Traccia d'Esame 1: Analisi delle Scuderie Dominanti

Obiettivo: Costruire un grafo che rappresenti le rivalità tra scuderie (costruttori) in una data stagione, basandosi sulle vittorie.

PUNTO 1

a. L'utente seleziona dal corrispondente menù a tendina l'anno di una stagione (tabella seasons) in cui si sia disputato un campionato.

b. Premendo sul tasto "Rivalità Scuderie", l'applicazione costruisce un grafo dove: \* I vertici sono le scuderie (constructors) che hanno vinto almeno una gara nella stagione selezionata. \*

select distinct constructorId

from results

where position =1

and raceId in (select raceId from races where year=2009);

Un arco orientato va dalla scuderia A alla scuderia B se la scuderia A ha totalizzato più punti totali (campo points nella tabella constructorresults) rispetto alla scuderia B in almeno una gara in cui entrambe hanno partecipato nella stagione selezionata. Il peso dell'arco è pari al numero di gare in cui questa condizione si è verificata.

select count(distinct *c1*.raceId)

from constructorresults *c1*,constructorresults *c2*

where *c1*.constructorId= 23 and *c2*.constructorId =7

and *c1*.raceId= *c2*.raceId and *c1*.points>*c2*.points

and *c2*.raceId in (select raceId from races where year=2009);

c. Costruito il grafo, l'applicazione visualizza: \* Il numero di nodi e di archi del grafo. \* La scuderia con il maggior numero di "vittorie" (ovvero, con il maggior grado uscente del grafo). == GRAFO.OUT\_DEGREE(NODO) \* La scuderia più "battuta" (ovvero, con il maggior grado entrante del grafo). ===GRFO.IN\_DEGREE(NODO)

\* Il numero di nodi e di archi del grafo.

\* La scuderia con il maggior numero di "vittorie" (ovvero, con il maggior grado uscente del grafo).

\* La scuderia più "battuta" (ovvero, con il maggior grado entrante del grafo).

PUNTO 2 (Avanzato/Ricorsione)

a. L'utente inserisce un valore intero K.

c. Visualizzare il percorso trovato e la sua lunghezza.

Traccia d'Esame 2: Connessioni tra Piloti e Circuiti

Obiettivo: Analizzare i circuiti preferiti dai piloti e le "specializzazioni" dei piloti in determinati circuiti.

PUNTO 1

a. L'utente seleziona dal corrispondente menù a tendina l'anno di una stagione (tabella seasons).

b. Premendo sul tasto "Piloti e Circuiti", l'applicazione costruisce un grafo bipartito dove:

\* Ci sono due tipi di vertici: i piloti (drivers) che hanno partecipato ad almeno una gara nella stagione selezionata e i circuiti (circuits) su cui si sono svolte gare in quella stagione.

\* Un arco non orientato collega un pilota ad un circuito se il pilota ha corso su quel circuito nella stagione selezionata.

\* Il peso dell'arco è pari al numero di volte che il pilota ha ottenuto una posizione (position nella tabella results) tra i primi 3 su quel circuito in quella stagione.

c. Costruito il grafo, l'applicazione visualizza:

\* Il numero di nodi (suddivisi per tipo: piloti e circuiti) e di archi.

\* Per ogni circuito, il pilota che ha ottenuto il punteggio maggiore su quel circuito (somma dei pesi degli archi uscenti dal circuito e collegati ai piloti).

\* Per ogni pilota, il circuito su cui ha ottenuto il punteggio maggiore.

Traccia d'Esame 3: Dinamiche di Gara e Affidabilità

Obiettivo: Esplorare le relazioni tra i piloti che hanno concluso le gare e quelli che si sono ritirati, e le cause di ritiro.

PUNTO 1

a. L'utente seleziona dal corrispondente menù a tendina l'anno di una stagione (tabella seasons).

b. Premendo sul tasto "Affidabilità Piloti", l'applicazione costruisce un grafo dove:

\* I vertici sono tutti i piloti (drivers) che hanno partecipato ad almeno una gara nella stagione selezionata.

SELECT DISTINCT driverId

FROM results

WHERE raceId in (select raceId from races where year = %s)

\* Un arco orientato va dal pilota A al pilota B se, in una qualsiasi gara della stagione selezionata, il pilota A ha terminato la gara (status 1 nella tabella status) mentre il pilota B si è ritirato (qualsiasi altro statusId diverso da 1).

SELECT COUNT(DISTINCT r1.raceId)

FROM results r1, results r2

WHERE r1.raceId=r2.raceId and r1.statusId=1 and r2.statusId!=1

And raceId in (select raceId from races where year = %s)

\* Il peso dell'arco è pari al numero di gare in cui questa coppia (A termina, B si ritira) si è verificata.

c. Costruito il grafo, l'applicazione visualizza:

\* Il numero di nodi e di archi.

\* Il pilota che ha il maggior numero di "ritirati" dietro di sé (massimo grado uscente).

\* Il pilota che si è ritirato più volte mentre altri hanno terminato (massimo grado entrante).

\* Una lista delle 5 cause di ritiro (status) più comuni tra i piloti che sono "bersaglio" di archi nel grafo.

**SELECT** **COUNT**(**DISTINCT** raceId) **as** *occorrenze*, statusId

**FROM** results

**WHERE** statusId != 1 **and** raceId **in** (**select** raceId **from** races **where** **year** = 2009)

**GROUP** **BY** statusId

**Order** **by** *occorrenze* **desc**

**limit** 5;

PUNTO 2 (Avanzato/Ricorsione)

a. L'utente seleziona un pilota specifico dal menù a tendina.

b. L'applicazione deve trovare, usando una funzione ricorsiva, la catena più lunga di piloti in cui ogni pilota successivo si è ritirato in una gara in cui il precedente ha terminato, partendo dal pilota selezionato. (Esempio: Pilota A termina, Pilota B si ritira; Pilota B termina, Pilota C si ritira; Pilota C termina, Pilota D si ritira, ecc., ma considerando la relazione del grafo Pilota A -> Pilota B).== trova la più lunga, devo usare i successors()

c. Visualizzare la catena trovata e la sua lunghezza.

**raccia d'Esame 6: Gerarchie di Prestazione su Circuiti Comuni**

**Obiettivo:** Costruire un grafo che rappresenti le gerarchie di prestazione tra piloti su circuiti su cui entrambi hanno avuto un'esperienza significativa, indipendentemente dalla loro partecipazione alla stessa gara.

**PUNTO 1**

a. L'utente seleziona dal corrispondente menù a tendina un **intervallo di anni** (es. Anno Inizio, Anno Fine) per l'analisi (tabella seasons).

b. Premendo sul tasto "Gerarchie Circuiti", l'applicazione costruisce un grafo orientato dove: \* I **vertici** sono tutti i **piloti** (drivers) che hanno partecipato ad almeno una gara nell'intervallo di anni selezionato.

\* Un **arco orientato** va dal Pilota A al Pilota B se esiste **almeno un circuito** (circuitId) nell'intervallo di anni selezionato, su cui Pilota A ha dimostrato una prestazione media (misurata in position, quindi un valore numericamente più basso è migliore) **strettamente migliore** di Pilota B, considerando solo le gare in cui *entrambi hanno completato la gara* (status = 'Finished'). \* Per considerare la prestazione "significativa", entrambi i piloti (A e B) devono aver **completato almeno 3 gare** su quel circuito specifico nell'intervallo di anni.

\* Il **peso dell'arco** è la **somma delle differenze positive** (avg\_pos\_B - avg\_pos\_A) per tutti i circuiti in cui Pilota A ha soddisfatto la condizione di "prestazione migliore" rispetto a Pilota B. Un peso maggiore indica una dominanza più marcata o su più circuiti.

c. Costruito il grafo, l'applicazione visualizza: \* Il numero totale di nodi e il numero totale di archi. \* Il pilota con il **maggior grado uscente** (ovvero, colui che ha superato in prestazione più piloti su circuiti comuni). \* Il pilota con il **maggior grado entrante** (ovvero, colui che è stato superato in prestazione da più piloti su circuiti comuni). \* Il pilota "più resiliente": quello con il maggior numero di gare terminate (status = 'Finished') su circuiti storici (definiti come nel Lab13: circuiti con più di 50 gare totali nella storia della F1) nell'intervallo di anni selezionato.

**PUNTO 2** (Avanzato/Ricorsione)

a. L'utente seleziona un pilota di partenza (P\_start) e un valore intero L (lunghezza massima del percorso in numero di archi). b. L'applicazione deve trovare, utilizzando un **algoritmo ricorsivo (come una DFS)**, il cammino **più lungo** che parte da P\_start e che: \* Non supera L archi (e quindi L+1 nodi diversi). Il cammino deve essere semplice (senza ripetere piloti). \* Ogni arco nel percorso deve avere un peso superiore a una soglia T (ad esempio, T = 2.5, un valore fisso o inseribile dall'utente) per indicare una "dominanza significativa". c. Visualizzare il cammino trovato, la sua lunghezza (numero di nodi) e il peso totale del cammino (somma dei pesi degli archi).