

Basi di Dati
A.A. 2017-2018
Prof. Alfredo Pulvirenti
Prova in Itinere 5 dicembre 2017

Compito	Matricola	Cognome Nome	Valutazione
A			

Si consideri il seguente schema relazionale relativo ad un noto gioco di caramelle:

Utente(id, nome, cognome, livello_corrente, episodio_corrente)

Episodio(id, nome, numero)

Livello(numero, idepisodio, tipo)

Amicizia(idutente, idutente)

LivelliUtente(numeroepisodio, idutente, datasuperamento, punteggio)

Un episodio è composto da tanti livelli, ogni livello è di un certo tipo descritto da una stringa alfanumerica. Per ogni utente si registra il livello dove è arrivato ed il corrispondente episodio.

1. Identificare le chiavi primarie ed esterne dello schema [1 punto];
2. Risolvere le seguenti query in algebra relazionale:
 - a. Trovare gli utenti che non hanno superato nessun livello con punteggio superiore a 1000 [4 punti];
 - b. Trovare l'utente che ha giocato in tutti gli episodi [4 punti];
 - c. Per ogni episodio trovare gli utenti che hanno raggiunto il punteggio massimo [4 punti];
3. Risolvere le seguenti query in SQL:
 - a. Trovare gli utenti che hanno superato un numero di livelli maggiore a quello medio calcolato su tutti gli utenti [4 punti];
 - b. Tra gli episodi con il numero massimo di livelli di tipo "A tempo", trovare quelli superati da almeno due utenti [5 punti];
4. Implementare un vincolo che imponga che il livello corrente dell'utente deve appartenere al corrispondente episodio [4 punti];
5. Implementare un trigger che dopo l'inserimento del superamento del livello nella tabella LivelliUtente aggiorna opportunamente gli attributi livello_corrente e episodio_corrente nella tabella Utente [4 punti].

Basi di Dati
A.A. 2017-2018
Prof. Alfredo Pulvirenti
Prova in Itinere 5 dicembre 2017

Compito	Matricola	Cognome Nome	Valutazione
C			

Si consideri il seguente schema relazionale relativo a un noto gioco di parole:

Giocatore(id, nome, cognome, livello)

Partita(idsfida,idpartita,numpartita,puntiG1,PuntiG2)

Sfida(idsfida,giocatore1,giocatore2,data,puntiG1Tot,PuntiG2Tot)

Torneo(id, datainizio, datafine, fine)

Partecipante(idtorneo, idpartecipante, punti, numeroPartiteTorneo)

Una Sfida è composta da 3 partite, ogni partita da un certo numero di punti ad ogni giocatore. Tali punti saranno sommati ai punti totali.

Ad ogni torneo partecipano tanti utenti, ogni utente può partecipare a un numero arbitrario di partite (diverse da quelle della sfida) dove collezionerà un numero di punti. Il livello di ogni utente è un valore numerico.

1. Identificare le chiavi primarie ed esterne dello schema [1 punto];
2. Rispondere alle seguenti query in algebra relazionale:
 - a. Trovare i giocatori con il livello minimo [4 punti];
 - b. Trovare i giocatori che hanno partecipato a tutti i tornei [4 punti];
 - c. Per ogni torneo trovare il giocatore che ha giocato più partite [4 punti];
3. Rispondere alle seguenti query in SQL:
 - a. Trovare i giocatori che hanno giocato un numero di partite maggiore di quello medio giocato da tutti gli utenti [4 punti];
 - b. Trovare i giocatori che hanno giocato con tutti i giocatori [3 punti];
 - c. Trovare i giocatori con il livello massimo che non hanno mai partecipato a tornei nel 2017 e che non hanno mai pareggiato con altri giocatori [5 punti];
4. Implementare un trigger che dopo la fine di una sfida (la sfida è conclusa quando viene inserita la terza partita nella tabella Partita) incrementa il livello del giocatore vincente a +5% e decrementa il livello del giocatore perdente a -10% [5 punti].

Basi di Dati
A.A. 2017-2018
Prof. Alfredo Pulvirenti
Prova in Itinere 5 dicembre 2017

Compito	Matricola	Cognome Nome	Valutazione
B			

Si consideri il seguente schema relazionale relativo alle segnalazioni di emergenza in una città:

Emergenza(id, idPersonaChiamante, data, ora, CAP, indirizzo, descrizione)

Persona(CF, nome, cognome)

Citta(id, nome, provincia)

Zona(idcitta, CAP)

Pericolosità(CAP, numeroChiamate, fasciaOraria)

Ogni cittadino può segnalare diverse emergenze. Ogni emergenza si riferisce ad una certa zona della città identificata da un CAP. Per ogni zona è definito un indice di pericolosità per fascia oraria determinato dal numero di chiamate.

1. Identificare le chiavi primarie ed esterne dello schema [1 punto];
2. Rispondere alle seguenti query in algebra relazionale:
 - a. Trovare le città che non hanno avuto emergenze segnalate [4 punti];
 - b. Trovare le città che hanno avuto emergenze in tutte le zone [4 punti];
 - c. Trovare la città con il massimo numero di chiamate (indipendentemente dal CAP) per ogni fascia oraria [4 punti];
3. Rispondere alle seguenti query in SQL:
 - a. Trovare le città che hanno avuto un numero di chiamate d'emergenza minore a quello medio di tutte le città [4 punti];
 - b. Trovare le persone che hanno effettuato delle chiamate d'emergenza per tutte le città [4 punti].
4. Implementare un vincolo di integrità che non consenta ad una persona di effettuare più di 5 chiamate d'emergenza nello stesso mese per la stessa città [4 punti].
5. Implementare un trigger che dopo l'inserimento di una emergenza aggiorni il valore di pericolosità nella tabella Pericolosità. Considerare l'attributo fasciaOraria con quattro valori: 00.00-6.00 , 6.00-12.00, 12.00-19.00, 19.00-24.00 [5 punti].

Basi di Dati
A.A. 2017-2018
Prof. Alfredo Pulvirenti
Prova in Itinere 5 dicembre 2017

Compito	Matricola	Cognome Nome	Valutazione
D			

Si consideri il seguente schema relativo alle rilevazioni di temperatura, precipitazioni e Co2 di una città:

Rilevatore(id, idcitta, CAP)

Citta(codiceIstat, nome, provincia)

Zona(codiceIstat, CAP)

RilevazioneOrariaTP(idrilevatore, data, ora, temperatura, precipitazioni)

RilevazioneOrariaCO2(idrilevatore, data, ora, concentrazioneCO2)

RevisioneRilevatore(id, dataRevisione, Descrizione)

In ogni città sono presenti diversi rilevatori (chiamati anche sensori), uno per ogni CAP. Ogni rilevatore effettua misure su Temperatura e Precipitazioni oppure sulla concentrazione di Co2. Ogni rilevazione è memorizzata nelle rispettive tabelle a diverse ore di ogni giorno dell'anno. Ogni rilevatore ogni 6 mesi necessita di una revisione programmata.

1. Identificare le chiavi primarie ed esterne dello schema [1 punto];
2. Rispondere alle seguenti query in algebra relazionale:
 - a. Trovare per ogni città la zona con la massima concentrazione assoluta di Co2 [4 punti];
 - b. Trovare le città che hanno rilevatori in tutte le zone [4 punti];
 - c. Trovare i rilevatori che non hanno riportato rilevazioni di Co2 [4 punti].
3. Rispondere alle seguenti query in SQL:
 - a. Trovare le città con la quantità totale di precipitazioni inferiore a quella media di tutte le città [4 punti];
 - b. Trovare le città che hanno avuto una concentrazione di Co2 maggiore di 1000 ppm in tutte le zone in tutte le rilevazioni del 2017 [4 punti].
4. Implementare un vincolo di integrità che non consenta di effettuare più di 5 rilevazioni di Co2 nella stessa giornata [4 punti].
5. Implementare un trigger che dopo l'inserimento di una rilevazione di temperatura e precipitazioni programmi automaticamente una revisione per il sensore 10 giorni dopo la rilevazione. La revisione è programmata solo se non esiste per lo stesso sensore una revisione avvenuta al più 6 mesi prima [5 punti].

Basi di Dati
A.A. 2017-2018
Prof. Alfredo Pulvirenti
Prova in Itinere 5 dicembre 2017

Compito	Matricola	Cognome Nome	Valutazione
E			

Si consideri il seguente schema relazionale relativo a dei viaggi:

Luogo(id, stato, città)

Viaggio(id, datapartenza, datarientro)

Partecipante(idp, idv)

Tappa(idviaggio, idtappa, nome, numero_giorni, idluogo)

Persona(id, nome, cognome, frequentTraveler)

Ogni persona può partecipare a uno o più viaggi. Ogni viaggio ha una data di inizio e una di fine, è composto da più tappe, ognuna delle quali prevede un certo numero di giorni. Ogni persona che ha fatto un certo numero di viaggi è eletto a frequent traveler.

1. Identificare le chiavi primarie ed esterne dello schema [1 punto];
2. Rispondere alle seguenti query in algebra relazionale:
 - a. Trovare i luoghi che non sono stati visitati da nessun viaggio [4 punti];
 - b. Trovare i luoghi visitati in tutti i viaggi [4 punti];
 - c. Trovare le persone che hanno partecipato ad almeno 2 viaggi con almeno 2 tappe [4 punti];
3. Rispondere alle seguenti query in SQL:
 - a. Trovare le persone che hanno partecipato ad un numero di viaggi maggiore al numero medio di tutte le persone [4 punti];
 - b. Trovare le persone che hanno partecipato a tutti i viaggi della durata di 15 giorni [5 punti];
4. Implementare un vincolo di integrità che non consenta di avere più di 5 tappe per ogni viaggio [4 punti].
5. Implementare un trigger elegga a frequentTraveler le persone che hanno fatto più di 10 viaggi nello stesso anno [4 punti].