**Abstract**

Un gestore di case da gioco vuole centralizzare e modernizzare le informazioni riguardanti le proprie strutture gestite, in modo da poter compiere operazioni di statistica fiscale, gestire le risorse umane e la manutenzione delle strutture (cazzata non sapevo che scrivere).  
Viene dunque sviluppata una base di dati nella quale raccogliere le informazioni riguardanti il personale, gli ambienti presenti nel casinò, i tipi di gioco presenti e le transazioni effettuate in gioco (vincite perdite).

**Analisi dei requisiti**

Il progetto mira a generare una base di dati, che rappresenti il gestionale di uno o più casino.

Ogni **casinò** è caratterizzato da: il proprio indirizzo, un nome, del personale, delle sale e dei clienti.

Il **Personale** rappresenta i lavoratori del casinò (croupier, cassieri, sicurezza, dirigenti, baristi), ogni persona viene identificata con codice fiscale, nome, cognome, ruolo.  
Inoltre viene salvato l’indirizzo di ogni dipendente.

Le **Sale** rappresentano i vari ambienti del casinò, essi possono essere divisi in 3 tipi: sale gioco(gambling), sale ristoro e sale congressi.  
Ogni sala ha un proprio nome, un numero, e la posizione all’interno del casinò.

All’interno delle sale gambling sono presenti una o più **postazioni** da gioco, ogni postazione ha un numero identificativo ed un Nome.  
Ogni **postazione** ha un proprio tipo di gioco.  
  
Le varie tipologie di **Gioco** sono caratterizzate dal nome, la descrizione, le regole e il tipo.  
  
Ogni postazione registra le **transazioni** effettuate (vincite/perdite).  
Una transazione viene individuata da un id, il timestamp

(data e ora), l’importo e il cliente che l’ha effettuata e il tipo.

Vengono inoltre registrati tutti i **Clienti**, di ogni cliente si registra il CF, Nome, Cognome, un contatto, l’indirizzo e il saldo dei crediti di gioco.

**Glossario dei termini**

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Collegamenti |
| Casino | una struttura di gioco, identificata da un id univoco | Personale, Sale, Clienti |
| Personale | Persona che lavora all’interno della struttura. (Dirigenti, croupier, barman, sicurezza) | Casino, Sale, Postazione |
| Gambling | Stanza/ambiente nella quale sono posizionate le postazioni da gioco | Casino, Postazione, Personale |
| Ristoro | Stanza/Ambiente dedicata al bar/ristorante | Casino, Personale |
| Congressi | Stanza/Ambiente dedicata ad eventi e presentazioni | Casino, Personale |
| Postazione | Descrive le postazioni di gioco presenti all’interno di una sala, come tavoli, giochi elettronici | Gambling, Personale, Gioco, Transazione |
| Gioco | Descrive il gioco giocato nella postazione | Postazione |
| Ricarica | Transazione rappresentante la ricarica del saldo di gioco | Postazione, Clienti |
| Perdita | Transazione di gioco perdente | Postazione, Clienti |
| Vincita | Transazione di gioco vincente | Postazione, Clienti |
| Clienti | Rappresenta tutta la clientela, dei casinò | Casino, Transazione |
| Torneo | Rappresenta un torneo di un gioco | Match, Gioco |
| Match | Rappresenta i vari round del torneo | Clienti, Torneo |

**Operazioni tipiche**

Le seguenti operazioni vengono calcolate in una settimana regolare di attività.

|  |  |
| --- | --- |
| 1. Inserimento Clienti | 200 al giorno |
| 2. Controllo turni di lavoro | 50 al giorno |
| 3. Controllo saldo clienti | 300 al giorno |
| 4. Controllo vincite giornaliero | 1 al giorno |
| 5. Inserimento turni di lavoro | 200 settimanali |
| 6. Inserimento transazioni | 1000 giornaliere |

Le seguenti operazioni vengono invece distribuite durante l’anno:

|  |  |
| --- | --- |
| 1. Inserimento Sale | 10 alla creazione |
| 2. Inserimento Postazioni | 10 all’anno |
| 3. Inserimento Giochi | 5 all’anno |
| 4. Inserimento Personale | 20 all’anno |
| 5. Inserimento tornei | 15/20 anno |

**Progettazione Concettuale**

Lista entità

|  |  |  |
| --- | --- | --- |
| **Casinò:** una struttura adibita al gioco | | |
| IDCasinò | int primary key | Codice univoco del casinò |
| Nome | varchar(40) | Nome della struttura |
| Indirizzo | Attributo composto |  |
| Via | varchar(30) |  |
| CAP | varchar(6) |  |
| Provincia | varchar(2) |  |
| Stato | varchar(2) |  |

|  |  |  |
| --- | --- | --- |
| **Personale:** lavoratore della struttura | | |
| CF | varchar(16) | Codice univoco |
| Nome | varchar(20) |  |
| Cognome | varchar(20) |  |
| Contatto | varchar(20) | Contatto telefonico |
| Ruolo | varchar(15) | Ruolo ricoperto all’interno della struttura |
| Indirizzo | Attributo composto |  |
| Via | varchar(20) |  |
| CAP | int |  |
| Provincia | varchar(2) |  |
| Stato | varchar(2) |  |

|  |  |  |
| --- | --- | --- |
| **Sale:** stanze della strutturaz | | |
| Numero | int primary key | Numero identificativo della stanza |
| Nome | varchar(20) primary key | Nome della sala |
| Posizione | varchar(20) | Posizione all’interno del casinò |
| Tipo | Tipo\_sale |  |

**Sale** si specializza in **Gambling, Ristoro** e **Congressi**

|  |
| --- |
| **Gambling:** sale dedicate al gioco |

|  |
| --- |
| **Congressi:** sale dedicate ad eventi e congressi |

|  |
| --- |
| **Ristoro:** sale dedicate al consumo di alimenti |

|  |  |  |
| --- | --- | --- |
| **Postazioni:** postazioni di gioco | | |
| Numero | int primary key | Numero identificativo della postazione |
| Nome | varchar(15) | Nome della postazione |

|  |  |  |
| --- | --- | --- |
| **Gioco:** gioco praticato nella postazione | | |
| Tipo | varchar(10) primary key | Tipologia di gioco |
| Nome | varchar(10) primary key | Nome della versione del gioco |
| Descrizione | varchar(50) | Breve descrizione del gioco |
| Regole | varchar(100) | Regole di gioco |

|  |  |  |
| --- | --- | --- |
| **Transazioni:** vincita o perdita al gioco | | |
| IDTx | integer primary key | Numero transazione |
| Tipo | tipo\_tx | Indica se si tratta di vincita o perdita |
| TimestampTX | timestamp | Tempo di esecuzione della tx |
| Importo | integer |  |

**Transazioni** si divide in **Vincita, Perdita** e **Ricarica**

|  |
| --- |
| **Vincita:** transazione rappresentante una vittoria ad un gioco |

|  |
| --- |
| **Perdita:** transazione rappresentante una perdita ad un gioco |

|  |
| --- |
| **Ricarica:** transazione rappresentante una ricarica del saldo di gioco |

|  |  |  |
| --- | --- | --- |
| **Clienti:** clienti della struttura | | |
| CF | varchar(16) | Codice univoco che rappresenta la persona |
| Nome | varchar(20) |  |
| Cognome | varchar(20) |  |
| Contatto | varchar(20) |  |
| Saldo | integer | Saldo |
| Indirizzo | Attributo composto |  |
| Via | varchar(20) |  |
| CAP | int |  |
| Provincia | varchar(2) |  |
| Stato | varchar(2) |  |

|  |  |  |
| --- | --- | --- |
| **Tornei:** | | |
| IDTorneo | Integer primary key | Codice univoco che rappresenta il torneo |
| Nome | Varchar(20) | Nome del torneo |
| DataTorneo | DATE |  |
| BuyIN | integer | Quota d’entrata |
| Premio | integer | Premio vincitore |

|  |  |  |
| --- | --- | --- |
| **Match:** | | |
| IDMatch | Integer primary key | Codice univoco che rappresenta il match |
| IsFinal | boolean | Indica l’ultimo match del torneo |
| Ora | TIME |  |

Lista relazioni

Casino-Sale: **HA**

* Un casinò può avere una o più sale (1, N)
* Una sala può avere un solo casinò (1,1)

Casino-Personale: **DIPENDE**

* Un casinò può avere più dipendenti (1, N)
* Un dipendente può lavorare ad un solo casinò (1,1)

Casino-Clienti: **SPENDE**

* Un casinò può avere 0 o più clienti (0, N)
* Un cliente può essere associato a più casinò (1, N)

Personale-Postazioni: **LAVORA**

* Un dipendente può lavorare a più postazioni non contemporaneamente (1, N)
* Una postazione può avere un solo lavoratore in un determinato momento (1,1)

Personale-Sale: **LAVORA**

* Un dipendente può essere associato a più sale (1, N)
* Una sala può avere più dipendenti assegnati (1, N)

Sale-Postazioni: **GIOCA**

* Una sala può avere più postazioni (1, N)
* Una postazione può avere una sola sala assegnata (1,1)

Sale-Tornei: **SVOLTO**

* Una sala può avere un solo torneo contemporaneamente (1, 1)
* Un torneo può essere eseguito in una sola sala (1,1)

Postazioni-Transazioni: **TRANSA**

* Una postazione può essere associata a 0 o più transazioni (0, N)
* Una transazione può avere una sola postazione associata (1,1)

Postazioni-Gioco: **CARATERIZZATO**

* Una postazione può avere un solo gioco assegnato (1, 1)
* Un gioco può essere associato a nessuna postazione o più (0, N)

Gioco-Tornei: **TIPO DI GIOCO**

* Un gioco può avere nessuno o più tornei (0, N)
* Un torneo può avere un solo tipo di gioco (1,1)

Tornei-Match: **CLASSIFICA**

* Un torneo può avere uno o più match (1, N)
* Un match può avere un solo torneo associato (1,1)

Match-Clienti: **DISPUTANO**

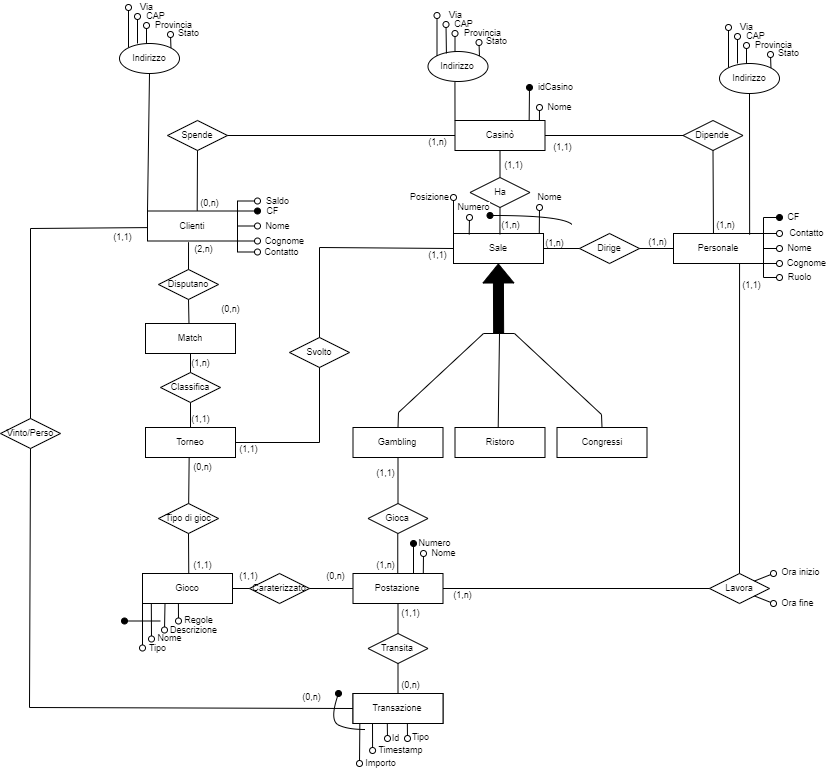
* Un match deve avere 2 o più clienti (2, N)
* Un cliente può avere uno o più match (0,1)

Clienti-Transazioni: **VINTO/PERSO**

* Un cliente può avere nessuna o più transazioni (0, N)
* Una transazione può avere un solo cliente associato (1,1)

Clienti-Tornei: **VINCE**

* Un cliente può vincere nessuno o più tornei (0, N)
* Un torneo può avere solo un vincitore (1,1)



Progettazione Logica

L’attributo “Ruolo” (di personale) e “Tipo” (di transazione) vengono trasformati in tipi custom.  
L’attributo “Regole” (di gioco) viene trasformato in una nuova entità e collegato all’entità tramite una relazione molti a molti.

Analisi delle ridondanze  
L’attributo “Saldo” (di cliente) potrebbe essere considerata una ridondanza in quanto potremmo calcolare facilmente il saldo di ogni cliente tramite le transazioni associate al cliente, ma questo causerebbe un forte peso sull’operazione: *Controllo saldo clienti. 300 al giorno*

|  |  |  |
| --- | --- | --- |
| Tavola dei volumi | | |
| **Concetto** | **Costrutto** | **Volume** |
| Vinto/Perso | R |  |
| Transazione | E |  |

Se il saldo non venisse salvato per ogni cliente la base di dati si troverebbe a dover gestire 300\* (numero di transazioni di ogni cliente) operazioni al giorno sulla tabella Transazione.  
Essendo il campo saldo di tipo integer avremo un dispendio maggiore di memoria di 4 byte per ogni cliente, ipotizzando un casinò con 100000 clienti ci ritroveremo con una base di dati con 400Kb in più di dati.

La relazione “Vince” (tra Tornei e Clienti), può essere rimossa in quanto il cliente vincitore del torneo può essere facilmente recuperato controllando l’utente con saldo diverso da 0 nell’ultimo match del torneo.

Eliminazione delle generalizzazioni:

Per entrambe le generalizzazioni è stato scelto di accorpare le tre entità figlie nella genitrice.  
In questo modo potremmo risparmiare 2 tabelle e i rispettivi identificatori, andando ad inserire un attributo per distinguere i 3 tipi.

Partizionamento / Accorpamento di entità e associazioni

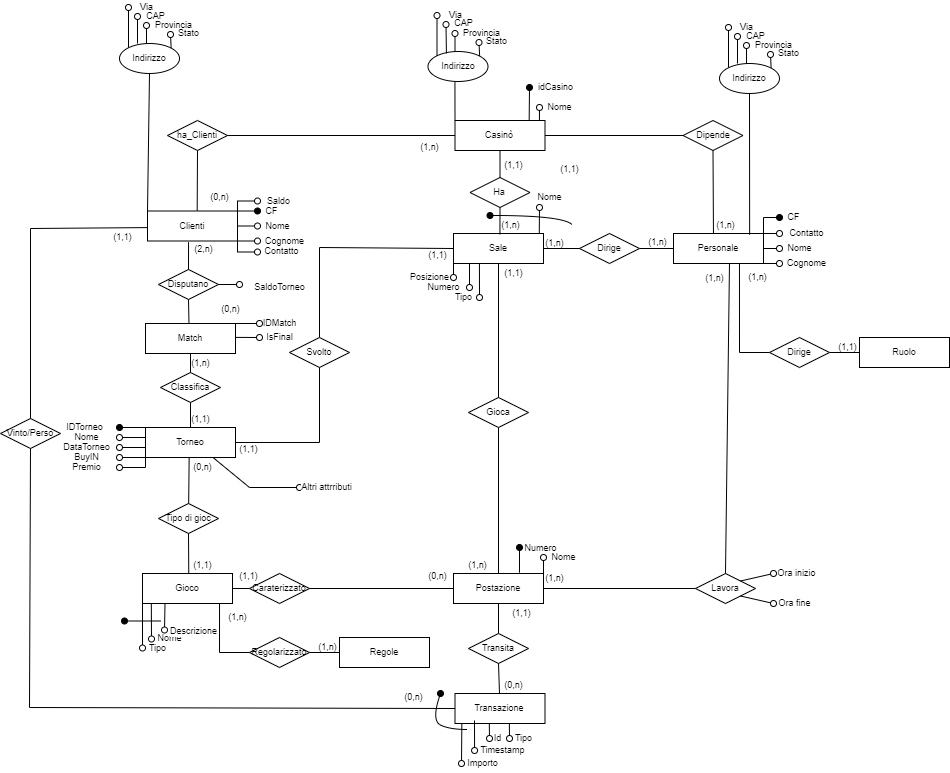
La scelta è quella di accorpare in un’unica entità i tipi di sale

Scelta degli identificatori principali:

Per le tabelle “Clienti” e “Personale” è stato scelto di utilizzare un id univoco di tipo integer, nonostante avessimo già il campo “CF” univoco per ogni occorrenza delle tabelle.  
Se avremo utilizzato il campo “CF” come chiave primaria, ci saremo ritrovati con il dato replicato in ogni tabella che lo utilizzava come chiave esterna, replicando dunque 16 Char invece di un semplice intero.

Per la tabella “Sale” è stato scelto di utilizzare una chiave primaria composta dal numero di sala e l’id del casinò a cui essa appartiene.  
Il numero sala non sarebbe bastato per rendere univoca l’entità assieme alle sale degli altri casinò, perciò si è deciso di inserire anche l’IDCasinò.  
In questo modo la base di dati è stata leggermente appesentita dovendo referenziare anche questo dato.  
  
Per l’entità “Gioco” si è scelto di inserire un Id integer univoco, per facilitare la referenziazione di essa nelle tabelle in cui è impiegata.

Diagramma E-R ristrutturato:



Schema relazionale e vincoli di integrità referenziale

Gli attributi con l’asterisco possono avere valori nulli, in seguito ad ogni relazione sono riportate le chiavi esterne e i relativi attributi coinvolti

**Casinò** (IDCasinò, Nome, Via, CAP, Provincia, Stato)

**Ruoli** (IDRuolo, Nome, Descrizione, Salario)

**Personale** (IDPersonale, CF, Nome, Cognome, Contatto, Via, CAP, Provincia, Stato, IDRuolo, IDCasinò)

* Personale.IDRuolo -> Ruoli.IDRuolo
* Personale.IDCasinò -> Casinò.IDCasinò

**Sale** (Numero, Nome, Posizione, Tipo, IDCasinò)

* Sale.IDCasinò -> Casinò.IDCasinò

**Dirige** (IDSala, IDPersonale)

* Dirige.IDSala -> Sale.Numero
* Dirige.IDPersonale -> Personale.IDPersonale

**Giochi** (IDGioco, Tipo, Nome, Descrizione)

**Postazioni** (Numero, Nome, NomeSala, NumeroSala, IDGioco)

* Postazioni.NomeSala -> Sale.Nome
* Postazioni.NumeroSala -> Sale.Numero
* Postazioni.IDGioco -> Giochi.IDGioco

**Lavora** (NumeroPs, IDPersonale, Ora\_Inizio, Ora\_Fine)

* Lavora.NumeroPS -> Postazioni.Numero
* Lavora.IDPersonale -> Personale.IDPersonale

**Clienti** (IDCliente, CF, Nome, Cognome, Contatto, Via, CAP, Provincia, Stato, Saldo)

**ha\_clienti** (IDCliente, IDCasinò)

* Ha\_clienti.IDCliente -> Clienti.IDCliente)
* Ha\_clienti.IDCasinò -> Clienti.IDCasinò)

**Tornei** (IDTorneo, Nome, DataTorneo, BuyIN, Premio, NomeSala, NumeroSala, IDGioco)

* Tornei.NomeSala -> Sale.Nome
* Tornei.NumeroSala -> Sale.Numero
* Tornei.IDGioco -> Giochi.IDGioco

**Match** (IDMatch, IsFinal, Ora, IDTorneo)

* Match.IDTorneo -> Tornei.IDTorneo

**Disputano** (IDMatch, IDCliente, SaldoMano)

* Disputano.IDMatch -> Match.IDMatch
* Disputano.IDCliente -> Clienti.IDCliente

**Regole** (IDRegola, Nome, Descrizione, Penalità)

**Regolamento** (IDGioco, IDRegola)

* Regolamento.IDGioco -> Giochi.IDGioco
* Regolamento.IDRegola -> Regole.IDRegola

**Transazioni** (IDTx, Tipo, TimestampTX, Importo, IDPostazione, IDCliente)

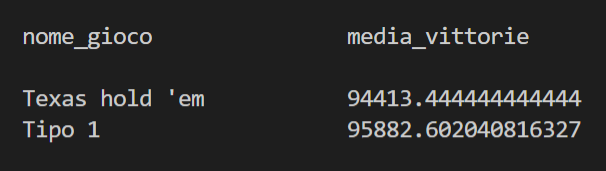
* Transazioni.IDPostazione -> Postazioni.Numero
* Transazioni.IDCliente -> Clienti.IDCliente

Query e Indice

1. Mostrare i giochi con un payout superiore alla media

Immagine che contiene testo, screenshot, schermo

Descrizione generata automaticamente



1. Mostrare le statistiche dei tavoli di un casinò

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

1. Mostrare il cliente che ha vinto la maggior somma di denaro ad un tipo di gioco

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo, schermo, screenshot

Descrizione generata automaticamente

1. Mostrare il vincitore di un torneo

Immagine che contiene testo

Descrizione generata automaticamente

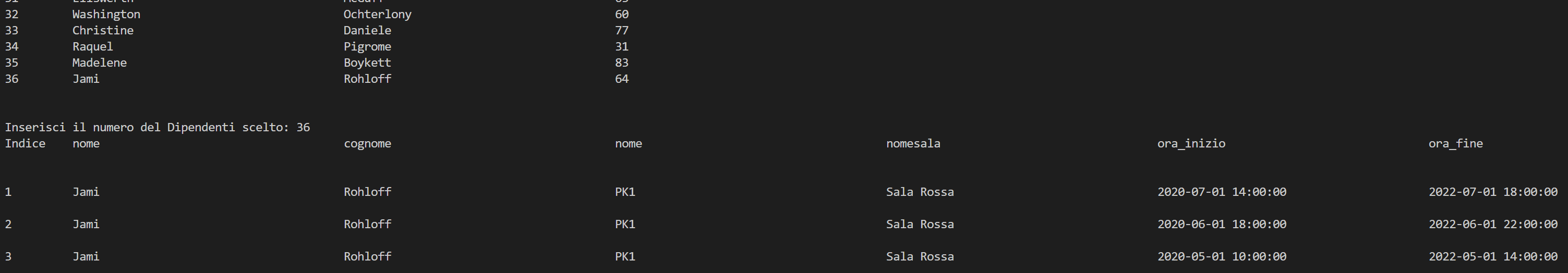
Immagine che contiene testo

Descrizione generata automaticamente

1. Vedere i turni di lavoro di una dipendete

Immagine che contiene testo

Descrizione generata automaticamente



1. Dipendenti con media vittorie sopra la media durante il loro turno

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamenteUtile per identificare dipendenti distratti o malintenzionati che arrecano danni al casino se molto al disopra della media

7. Indice:

Disponendo di un grande database frequentemente interrogato si prende in considerazione l’idea di creare un index. Le transizioni effettuate possono essere lette milioni di volte in una giornata dalle app dei clienti, dalle query per l’analisi statistica, dai cartelloni pubblicitari che mostrano le vincite, dai terminali dello staff e dalla zona di ritiro contante. Quindi prevedendo un uso su larga scala il legame al cliente di questa entità si sviluppa un indice che permette di indicizzare idpostazione e idcliente:

CREATE INDEX idx\_transazione ON transazione(idpostazione,idcliente)

Codice C++