

BASI DI DATI 1 – PROFF. S. CERI, S. COMAI, L. TANCA – A.A. 18/19

APPELLO DEL 22 GENNAIO 2019 – DURATA DELLA PROVA : 2 h 30 m

Scrivete le soluzioni delle due parti su DUE FOGLI DISTINTI, entrambi intestati

Parte 1: LINGUAGGI DI INTERROGAZIONE (su un foglio a parte rispetto alla Parte 2)

CLIENTE (IDCLIENTE, NOME, COGNOME, DATA_{NASCITA}, EMAIL)
RISTORANTE (IDRISTORANTE, NOME, CITTÀ, INDIRIZZO, NUMERO_{TELEFONO},
NUMERO_{COPERTI}, ORA_{APERTURA}, ORA_{CHIUSURA})
PRENOTAZIONE (IDCLIENTE, IDRISTORANTE, DATA, ORA_{INIZIO}, NUMERO_{PERSONE})
CENA (IDCLIENTE, IDRISTORANTE, DATA, ORA_{INIZIO}, ORA_{FINE}, NUMERO_{PERSONE},
SPESA_{TOTALE})

Lo schema soprastante descrive le prenotazioni effettuate dai clienti di una catena di ristoranti. Viene tenuta traccia solo delle cene associate ad una prenotazione. Come ora di inizio della cena si tiene presente quella indicata al momento della prenotazione, a prescindere dall'orario effettivo di arrivo del cliente. L'attributo SpesaTotale memorizza il conto della cena, ovvero la spesa sostenuta da tutte le persone partecipanti.

A. SQL

1. Specificare i comandi di creazione delle tabelle PRENOTAZIONE e CENA, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo gli eventuali vincoli di integrità referenziale verso le altre tabelle.

```
Create table Prenotazione (  
  IdCliente integer references Cliente.IdCliente on delete no action on update cascade,  
  IdRistorante integer references Ristorante.IdRistorante on delete cascade on update cascade,  
  Data date,  
  OraInizio time,  
  NumeroPersone integer,  
  Primary key (IdCliente, IdRistorante, Data, OraInizio) )
```

```
Create table Cena (  
  Create table Prenotazione (  
    IdCliente integer,  
    IdRistorante integer,  
    Data date,  
    OraInizio time,  
    OraFine time,  
    NumeroPersone integer,  
    Primary key (IdCliente, IdRistorante, Data, OraInizio),  
    Foreign key (IdCliente, IdRistorante, Data, OraInizio) references Prenotazione (IdCliente,  
    IdRistorante, Data, OraInizio) on delete cascade on update cascade)
```

2. Estrarre il cliente che ha effettuato la spesa media pro-persona più alta.

```
Select Top 1 C.IdCliente  
From Cliente as C join prenotazione as P join Cena as Cn  
Group by C.IdCliente  
Order by AVG(Cn.SpeseTotale / Cn.NumeroPersone) desc
```

3. Estrarre nome e cognome dei clienti “under 30” che hanno fatto prenotazioni sempre in ristoranti diversi.

```

Select C.Nome, C.Cognome
From Cliente as C join Prenotazione as P2
Where C.DataNascita > 1/1/1989 and C.IdCliente not in (Select P.IdCliente
From Prenotazione as P
Group by P.IdCliente, P.IdRistorante
Having Count (*) > 1 )

```

4. Estrarre nome e cognome dei clienti che non hanno mai “dato buca” in ristoranti di Milano, ma che si sono presentati almeno una volta con un numero di persone inferiore a quanto indicato nella prenotazione.

```

Select C.Nome, C.Cognome
From Cliente as C join Prenotazione as P join Cena as Cn join Ristorante as R
Where Cn.NumeroPersone < P.NumeroPersone and R.Città = “Milano” and C.IdCliente in (select
P3.IdCliente
From Cena as C3 join Prenotazione as P3 join Ristorante as R3
Where R3.Città = “Milano”
Group by P3.IdCliente
Having count(*) = (Select count(*)
From Prenotazione as P2 join Ristorante as R2
Where P2.IdCliente = C.IdCliente and R2.Città = “Milano”) )

```

5. Esprimere il vincolo che verifica che non ci siano prenotazioni effettuate fuori dall’orario di apertura del ristorante.

```

Create assertion NoPrenotazioniFuoriOrario check (not exists (select *
from Prenotazione as P join Ristorante as R
where P.OraInizio < R.OraApertura or P.OraInizio < R.OraChiusura) )

```

B. Linguaggi formali

6. Formulare in Algebra Relazionale, Calcolo Relazionale e Datalog l’interrogazione A.3

$\Pi_{\text{Nome, Cognome}} \text{CLIENTE} \bowtie$
 $[\Pi_{\text{IDCliente}} (\sigma_{\text{DataNascita} > "31/12/1988"} \text{CLIENTE}) - \Pi_{\text{IDCliente}} \sigma_{\text{D} \neq \text{Data} \vee \text{O} \neq \text{OraInizio}} (\text{PRENOTAZIONE} \bowtie \rho_{\text{D, O} \leftarrow \text{Data, OraInizio}} \text{PRENOTAZIONE})]$

$\text{AlmenoUnRist2Volte}(\text{IDC}) :- \text{PRENOTAZIONE}(\text{IDC}, \text{IDR}, \text{D1}, _, _),$
 $\text{PRENOTAZIONE}(\text{IDC}, \text{IDR}, \text{D2}, _, _), \text{D1} \neq \text{D2}$
 $\text{AlmenoUnRist2Volte}(\text{IDC}) :- \text{PRENOTAZIONE}(\text{IDC}, \text{IDR}, _, \text{O1}, _),$
 $\text{PRENOTAZIONE}(\text{IDC}, \text{IDR}, _, \text{O2}, _), \text{O1} \neq \text{O2}$
 $\text{Risultato}(\text{Nome}, \text{Cognome}) :- \text{CLIENTE}(\text{IDC}, \text{NOME}, \text{COGNOME}, \text{DATANASCITA}, _),$
 $\neg \text{AlmenoUnRist2Volte}(\text{IDC}), \text{DATANASCITA} > "31/12/1988"$

$\{ t \mid \exists t1 \in \text{CLIENTE} \mid t[\text{Nome}, \text{Cognome}] = t1[\text{Nome}, \text{Cognome}] \wedge$
 $t1[\text{DataNascita}] > "31/12/1988" \wedge \neg (\exists t2 \in \text{PRENOTAZIONE}, \exists t3 \in \text{PRENOTAZIONE} \mid$
 $t1[\text{IDCLiente}] = t2[\text{IDCLiente}] \wedge t2[\text{IDCLiente}, \text{IDRistorante}] = t3[\text{IDCLiente}, \text{IDRistorante}] \wedge$
 $(t2[\text{Data}] \neq t3[\text{Data}] \vee t2[\text{OraInizio}] \neq t3[\text{OraInizio}]))$
 $\}$

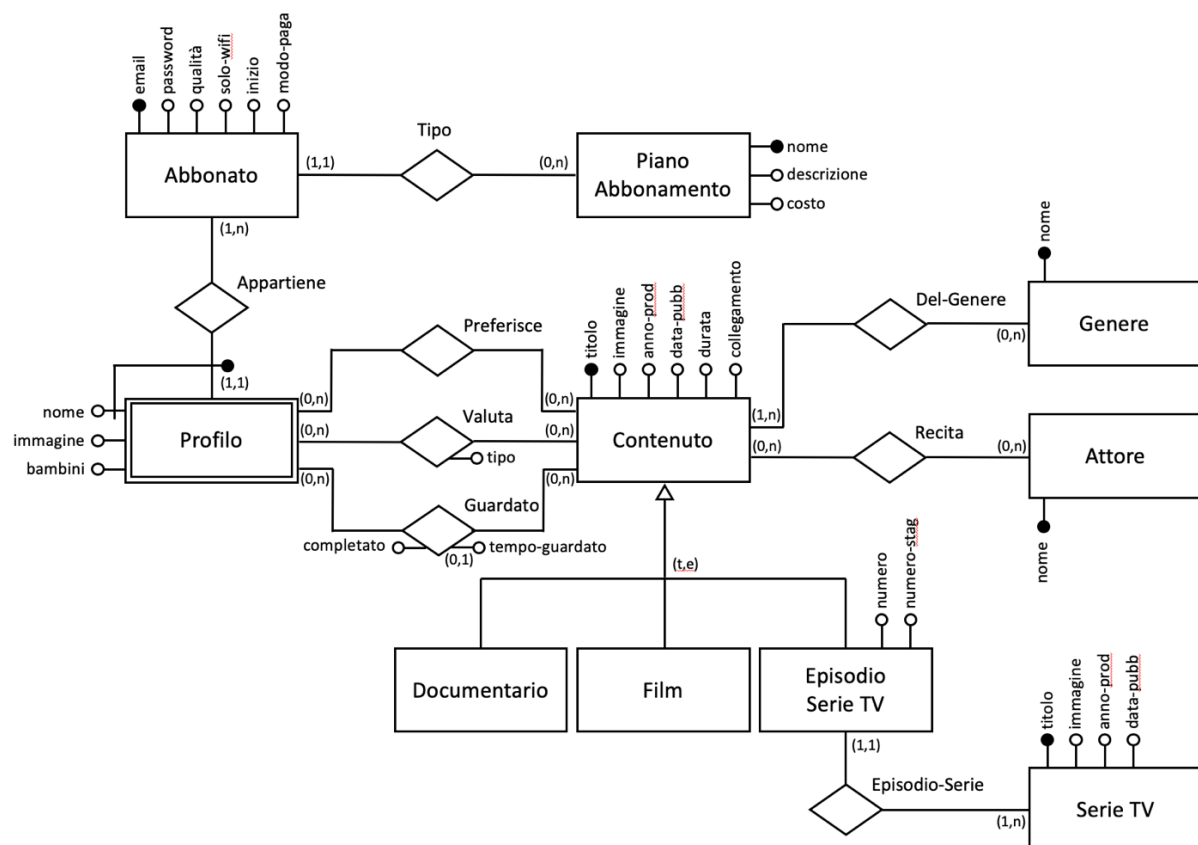
Parte 2: PROGETTAZIONE E

C. Progetto Concettuale e Progetto Logico (13 pt.)

Si vogliono rappresentare i dati di un servizio di streaming on demand che consente agli abbonati di guardare una varietà di serie TV, film e documentari. Gli abbonati si registrano attraverso un'email e una password; possono indicare la qualità video desiderata e se utilizzare il servizio solamente quando vi è la connessione wifi. Il servizio offre diversi piani di abbonamento (base, standard, premium), ciascuno caratterizzato da una descrizione ed una quota mensile. Per ogni abbonato sono noti la data di inizio dell'abbonamento e la modalità di pagamento prescelta (ad es. carta di credito, carta regalo, ecc.). L'abbonato può creare dei profili utente, caratterizzati da un'immagine, un nome e l'indicazione se si tratta di un profilo per bambini o meno. Le serie tv sono organizzate in stagioni (caratterizzate da un numero): ogni stagione è composta da un insieme di episodi, anch'essi numerati. Le serie tv, gli episodi, i film e i documentari hanno un titolo, un'immagine, l'anno di produzione, il cast di attori, il genere (sono possibili più generi per uno stesso titolo) e la data di pubblicazione sul servizio di streaming. Gli episodi, i film e i documentari hanno una durata ed il collegamento al video. Il sistema tiene traccia di quali film sono stati visti per ciascun profilo o se il film è stato visto solo parzialmente; in quest'ultimo caso si tiene traccia di quanti minuti/secondi sono già stati visualizzati. Gli abbonati possono anche valutare film, documentari e serie tv con un semplice clic (pollice alzato o abbassato) e crearsi una propria lista di preferenze.

- Il progetto è valutato per completezza, correttezza, leggibilità, minimalità e autoesplicitività.
- Nel progetto concettuale, ricordare di specificare gli **identificatori** di tutte le entità e le **cardinalità** di tutte le associazioni e degli attributi, e di disporlo insieme al progetto logico su due facciate affiancate, in modo da poterli osservare simultaneamente.
- Nel progetto logico, specificare le chiavi primarie e le chiavi esterne (integrità referenziale)

Progetto Concettuale



Progetto Logico

PianoAbbonamento (nome, descrizione, costo)

Abbonato (email, password, qualità, solo-wifi, inizio, modo-paga, pianoAbbonamento)

Profilo (email, nome, immagine, bambini)

Preferisce (email, nome, titolo)

Valuta (email, nome, titolo, tipo)

Guardato (email, nome, titolo, completato, tempo-guardato*)

Contenuto (titolo, immagine, anno-prod, data-pubb, durata, collegamento)

Film (titolo-film)

Documentario (titolo-documentario)

EpisodioSerieTV (titolo-episodio, titolo-serie, numero, numero-stagione)

SerieTV (titolo, immagine, anno-prod, data-pubb)

Attore (nome)

Genere (nome)

DelGenere (titolo, nome-genere)

Recita (titolo, nome-attore)

D. Teoria (1 pt.)

Discutere vantaggi e svantaggi di un modello dei dati come quello relazionale, basato su valori, rispetto ad altri modelli dove per i riferimenti si usano i puntatori.

Buona soluzione di uno studente

I possibili vantaggi derivanti dall'utilizzo di valori per le relazioni invece che puntatori sono:

- Indipendenza dal livello fisico, infatti i valori possono avere una rappresentazione fisica arbitraria mentre i puntatori dipendono dall'architettura;
- Semplicità di lettura, per le persone è più immediato cogliere una relazione costituita da valori invece che sequenze di numeri;
- Simmetria delle relazioni: pur essendo l'uguaglianza tra valori simmetrica non è possibile avere una relazione asimmetrica, il viceversa vale utilizzando i puntatori, i quali identificano un'area di memoria unidirezionalmente. Per un'entità puntata non è facile capire con chi è in relazione.

Esistono tuttavia svantaggi derivanti dall'utilizzo dei valori invece che dei puntatori:

- Peso maggiore su disco in quanto una relazione basata su valori ha bisogno di un'intera chiave esterna, che in alcuni casi può essere relativamente pesante;
- Velocità nel confronto: il confronto tra valori porrebbe essere computazionalmente dispendioso in caso di chiavi grandi.