



POLITECNICO MILANO 1863

“FUNDRAISER TEAMTREES.ORG”

PROGETTO DEL CORSO DI SISTEMI INFORMATIVI

Ingegneria Gestionale | Scaglione A-M | Prof. Plebani Pierluigi

Anno accademico 2019/2020

GRUPPO 59 formato da:

Di Grandi Daniele, matricola: 887284

Hartog Alice, matricola: 865928

TRACCIA:

Lo Youtuber americano Jimmy Donaldson, conosciuto come “MrBeast”, aveva intenzione di fare qualcosa di memorabile per celebrare il raggiungimento dei 20.000.000 di iscritti al suo canale, così i suoi fans l’hanno sfidato a piantare 20.000.000 di alberi nel mondo.

Raccolta la sfida, ha iniziato, in collaborazione con il sito “Teamtrees.org”, la raccolta fondi più grande che si sia mai vista in questo ambito: 20.000.000 di dollari entro il 01/01/2020 (1 dollaro = 1 albero piantato).

Molte personalità di spicco nel mondo delle imprese hanno accolto molto positivamente questa iniziativa, donando personalmente ingenti quantità di denaro: si pensi a 1.000.000 di dollari donati da Elon Musk (CEO di Tesla e SpaceX), a 1.000.001 di dollari donati da Tobias Lütke (CEO di Shopify), 200.000 dollari donati da Susan Wojcicki (CEO di Youtube) e molti altri.

Ci siamo quindi immaginati come può essere progettata una base di dati per la gestione di un progetto “dinamico” e temporaneo come una raccolta fondi, nello specifico, quella avviata da Jimmy Donaldson.

In primo luogo si vogliono salvare i dati delle persone che prenderanno parte al progetto.

Le persone possono essere solo donatori e/o piantatori. Per ogni persona, si vuole salvare la mail, il nome, il cognome, il telefono, l’età (opzionale), il sesso (opzionale) e lo Stato di appartenenza.

Inoltre dei piantatori si vuole salvare anche l’associazione a cui appartengono, la Nazione in cui si trovano al momento (per decidere al meglio gli spostamenti dei vari piantatori tra le varie Nazioni del mondo) e si vuole tenere traccia degli eventuali alberi che hanno piantato.

Gli alberi sono contrassegnati da un ID univoco, saranno piantati in una certa foresta nel mondo, appartengono ad una certa specie e sono stati donati da un certo donatore.

La piattaforma funziona infatti per donazioni: ogni donatore può donare una certa quantità di denaro, ricordando che ad ogni dollaro donato equivale 1 albero piantato.

Per ogni donazione si salva un ID univoco, la data e l’ora in cui è stata effettuata, l’importo donato (in €), il nome visualizzato nel feed delle donazioni (opzionale) e il messaggio visualizzato (opzionale).

Per ogni foresta si vuole salvare un ID univoco, il nome, la nazione sulla quale è principalmente localizzata, oltre all’estensione che ricopre (in Km^2), al livello di rischio ambientale (scala di valori da 1 a 5 con 1-> rischio basso, 5-> rischio critico) e alle specie native di alberi.

Infatti le specie degli alberi donati saranno scelte in modo automatico in base alla foresta cui sono destinati (nasce quindi il bisogno di tenere traccia delle specie native di ogni foresta, sapendo anche che ogni specie può essere nativa di più foreste), e verrà data priorità alle foreste con un livello critico di rischio ambientale.

Per ogni specie di albero, si vuole salvare il nome della specie, la famiglia a cui appartiene, l’altezza media dell’albero adulto e la quantità in stock presente nei magazzini di Teamtrees.

Sia i fornitori degli alberi che Teamtrees effettuano le spedizioni verso le foreste utilizzando dei bancali.

I fornitori possono decidere a chi spedire: direttamente ad una foresta (se per esempio il fornitore si trova geograficamente vicino alla foresta destinataria), oppure ai magazzini di Teamtrees.

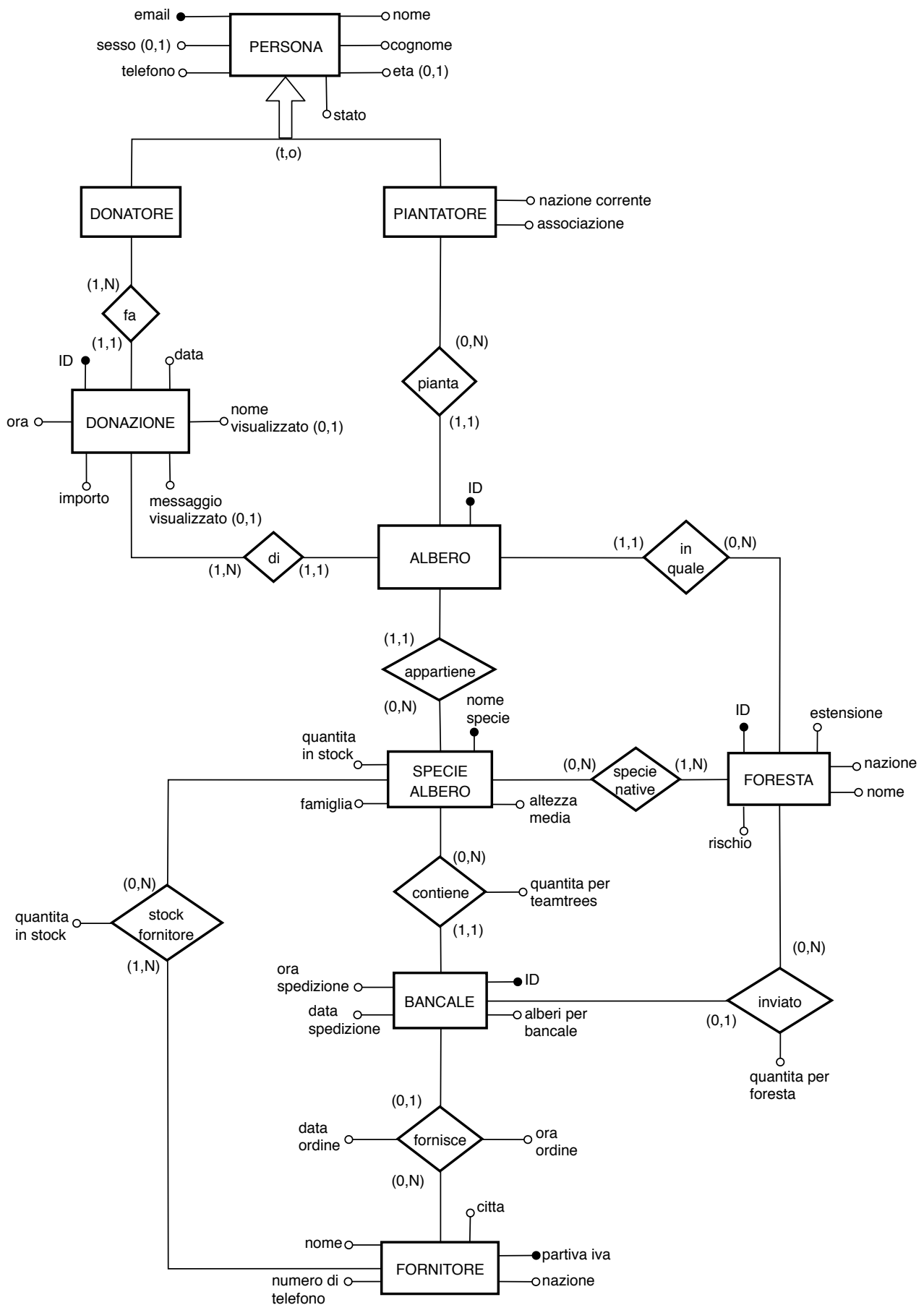
Ogni banchale può contenere una sola specie di albero e si vuole salvare un ID univoco, la data e l’ora in cui è stato spedito, il fornitore dal quale arriva (solo se a spedire è stato il fornitore, altrimenti se la spedizione è effettuata dai magazzini di Teamtrees verso le foreste, allora sarà un campo vuoto), e il numero di alberi che nominalmente può contenere. Se il banchale è stato spedito da un fornitore, si tiene traccia anche della data e ora dell’ordine.

Si vuole infine tenere traccia, in base alla destinazione finale (Teamtrees o foreste), della quantità di alberi presenti veramente per ogni banchale spedito.

Di ogni fornitore si vuole tenere traccia della partita IVA, del nome, della città, della nazione di appartenenza (in modo da capire quali fornitori sono vicini a quali foreste) e del numero di telefono.

In più si vuole anche tenere traccia dello stock delle specie di alberi nei magazzini di ogni fornitore.

SCHEMA CONCETTUALE:



SCHEMA LOGICO:

PERSONA (email, nome, cognome, telefono, eta(0, 1), sesso(0,1),
associazione(0,1), donatore, piantatore, stato, nazionecorrente(0,1))

DONAZIONE (ID, emaildonatore, data, ora, importo, nomevisualizzato(0,1),
messaggiovisualizzato(0,1))

FORESTA (ID, nome, nazione, estensione, rischio)

ALBERO (ID, IDdonazione, emailpiantatore, IDforesta, nomespecie)

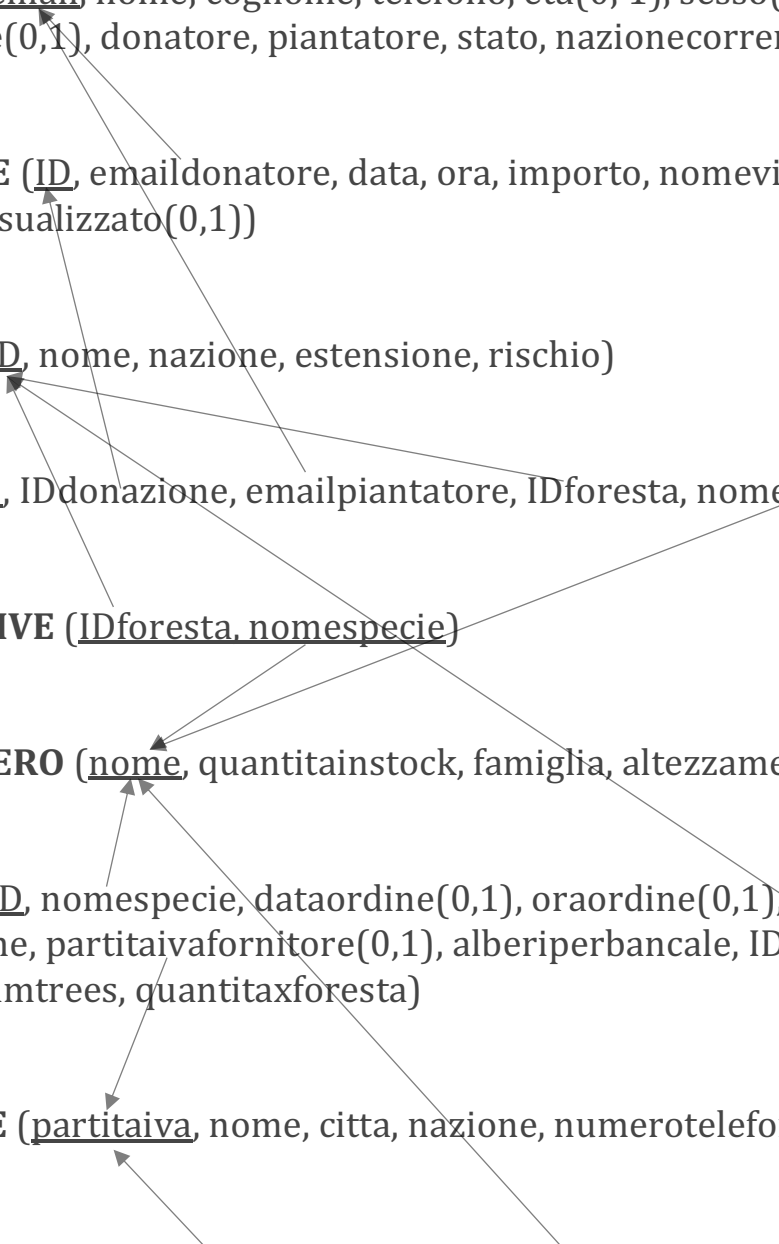
SPECIENATIVE (IDforesta, nomespecie)

SPECIEALBERO (nome, quantitainstock, famiglia, altezzamedia)

BANCALE (ID, nomespecie, dataordine(0,1), oraordine(0,1), dataspedizione,
oraspedizione, partitaivaforitore(0,1), alberiperbancale, IDforesta(0,1),
quantitaxteamtrees, quantitaxforesta)

FORNITORE (partitaiva, nome, citta, nazione, numerotelefono)

STOCKFORNITORE (partitaivaforitore, nomespecie, quantitainstock)



QUERY:

- Una query a scelta che contenga almeno una operazione di selezione (ovvero la presenza della clausola WHERE):

Trovare nome, città, nazione, numero di telefono e quantità presente in magazzino dei fornitori che hanno in stock la specie di albero “araucaria angustifolia”.

```
USE Fundraiser_Teamtrees;
SELECT f.nome, f.città, f.nazione, f.numerotelefono, s.quantitainstock
FROM fornitore AS f JOIN stockfornitore AS s ON (f.partitaiva = s.partitaivafornitore)
WHERE s.nomespecie = 'araucaria angustifolia' AND s.quantitainstock > 0;
```

- Una query che contenga il costrutto GROUP BY con HAVING:

Trovare l'associazione che ha piantato più alberi utilizzando piantatori del luogo (quindi provenienti dallo stato in cui si trova la foresta dove hanno piantato un albero) e restituire anche il nome dello stato.

```
USE Fundraiser_Teamtrees;
SELECT p.associazione, p.stato
FROM persona AS p JOIN albero AS a ON (p.email = a.emailpiantatore) JOIN foresta AS f ON (a.idforesta = f.id)
GROUP BY p.associazione, p.stato, f.nazione
HAVING p.stato = f.nazione
ORDER BY COUNT(*) DESC
LIMIT 1;
```

- Una query che faccia uso di query annidate (senza l'utilizzo di costrutti “avanzati”):

Top donatore/i del giorno: trovare il nome visualizzato (eliminare i NULL), la data della donazione e l'importo donato di chi, per ogni giorno, ha donato l'importo maggiore.

```
USE Fundraiser_Teamtrees;
SELECT d1.data, d1.nomevisualizzato, d1.importo
FROM donazione AS d1
WHERE d1.nomevisualizzato IS NOT NULL AND d1.importo =
(SELECT MAX(d2.importo)
FROM donazione AS d2
WHERE d2.data=d1.data);
```

- Una query che faccia uso di uno dei costrutti "avanzati" (IN, NOT IN, ANY, ALL, ...):

Trovare nome e nazione delle foreste a rischio almeno 4 che non hanno mai ricevuto alberi in consegna.

```
USE Fundraiser_Teamtrees;
SELECT f1.nome, f1.nazione
FROM foresta AS f1
WHERE f1.rischio > 3 AND f1.nome NOT IN
(SELECT DISTINCT f2.nome
FROM foresta AS f2 JOIN bancali AS b ON (f2.id = b.idforesta)
WHERE b.quantitaxforesta > 0);
```

- Una query che faccia uso di una vista:

Trovare nome specie, nome foresta e le rimanenze (alberi inviati alle foreste sia dai fornitori che da Teamtrees ma non ancora piantati) di ogni specie albero in ogni foresta.

```
USE Fundraiser_Teamtrees;
DROP VIEW IF EXISTS sommabancali;
DROP VIEW IF EXISTS sommaforesta;
```

```
CREATE VIEW sommabancali(idforesta, nomespecie, somma) AS
(SELECT b.idforesta, b.nomespecie, SUM(quantitaxforesta)
FROM bancali AS b
WHERE b.idforesta IS NOT NULL
GROUP BY b.idforesta, b.nomespecie);
```

```
CREATE VIEW sommaforesta(idforesta, nomespecie, somma) AS
(SELECT a.idforesta, a.nomespecie, COUNT(*)
FROM albero AS a
GROUP BY a.idforesta, a.nomespecie);
```

```
SELECT f.nome, sb.nomespecie, sb.somma - sf.somma
FROM sommabancali AS sb JOIN sommaforesta AS sf ON (sb.nomespecie = sf.nomespecie) JOIN foresta as f
ON (sb.idforesta = f.id)
WHERE sb.idforesta = sf.idforesta;
```