e ALBERGO con relativo identificatore esterno tra CAMERA e ALBERGO .
- L'entità CAMERA ha un attributo multivalore , tale attributo è 'servizi'.
- Una relazione ternaria tra PRENOTAZIONE , UTENTE e PAGAMENTO .
- L'entità PAGAMENTO presenta due specializzazioni , cioè due entità CARTA e BONIFICO .
- L'entità SPOSTAMENTO possiede due attributi composti 'GPS_partenza' e 'GPS_arrivo' .



2.3 DESCRIZIONI

Entità

Inizialmente è stato deciso che un pacchetto debba avere un codice identificativo e un numero di persone per attestare la quantità di coloro che prediligono usare quel pacchetto, inoltre il pacchetto entra in relazione con le entità SPOSTAMENTO, SOGGIORNO, PRENOTAZIONE poiché un pacchetto comprende le combinazioni tra le tre entità.

Più nello specifico uno SPOSTAMENTO può prevedere più mezzi di trasporto e ciò giustifica la relazione tra le due entità nello schema, inoltre il SOGGIORNO è una combinazione tra ALBERGHI e CAMERE .

Infine, l'entità PRENOTAZIONE è legato sia ad UTENTE che a PAGAMENTO in quanto un utente può effettuare diverse prenotazioni e diversi utenti possono effettuare determinati pagamenti, inoltre così' è facile determinare sia gli utenti che effettuano i pagamenti e gli utenti viaggiatori.

SPECIALIZZAZIONE

In questo schema è presente una specializzazione dell'entità PAGAMENTO, che può essere effettuato tramite una carta e tramite un bonifico.

CARDINALITA'

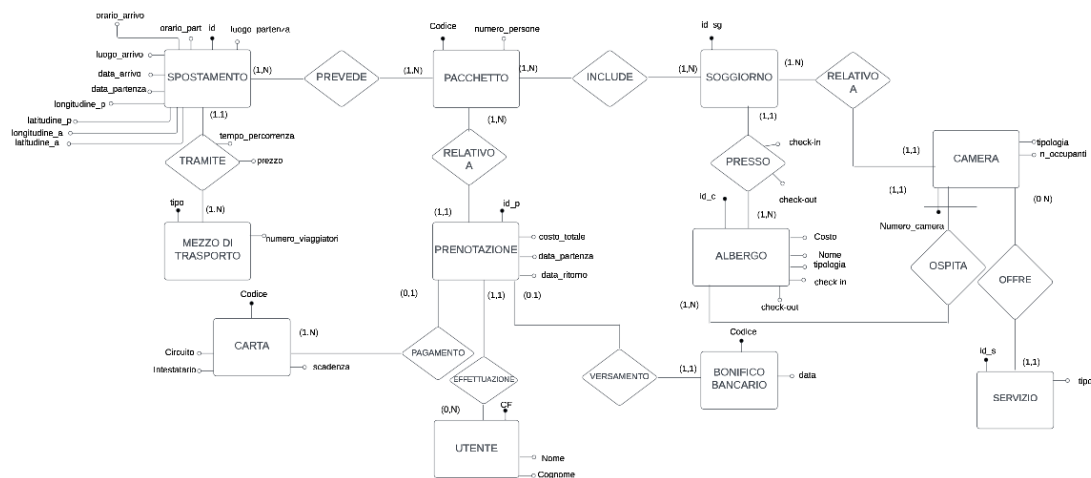
- L'associazione pacchetto e spostamento è del tipo molti a molti perché un pacchetto può prevedere più spostamenti e a più spostamenti possono corrispondere diversi spostamenti
- L'associazione pacchetto e soggiorno è del tipo molti a molti perché un pacchetto può prevedere più soggiorni e a più soggiorni corrispondono diversi pacchetti
- L'associazione pacchetto e prenotazione è del tipo uno a molti perché un pacchetto è pensato per più persone e le prenotazioni sono effettuate da una persona .
- L'associazione spostamento e mezzo di trasporto è del tipo molti a molti perché uno spostamento è composto da più mezzi di trasporto e ad un mezzo di trasporto è associato uno spostamento.
- L'associazione soggiorno e albergo è del tipo uno a molti perché un soggiorno comprendere un albergo e un albergo può gestire più soggiorni.
- L'associazione prenotazione e pagamento è del tipo uno a molti perché ad una prenotazione è associata un metodo di pagamento e ad un metodo di pagamento possono corrispondere diverse prenotazioni nel tempo.
- L'associazione prenotazione e utente è del tipo uno a molti in quanto ad una prenotazione è associata un utente, ma un utente può effettuare più prenotazioni nel corso del tempo.
- L'associazione soggiorno e camera è del tipo molti a uno in quanto diversi soggiorni sono associati ad una camera, una camera è relativa ad un soggiorno.
- L'associazione albergo e camera è del tipo molti a uno perché l'albergo è composto da diverse camere, mentre una camera appartiene uno e ad un solo albergo.

3. PROGETTAZIONE LOGICA

3.1 TRADUZIONE DELLO SCHEMA E/R

La progettazione logica si divide in due parti

- 1) Trasformazione dello schema concettuale in uno schema semplificato.
- 2) Traduzione dello schema semplificato nello schema relazionale.



3.2 TRASFORMAZIONE

- 1) Sostituzione degli attributi multivalore introducendo una nuova entità, come nel caso dell'entità CAMERA che aveva 'servizio' come attributo multivalore, che è stato tolto aggiungendo al suo posto un'entità, questa è associato a camera attraverso una relazione uno a molti.
- 2) Decomponendo gli attributi composti in attributi semplici come quelli presenti nell'entità SPOSTAMENTI, in cui all'entità SPOSTAMENTI si aggiungono gli attributi: longitudine_a, longitudine_p, latitudine_a, latitudine_p.
- 3) La generalizzazione della classe pagamento è risolta accorpendo la superclasse nella sottoclasse in quanto la generalizzazione è del tipo totale-disgiunta. Quindi avremo due entità CARTA e BONIFICO BANCARIO.

3.3 TRADUZIONE

Ogni entità si traduce in una relazione avente come attributi dell'entità e come chiave primaria l'identificatore dell'entità:

PACCHETTI (Codice, numero persone)

SPOSTAMENTI (Id, luogo di partenza, luogo di arrivo, orario di partenza, orario di arrivo, data partenza, data di arrivo, latitudine_p, latitudine_a, longitudine_a, longitudine_p)

MEZZI DI TRASPORTO (Tipo, numero viaggiatori)

PRENOTAZIONI (id_p, costo totale, data di partenza, data di ritorno)

UTENTI (CF, Nome, Cognome)

CARTE (Codice, circuito, intestatario, scadenza)

BONIFICI BANCARI (Codice, data)

SOGGIORNI (id_sg)

ALBERGHI (id_c , costo, nome, tipologia, check-in, check-out)

CAMERE (Numero camera, tipologia, numero occupanti, id_c: ALBERGHI)

SERVIZI (id_s , tipo)

Per finire bisogna tradurre l'associazione tra le entità:

Sono presenti associazioni del tipo molti a molti che possono essere tradotte tramite la regola 2 che consiste in:

si traduce l'associazione in una relazione dello schema logico avente lo stesso nome (ma al plurale) dell'associazione e per attributi gli attributi dell'associazione e gli identificatori delle entità coinvolte. L'insieme di tali identificatori costruisce la chiave primaria della relazione, ed ognuno di essi ha un vincolo di integrità referenziale con il corrispondente attributo dell'entità da cui discende.

Queste associazioni sono:

Tra PACCHETTO e SPOSTAMENTO:

COMBINAZIONISPOSTAMENTI (Codice, id : SPOSTAMENTO)

Tra SPOSTAMENTO e MEZZO DI TRASPORTO

ITINERARI (id : SPOSTAMENTO , tipo : MEZZO DI TRASPORTO , tempo percorrenza , prezzo)

Tra PACCHETTO e SOGGIORNO

COMBINAZIONISOGGIORNO (Codice : PACCHETTO, id_sg : SOGGIORNO)

Sono presenti anche associazioni uno a molti che vengono tradotte tramite la regola 4 che consiste in:

Una associazione uno a molti dello schema concettuale semplificato si traduce aggiungendo alla relazione che traduce l'entità dal lato uno gli attributi dell'associazione e l'identificatore dell'entità dal lato molti, eventualmente ridenominato .Esiste un vincolo di integrità referenziale tra questo attributo e il corrispondente attributo dell'entità dal lato molti.

Queste associazioni sono:

Tra CAMERA e SOGGIORNO

CAMERE (Numero camera, tipologia, numero occupanti, id_c : ALBERGHI, id_sg: SOGGIORNI)

Tra ALBERGO e CAMERA

CAMERE (Numero camera, tipologia, numero occupanti, id_c : ALBERGHI , id_sg: SOGGIORNO)

Tra CAMERA e SERVIZIO

SERVIZI (id_s , tipo , (Numero camera , id_c): CAMERE)

Tra PRENOTAZIONE e PACCHETTO

PRENOTAZIONI (id_p, costo totale, data partenza, data ritorno, codicePacchetto: PACCHETTI)

Tra PRENOTAZIONE e UTENTE

PRENOTAZIONI (id_p, costo totale, data partenza, data ritorno, codicePacchetto :PACCHETTI , CodiceFiscaleUtente : UTENTI)

Tra PRENOTAZIONE e CARTA DI CREDITO

PRENOTAZIONI (id_p, costo totale, data partenza, data ritorno, codicePacchetto :PACCHETTI , CodiceFiscaleUtente : UTENTI , CodiceCarta : CARTE DI CREDITO)

Tra SOGGIORNO e ALBERGO

SOGGIORNI (id_sg, check-in, check-out, id_c : ALBERGO)

Associazione uno a uno si traduce con la Regola 6:

Una associazione uno a uno dello schema concettuale si traduce aggiungendo alla relazione che traduce una delle due entità , gli attributi dell'associazione e l'identificatore dell'altra entità , eventualmente ridenominato. Esiste un vincolo di unicità sul nuovo attributo aggiunto ed un vincolo di integrità referenziale tra questo ultimo e il corrispondente attributo dell'altra entità.

Tra PRENOTAZIONE e BONIFICO BANCARIO

PRENOTAZIONI (id_p, costo totale, data partenza, data ritorno, codicePacchetto :PACCHETTI , CodiceFiscaleUtente : UTENTI , CodiceCarta : CARTE DI CREDITO , CodiceBancario: BONIFICI BANCARIO)

Di seguito è riportato lo schema relazionale completo per la base di dati in oggetto:

PACCHETTI (Codice, numero persone)

SPOSTAMENTI (Id, luogo di partenza, luogo di arrivo, orario di partenza, orario di arrivo, data partenza, data di arrivo, latitudine_p, latitudine_a, longitudine_a, longitudine_p)

MEZZI DI TRASPORTO (Tipo, numero viaggiatori)

PRENOTAZIONI (id_p, costo totale, data partenza, data ritorno, codicePacchetto :PACCHETTI , CodiceFiscaleUtente : UTENTI , CodiceCarta : CARTE DI CREDITO , CodiceBancario: BONIFICI BANCARIO)

UTENTI (CF, Nome, Cognome)

CARTE (Codice, circuito, intestatario, scadenza)

BONIFICI BANCARI (Codice, data ,Codice_prenotazione)

SOGGIORNI (id_sg, check-in, check-out, id_c : ALBERGO)

ALBERGHI (id_c , costo, nome, tipologia)

CAMERE (Numero camera, tipologia, numero occupanti, id_c : ALBERGHI , id_sg: SOGGIORNO)

SERVIZI (id_s , tipo , (Numero camera , id_c): CAMERE)

COMBINAZIONISPOSTAMENTO (Codice:PACCHETTI, id : SPOSTAMENTI)

ITINERARI (id : SPOSTAMENTO , tipo : MEZZO DI TRASPORTO , tempo percorrenza , prezzo)

COMBINAZIONISOGGIORNO (Codice : PACCHETTI, id_sg : SOGGIORNI)

4 DIMENSIONAMENTO

4.1 TIPI DI DATO

Terminata dunque la fase di progettazione logica e dopo aver esposto le informazioni che saranno all'interno della nostra base di dati, ci occupiamo di specificare i tipi di dato che abbiamo utilizzato nella creazione delle tabelle:

- **VARCHAR2(n)**: è stato utilizzato il tipo varchar2 per gli attributi di tipo testuale a lunghezza variabile che sono più propensi a variazioni (come NOME, COGNOME . . .). Il tipo varchar2 prevede in ORACLE una collocazione di n byte per n caratteri testuali entro la dimensione dei 4000 byte per valore.
- **CHAR**: è stato utilizzato per i codici fiscali con un valore di 16 bit (2 byte).
- **NUMBER**: dato l'utilizzo di codici identificativi numerici per la loro gestione, è stato utilizzato il tipo Number per questi campi. Esso prevede in ORACLE una collocazione di $n \cdot 2 + 2$ byte per n cifre.
- **DATE**: è stato utilizzato il tipo Date negli attributi di tipo data. Esso prevede in ORACLE una collocazione di 7byte.
- **TIMESTAMP**: è stato utilizzato per i valori che contengono sia data che ora. Il suo valore è di 8 byte.
- **INTEGER**: è stato invece utilizzato per la gestione dei prezzi. Con un valore di 4 byte

4.2 TABELLE

Date le informazioni sul volume dei dati e scelti i tipi degli attributi utilizzati per l'implementazione della base di dati è possibile dare una stima della quantità di memoria che le tabelle andranno ad occupare.

DIMENSIONI TIPI:

NUMBER(x): $(\lceil x/2 \rceil + 2)$ byte

DATE: 7 byte

CHAR(x): x byte

VARCHAR(x): x byte

INTEGER: 4 byte

TIMESTAMP: 4 byte

PACCHETTI		
ATTRIBUTO	TIPO	BYTE
Codice	NUMBER(6)	5
Numero_persone	NUMBER(2)	3

STIMA DIMENSIONE PACCHETTI:

Spazio pacchetti = $N(\text{tuple}) * (S_{\text{codice}} + S_{\text{numero_persone}}) = 100 * (5+3) = 800 \text{ B}$

SPOSTAMENTI		
ATTRIBUTO	TIPO	BYTE
Id	NUMBER(6)	5
Luogo_partenza	VARCHAR2(50)	50
Luogo_arrivo	VARCHAR2(50)	50
Data_ora_partenza	TIMESTAMP	4
Data_ora_arrivo	TIMESTAMP	4
Longitudine_p	VARCHAR2(30)	30
Latitudine_p	VARCHAR2(30)	30
Longitudine_a	VARCHAR2(30)	30
Latitudine_a	VARCHAR2(30)	30

STIMA DIMENSIONI SPOSTAMENTI

$S = 500 * (5 + 50 + 50 + 4 + 4 + 30 + 30 + 30 + 30) = 116,5 \text{ kB}$

MEZZI DI TRASPORTO		
ATTRIBUTO	TIPO	BYTE
Tipo	VARCHAR2(20)	20
Numero_viaggiatori	INTEGER	4

STIMA DIMENSIONI MEZZI DI TRASPORTO

$$S = 15 * (20 + 4) = 360 \text{ B}$$

UTENTI		
ATTRIBUTO	TIPO	BYTE
CF	CHAR(16)	16
Nome	VARCHAR2(30)	30
Cognome	VARCHAR2(30)	30

STIMA DIMENSIONI UTENTI

$$S = 12000 * (16 + 30 + 30) = 912 \text{ kB}$$

CARTE		
ATTRIBUTO	TIPO	BYTE
Codice	NUMBER(16)	10
Circuito	VARCHAR2(30)	30
Intestatario	VARCHAR2(50)	50
Scadenza	DATE	7

STIMA DIMENSIONI CARTE

$$S = 6000 * (10 + 30 + 50 + 7) = 582 \text{ kB}$$

PRENOTAZIONI		
ATTRIBUTO	TIPO	BYTE
Id_p	NUMBER(6)	5
Costo_totale	INTEGER	4
Data_partenza	DATE	7
Data_ritorno	DATE	7
Codice_pacchetti	NUMBER(6)	5
Codice_fiscale_utenti	CHAR(16)	16
Codice_carta	NUMBER(16)	10
Codice_bancario	VARCHAR2(30)	30

STIMA DIMENSIONI PRENOTAZIONI

$$S = 3000 * (5 + 4 + 7 + 7 + 5 + 16 + 10 + 30) = 252 \text{ kB}$$

BONIFICI BANCARI		
ATTRIBUTO	TIPO	BYTE
Codice	VARCHAR2(30)	30
Data	DATE	7
Codice_prenotazione	NUMBER(6)	5

STIMA DIMENSIONI BONIFICI BANCARI

$$S = 6000 * (30 + 7 + 5) = 252 \text{ kB}$$

ALBERGHI

ATTRIBUTO	TIPO	BYTE
Id_c	NUMBER(6)	5
Costo	INTEGER	4
Nome	VARCHAR2(50)	50
Tipologia	VARCHAR2(50)	50

STIMA DIMENSIONI ALBERGHI

$$S = 100 * (5 + 4 + 50 + 50) = 10,9 \text{ kB}$$

SOGGIORNI		
ATTRIBUTO	TIPO	BYTE
Id_sg	NUMBER(6)	5
Check_in	DATE	7
Check_out	DATE	7
Codice_albergo	NUMBER(6)	5

STIMA DIMENSIONI SOGGIORNI

$$S = 200 * (5 + 7 + 7 + 5) = 4,8 \text{ kB}$$

CAMERE		
ATTRIBUTO	TIPO	BYTE
Numero_camera	NUMBER(2)	3
Tipologia	VARCHAR2(50)	50
Numero_occupanti	INTEGER	4
Codice_alberghi	NUMBER(6)	5
Codice_soggiorno	NUMBER(6)	5

STIME DIMENSIONI CAMERE

$$S = 200 * (3 + 50 + 4 + 5 + 5) = 13,4 \text{ kB}$$

SERVIZI		
ATTRIBUTO	TIPO	BYTE
Id_s	NUMBER(2)	3
Tipo	VARCHAR2(100)	100
Codice_camera	NUMBER(2)	3
Codice_albergo	NUMBER(6)	5

STIMA DIMENSIONI SERVIZI

$$S = 9 * (3 + 100 + 3 + 5) = 1 \text{ kB}$$

COMBINAZIONI SPOSTAMENTO

ATTRIBUTO	TIPO	BYTE
Codice_comb	NUMBER(6)	5
Codice_spostamento	NUMBER(6)	5

STIMA DIMENSIONI COMBINAZIONI SPOSTAMENTO

$$S = 500 * (5 + 5) = 5 \text{ kB}$$

ITINERARI		
ATTRIBUTO	TIPO	BYTE
Id	NUMBER(6)	5
Tipo	VARCHAR2(20)	20
Tempo_percorrenza	INTEGER	4
Prezzo	INTEGER	4

STIMA DIMENSIONI ITINERARI

$$S = 500 * (5 + 20 + 4 + 4) = 16,5 \text{ kB}$$

COMBINAZIONI SOGGIORNO		
ATTRIBUTO	TIPO	BYTE
Codice	NUMBER(6)	5
Id_sg	NUMBER(6)	5

STIMA DIMENSIONI COMBINAZIONI SOGGIORNO

$$S = 500 * (5 + 5) = 5 \text{ kB}$$

5 PROGETTAZIONE FISICA

5.1 CREAZIONE TABELLE

```
CREATE TABLE PACCHETTI (
  Codice NUMBER(6),
  Numero_persone NUMBER(2),

  CONSTRAINT PK_PACCHETTI PRIMARY KEY (Codice),
  CONSTRAINT CK_PACCHETTI CHECK(Numero_persone>0)
);

CREATE TABLE SPOSTAMENTI (
```

```

id NUMBER(6),
luogo_partenza varchar2(50) NOT NULL,
luogo_arrivo varchar2(50) NOT NULL ,
data_ora_partenza TIMESTAMP,
data_ora_arrivo TIMESTAMP,
latitudine_p varchar2(30) NOT NULL,
latitudine_a varchar2(30) NOT NULL,
longitudine_p varchar2(30) NOT NULL,
longitudine_a varchar2(30) NOT NULL,
CONSTRAINT PK_SPOSTAMENTI PRIMARY KEY (id)
);

CREATE TABLE MEZZI_DI TRASPORTO (
Tipo VARCHAR2(20),
Numero_viaggiatori INTEGER,
CONSTRAINT PK_MEZZI_DI TRASPORTO PRIMARY KEY (Tipo),
CONSTRAINT CK_MEZZI_DI TRASPORTO CHECK(Numero_viaggiatori>0)
);

CREATE TABLE UTENTI (
CF CHAR(16),
Nome VARCHAR2(30) NOT NULL,
Cognome VARCHAR2(30) NOT NULL,
CONSTRAINT PK_UTENTI PRIMARY KEY (CF)
);

CREATE TABLE CARTE (
Codice NUMBER(16),
Circuito VARCHAR2(30) NOT NULL ,
Intestatario VARCHAR2(50) NOT NULL ,
Scadenza DATE NOT NULL,
CONSTRAINT PK_CARTE PRIMARY KEY (Codice)
);

CREATE TABLE PRENOTAZIONI (
id_p NUMBER(6),
costo_totale INTEGER,
data_partenza DATE NOT NULL,
data_ritorno DATE NOT NULL ,
codicePacchetti NUMBER(6),
codiceFiscaleUtenti CHAR(16),
codiceCarta NUMBER(16),
codiceBancario VARCHAR2(30),
CONSTRAINT PK_PRENOTAZIONI PRIMARY KEY (id_p),
CONSTRAINT CK_PRENOTAZIONI CHECK(costo_totale>0)
);

```



```
CREATE TABLE BONIFICI_BANCARI (
Codice VARCHAR2(30),
Data DATE NOT NULL ,
Codice_prenotazione NUMBER(6) UNIQUE NOT NULL,
CONSTRAINT PK_BONIFICI_BANCARI PRIMARY KEY (Codice)
);
```

```
CREATE TABLE ALBERGHI (
id_c NUMBER(6),
costo INTEGER,
nome VARCHAR2(50) NOT NULL,
tipologia VARCHAR2(50) NOT NULL,
CONSTRAINT PK_ALBERGHI PRIMARY KEY (id_c),
CONSTRAINT CK_ALBERGHI CHECK(costo>0)
);
```

```
CREATE TABLE SOGGIORNI (
id_sg NUMBER(6),
check_in DATE NOT NULL,
check_out DATE NOT NULL,
codice_albergo NUMBER(6),
CONSTRAINT PK_SOGGIORNI PRIMARY KEY (id_sg)
);
```

```
CREATE TABLE CAMERE (
Numero_camera NUMBER(2),
tipologia VARCHAR2(50) NOT NULL,
numero_occupanti INTEGER,
codice_alberghi NUMBER(6),
codice_soggiorno NUMBER(6),

CONSTRAINT PK_CAMERE PRIMARY KEY (Numero_camera) ,
CONSTRAINT CK_CAMERE CHECK(numero_occupanti>0)
);
```

```
CREATE TABLE SERVIZI (
id_s NUMBER(2),
tipo VARCHAR2(100) NOT NULL,
codice_camera NUMBER(2),
codice_albergo NUMBER(6),

CONSTRAINT PK_SERVIZI PRIMARY KEY (id_s)
);
```

```
CREATE TABLE COMBINAZIONI_SPOSTAMENTO (
Codice_comb NUMBER(6),
```

```

Codice_spostamento NUMBER(6),

CONSTRAINT PK_COMBINAZIONE_SPOSTAMENTO PRIMARY KEY (Codice_comb,
Codice_spostamento )
);

CREATE TABLE ITINERARI (
id_i NUMBER(6),
Tipo_i VARCHAR2(20),
Tempo_percorrenza INTEGER,
Prezzo INTEGER,

CONSTRAINT PK_ITINERARI PRIMARY KEY (id_i, Tipo_i),
CONSTRAINT CK_SPOSTAMENTO CHECK(Prezzo>0)
);

CREATE TABLE COMBINAZIONI_SOGGIORNO (
Codice_S NUMBER(6),
Id_sg NUMBER(6),

CONSTRAINT PK_COMBINAZIONI_SOGGIORNO PRIMARY KEY (Codice_S, Id_sg)
);

```

5.2 VINCOLI

I vincoli di integrità referenziale sono aggiunti tramite il comando ALTER TABLE che consente di modificare una tabella che è stata già creata . Tramite al comando ALTER TABLE non siamo vincolati all'ordine delle creazioni delle tabelle per aggiungere i vincoli necessari .

I vincoli sono i seguenti :

NO ACTION : non compie nessuna operazione sulla base di dati (di default)

SET NULL : setta automaticamente a NULL tutti gli attributi che sono collegati , come chiave esterna a quello eliminato

SET DEFAULT : imposta gli attributi eliminati a 0 .

CASCADE : elimina automaticamente tutti gli oggetti associati all'attributo eliminato .

```

ALTER TABLE SOGGIORNI ADD CONSTRAINT FK_ALBERGHI FOREIGN KEY (codice_albergo)
REFERENCES ALBERGHI (id_c) ON DELETE CASCADE ;

ALTER TABLE CAMERE ADD CONSTRAINT FK_ALBERGHI_C FOREIGN KEY (codice_alberghi)
REFERENCES ALBERGHI (id_c) ON DELETE CASCADE ;

```

```

ALTER TABLE CAMERE ADD CONSTRAINT FK_SOGGIORNI_C FOREIGN KEY (
codice_soggiorno) REFERENCES SOGGIORNI (id_sg) ;

ALTER TABLE SERVIZI ADD CONSTRAINT FK_ALBERGHI_SERVIZI FOREIGN KEY
(codice_albergo) REFERENCES ALBERGHI (id_c) ON DELETE CASCADE ;
ALTER TABLE SERVIZI ADD CONSTRAINT FK_SOGGIORNI_SERVIZI FOREIGN KEY
(codice_camera) REFERENCES CAMERE (Numero_camera) ;

ALTER TABLE COMBINAZIONI_SPOSTAMENTO ADD CONSTRAINT
FK_SPOSTAMENTO_COMB FOREIGN KEY (Codice_spostamento) REFERENCES
SPOSTAMENTI (id) ON DELETE CASCADE ;
ALTER TABLE COMBINAZIONI_SPOSTAMENTO ADD CONSTRAINT FK_PACCHETTI_COMB
FOREIGN KEY (Codice_comb) REFERENCES PACCHETTI (Codice) ON DELETE CASCADE ;

ALTER TABLE ITINERARI ADD CONSTRAINT FK_SPOSTAMENTO_IT FOREIGN KEY (id_i)
REFERENCES SPOSTAMENTI (id) ON DELETE CASCADE ;
ALTER TABLE ITINERARI ADD CONSTRAINT FK_MEZZO_DI TRASPORTO FOREIGN KEY
(Tipo_i) REFERENCES MEZZI_DI TRASPORTO (Tipo) ON DELETE CASCADE ;

ALTER TABLE COMBINAZIONI_SOGGIORNO ADD CONSTRAINT FK_PACCHETTO FOREIGN
KEY (Codice_S) REFERENCES PACCHETTI (Codice) ON DELETE CASCADE ;
ALTER TABLE COMBINAZIONI_SOGGIORNO ADD CONSTRAINT FK_SOGGIORNO FOREIGN
KEY (Id_sg) REFERENCES SOGGIORNI (id_sg) ON DELETE CASCADE ;

ALTER TABLE PRENOTAZIONI ADD CONSTRAINT FK_PACCHETTI FOREIGN KEY
(codicePacchetti) REFERENCES PACCHETTI (Codice) ON DELETE CASCADE;
ALTER TABLE PRENOTAZIONI ADD CONSTRAINT FK_UTENTI FOREIGN KEY
(CodiceFiscaleUtenti) REFERENCES UTENTI (CF) ON DELETE CASCADE ;
ALTER TABLE PRENOTAZIONI ADD CONSTRAINT FK_CARTE FOREIGN KEY (CodiceCarta)
REFERENCES CARTE(Codice) ON DELETE CASCADE ;
ALTER TABLE PRENOTAZIONI ADD CONSTRAINT FK_BONIFICI FOREIGN KEY
(CodiceBancario) REFERENCES BONIFICI_BANCARI (Codice) ON DELETE CASCADE;

```

5.3 INDICI

Gli indici servono per ottimizzare gli accessi nel caso in cui si ha un database di grandi dimensioni . quindi è necessario stabilire degli indici per le informazioni più richieste .

Abbiamo scelto :

- 1) Spostamenti : sugli attributi luogo di partenza e luogo di arrivo , in quanto potrebbe essere utile cercare gli elementi in base a questi due attributi .
- 2) Alberghi: sull'attributo costo in quanto un utente potrebbe avere più preferenze a cercare un albergo con costo minore .
- 3) Camere: sull'attributo tipologia, in quanto in base al numero dei viaggiatori , l'utente è più predisposto a prenotare una camera con più posti .

```
CREATE INDEX INDICE_SPOSTAMENTI ON SPOSTAMENTI (luogo_partenza , luogo_arrivo ) ;  
CREATE INDEX INDICE_CAMERE ON CAMERE (tipologia) ;  
CREATE INDEX INDICE_ALBERGO ON ALBERGHI (costo) ;
```

5.4 CONCORRENZA

In SQL è possibile differenziare transazioni che effettuano operazioni in lettura/scrittura da quelle che effettuano la sola lettura. La prima per evitare anomalie preferiamo scegliere il meccanismo 2PL stretto (Two phase locking) mentre per le seconde si preferisce scegliere il REPEATABLE READ che consiste nel fatto che la transazione accetta di leggere i dati modificati solo dopo aver effettuato il commit , inoltre se un dato è letto due volte rilegge sempre lo stesso valore .

5.5 AFFIDABILITA'

Si è scelto come metodo di storage il RAID 1 , consiste nella copia di un disco per ogni disco , quindi se ho un disco ho una copia per questo disco , se ho due dischi ne ho due di copia e così' via . Questo metodo garantisce molta affidabilità grazie alla tecnica del mirroring , che in casi di guasti consente di sostituire il disco danneggiato .Inoltre favorisce operazioni in lettura sfavorendo quelle in scrittura .

Per quanto riguarda l'affidabilità sui guasti si differenziano i guasti in due tipi , soft inerenti agli errori di programma , crash di sistema e sono ripristinabili tramite la ripresa a caldo che consente il ripristino dei dati a partire dall'ultimo checkpoint .

Per quelli hard si preferisce una ripresa a freddo che consiste nel ripristinare i dati a partire dall'ultimo backup , eseguendo tutte le operazioni registrate sul file log relative alla parte danneggiata e per finire effettuare una ripresa a caldo .

5.6 SICUREZZA

Abbiamo scelto tre tipi di utenti :

- 1) il DBA (data base administrator) che possiede tutti i privilegi .
- 2) l'operatore che ha il privilegio di vedere tutto .Per quanto riguarda i pacchetti , mezzi e spostamenti ha anche l'autorizzazione di modificarli ,mentre per quanto riguarda il lato utenti quindi pagamenti e prenotazioni possono soltanto essere visionate .
- 3) il cliente che può effettuare le prenotazioni e i pagamenti ,in più visionare i pacchetti e i soggiorni e spostamenti associati .

```
CREATE USER dba_azienda DEFAULT TABLESPACE ts_azienda IDENTIFIED BY prop_00  
GRANT dba, UNLIMITED TABLESPACE to dba_azienda
```

```
CREATE ROLE operatore
```

```

GRANT CONNECT TO operatore ;
GRANT all privileges on dba_azienza.Pacchetti *TO operatore ;
GRANT all privileges on dba_azienza.Spostamenti *TO operatore ;
GRANT all privileges on dba_azienza.Soggiorni *TO operatore ;
GRANT select on dba_azienza.Pagamenti *TO operatore ;
GRANT select, update, insert , delete on dba_azienza.Prenotazioni *TO operatore ;
GRANT select, update , insert , delete on dba_azienza.Utenti *TO operatore ;
GRANT select on dba_azienza.combiazioni_soggiorno *TO operatore ;
GRANT select on dba_azienza.combinazioni_spostamenti *TO operatore ;
GRANT select on dba_azienza.itinerari *TO operatore ;
GRANT select on dba_azienza.Alberghi *TO operatore ;
GRANT select on dba_azienza.Camere*TO operatore ;
GRANT select on dba_azienza.Servizi* TO operatore ,

```

```

CREATE ROLE utente ;
GRANT CONNECT TO utente ;
GRANT select on dba_azienza.Pacchetti *TO utente ;
GRANT select on dba_azienza.Spostamenti *TO utente ;
GRANT select on dba_azienza.Soggiorni *TO utente ;
GRANT select on dba_azienza.Alberghi *TO utente ;
GRANT select on dba_azienza.Camere *TO utente ;
GRANT select on dba_azienza.Servizi *TO utente ;
GRANT select on dba_azienza.Combinazioni_spostamento *TO utente ;
GRANT select on dba_azienza.itinerari *TO utente ;
GRANT select on dba_azienza.Mezzi_di_trasporto *TO utente ;
GRANT update , delete , insert on dba_azienza.Pagamenti *TO utente ;
GRANT select , insert , update , delete on dba_azienza.Prenotazioni *TO utente ;

```

5.7 POPOLAMENTO

PACCHETTI

CODICE	NUMERO_PERSONE
1111	3
2222	5
3333	2
4444	4
5555	10

INSERT INTO PACCHETTI VALUES (1111,3);

INSERT INTO PACCHETTI VALUES (2222,5);

INSERT INTO PACCHETTI VALUES (3333,2);

INSERT INTO PACCHETTI VALUES (4444,4);

INSERT INTO PACCHETTI VALUES (5555,10);

SPOSTAMENTI

INSERT INTO SPOSTAMENTI VALUES (1000, 'Napoli', 'Roma', '12-Feb-2018 10:30:00', '12-Feb-2018 12:00:00', '40°50'49.20" NORD', '41°53'57.12" NORD', '14°15'54.00" EST', '12°32'42.00" EST') ;

INSERT INTO SPOSTAMENTI VALUES (1001, 'Napoli', 'Milano', '10-Mar-2019 05-00-00', '10-Mar-2019 12-00-00', '40°50'49.20" NORD', '45°46'42.19" NORD', '14°15'54.00" EST', '09°09'34.20" EST');

INSERT INTO SPOSTAMENTI VALUES (1002, 'Napoli', 'Venezia', '22-Apr-2022 04-30-00', '22-Apr-2022 12-25-00', '40°50'49.20" NORD', '45°26'23" NORD', '14°15'54.00" EST', '12°19'55" EST');

INSERT INTO SPOSTAMENTI VALUES (5000, 'Roma', 'Napoli', '18-Feb-2018 9-30-00', '18-Feb-2018 11-00-00', '41°53'57.12" NORD', '40°50'49.20" NORD', '12°32'42.00" EST', '14°15'54.00" EST');

INSERT INTO SPOSTAMENTI VALUES (5001, 'Milano', 'Napoli', '15-Mar-2019 05-00-00', '15-Mar-2019 12-00-00', '45°46'42.19" NORD', '40°50'49.20" NORD', '09°09'34.20" EST', '14°15'54.00" EST');

INSERT INTO SPOSTAMENTI VALUES (5002, 'Venezia', 'Napoli', '28-Apr-2022 04-15-00', '28-Apr-2022 12-00-00', '45°26'23" NORD', '40°50'49.20" NORD', '12°19'55" EST', '14°15'54.00" EST');

ID	LUOGO_PARTENZA	LUOGO_ARRIVO	DATA_ORA_PARTENZA	DATA_ORA_ARRIVO	LATITUDINE_P	LATITUDINE_A	LONGITUDINE_P	LONGITUDINE_A
1000	Napoli	Roma	12-FEB-18 10.30.00.000000 AM	12-FEB-18 12.00.00.000000 PM	40°50'49.20" NORD	41°53'57.12" NORD	14°15'54.00" EST	12°32'42.00" EST
1001	Napoli	Milano	10-MAR-19 05.00.00.000000 AM	10-MAR-19 12.00.00.000000 PM	40°50'49.20" NORD	45°46'42.19" NORD	14°15'54.00" EST	09°09'34.20" EST
1002	Napoli	Venezia	22-APR-22 04.30.00.000000 AM	22-APR-22 12.25.00.000000 PM	40°50'49.20" NORD	45°26'23" NORD	14°15'54.00" EST	12°19'55" EST
5000	Roma	Napoli	18-FEB-18 09.30.00.000000 AM	18-FEB-18 11.00.00.000000 AM	41°53'57.12" NORD	40°50'49.20" NORD	12°32'42.00" EST	14°15'54.00" EST
5001	Milano	Napoli	15-MAR-19 05.00.00.000000 AM	15-MAR-19 12.00.00.000000 PM	45°46'42.19" NORD	40°50'49.20" NORD	09°09'34.20" EST	14°15'54.00" EST
5002	Venezia	Napoli	28-APR-22 04.15.00.000000 AM	28-APR-22 12.00.00.000000 PM	45°26'23" NORD	40°50'49.20" NORD	12°19'55" EST	14°15'54.00" EST

MEZZI_DI_TRASPORTO

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Automobile',5);

TIPO	NUMERO_VIAGGIATORI
Automobile	5
Aereo	54
Treno	40
Bus	42
Metropolitana	35
Funicolare	16

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Aereo',54);

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Treno',40);

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Bus',42);

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Metropolitana',35);

INSERT INTO MEZZI_DI_TRASPORTO VALUES ('Funicolare',16);

PRENOTAZIONI

INSERT INTO PRENOTAZIONI VALUES (123, 100 , '12-Feb-2018', '18-Feb-2018', 1111,'FLCNMMM00LHGFF',0293256348952015,NULL) ;

INSERT INTO PRENOTAZIONI VALUES (345, 200 , '10-Mar-2019', '15-Mar-2019',3333,'MFR222RFTMTMTM',2598745632154255, NULL) ;

INSERT INTO PRENOTAZIONI VALUES (567, 300 , '22-Apr-2022', '28-Apr-2022',4444,'FLCMHMM22LHGFF',NULL,'3347563288') ;

ID_P	COSTO_TOTALE	DATA_PARTENZA	DATA_RITORNO	CODICEPACCHETTI	CODICEFISCALEUTENTI	CODICECARTA	CODICEBANCARIO
123	100	12-FEB-18	18-FEB-18	1111	FLCNMMM00LHGFF	293256348952015	-
345	200	10-MAR-19	15-MAR-19	3333	MFR222RFTMTMTM	2598745632154255	-
567	300	22-APR-22	28-APR-22	4444	FLCMHMM22LHGFF	-	3347563288

UTENTI

```

INSERT INTO UTENTI VALUES ('FLCNMMM00LHGFF','Giovanni','Falco');
INSERT INTO UTENTI VALUES ('FLCMHMM22LHGFF','Michele','Falco');
INSERT INTO UTENTI VALUES ('FLCARCM33LHGFF','Arcangelo','Falco');
INSERT INTO UTENTI VALUES ('DICC101VTCCCC00','Salvatore','Di Cicco');
INSERT INTO UTENTI VALUES('MFR222RFTMTMTM','Mario','Formato');
INSERT INTO UTENTI VALUES ('DICC101VTCCCLDL','Emanuela','Di Cicco');

```

CF	NOME	COGNOME
FLCNMMM00LHGFF	Giovanni	Falco
FLCMHMM22LHGFF	Michele	Falco
FLCARCM33LHGFF	Arcangelo	Falco
DICC101VTCCCC00	Salvatore	Di Cicco
MFR222RFTMTMTM	Mario	Formato
DICC101VTCCCLDL	Emanuela	Di Cicco
CMGL123LRLRLR	Laura	Cimiglio
MCC456LCLCLC	Luca	Moccia

CARTE

```

INSERT INTO CARTE VALUES (0293256348952015,'Mastercard','Giovanni Falco','16-Jun-2026');

```


INSERT INTO CARTE VALUES (2598745632154255,'Visa','Mario Formato','11-Mar-2029');

INSERT INTO CARTE VALUES (2548726592716489,'American Express','Salvatore Di Cicco','06-Aug-2026');

CODICE	CIRCUITO	INTESTATARIO	SCADENZA
293256348952015	Mastercard	Giovanni Falco	16-JUN-26
2598745632154255	Visa	Mario Formato	11-MAR-29
2548726592716489	American Express	Salvatore Di Cicco	06-AUG-26

BONIFICI_BANCARI

CODICE	DATA	CODICE_PRENOTAZIONE
3347563288	21-APR-22	567

INSERT INTO BONIFICI_BANCARI
VALUES ('3347563288', '21-Apr-2022',
567);

SOGGIORNI

ID_SG	CHECK_IN	CHECK_OUT	CODICE_ALBERGO
10101	10-MAR-19	15-MAR-19	101010
20202	22-APR-22	28-APR-22	202020
30303	12-FEB-18	18-FEB-18	303030

INSERT INTO SOGGIORNI VALUES
(010101,'10-Mar-2019','15-Mar-
2019',101010) ;

INSERT INTO SOGGIORNI VALUES
(020202,'22-Apr-2022', '28-Apr-
2022',202020) ;

INSERT INTO SOGGIORNI VALUES
(030303 , '12-Feb-2018', '18-Feb-
2018',303030) ;

ALBERGHI

ID_C	COSTO	NOME	TIPOLOGIA
202020	33	Concaverde	Residenza d'epoca
303030	50	Barbara	Motel
404040	28	Padula	b&b
101010	40	Sant'Ilario	b&b

INSERT INTO ALBERGHI VALUES(101010,40,
'Sant'Ilario', 'b&b');

INSERT INTO ALBERGHI VALUES(202020,33,
'Concaverde', 'Residenza d'epoca');

INSERT INTO ALBERGHI VALUES(303030,50 ,
'Barbara ', 'Motel ');

INSERT INTO ALBERGHI VALUES(404040,28, 'Padula', 'b&b');

CAMERE

INSERT INTO CAMERE VALUES (10, 'Doppia', 2, 101010, 010101);

INSERT INTO CAMERE VALUES (11, 'Singola', 1, 202020, 020202);

INSERT INTO CAMERE VALUES (12, 'Tripla', 3, 303030, 030303);

NUMERO_CAMERA	TIPOLOGIA	NUMERO_OCCUPANTI	CODICE_ALBERGHI	CODICE_SOGGIORNO
10	Doppia	2	101010	10101
11	Singola	1	202020	20202
12	Tripla	3	303030	30303

SERVIZI

ID_S	TIPO	CODICE_CAMERA	CODICE_ALBERGO
5	Bagno privato	10	101010
6	frigo bar	11	202020
7	colazione in camera	12	303030
1	Wi-fi	10	101010
2	aria condizionata	11	202020
3	TV	12	303030
4	Netflix	10	101010

INSERT INTO SERVIZI VALUES (1, 'Wi-fi', 10 , 101010) ;

INSERT INTO SERVIZI VALUES (2, 'aria condizionata', 11 , 202020) ;

INSERT INTO SERVIZI VALUES(3, 'TV', 12 , 303030) ;

INSERT INTO SERVIZI VALUES (4, , 'Netflix', 10 , 101010) ;

INSERT INTO SERVIZI VALUES (5, , 'Bagno privato', 10 , 101010) ;

INSERT INTO SERVIZI VALUES (6, 'frigo bar', 11 , 202020) ;

INSERT INTO SERVIZI VALUES (7, 'colazione in camera', 12 , 303030) ;

COMBINAZIONI_SPOSTAMENTO

CODICE_COMB	CODICE_SPOSTAMENTO
1111	1000
1111	5000
3333	1001
3333	5001
4444	1002
4444	5002

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (1111, 1000);

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (3333, 1001);

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (4444, 1002);

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (4444, 5002);

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (3333, 5001);

INSERT INTO COMBINAZIONI_SPOSTAMENTO VALUES (1111, 5000);

ITENERARI

ID_I	TIPO_I	TEMPO_PERCORRENZA	PREZZO
1000	Bus	2	10
1001	Aereo	1	80
1002	Aereo	3	100
5000	Treno	3	10
5001	Treno	8	40
5002	Aereo	3	100

```

INSERT INTO ITINERARI VALUES
(1000,'Bus',2 , 10) ;
INSERT INTO ITINERARI VALUES
(1001,'Aereo',1 , 80) ;
INSERT INTO ITINERARI VALUES
(1002,'Aereo',3 , 100) ;
INSERT INTO ITINERARI VALUES
(5000,'Treno',3 , 10) ;
INSERT INTO ITINERARI VALUES
(5001,'Treno',8 , 40) ;
INSERT INTO ITINERARI VALUES
(5002,'Aereo',3 , 100) ;

```

COMBINAZIONI_SOGGIORNO

CODICE_S	ID_SG
1111	10101
2222	20202
3333	30303

```

INSERT INTO COMBINAZIONI_SOGGIORNO VALUES (1111,10101);
INSERT INTO COMBINAZIONI_SOGGIORNO VALUES (2222,20202);
INSERT INTO COMBINAZIONI_SOGGIORNO VALUES (3333,30303);

```

5.8 QUERY

1 trovare i diversi pacchetti con i rispettivi spostamenti e soggiorni

PACCHETTO	SOGGIORNO	SPOSTAMENTO	SELECT So.Codice_S AS Pacchetto , So.id_sg AS Soggiorno, Sp.Codice_Spostamento AS Spostamento
1111	10101	1000	FROM COMBINAZIONI_SPOSTAMENTO Sp JOIN COMBINAZIONI_SOGGIORNO So ON Sp.Codice_comb=So.Codice_S
1111	10101	5000	
3333	30303	1001	
3333	30303	5001	

2 Trovare per ogni pacchetto il numero di prenotazioni

CODICE	NUMERO_PRENOTAZIONI	SELECT P.Codice, COUNT(*) AS Numero_Prenotazioni
1111	1	FROM PACCHETTI P JOIN PRENOTAZIONI P1 ON P.Codice=P1.codicePacchetti
3333	1	GROUP BY P.Codice
4444	1	

3 Trovare nome e cognome di tutti gli utenti che hanno prenotato un pacchetto del 2019

NOME	COGNOME	SELECT DISTINCT U.Nome, U.Cognome
Mario	Formato	FROM (UTENTI U JOIN PRENOTAZIONI P ON U.CF=P.codiceFiscaleUtenti)
		WHERE EXTRACT(YEAR FROM P.data_partenza)=2019

4 Trovare i mezzi di trasporto utilizzati

TIPO	NUMERO_UTILIZZI	CREATE VIEW Mezzi_usati AS
Bus	1	SELECT M.Tipo , COUNT(*) AS Numero_utilizzi
Treno	2	FROM MEZZI_DI TRASPORTO M JOIN ITINERARI I ON
Aereo	3	M.TIPO=I.TIPO_I
		GROUP BY M.Tipo ;
		select *

from Mezzi_usati ;

5 Trovare il mezzo ut utilizzato più volte

TIPO	select TIPO
Aereo	from Mezzi_usati
	WHERE NUMERO_UTILIZZI =(
	SELECT MAX(M.NUMERO_UTILIZZI)
	FROM MEZZI_USATI M)

6 Trovare per ogni pacchetto il numero di prenotati

CODICE_PACCHETTO	NUMERO_PRENOTAZIONI	SELECT Pa.Codice AS Codice_Pacchetto, COUNT(*)
1111	1	AS Numero_prenotazioni
3333	1	FROM PACCHETTI Pa JOIN PRENOTAZIONI Pr ON
4444	1	Pa.Codice=Pr.codicePacchetti
		GROUP BY Pa.Codice

7 Trovare i pacchetti del 2022 prenotati da utenti con il cognome Falco

PACCHETTO	SELECT P.CodicePacchetti AS Pacchetto
4444	FROM PRENOTAZIONI P JOIN UTENTI U ON P.codiceFiscaleUtenti= U.CF
	WHERE U.Cognome='Falco' AND EXTRACT(YEAR FROM
	P.data_partenza)=2022

8 Trovare i servizi per ogni camera

NUMERO_CAMERA	NUMERO_SERVIZI
11	2
12	2
10	3

```
create view servizi_per_camera as
select c.numero_camera , count (*) numero_servizi
from camere c join servizi s on
c.numero_camera=s.codice_camera
group by c.numero_camera ;
select*
```

from servizi_per_camera ;

9 Trovare la camera con più servizi

NUMERO_CAMERA	NUMERO_SERVIZI
10	3

```
select *
from servizi_per_camera s
where s.numero_servizi = (
```

```
select max(s1.numero_servizi)
```

```
from servizi_per_camera s1
```

```
)
```

10 Trovare il nome degli utenti che hanno prenotato un pacchetto con il soggiorno in un b&b

CF	NOME	COGNOME
FLCNMM00LHGFF	Giovanni	Falco

```
select *
from utenti u
where u.cf=(
```

```
select p.codicefiscaleutenti
```

```
from prenotazioni p
```

```
where p.codicePacchetti=(
```

```
select cs.Codice_s
```

```
from combinazioni_soggiorno cs
```

```
where cs.id_sg =(
```

```

select s.id_sg
from soggiorni s join alberghi al on s.codice_albergo=al.id_c
where al.tipologia='b&b'
)))

```

5.9 TRIGGERS

1 Controlla se la data di partenza è precedente alla data di ritorno

```

CREATE OR REPLACE TRIGGER CONTROLLO_DATE
BEFORE INSERT ON PRENOTAZIONI
FOR EACH ROW
BEGIN
IF :new.data_partenza>:new.data_ritorno
THEN RAISE_APPLICATION_ERROR(-20000, 'Ordine delle date non valido');
END IF;
END;

select *
from prenotazioni ;

insert into prenotazioni values ( 789, 1000 , '3-APR-2022','1-Apr-
2022',5555,'FLCNMMM00LHGFF',293256348952015, NULL);

```

ORA-20000: Ordine delle date non valido ORA-06512: at "SQL_IBNMFFODSLTMKXYGSSYBTXUET.CONTROLLO_DATE", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

2 Impostare come mezzo di trasporto "AEREO" se il viaggio è da Napoli a Palermo

ID_I	TIPO_I	TEMPO_PERCORRENZA	PREZZO
33302	Aereo	75	17
1000	Bus	2	10
1001	Aereo	1	80
1002	Aereo	3	100
5000	Treno	3	10
5001	Treno	8	40
5002	Aereo	3	100

```

CREATE OR REPLACE TRIGGER
Airplane_Na_Pa
AFTER INSERT ON SPOSTAMENTI
FOR EACH ROW
BEGIN
IF :new.luogo_partenza='Napoli' AND
:new.luogo_arrivo='Palermo'
THEN INSERT INTO ITINERARI
VALUES(:new.id, 'Aereo', 75, 17);
END IF;
END;

```

```

INSERT INTO SPOSTAMENTI VALUES (033302, 'Napoli', 'Palermo', '12-mar-2019 02-00-00',
'17-mar-2019 03-15-00', '41°53'57.12" NORD', '40°50'49.20" NORD', '12°32'42.00" SUD',
'14°15'54.00" SUD');

```

```
);
```

```
select *
```

```
FROM itinerari;
```

3 Non permettere all'utente di prenotare un pacchetto se manca meno di una settimana alla partenza

```
CREATE OR REPLACE TRIGGER Scadenza_prenotazioni
```

```
BEFORE INSERT ON PRENOTAZIONI
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF :new.data_partenza<SYSDATE - INTERVAL '7' DAY
```

```
THEN RAISE_APPLICATION_ERROR(-20001, 'pacchetto non valido');
```

```
END IF;
```

```
END;
```

```
SELECT *
```

```
FROM ;
```

```
INSERT INTO PRENOTAZIONI VALUES (107, 100, '12-FEB-2018', '18-FEB-2018', 1111,  
'FLCNMMM00LHGFF', 293256348952015, NULL);
```

ORA-20001: pacchetto non valido ORA-06512: at "SQL_IBNMFFODSLTMKXYGSSYBTXUET.SCADENZA_PRENOTAZIONI", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

5.10 PROCEDURE

1) Numero prenotazioni per utente

```
CREATE OR REPLACE PROCEDURE CALCOLA_NUMERO(  
CFUTENTE PRENOTAZIONI.codiceFiscaleUtenti%TYPE  
)  
AS  
BEGIN  
DECLARE  
NUM_PRENOTAZIONI INTEGER;  
BEGIN  
SELECT COUNT(P.CODICEFISCALEUTENTI) INTO NUM_PRENOTAZIONI  
FROM PRENOTAZIONI P JOIN UTENTI U ON U.CF=P.CODICEFISCALEUTENTI  
WHERE cfutente =U.CF  
GROUP BY P.codiceFiscaleUtenti;  
DBMS_OUTPUT.PUT_LINE('UTENTE ' || CFUTENTE || ' HA COMPIUTO: ' ||  
NUM_PRENOTAZIONI || ' PRENOTAZIONI ');  
END;  
END CALCOLA_NUMERO;  
execute calcola_numero('MFR222RFTMTMTM');  
  
Statement processed.  
UTENTE MFR222RFTMTMTM HA COMPIUTO : 1 PRENOTAZIONI
```

2 Trova il numero di camera per ogni albergo

```
CREATE OR REPLACE PROCEDURE CAMERA_PER_ALBERGO (  
Nome_albergo ALBERGHI.NOME%TYPE )  
AS
```

```

BEGIN
DECLARE
numero_camera_max integer ;
begin
select count(C.numero_camera) INTO numero_camera_max
from alberghi al join camere c on c.codice_alberghi=al.id_c
where al.nome=Nome_albergo
group by al.nome ;
DBMS_OUTPUT.PUT_LINE (' ALBERGO '||NOME_ALBERGO || ' HA '||
numero_camera_max|| ' CAMERE ');
END ;
END CAMERA_PER_ALBERGO ;
execute CAMERA_PER_ALBERGO ('Concaverde') ;

Statement processed.
ALBERGO Concaverde HA 1 CAMERE

```

3 Confronto pacchetti

```

CREATE OR REPLACE PROCEDURE Confronto_pacchetti (
Pacchetto1 PRENOTAZIONI.codicepacchetti%TYPE,
Pacchetto2 PRENOTAZIONI.codicepacchetti%TYPE
)
AS
BEGIN
    DECLARE
        Num_pac1 INTEGER;
        Num_pac2 INTEGER;
    BEGIN
        SELECT COUNT(*) INTO Num_pac1 FROM PRENOTAZIONI WHERE
codicePacchetti=Pacchetto1;

```

```

        SELECT COUNT(*) INTO Num_pac2 FROM PRENOTAZIONI WHERE
codicePacchetti=Pacchetto2;

        IF Num_pac1>Num_pac2
THEN DBMS_OUTPUT.PUT_LINE ('Si hanno più prenotazioni per il pacchetto ' || Pacchetto1);
ELSE IF Num_pac1<Num_pac2
THEN DBMS_OUTPUT.PUT_LINE('Si hanno più prenotazioni per il pacchetto ' || Pacchetto2 );
ELSE
DBMS_OUTPUT.PUT_LINE('Il numero di prenotazione è equivalente ');
END IF;
END IF;
END;

END Confronto_pacchetti;
execute Confronto_pacchetti (1111,2222) ;

Statement processed.
Si hanno più prenotazioni per il pacchetto  1111

```