



PROGETTO DI BASI DI DATI

GAME LAB

Progetto di basi di Dati

~Game Lab~

1. Raccolta delle specifiche della realtà d'interesse

-Descrizione-

Si vuole progettare un database per la gestione di un **Game Lab**.

L'edificio è suddiviso in due piani, al piano terra si trova la sala bar ed il punto di vendita elettronica, al piano superiore troviamo la sala intrattenimento dove i clienti possono giocare all'interno di una postazione.

L'edificio è fornito di personale a cui è associato un **ID** univoco, e di loro si vuole conoscere il nome, cognome, IBAN, telefono, e-mail, stipendio. A sua volta il personale si suddivide in:

Tecnico: addetto alla manutenzione delle postazioni di gioco, provvisto di certificazione.

Si vuole conoscere quando è stata effettuata la manutenzione, il tipo ed il prezzo; ogni manutenzione viene effettuata da un solo tecnico ed è relativa ad una singola postazione.

Supporto: addetto al supporto ai clienti nell'utilizzo delle postazioni;

Specializzato elettronica: addetto alla vendita di prodotti elettronici, provvisto di attestato;

Specializzato bar: addetto alla vendita di prodotti da ristoro e consumazione, provvisto di attestato HACCP che certifica la vendita.

Per ogni cliente si vuole conoscere il **CF**, nome, cognome e il numero di telefono. Egli può ordinare uno scontrino al bar, fare acquisti di elettronica o prenotare una postazione da gioco.

Agli scontrini viene associato un **codice** univoco, data, ora, tipo ed il costo totale dell'acquisto. Del singolo prodotto del bar si vuole conoscere il **codice seriale**, che identifica il prodotto, la marca, il prezzo, il nome, la dimensione (piccolo, medio, grande), la data di scadenza e il tipo di prodotto. Per i prodotti di elettronica si vogliono invece conoscere il **codice seriale**, il nome, la marca, il prezzo, il tipo, la tipologia di connessione e il colore.

Per accedere alle postazioni ogni cliente deve effettuare una prenotazione; per ciascuna di esse si deve salvare l'ora di prenotazione, la data, l'ora, il costo, il numero di ore, ed il **numero progressivo**, che la identifica.

Ogni prenotazione effettuata è associata ad una singola postazione di gioco, ognuna con un **numero seriale** associato. Si vuole tenere conto dello stato della postazione, il nome, a che generazione appartiene ed il numero massimo di giocatori. Inoltre, le postazioni si suddividono in: **Console, PC, Realtà virtuale, Simulatore di guida**.

Su di esse sono installati dei giochi, di cui si vuole salvare il **titolo**, editore, il genere e l'anno d'uscita.

-Specifiche della realtà d'interesse-

La realtà che andremo a rappresentare, comprende la gestione di un edificio LAN con punto vendita elettronica e una zona ristoro. L'edificio comprende due piani, al piano terra è possibile rifornirsi alla zona ristoro oppure effettuare acquisti di componenti elettroniche o key da riscattare, al piano superiore è presente la sala LAN, dove pagando una tariffa di 3€\h si può usufruire delle varie postazioni presenti. Tra le varie postazioni si trovano principalmente computer da gaming, ma anche simulatori di guida, simulatori VR e varie console, tra cui anche console per il retro-gaming.

L'edificio è stato creato per puro scopo ricreativo ed è aperto tutti i giorni dalle ore 8:00 alle 24:00.

È anche possibile effettuare streaming siccome la sala da gaming è fornita della migliore attrezzatura per streaming di eventi, la qualità di gioco è molto elevata, questo perché l'hardware dei vari dispositivi rende l'esperienza di gioco fluida, inoltre le postazioni presentano tavoli con livello di altezza regolabile, tastiere meccaniche e sedili da gaming. Facendo riferimento all'help desk è possibile creare sessioni di gioco personalizzate in caso si volessero organizzare tornei in compagnia.

Le varie postazioni offrono principalmente giochi multiplayer, per i quali viene utilizzato un account da gioco standard; quindi, non è necessario tenere conto dell'account di gioco dei singoli clienti. Tutte le postazioni presentano un timer in alto a destra sul monitor per tenere conto del proprio tempo a disposizione, quando raggiunge il termine, il sistema andrà in shut down automaticamente.

Se si è interessati all'acquisto di console da gaming, è possibile far riferimento allo store che si trova al piano terra, dove il cliente potrà trovare oltre a prodotti come, stampanti, tastiere, mouse, singole componenti per assemblare un computer, anche gli stessi computer preassemblati della sala da gaming, perfettamente funzionanti e pronti all'utilizzo.

Mensilmente viene effettuata una manutenzione a tutti i dispositivi nella sala da gaming da parte di un tecnico specializzato, di questa sarà necessario tenere conto di quando è stata effettuata.

La base di dati tiene conto dello stato di ogni singola postazione, e ogni mese per i prodotti maggiormente acquistati.

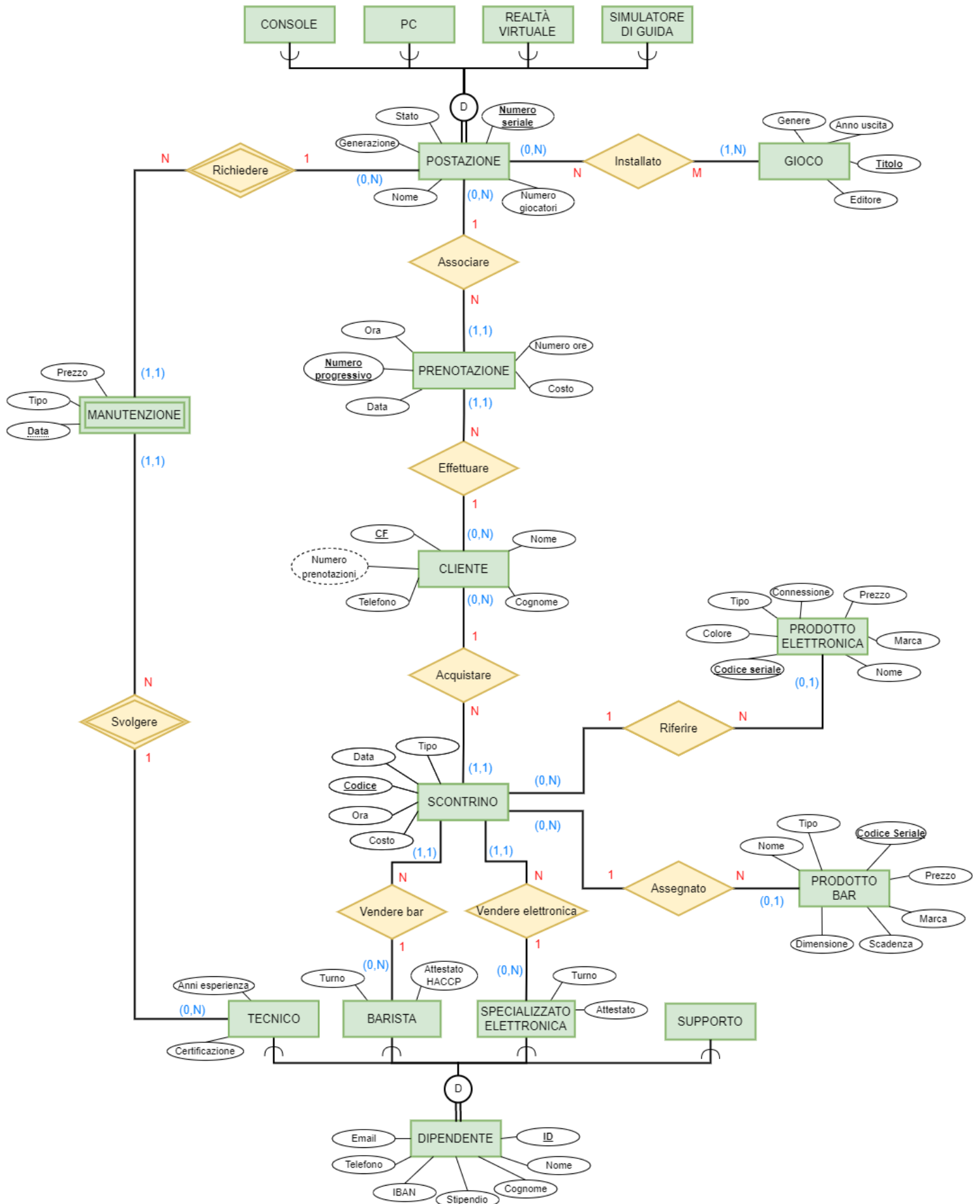
-Glossario dei termini-

TERMINE	SIGNIFICATO
Sala LAN	È uno spazio dedicato al gaming su una rete locale o multiplayer.
LAN	Utilizzata per connettere computer e altri dispositivi all'interno di un'area geografica limitata, come una casa, un ufficio o una scuola.
Editore	Società che pubblica e distribuisce videogiochi. Sono responsabili del marketing, della promozione e della vendita del gioco al pubblico.
Realtà virtuale	Simulazione di un ambiente tridimensionale che può essere sperimentato e con cui si può interagire come se fosse reale. Si ottiene attraverso l'uso di un auricolare o altro dispositivo che fornisce uno o più schermi davanti agli occhi.
Simulatore di guida	Dispositivo che simula l'esperienza di guida di un veicolo.
Postazione	Progettato per giocare ai videogiochi. È in grado di gestire i requisiti esigenti dei giochi moderni.
Generazione	Si riferisce all'insieme specifico di tecnologie hardware e software utilizzate nel dispositivo. Ogni nuova generazione rappresenta un progresso significativo nella tecnologia e nelle capacità.
IBAN	È un codice univoco utilizzato per identificare un conto bancario specifico.
Attestato / Certificazione	È un documento formale che funge da prova o riconoscimento di uno specifico risultato o realizzazione.
Attestato HACCP	È un corso di formazione atto a fornire gli strumenti e le conoscenze per garantire che i prodotti alimentari siano sicuri per il consumo umano.
Key	È un codice di attivazione univoco utilizzato per riscattare giochi, saldo associato ad un account su uno store o licenze software
Retro gaming	Consiste nel giocare a videogiochi classici o vecchie console che sono state rilasciate in passato

2. Progettazione concettuale della base di dati

-Schema EER-

Procedendo con la fase di progettazione concettuale della base di dati, si perviene al seguente schema EER:



-Dizionario delle entità-

Legenda: entità debole, chiave candidata, sotto entità, attributo ridondante

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Dipendente	Persona che lavora all'interno della struttura	ID, Nome, Cognome, IBAN, Telefono, Email, Stipendio	ID
Supporto	Persona dedicata al supporto generale delle postazioni	/ /	/ /
Specializzato bar	Persona dedicata alla vendita nel bar	Attestato HACCP, Turno	/ /
Specializzato elettronica	Persona dedicata alla vendita di prodotti di elettronica	Attestato, Turno	/ /
Tecnico	Persona dedicata alla manutenzione delle postazioni	Certificazione, Anni esperienza	/ /
Scontrino	Scontrino rilasciato al cliente con il riepilogo dei prodotti acquistati	Codice, Costo, Tipo, Data, Ora	Codice
Prodotto bar	Prodotto cibario o di bevveraggio servito ai clienti	Codice seriale, Tipo, Nome, Prezzo, Marca, Scadenza, Dimensione	Codice seriale
Prodotto elettronica	Key o vari prodotti elettronici, come console, smartphone e pc preassemblati	Codice seriale, Nome, Marca, Prezzo, Tipo, Colore, Connessione	Codice seriale
Cliente	Persona che acquista un prodotto o utilizza una postazione	CF, Nome, Cognome, Telefono, Numero prenotazioni	CF
Prenotazione	Permette al cliente di utilizzare la postazione	Numero progressivo, Ora, Numero ore, Costo, Data	Numero progressivo
Postazione	Dispositivo usato dal cliente per giocare al gioco di suo interesse	Numero seriale, Stato, Generazione, Numero giocatori, Nome	Numero seriale
Manutenzione	Insieme di attività volte alla risoluzione di un problema alla postazione	Data, Prezzo, Tipo	Data

Gioco	Software installato nella postazione volto all'intrattenimento del cliente	Anno uscita, Titolo, Editore, Genere	Titolo
Console	Dispositivo pensato per i giochi che fanno uso di un controller	/ /	/ /
PC	Dispositivo la cui funzione è di permettere al cliente di usare determinati giochi	/ /	/ /
Realtà virtuale	Dispositivo che si applica all'altezza degli occhi, che permette al cliente l'uso di determinati giochi	/ /	/ /
Simulatore di guida	Hardware composto da un sedile reclinabile e un volante che permettono l'uso di giochi di guida al cliente	/ /	/ /

-Dizionario delle relazioni-

RELAZIONE	DESCRIZIONE	ENTITÀ COINVOLTE	ATTRIBUTI
Vendere Bar	Un personale specializzato vende un prodotto del bar	Barista (0,N) Scontrino (1,1)	/ /
Vendere Elettronica	Un personale specializzato vende un prodotto di elettronica	Specializzato elettronica (0,N) Scontrino (1,1)	/ /
Assegnato	Un prodotto del bar viene assegnato ad uno scontrino	Prodotto bar (0,1) Scontrino bar (0,N)	/ /
Acquistare	Un cliente ordina uno scontrino	Scontrino (1,1) Cliente (0,N)	/ /
Riferire	Un prodotto di elettronica si riferisce ad un acquisto di elettronica	Prodotto elettronica (0,1) Scontrino elettronica(0,N)	/ /
Effettuare	Un cliente effettua una prenotazione	Cliente (0,N) Prenotazione (1,1)	/ /
Associare	Una prenotazione è associata ad una postazione	Prenotazione (1,1) Postazione (0,N)	/ /
Richiedere	Una postazione richiede manutenzione	Postazione (0,N) Manutenzione (1,1)	/ /
Installato	Un gioco è installato in una postazione	Postazione (0,N) Gioco (1,N)	/ /
Svolgere	Un tecnico svolge una manutenzione	Manutenzione (1,1) Tecnico (0,N)	/ /

-Vincoli non esprimibili-

ENTITÀ	ATTRIBUTO	VINCOLO
Postazione	Numero giocatori	≥ 1 e ≤ 6
Postazione	Numero seriale	di 12 caratteri
Cliente	CF	di 16 caratteri
Cliente	Telefono	di massimo 12 cifre
Dipendente	IBAN	di 27 caratteri
Dipendente	Telefono	di massimo 12 cifre
Scontrino	Codice	di 8 caratteri

3. Definizione delle procedure per la gestione della base di dati

-Tavola dei volumi-

Definiamo di seguito la tavola dei volumi della base di dati.

CONCETTO	TIPO	CARICO APPLICATIVO
Dipendente	E	40
Tecnico	SE	6
Specializzato bar	SE	12
Specializzato elettronica	SE	8
Supporto	SE	14
Scontrino	E	3250
Prodotto bar	E	3800
Prodotto elettronica	E	3150
Gioco	E	200
Postazione	E	26
Pc	SE	10
Console	SE	6
Simulatore di guida	SE	5
Realtà virtuale	SE	5
Cliente	E	1000
Prenotazione	E	1600
Manutenzione	E	420
<i>Vendere</i>	R	3250
<i>Assegnato</i>	R	3600
<i>Acquistare</i>	R	3250
<i>Effettuare</i>	R	1500
<i>Riferire</i>	R	3000
<i>Associare</i>	R	1600
<i>Richiedere</i>	R	420
<i>Svolgere</i>	R	420
<i>Installato</i>	R	390

Assegnato (in media ogni scontrino riguarda due prodotti)

Riferire (in media ogni scontrino riguarda due prodotti)

Installato (in media su ogni postazione sono installati 15 giochi)

-Tavola delle operazioni-

Definiamo di seguito la tavola delle operazioni per la gestione dei dati memorizzati nella base di dati.

	OPERAZIONE	TIPO	FREQUENZA	FREQUENZA ANNUALE
1	Inserire un nuovo cliente	I	50/mm	600/aa
2	Inserire un nuovo prodotto bar	I	210/mm	2520/aa
3	Inserire un nuovo prodotto elettronico	I	60/mm	720/aa
4	Inserire una nuova postazione	I	2/aa	2/aa
5	Inserire un nuovo dipendente	I	3/aa	3/aa
6	Effettuare una prenotazione	I	12/gg	4380/aa
7	Effettuare un acquisto al bar	I	35/gg	12775/aa
8	Effettuare un acquisto di elettronica	I	20/gg	7300/aa
9	Richiedere una manutenzione	I	28/mm	24/aa
10	Installare un gioco	I	25/mm	300/aa
11	Stampare numero seriale, nome, tipo e stato di ogni postazione, compreso il numero di prenotazioni	B	1/mm	
12	Stampare nome, cognome, numero di telefono ed email del tecnico con più anni di esperienza	B	1/mm	12/aa
13	Stampare i dati del cliente con maggior numero di prenotazioni	B	1/mm	12/aa
14	Stampare su quali postazioni è installato un determinato gioco	I	10/gg	3650/aa
15	Stampare i dati del cliente con maggior numero di ordini al bar e acquisti di elettronica, che non abbia effettuato nessuna prenotazione	B	1/mm	12/aa
16	Stampare i dati del cliente con la spesa maggiore in prenotazioni			
17	Stampare la postazione maggiormente prenotata per un certo numero di giocatori	I	1/mm	12/aa
18	Stampare i dati del cliente che ha speso di più	B	1/mm	12/aa

4. Progettazione logica

-Analisi delle ridondanze-

Lo schema presentato in precedenza possiede il seguente attributo ridondante: "numero prenotazioni" dell'entità Cliente. Questo attributo è infatti derivabile attraverso la partecipazione dell'istanza di Cliente nella relazione "Cliente effettuare Prenotazione". Per decidere se mantenere o meno il dato derivabile è necessario calcolare il numero di accessi, nelle operazioni che lo coinvolgono, con e senza quest'ultimo. Supponiamo che l'attributo "numero prenotazioni" sia uno SMALL INT UNSIGNED, quindi esso occuperà 2 byte. Considerando che il volume dell'entità Cliente è 1000 avremo una quantità di memoria occupata dall'attributo di 2000 byte, cioè 2 kb.

-Tavola degli accessi-

Operazione 6 (Effettuare una prenotazione)

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	S	Prenotazione	E	1	S
Associare	R	1	S	Associare	R	1	S
Effettuare	R	1	S	Effettuare	R	1	S
Cliente	E	1	L				
Cliente	E	1	S				
Totale	$(4S + 1L) * 4380/aa = 39420/aa$			$(3S) * 4380/aa = 26280/aa$			

Operazione 13 (Stampare i dati del cliente con maggior numero di prenotazioni)

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Cliente	E	1000	L	Effettuare	R	1500	L
Totale	$(1000L) * 12/aa = 12000/aa$			$(1500L) * 12/aa = 18000/aa$			

Operazione 15 (Stampare i dati del cliente con maggior numero di ordini al bar e acquisti di elettronica, che non abbia effettuato nessuna prenotazione)

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accessi	Tipo	Concetto	Costrutto	Accessi	Tipo
Cliente	E	1000	L	Effettuare	R	1500	L
Acquistare	R	3250	L	Acquistare	R	3250	L
Totale	$(4250L) * 12/aa = 51000/aa$			$(4750L) * 12/aa = 57000/aa$			

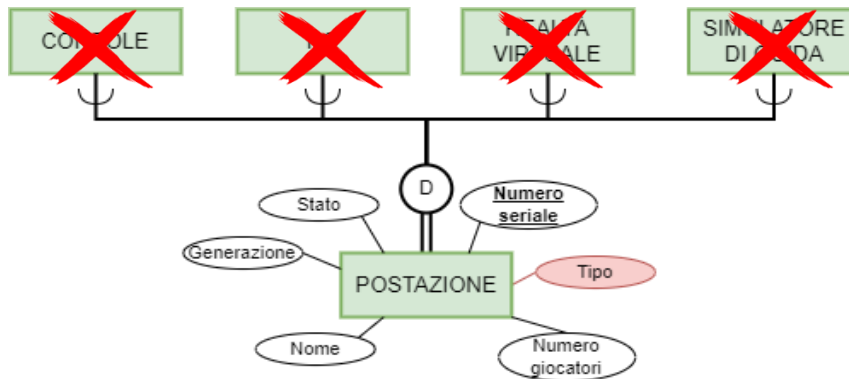
Totale accessi con ridondanza	Totale accessi senza ridondanza
$102420/aa + 2kb$	$101280/aa$

Dato l'assente spreco di memoria e il minor numero di accessi conviene eliminare il dato ridondante:

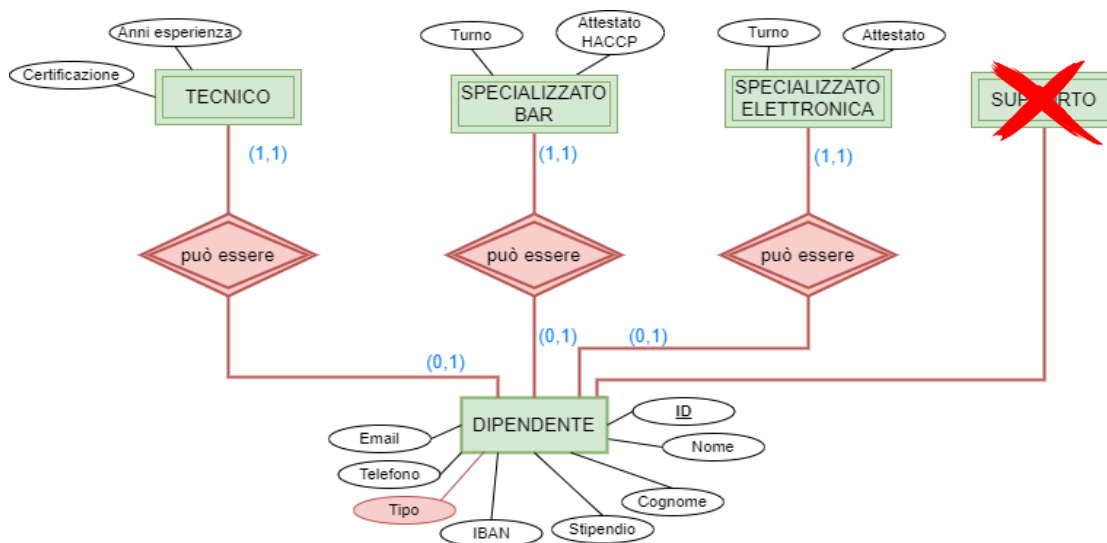


-Eliminazione delle gerarchie-

Nello schema EER presentato in precedenza è presente la seguente specializzazione. Data l'assenza di attributi per le entità figlie e l'assenza di relazioni che coinvolgano un'entità figlia singolarmente, conviene accorpare le entità figlie nel padre e aggiungere l'attributo Tipo all'entità Postazione.

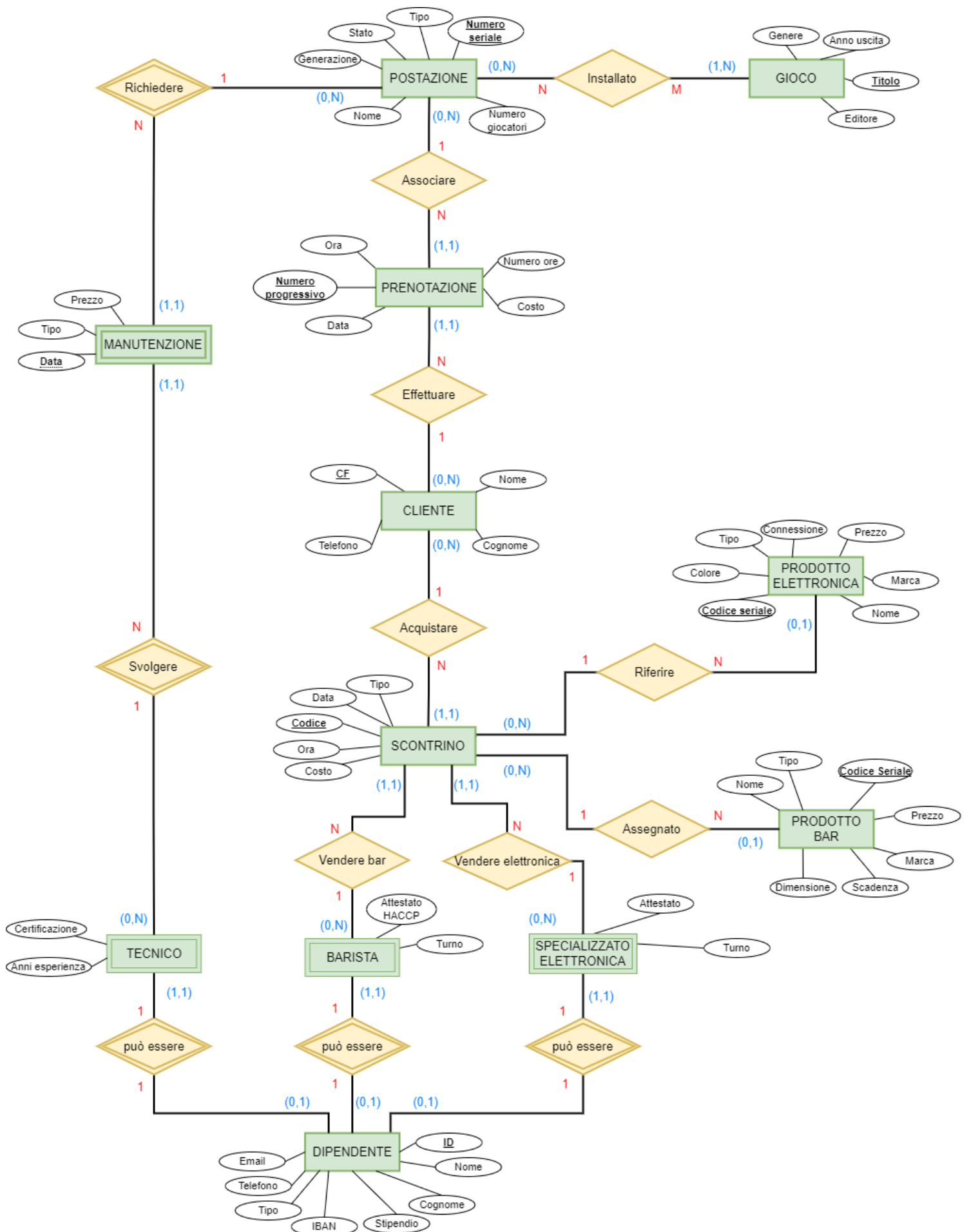


Diverso è invece il caso della specializzazione di Dipendente, dove le entità figlie possiedono dei propri attributi e sono coinvolte singolarmente in delle associazioni, ad eccezione dell'entità Supporto. Conviene quindi adottare una soluzione ibrida accorpare l'entità Supporto nel padre e creare invece tre nuove relazioni per Tecnico, Barista e Specializzato Elettronica. Verrà inoltre aggiunto, come per la precedente specializzazione, l'attributo Tipo all'entità Dipendente.



-Schema EER ristrutturato-

Al termine della fase di ristrutturazione ne deriva il seguente schema EER:



-Schema relazionale-

Dipendente(ID, Nome, Cognome, Stipendio, IBAN, Telefono, Email, Tipo)
Tecnico(Dipendente.ID↑, Anni esperienza, Certificazione)
Barista(Dipendente.ID↑, Attestato HACCP, Turno)
Specializzato elettronica(Dipendente.ID↑, Attestato, Turno)
Cliente(CF, Nome, Cognome, Telefono)
Scontrino (Codice, Data, Ora, Costo, Tipo, Cliente.CF↑)
Prodotto bar(Codice seriale, Nome, Marca, Prezzo, Tipo, Dimensione, Scadenza)
Prodotto elettronica(Codice seriale, Nome, Marca, Prezzo, Tipo, Colore, Connessione)
Prenotazione(Numero Progressivo, Data, Ora, Costo, Numero ore, Postazione.Numero seriale↑, Cliente.CF↑)
Postazione(Numero seriale, Stato, Generazione, Numero giocatori, Nome, Tipo)
Manutenzione(Data, Postazione.Numero seriale↑, Tecnico. Dipendente.ID↑, Tipo, Prezzo)
Gioco(Titolo, Editore, Genere, Anno uscita)
Installato(Postazione.Numero seriale↑, Gioco.Titolo↑)
Acquisto bar(Prodotto bar.Codice seriale↑, Scontrino.Codice↑)
Acquisto elettronica(Prodotto elettronica.Codice seriale↑, Scontrino.Codice↑)
Vendere bar(Barista.Dipendente.ID↑, Scontrino.Codice↑)
Vendere elettronica(Specializzato elettronica.Dipendente.ID↑, Scontrino.Codice↑)

-Normalizzazione-

Il database si presenta già normalizzato. È infatti in prima forma normale in quanto tutti gli attributi sono atomici. È in seconda forma normale perché, oltre ad essere già in 1NF, tutti gli attributi non primi sono pienamente dipendenti funzionalmente. È in terza forma normale perché, oltre ad essere già in 2NF, in tutte le tabelle non sono presenti dipendenze transitive fra attributi non primi e la chiave primaria.

5. Realizzazione della base di dati con MySQL

- `CREATE SCHEMA `game_lab`;`
- `USE `game_lab`;`
- `CREATE TABLE `dipendente` (
 `ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
 `nome` VARCHAR(30) NOT NULL,
 `cognome` VARCHAR(30) NOT NULL,
 `stipendio` DECIMAL(7,2),
 `IBAN` CHAR(27) NOT NULL,
 `telefono` VARCHAR(12) NOT NULL,
 `email` VARCHAR(319) NOT NULL,
 `tipo` ENUM('tecnico','barista','specializzato elettronico','supporto') NOT NULL,
 PRIMARY KEY (`ID`)
);`
- `CREATE TABLE `tecnico` (
 `ID` INT UNSIGNED NOT NULL,
 `anni_esperienza` TINYINT,
 `certificazione` VARCHAR(80) NOT NULL,
 PRIMARY KEY(`ID`),
 FOREIGN KEY(`ID`) REFERENCES dipendente(`ID`) ON UPDATE CASCADE ON DELETE CASCADE
);`
- `CREATE TABLE `barista` (
 `ID` INT UNSIGNED NOT NULL,
 `attestato_HACCP` VARCHAR(50) NOT NULL,
 `turno` VARCHAR(30) NOT NULL,
 PRIMARY KEY(`ID`),
 FOREIGN KEY(`ID`) REFERENCES dipendente(`ID`) ON UPDATE CASCADE ON DELETE CASCADE
);`
- `CREATE TABLE `specializzato_elettronica` (
 `ID` INT UNSIGNED NOT NULL,
 `attestato` VARCHAR(50) NOT NULL,
 `turno` VARCHAR(30) NOT NULL,
 PRIMARY KEY(`ID`),
 FOREIGN KEY(`ID`) REFERENCES dipendente(`ID`) ON UPDATE CASCADE ON DELETE CASCADE
);`
- `CREATE TABLE `cliente` (
 `CF` CHAR(16) NOT NULL,
 `nome` VARCHAR(30) NOT NULL,
 `cognome` VARCHAR(30) NOT NULL,
 `telefono` VARCHAR(12) NOT NULL,
 PRIMARY KEY(`CF`)
);`
- `CREATE TABLE `vendere_elettronica` (
 `ID` INT UNSIGNED NOT NULL,
 `codice` CHAR(8) NOT NULL,
 PRIMARY KEY(`ID`,`codice`),
 FOREIGN KEY(`ID`) REFERENCES specializzato_elettronica(`ID`) ON UPDATE CASCADE ON DELETE CASCADE,
 FOREIGN KEY(`codice`) REFERENCES scontrino(`codice`) ON UPDATE CASCADE ON DELETE CASCADE
);`

- ```
CREATE TABLE `prodotto_bar` (
 `codice_seriale` VARCHAR(22) NOT NULL,
 `nome` VARCHAR(30) NOT NULL,
 `marca` VARCHAR(30) NOT NULL,
 `prezzo` DECIMAL(7,2) NOT NULL,
 `tipo` VARCHAR(30) NOT NULL,
 `scadenza` DATE NOT NULL,
 `dimensione` ENUM('piccolo', 'medio', 'grande'),
 PRIMARY KEY(`codice_seriale`)
);
```
- ```
CREATE TABLE `prodotto_elettronica` (
  `codice_seriale` VARCHAR(22) NOT NULL,
  `nome` VARCHAR(30) NOT NULL,
  `marca` VARCHAR(30) NOT NULL,
  `prezzo` DECIMAL(7,2) NOT NULL,
  `tipo` VARCHAR(30) NOT NULL,
  `colore` VARCHAR(25),
  `connessione` VARCHAR(30),
  PRIMARY KEY(`codice_seriale`)
);
```
- ```
CREATE TABLE `scontrino` (
 `codice` CHAR(8) NOT NULL,
 `data` DATE NOT NULL,
 `ora` TIME NOT NULL,
 `costo` DECIMAL(7,2) NOT NULL,
 `tipo` ENUM('bar', 'elettronica') NOT NULL,
 `CF` CHAR(16) NOT NULL,
 PRIMARY KEY(`codice`),
 FOREIGN KEY(`CF`) REFERENCES cliente(`CF`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `postazione` (
  `numero_seriale` CHAR(12) NOT NULL,
  `stato` VARCHAR(30) NOT NULL,
  `generazione` TINYINT NOT NULL,
  `nome` VARCHAR(30) NOT NULL,
  `tipo` ENUM('PC', 'realità virtuale', 'simulatore di guida', 'console') NOT NULL,
  `numero_giocatori` ENUM('1', '2', '3', '4', '5', '6') NOT NULL,
  PRIMARY KEY(`numero_seriale`)
);
```
- ```
CREATE TABLE `gioco` (
 `titolo` VARCHAR(50) NOT NULL,
 `editore` VARCHAR(30) NOT NULL,
 `genere` VARCHAR(30) NOT NULL,
 `anno_uscita` DATE NOT NULL,
 PRIMARY KEY(`titolo`)
);
```



- ```
CREATE TABLE `prenotazione` (
    `numero_progressivo` INT UNSIGNED NOT NULL AUTO_INCREMENT,
    `data` DATE NOT NULL,
    `ora` TIME NOT NULL,
    `costo` DECIMAL(7,2) NOT NULL,
    `numero_ore` TINYINT NOT NULL,
    `CF` CHAR(16) NOT NULL,
    `numero_seriale` CHAR(12) NOT NULL,
    PRIMARY KEY(`numero_progressivo`),
    FOREIGN KEY(`CF`) REFERENCES cliente(`CF`) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY(`numero_seriale`) REFERENCES postazione(`numero_seriale`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `manutenzione` (
 `data` DATE NOT NULL,
 `numero_seriale` CHAR(12) NOT NULL,
 `ID` INT UNSIGNED NOT NULL,
 `tipo` VARCHAR(30) NOT NULL,
 `prezzo` DECIMAL(7,2) NOT NULL,
 PRIMARY KEY(`data`,`numero_seriale`,`ID`),
 FOREIGN KEY(`numero_seriale`) REFERENCES postazione(`numero_seriale`) ON UPDATE CASCADE ON DELETE CASCADE,
 FOREIGN KEY(`ID`) REFERENCES tecnico(`ID`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `installato` (
    `numero_seriale` CHAR(12) NOT NULL,
    `titolo` VARCHAR(50) NOT NULL,
    PRIMARY KEY(`numero_seriale`,`titolo`),
    FOREIGN KEY(`numero_seriale`) REFERENCES postazione(`numero_seriale`) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY(`titolo`) REFERENCES gioco(`titolo`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `acquisto_bar` (
 `codice_seriale` VARCHAR(22) NOT NULL,
 `codice` CHAR(8) NOT NULL,
 PRIMARY KEY(`codice_seriale`,`codice`),
 FOREIGN KEY(`codice_seriale`) REFERENCES prodotto_bar(`codice_seriale`) ON UPDATE CASCADE ON DELETE CASCADE,
 FOREIGN KEY(`codice`) REFERENCES scontrino(`codice`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `acquisto_elettronica` (
    `codice_seriale` VARCHAR(22) NOT NULL,
    `codice` CHAR(8) NOT NULL,
    PRIMARY KEY(`codice_seriale`,`codice`),
    FOREIGN KEY(`codice_seriale`) REFERENCES prodotto_elettronica(`codice_seriale`) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY(`codice`) REFERENCES scontrino(`codice`) ON UPDATE CASCADE ON DELETE CASCADE
);
```
- ```
CREATE TABLE `vendere_bar` (
 `ID` INT UNSIGNED NOT NULL,
 `codice` CHAR(8) NOT NULL,
 PRIMARY KEY(`ID`,`codice`),
 FOREIGN KEY(`ID`) REFERENCES barista(`ID`) ON UPDATE CASCADE ON DELETE CASCADE,
 FOREIGN KEY(`codice`) REFERENCES scontrino(`codice`) ON UPDATE CASCADE ON DELETE CASCADE
);
```

## 6. Implementazione query SQL

- **Operazione 1** (Inserire un nuovo cliente):

```
INSERT INTO cliente (CF, nome, cognome, telefono)
VALUES (
 ?, ?, ?, ?
);
```

- **Operazione 2** (Inserire un nuovo prodotto bar):

```
INSERT INTO prodotto_bar(codice_seriale, nome, marca, prezzo, tipo, scadenza, dimensione)
VALUES (
 ?, ?, ?, ?, ?, ?, ?
);
```

- **Operazione 3** (Inserire un nuovo prodotto di elettronica):

```
INSERT INTO prodotto_elettronica(codice_seriale, nome, marca, prezzo, tipo, colore, connessione)
VALUES (
 ?, ?, ?, ?, ?, ?, ?
);
```

- **Operazione 4** (Inserire una nuova postazione):

```
INSERT INTO postazione(numero_seriale, stato, generazione, nome, tipo, numero_giocatori)
VALUES (
 ?, ?, ?, ?, ?, ?
);
```

- **Operazione 5** (Inserire un nuovo dipendente):

```
INSERT INTO dipendente(ID, nome, cognome, stipendio, IBAN, telefono, email, tipo)
VALUES (
 ?, ?, ?, ?, ?, ?, ?, ?
);
```

In base alla scelta dell'utente verranno effettuati gli inserimenti nelle tabelle tecnico, barista o specializzato\_elettronica:

```
INSERT INTO tecnico(ID, anni_esperienza, certificazione)
VALUES (
 ?, ?, ?
);
```

```
INSERT INTO barista(ID, attestato_HACCP, turno)
VALUES (
 ?, ?, ?
);
```

```
INSERT INTO specializzato_elettronica(ID, attestato, turno)
VALUES (
 ?, ?, ?
);
```

- **Operazione 6** (Effettuare una prenotazione):

```
INSERT INTO prenotazione(numero_progressivo, data, ora, costo, numero_ore, CF, numero_seriale)
VALUES (
 ?, ?, ?, ?, ?, ?, ?
);
```

- **Operazione 7** (Effettuare un acquisto al bar):

Creo l'istanza di scontrino:

```
INSERT INTO scontrino(codice, data, ora, tipo, CF, costo)
VALUES (
 ?, ?, ?, ?, ?, 0
);
```

Inserisco tutti i prodotti acquistati:

```
INSERT INTO acquisto_bar(codice_seriale,codice)
VALUES (
 ?, ?
);
```

Calcolo il costo totale dell'acquisto e inserisco il costo nell'istanza di scontrino:

```
UPDATE scontrino
SET costo = (
 SELECT SUM(prezzo) AS costo
 FROM acquisto_bar AS a, prodotto_bar AS p
 WHERE a.codice_seriale = p.codice_seriale AND a.codice = ?
)
WHERE codice = ?;
```

Infine collego lo scontrino al dipendente che ha venduto i prodotti:

```
INSERT INTO vendere_bar(ID,codice)
VALUES (
 ?, ?
);
```

- **Operazione 8** (Effettuare un acquisto di elettronica):

Creo l'istanza di scontrino:

```
INSERT INTO scontrino(codice, data, ora, tipo, CF, costo)
VALUES (
 ?, ?, ?, ?, ?, 0
);
```

Inserisco tutti i prodotti acquistati:

```
INSERT INTO acquisto_elettronica(codice_seriale,codice)
VALUES (
 ?, ?
);
```

Calcolo il costo totale dell'acquisto e inserisco il costo nell'istanza di scontrino:

```
UPDATE scontrino
SET costo = (
 SELECT SUM(prezzo) AS costo
 FROM acquisto_elettronica AS a, prodotto_bar AS p
 WHERE a.codice_seriale = p.codice_seriale AND a.codice = ?
)
WHERE codice = ?;
```

Infine collego lo scontrino al dipendente che ha venduto i prodotti:

```
INSERT INTO vendere_elettronica(ID,codice)
VALUES (
 ?, ?
);
```

- **Operazione 9** (Richiedere una manutenzione per una postazione):

```
INSERT INTO manutenzione(data, numero_seriale, ID, tipo, prezzo)
VALUES (
 ?, ?, ?, ?, ?
);
```

A seconda dell'esito della manutenzione verrà aggiornato lo stato della postazione:

```
UPDATE postazione SET stato = '???' WHERE numero_seriale = ?
```

- **Operazione 10** (Installare un gioco su una postazione):

```
INSERT INTO installato(numero_seriale, titolo)
VALUES (
 ?, ?
);
```

Se il gioco non è ancora presente nel database verrà effettuato anche un inserimento nella tabella gioco:

```
INSERT INTO gioco(titolo, editore, genere, anno_uscita)
VALUES (
 ?, ?, ?, ?
);
```

- **Operazione 11** (Stampare numero seriale, nome, tipo e stato di ogni postazione, compreso il numero di prenotazioni):

```
SELECT po.numero_seriale, po.nome, po.tipo, po.stato, COUNT(*) AS numero_prenotazioni
FROM postazione AS po, prenotazione AS pr
WHERE po.numero_seriale = pr.numero_seriale
GROUP BY po.numero_seriale
```

- **Operazione 12** (Stampare nome, cognome, numero di telefono ed email del tecnico con più anni di esperienza):

```
SELECT d.nome, d.cognome, d.telefono, d.email
FROM tecnico AS t, dipendente AS d
WHERE t.ID = d.ID AND t.anni_esperienza = (
 SELECT MAX(anni_esperienza)
 FROM tecnico
);
```

- **Operazione 13** (Stampare i dati del cliente con maggior numero di prenotazioni):

```
SELECT c.CF, c.nome, c.cognome, c.telefono, COUNT(*) AS numero_prenotazioni
FROM cliente AS c, prenotazione AS p
WHERE c.CF = p.CF
GROUP BY c.CF, c.nome, c.cognome, c.telefono
ORDER BY numero_prenotazioni DESC
LIMIT 1;
```

- **Operazione 14** (Stampare su quali postazioni è installato un determinato gioco):

```
SELECT p.numero_seriale, p.nome, p.stato, p.numero_giocatori
FROM postazione AS p, installato AS i, gioco AS g
WHERE p.numero_seriale = i.numero_seriale AND g.titolo = i.titolo AND g.titolo = '?'
```

- **Operazione 15** (Stampare i dati del cliente con maggior numero di ordini al bar e acquisti di elettronica, che non abbia effettuato nessuna prenotazione):

```
SELECT c.CF, c.nome, c.cognome, c.telefono, COUNT(*) AS numero_acquisti
FROM cliente AS c, scontrino AS s
WHERE c.CF = s.CF AND c.CF NOT IN (
 SELECT p.CF
 FROM prenotazione AS p
)
GROUP BY c.CF, c.nome, c.cognome, c.telefono
ORDER BY numero_acquisti DESC
LIMIT 1;
```

- **Operazione 16** (Stampare i dati del cliente con la spesa maggiore in prenotazioni):

```
SELECT c.CF, c.nome, c.cognome, c.telefono, SUM(costo) AS spesa
FROM cliente AS c, prenotazione AS p
WHERE c.CF = p.CF
GROUP BY c.CF, c.nome, c.cognome, c.telefono
ORDER BY spesa DESC
LIMIT 1;
```

- **Operazione 17** (Stampare la postazione maggiormente prenotata per un certo numero di giocatori):

```
SELECT po.numero_seriale, COUNT(*) AS max_prenotata
FROM postazione AS po, prenotazione AS pr
WHERE po.numero_seriale = pr.numero_seriale AND po.numero_giocatori = '?' AND po.stato = 'abilitata'
GROUP BY numero_seriale
HAVING max_prenotata > 0
ORDER BY max_prenotata DESC
LIMIT 1;
```

- **Operazione 18** (Stampare i dati del cliente che ha speso di più):

```
SELECT cliente.CF, SUM(ts.spesa) AS spesa_totale
FROM cliente, (
 (SELECT c.CF, SUM(p.costo) AS spesa
 FROM cliente AS c, prenotazione AS p
 WHERE c.CF = P.CF
 GROUP BY c.CF
 ORDER BY spesa DESC
 LIMIT 1)
 UNION ALL
 (SELECT C.CF, SUM(s.costo) AS spesa
 FROM cliente AS C, scontrino AS s
 WHERE C.CF = s.CF
 GROUP BY C.CF
 ORDER BY spesa DESC
 LIMIT 1)
) AS ts
GROUP BY cliente.CF
ORDER BY spesa_totale DESC
LIMIT 1;
```