

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II



DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA IN INFORMATICA
INSEGNAMENTO DI BASI DI DATI I
ANNO ACCADEMICO 2021/2022

Progettazione e sviluppo di una base di dati relazionale per la gestione di un sistema di tracciamento contatti per ristoranti

Autori:

Angelo DI MAIO
MATRICOLA N86003699
angelo.dimaio3@studenti.unina.it

Santolo BARRETTA
MATRICOLA N86003666
san.barretta@studenti.unina.it

Docenti:

Prof. Adriano PERON
Prof. Sergio DI MARTINO
Prof. Luigi Libero Lucio STARACE

23 dicembre 2021

Questa pagina è stata lasciata intenzionalmente bianca.

Indice

Indice	3
Capitolo 1.....	6
Descrizione del progetto	6
1.1 Descrizione sintetica e analisi del problema.....	6
1.2 L'organizzazione del sistema di tracciamento contatti.....	6
Capitolo 2.....	8
Progettazione concettuale	8
Introduzione.....	8
2.1 Alcune convenzioni per la lettura dei class diagram.....	8
2.2 Class diagram	9
2.3 Ristrutturazione del class diagram.....	10
2.3.1 Analisi delle informazioni ridondanti.....	10
2.3.2 Analisi degli identificativi.....	10
2.3.3 Rimozione degli attributi strutturati	10
2.3.4 Rimozione delle classi di associazione.....	11
2.3.5 Rimozione delle gerarchie di specializzazione.....	11
2.3.6 Class diagram ristrutturato.....	12
2.4 Dizionario dei dati per il class diagram ristrutturato	13
2.4.1 Dizionario delle classi.....	13
2.4.2 Dizionario delle associazioni	17
2.4.3 Dizionario dei vincoli	19
2.4.4 Dizionario delle interrogazioni.....	21
Capitolo 3.....	23
Progettazione logica.....	23
Introduzione.....	23
3.1 Schema logico	23
3.1.1 Traduzione in schemi relazionali	23
3.1.2 Traduzione delle associazioni.....	25
3.1.3 Schema logico generale	26

Capitolo 4.....	27
Progettazione fisica.....	27
Introduzione.....	27
4.1 Note sull'implementazione.....	27
4.2 Definizione delle tabelle.....	27
4.2.1 Definizione della tabella PROPRIETARIO	28
4.2.2 Definizione della tabella RISTORANTE	28
4.2.3 Definizione della tabella MANAGERRISTORANTE.....	29
4.2.4 Definizione della tabella SALA	29
4.2.5 Definizione della tabella CAMERIERE	30
4.2.6 Definizione della tabella TAVOLO	30
4.2.7 Definizione della tabella TAVOLATA	31
4.2.8 Definizione della tabella AVVENTORE.....	32
4.2.9 Definizione della tabella CASO	33
4.3 Funzioni, procedure ed altre automazioni	34
4.3.1 Stored function IS_NUMBER.....	34
4.3.2 Stored procedure NUMERO_DI_TELEFONO_LEGALE	34
4.3.3 Stored procedure PASSWORD_LEGALE.....	35
4.4 Viste	36
4.4.1 Vista TAVOLATE	36
4.5 Implementazione dei vincoli	37
4.5.1 Implementazione del vincolo Password legale	37
4.5.2 Implementazione del vincolo Numero di telefono legale	38
4.5.3 Implementazione del vincolo Cap legale.....	39
4.5.4 Implementazione del vincolo Età cameriere legale.....	39
4.5.5 Implementazione del vincolo MaxAvventori legale.....	40
4.5.6 Implementazione del vincolo Capienza legale.....	40
4.5.7 Implementazione del vincolo Data nascita legale	41
4.5.8 Implementazione del vincolo Has greenpass	41
4.5.9 Implementazione del vincolo Temperatura avventore legale.....	42
4.5.10 Implementazione del vincolo Somma avventori a tavolata legale	42
4.5.11 Implementazione del vincolo Data registrazione caso	43

Capitolo 5	44
Esempio d'uso	44
Introduzione	44
5.1 Popolamento del database	44
5.2 Prospettiva delle tabelle implementate su ORACLE	50
5.3 Implementazione delle query	53
5.3.1 Numero giornaliero di avventori per ristorante	53
5.3.2 Numero mensile di avventori per ristorante	53
5.3.3 Numero giornaliero di avventori per tutti i ristoranti di un proprietario	53
5.3.4 Numero mensile di avventori per tutti i ristoranti di un proprietario	54
5.3.5 Casi positivi di un determinato ristorante per data di arrivo della tavolata	54
5.3.6 Casi positivi di un determinato ristorante per mese di arrivo della tavolata	54
5.3.7 Casi positivi di un determinato ristorante per anno di arrivo della tavolata	54
5.3.8 Casi positivi di tutti i ristoranti di un proprietario per data di arrivo della tavolata	55
5.3.9 Casi positivi di tutti i ristoranti di un proprietario per mese di arrivo della tavolata	55
5.3.10 Casi positivi di tutti i ristoranti di un proprietario per anno di arrivo della tavolata	55
5.3.11 Informazioni sugli avventori risultati positivi in un ristorante	55
5.3.12 Informazioni sugli avventori risultati positivi in tutti i ristoranti di un proprietario	56
5.3.13 Informazioni sui camerieri risultati positivi in un ristorante	56
5.3.14 Informazioni sui camerieri risultati positivi in tutti i ristoranti di un proprietario	56
5.3.15 Avventori positivi con o senza green pass	57
5.3.16 Numero di avventori medio per tavolata di un ristorante	57

Capitolo 1

Descrizione del progetto

1.1 Descrizione sintetica e analisi del problema

Si provvederà alla progettazione e allo sviluppo di una base di dati relazionale per la gestione del tracciamento di contatti COVID-19 in ristoranti. Il sistema permetterà di tenere traccia dei contatti di un avventore in un ristorante, fornendo ulteriori informazioni per il tracciamento degli eventuali avventori risultati positivi al virus SARS-CoV-2. In particolare, il sistema deve permettere di gestire il tracciamento dei contatti di uno o più ristoranti. Ciascun ristorante è organizzato in una o più sale, identificate da un nome, le quali contengono uno o più tavoli. Ciascun tavolo presente all'interno di una sala sarà identificato da un codice univoco, avrà l'indicazione del numero massimo di avventori che possono sedersi a quel tavolo e dei tavoli adiacenti ad esso. Il sistema viene gestito da operatori che tengono traccia delle tavolate formate dagli avventori nei ristoranti. Un avventore sarà identificato da un nome, un cognome, un numero di carta d'identità e un numero di telefono. La tavolata viene identificata dalla data di arrivo degli avventori, i camerieri che l'hanno servita e il tavolo ad essa assegnata. I ristoranti effettueranno soltanto servizio serale nella fascia oraria 20 – 22, quindi in ciascuna data, al più una tavolata deve essere associata ad un dato tavolo, poiché il servizio è unico. Inoltre, il numero di avventori che partecipano ad una tavolata non deve superare il numero massimo di avventori che possono sedersi al tavolo a cui essa è assegnata. Infine, il sistema dovrà permettere la visualizzazione di alcune statistiche su base mensile e giornaliera, tra cui il numero totale di avventori accolti da ciascun ristorante.

1.2 L'organizzazione del sistema di tracciamento contatti

Di seguito verrà descritta brevemente la **chiave di lettura** data al sistema da implementare, sulla quale si basa la progettazione della base di dati relazionale.

Si suppone che il sistema di tracciamento contatti COVID-19 venga utilizzato da uno o più ristoranti appartenenti ad una stessa catena, ubicati fisicamente nel territorio italiano e con gruppi di avventori accolti prevalentemente italiani. Il sistema verrà gestito dalla figura dell'**Operatore** il quale, una volta autenticato nella piattaforma sviluppata, con un **Username** e una **Password**, si occuperà di alcuni aspetti rilevanti, fra cui:

- **la gestione e l'inserimento nella piattaforma di tavolate di avventori:** si intende come **Tavolata** l'atto di associare i tavoli presenti in una sala del ristorante ad un singolo o ad un gruppo di avventori. La tavolata verrà identificata quindi da una data di arrivo degli avventori facenti parte della stessa e da un orario di arrivo e uno di uscita, che coincideranno sempre con quelli dell'unico servizio serale effettuato (nella fascia oraria 20-22). L'**Operatore** che gestisce la piattaforma si occuperà poi di registrare anche le informazioni sugli avventori facenti parte di una determinata tavolata, tra cui: le informazioni anagrafiche, la temperatura registrata dall'avventore al suo arrivo al ristorante (se oltre i 37.5° non sarà possibile registrare l'**Avventore** nel sistema) ed inoltre se possiede o meno green pass (se un **Avventore** non

possiede green pass non sarà possibile associarlo ad un tavolo presente in una sala Interna ma solo ad un tavolo presente in una sala Esterna, se il ristorante ne è dotato). In questo modo gli operatori che gestiscono il sistema terranno traccia di tutte le tavolate formate in un ristorante;

- **la gestione e l’inserimento nella piattaforma di casi COVID-19 avvenuti nel ristorante:** nel caso in cui un avventore registrato ad una tavolata in un determinato ristorante risultasse positivo nei giorni seguenti al virus SARS-CoV-2, dopo sua preventiva segnalazione al ristorante (ad esempio tramite form online o telefonata), sarà compito degli operatori aprire un nuovo Caso nella piattaforma e gestirlo. La gestione di un Caso comprende l’inserimento in piattaforma dei dati dell’avventore risultato positivo, quindi la sua anagrafica, la data di registrazione del caso, il suo stato e delle eventuali note associate al caso. Lo stesso vale anche per eventuali camerieri risultati positivi. Si suppone che lo stato di un caso possa essere “NonRisolto” quando viene solamente inserito in piattaforma, “InRisoluzione” quando un operatore prenderà in carico il caso e avviserà tutti gli avventori che hanno avuto contatti con l’avventore risultato positivo e “Risolto” quando saranno stati avvisati tutti gli avventori venuti in contatto con l’avventore positivo. In questo modo il sistema permetterà di tenere traccia dei contatti di un avventore in un ristorante.

La figura dell’Operatore che gestisce l’intero sistema descritto può essere svolta da:

- uno o più Manager che gestiscono un determinato ristorante, i quali avranno dei privilegi limitati nel sistema relativi solo alla gestione di casi e all’inserimento di tavolate;
- direttamente dal Proprietario che amministra il ristorante o la catena formata da più ristoranti ed il quale avrà privilegi “ad hoc” nel sistema come la possibilità di aggiungere, modificare o rimuovere ristoranti, sale, tavoli presenti in piattaforma, oltre che la gestione del personale presente per quanto riguarda le figure del Manager e del Cameriere.

Per ulteriori chiarimenti generali si rimanda al successivo capitolo riguardante la **progettazione concettuale**, in cui è definito il **class diagram** della base di dati e viene riportato il **dizionario dei dati**.

Capitolo 2

Progettazione concettuale

Introduzione

In questo capitolo viene affrontata la progettazione della base di dati al livello di astrazione più alto, ossia quello della progettazione concettuale. Dall'analisi iniziale del problema e dei requisiti che devono essere soddisfatti si arriverà ad uno schema concettuale indipendente dalla struttura dei dati e dall'implementazione fisica della base di dati. Lo schema concettuale verrà rappresentato usando un class diagram UML, in cui saranno presenti le entità rilevanti individuate ai fini della rappresentazione dei dati e le relazioni che intercorrono tra esse.

2.1 Alcune convenzioni per la lettura dei class diagram

Al fine di semplificare la lettura dei class diagram UML che seguono si è scelto di adottare le seguenti convenzioni:

- tutti gli attributi, salvo quelli in cui è specificata, hanno cardinalità $(1, 1)$ o brevemente 1, ossia sono attributi a valore singolo totali;
- per le associazioni: la cardinalità di una partecipazione è situata a destra se la linea di collegamento è verticale, in alto se la linea di collegamento è orizzontale;
- le enumerazioni sono identificabili dal colore grigio.

NOTA - Nel caso in cui i class diagram non risultassero essere abbastanza leggibili si riportano di seguito i link alle rispettive versioni digitali esterne:

- **Class diagram:** <https://ibb.co/7WC7zWC>
- **Class diagram ristrutturato:** <https://ibb.co/VpHcWvx>

2.2 Class diagram

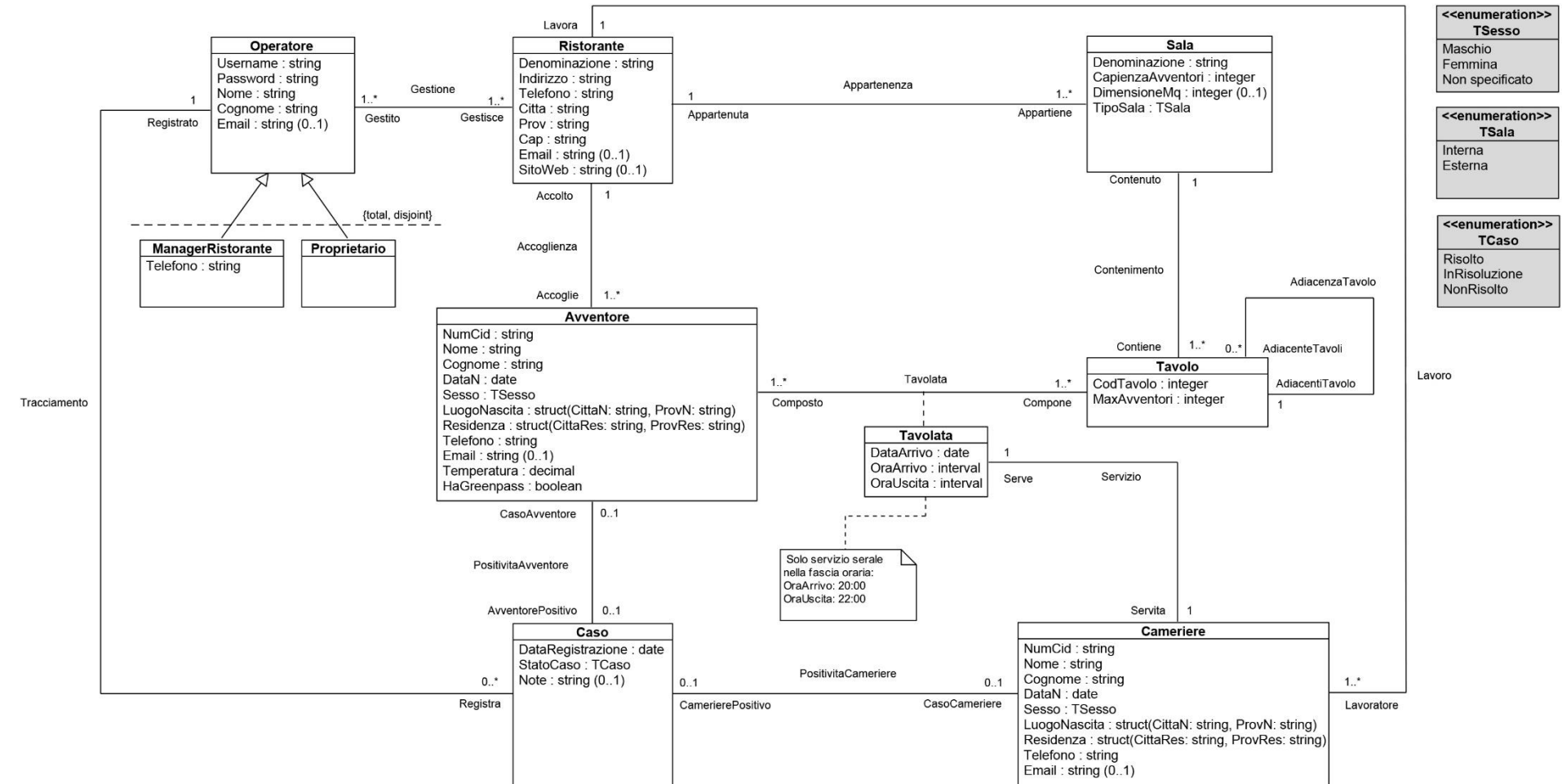


Immagine 2.1: Class diagram della base di dati

2.3 Ristrutturazione del class diagram

Al fine di rendere il class diagram conforme al modello relazionale che sarà adottato successivamente nella fase di progettazione logica e per migliorare l'efficienza dell'implementazione fisica si procede alla ristrutturazione dello stesso. Al termine del procedimento di ristrutturazione il class diagram non conterrà attributi strutturati e gerarchie di specializzazione.

2.3.1 Analisi delle informazioni ridondanti

Sono presenti due informazioni ridondanti nel class diagram di partenza. La prima riguarda gli attributi `OraArrivo` e `OraUscita` appartenenti alla classe di associazione `Tavolata`. Tali attributi avranno sempre lo stesso valore per ogni istanza di `Tavolata`, poiché dall'analisi dei requisiti sappiamo che ogni singolo ristorante effettuerà soltanto servizio serale nella fascia oraria `OraArrivo: 20:00`, `OraUscita: 22:00`. Per completezza si è deciso di non rimuovere i suddetti attributi nella ristrutturazione del class diagram. La seconda informazione ridondante riguarda l'attributo `CapienzaAvventori` appartenente all'entità `Sala`. È infatti possibile ottenere la `CapienzaAvventori` di una determinata `Sala` sommando l'attributo `MaxAvventori` dell'entità `Tavolo`, per tutti i tavoli appartenenti alla `Sala` presa in considerazione. Anche in questo caso, per completezza, si è deciso di non rimuovere l'attributo nella fase di ristrutturazione poiché si è pensato che in ottica pandemica, la capienza di avventori per una determinata sala rimanga fissa per un lungo periodo di tempo nello schema della base di dati e non variabile a seconda dei tavoli che vengono associati ad una sala. In conclusione, per ogni `Sala` avremo una capienza fissa ed un massimo numero di tavoli che possono essere ad essa associati, la cui somma totale di `MaxAvventori` ovviamente dovrà essere minore o uguale alla `CapienzaAvventori` della `Sala`.

2.3.2 Analisi degli identificativi

Per alcune entità del class diagram è stata decisa l'introduzione di chiavi primarie "surrogate" in modo da evitare l'impiego di chiavi candidate composte da più attributi. Tali chiavi primarie saranno degli identificativi numerici che permetteranno di discriminare con maggiore facilità le istanze. Le chiavi primarie surrogate introdotte sono le seguenti: `CodProprietario`, `CodManager`, `CodRistorante`, `CodSala`, `CodTavolo`, `CodTavolata`. Le entità `Avventore` e `Cameriere` saranno identificate dalla chiave primaria `NumCid` ossia una stringa che corrisponde al numero di carta d'identità, univoco per ogni persona.

2.3.3 Rimozione degli attributi strutturati

È necessario gestire gli attributi strutturati `LuogoNascita` e `Residenza` presenti rispettivamente nelle entità `Avventore` e `Cameriere`. È stato deciso di estendere gli attributi che formano gli strutturati `LuogoNascita` e `Residenza` nell'entità cui appartengono, rimuovendo questi ultimi. Dunque, alle entità `Avventore` e `Cameriere` verranno aggiunti i seguenti attributi: `CittaN`, `ProvN` (che costituiscono l'attributo strutturato `LuogoNascita`) e `CittaRes`, `ProvRes` (che costituiscono l'attributo strutturato `Residenza`).

È stata scelta questa gestione poiché tali attributi potrebbero essere utili per alcune interrogazioni riguardanti statistiche sul tracciamento dei contatti (ad esempio per stabilire quante persone di una determinata CittàRes siano risultate positive in un determinato ristorante, su base giornaliera o mensile).

2.3.4 Rimozione delle classi di associazione

Si procede con l'eliminazione della classe di associazione Tavolata e la reintroduzione della stessa come entità, contenente gli stessi attributi della classe di associazione con l'aggiunta di un identificativo numerico, CodTavolata. Sono state introdotte dunque due nuove associazioni:

- Partecipazione fra le entità Avventore e Tavolata;
- Composizione fra le entità Tavolo e Tavolata.

Ciò faciliterà la creazione e la gestione delle tavolate di avventori.

2.3.5 Rimozione delle gerarchie di specializzazione

Si procede con l'eliminazione della specializzazione riguardante l'entità Operatore che può essere specializzata in Proprietario oppure ManagerRistorante. Si tratta di una specializzazione totale e disgiunta; per tanto si procede alla sua eliminazione attraverso la “compressione” della superclasse nelle sottoclassi, le quali ereditano gli attributi e le associazioni della superclasse Operatore. Sono quindi state aggiunte le associazioni:

- Amministrazione e Gestione (in sostituzione della precedente, Gestione);
- TracciamentoProprietario e TracciamentoManager (in sostituzione della precedente, Tracciamento).

In questo modo la figura dell'operatore generico che gestirà la piattaforma per il tracciamento verrà dettagliata maggiormente, così come descritto al punto **1.2**.

2.3.6 Class diagram ristrutturato

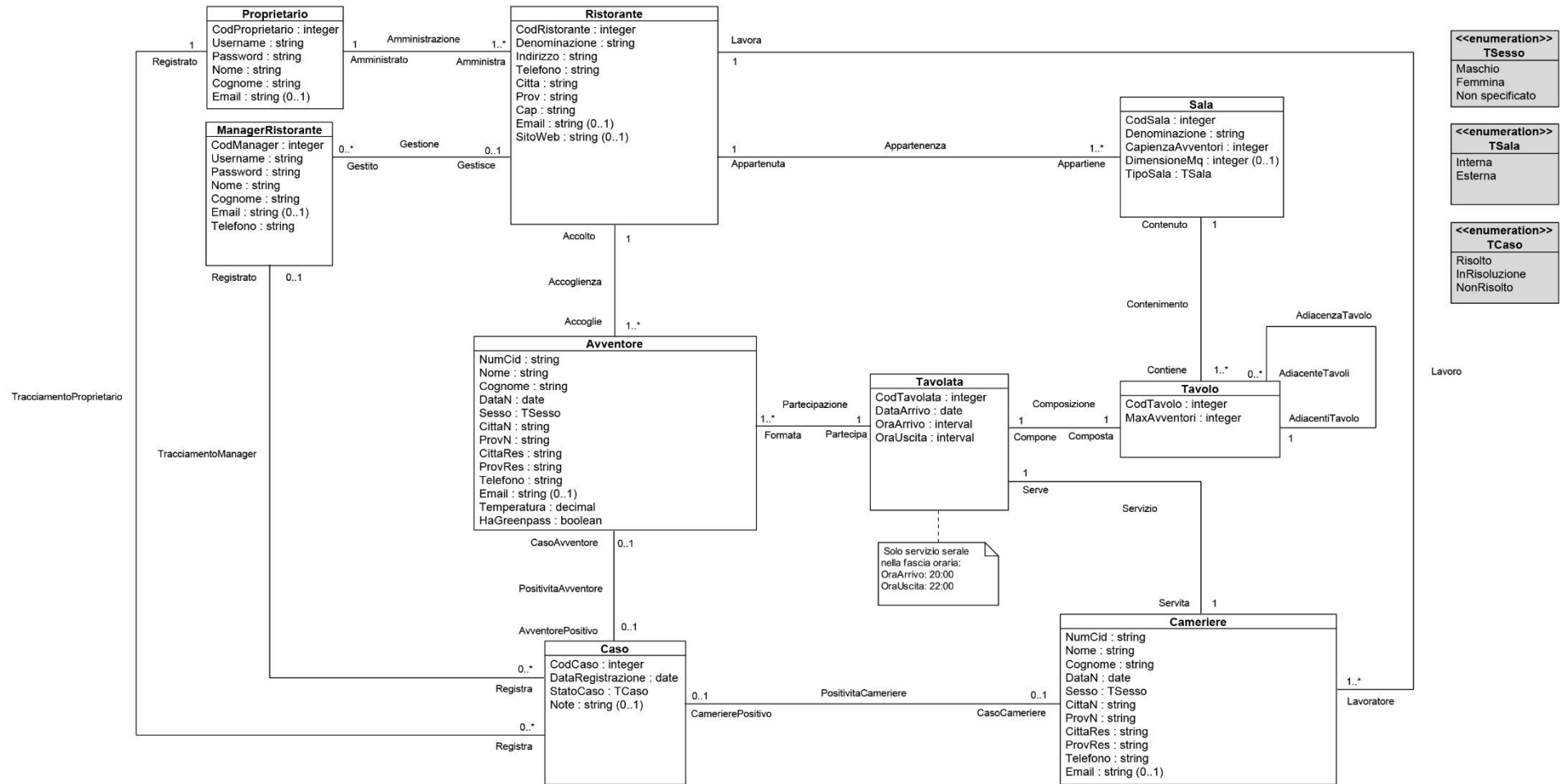


Immagine 2.2: Class diagram ristrutturato della base di dati

2.4 Dizionario dei dati per il class diagram ristrutturato

2.4.1 Dizionario delle classi

<i>Classe</i>	<i>Descrizione</i>	<i>Attributi</i>
Proprietario	Descrittore di un generico proprietario di uno o più ristoranti	CodProprietario (<i>integer</i>): chiave surrogata che serve a identificare un proprietario. Username (<i>string</i>): username associato ad un proprietario. Password (<i>string</i>): password associata ad un proprietario. Nome (<i>string</i>): nome di un proprietario. Cognome (<i>string</i>): cognome di un proprietario. Email (<i>string, opzionale</i>): email associata ad un proprietario.
ManagerRistorante	Descrittore del manager di un ristorante	CodManager (<i>integer</i>): chiave surrogata che serve a identificare univocamente un manager di un ristorante. Username (<i>string</i>): username associato al manager. Password (<i>string</i>): password associata al manager. Nome (<i>string</i>): nome di un manager. Cognome (<i>string</i>): cognome associato di un manager. Email (<i>string, opzionale</i>): email associata al manager. Telefono (<i>string</i>): numero di telefono associato al manager.

Ristorante	Descrittore di un ristorante	<p>CodRistorante (<i>integer</i>): chiave surrogata che serve a identificare univocamente un ristorante.</p> <p>Denominazione (<i>string</i>): nome di un ristorante.</p> <p>Indirizzo (<i>string</i>): indirizzo di un ristorante.</p> <p>Telefono (<i>string</i>): numero di telefono di un ristorante.</p> <p>Citta (<i>string</i>): città di ubicazione di un ristorante.</p> <p>Prov (<i>string</i>): provincia di ubicazione di un ristorante.</p> <p>Cap (<i>string</i>): cap della provincia di ubicazione di un ristorante.</p> <p>Email (<i>string, opzionale</i>): email associata ad un ristorante.</p> <p>SitoWeb (<i>string, opzionale</i>): sito web associato ad un ristorante.</p>
Sala	Descrittore della sala di un ristorante	<p>CodSala (<i>integer</i>): chiave surrogata che serve a identificare univocamente una sala.</p> <p>Denominazione (<i>string</i>): nome di una sala.</p> <p>CapienzaAvventori (<i>integer</i>): capienza massima di avventori per una sala.</p> <p>DimensioneMq (<i>string, opzionale</i>): dimensione in metri quadri di una sala.</p> <p>TipoSala (<i>TSala</i>): attributo che indica se la sala è interna oppure esterna.</p>
Tavolo	Descrittore di un tavolo del ristorante	<p>CodTavolo (<i>integer</i>): chiave surrogata che serve a identificare univocamente un tavolo.</p> <p>MaxAvventori (<i>integer</i>): numero massimo di avventori per ogni tavolo.</p>

Tavolata	Descrittore della tavolata di avventori	<p>CodTavolata (<i>integer</i>): chiave surrogata che serve a identificare univocamente ciascuna tavolata.</p> <p>DataArrivo (<i>date</i>): data in cui si svolge la tavolata.</p> <p>OraArrivo (<i>interval</i>): orario di arrivo della tavolata (vale sempre 20:00).</p> <p>OraUscita (<i>interval</i>): orario di uscita della tavolata (vale sempre 22:00).</p>
Avventore	Descrittore di un generico avventore	<p>NumCid (<i>string</i>): numero della carta di identità che serve anche da chiave primaria per identificare univocamente ciascun avventore.</p> <p>Nome (<i>string</i>): nome di un avventore.</p> <p>Cognome (<i>string</i>): cognome di un avventore.</p> <p>DataN (<i>date</i>): data di nascita di un avventore.</p> <p>Sesso (<i>TSesso</i>): sesso di un avventore.</p> <p>CittaN (<i>string</i>): città di nascita di un avventore.</p> <p>ProvN (<i>string</i>): provincia di nascita di un avventore.</p> <p>CittaRes (<i>string</i>): città di residenza di un avventore.</p> <p>ProvRes (<i>string</i>): provincia di residenza di un avventore.</p> <p>Telefono (<i>string</i>): numero di telefono di un avventore.</p> <p>Email (<i>string, opzionale</i>): email associata ad un avventore.</p> <p>Temperatura (<i>decimal</i>): temperatura misurata all'ingresso di un ristorante.</p> <p>HaGreenpass (<i>boolean</i>): indica il possesso o meno del green pass da parte di un avventore.</p>

Cameriere	Descrittore di un cameriere del ristorante	<p>NumCid (<i>string</i>): numero della carta di identità che serve anche da chiave primaria per identificare univocamente ciascun cameriere.</p> <p>Nome (<i>string</i>): nome di un cameriere.</p> <p>Cognome (<i>string</i>): cognome di un cameriere.</p> <p>DataN (<i>date</i>): data di nascita di un cameriere.</p> <p>Sesso (<i>TSesso</i>): sesso di un cameriere.</p> <p>CittaN (<i>string</i>): città di nascita di un cameriere.</p> <p>ProvN (<i>string</i>): provincia di nascita di un cameriere.</p> <p>CittaRes (<i>string</i>): città di residenza di un cameriere.</p> <p>ProvRes (<i>string</i>): provincia di residenza di un cameriere.</p> <p>Telefono (<i>string</i>): numero di telefono di un cameriere.</p> <p>Email (<i>string, opzionale</i>): email associata ad un cameriere.</p>
Caso	Descrittore di un caso covid	<p>CodCaso (<i>integer</i>): chiave surrogata che serve per identificare univocamente ciascun caso.</p> <p>DataRegistrazione (<i>date</i>): data di registrazione di un caso.</p> <p>StatoCaso (<i>TStato</i>): stato di un caso (Risolto, InRisoluzione, NonRisolto).</p> <p>Note (<i>string, opzionale</i>): note relative al caso.</p>

Tabella 2.1: Dizionario delle classi - fine.

2.4.2 Dizionario delle associazioni

Associazione	Descrizione	Classi coinvolte
Amministrazione	Esprime l'amministrazione da parte di un proprietario di uno o più ristoranti	<p>Proprietario [1] ruolo (Amministrato): indica il proprietario che amministra uno o più ristoranti.</p> <p>Ristorante [1..*] ruolo (Amministra): indica il ristorante che viene amministrato da un proprietario.</p>
Gestione	Esprime la possibile gestione da parte di uno o più manager di un ristorante.	<p>ManagerRistorante [0..*] ruolo (Gestito): indica il manager che può gestire un ristorante.</p> <p>Ristorante [0..1] ruolo (Gestisce): indica il ristorante che può essere gestito da uno o più manager.</p>
Appartenenza	Esprime l'appartenenza di una o più sale ad un ristorante.	<p>Ristorante [1] ruolo (Appartenuta): indica il ristorante a cui appartengono una o più sale.</p> <p>Sala [1..*] ruolo (Appartiene): indica una sala che appartiene ad un ristorante.</p>
Contenimento	Esprime il contenimento di tavoli da parte di una sala.	<p>Sala [1] ruolo (Contenuto): indica una sala contenente uno o più tavoli.</p> <p>Tavolo [1..*] ruolo (Contiene): indica un tavolo contenuto in una sala.</p>
Servizio	Esprime il servizio di un cameriere ad una tavolata.	<p>Cameriere [1] ruolo (Servita): indica il cameriere che serve una tavolata</p> <p>Tavolata [1] ruolo (Serve): indica la tavolata servita da un cameriere.</p>
Partecipazione	Esprime uno o più avventori che partecipano ad una tavolata.	<p>Tavolata [1..*] ruolo (Partecipa): indica la tavolata a cui partecipa uno o più avventori.</p> <p>Avventore [1] ruolo (Formata): indica uno o più avventori che partecipano ad una tavolata.</p>

Composizione	Esprime la composizione di una tavolata con un tavolo.	<p>Tavolata [1] ruolo (Compone): indica la tavolata composta da un tavolo.</p> <p>Tavolo [1] ruolo (Composta): indica il tavolo che compone una tavolata.</p>
AdiacenzaTavolo	Esprime la possibile adiacenza di altri tavoli ad un tavolo.	<p>Tavolo [1] ruolo (AdiacentiTavolo): indica il tavolo a cui possono essere adiacenti altri tavoli.</p> <p>Tavolo [0..*] ruolo (AdiacenteTavoli): indica la possibilità di adiacenza di altri tavoli ad un tavolo.</p>
Lavoro	Esprime il rapporto lavorativo tra uno o più camerieri ed un ristorante.	<p>Cameriere [1..*] ruolo (Lavoratore): indica il cameriere che lavora per un ristorante.</p> <p>Ristorante [1] ruolo (Lavora): indica il ristorante per cui lavorano uno o più camerieri.</p>
Accoglienza	Esprime l'accoglienza da parte di un ristorante di uno o più avventori.	<p>Ristorante [1] ruolo (Accolto): indica il ristorante che accoglie uno o più avventori</p> <p>Avventore [1..*] ruolo (Accoglie): indica uno o più avventori accolti da un ristorante.</p>
PositivitaAvventore	Esprime la possibilità di esistenza di un caso di positività per un avventore.	<p>Avventore [0..1] ruolo (CasoAvventore): indica l'avventore che può risultare positivo.</p> <p>Caso [0..1] ruolo (AvventorePositivo): indica il possibile caso di un avventore positivo.</p>
PositivitaCameriere	Esprime la possibilità di esistenza di un caso di positività per un cameriere.	<p>Caso [0..1] ruolo (CamerierePositivo): indica il cameriere che può risultare positivo.</p> <p>Cameriere [0..1] ruolo (CasoCameriere): indica il possibile caso di un cameriere positivo.</p>

TracciamentoProprietario	Esprime il tracciamento da parte di un proprietario di uno o più possibili casi.	Proprietario [1] ruolo (Registrato): indica il proprietario che registra uno o più possibili casi. Caso [0..*] ruolo (Registra): indica uno o più possibili casi registrati da un proprietario.
TracciamentoManager	Esprime il possibile tracciamento da parte di un manager di uno o più possibili casi.	Manager [0..1] ruolo (Registrato): indica il manager che può registrare uno o più possibili casi. Caso [0..*] ruolo (Registra): indica uno o più possibili casi registrati da un manager.

Tabella 2.2: Dizionario delle associazioni - fine.

2.4.3 Dizionario dei vincoli

Vincolo	Tipo	Descrizione
Has greenpass	Interrelazionale	Se l'attributo <i>HaGreenPass</i> dell'entità <i>Avventore</i> vale 'F' (<i>falso</i>), allora un avventore può partecipare solo ad una tavolata composta da un tavolo la cui ubicazione è in una sala di tipo 'Esterna'.
Email legale	Dominio	Vale per tutte le entità che possiedono l'attributo <i>Email</i> : <i>Proprietario</i> , <i>ManagerRistorante</i> , <i>Ristorante</i> , <i>Avventore</i> , <i>Cameriere</i> . L' <i>Email</i> deve rispettare la forma standard ovvero: contenere almeno un carattere prima della @, almeno un carattere tra essa e il punto e almeno due caratteri nella parte finale dopo il punto.
Password legale	Dominio	Vale per tutte le entità che possiedono l'attributo <i>Password</i> : <i>Proprietario</i> , <i>ManagerRistorante</i> . La <i>Password</i> deve contenere minimo 8 caratteri e rispettare il seguente formato: deve esserci almeno una lettera e un numero.
Sito web legale	Dominio	Il <i>SitoWeb</i> deve rispettare la forma standard ovvero: contenere tre caratteri iniziali corrispondenti a "www" seguiti da un punto, successivamente almeno due caratteri ed infine un punto seguito da almeno due caratteri.
Numero di telefono legale	Dominio	Vale per tutte le entità che possiedono l'attributo <i>Telefono</i> : <i>ManagerRistorante</i> , <i>Ristorante</i> , <i>Avventore</i> , <i>Cameriere</i> . Il numero di <i>Telefono</i> può contenere solo numeri ad eccezione del carattere iniziale '+' per il prefisso.
Data nascita legale	Interrelazionale	La <i>DataN</i> di un avventore deve essere precedente o uguale alla <i>DataArrivo</i> della tavolata a cui esso partecipa.

Orario tavolata legale	Dominio	Gli attributi <i>OraArrivo</i> e <i>OraUscita</i> di tavolata sono di default sempre uguali a <i>OraArrivo: 20:00</i> , <i>OraUscita: 22:00</i> .
CAP legale	Dominio	Il <i>Cap</i> deve rispettare il formato standard italiano, ossia deve essere formato esattamente da 5 caratteri numerici.
CapienzaAvventori legale	Dominio	La <i>CapienzaAvventori</i> di una sala deve essere maggiore di 0.
DimensioneMq legale	Dominio	La <i>DimensioneMq</i> di una sala deve essere maggiore di 0.
MaxAvventori legale	Interrelazionale	Il valore di <i>MaxAvventori</i> di un tavolo deve essere maggiore di 0 e minore uguale del valore di <i>CapienzaAvventori</i> della sala in cui il tavolo è contenuto.
Capienza legale	Interrelazionale	La somma totale di <i>MaxAvventori</i> per tutti i tavoli di una sala deve essere minore o uguale alla <i>CapienzaAvventori</i> della sala che contiene i tavoli in questione.
Somma avventori a tavolata legale	Interrelazionale	La somma totale di tutti gli avventori partecipanti ad una stessa tavolata deve essere minore o uguale al <i>MaxAvventori</i> del tavolo che compone la tavolata.
Data registrazione caso legale	Interrelazionale	La <i>DataRegistrazione</i> di un caso deve essere successiva alla <i>DataN</i> dell'avventore o cameriere a cui il caso è associato.
Temperatura avventore legale	Dominio	La <i>Temperatura</i> registrata di un avventore deve essere maggiore o uguale a 35 e minore uguale a 37,5.
Età cameriere legale	Dominio	L'età di un cameriere deve essere maggiore o uguale a 18 anni.
Unica composizione tavolo a tavolata	Interrelazionale	Per ogni <i>DataArrivo</i> , un tavolo può essere associato al più ad una tavolata.
Unico username proprietario	Intrarelazionale	L' <i>Username</i> di un proprietario deve essere unico.
Unico username manager	Intrarelazionale	L' <i>Username</i> di un manager deve essere unico.
TSesso	Dominio	L'attributo <i>Sesso</i> può assumere solo i valori: 'Maschio', 'Femmina', 'Non specificato'.

TSala	Dominio	L'attributo <i>TipoSala</i> può assumere solo i valori: 'Interna', 'Esterna'.
TCaso	Dominio	L'attributo <i>StatoCaso</i> può assumere solo i valori: 'Risolto', 'InRisoluzione', 'NonRisolto'.

Tabella 2.3: Dizionario dei vincoli - fine.

2.4.4 Dizionario delle interrogazioni

<i>Interrogazione</i>	<i>Descrizione</i>	<i>Frequenza</i>	<i>Classi coinvolte</i>
Numero giornaliero di avventori per ristorante	La query, dato il codice di un ristorante e specificata una data, calcola il numero totale di avventori che sono stati accolti dal ristorante in quella data.	Può essere eseguita giornalmente	Ristorante, Avventore, Tavolata
Numero mensile di avventori per ristorante	La query, dato il codice di un ristorante e specificato il mese di un determinato anno, calcola il numero totale di avventori che sono stati accolti dal ristorante in quel mese dell'anno specificato.	Può essere eseguita mensilmente ma anche giornalmente	Ristorante, Avventore, Tavolata
Numero giornaliero di avventori per tutti i ristoranti di un proprietario	La query, dato il codice di un proprietario e specificata una data, calcola il numero totale di avventori che sono stati accolti da tutti i ristoranti appartenenti al proprietario, in quella data.	Può essere eseguita giornalmente	Ristorante, Avventore, Tavolata
Numero mensile di avventori per tutti i ristoranti di un proprietario	La query, dato il codice di un proprietario e specificato il mese di un determinato anno, calcola il numero totale di avventori che sono stati accolti da tutti i ristoranti appartenenti al proprietario, in quel mese dell'anno.	Può essere eseguita mensilmente ma anche giornalmente	Ristorante, Avventore, Tavolata
Casi positivi di un determinato ristorante per data di arrivo della tavolata	La query, dato il codice di un ristorante e specificata una data di arrivo, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti dal ristorante in quella data.	Può essere eseguita giornalmente	Ristorante, Tavolata, Avventore, Caso
Casi positivi di un determinato ristorante per mese di arrivo della tavolata	La query, dato il codice di un ristorante e specificato il mese di un anno in cui si è svolta la tavolata, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti dal ristorante in quel mese dell'anno.	Può essere eseguita mensilmente ma anche giornalmente	Ristorante, Tavolata, Avventore, Caso
Casi positivi di un determinato ristorante per anno di arrivo della tavolata	La query, dato il codice di un ristorante e specificato un anno in cui si è svolta la tavolata, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti dal ristorante in quell'anno.	Può essere eseguita annualmente ma anche mensilmente	Ristorante, Tavolata, Avventore, Caso

Casi positivi di tutti i ristoranti di un proprietario per data di arrivo della tavolata	La query, dato il codice di un proprietario e specificata una data di arrivo, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti da tutti i ristoranti appartenenti al proprietario in quella data.	Può essere eseguita giornalmente	Ristorante, Avventore, Tavolata, Caso
Casi positivi di tutti i ristoranti di un proprietario per mese di arrivo della tavolata	La query, dato il codice di un proprietario e specificato il mese di un anno in cui si è svolta la tavolata, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti nei ristoranti appartenenti al proprietario in quel mese dell'anno.	Può essere eseguita mensilmente ma anche giornalmente	Ristorante, Avventore, Tavolata, Caso
Casi positivi di tutti i ristoranti di un proprietario per anno di arrivo della tavolata	La query, dato il codice di un proprietario e specificato un anno in cui si è svolta la tavolata, calcola il numero totale di avventori risultati successivamente positivi che sono stati accolti nei ristoranti appartenenti al proprietario in quell'anno.	Può essere eseguita annualmente ma anche mensilmente	Ristorante, Avventore, Tavolata, Caso
Informazioni sugli avventori risultati positivi in un ristorante	La query, dato un ristorante, permette di visualizzare l'anagrafica e alcune informazioni sul relativo caso di tutti gli avventori risultati positivi nel ristorante.	Può essere eseguita giornalmente	Ristorante, Avventore, Caso, Tavolata, Tavolo
Informazioni sugli avventori risultati positivi in tutti i ristoranti di un proprietario	La query, dato un determinato proprietario, permette di visualizzare l'anagrafica e alcune informazioni sul relativo caso di tutti gli avventori risultati positivi nei ristoranti da lui amministrati.	Può essere eseguita giornalmente	Ristorante, Avventore, Caso, Tavolata, Tavolo
Informazioni sui camerieri risultati positivi in un ristorante	La query, dato un ristorante, permette di visualizzare l'anagrafica e alcune informazioni sul relativo caso di tutti i camerieri risultati positivi nel ristorante.	Può essere eseguita giornalmente	Ristorante, Cameriere, Caso
Informazioni sui camerieri risultati positivi in tutti i ristoranti di un proprietario	La query, dato un determinato proprietario, permette di visualizzare l'anagrafica e alcune informazioni sul relativo caso di tutti i camerieri risultati positivi nei ristoranti da lui amministrati.	Può essere eseguita giornalmente	Ristorante, Cameriere, Caso
Avventori positivi con o senza green pass	La query mostra il numero complessivo totale di casi di avventori risultati positivi che possiedono green pass. Lo stesso per gli avventori risultati positivi sprovvisti di green pass.	Può essere eseguita giornalmente	Avventore, Caso
Numero di avventori medio per tavolata di un ristorante	La query, dato un determinato ristorante, mostra il numero di avventori medio per tavolata.	Può essere eseguita giornalmente ma anche mensilmente	Avventore, Tavolata

Tabella 2.4: Dizionario delle interrogazioni - fine.

Capitolo 3

Progettazione logica

Introduzione

In questo capitolo viene affrontata la fase successiva alla progettazione concettuale della base di dati, scendendo ad un livello di astrazione più basso e vicino all'implementazione vera e propria della base di dati. Si tradurrà lo schema concettuale ristrutturato in uno **schema logico**, dipendente dal **modello dei dati** scelto ossia quello **relazionale**. Negli schemi relazionali che seguiranno le **chiavi primarie** sono indicate con una singola sottolineatura mentre le **chiavi esterne** con una doppia sottolineatura.

3.1 Schema logico

3.1.1 Traduzione in schemi relazionali

```
Proprietario(CodProprietario, Username, Password, Nome, Cognome, Email)
```

Chiavi esterne: nessuna.

```
Ristorante(CodRistorante, Denominazione, Indirizzo, Telefono, Citta, Prov, Cap, Email, SitoWeb, Proprietario)
```

Chiavi esterne: Proprietario → Proprietario.CodProprietario

```
ManagerRistorante(CodManager, Username, Password, Nome, Cognome, Email, Telefono, RistoranteGestito)
```

Chiavi esterne: RistoranteGestito → Ristorante.CodRistorante

```
Sala(CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
```

Chiavi esterne: Ristorante → Ristorante.CodRistorante

Cameriere(NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)

Chiavi esterne: Ristorante → Ristorante.CodRistorante

Tavolo(CodTavolo, MaxAvventori, Sala, TavoloAdiacente)

Chiavi esterne: Sala → Sala.CodSala
TavoloAdiacente → Tavolo.CodTavolo

Tavolata(CodTavolata, DataArrivo, OraArrivo, OraUscita, Tavolo, Cameriere)

Chiavi esterne: Tavolo → Tavolo.CodTavolo
Cameriere → Cameriere.NumCid

Avventore(NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)

Chiavi esterne: Ristorante → Ristorante.CodRistorante
Tavolata → Tavolata.CodTavolata

Caso(CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)

Chiavi esterne: AvventorePositivo → Avventore.NumCid
CamerierePositivo → Cameriere.NumCid
RegistraProprietario → Proprietario.CodProprietario
RegistraManager → ManagerRistorante.CodManager

3.1.2 Traduzione delle associazioni

Nella seguente tabella vengono riportate le traduzioni delle associazioni individuate nello schema logico.

<i>Associazione</i>	<i>Implementazione</i>
Amministrazione	Chiave esterna in Ristorante → Proprietario
Gestione	Chiave esterna in ManagerRistorante → Ristorante
Appartenenza	Chiave esterna in Sala → Ristorante
Contenimento	Chiave esterna in Tavolo → Sala
Servizio	Chiave esterna in Tavolata → Cameriere
Partecipazione	Chiave esterna in Avventore → Tavolata
Composizione	Chiave esterna in Tavolata → Tavolo
AdiacenzaTavolo	Chiave esterna in Tavolo → Tavolo
Lavoro	Chiave esterna in Cameriere → Ristorante
Accoglienza	Chiave esterna in Avventore → Ristorante
PositivitaAvventore	Chiave esterna in Caso → Avventore
PositivitaCameriere	Chiave esterna in Caso → Cameriere
TracciamentoProprietario	Chiave esterna in Caso → Proprietario
TracciamentoManager	Chiave esterna in Caso → ManagerRistorante

Tabella 3.1: Traduzione delle associazioni

3.1.3 Schema logico generale

Alla fine del procedimento di traduzione, in conclusione, perveniamo al seguente schema logico:

Proprietario	(<u>CodProprietario</u> , Username, Password, Nome, Cognome, Email)
Ristorante	(<u>CodRistorante</u> , Denominazione, Indirizzo, Telefono, Citta, Prov, Cap, Email, SitoWeb, <u>Proprietario</u>)
ManagerRistorante	(<u>CodManager</u> , Username, Password, Nome, Cognome, Email, Telefono, <u>RistoranteGestito</u>)
Sala	(<u>CodSala</u> , Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, <u>Ristorante</u>)
Cameriere	(<u>NumCid</u> , Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, <u>Ristorante</u>)
Tavolo	(<u>CodTavolo</u> , MaxAvventori, <u>Sala</u> , <u>TavoloAdiacente</u>)
Tavolata	(<u>CodTavolata</u> , DataArrivo, OraArrivo, OraUscita, <u>Tavolo</u> , <u>Cameriere</u>)
Avventore	(<u>NumCid</u> , Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, <u>Ristorante</u> , <u>Tavolata</u>)
Caso	(<u>CodCaso</u> , DataRegistrazione, StatoCaso, Note, <u>AvventorePositivo</u> , <u>CamerierePositivo</u> , <u>RegistraProprietario</u> , <u>RegistraManager</u>)

Tabella 3.2: Schema logico

Capitolo 4

Progettazione fisica

Introduzione

In questo capitolo viene trattata l'ultima fase relativa alla progettazione della base di dati, ossia la progettazione fisica. Dopo aver scelto il **DBMS** su cui implementare la base di dati, si passerà alla traduzione dello schema logico in uno **schema fisico** dei dati, attraverso la definizione delle tabelle, l'implementazione dei relativi vincoli di integrità, dei trigger e delle procedure individuate in **SQL**.

4.1 Note sull'implementazione

Il DBMS scelto per l'implementazione della base di dati è **Oracle Database** nella versione **19C**. Siccome nel DBMS Oracle non è implementato il tipo di dato **boolean**, esso è stato simulato con un carattere (CHAR) che può assumere solo i valori in { 'V' , 'F' } che rappresentano rispettivamente VERO e FALSO.

4.2 Definizione delle tabelle

Seguono le definizioni delle tabelle estratte dallo script di creazione del database tratte dal file "DBRistorante.sql".

4.2.1 Definizione della tabella PROPRIETARIO

```
-- Creazione della tabella PROPRIETARIO
CREATE TABLE PROPRIETARIO
(
    CodProprietario INTEGER GENERATED BY DEFAULT AS IDENTITY,
    Username        VARCHAR2(64) NOT NULL,
    Password        VARCHAR2(64) NOT NULL,
    Nome            VARCHAR2(64) NOT NULL,
    Cognome         VARCHAR2(64) NOT NULL,
    Email           VARCHAR2(320)
);
/

-- Definizione dei vincoli per la tabella PROPRIETARIO
ALTER TABLE PROPRIETARIO ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_PROPRIETARIO PRIMARY KEY (CodProprietario),

    -- Vincolo Email legale
    CONSTRAINT EMAIL_LEGALE_PROPRIETARIO CHECK (Email LIKE '_%@%.____%' OR Email IS NULL),

    -- Vincolo Unico username proprietario
    CONSTRAINT UNICO_USERNAME_PROPRIETARIO UNIQUE (Username)
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella PROPRIETARIO

4.2.2 Definizione della tabella RISTORANTE

```
-- Creazione della tabella RISTORANTE
CREATE TABLE RISTORANTE
(
    CodRistorante    INTEGER GENERATED BY DEFAULT AS IDENTITY,
    Denominazione    VARCHAR2(64) NOT NULL,
    Indirizzo        VARCHAR2(64) NOT NULL,
    Telefono         VARCHAR2(20) NOT NULL,
    Citta            VARCHAR2(64) NOT NULL,
    Prov            VARCHAR2(2)  NOT NULL,
    Cap              VARCHAR2(5) NOT NULL,
    Email            VARCHAR2(320) ,
    SitoWeb          VARCHAR2(100) ,
    Proprietario     INTEGER     NOT NULL
);
/

-- Definizione dei vincoli per la tabella RISTORANTE
ALTER TABLE RISTORANTE ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_RISTORANTE PRIMARY KEY (CodRistorante),

    -- Vincolo di chiave esterna
    /*Non sarà possibile eliminare un proprietario a cui sono associati uno o più ristoranti
    ON DELETE NO ACTION è implementato di default da ORACLE*/
    CONSTRAINT FK_PROPRIETARIO_RISTORANTE FOREIGN KEY (Proprietario) REFERENCES PROPRIETARIO(CodProprietario),

    -- Vincolo Email legale
    CONSTRAINT EMAIL_LEGALE_RISTORANTE CHECK (Email LIKE '_%@%.____%' OR Email IS NULL),

    -- Vincolo Sito Web Legale
    CONSTRAINT SITO_WEB_LEGALE CHECK (SitoWeb LIKE 'www.____%' OR SitoWeb IS NULL)
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella RISTORANTE

4.2.3 Definizione della tabella MANAGERRISTORANTE

```
-- Creazione della tabella MANAGERRISTORANTE
CREATE TABLE MANAGERRISTORANTE
(
    CodManager          INTEGER GENERATED BY DEFAULT AS IDENTITY,
    Username            VARCHAR2(64) NOT NULL,
    Password            VARCHAR2(64) NOT NULL,
    Nome                VARCHAR2(64) NOT NULL,
    Cognome             VARCHAR2(64) NOT NULL,
    Email               VARCHAR2(320) ,
    Telefono            VARCHAR2(20) NOT NULL,
    RistoranteGestito   INTEGER
);
/

-- Definizione dei vincoli per la tabella MANAGERRISTORANTE
ALTER TABLE MANAGERRISTORANTE ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_MANAGER_RISTORANTE PRIMARY KEY (CodManager),

    -- Vincolo di chiave esterna
    -- Se si cancella un ristorante vengono eliminati anche i manager che lo gestiscono
    CONSTRAINT FK_MANAGER_RISTORANTE FOREIGN KEY (RistoranteGestito) REFERENCES RISTORANTE(CodRistorante) ON DELETE CASCADE,

    -- Vincolo Email legale
    CONSTRAINT EMAIL_LEGALE_MANAGER_RISTORANTE CHECK (Email LIKE '%@%.%' OR Email IS NULL),

    -- Vincolo Unico username manager
    CONSTRAINT UNICO_USERNAME_MANAGER_RISTORANTE UNIQUE (Username)
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella MANAGERRISTORANTE

4.2.4 Definizione della tabella SALA

```
-- Creazione della tabella SALA
CREATE TABLE SALA
(
    CodSala             INTEGER GENERATED BY DEFAULT AS IDENTITY,
    Denominazione       VARCHAR2(64) NOT NULL,
    CapienzaAvventori   INTEGER NOT NULL,
    DimensioneMq        INTEGER ,
    TipoSala            VARCHAR(10) NOT NULL,
    Ristorante          INTEGER NOT NULL
);
/

-- Definizione dei vincoli per la tabella SALA
ALTER TABLE SALA ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_SALA PRIMARY KEY (CodSala),

    -- Vincolo di chiave esterna
    -- Se si cancella un ristorante vengono eliminate anche le sale che vi appartengono
    CONSTRAINT FK_SALA_RISTORANTE FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(CodRistorante) ON DELETE CASCADE,

    -- Vincolo CapienzaAvventori legale
    CONSTRAINT CAPIENZA_AVVENTORI_LEGALE CHECK (CapienzaAvventori > 0),

    -- Vincolo DimensioneMq legale
    CONSTRAINT DIMENSIONEMQ_LEGALE CHECK (DimensioneMq > 0 OR DimensioneMq IS NULL),

    -- Vincolo TSala
    CONSTRAINT TSALA CHECK (TipoSala IN ('Interna', 'Esterna'))
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella SALA

4.2.5 Definizione della tabella CAMERIERE

```
-- Creazione della tabella CAMERIERE
CREATE TABLE CAMERIERE
(
    NumCid          VARCHAR2(9)      NOT NULL,
    Nome            VARCHAR2(64)     NOT NULL,
    Cognome         VARCHAR2(64)     NOT NULL,
    DataN          DATE              NOT NULL,
    Sesso          VARCHAR2(20)      NOT NULL,
    CittaN         VARCHAR2(64)     NOT NULL,
    ProvN          VARCHAR2(2)       NOT NULL,
    CittaRes       VARCHAR2(64)     NOT NULL,
    ProvRes        VARCHAR2(2)       NOT NULL,
    Telefono       VARCHAR2(20)     NOT NULL,
    Email          VARCHAR2(320)
    Ristorante     INTEGER          NOT NULL
);
/
-- Definizione dei vincoli per la tabella CAMERIERE
ALTER TABLE CAMERIERE ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_CAMERIERE PRIMARY KEY (NumCid),

    -- Vincolo di chiave esterna
    -- Se si cancella un ristorante vengono eliminate anche i camerieri che vi lavorano
    CONSTRAINT FK_CAMERIERE_RISTORANTE FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(CodRistorante) ON DELETE CASCADE,

    -- Vincolo Email legale
    CONSTRAINT EMAIL_LEGALE_CAMERIERE CHECK (Email LIKE '%@%.%' OR Email IS NULL),

    -- Vincolo TSesso
    CONSTRAINT TSesso CHECK (Sesso IN ('Maschio','Femmina','Non specificato'))
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella CAMERIERE

4.2.6 Definizione della tabella TAVOLO

```
-- Creazione della tabella Tavolo
CREATE TABLE TAVOLO
(
    CodTavolo       INTEGER GENERATED BY DEFAULT AS IDENTITY,
    MaxAvventori    INTEGER          NOT NULL,
    Sala            INTEGER          NOT NULL,
    TavoloAdiacente INTEGER
);
/
-- Definizione dei vincoli per la tabella TAVOLO
ALTER TABLE TAVOLO ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_TAVOLO PRIMARY KEY (CodTavolo),

    -- Vincolo di chiave esterna
    -- Se si cancella una sala vengono eliminati anche i tavoli in essa contenuti
    CONSTRAINT FK_SALA_TAVOLO FOREIGN KEY (Sala) REFERENCES SALA(CodSala) ON DELETE CASCADE,

    -- Vincolo di chiave esterna
    -- Se si cancella un tavolo adiacente ad un altro, l'adiacenza diventa NULL
    CONSTRAINT FK_TAVOLO_ADIACENTE FOREIGN KEY (TavoloAdiacente) REFERENCES TAVOLO(CodTavolo) ON DELETE SET NULL
);
/
```

Estratto da DBRistorante.sql - Definizione della tabella TAVOLO

4.2.7 Definizione della tabella TAVOLATA

```
-- Creazione della tabella TAVOLATA
CREATE TABLE TAVOLATA
(
    CodTavolata      INTEGER GENERATED BY DEFAULT AS IDENTITY,
    DataArrivo       DATE                DEFAULT SYSDATE NOT NULL,
    OraArrivo        INTERVAL DAY(0) TO SECOND(0)  DEFAULT INTERVAL '20:00' HOUR TO MINUTE NOT NULL,
    OraUsicta        INTERVAL DAY(0) TO SECOND(0)  DEFAULT INTERVAL '22:00' HOUR TO MINUTE NOT NULL,
    Tavolo           INTEGER              NOT NULL,
    Cameriere        VARCHAR2(9)          NOT NULL
);
/
-- Definizione dei vincoli per la tabella TAVOLATA
ALTER TABLE TAVOLATA ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_TAVOLATA PRIMARY KEY (CodTavolata),

    -- Vincolo di chiave esterna
    -- Se si elimina un tavolo viene eliminata anche la tavolata ad esso associato
    CONSTRAINT FK_TAVOLO_TAVOLATA FOREIGN KEY (Tavolo) REFERENCES TAVOLO(CodTavolo) ON DELETE CASCADE,

    -- Vincolo di chiave esterna
    CONSTRAINT FK_CAMERIERE_TAVOLATA FOREIGN KEY (Cameriere) REFERENCES CAMERIERE(NumCid),

    -- Vincolo Unica composizione tavolo a tavolata
    CONSTRAINT UNICA_COMPOSIZIONE_TAVOLO_A_TAVOLATA UNIQUE(DataArrivo,Tavolo)
);
/
```

Estratto da **DBRistorante.sql** - Definizione della tabella **TAVOLATA**

4.2.8 Definizione della tabella AVVENTORE

```
-- Creazione della tabella AVVENTORE
CREATE TABLE AVVENTORE
(
    NumCid          VARCHAR2(9)      NOT NULL,
    Nome            VARCHAR2(64)     NOT NULL,
    Cognome         VARCHAR2(64)     NOT NULL,
    DataN          DATE              NOT NULL,
    Sesso           VARCHAR2(15)     NOT NULL,
    CittaN         VARCHAR2(64)     NOT NULL,
    ProvN          VARCHAR2(2)       NOT NULL,
    CittaRes       VARCHAR2(64)     NOT NULL,
    ProvRes        VARCHAR2(2)       NOT NULL,
    Telefono       VARCHAR2(20)     NOT NULL,
    Email          VARCHAR2(320)     ,
    Temperatura    DECIMAL(3,1)     NOT NULL,
    HaGreenpass    CHAR              DEFAULT 'F' NOT NULL,
    Ristorante     INTEGER           NOT NULL,
    Tavolata       INTEGER           NOT NULL
);

-- Definizione dei vincoli per la tabella AVVENTORE
ALTER TABLE AVVENTORE ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_AVVENTORE PRIMARY KEY (NumCid),

    -- Vincolo di chiave esterna
    -- Se si elimina un ristorante vengono eliminati anche gli avventori da esso accolti
    CONSTRAINT FK_AVVENTORE_RISTORANTE FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(CodRistorante) ON DELETE CASCADE,

    -- Vincolo di chiave esterna
    -- Se si elimina una tavolata vengono eliminati anche gli avventori che la formano
    CONSTRAINT FK_AVVENTORE_TAVOLATA FOREIGN KEY (Tavolata) REFERENCES TAVOLATA(CodTavolata) ON DELETE CASCADE,

    -- Vincolo Email legale
    CONSTRAINT EMAIL_LEGALE_AVVENTORE CHECK (Email LIKE '%@%.%' OR Email IS NULL),

    -- Vincolo TSesso
    CONSTRAINT TSesso_AVVENTORE CHECK (Sesso IN ('Maschio','Femmina','Non specificato')),

    -- Vincolo per il dominio dell'attributo booleano HaGreenPass
    CONSTRAINT HAGREENPASS_VALUES CHECK (HaGreenPass IN('V','F')),

    -- Vincolo Temperatura avventore legale
    CONSTRAINT TEMPERATURA_AVVENTORE_LEGALE CHECK (Temperatura BETWEEN 35.0 AND 37.5)
);
```

Estratto da **DBRistorante.sql** - Definizione della tabella AVVENTORE

4.2.9 Definizione della tabella CASO

```
-- Creazione della tabella CASO
CREATE TABLE CASO
(
    CodCaso INTEGER GENERATED BY DEFAULT AS IDENTITY,
    DataRegistrazione DATE NOT NULL,
    StatoCaso VARCHAR2(20) DEFAULT 'NonRisolto' NOT NULL,
    Note VARCHAR2(100) ,
    AvventorePositivo VARCHAR2(9) ,
    CamerierePositivo VARCHAR2(9) ,
    RegistraProprietario INTEGER ,
    RegistraManager INTEGER
);
/
-- Definizione dei vincoli per la tabella CASO
ALTER TABLE CASO ADD
(
    -- Vincolo di chiave primaria
    CONSTRAINT PK_CASO PRIMARY KEY (CodCaso),

    -- Vincolo di chiave esterna
    CONSTRAINT FK_AVVENTORE_POSITIVO FOREIGN KEY (AvventorePositivo) REFERENCES AVVENTORE(NumCid) ON DELETE SET NULL,

    -- Vincolo di chiave esterna
    CONSTRAINT FK_CAMERIERE_POSITIVO FOREIGN KEY (CamerierePositivo) REFERENCES CAMERIERE(NumCid) ON DELETE SET NULL,

    -- Vincolo di chiave esterna
    CONSTRAINT FK_REGISTRA_PROPRIETARIO FOREIGN KEY (RegistraProprietario) REFERENCES PROPRIETARIO(CodProprietario) ON DELETE SET NULL,

    -- Vincolo di chiave esterna
    CONSTRAINT FK_REGISTRA_MANAGER FOREIGN KEY (RegistraManager) REFERENCES MANAGERRISTORANTE(CodManager) ON DELETE SET NULL,

    -- Vincolo TCaso
    CONSTRAINT TCaso CHECK (StatoCaso IN ('Risolto','InRisoluzione','NonRisolto'))
);
/
```

Estratto da **DBRistorante.sql** - Definizione della tabella **CASO**

- fine definizione delle tabelle -

4.3 Funzioni, procedure ed altre automazioni

Seguono le definizioni delle procedure e funzioni utilizzate nel database tratte dal file “DBProcedureFunzioni.sql”.

4.3.1 Stored function IS_NUMBER

Lo scopo della stored function IS_NUMBER è quello di verificare che la stringa data in input contenga solo caratteri numerici. Verrà utilizzata successivamente nella definizione di alcuni trigger.

```
CREATE FUNCTION IS_NUMBER (stringa IN VARCHAR2) RETURN INT
IS
    valorenumerico NUMBER;
BEGIN
    -- Converte da valore numerico a stringa e ritorna 1
    valorenumerico := TO_NUMBER(stringa);
    RETURN 1;

    -- Se fallisce la conversione, errore e ritorna 0
    EXCEPTION WHEN VALUE_ERROR THEN
        RETURN 0;
END;
/
```

Estratto da DBProcedureFunzioni.sql - Definizione della stored function IS_NUMBER

4.3.2 Stored procedure NUMERO_DI_TELEFONO_LEGALE

Lo scopo della stored procedure NUMERO_DI_TELEFONO_LEGALE è quello di verificare che una stringa data in input rispetti il formato standard di un numero telefonico. Il controllo esclude il numero di cifre del numero telefonico, che possono essere variabili. Verrà utilizzata successivamente nella definizione di alcuni trigger.

```
CREATE PROCEDURE NUMERO_DI_TELEFONO_LEGALE (numerotelefonico IN VARCHAR2)
IS
    stringa VARCHAR2(15);
BEGIN
    -- Rimuove inizialmente tutti gli spazi, se presenti, dalla stringa del numero telefonico
    -- per effettuare il successivo controllo
    stringa := REPLACE(numerotelefonico, ' ', '');

    -- Se presente un prefisso nel numero telefonico con il carattere +, rimuove quest'ultimo
    -- per effettuare il successivo controllo
    IF stringa LIKE '+%' THEN
        stringa := TRIM('+ ' FROM stringa);
    END IF;

    -- Controlla che i restanti caratteri associati al numero telefonico siano numeri.
    -- Se il controllo fallisce, è presente un carattere diverso da un numero
    -- quindi il numero di telefono non è valido.
    IF (is_number(stringa)=0) THEN
        RAISE_APPLICATION_ERROR(-20010, 'Numero di telefono non valido. ');
    END IF;
END;
/
```

Estratto da DBProcedureFunzioni.sql - Definizione della stored procedure NUMERO_DI_TELEFONO_LEGALE

4.3.3 Stored procedure PASSWORD_LEGALE

Lo scopo della stored procedure PASSWORD_LEGALE è quello di verificare che una stringa data in input rispetti il formato stabilito per una password. Verrà utilizzata successivamente nella definizione di alcuni trigger.

```
-- 3. Procedure Password legale (viene riutilizzata più volte nei trigger)
CREATE PROCEDURE PASSWORD_LEGALE (password IN VARCHAR2)
IS
    password_okay INTEGER;
BEGIN
    -- Controlla se la password contiene almeno una lettera ed un numero
    IF REGEXP_LIKE(password, '^[A-Z].*$') AND REGEXP_LIKE(password, '^[0-9].*$') THEN
        password_okay := 1;
    ELSE
        password_okay := 0; -- La password non contiene almeno una lettera ed un numero
    END IF;

    -- Se la password inserita è lunga meno di 8 caratteri o
    -- non contiene almeno una lettera ed un numero allora non è valida
    IF (LENGTH(password) < 8) OR (password_okay = 0) THEN
        RAISE_APPLICATION_ERROR(-20011, 'Password non valida.
        Deve contenere almeno 8 caratteri, una lettera ed un numero!');
    END IF;
END;
/
```

Estratto da **DBProcedureFunzioni.sql** - Definizione della stored procedure **PASSWORD_LEGALE**

- fine definizione delle procedure e funzioni -

4.4 Viste

Segue la definizione dell'unica vista utilizzata nel database tratta dal file "DBRistorante.sql".

4.4.1 Vista TAVOLATE

Lo scopo della vista TAVOLATE di calcolare il numero totale di partecipanti alla tavolata, per ogni tavolata presente in ciascun ristorante. Verrà utilizzata successivamente per l'implementazione di una query.

```
-- Creazione della VISTA 'Tavolate' :  
-- per ogni tavolata di un ristorante calcola il numero totale di partecipanti alla tavolata.  
CREATE VIEW TAVOLATE (Ristorante, CodiceTavolata, Partecipanti) AS  
SELECT A.Ristorante, A.Tavolata, COUNT(A.NumCid) AS TOT_TAVOLATA  
FROM AVVENTORE A JOIN TAVOLATA T ON A.Tavolata = T.CodTavolata  
GROUP BY A.Ristorante, A.Tavolata  
ORDER BY A.Tavolata ASC;  
/
```

Estratto da **DBRistorante.sql** - Definizione della vista **TAVOLATE**

- fine definizione delle viste -

4.5 Implementazione dei vincoli

Seguono le implementazioni dei vari vincoli che non sono già stati mostrati nella definizione delle tabelle. Tutti i vincoli sono tratti dal file “DBRistorante.sql”.

4.5.1 Implementazione del vincolo Password legale

Il vincolo si ripete nella definizione di diverse tabelle. Per questo scopo è stata definita e riutilizzata più volte la stored procedure PASSWORD_LEGALE.

```
-- Trigger per il vincolo Password legale
CREATE OR REPLACE TRIGGER PASSWORD_PROPRIETARIO_LEGALE
BEFORE INSERT OR UPDATE ON PROPRIETARIO
FOR EACH ROW
BEGIN
    PASSWORD_LEGALE (:NEW.Password) ;
END;
/

-- Trigger per il vincolo Password legale
CREATE OR REPLACE TRIGGER PASSWORD_MANAGER_RISTORANTE_LEGALE
BEFORE INSERT OR UPDATE ON MANAGERRISTORANTE
FOR EACH ROW
BEGIN
    PASSWORD_LEGALE (:NEW.Password) ;
END;
/
```

Estratto da DBRistorante.sql - Implementazione del vincolo **Password legale**

4.5.2 Implementazione del vincolo Numero di telefono legale

Il vincolo si ripete nella definizione di diverse tabelle. Per questo scopo è stata definita e riutilizzata più volte la stored procedure NUMERO_DI_TELEFONO_LEGALE.

```
-- Trigger per il vincolo Numero di telefono legale
CREATE OR REPLACE TRIGGER NUMERO_DI_TELEFONO_RISTORANTE_LEGALE
BEFORE INSERT OR UPDATE ON RISTORANTE
FOR EACH ROW
BEGIN
    NUMERO_DI_TELEFONO_LEGALE (:NEW.Telefono) ;
END;
/

-- Trigger per il vincolo Numero di telefono legale
CREATE OR REPLACE TRIGGER NUMERO_DI_TELEFONO_MANAGER_RISTORANTE_LEGALE
BEFORE INSERT OR UPDATE ON MANAGERRISTORANTE
FOR EACH ROW
BEGIN
    NUMERO_DI_TELEFONO_LEGALE (:NEW.Telefono) ;
END;
/

-- Trigger per il vincolo Numero di telefono legale
CREATE OR REPLACE TRIGGER NUMERO_DI_TELEFONO_CAMERIERE_LEGALE
BEFORE INSERT OR UPDATE ON CAMERIERE
FOR EACH ROW
BEGIN
    NUMERO_DI_TELEFONO_LEGALE (:NEW.Telefono) ;
END;
/

-- Trigger per il vincolo Numero di telefono legale
CREATE OR REPLACE TRIGGER NUMERO_DI_TELEFONO_AVVENTORE_LEGALE
BEFORE INSERT OR UPDATE ON AVVENTORE
FOR EACH ROW
BEGIN
    NUMERO_DI_TELEFONO_LEGALE (:NEW.Telefono) ;
END;
/
```

Estratto da DBRistorante.sql - Implementazione del vincolo Numero di telefono legale

4.5.3 Implementazione del vincolo Cap legale

```
-- Trigger per il vincolo Cap legale
CREATE OR REPLACE TRIGGER CAP_LEGALE
BEFORE INSERT OR UPDATE ON RISTORANTE
FOR EACH ROW
BEGIN
    -- Controlla che i caratteri associati al CAP inserito siano 5 e tutti numeri.
    -- Se il controllo fallisce, quindi è presente un carattere diverso da un numero oppure
    -- ci sono meno di 5 caratteri allora il CAP non è valido.
    IF (is_number(:NEW.Cap)=0) OR (LENGTH(:NEW.Cap)<5) THEN
        RAISE_APPLICATION_ERROR(-20012,'CAP inserito non valido.');
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Cap legale**

4.5.4 Implementazione del vincolo Età cameriere legale

```
-- Trigger per il vincolo Età cameriere legale
CREATE OR REPLACE TRIGGER ETA_CAMERIERE_LEGALE
BEFORE INSERT OR UPDATE ON CAMERIERE
FOR EACH ROW
DECLARE etacameriere INTEGER;
BEGIN
    -- Se l'età del cameriere è inferiore a 18 anni,
    -- allora non sarà possibile inserire il cameriere
    etacameriere := TRUNC((TO_NUMBER(SYSDATE - :NEW.DataN))/365.25);

    IF etacameriere < 18 THEN
        RAISE_APPLICATION_ERROR(-20014,'DataN per cameriere non valida. Un cameriere deve essere maggiorenne!');
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Età cameriere legale**

4.5.5 Implementazione del vincolo MaxAvventori legale

```
-- Trigger per il vincolo MaxAvventori legale
CREATE OR REPLACE TRIGGER MAXAVVENTORI_LEGALE
AFTER INSERT OR UPDATE ON TAVOLO
FOR EACH ROW
DECLARE
    capienza INTEGER;
BEGIN
    -- Calcolo di CapienzaAvventori
    SELECT S.CapienzaAvventori INTO capienza
    FROM SALA S
    WHERE S.CodSala = :NEW.Sala;

    -- Se MaxAvventori è <=0 allora non è valido
    IF :NEW.MaxAvventori <=0 THEN
        RAISE_APPLICATION_ERROR(-20015,'Il valore per MaxAvventori deve essere maggiore di 0!');
    ELSE
        -- Se MaxAvventori è > di CapienzaAvventori della sala in cui il tavolo è contenuto
        -- allora non è valido
        IF :NEW.MaxAvventori > capienza THEN
            RAISE_APPLICATION_ERROR(-20016,'Il valore di MaxAvventori per il tavolo deve essere
            minore o uguale alla CapienzaAvventori della sala che contiene il tavolo in questione!');
        END IF;
    END IF;
END;
/
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **MaxAvventori legale**

4.5.6 Implementazione del vincolo Capienza legale

```
-- Trigger per il vincolo Capienza legale
CREATE OR REPLACE TRIGGER CAPIENZA_LEGALE
BEFORE INSERT ON TAVOLO
FOR EACH ROW
DECLARE
    capienza INTEGER;
    capienzacorrente INTEGER;
BEGIN
    -- Calcolo di CapienzaAvventori
    SELECT S.CapienzaAvventori INTO capienza
    FROM SALA S
    WHERE S.CodSala = :NEW.Sala;

    -- Calcolo della CapienzaAvventori corrente per la Sala
    SELECT SUM(T.MAXAVVENTORI) INTO capienzacorrente
    FROM TAVOLO T
    WHERE T.Sala =:NEW.Sala;

    -- Se la somma fra la Capienza corrente e il nuovo MaxAvventori supera
    -- la Capienza massima della Sala allora blocca l'inserimento
    IF :NEW.MAXAVVENTORI+capienzacorrente > capienza THEN
        RAISE_APPLICATION_ERROR(-20017,'Errore. Capienza massima della sala di riferimento superata!');
    END IF;
END;
/
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Capienza legale**

4.5.7 Implementazione del vincolo Data nascita legale

```
-- Trigger per il vincolo Data nascita legale
CREATE OR REPLACE TRIGGER DATA_NASCITA_LEGALE
BEFORE INSERT OR UPDATE ON AVVENTORE
FOR EACH ROW
DECLARE
datatavolata TAVOLATA.DataArrivo%TYPE;
BEGIN
    -- Recupera la data della tavolata a cui partecipa l'avventore
    SELECT T.DataArrivo INTO datatavolata
    FROM TAVOLATA T
    WHERE T.CodTavolata = :NEW.Tavolata;

    -- Se la data di nascita dell'avventore è successiva alla data della tavolata allora
    -- blocca l'inserimento dell'avventore
    IF :NEW.DataN > datatavolata THEN
        RAISE_APPLICATION_ERROR( -20018, 'La data di nascita di un avventore deve essere
        precedente alla data di arrivo della tavolata!');
    END IF;
END;
/
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Data nascita legale**

4.5.8 Implementazione del vincolo Has greenpass

```
-- Trigger per il vincolo Has greenpass
CREATE OR REPLACE TRIGGER HAS_GREENPASS
BEFORE INSERT OR UPDATE ON AVVENTORE
FOR EACH ROW
DECLARE
tipologiasala SALA.TipoSala%TYPE;
BEGIN
    -- Recupera la tipologia della sala in cui è ubicato il tavolo della tavolata a cui partecipa l'avventore
    SELECT S.TipoSala INTO tipologiasala
    FROM TAVOLATA T JOIN TAVOLO TA ON T.Tavolo = TA.CodTavolo JOIN SALA S ON S.CodSala = TA.Sala
    WHERE T.CodTavolata = :NEW.Tavolata;

    -- Se l'avventore è sprovvisto di green pass ed è stato associato ad una tavolata il cui tavolo
    -- si trova in una sala interna allora non può partecipare alla tavolata.
    IF :NEW.HasGreenpass='F' AND tipologiasala='Interna' THEN
        RAISE_APPLICATION_ERROR( -20019, 'Un avventore sprovvisto di green pass puo partecipare unicamente
        ad una tavolata composta da un tavolo ubicato in una sala esterna!');
    END IF;
END;
/
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Has greenpass**

4.5.9 Implementazione del vincolo Temperatura avventore legale

```
-- Trigger per il vincolo Temperatura avventore legale
CREATE OR REPLACE TRIGGER TEMPERATURA_AVVENTORE
BEFORE INSERT OR UPDATE ON AVVENTORE
FOR EACH ROW
BEGIN
    -- Se la temperatura dell'avventore supera i 37.5 gradi allora comunica che l'avventore
    -- puo' essere un potenziale caso da registrare nella tabella CASO.
    IF :NEW.Temperatura > 37.5 THEN
        RAISE_APPLICATION_ERROR( -20020, 'Temperatura avventore illegale!
        Potrebbe essere un potenziale CASO da registrare.');
```

Estratto da DBRistorante.sql - Implementazione del vincolo Temperatura avventore legale

4.5.10 Implementazione del vincolo Somma avventori a tavolata legale

```
-- Trigger per il vincolo Somma avventori a tavolata legale
CREATE OR REPLACE TRIGGER SOMMA_AVVENTORI_A_TAVOLATA_LEGALE
BEFORE INSERT ON AVVENTORE
FOR EACH ROW
DECLARE
    maxavventoritavolata TAVOLO.MaxAvventori%TYPE;
    numavventoricorrente INTEGER;
    avventorecorrente INTEGER;
BEGIN
    -- Recupera il massimo numero dei posti del tavolo associato alla tavolata
    SELECT TA.MaxAvventori INTO maxavventoritavolata
    FROM TAVOLATA T JOIN TAVOLO TA ON T.Tavolo = TA.CodTavolo
    WHERE T.CodTavolata = :NEW.Tavolata;

    -- Conta gli avventori correnti che partecipano alla tavolata
    SELECT COUNT(A.NumCid) INTO numavventoricorrente
    FROM AVVENTORE A
    WHERE A.Tavolata = :NEW.Tavolata;

    -- Se il numero di avventori correnti che partecipano alla tavolata
    -- più il nuovo avventore che si sta per registrare
    -- superano complessivamente il numero dei posti del tavolo
    -- allora non è possibile registrare il nuovo avventore
    IF numavventoricorrente+1 > maxavventoritavolata THEN
        RAISE_APPLICATION_ERROR( -20021, 'Impossibile registrare avventore alla tavolata.
        Il tavolo di riferimento ha tutti i posti occupati!');
```

Estratto da DBRistorante.sql - Implementazione del vincolo Somma avventori a tavolata legale

4.5.11 Implementazione del vincolo Data registrazione caso

```
-- Trigger per il vincolo Data registrazione caso
CREATE OR REPLACE TRIGGER DATA_REGISTRAZIONE_CASO
BEFORE INSERT OR UPDATE ON CASO
FOR EACH ROW
DECLARE
datanascita_avventore AVVENTORE.DataN%TYPE;
datanascita_cameriere CAMERIERE.DataN%TYPE;
BEGIN
    IF :NEW.AvventorePositivo IS NOT NULL THEN
        -- Recupera la data di nascita dell'avventore positivo
        SELECT A.DataN INTO datanascita_avventore
        FROM AVVENTORE A
        WHERE A.NumCid = :NEW.AvventorePositivo;

        -- Se la data di registrazione del caso precede la data di nascita
        -- allora non e' possibile registrare il caso
        IF :NEW.DataRegistrazione < datanascita_avventore THEN
            RAISE_APPLICATION_ERROR( -20022, 'Impossibile registrare caso: data registrazione non valida.' );
        END IF;
    END IF;

    IF :NEW.CamerierePositivo IS NOT NULL THEN
        -- Recupera la data di nascita del cameriere positivo
        SELECT C.DataN INTO datanascita_cameriere
        FROM CAMERIERE C
        WHERE C.NumCid = :NEW.CamerierePositivo;

        -- Se la data di registrazione del caso precede la data di nascita
        -- allora non e' possibile registrare il caso
        IF :NEW.DataRegistrazione < datanascita_cameriere THEN
            RAISE_APPLICATION_ERROR( -20022, 'Impossibile registrare caso: data registrazione non valida.' );
        END IF;
    END IF;
END;
/
```

Estratto da **DBRistorante.sql** - Implementazione del vincolo **Data registrazione caso**

- fine definizione dei vincoli -

Capitolo 5

Esempio d'uso

Introduzione

In questo capitolo viene mostrato un esempio d'uso del database implementato. Si inizierà mostrando lo script definito per il popolamento del db contenuto nel file “DBInsert.sql”. Successivamente viene mostrata l'implementazione delle possibili query da effettuare sul database, tratte dal file “DBQuery.sql” e definite precedentemente nel dizionario delle interrogazioni.

5.1 Popolamento del database

```
-- POPOLAMENTO TABELLE DEL DB
/*=====*/
/*=====*/
-- Insert per la tabella PROPRIETARIO: aggiunge il proprietario dei ristoranti.
INSERT INTO PROPRIETARIO (CodProprietario, Username, Password, Nome, Cognome, Email)
VALUES (1,'sandro05','password00','Santolo','Barretta','santolobarretta05@gmail.com');
COMMIT;
/*=====*/
/*=====*/
-- Insert per la tabella RISTORANTE: aggiunge i ristoranti amministrati da un proprietario.
INSERT INTO RISTORANTE (CodRistorante, Denominazione, Indirizzo, Telefono, Città, Prov, Cap, Email, SitoWeb, Proprietario)
VALUES (1,'Bella Napoli','Via Francesco Caracciolo, 1','0813509900','Napoli','NA','80122','bellanapoli@gmail.com','www.ristorantebellanapoli.it',1);

INSERT INTO RISTORANTE (CodRistorante, Denominazione, Indirizzo, Telefono, Città, Prov, Cap, Email, SitoWeb, Proprietario)
VALUES (2,'Bella Napoli 2','Via Toledo, 30','0813339900','Napoli','NA','80134','bellanapoli@gmail.com','www.ristorantebellanapoli.it',1);

INSERT INTO RISTORANTE (CodRistorante, Denominazione, Indirizzo, Telefono, Città, Prov, Cap, Proprietario)
VALUES (3,'La Madonnina','Via Alessandro Manzoni, 5','+390289091122','Milano','MI','20121',1);
COMMIT;
/*=====*/
/*=====*/
-- Insert per la tabella MANAGERRISTORANTE: aggiunge i manager che gestiscono un ristorante, se previsti.
INSERT INTO MANAGERRISTORANTE (CodManager, Username, Password, Nome, Cognome, Email, Telefono, RistoranteGestito)
VALUES (1,'angelo02','qwerty0123','Angelo','Di Maio','angelodimaio@gmail.com','+393337060999',2);

INSERT INTO MANAGERRISTORANTE (CodManager, Username, Password, Nome, Cognome, Telefono, RistoranteGestito)
VALUES (2,'mariorossi33','mrl123456','Mario','Rossi','+393517799111',3);
COMMIT;
/*=====*/
/*=====*/

-- Insert per la tabella SALA: aggiunge le sale ai relativi ristoranti cui appartengono.
INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (1,'Sala Pulcinella',50,100,'Interna',1);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (2,'Sala Vesuvio',20,60,'Interna',1);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (3,'Sala Garden',10,30,'Esterna',1);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (4,'Sala Maradona',50,100,'Interna',2);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (5,'Sala Partenope',18,25,'Interna',2);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (6,'Sala Duomo',30,80,'Interna',3);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (7,'Sala Meneghino',30,80,'Interna',3);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (8,'Sala Biscione',20,60,'Interna',3);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (9,'Sala Exclusive',10,20,'Esterna',3);

INSERT INTO SALA (CodSala, Denominazione, CapienzaAvventori, DimensioneMq, TipoSala, Ristorante)
VALUES (10,'Sala Vip',10,NULL,'Esterna',3);
COMMIT;
/*=====*/
/*=====*/
```

```

-- Insert per la tabella CAMERIERE: aggiunge i camerieri ai rispettivi ristoranti per cui lavorano.
INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA78432DB', 'Ciro', 'Esposito', TO_DATE('08/01/2000', 'dd/mm/yyyy'), 'Maschio', 'Napoli', 'NA', 'Qualiano', 'NA', '3517486042', 'ciroespo@gmail.com', 1);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA66421DA', 'Andrea', 'Russo', TO_DATE('10/10/1990', 'dd/mm/yyyy'), 'Maschio', 'Villaricca', 'NA', 'Quarto', 'NA', '3591676343', 'andrearusso@outlook.it', 1);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('AU0199811', 'Maria', 'De Rosa', TO_DATE('22/03/1989', 'dd/mm/yyyy'), 'Femmina', 'Pozzuoli', 'NA', 'Napoli', 'NA', '3722282194', 'mariaderosa@gmail.com', 1);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA64321CE', 'Rosaria', 'Romano', TO_DATE('30/08/1993', 'dd/mm/yyyy'), 'Femmina', 'Napoli', 'NA', 'Napoli', 'NA', '3218505681', 'romano01@gmail.com', 1);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('AU3425442', 'Pasquale', 'Ricci', TO_DATE('12/04/1988', 'dd/mm/yyyy'), 'Maschio', 'Caserta', 'CE', 'Casoria', 'NA', '3273256337', 'ricciopas22@outlook.it', 1);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA53216BE', 'Gabriele', 'Marino', TO_DATE('21/02/1993', 'dd/mm/yyyy'), 'Maschio', 'Napoli', 'NA', 'Pozzuoli', 'NA', '3553493696', 'marino0@gmail.com', 2);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA73121FE', 'Salvatore', 'Neri', TO_DATE('12/03/1996', 'dd/mm/yyyy'), 'Maschio', 'Napoli', 'NA', 'Bacoli', 'NA', '3215642156', 'sasinerill@gmail.com', 2);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('AU567231', 'Luisa', 'Esposito', TO_DATE('08/01/2000', 'dd/mm/yyyy'), 'Femmina', 'Napoli', 'NA', 'Quarto', 'NA', '3445768683', 'espolui0@outlook.com', 2);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA32131PV', 'Carmen', 'Granata', TO_DATE('28/05/1995', 'dd/mm/yyyy'), 'Femmina', 'Mariglianese', 'CE', 'Pozzuoli', 'NA', '3830046533', 'carmen233@gmail.com', 2);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA97452TB', 'Laura', 'Galli', TO_DATE('17/11/1999', 'dd/mm/yyyy'), 'Femmina', 'Milano', 'MI', 'Legnano', 'MI', '3746409289', 'lauragalli56@gmail.com', 3);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA33145OD', 'Sara', 'Sala', TO_DATE('13/12/2000', 'dd/mm/yyyy'), 'Femmina', 'Milano', 'MI', 'Sesto San Giovanni', 'MI', '3227317200', 'sara123@gmail.com', 3);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('AU0236754', 'Fiorella', 'Bianchi', TO_DATE('25/10/1990', 'dd/mm/yyyy'), 'Femmina', 'Milano', 'MI', 'Rho', 'MI', '3229731501', 'bianchi0@outlook.com', 3);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA10946DE', 'Raffaele', 'Fontana', TO_DATE('04/09/1994', 'dd/mm/yyyy'), 'Maschio', 'Milano', 'MI', 'Legnano', 'MI', '3619439924', 'raffaele444@gmail.com', 3);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA32410A', 'Achille', 'Rinaldi', TO_DATE('21/05/1987', 'dd/mm/yyyy'), 'Maschio', 'Milano', 'MI', 'Milano', 'MI', '3283664283', 'achillerinaldi2@gmail.com', 3);

INSERT INTO CAMERIERE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Ristorante)
VALUES ('CA13264UI', 'Morena', 'Costa', TO_DATE('19/03/1999', 'dd/mm/yyyy'), 'Femmina', 'Milano', 'MI', 'Assago', 'MI', '3194130484', 'morena43@outlook.com', 3);
COMMIT;
/*=====*/
/*=====*/

-- Insert per la tabella TAVOLO
INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (1, 10, 1, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (2, 5, 1, 1);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (3, 5, 1, 2);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (4, 2, 1, 3);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (5, 2, 1, 4);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (6, 2, 1, 5);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (7, 2, 1, 6);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (8, 2, 1, 7);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (9, 20, 1, 8);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (10, 5, 2, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (11, 5, 2, 10);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (12, 2, 2, 11);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (13, 2, 2, 12);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (14, 2, 2, 13);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (15, 2, 2, 14);

```

```

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (16, 2, 2, 15);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (17, 2, 3, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (18, 2, 3, 17);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (19, 2, 3, 18);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (20, 2, 3, 19);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (21, 2, 3, 20);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (22, 10, 4, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (23, 5, 4, 22);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (24, 5, 4, 23);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (25, 2, 4, 24);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (26, 2, 4, 25);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (27, 2, 4, 26);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (28, 2, 4, 27);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (29, 2, 4, 28);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (30, 20, 4, 29);


INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (32, 2, 5, 31);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (33, 4, 5, 32);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (34, 2, 5, 33);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (35, 2, 5, 34);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (36, 2, 5, 35);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (37, 4, 5, 36);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (38, 10, 6, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (39, 5, 6, 38);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (40, 5, 6, 39);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (41, 10, 6, 40);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (42, 10, 7, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (43, 5, 7, 42);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (44, 5, 7, 43);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (45, 10, 7, 44);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (46, 5, 8, NULL);

```



```

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (47, 5, 8, 46);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (48, 2, 8, 47);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (49, 2, 8, 48);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (50, 2, 8, 49);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (51, 2, 8, 50);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (52, 2, 8, 51);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (53, 2, 9, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (54, 2, 9, 53);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (55, 2, 9, 54);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (56, 2, 9, 55);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (57, 2, 9, 56);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (58, 2, 10, NULL);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (59, 2, 10, 58);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (60, 2, 10, 59);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (61, 2, 10, 60);

INSERT INTO TAVOLO (CodTavolo, MaxAvventori, Sala, TavoloAdiacente)
VALUES (62, 2, 10, 61);
COMMIT;

/*=====*/
/*=====*/
-- Insert per la tabella TAVOLATA
INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (1, TO_DATE('17/11/2021','dd/mm/yyyy'),4,'CA78432DB');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (2, TO_DATE('17/11/2021','dd/mm/yyyy'),5,'CA66421DA');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (3, TO_DATE('18/11/2021','dd/mm/yyyy'),12,'AU0199811');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (4, TO_DATE('18/11/2021','dd/mm/yyyy'),13,'CA64321CE');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (5, TO_DATE('19/11/2021','dd/mm/yyyy'),18,'AU3425442');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (6, TO_DATE('20/11/2021','dd/mm/yyyy'),8,'CA78432DB');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (7, TO_DATE('20/11/2021','dd/mm/yyyy'),25,'CA53216BE');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (8, TO_DATE('22/11/2021','dd/mm/yyyy'),26,'CA73121FE');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (9, TO_DATE('22/11/2021','dd/mm/yyyy'),27,'AU0567231');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (10, TO_DATE('25/11/2021','dd/mm/yyyy'),37,'CA52131PV');

```

```

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (11, TO_DATE('19/12/2021','dd/mm/yyyy'),26,'CA53216BE');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (12, TO_DATE('20/12/2021','dd/mm/yyyy'),39,'CA87452TB');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (13, TO_DATE('20/12/2021','dd/mm/yyyy'),49,'CA33145OD');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (14, TO_DATE('24/12/2021','dd/mm/yyyy'),50,'AU0236754');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (15, TO_DATE('25/12/2021','dd/mm/yyyy'),53,'CA10946DE');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (16,TO_DATE('25/01/2022','dd/mm/yyyy'),55,'CA332410A');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (17,TO_DATE('25/01/2022','dd/mm/yyyy'),62,'CA13264UI');

INSERT INTO TAVOLATA (CodTavolata, DataArrivo, Tavolo, Cameriere)
VALUES (18,TO_DATE('25/01/2022','dd/mm/yyyy'),50,'CA13264UI');
COMMIT;

/*****
-- Insert per la tabella Avventure
INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('1C3159530','Baldassarre','Pinto',TO_DATE('04/06/1953','dd/mm/yyyy'),'Maschio','Napoli','NA','Napoli','NA','+393493302038','baldassarepinto@gmail.com',35.8,'V',1,1);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('PW824173','Filliberto','Milano',TO_DATE('19/11/1940','dd/mm/yyyy'),'Maschio','Napoli','NA','Napoli','NA','+393598741558','fillibertomilano@outlook.com',36.8,'V',1,1);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('WFI996739','Felicità','Piazza',TO_DATE('16/06/1973','dd/mm/yyyy'),'Femmina','Vindoli','RN','Vindoli','RN','+393658942018','felicitapiazza3@libero.it',36.9,'V',1,2);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('SU2810531','Lea','Barese',TO_DATE('26/07/1985','dd/mm/yyyy'),'Femmina','Cino','80','Caivano','NA','+393259847862','leaBarese26@gmail.com',36.5,'V',1,2);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('BD8518049','Nazzareno','Onio',TO_DATE('07/12/2001','dd/mm/yyyy'),'Maschio','Badia','VV','Marcanise','CE','+393517788436','nazaon2001@outlook.it',37.0,'V',1,3);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('RT5802431','Edoardo','Ricci',TO_DATE('29/01/2002','dd/mm/yyyy'),'Maschio','Ferticani','PG','Benevento','BN','+393254159687','edoRich@libero.it',36.4,'V',1,3);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('WI0726870','Prospero','Padovesi',TO_DATE('24/04/1963','dd/mm/yyyy'),'Maschio','Dorno','PV','Caserta','CE','+393659852014','prPa2404@gmail.com',36.4,'V',1,4);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('CY1881570','Alice','Cocci',TO_DATE('24/05/1951','dd/mm/yyyy'),'Femmina','Calvignasco','MI','Roma','RM','+393654002598','AliceCC@outlook.com',36.2,'V',1,4);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('RT1739919','Quinzio','Pugliesi',TO_DATE('18/05/1989','dd/mm/yyyy'),'Maschio','Ascensione','RA','Recale','CE','+393358947652','QuinzioPugliesi0589@gmail.com',37.0,'F',1,5);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('Y83432289','Ida','Pagnotto',TO_DATE('01/03/1966','dd/mm/yyyy'),'Femmina','Martano','LE','Martano','LE','+393698521655','IdaPagno@gmail.com',36.8,'F',1,5);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('GU1160846','Geronima','Endrizzi',TO_DATE('09/11/2000','dd/mm/yyyy'),'Non specificato','Nago','TN','Vico Equense','NA','+393854759624','GeronimaEnd@gmail.com',35.9,'V',1,6);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('AD8695321','Carolina','Pugliese',TO_DATE('07/04/1953','dd/mm/yyyy'),'Femmina','Casavatore','NA','Casoria','NA','+393275273639','CarolinaPugliese@gmail.com',35.4,'V',1,6);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('BG2548759','Marcella','Pezzali',TO_DATE('28/12/1977','dd/mm/yyyy'),'Femmina','Arzano','NA','Arzano','NA','+393732068075','MarcellaPezzali@gmail.com',36.1,'V',2,7);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('GH5632104','Dolores','Zanzi',TO_DATE('18/01/1955','dd/mm/yyyy'),'Femmina','Volla','NA','Cetara','SA','+393380722982','DoloresDePanza@gmail.com',36.4,'V',2,7);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('FR5849625','Fabiola','Mengolo',TO_DATE('19/09/1956','dd/mm/yyyy'),'Femmina','Giffoni Sei Casali','SA','San Giorgio a Cremano','NA','+393416755818','FabiMengolo@gmail.com',36.5,'V',2,8);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('RS5625487','Daniele','Bernardi',TO_DATE('03/12/1959','dd/mm/yyyy'),'Maschio','Cercola','NA','Portici','NA','+393562319517','DaanBernardi@gmail.com',37.1,'V',2,8);

```



```

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('BR2648058', 'Carolina', 'Lombardi', TO_DATE('07/11/1961', 'dd/mm/yyyy'), 'Femmina', 'Melito di Napoli', 'NA', 'Melito di Napoli', 'NA', '+393516530360', 'CaroLombardi@gmail.com', 37.0, 'V', 2, 9);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('LE2615481', 'Rossana', 'Filizi', TO_DATE('23/09/1965', 'dd/mm/yyyy'), 'Femmina', 'Afragola', 'NA', 'Montecorvino Pugliano', 'SA', '+393316825631', 'RossanaFilizi@gmail.com', 36.8, 'V', 2, 9);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('ML6215403', 'Pierluigi', 'Trapanese', TO_DATE('30/09/1965', 'dd/mm/yyyy'), 'Maschio', 'Casandrino', 'NA', 'Fisciano', 'SA', '+393303659368', 'LuigiTrapanese@gmail.com', 36.8, 'V', 2, 10);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('RS4697315', 'Silvia', 'Golgi', TO_DATE('22/04/1968', 'dd/mm/yyyy'), 'Femmina', 'Casalnuovo di Napoli', 'NA', 'Calvanico', 'SA', '+393869408344', 'SilviGolgi@gmail.com', 36.6, 'V', 2, 10);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('GC9674582', 'Nicola', 'Ricci', TO_DATE('21/08/1969', 'dd/mm/yyyy'), 'Maschio', 'Bellizzi', 'SA', 'San Sebastiano al Vesuvio', 'NA', '+393362582808', 'NicoRic@gmail.com', 36.5, 'V', 2, 10);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('AE6164555', 'Antonio', 'Bettioni', TO_DATE('20/08/1971', 'dd/mm/yyyy'), 'Maschio', 'Mugnano di Napoli', 'NA', 'Mercato San Severino', 'SA', '+393410197804', 'AntoBet@gmail.com', 36.3, 'V', 2, 10);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('PO3031697', 'Amleto', 'Virgilio', TO_DATE('05/01/1975', 'dd/mm/yyyy'), 'Maschio', 'Marano di Napoli', 'NA', 'Maiori', 'SA', '+393488969310', 'AmletoVirgilio@gmail.com', 36.2, 'V', 2, 11);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('MR9748683', 'Mauro', 'Priuli', TO_DATE('15/04/1988', 'dd/mm/yyyy'), 'Maschio', 'Marano di Napoli', 'NA', 'Marano di Napoli', 'NA', '+393394922230', 'MauroPriuli@gmail.com', 36.2, 'V', 2, 11);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('DR0301264', 'Rosaria', 'Argurio', TO_DATE('25/05/1976', 'dd/mm/yyyy'), 'Femmina', 'Dugenta', 'BN', 'Durazzano', 'BN', '+393250404211', 'RosiaArgurio@gmail.com', 36.5, 'V', 3, 12);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('LE9575684', 'Mauro', 'Zola', TO_DATE('17/10/1979', 'dd/mm/yyyy'), 'Maschio', 'Bellona', 'CE', 'Arienza', 'CE', '+393411767562', 'MauroZola@gmail.com', 36.5, 'V', 3, 12);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('IC2558313', 'Donatello', 'Pellegrini', TO_DATE('19/03/1980', 'dd/mm/yyyy'), 'Maschio', 'Vitulazio', 'CE', 'Vitulazio', 'CE', '+393397508638', 'DonaPellegrini@gmail.com', 37.0, 'V', 3, 12);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('TE0864370', 'Giancarlo', 'Boitani', TO_DATE('05/04/1982', 'dd/mm/yyyy'), 'Maschio', 'Cervino', 'CE', 'San Felice a Cancellò', 'CE', '+393416061365', 'GianBoitani@gmail.com', 37.0, 'V', 3, 12);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('Z06283162', 'Melania', 'Poeiro', TO_DATE('23/04/1984', 'dd/mm/yyyy'), 'Femmina', 'Santa Maria a Vico', 'CE', 'Santa Maria la Fossa', 'CE', '+393533370100', 'MelaniaPoeiro@gmail.com', 36.7, 'V', 3, 12);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('IH2648039', 'Renata', 'Pizzetti', TO_DATE('11/11/1985', 'dd/mm/yyyy'), 'Femmina', 'Piana di Monte Verna', 'CE', 'Piana di Monte Verna', 'CE', '+393516025201', 'RenataPizz@gmail.com', 36.7, 'V', 3, 13);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('FU2645786', 'Giuseppina', 'Giannuzzi', TO_DATE('20/07/1990', 'dd/mm/yyyy'), 'Femmina', 'Grazzanise', 'CE', 'Grazzanise', 'CE', '+393520418674', 'GeppyGianni@gmail.com', 36.3, 'V', 3, 13);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('UR6437280', 'Angelica', 'Barbaggio', TO_DATE('02/07/1991', 'dd/mm/yyyy'), 'Femmina', 'San Tammaro', 'CE', 'Santa Maria Capua Vetere', 'CE', '+393510895615', 'LikaBarbaggio@gmail.com', 36.3, 'V', 3, 14);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('OG6407351', 'Melissa', 'Tasso', TO_DATE('16/08/1992', 'dd/mm/yyyy'), 'Femmina', 'Santa Maria Capua Vetere', 'CE', 'Caiazzo', 'CE', '+393309496253', 'MeliTasso@gmail.com', 36.4, 'V', 3, 14);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('OG6427318', 'Gianluca', 'Murri', TO_DATE('19/05/1995', 'dd/mm/yyyy'), 'Maschio', 'Marcianise', 'CE', 'Marcianise', 'CE', '+393528931547', 'GianMurri@gmail.com', 36.5, 'F', 3, 15);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('LR9137648', 'Jolanda', 'Adinolfi', TO_DATE('06/05/1996', 'dd/mm/yyyy'), 'Femmina', 'Portico di Caserta', 'CE', 'Macerata Campania', 'CE', '+393519447992', 'JoleAdi@gmail.com', 37.0, 'F', 3, 15);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('HE5467974', 'Fernando', 'Cilibrazi', TO_DATE('19/07/1997', 'dd/mm/yyyy'), 'Maschio', 'Pozzuoli', 'NA', 'Pozzuoli', 'NA', '+393342696918', 'FernandoCilibrazi@gmail.com', 36.5, 'F', 3, 16);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('TR9746286', 'Roberto', 'Camanni', TO_DATE('24/11/1998', 'dd/mm/yyyy'), 'Maschio', 'Quarto', 'NA', 'Quarto', 'NA', '+393438255310', 'RobiCamanni@gmail.com', 36.4, 'F', 3, 16);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('IT3146975', 'Carmelo', 'Mastroianni', TO_DATE('12/02/1999', 'dd/mm/yyyy'), 'Maschio', 'Cardito', 'NA', 'Curti', 'CE', '+393309290526', 'CarmeloMastroianni@gmail.com', 36.5, 'F', 3, 17);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('HY9746580', 'Ezio', 'Ammaniti', TO_DATE('11/12/2001', 'dd/mm/yyyy'), 'Maschio', 'Frattamaggiore', 'NA', 'Frattamaggiore', 'NA', '+393353722101', 'EzioAmni@gmail.com', 36.8, 'F', 3, 17);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('LR9137684', 'Aria', 'Moschino', TO_DATE('02/05/2002', 'dd/mm/yyyy'), 'Femmina', 'Capodrise', 'CE', 'Massa di Somma', 'NA', '+393301539510', 'AriMoschino@gmail.com', 36.7, 'V', 3, 18);

INSERT INTO AVVENTORE (NumCid, Nome, Cognome, DataN, Sesso, CittaN, ProvN, CittaRes, ProvRes, Telefono, Email, Temperatura, HaGreenpass, Ristorante, Tavolata)
VALUES ('GE9197735', 'Vittorio', 'Pulci', TO_DATE('10/12/2002', 'dd/mm/yyyy'), 'Maschio', 'Grumo Nevano', 'NA', 'Casapulla', 'CE', '+393273531946', 'VikPulci@gmail.com', 36.5, 'V', 3, 18);
COMMIT;

/*
/*
-- Insert per la tabella CASO
INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (1, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, IC3159530, NULL, 1, NULL);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (2, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, SU2810531, NULL, 1, NULL);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (3, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, RT1739919, NULL, 1, NULL);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (4, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'InRisoluzione', NULL, LE2615481, NULL, NULL, 1);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (5, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, LZ9575684, NULL, NULL, 2);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (6, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, OG6427318, NULL, NULL, 2);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (7, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'Risolto', NULL, GE9197735, NULL, NULL, 2);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (8, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, NULL, CA78432DB, 1, NULL);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (9, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, NULL, CA332410A, NULL, 2);

INSERT INTO CASO (CodCaso, DataRegistrazione, StatoCaso, Note, AvventorePositivo, CamerierePositivo, RegistraProprietario, RegistraManager)
VALUES (10, TO_DATE('25/01/2022', 'dd/mm/yyyy'), 'NonRisolto', NULL, NULL, CA66421DA, 1, NULL);
COMMIT;
/*
/*







```

Estratto da DBInsert.sql - Popolamento del database - fine.

5.2 Prospettiva delle tabelle implementate su ORACLE

CODPROPRIETARIO	USERNAME	PASSWORD	NOME	COGNOME	EMAIL
1	sandro05	password00	Santolo	Barretta	santolobarretta05@gmail.com

	CODRISTORANTE	DENOMINAZIONE	INDIRIZZO	TELEFONO	CITTA	PROV	CAP	EMAIL	SITOWEB	PROPRIETARIO
1		1 Bella Napoli	Via Francesco Caracciolo, 1	0813509900	Napoli NA		80122	bellanapoli@gmail.com	www.ristorantebellanapoli.it	1
2		2 Bella Napoli	2 Via Toledo, 30	0813339900	Napoli NA		80134	bellanapoli@gmail.com	www.ristorantebellanapoli.it	1
3		3 La Madonnina	Via Alessandro Manzoni, 5	+390289091122	Milano MI		20121	(null)	(null)	1

MANAGERRISTORANTE									
Colonne: Dati Model Vincoli Autorizzazioni Statistiche Trigger Flashback Dipendenze Dettagli Partizioni Indici SQL									
      Ordina... Filtro:									
	CODMANAGER	USERNAME	PASSWORD	NOME	COGNOME	EMAIL	TELEFONO	RISTORANTEGESTITO	
1	1	angelo02	qwerty0123	Angelo	Di Maio	angelodimaio@gmail.com	+393337060999		2
2	2	mariorossi33	mrl123456	Mario	Rossi	(null)	+393517799111		3

SALA

Colonne

Dati

Model

Vincoli

Autorizzazioni

Statistiche

Trigger

Flashback

Dipendenze

Dettagli

Partizioni







Indici

SQL







Ordina...

Filtro:

	CODSALA	DENOMINAZIONE	CAPENZA	AAVVENTORI	DIMENSIONE	MQ	TIPO	SALA	RISTORANTE
1	1	Sala Pulcinella	50	100	Interna	1			
2	2	Sala Vesuvio	20	60	Interna	1			
3	3	Sala Garden	10	30	Esterna	1			
4	4	Sala Maradona	50	100	Interna	2			
5	5	Sala Partenope	18	25	Interna	2			
6	6	Sala Duomo	30	80	Interna	3			
7	7	Sala Meneghino	30	80	Interna	3			
8	8	Sala Biscione	20	60	Interna	3			
9	9	Sala Exclusive	10	20	Esterna	3			
10	10	Sala Vip	10	(null)	Esterna	3			

CAMERIERE												
Colonne	Dati	Model	Vincoli	Autorizzazioni	Statistiche	Trigger	Flashback	Dipendenze	Dettagli	Partizioni	Indici	SQL
      Ordina... Filtro:												
	NUMCID	NOME	COGNOME	DATAN	SESSO	CITTAN	PROVN	CITTARES	PROVRES	TELEFONO	EMAIL	RISTORANTE
1	CA78432DB	Ciro	Esposito	08-GEN-00	Maschio	Napoli	NA	Qualiano	NA	3517486042	ciroespo@gmail.com	1
2	CA66421DA	Andrea	Russo	10-OTT-90	Maschio	Villaricca	NA	Quarto	NA	3591676343	andrearusso@outlook.it	1
3	AU0199811	Maria	De Rosa	22-MAR-89	Femmina	Pozzuoli	NA	Napoli	NA	3722282194	mariaderosa@gmail.com	1
4	CA64321CE	Rosaria	Romano	30-AGO-93	Femmina	Napoli	NA	Napoli	NA	3218505681	romano01@gmail.com	1
5	AU3425442	Pasquale	Riccio	12-APR-88	Maschio	Caserta	CE	Casoria	NA	3273256337	ricciopas22@outlook.it	1
6	CA53216BE	Gabriele	Marino	21-FEB-93	Maschio	Napoli	NA	Pozzuoli	NA	3553493696	marino0@gmail.com	2
7	CA73121FE	Salvatore	Neri	12-MAR-96	Maschio	Napoli	NA	Bacoli	NA	3215642156	sasineri11@gmail.com	2
8	AU0567231	Luisa	Esposito	08-GEN-00	Femmina	Napoli	NA	Quarto	NA	3445768683	espouli0@outlook.com	2
9	CA52131PV	Carmen	Granata	28-MAG-95	Femmina	Marcianise	CE	Pozzuoli	NA	3830046533	carmen233@gmail.com	2
10	CA87452TB	Laura	Galli	17-NOV-99	Femmina	Milano	MI	Legnano	MI	3746409289	lauragalli56@gmail.com	3
11	CA331450D	Sara	Sala	13-DIC-00	Femmina	Milano	MI	Sesto San Giovanni	MI	3227317200	sara123@gmail.com	3
12	AU0236754	Riella	Bianchi	25-OTT-90	Femmina	Milano	MI	Rho	MI	3229731501	bianchi0@outlook.com	3
13	CA10946DE	Raffaele	Fontana	04-SET-94	Maschio	Milano	MI	Legnano	MI	3619439924	raffaefae144@gmail.com	3
14	CA332410A	Achille	Rinaldi	21-MAG-87	Maschio	Milano	MI	Milano	MI	3283664283	achillerinaldi2@gmail.com	3
15	CA13264UI	Morena	Costa	19-MAR-99	Femmina	Milano	MI	Assago	MI	3194130484	morena43@outlook.com	3

TAVOLO					
Colonne	Dati	Model	Vincoli	Autorizzazioni	Statistiche
Trigger	Flashback	Dipendenze	Dettagli	Partizioni	Indici
SQL					
Ordina...	Filtro:				
1	SALA	2	CODTAVOLO	MAXAVVENTORI	TAVOLOADIACENTE
1	1	1	1	10	(null)
2	1	2	2	5	1
3	1	3	3	5	2
4	1	4	4	2	3
5	1	5	5	2	4
6	1	6	6	2	5
7	1	7	7	2	6
8	1	8	8	2	7
9	1	9	9	20	8
10	2	10	10	5	(null)
11	2	11	11	5	10
12	2	12	12	2	11
13	2	13	13	2	12
14	2	14	14	2	13
15	2	15	15	2	14
16	2	16	16	2	15
17	3	17	17	2	(null)
18	3	18	18	2	17
19	3	19	19	2	18
20	3	20	20	2	19
21	3	21	21	2	20
22	4	22	22	10	(null)
23	4	23	23	5	22
24	4	24	24	5	23
25	4	25	25	2	24
26	4	26	26	2	25
27	4	27	27	2	26
28	4	28	28	2	27
29	4	29	29	2	28
30	4	30	30	20	29
31	5	31	31	2	(null)
32	5	32	32	2	31
33	5	33	33	4	32
34	5	34	34	2	33
35	5	35	35	2	34
36	5	36	36	2	35
37	5	37	37	4	36
38	6	38	38	10	(null)
39	6	39	39	5	38
40	6	40	40	5	39
41	6	41	41	10	40
42	7	42	42	10	(null)
43	7	43	43	5	42
44	7	44	44	5	43
45	7	45	45	10	44
46	8	46	46	5	(null)
47	8	47	47	5	46
48	8	48	48	2	47
49	8	49	49	2	48
50	8	50	50	2	49
51	8	51	51	2	50
52	8	52	52	2	51
53	9	53	53	2	(null)
54	9	54	54	2	53
55	9	55	55	2	54
56	9	56	56	2	55
57	9	57	57	2	56
58	10	58	58	2	(null)
59	10	59	59	2	58
60	10	60	60	2	59
61	10	61	61	2	60
62	10	62	62	2	61

TAVOLATA												
Colonne Dati Model Vincoli Autorizzazioni Statistiche Trigger Flashback Dipendenze Dettagli Partizioni Indici SQL												
Ordina... Filtro:												
	 1	CODTAVOLATA	 3	DATAARRIVO		ORAARRIVO		ORAUSICTA	 2	TAVOLO		CAMERIERE
1		1	17-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				4	CA78432DB	
2		2	17-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				5	CA66421DA	
3		3	18-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				12	AU0199811	
4		4	18-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				13	CA64321CE	
5		5	19-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				18	AU3425442	
6		6	20-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				8	CA78432DB	
7		7	20-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				25	CA53216BE	
8		8	22-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				26	CA73121FE	
9		9	22-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				27	AU0567231	
10		10	25-NOV-21	+00 20:00:00.000000		+00 22:00:00.000000				37	CA52131PV	
11		11	19-DIC-21	+00 20:00:00.000000		+00 22:00:00.000000				26	CA53216BE	
12		12	20-DIC-21	+00 20:00:00.000000		+00 22:00:00.000000				39	CA87452TB	
13		13	20-DIC-21	+00 20:00:00.000000		+00 22:00:00.000000				49	CA33145OD	
14		14	24-DIC-21	+00 20:00:00.000000		+00 22:00:00.000000				50	AU0236754	
15		15	25-DIC-21	+00 20:00:00.000000		+00 22:00:00.000000				53	CA10946DE	
16		16	26-GEN-22	+00 20:00:00.000000		+00 22:00:00.000000				55	CA33241OA	
17		17	25-GEN-22	+00 20:00:00.000000		+00 22:00:00.000000				62	CA13264UI	
18		18	25-GEN-22	+00 20:00:00.000000		+00 22:00:00.000000				50	CA13264UI	

AVVENTORE																											
Colonne Dati Model Vincoli Autorizzazioni Statistiche Trigger Flashback Dipendenze Dettagli Partizioni Indici SQL																											
Ordina... Filtro:																											
	NUMCID	NOME	COGNOME	DATAN	SESSO	CITTAN	PROVIN	CITTARES	PROVRES	TELEFONO	EMAIL	TEMPERATURA	HAGREENPASS	RISTORANTE	TAVOLATA												
1	PW824173	Filiberto	Milano	19-NOV-40	Maschio	Napoli	NA	Napoli	NA	+393598741558	filiberto...	36,8 V		1	1												
2	IC3159530	Baldassarre	Pinto	04-GIU-53	Maschio	Napoli	NA	Napoli	NA	+393493302038	baldassar...	35,8 V		1	1												
3	WF1996739	Felicita	Piazza	16-GIU-73	Femmina	Vindoli	RN	Vindoli	RN	+393658942018	felicitaP...	36,9 V		1	2												
4	SU2810531	Lea	Barese	26-LUG-85	Femmina	Cino	SO	Calvano	NA	+393259947862	leaBarese...	36,5 V		1	2												
5	BD8518049	Mazzareno	Onio	07-DIC-01	Maschio	Badia	VV	Marcianise	CE	+393517788436	nazaon200...	37 V		1	3												
6	KT5802431	Edoardo	Ricci	29-GEN-02	Maschio	Perticani	PV	Benevento	BN	+393254159687	edoRic81...	36,4 V		1	3												
7	W10726870	Prospero	Padovesi	24-APR-63	Maschio	Dorno	FG	Caserta	CE	+393659852014	prPa24048...	36,4 V		1	4												
8	CY1881570	Alice	Cocci	24-MAG-51	Femmina	Calvignasco	MI	Roma	RM	+393654002598	AliceCC8o...	36,2 V		1	4												
9	RT1739919	Quinzio	Pugliesi	18-MAG-89	Maschio	Ascensione	RA	Recale	CE	+393398947852	QuinzioPu...	37 F		1	5												
10	TS9432289	Ida	Pagnotto	01-MAR-66	Femmina	Martano	LE	Martano	LE	+393698521655	IdaPagno8...	36,8 F		1	5												
11	GU1100946	Geronima	Endrizzi	09-NOV-50	Non specificato	Nago	TN	Vico Eguense	NA	+393854759624	GeronimaE...	35,9 V		1	6												
12	AD8695321	Carolina	Pugliese	07-APR-53	Femmina	Casavatore	NA	Casoria	NA	+393275273639	CarolinaP...	35,4 V		1	6												
13	BO2548759	Marcella	Perzali	28-DIC-77	Femmina	Azzano	NA	Azzano	NA	+393732068075	MarcellaP...	36,1 V		2	7												
14	GH5632104	Dolores	Zanzi	18-GEN-55	Femmina	Volla	NA	Cetara	SA	+393380722982	DoloresDe...	36,4 V		2	7												
15	FR5949625	Fabiola	Mengolo	19-SET-56	Femmina	Giffoni Sei Casali	SA	San Giorgio a Cremano	NA	+393416755818	FabiMengo...	36,5 V		2	8												
16	TR5625487	Daniela	Bernardi	03-DIC-59	Maschio	Cercola	NA	Portici	NA	+393562319517	DanBerna...	37,1 V		2	8												
17	BR2649058	Carolina	Lombardi	07-NOV-61	Femmina	Melito di Napoli	NA	Melito di Napoli	NA	+393516530360	CaroLomba...	37 V		2	9												
18	LE2615491	Rossana	Filzi	23-SET-65	Femmina	Afragola	NA	Montecorvino Pugliano	SA	+393316825631	RossanaFi...	36,8 V		2	9												
19	RS4697315	Silvia	Golgi	22-APR-68	Femmina	Caselnovo di Napoli	NA	Calvanico	SA	+393669408344	SilviGolgi...	36,6 V		2	10												
20	ML6215403	Pierluigi	Trapanese	30-SET-65	Maschio	Casandrino	NA	Fisciano	SA	+393303659368	LuigiTrap...	36,8 V		2	10												
21	GC9674582	Nicola	Ricci	21-AGO-69	Maschio	Bellizzi	SA	San Sebastiano al Ve...	NA	+393362582808	NicoRic8g...	36,5 V		2	10												
22	AE6164595	Antonio	Bettoni	20-AGO-71	Maschio	Mugnano di Napoli	NA	Mercato San Severino	SA	+393410197804	AntoBet8g...	36,3 V		2	10												
23	MR9748683	Mauro	Priuli	15-APR-88	Maschio	Marano di Napoli	NA	Marano di Napoli	NA	+393394922230	MauroPriu...	36,2 V		2	11												
24	PO3031697	Amleto	Virgilio	05-GEN-75	Maschio	Marano di Napoli	NA	Maiori	SA	+393488969310	AmletoVir...	36,2 V		2	11												
25	LZ9575684	Mauro	Zola	17-OTT-79	Maschio	Bellona	CE	Arienzo	CE	+393411767562	MauroZola...	36,5 V		3	12												
26	DR0301264	Rosaria	Argurio	25-MAG-76	Femmina	Durazzano	BN	Durazzano	BN	+393250404211	RosiArgur...	36,5 V		3	12												
27	IC2558319	Donatello	Pellegrini	19-MAR-80	Maschio	Vitulazio	CE	Vitulazio	CE	+393397508638	DonaFelle...	37 V		3	12												
28	ZO6283162	Melania	Porio	23-APR-84	Femmina	Santa Maria a Vico	CE	Santa Maria la Fossa	CE	+393533370100	MelaniaPo...	36,7 V		3	12												
29	TE0864370	Giancarlo	Boitani	05-APR-82	Maschio	Cervino	CE	San Felice a Cancelli	CE	+393416061365	GianBoita...	37 V		3	12												
30	TH2648039	Renata	Pizzetti	11-NOV-85	Femmina	Piana di Monte Verna	CE	Piana di Monte Verna	CE	+393516025201	RenataPiz...	36,7 V		3	13												
31	FU2645786	Giuseppina	Gianuzzi	20-LUG-90	Femmina	Grazzanise	CE	Grazzanise	CE	+393520418674	GeppyGian...	36,3 V		3	13												
32	UR6437280	Angelica	Barbarigo	02-LUG-91	Femmina	San Tamarro	CE	Santa Maria Capua Ve...	CE	+393510895615	LiKaBarba...	36,3 V		3	14												
33	OG6407351	Melissa	Tasso	16-AGO-92	Femmina	Santa Maria Capua V...	CE	Caiazzo	CE	+393309496253	GliTasso...	36,4 V		3	14												
34	OG6427318	Gianluca	Murri	19-MAG-95	Maschio	Marcianise	CE	Marcianise	CE	+393528831547	GianMurri...	36,5 F		3	15												
35	LR9137648	Jolanda	Adinolfi	06-MAG-96	Femmina	Portico di Caserta	CE	Macerata Campania	CE	+393519447992	JoleAdi8g...	37 F		3	15												
36	HE5467974	Fernando	Cilibrasi	19-LUG-97	Maschio	Pozzuoli	NA	Pozzuoli	NA	+393426696918	FernandoC...	36,5 F		3	16												
37	TR9746286	Roberto	Camanni	24-NOV-98	Maschio	Quarto	NA	Quarto	NA	+393438255310	RobiCaman...	36,4 F		3	16												
38	IT31466975	Carmelo	Mastroianni	12-FEB-99	Maschio	Cardito	NA	Curti	CE	+393309290526	CarmeloMa...	36,5 F		3	17												
39	HY9746580	Ezio	Ammaniti	11-DIC-01	Maschio	Frattamaggiore	NA	Frattamaggiore	NA	+393353722101	EzioAmma8...	36,8 F		3	17												
40	LH9137684	Aria	Moschino	02-MAG-02	Femmina	Capodrise	CE	Massa di Somma	NA	+393301539510	AriMoschi...	36,7 V		3	18												
41	GE9197735	Vittorio	Pulci	10-DIC-02	Maschio	Grumo Nevano	NA	Casapulla	CE	+393273531946	VikPulci8...	36,5 V		3	18												

CASO								
Colonne Dati Model Vincoli Autorizzazioni Statistiche Trigger Flashback Dipendenze Dettagli Partizioni Indici SQL								
Ordina... Filtro:								
	CODCASO	DATAREGISTRAZIONE	STATOCASO	NOTE	AVVENTOREPOSITIVO	CAMERIEREPOSITIVO	REGISTRAPROPRIETA...	REGISTRAMANAGER
1	1	25-GEN-22	NonRisolto	(null)	IC3159530	(null)	1	(null)
2	2	25-GEN-22	NonRisolto	(null)	SU2810531	(null)	1	(null)
3	3	325-GEN-22	NonRisolto	(null)	RT1739919	(null)	1	(null)
4	4	425-GEN-22	InRisoluzione	(null)	LE2615481	(null)	(null)	1
5	5	525-GEN-22	NonRisolto	(null)	LZ9575684	(null)	(null)	2
6	6	625-GEN-22	NonRisolto	(null)	OG6427318	(null)	(null)	2
7	7	725-GEN-22	Risolto	(null)	GE9197735	(null)	(null)	2
8	8	825-GEN-22	NonRisolto	(null)	(null)	CA78432DB	1	(null)
9	9	925-GEN-22	NonRisolto	(null)	(null)	CA332410A	(null)	2
10	10	1025-GEN-22	NonRisolto	(null)	(null)	CA66421DA	1	(null)

Prospettiva delle tabelle implementate su ORACLE - fine.

5.3 Implementazione delle query

Seguono le definizioni delle query in SQL tratte dal file “DBQuery.sql”, mostrando successivamente l’output ottenuto dall’esecuzione sul database implementato. Per la descrizione del funzionamento delle query si rimanda al [dizionario delle interrogazioni](#).

5.3.1 Numero giornaliero di avventori per ristorante

```
-- 1.Numero giornaliero di avventori per ristorante
-- Supponiamo che il ristorante dato abbia CodRistorante = 1 e la DataArrivo = '17/11/2021'.
SELECT R.Denominazione AS RISTORANTE, T.DataArrivo AS DATA, COUNT(A.NumCid) AS TOT_GIORNALIERO_AVVENTORI
FROM AVVENTORE A JOIN TAVOLATA T ON A.Tavolata = T.CodTavolata JOIN RISTORANTE R ON R.CodRistorante = A.Ristorante
WHERE A.Ristorante = 1 AND T.DataArrivo = TO_DATE('17/11/2021','dd/mm/yyyy')
GROUP BY R.Denominazione, T.DataArrivo;
```

Output:

RISTORANTE	DATA	TOT_GIORNALIERO_AVVENTORI
Bella Napoli	17-NOV-21	4

5.3.2 Numero mensile di avventori per ristorante

```
-- 2.Numero mensile di avventori per ristorante
-- Supponiamo che il ristorante dato abbia CodRistorante = 1 e il mese di riferimento sia 11/2021
SELECT R.Denominazione AS RISTORANTE, TO_CHAR(T.DataArrivo, 'mm ') AS MESE, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(A.NumCid) AS TOT_MENSILE_AVVENTORI
FROM AVVENTORE A JOIN TAVOLATA T ON A.Tavolata = T.CodTavolata JOIN RISTORANTE R ON R.CodRistorante = A.Ristorante
WHERE A.Ristorante = 1 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021 AND TO_CHAR(T.DataArrivo, 'mm') = 11
GROUP BY R.Denominazione, TO_CHAR(T.DataArrivo, 'mm '), TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

RISTORANTE	MESE	ANNO	TOT_MENSILE_AVVENTORI
Bella Napoli	11	2021	12

5.3.3 Numero giornaliero di avventori per tutti i ristoranti di un proprietario

```
-- 3.Numero giornaliero di avventori per tutti i ristoranti di un proprietario
-- Supponiamo che il proprietario abbia CodProprietario = 1 e la DataArrivo = '17/11/2021'
SELECT R.Proprietario, T.DataArrivo AS DATA, COUNT(A.NumCid) AS TOT_GIORNALIERO_AVVENTORI_RISTORANTI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON A.Tavolata = T.CodTavolata
WHERE R.Proprietario = 1 AND T.DataArrivo = TO_DATE('17/11/2021','dd/mm/yyyy')
GROUP BY R.Proprietario, T.DataArrivo;
```

Output:

PROPRIETARIO	DATA	TOT_GIORNALIERO_AVVENTORI_RISTORANTI
1	17-NOV-21	4

5.3.4 Numero mensile di avventori per tutti i ristoranti di un proprietario

```
-- 4. Numero mensile di avventori per tutti i ristoranti di un proprietario
-- Supponiamo che il proprietario abbia CodProprietario = 1 e il
SELECT R.Proprietario, TO_CHAR(T.DataArrivo, 'mm ') AS MESE, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(A.NumCid) AS TOT_MENSILE_AVVENTORI_RISTORANTI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON A.Tavolata = T.CodTavolata
WHERE R.Proprietario = 1 AND TO_CHAR(T.DataArrivo, 'mm') = 11 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021
GROUP BY R.Proprietario, TO_CHAR(T.DataArrivo, 'mm '), TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

PROPRIETARIO	MESE	ANNO	TOT_MENSILE_AVVENTORI_RISTORANTI
1	11	2021	22

5.3.5 Casi positivi di un determinato ristorante per data di arrivo della tavolata

```
-- 5. Casi positivi di un determinato ristorante per data di arrivo della tavolata
-- Supponiamo che il codice del ristorante sia CodRistorante = 1 e che la data di arrivo della tavolata sia il: '17/11/2021'
SELECT R.Denominazione AS RISTORANTE, T.DataArrivo AS DATA_ARV, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE A.Ristorante = 1 AND T.DataArrivo = TO_DATE('17/11/2021', 'dd/mm/yyyy')
GROUP BY R.Denominazione, T.DataArrivo;
```

Output:

RISTORANTE	DATA_ARV	TOT_RISULTATI_POSITIVI
Bella Napoli	17-NOV-21	2

5.3.6 Casi positivi di un determinato ristorante per mese di arrivo della tavolata

```
-- 6. Casi positivi di un determinato ristorante per mese di arrivo della tavolata
-- Supponiamo che il codice del ristorante sia CodRistorante = 1 e che il mese di arrivo della tavolata sia 11/2021
SELECT R.Denominazione AS RISTORANTE, TO_CHAR(T.DataArrivo, 'mm ') AS MESE, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE A.Ristorante = 1 AND TO_CHAR(T.DataArrivo, 'mm') = 11 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021
GROUP BY R.Denominazione, TO_CHAR(T.DataArrivo, 'mm '), TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

RISTORANTE	MESE	ANNO	TOT_RISULTATI_POSITIVI
Bella Napoli	11	2021	3

5.3.7 Casi positivi di un determinato ristorante per anno di arrivo della tavolata

```
-- 7. Casi positivi di un determinato ristorante per anno di arrivo della tavolata
-- Supponiamo che il codice del ristorante sia CodRistorante = 1 e che l'anno di arrivo della tavolata sia il 2021
SELECT R.Denominazione AS RISTORANTE, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE A.Ristorante = 1 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021
GROUP BY R.Denominazione, TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

RISTORANTE	ANNO	TOT_RISULTATI_POSITIVI
Bella Napoli	2021	3

5.3.8 Casi positivi di tutti i ristoranti di un proprietario per data di arrivo della tavolata

```
-- 8.Casi positivi di tutti i ristoranti di un proprietario per data di arrivo della tavolata
-- Supponiamo che il codice del proprietario sia CodProprietario = 1 e che la data di arrivo della tavolata sia il: '17/11/2021'
SELECT R.Proprietario, T.DataArrivo AS DATA_ARV, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI_RISTORANTI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE R.Proprietario = 1 AND T.DataArrivo = TO_DATE('17/11/2021','dd/mm/yyyy')
GROUP BY R.Proprietario, T.DataArrivo;
```

Output:

PROPRIETARIO	DATA_ARV	TOT_RISULTATI_POSITIVI_RISTORANTI
1	17-NOV-21	2

5.3.9 Casi positivi di tutti i ristoranti di un proprietario per mese di arrivo della tavolata

```
-- 9.Casi positivi di tutti i ristoranti di un proprietario per mese di arrivo della tavolata
-- Supponiamo che il codice del proprietario sia CodProprietario = 1 e che il mese di arrivo della tavolata sia 11/2021
SELECT R.Proprietario, TO_CHAR(T.DataArrivo, 'mm ') AS MESE, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI_RISTORANTI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE R.Proprietario = 1 AND TO_CHAR(T.DataArrivo, 'mm') = 11 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021
GROUP BY R.Proprietario, TO_CHAR(T.DataArrivo, 'mm '), TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

PROPRIETARIO	MESE	ANNO	TOT_RISULTATI_POSITIVI_RISTORANTI
1	11	2021	4

5.3.10 Casi positivi di tutti i ristoranti di un proprietario per anno di arrivo della tavolata

```
-- 10.Casi positivi di tutti i ristoranti di un proprietario per anno di arrivo della tavolata
-- Supponiamo che il codice del proprietario sia CodProprietario = 1 e che l'anno di arrivo della tavolata sia il 2021
SELECT R.Proprietario, TO_CHAR(T.DataArrivo, 'yyyy') AS ANNO, COUNT(C.AvventorePositivo) TOT_RISULTATI_POSITIVI_RISTORANTI
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN CASO C ON C.AvventorePositivo = A.NumCid
WHERE R.Proprietario = 1 AND TO_CHAR(T.DataArrivo, 'yyyy') = 2021
GROUP BY R.Proprietario, TO_CHAR(T.DataArrivo, 'yyyy');
```

Output:

PROPRIETARIO	ANNO	TOT_RISULTATI_POSITIVI_RISTORANTI
1	2021	6

5.3.11 Informazioni sugli avventori risultati positivi in un ristorante

```
-- 11.Informazioni sugli avventori risultati positivi in un ristorante
-- Supponiamo che il codice del ristorante che ha accolto gli avventori risultati positivi sia CodRistorante = 1
SELECT C.CodCaso, C.DataRegistrazione AS DATAR, CAST(R.Denominazione AS VARCHAR2(20)) AS RISTORANTE, A.NumCid, CAST(A.Nome AS VARCHAR2(30)) AS Nome,
CAST(A.Cognome AS VARCHAR2(30)) AS COGNOME, A.DataN, A.Telefono, A.HaGreenpass AS G, T.CodTavolata, T.DataArrivo, T.Cameriere, T.Tavolo, TA.TavoloAdiacente
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN CASO C ON C.AvventorePositivo = A.NumCid
JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN TAVOLO TA ON TA.CodTavolo = T.Tavolo
WHERE A.Ristorante = 1
ORDER BY C.CodCaso;
```

Output:

CODCASO	DATAR	RISTORANTE	NUMCID	NOME	COGNOME	DATAN	TELEFONO	G	CODTAVOLATA	DATAARRIV	CAMERIERE	TAVOLO	TAVOLOADIACENTE
1	25-GEN-22	Bella Napoli	IC3159530	Baldassarre	Pinto	04-GIU-53	+393493302038	V	1	17-NOV-21	CA78432DB	4	3
2	25-GEN-22	Bella Napoli	SU2810531	Lea	Barese	26-LUG-85	+393259847862	V	2	17-NOV-21	CA66421DA	5	4
3	25-GEN-22	Bella Napoli	RT1739919	Quinzio	Pugliesi	18-MAG-89	+393398947852	F	5	19-NOV-21	AU3425442	18	17

5.3.12 Informazioni sugli avventori risultati positivi in tutti i ristoranti di un proprietario

```
-- 12. Informazioni sugli avventori risultati positivi in tutti i ristoranti di un proprietario
-- Supponiamo che il codice del proprietario dei ristoranti sia CodProprietario = 1
SELECT C.CodCaso, C.DataRegistrazione AS DATAR, CAST(R.Denominazione AS VARCHAR2(20)) AS RISTORANTE, A.NumCid, CAST(A.Nome AS VARCHAR2(30)) AS Nome,
CAST(A.Cognome AS VARCHAR2(30)) AS Cognome, A.DataN, A.Telefono, A.HaGreenpass AS G, T.CodTavolata, T.DataArrivo, T.Cameriere, T.Tavolo, TA.TavoloAdiacente
FROM RISTORANTE R JOIN AVVENTORE A ON R.CodRistorante = A.Ristorante JOIN CASO C ON C.AvventorePositivo = A.NumCid
JOIN TAVOLATA T ON T.CodTavolata = A.Tavolata JOIN TAVOLO TA ON TA.CodTavolo = T.Tavolo
WHERE R.Proprietario = 1
ORDER BY C.CodCaso;
```

Output:

CODCASO	DATAR	RISTORANTE	NUMCID	NOME	COGNOME	DATAN	TELEFONO	G	CODTAVOLATA	DATAARRIV	CAMERIERE	TAVOLO	TAVOLOADIACENTE
1	25-GEN-22	Bella Napoli	IC3159530	Baldassarre	Pinto	04-GIU-53	+393493302038	V	1	17-NOV-21	CA78432DB	4	3
2	25-GEN-22	Bella Napoli	SU2810531	Lea	Barese	26-LUG-85	+393259847862	V	2	17-NOV-21	CA66421DA	5	4
3	25-GEN-22	Bella Napoli	RT1739919	Quinzio	Pugliesi	18-MAG-89	+393399847852	F	5	19-NOV-21	AU3425442	18	17
4	25-GEN-22	Bella Napoli 2	LE2615481	Rossena	Filzi	23-SET-65	+393316825631	V	9	22-NOV-21	AU0567231	27	26
5	25-GEN-22	La Madonnina	LZ9575684	Mauro	Zola	17-OTT-79	+393411767562	V	12	20-DIC-21	CA87452TB	39	38
6	25-GEN-22	La Madonnina	OG6427318	Gianluca	Murri	19-MAG-95	+393528831547	F	15	25-DIC-21	CA10946DE	53	
7	25-GEN-22	La Madonnina	GE9197735	Vittorio	Fulci	10-DIC-02	+393273531946	V	18	25-GEN-22	CA13264UI	50	49

5.3.13 Informazioni sui camerieri risultati positivi in un ristorante

```
-- 13. Informazioni sui camerieri risultati positivi in un ristorante
-- Supponiamo che il codice del ristorante per cui lavorano i camerieri risultati positivi sia CodRistorante = 1
SELECT CA.CodCaso, CA.DataRegistrazione AS DATAR, CAST(R.Denominazione AS VARCHAR2(20)) AS RISTORANTE, C.NumCid, CAST(C.Nome AS VARCHAR2(30)) AS Nome,
CAST(C.Cognome AS VARCHAR2(30)) AS Cognome, C.DataN, C.Telefono
FROM RISTORANTE R JOIN CAMERIERE C ON R.CodRistorante = C.Ristorante JOIN CASO CA ON CA.CamerierePositivo = C.NumCid
WHERE C.Ristorante = 1
ORDER BY CA.CodCaso;
```

Output:

CODCASO	DATAR	RISTORANTE	NUMCID	NOME	COGNOME	DATAN	TELEFONO
8	25-GEN-22	Bella Napoli	CA78432DB	Ciro	Esposito	08-GEN-00	3517486042
10	25-GEN-22	Bella Napoli	CA66421DA	Andrea	Russo	10-OTT-90	3591676343

5.3.14 Informazioni sui camerieri risultati positivi in tutti i ristoranti di un proprietario

```
-- 14. Informazioni sui camerieri risultati positivi in tutti i ristoranti di un proprietario
-- Supponiamo che il codice del proprietario dei ristoranti sia CodProprietario = 1
SELECT CA.CodCaso, CA.DataRegistrazione AS DATAR, CAST(R.Denominazione AS VARCHAR2(20)) AS RISTORANTE, C.NumCid, CAST(C.Nome AS VARCHAR2(30)) AS Nome,
CAST(C.Cognome AS VARCHAR2(30)) AS Cognome, C.DataN, C.Telefono
FROM RISTORANTE R JOIN CAMERIERE C ON R.CodRistorante = C.Ristorante JOIN CASO CA ON CA.CamerierePositivo = C.NumCid
WHERE R.Proprietario = 1
ORDER BY CA.CodCaso;
```

Output:

CODCASO	DATAR	RISTORANTE	NUMCID	NOME	COGNOME	DATAN	TELEFONO
8	25-GEN-22	Bella Napoli	CA78432DB	Ciro	Esposito	08-GEN-00	3517486042
9	25-GEN-22	La Madonnina	CA332410A	Achille	Rinaldi	21-MAG-87	3283664283
10	25-GEN-22	Bella Napoli	CA66421DA	Andrea	Russo	10-OTT-90	3591676343

5.3.15 Avventori positivi con o senza green pass

```
-- 15.Avventori positivi con o senza green pass
SELECT COUNT(C.CodCaso) AS AVVENTORI_POSITIVI_CON_GREENPASS
FROM CASO C JOIN AVVENTORE A ON C.AvventorePositivo = A.NumCid
WHERE A.HaGreenpass='V';

SELECT COUNT(C.CodCaso) AS AVVENTORI_POSITIVI_SENZA_GREENPASS
FROM CASO C JOIN AVVENTORE A ON C.AvventorePositivo = A.NumCid
WHERE A.HaGreenpass='F';
```

Output:

AVVENTORI_POSITIVI_CON_GREENPASS

5

AVVENTORI_POSITIVI_SENZA_GREENPASS

2

5.3.16 Numero di avventori medio per tavolata di un ristorante

```
-- 16.Numero di avventori medio per tavolata di un ristorante
-- Supponendo di voler conoscere la media di avventori per tavolata del ristorante di CodRistorante = 2
SELECT T.Ristorante, AVG(T.Partecipanti) AS MEDIA_AVVENTORI_PER_TAVOLATA
FROM TAVOLATE T
WHERE T.Ristorante = 2
GROUP BY T.Ristorante;
```

Output:

RISTORANTE MEDIA_AVVENTORI_PER_TAVOLATA

2

2,4

Estratti da DBQuery.sql - Implementazione delle query - fine.



DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

Fine

Angelo DI MAIO
MATRICOLA N86003699
angelo.dimaio3@studenti.unina.it

Santolo BARRETTA
MATRICOLA N86003666
san.barretta@studenti.unina.it