**STABILIMENTO BALNEARE**

Lo stabilimento balneare "Sole & Mare" desidera un sistema per gestire le proprie attività. Ogni ombrellone è identificato da un numero e può avere uno o più lettini associati. Ciascun ombrellone può essere prenotato per specifici giorni da diversi clienti. I clienti sono identificati tramite codice fiscale, nome, cognome e contatti (telefono, email). Lo stabilimento dispone anche di un chiosco bar, dove vengono proposti prodotti da cucina, distinti in piatti unici (come insalate, focacce, friselle al pomodoro, etc.) e proposte di menù, a loro volta composte da due piatti unici (i.e. di pasta, focacce, pizze, frisella, cotoletta, etc.) e da un contorno (i.e. verdure o insalate). Ogni elemento della proposta di menu ha nome e prezzo, e ogni consumazione deve essere associata alla prenotazione che l'ha richiesta. Inoltre, il personale (distinto in bagnini e addetti al bar) va registrato, indicando nome, cognome, ruolo, data di assunzione e retribuzione. Alcuni addetti al bar possono essere assegnati a più ombrelloni (ad esempio per il servizio in spiaggia). Lo stabilimento offre anche la possibilità di aggiungere servizi extra (es. lettini, sedie) e attività organizzate dallo stabilimento (es. corsi di yoga, tornei), che hanno un nome, una data e un orario, e possono essere prenotate dai clienti. Infine, si desidera tracciare i pagamenti dei clienti, specificando l'importo e la data di pagamento, distinguendo tra quelli relativi al noleggio e quelli relativi a consumazioni o extra.

---

Svolgere le seguenti attività

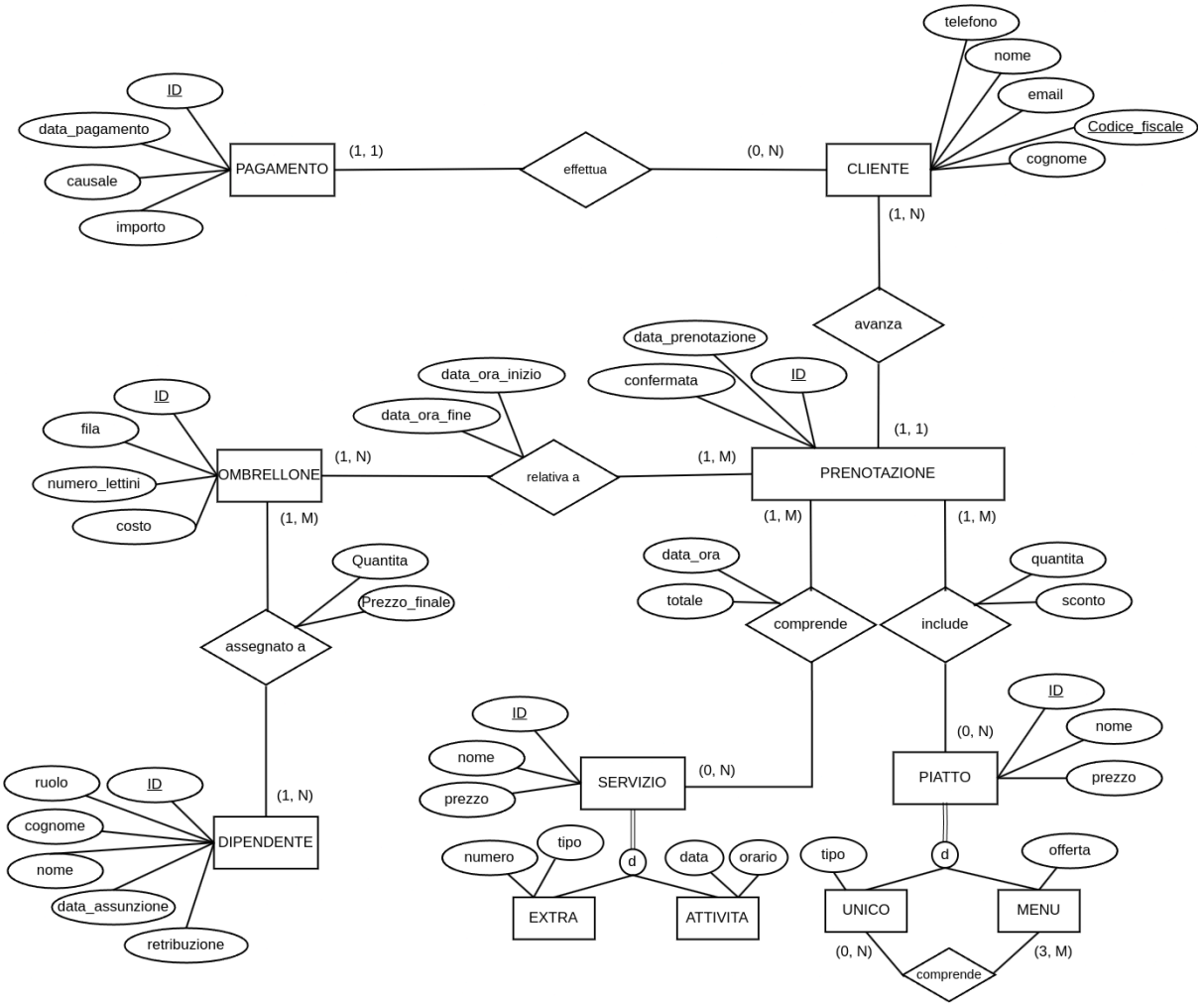
1. Si definisca un diagramma EER che modelli la realtà commentandolo ed indicando le ipotesi fatte.
2. Si derivi dal diagramma EER il modello relazionale corrispondente, indicando eventuali ottimizzazioni o normalizzazioni applicabili.
3. Dato il seguente modello relazionale di una Ludoteca, scrivere in SQL o in algebra relazionale le seguenti query

**GIOCO**: IDGioco (PK), Titolo, Categoria, EtàMinima
**UTENTE**: IDUtente (PK), Nome, Cognome, DataNascita, Telefono, Email
**PRESTITO**: IDPrestito (PK), DataInizio, DataFine, IDUtente (FK → UTENTE.IDUtente), IDGioco (FK → GIOCO.IDGioco), Note
**POSTAZIONE**: IDPostazione (PK), NomePostazione, Tipo (es. "PC", "Console", "Tavolo da gioco"), Disponibilità (boolean)
**PRENOTAZIONE**: IDPrenotazione (PK), DataPrenotazione, OraInizio, OraFine, IDPostazione (FK → STAZIONE.IDPostazione), IDUtente (FK → UTENTE.IDUtente)

1. Elencare tutti i prestiti **attivi** e gli utenti che li hanno effettuati
2. Calcolare per ogni categoria di gioco il **numero di prestiti totali**, ordinando per numero decrescente di prestiti
3. Elencare tutti i giochi appartenenti a categorie che hanno più di 5 giochi in totale

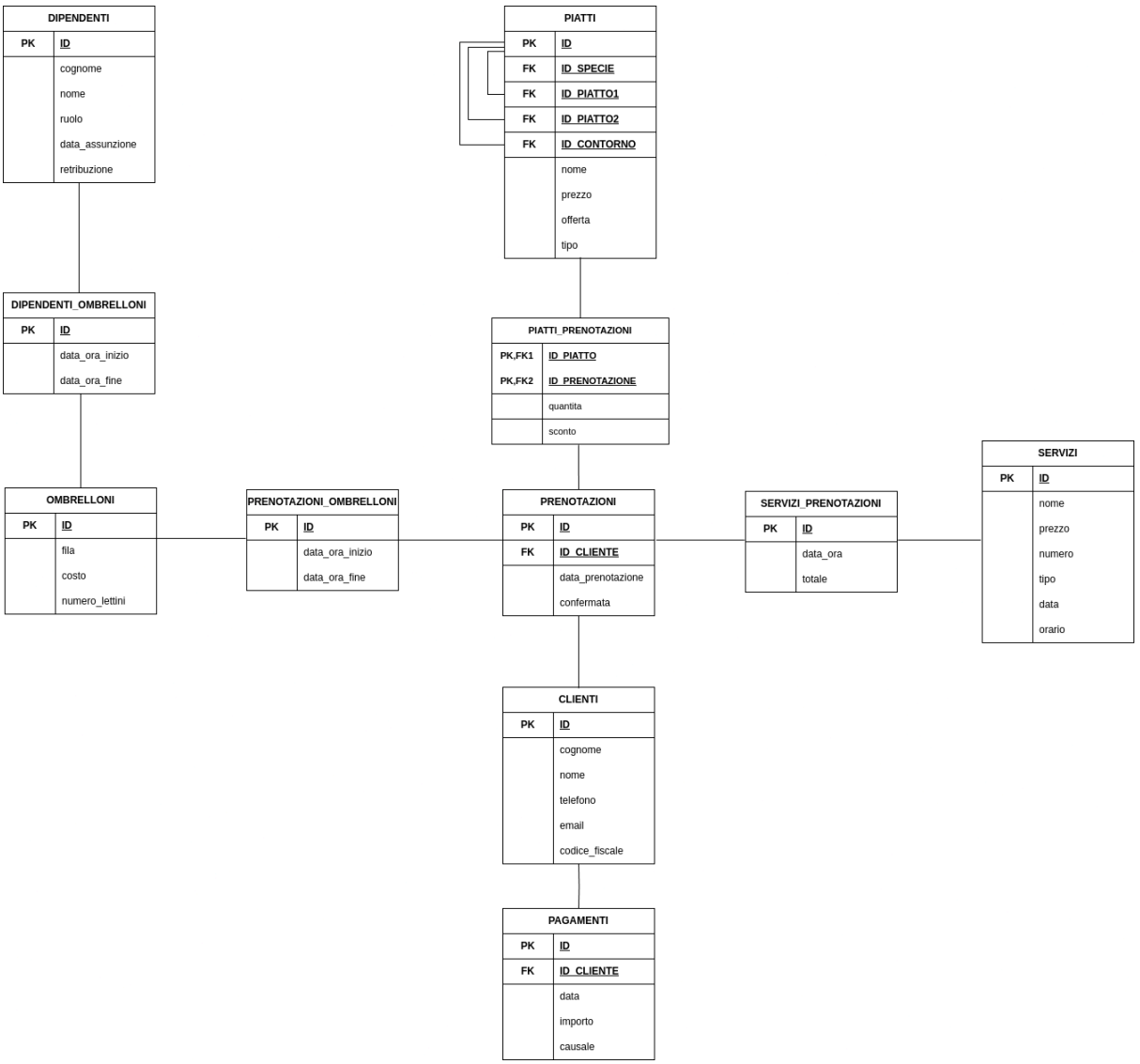
## PREMESSA

**La soluzione seguente rappresenta solo una proposta che può coincidere in tutto o in parte con quanto sviluppato da ciascuno di voi. Oltre a differenze in termini di numero e tipo di attributi di ciascuna entità, è possibile che una soluzione differisca da un'altra per numero di entità e/o associazioni. Un contributo significativo in tal senso può essere rappresentato da concetti ed estensioni previsti dall'Enhanced-Entity-Relationship (EER).**




## COMMENTO

**La soluzione poteva prevedere delle specializzazioni a supporto della proposta di un modello che presenti più caratteri di EER. L'entità DIPENDENTE ad esempio poteva essere specializzata in BAGNINO e ADDETTO BAR. In tal caso, risulta fondamentale dotare le specializzazioni di almeno un attributo ciascuna che la differenzi dalla relativa superclasse. Ad esempio, per il BAGNINO tale attributo potrebbe essere la data di acquisizione del brevetto, mentre per l'ADDETTO BAR si potrebbe pensare alla sua prerogativa di essere anche un bartender per la preparazione di cocktail.**


DIPENDENTI
PK ID
cognome
nome
ruolo
data_assunzione
retribuzione
PIATTI
PK ID
FK ID_SPECIE
FK ID_PIATTO1
FK ID_PIATTO2
FK ID_CONTORNO
nome
prezzo
offerta
tipo
DIPENDENTI_OMBRELLONI
PK ID
data_ora_inizio
data_ora_fine
PIATTI_PRENOTAZIONI
PK,FK1 ID_PIATTO
PK,FK2 ID_PRENOTAZIONE
quantita
sconto
SERVIZI
PK ID
nome
prezzo
numero
tipo
data
orario
OMBRELLONI
PK ID
fila
costo
numero_lettini
PRENOTAZIONI_OMBRELLONI
PK ID
data_ora_inizio
data_ora_fine
PRENOTAZIONI
PK ID
FK ID_CLIENTE
data_prenotazione
confermata
SERVIZI_PRENOTAZIONI
PK ID
data_ora
totale
CLIENTI
PK ID
cognome
nome
telefono
email
codice_fiscale
PAGAMENTI
PK ID
FK ID_CLIENTE
data
importo
causale

1. Elencare tutti i prestiti **attivi** e gli utenti che li hanno effettuati

```
SELECT
	P.IDPrestito, U.Nome, U.Cognome, G.Titolo, P.DataInizio, P.DataFine
FROM	PRESTITO P JOIN
	UTENTE U ON P.IDUtente = U.IDUtente JOIN
	GIOCO G ON P.IDGioco = G.IDGioco
WHERE
	P.DataFine > CURRENT_DATE;
```

2. Calcolare per ogni categoria di gioco il **numero di prestiti totali**, ordinando per numero decrescente di prestiti

```
SELECT
	G.Categoria, COUNT(*) AS NumPrestiti
FROM
	PRESTITO PR JOIN GIOCO G ON PR.IDGioco = G.IDGioco
GROUP BY G.Categoria
ORDER BY NumPrestiti DESC;
```

3. Elencare tutti i giochi appartenenti a categorie che hanno più di 5 giochi in totale

```
SELECT
	IDGioco, Titolo, Categoria
FROM
	GIOCO
WHERE
	Categoria IN (
		SELECT Categoria
		FROM GIOCO
		GROUP BY Categoria
	HAVING COUNT(*) > 5);
```