

Dato lo schema:

Escursione(id, titolo, descrizione, durata, difficoltà, costo)

DataEscursione(id, data, idescursione, id guida)

Partecipante(idpartecipante, idescursione)

Persona(id, nome, cognome)

1. Indicare le chiavi primarie ed esterne dello schema e le relazioni esistenti tra le tabelle .
2. Rispondere alle seguenti query in algebra relazionale ed SQL:
 - a. Trovare le escursioni (indicando titolo, descrizione e difficoltà) che hanno un costo massimo
 - b. Trovare i partecipanti (dando nome e cognome in output) che hanno partecipato a tutte le escursioni
 - c. Trovare le guide che non hanno mai partecipato ad escursioni di difficoltà massima
 - d. Trovare le coppie di persone che hanno partecipato sempre alle stesse escursioni
 - e. Dire ogni accompagnatore quante escursioni ha guidato;
3. Definiti 5 livelli di difficoltà per le escursioni. Creare un vincolo di integrità che garantisca che ogni accompagnatore prima di guidare una escursione di livello x abbia guidato almeno 5 escursioni di livello $(x-1)$

Si consideri un database SQL per memorizzare un grafo direzionato.

- Il database contiene un'unica tabella:

ARCO(n1,n2).

La tupla (X,Y) in questa tabella codifica il fatto che c'è un arco diretto dal nodo con l'identificatore X a quello con l'identificatore Y. Non ci sono duplicati. Si supponga che ogni nodo nel grafo fa parte di almeno un arco.

1. Scrivere una query SQL per trovare il nodo con il più alto out-degree.
2. Se (non) hai usato per il punto 1 la Group By, scrivi la stessa query senza (con) la Group By.
3. Modificare la soluzione per 1 e 2 e trovare gli identificatori con il più alto "in-degree".
4. Scrivere una query per trovare un cammino che va dal nodo X al nodo Y.
5. Scrivere una query SQL per trovare l'out-degree medio dei nodi nel grafo.