

# CASINO'

## INTRODUZIONE

In questo progetto abbiamo realizzato un database che permette di gestire e memorizzare informazioni relative ad un complesso di casinò, in particolare i dati inerenti ai casinò stessi, ai giocatori e alle loro giocate effettuate nei vari giochi e/o scommesse che sono resi disponibili.

Il tutto permette di fornire una panoramica generale sui movimenti di denaro che vengono effettuati attraverso le giocate, consentendo di estrarre statistiche e mantenere uno storico consistente nel tempo.

## ANALISI DEI REQUISITI

Il progetto vuole mostrare una base di dati che rappresenti i vari giocatori che si sono registrati in un complesso di casinò, i loro saldi associati, le loro giocate e scommesse effettuate e i conti dei casinò presenti nel database.

Ogni **giocatore** è caratterizzato da: un codice fiscale univoco, un nome, un cognome, una data di nascita e una nazionalità.

Ad ogni giocatore è associato almeno un **saldo**, costituito da un saldo reale e da un saldo bonus, che viene identificato da un ID univoco all'interno di un singolo casinò, motivo per il quale un giocatore può detenere più saldi.

Ogni **casinò** è caratterizzato da: un ID univoco, un indirizzo, da un paese e da una data di apertura.

Ad ogni casinò viene inoltre associato un proprio **conto**, il quale presenta a sua volta un ID che permette di identificarlo in modo univoco, un importo corrispondente alla quantità di denaro a disposizione del casinò corrispondente ed infine il nome della banca in cui questo conto risiede.

Ogni casinò organizza ed emette delle **scommesse**, le quali si suddividono in **scommesse sui cavalli** e in **scommesse calcistiche**.

La base di questa gerarchia, costituita da scommesse, è caratterizzata da un ID univoco, dalla quota, dalla data di apertura e da una data di chiusura della scommessa stessa.

Le scommesse sui cavalli presentano inoltre l'informazione in merito al nome del cavallo e della gara a cui la scommessa fa riferimento.

Le scommesse calcistiche invece offrono l'esito e la partita della scommessa stessa.

Ogni giocatore può eseguire una **effettuazione** su tali scommesse che vengono emesse.

Ogni effettuazione è quindi collegata al giocatore che la effettua e alla scommessa a cui fa riferimento, presentando inoltre l'importo giocato, l'esito finale della scommessa e la data in cui viene giocata.

I casinò, oltre alle scommesse, rendono disponibili dei **giochi**, i quali si suddividono in:

- **poker**
- **blackjack**
- **roulette**

## - slot

Ogni gioco è caratterizzato da un ID e da una puntata minima ed appartiene ad un solo casinò.

Il poker presenta inoltre l'informazione riguardante il limite di giocatori che si possono sedere al tavolo di gioco.

Il blackjack, come il poker, presenta sempre un limite di giocatori che si possono sedere al tavolo a cui si aggiunge il numero di mazzi che il gioco presenta.

Le roulette sono caratterizzate da un moltiplicatore della vincita.

Le slot presentano un moltiplicatore massimo, il numero di linee ed infine un proprio jackpot.

Ogni giocatore può effettuare una **giocata** sui giochi che sono resi disponibili dai casinò.

Una giocata è caratterizzata dalla data in cui viene emessa, dalla eventuale vincita in denaro, dell'importo giocato dal giocatore e l'eventuale numero scommesso, nel caso in cui la giocata faccia riferimento ad una puntata su una roulette.

## GLOSSARIO DEI TERMINI

termine	descrizione	collegamenti
Giocatore	Persona che ha effettuato l'iscrizione in almeno un casinò.	Giocata, Effettuazione, Saldo
Saldo	Saldo di un giocatore, costituito dal saldo reale e dal saldo bonus, associato ad un solo casinò.	Giocatore, Casinò
Casinò	Casinò che ha emesso almeno una scommessa o che offre almeno un gioco e che detiene i saldi dei suoi iscritti.	Conto, Saldo, Scommessa, Gioco
Conto	Conto di un casinò, costituito da un importo in denaro e che risiede in una banca.	Casinò
Gioco	Gioco reso disponibile da un casinò. Si distinguono in poker, blackjack, slot, roulette.	Casinò, Giocata
Giocata	Giocata effettuata da un giocatore su di un gioco.	Giocatore, Gioco
Scommessa	Scommessa emessa da un casinò. Si distinguono in scommesse sui cavalli e scommesse calcistiche.	Effettuazione, Casinò
Effettuazione	Effettuazione di una scommessa da parte di un giocatore su di una scommessa.	Giocatore, Scommessa

## LISTA ENTITA'

<b>Giocatore</b>		
<u>Codice Fiscale</u>	CHAR(16)	Indica il codice fiscale univoco per giocatore.
Nome	VARCHAR(255)	Rappresenta il nome del giocatore
Cognome	VARCHAR(255)	Rappresenta il Cognome del giocatore
Data_di_Nascita	DATE	Indica la data di nascita del giocatore
Nazionalità	VARCHAR(255)	Indica la Nazionalità del giocatore

<b>Conto</b>		
<u>ID_Conto</u>	CHAR(10)	Codice che rappresenta univocamente il conto
Importo	DECIMAL(12,2)	L'importo del conto
Banca	VARCHAR(255)	La banca a cui il conto e' depositato

<b>Casino</b>		
<u>ID_Casino</u>	VARCHAR(30)	Nome Univoco del casino
Indirizzo	VARCHAR(255)	Indirizzo del casino, ovvero la via.
Nazionalita	VARCHAR(255)	La nazionalita' del casino'
Data_Apertura	DATE	La data in cui ha aperto
Conto	CHAR(10)	il codice del conto del casino'

<b>Gioco</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente il gioco
Puntata_Minima	DECIMAL(4,2)	L'importo minimo per una puntata
Casino	VARCHAR(30)	Il casino' che possiede quel gioco

Si specializza in:

<b>Poker</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente il Poker
Limite_Tavolo	INT	Numeri Di giocatori per Tavolo
<b>BlackJack</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente il BlackJack
Numero_Mazzi	INT	Numeri Di mazzi usati in quel BlackJack
Limite_Tavolo	INT	Numeri Di giocatori per Tavolo
<b>Roulette</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente la <b>Roulette</b>
Moltiplicatore_Numero_Vincente	DECIMAL(5,2)	Moltiplicatore in caso di vincita
<b>Slot</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente la Slot
Moltiplicatore_Massimo	DECIMAL(5,2)	Indica il moltiplicatore massimo della slot
Numero_Linee	INT	Numeri di linee della slot
JackPot	DECIMAL(10,2)	Importo Jackpot

<b>Giocata</b>		
<u>ID_Gioco</u>	CHAR(10)	Codice che rappresenta univocamente il gioco
<u>CF_Giocatore</u>	CHAR(16)	Codice che rappresenta il CF del giocatore che ha fatto la giocata
Importo	DECIMAL(10, 2)	Importo della Giocata
Vincita	DECIMAL(10,2)	Importo Vinto
Data_Giocata	DATE	Data della giocata

Numero_Scommesso	INT	Numero Scommesso in caso di Roulette
Colore_Scommesso	CHAR(1)	Colore Scommesso in caso di Roulette

<b>Scommessa</b>		
<u>ID_Scommessa</u>	CHAR(10)	Codice che rappresenta univocamente la Scommessa
Quota	DECIMAL(4,2)	Indica la quota della scommessa
Data_Apertura	DATE,	Data apertura della scommessa
Data_Chiusura	DATE,	Data chiusura della scommessa
Casino	VARCHAR(30)	Indica il casino che ha aperto quella scommessa

Si specializza in :

<b>Scommessa_Cavallo</b>		
<u>ID_Scommessa</u>	CHAR(10)	Codice che rappresenta univocamente la Scommessa sui Cavalli
Cavallo	VARCHAR(255)	Nome del Cavallo
Gara	VARCHAR(20)	Nome della Gara

<b>Scommessa_Calcio</b>		
<u>ID_Scommessa</u>	CHAR(10)	Codice che rappresenta univocamente la Scommessa sul calcio
Risultato	VARCHAR(2)	Risultato della partita
Partita	VARCHAR(255)	Squadre

<b>Effettuazione</b>		
<u>ID_Scommessa</u>	CHAR(10)	Codice che rappresenta univocamente la scommessa
<u>CF_Giocatore</u>	CHAR(16)	Codice che rappresenta il CF del giocatore che ha fatto la scommessa
Importo	DECIMAL(10, 2)	Importo della Scommessa
Esito	BOOLEAN,	Indica se si ha vinto o meno
Data_Effettuazione	DATE	Data della effettuazione della

		scommessa
--	--	-----------

<b>Saldo</b>		
<u>ID_Saldo</u>	CHAR(10)	Rappresenta univocamente un saldo
<u>ID_Casino</u>	VARCHAR(30)	Indica il casino a cui appartiene il conto del Giocatore
Bonus	DECIMAL(10,2)	Importo Saldo Bonus
Saldo_Reale	DECIMAL(10,2)	Importo Saldo Reale
CF_Giocatore	CHAR(16)	Codice fiscale del giocatore a cui appartiene il conto

## LISTA DELLE RELAZIONI

### Giocatore-Saldo: **Detenimento**

- Un giocatore può detenere più saldi (1,N)
- Un saldo può essere detenuto da un solo giocatore (1,1)

### Saldo-Casinò: **Collegamento**

- Un saldo fa riferimento ad un casinò (1,1)
- Ad un casinò possono fare riferimento più saldi (1,N)

### Giocatore-Gioco: **Giocata**

- Un giocatore può effettuare zero o più giocate a giochi differenti (0,N)
- Un gioco può essere giocato da zero o più giocatori (0,N)

### Giocatore-Scommessa: **Effettuazione**

- Un giocatore può effettuare zero o più scommesse (0,N)
- Una scommessa può essere effettuata da zero o più giocatori (0,N)

### Casinò-Gioco: **Possessione**

- Un casinò può possedere più giochi (1,N)
- Un gioco è posseduto da un solo casinò (1,1)

### Casinò-Scommessa: **Organizzazione**

- Un casinò può organizzare più scommesse (1,N)
- Una scommessa viene organizzata da un solo casinò (1,1)

### Casinò-Conto: **Appartenenza**

- Ad un casinò appartiene un solo conto (1,1)
- Un conto appartiene un solo casinò (1,1)

## SCELTA DEGLI IDENTIFICATORI PRINCIPALI

La chiave di “**Giocatore**” è costituita da un codice fiscale univoco, composto da una lunghezza esatta di 16 caratteri.

Nell’entità “**Casino**” la chiave è composta da un attributo “id\_casino” di lunghezza variabile e massima di 30 caratteri.

Per identificare il “**Saldo**” si è aggiunto un attributo “id\_saldo” di lunghezza variabile ma con

un massimo di 10 caratteri, che insieme al campo "id\_casino" costituisce la chiave di questa entità: in questo modo ogni saldo avrà un "Id\_saldo" unico all'interno di un casinò.

Un "**Conto**" viene identificato in modo univoco da un "id\_conto" di lunghezza variabile ma con un massimo di 10 caratteri.

Ogni "**Gioco**" ha invece come chiave primaria un attributo chiamato "id\_gioco" composto da esattamente 10 caratteri. Questa chiave permette inoltre la realizzazione della gerarchia giochi composta da "**Poker**", "**BlackJack**", "**Roulette**" e "**Slot**".

Nell'entità "**Giocata**" la chiave è composta dal "cf\_giocatore", ovvero il codice fiscale del giocatore che effettua la giocata, insieme all' "id\_gioco" che permette di identificare in maniera univoca il gioco su cui tale giocata viene effettuata.

Ogni "**Scommessa**" in modo analogo viene identificata da un campo "id\_scommessa" composto esattamente da 10 caratteri e tale attributo permette la realizzazione della gerarchia scommesse composta da "**Scommessa\_Cavalli**" e "**Scommessa\_Calcio**".

L'entità "**Effettuazione**" presenta come chiave il "cf\_giocatore", ovvero il codice fiscale del giocatore che effettua la scommessa, insieme all' "id\_scommessa" che permette di identificare in modo univoco la scommessa piazzata.

## GESTIONE RIDONDANZE NELL'IMPLEMENTAZIONE DELLE GERARCHIE

Abbiamo creato dei trigger per assicurarci che un utente in possesso dell'accesso al db non possa aggiungere un nuovo Gioco (ovvero Poker, Roulette, Slot o BlackJack) con un ID\_Gioco già appartenente ad uno dei giochi.

Senza questo trigger infatti era possibile poter creare un Gioco con l'ID\_Gioco già usato in un altro gioco.

- Funzionamento : Eseguiamo una query prima di ogni insert che controlla se il nuovo ID\_Gioco è già presente nella unione delle tabelle dei vari giochi, ovvero se appartiene alla unione delle tabelle di Poker, Slot e Roulette

### Funzione Trigger Gioco

```
CREATE FUNCTION checkDuplicateGioco() RETURNS TRIGGER AS $$  
BEGIN
```

```
    IF EXISTS (  
        SELECT 1  
        FROM (  
            SELECT ID_Gioco FROM Poker  
            UNION  
            SELECT ID_Gioco FROM Slot  
            UNION  
            SELECT ID_Gioco FROM BlackJack  
            UNION  
            SELECT ID_Gioco FROM Roulette  
        ) AS CombinedGames  
        WHERE ID_Gioco = NEW.ID_Gioco  
    ) THEN  
        RAISE EXCEPTION 'Gioco già assegnato';
```

```

        ELSE
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER checkDuplicatePokerTrigger
BEFORE INSERT ON Poker
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateGioco();

CREATE TRIGGER checkDuplicateSlotTrigger
BEFORE INSERT ON Slot
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateGioco();

CREATE TRIGGER checkDuplicateBlackJackTrigger
BEFORE INSERT ON BlackJack
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateGioco();

CREATE TRIGGER checkDuplicateRouletteTrigger
BEFORE INSERT ON Roulette
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateGioco();

```

Per lo stesso motivo si è creato un trigger identico anche per le Scommesse. Lo scopo ed il funzionamento sono i rispettivi con le opportune modifiche di quello relativo ai Giochi.

#### Funzione Trigger Scommesse

```

CREATE FUNCTION checkDuplicateScommessa() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM (
            SELECT ID_Scommessa FROM Scommessa_Cavallo
            UNION
            SELECT ID_Scommessa FROM Scommessa_Calcio
        ) AS CombinedGames
        WHERE ID_Scommessa = NEW.ID_Scommessa
    ) THEN
        RAISE EXCEPTION 'Scommessa già assegnata';
    ELSE
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;

```



```
CREATE TRIGGER checkDuplicateScommessaCavalloTrigger
BEFORE INSERT ON Scommessa_Cavallo
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateScommessa();
```

```
CREATE TRIGGER checkDuplicateScommessaCalcioTrigger
BEFORE INSERT ON Scommessa_Calcio
FOR EACH ROW
EXECUTE FUNCTION checkDuplicateScommessa();
```

## SCHEMA RELAZIONALE E VINCOLI DI INTEGRITA' REFERENZIALE

Giocatore(**Codice\_Fiscale**, Nome, Cognome, Data\_di\_Nascita, Nazionalità);

Conto (**ID\_Conto**, Importo , Banca);

Casinò (**ID\_Casinò** , indirizzo , Paese , Data\_Apertura, Conto\* );  
 Casinò (Conto) -> Conto(ID\_Conto).

Gioco(**ID\_Gioco**, Puntata\_Minima, Casino\*);  
 Gioco(Casinò ) -> Casinò (ID\_Casinò ).

- Poker(**ID\_Gioco\***, Limite\_Tavolo);  
 Poker(ID\_Gioco) -> Gioco(ID\_Gioco).
- BlackJack(**ID\_Gioco\***, Numero\_Mazzi, Limite\_Tavolo);  
 BlackJack(ID\_Gioco) -> Gioco(ID\_Gioco).
- Roulette(**ID\_Gioco\***, Moltiplicatore\_Numero\_Vincente);  
 Roulette(ID\_Gioco) -> Gioco(ID\_Gioco).
- Slot(**ID\_Gioco\***, Moltiplicatore\_Massimo, Numero\_Linee, JackPot);  
 Slot(ID\_Gioco) -> Gioco(ID\_Gioco).

Giocata(**ID\_Gioco\***,**CF\_Giocatore\***, Importo, Vincita, Data\_Giocata,Numer\_Scommesso);  
 Giocata(ID\_Gioco) -> Gioco(ID\_Gioco).  
 Giocata(CF\_Giocatore) -> Giocatore(Codice\_Fiscale).

Scommessa(**ID\_Scommessa**, Quota, Data\_Apertura, Data\_Chiusura, Casino\*);  
 Scommessa(Casinò ) -> Casinò (ID\_Casinò ).

- Scommessa\_Cavallo(**ID\_Scommessa\***, Cavallo , Gara);  
 Scommessa\_Cavallo(ID\_Scommessa) -> Scommessa(ID\_Scommessa).
- Scommessa\_Calcio(**ID\_Scommessa\***, Risultato , Partita);  
 Scommessa\_Calcio(ID\_Scommessa) -> Scommessa(ID\_Scommessa).

Effettuazione(**ID\_Scommessa\***,**CF\_Giocatore\***, Importo, Esito, Data\_Effettuazione);  
 Effettuazione(ID\_Scommessa) -> Scommessa(ID\_Scommessa).  
 Effettuazione(CF\_Giocatore) -> Giocatore(Codice\_Fiscale).

Saldo(**ID\_Saldo**, **ID\_Casinò \***, Bonus , Saldo\_Reale , CF\_Giocatore \* );  
 Saldo(ID\_Casinò ) -> Casinò (ID\_Casino).  
 Saldo(CF\_Giocatore) -> Giocatore(Codice\_Fiscale).

DIAGRAMMA E-R

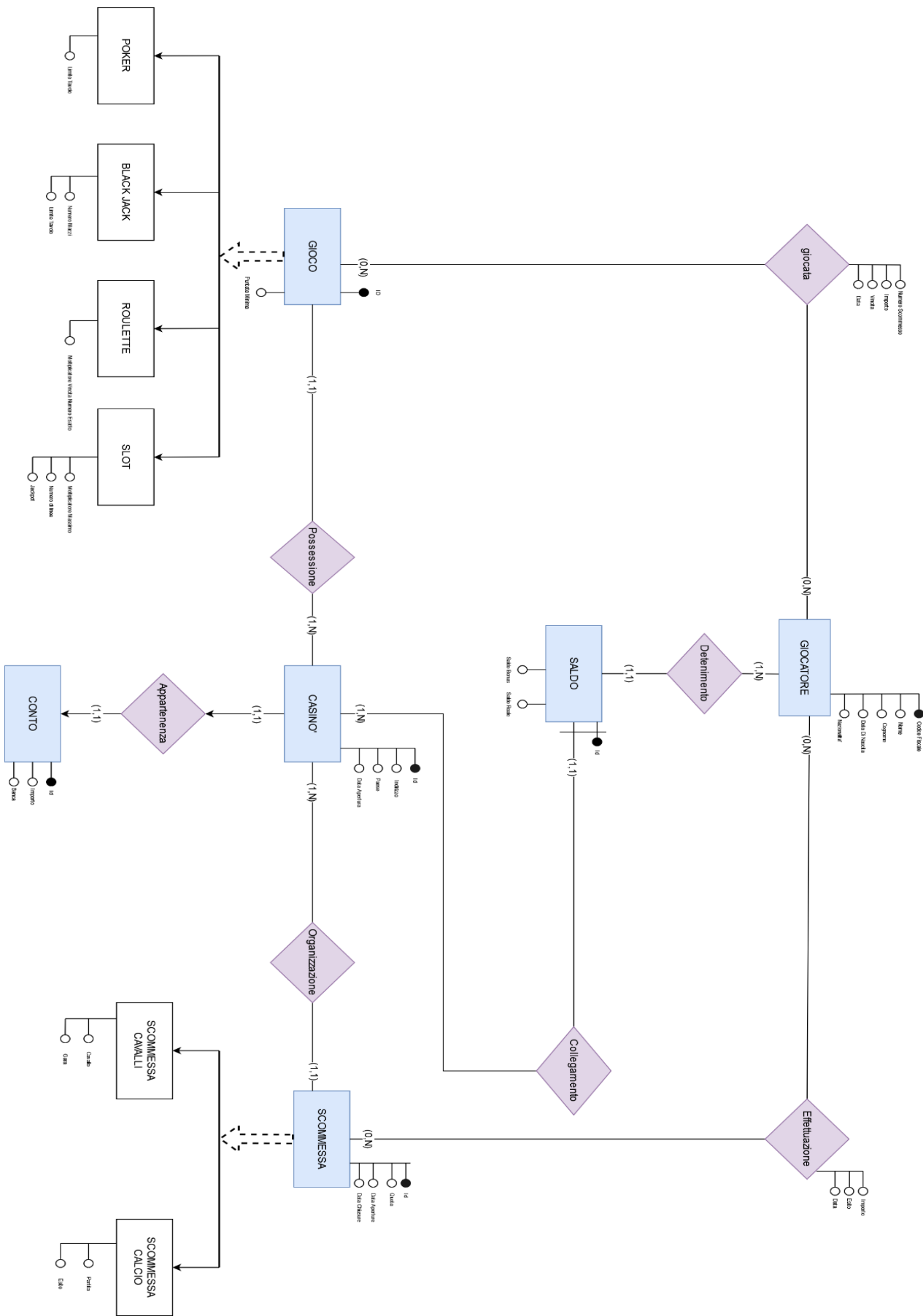
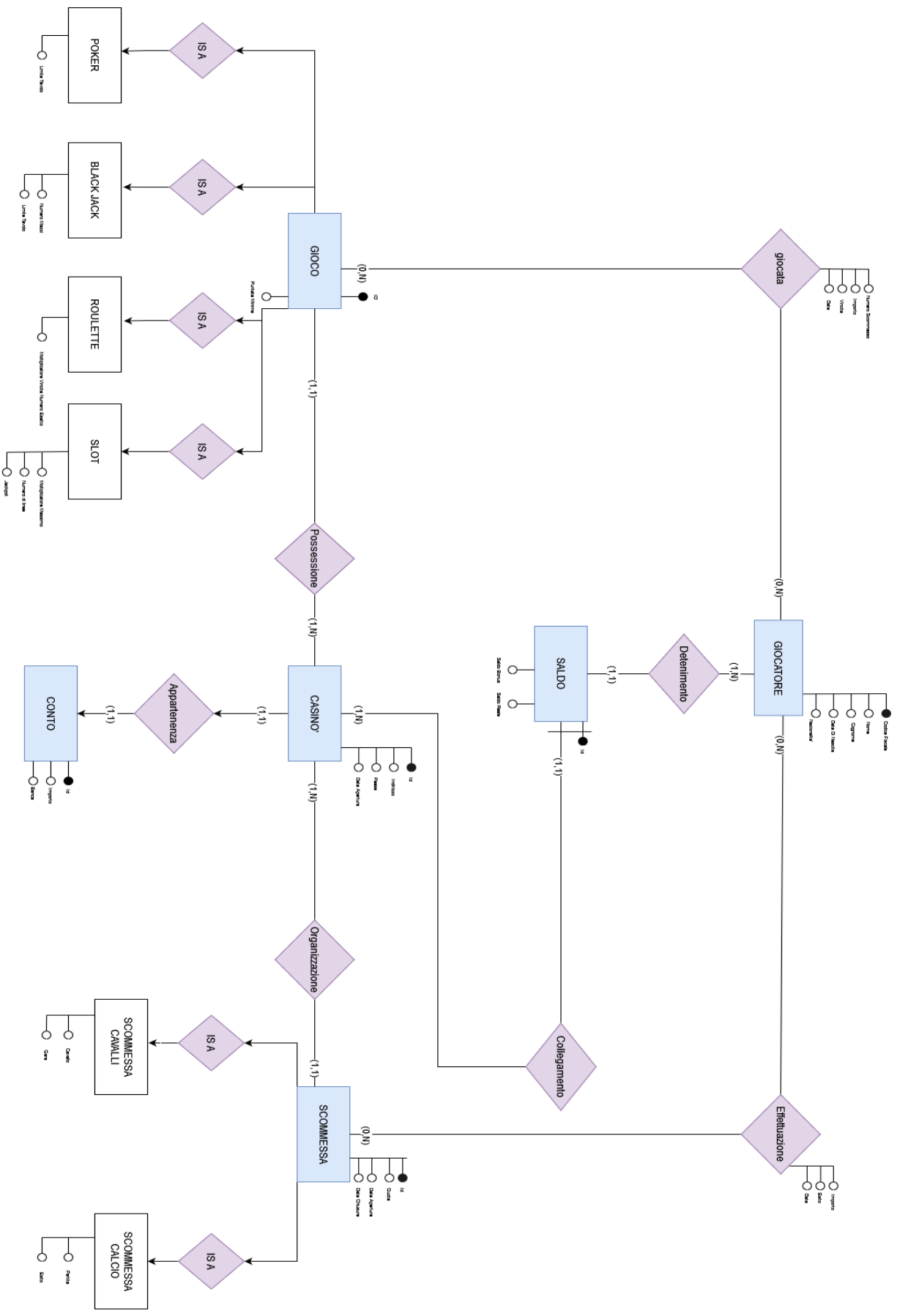


DIAGRAMMA E-R REIFICATO



## QUERY

1. *Mostrare la top 20 delle vincite sulle scommesse sui cavalli*

```
SELECT G.Nome, G.Cognome, E.Importo*Sc.Quota AS Vincita, S.Cavallo, S.Gara
FROM ((Giocatore AS G
      JOIN
      Effettuazione AS E
      ON G.Codice_Fiscale=E.CF_Giocatore)
      JOIN
      Scommessa_Cavallo AS S
      ON S.ID_Scommessa=E.ID_Scommessa)
      JOIN
      Scommessa AS Sc
      ON S.ID_Scommessa=Sc.ID_Scommessa
WHERE E.Esito=TRUE
ORDER BY Vincita DESC
LIMIT 20
```

2. *Mostrare i giocatori che detengono un complessivo di saldi reali che supera i 500€*

```
SELECT DISTINCT G.Nome, G.Cognome, SUM(S.Saldo_Reale)
FROM (Giocatore AS G
      JOIN
      Saldo AS S
      ON G.Codice_Fiscale=S.CF_Giocatore)
GROUP BY G.Nome, G.Cognome
HAVING SUM(S.Saldo_Reale) > 500
```

3. *Mostrare il totale scommesso da ogni giocatore su scommesse calcistiche e sui cavalli, in ordine decrescente*

```
SELECT G.Nome, G.Cognome, SUM(E.Importo) AS Tot_Scommesso
FROM (Giocatore AS G
      JOIN
      Effettuazione AS E
      ON G.Codice_Fiscale=E.CF_Giocatore)
GROUP BY G.Nome, G.Cognome
ORDER BY Tot_Scommesso DESC
```

4. *Mostrare le top 20 perdite avvenute su scommesse di calcio*

```
SELECT G.Nome, G.Cognome, E.Importo AS Perdita, S.Risultato, S.Partita
FROM (Giocatore AS G
      JOIN
      Effettuazione AS E
      ON G.Codice_Fiscale=E.CF_Giocatore)
      JOIN
      Scommessa_Calcio AS S
      ON S.ID_Scommessa=E.ID_Scommessa
WHERE E.Esito=FALSE
```

ORDER BY Perdita DESC  
LIMIT 20

5. *Mostra il codice fiscale, il nome e il numero di giocate a poker in ordine decrescente dei giocatori che hanno giocato a poker tra il '2020-05-22' e il '2022-05-22'.*

```
SELECT G.Codice_Fiscale ,G.nome, COUNT(*) AS Numero_Giocate
FROM Giocatore AS G
    JOIN Giocata AS J
    ON G.Codice_Fiscale = J.CF_Giocatore
    JOIN Gioco AS O
    ON O.ID_Gioco = J.ID_Gioco
    JOIN Poker AS P
    ON P.ID_Gioco = O.ID_Gioco
    JOIN Casino AS C
    ON O.Casino = C.ID_Casino
WHERE J.Data_Giocata >= '2020-05-22' AND J.Data_Giocata <= '2022-05-22'
GROUP BY G.Codice_Fiscale , G.nome
ORDER BY Numero_Giocate DESC
```

6. *Mostra i Casinò (ID\_casino, indirizzo e nazionalità) che possiedono un conto superiore a 7.000.000 euro e con almeno un tavolo da poker avente il limite di giocatori maggiore di 7 e il numero di tavoli da poker che hanno in totale ciascuno .*

```
SELECT DISTINCT C.ID_Casino, C.indirizzo, C.nazionalita , COUNT(P.ID_Gioco) as
Numero_Tavoli
FROM ((Casino as C
    JOIN Conto as Co
    ON C.Conto = Co.ID_conto)
    JOIN Gioco as G
    ON C.ID_Casino = G.casino)
    JOIN Poker as P
    ON G.ID_Gioco = P.ID_Gioco
WHERE Co.Importo >= 7000000 AND P.Limite_Tavolo > 7
GROUP BY C.ID_Casino, C.indirizzo, C.nazionalita
```

7. *Mostra i Casinò (ID\_casino, indirizzo e nazionalità) e il numero di giocate alle slot di quel casinò tra il '2015/01/01' e il '2022/12/31'*

```
SELECT DISTINCT C.ID_Casino , C.indirizzo, C.nazionalita, COUNT(*) AS
Numero_Giocate
FROM ((Casino as C
    JOIN Gioco as G
    ON C.ID_Casino = G.casino)
    JOIN Slot as SL
    ON G.ID_Gioco = SL.ID_Gioco)
    JOIN Giocata as S
    ON SL.ID_Gioco = S.ID_Gioco
WHERE S.Data_Giocata >= '2015/01/01'
AND S.Data_Giocata <= '2022/12/31'
GROUP BY C.ID_Casino, C.indirizzo, C.nazionalita
```

## INDICI

Questo database potenzialmente è utilizzabile su larga scala: è dunque necessaria la valutazione di inserimento di indici.

Per velocizzare l'esecuzione delle query che effettuano un controllo di uguaglianza sul codice fiscale dei giocatori all'interno della entità effettuazione, creiamo dunque l'indice *indice\_cf\_effettuazione* attraverso il seguente comando:

```
CREATE INDEX indice_cf_effettuazione ON Effettuazione USING  
hash(CF_Giocatore);
```

Utilizziamo l'hash in quanto velocizza i controlli sull'uguaglianza, controllo che viene eseguito all'interno del primo join della query. Tale indice porta ad esempio l'esecuzione della prima query dai 170-200 msec a 150 msec.

Nella prima e quarta query, per velocizzare la ricerca, è utile creare un indice composto sulla colonna Esito della tabella Effettuazione poiché stiamo appunto filtrando le righe in base al valore di Esito. Tale indice potrebbe aiutare il database a individuare più rapidamente le righe corrispondenti.

```
CREATE INDEX indice_esito ON Effettuazione (Esito);
```

Questo indice porta ad esempio l'esecuzione della quarta query dai 330 msec ai 250 msec.

Come già detto in precedenza, per l'esecuzione del controllo di uguaglianza sul codice fiscale dei giocatori all'interno della entità effettuazione, creiamo un indice intitolato *indice\_giocata\_cf\_giocatore* che permette di velocizzare l'esecuzione di query che effettuano il controllo del Cf degli giocatori all'interno della entità giocata.

```
CREATE INDEX indice_giocata_cf_giocatore ON Giocata USING  
hash(CF_Giocatore);
```

Nella quinta e settima query, per velocizzare la ricerca, è utile creare un indice sulla colonna *Data\_Giocata* della tabella Giocata poiché stiamo appunto filtrando le righe in base al valore di *Data\_Giocata*. Il comando per creare l'indice è il seguente e ci permette di individuare più rapidamente le righe corrispondenti.

```
CREATE INDEX indice_data_giocata ON Giocata (Data_Giocata);
```

## CODICE C++

Il codice C++ incluso nel progetto comprende il main che se eseguito permette di scegliere quale query eseguire tra quelle presentate in un elenco stampato su terminale, stampando a sua volta a schermo il risultato prodotto.

Per la compilazione è sufficiente eseguire il seguente comando all'interno della cartella del programma (attenzione: deve essere presente la cartella "dependencies", costituita a sua volta dalla cartella "include" contenente i file 'libpq-fe.h', 'pg\_config\_ext.h' e 'postgres\_ext.h' e dalla cartella "lib" contenente i file 'libpq.dll' e 'libpq.lib'):

```
g++ interfaccia.cpp -L dependencies\lib -lpq -o interfaccia
```

Per eseguire il programma dobbiamo invece utilizzare la linea di comando: *./interfaccia*

Al fine del login con il Database sono definite le seguenti costanti all'inizio del programma, che vanno ridefinite in base alla configurazione del database importato nella macchina locale.

PG\_HOST, PG\_USER, PG\_PASS, PG\_PORT, PG\_DB

Il programma attraverso un ciclo *while* continua a stampare le 7 query, chiedendo di inserire tramite tastiera il numero della query da eseguire, stampando in seguito su terminale il risultato della query selezionata.

La scelta della query da eseguire viene realizzata tramite uno *switch/case* sulla variabile che riceve come contenuto l'input preso da tastiera: come default tale costruzione condizionale prevede la chiusura del programma, attraverso il cambiamento di valore di una *flag* booleana dunque, per uscire dal programma, basta immettere da tastiera un carattere differente dagli interi che identificano le query nella lista.

Prima di poter eseguire le query il codice di assicura della corretta connessione al database, stampando a schermo lo stato del collegamento.

Ogni risultato della query eseguita viene inoltre controllato tramite l'esecuzione della funzione *checkResults*.

Il risultato delle query è di tipo *PGResult* e viene ottenuto attraverso l'esecuzione della classe *PGexec*, la quale riceve nel suo costruttore come argomento la query vera e propria. Successivamente dal risultato vengono estratti il numero delle tuple e dei campi attraverso *PQntuples* e *PQnfields*, valori che ci permetteranno di stampare correttamente a schermo attraverso dei cicli *for* i valori degli attributi prelevati dal risultato attraverso *PQfname* e *PQgetvalue*.