**Discorso Presentazione Data Management**

**PARTE DI GIO [x minuti totali] Tempi: 9:26, 9:05, 8:44, 9:01, 8:57,**

**Slide 1: Titolo Slide [5 sec]**

**Buongiorno. Il nostro progetto prende il nome di Competitive pokémon graph database.**

**Slide 2: Business Understanding [1.11 minuti slide, 1.16 minuti totale]**

**L’idea** alla base del progetto è stata quella di realizzare un database a grafo che racchiuda tutte le informazioni utili ad addentrarsi nel mondo delle lotte competitive dei videogiochi della serie Pokémon, con particolare riferimento al regolamento chimato Video Game Championship 2022 Series 12, che definisce il formato ufficiale in vigore nei tornei e negli eventi che si svolgono nel periodo Febbraio – Agosto 2022 e valido per i Pokémon World Championships che si svolgeranno a Londra in agosto.

**L’obiettivo** è quello di ottenere uno strumento utile come supporto ai giocatori del competitivo, inglobando quindi: statistiche e descrizioni relative a Pokémon, mosse, strumenti, abilità e altre meccaniche e entità di gioco; le relazioni che legano questi elementi di gameplay all’interno dei videogiochi e soprattutto i dati rilevati durante i match competitivi svolti a Marzo 2022 sulla piattaforma online Showdown!, che ci hanno permesso di estrarre le relazioni che legano ogni Pokémon con i compagni di squadra, le mosse, gli strumenti, le abilità e le nature con cui risultano più frequentemente abbinati all’interno dei team schierati.

~~Data la forte enfasi posta sulle relazioni tra gli elementi di gioco si è scelto quindi di realizzare un~~ **~~database a grafo~~**~~.~~

**Slide 3: DATA ACQUISITION:API & WEB SCRAPING [0.10 minuti slide, 1.20 minuti totale]**

Il processo di Data Acqusition è stato effettuato mediante l’utilizzo di una API e l’applicazione di tecniche di Web Scraping

**Slide 4: API: PokéAPI [1.08 minuti slide, 2.28 minuti totale]**

La Web API utilizzata è fornita pubblicamente dal sito PokéAPI, che permette mediante un’interfaccia RESTful API l’accesso a dati relativia tutti gli elementi e le meccaniche dei videogiochi Pokémon.

Il **processo** ha visto: l’individuazione degli API endpoints delle risorse relative all’entità di interesse (ad esempio, per l’entità Pokémon~~, è stato individuata la porzione di endpoint relativa genericamente ai Pokémon~~); la successiva creazione di una funzione che passi in modo automatizzato gli endpoints specifici al metodo .get() della libreria Python Requests, per inviare le GET request al server (~~ad esempio, nel caso dell’entità Pokémon la funzione si occupava di completare a ogni iterazione il path generico con l’id relativo al Pokémon specifico, inviare la get request, estrarre i dati mediante il metodo .json del request object e aggiungerli a una lista~~); La funzione ritorna poi, per ogni entità, una lista di dizionari (~~nel caso dei Pokémon si avrà una lista dove ogni elemento è il dizionario relativo al Pokémon specifico)~~, la quale viene convertita in un unico dizionario di dizionari e salvata in formato .json, mediante l’utilizzo della funzione json.dump sulla lista. ~~(si avrà quindi un unico dizionario di dizionari per l’interà entità Pokémon)~~. ~~Conversione e salvataggio sono avvenuti contestualmente usando json.dump sulla lista.~~

**Slide 5: API: dataset ottenuti [0.24 minuti slide, 2.52 minuti totale]**

Sono stati ottenuti i 5 dataset: Species API; Items API; Abilities API, Moves API e Types API, contenenti rispettivamente le specie di Pokémon, gli Oggetti, le abilità, le mosse e i tipi introdotti nella serie di videogiochi Pokémon fino a **Pokémon Spada e Scudo** (~~ovvero la versione su cui si basa questa stagione attuale del competitivo~~), accompagnati da ~~peculiari~~ attributi qualitativi e quantitativi che ne descrivono le caratteristiche e le relazioni con gli altri elementi di gioco.

~~Definizione di cosa siano le mosse, oggetti, abilità etc ?~~

**Slide 6: Web Scraping [2:22 minuti slide, 5.15 minuti totale]**

La tecnica del Web Scraping è stata utilizzata per acquisire dati da diverse sorgenti~~,~~ **siti target** sono stati: **Pokémon Database**: ovvero uno dei fansite ~~più popolari~~ ~~a tema videogiochi Pokémon~~, che raccoglie informazioni aggiornate relative ~~alle statistiche e alle meccaniche dei~~ videogiochi della serie principale, proponendole in buona parte in formato tabellare. **Pikalytics**: ~~un altro fansite~~ che fornisce però analisi e statistiche di utilizzo relative ai match competitivi disputati sulle diverse piattaforme e nei diversi formati lotta disponibili. ~~Per ogni Pokémon sono generati \textit{spreadsheet} contenenti informazioni riguardo i compagni di squadra, gli oggetti, le mosse, le abilità e le statistiche più popolari. Il sito si aggiorna ogni primo del mese, con i dati mostrati che fanno riferimento alla totalità dei match registrati durante il mese precedente~~. **Serebii.net** altro fansite contenente notizie, ~~dati in formato tabellare~~ e informazioni aggiornate relative al mondo Pokémon nel suo complesso **VictoryRoad**: un sito gestito da organizzatori e giudici del Pokémon VGC che fornisce informazioni, regolamenti e know-how relativi alle lotte competitive. e infine **Bulbapedia**, un’enciclopedia community-driven a tema Pokémon ~~che costituisce una delle risorse sul tema più complete e note del web.~~

ll **processo di Web Scraping** é stato effettuato in **Python**, con metodi e librerie ~~utilizzate~~ che variano in funzione della struttura delle pagine web ~~nei diversi siti menzionati~~. Tipicamente, il processo ha visto una fase iniziale di esplorazione del sito target con successiva ispezione della struttura del documento HTML con il tool apposito ~~(del Document Object Model del sito) mediante l’utilizzo del developer tool apposito del browser~~. La fase successiva ha previsto l'utilizzo della libreria **Python Requests**, e in particolare del metodo get(), al fine di inviare le GET request delle pagine target e ottenere gli specifici Response Object. Il parsing dei documenti HTML estratti dai Response Object e la loro successiva navigazione al fine di estrarre i dati di interesse è stata effettuata mediante le funzioni e i metodi di **Beautiful Soup**. ~~In particolare il processo di acquisizione ha visto lo sviluppo di \textit{script} che permettessero, in modo automatizzato, la navigazione del BeautifulSoup object, la ricerca e l'acquisizione dei dati di interesse, l'organizzazione degli stessi~~ e i dati sono stati organizzati all'interno di dataframe **Pandas** e ~~il loro salvataggio~~ salvati come file .csv e .JSON. In alcuni dei siti citati ~~(Pokémon Database, Serebii.net e Bulbapedia},~~ parte dei dati sono organizzati ~~in formato tabellare~~ all'interno delle **HTML Tables**~~. In questo caso~~ le quali sono state ~~tabelle sono state~~ lette facilmente~~, e salvate in \textit{dataframe},~~ utilizzando la funzione pandas.read\\_html(). Inoltre, l'acquisizione ~~dei dati relativi al videogioco competitivo~~ dal sito \textit{Pikalytics}, ha richiesto~~, in fase iniziale,~~ l'utilizzo della libreria **Selenium** ~~con \textit{Microsoft Edge WebDriver}\cite{EdgeDriver~~}, con lo scopo di simulare l'azione di scrolling nella pagina ~~fino al fondo della pagina principale~~ e permettere il caricamento completo ~~di tutti i dati da acquisire~~ del documento HTML.

**Slide 7: Scraping da Pokémon Database [0.42 minuti slide, 6 minuti totale]**

I dati ottenuti da Pokémon Database **coprono** tutti gli aspetti tecnici e le statistiche relative agli elementi e alle meccaniche di gioco ~~dei videogiochi della serie principale~~, lo **scopo** dell’acquisizione da questa fonte è stato integrare quanto ottenuto tramite l'API, con attributi e record aggiuntivi e valori più affidabili per le statistiche tecniche (~~il sito presenta una qualità maggiore rispetto all’API dal punto di vista della currency, aggiornato più frequentemente~~). Da Pokémon Database sono stati ottenuti un totale di **5 dataset e 1 tabella ponte**. Quest’utlima è la **tabella a doppia entrata Type Chart** ~~è una tabella ponte~~ che pone in relazione il tipo elementale della mossa usata ~~in battaglia~~ da un Pokémon attaccante con il tipo elementale del Pokémon che subisce la mossa~~. Tale~~ relazione definisce l'efficacia della mossa in battaglia, ~~determinando un eventuale incremento o riduzione del danno inflitto, rappresentato da un moltiplicatore il cui valore è rappresentato nelle celle della stessa.~~

**Slide 8: Scraping da Pokémon Database [0.15 minuti slide, 6.15 minuti totale]**

I 5 dataset ottenuti: Complete Pokémon Pokédex; Items; Abilities, Moves e Natures, ~~contenenti~~ contengono rispettivamente i Pokémon, gli Oggetti, le abilità, le mosse e le nature introdotti fino all’ultima generazione di videogiochi, accompagnati da ~~peculiari~~ attributi qualitativi e quantitativi ~~che ne descrivono le caratteristiche~~.

**Slide 9: Scraping da Pikalytics [0.31 minuti slide, 6.46 minuti totale]**

Il processo di Web Scraping su Pikalytics ha permesso l'acquisizione dei dati relativi ai match competitivi svolti a Marzo 2022, con particolare riferimento ~~a~~ quelli disputati online sulla piattaforma **Pokémon Showdown!**, seguendo il regolamento ufficiale di **VGC 2022 Series 12**. È stato ottenuto **un dataset e 5 tabelle ponte**.

Il dataset **Pokémon Usage**, associa ad ogni Pokémon utilizzato a Marzo 2022 durante almeno un match competitivo, il rank e le frequenze assolute e relative mensili di utilizzo nei team schierati.

**Slide 10: Scraping da Pikalytics** **[0.15 minuti slide, 7.01 minuti totale]**

Le 5 tabelle ponte (indicale) ottenute definiscono invece le relazioni che legano i Pokémon ai compagni di squadra, alle mosse, agli oggetti, alle abilità e alle nature a cui vengono più frequentemente abbinati nei team rilevati.

**Slide 11: Scraping Su VictoryRoad [0.46 minuti slide, 7.47 minuti totale]**

Il ~~processo di~~ Web Scraping su VictoryRoad ha avuto come target specifico la pagina contenente il **regolamento ufficiale** e le informazioni generali riguardanti la stagione 2020-2022 del competitivo~~, che terminerà con il mondiale di londra 2022,~~ con particolare riferimento al formato di lotta ufficiale attuale ~~che prende il nome di~~ **VGC 2022 Series 12**. Il processo ha portato ad ottenere un totale di **7 liste**, le quali contengono i riferimenti ai Pokémon Banned (~~non ammessi dal regoalmento~~), Restricted (per cui è permessa la presenza di al piú un Pokémon della lista nella squadra), Permitted (~~ovvero ammessi dal regolamento~~) e **Allowed Gigantamax**. ~~In particolare quest’ultima lista~~ che contiene i ~~nomi dei~~ Pokémon per cui è permessa una peculiare trasformazione temporanea in battaglia detta appunto Gigantamax.

**Slide 12: Scraping Su serebii.net [1 minuti slide, 8.47 minuti totale]**

Alcune delle liste del regolamento ottenute da victoryroad utilizzano come riferimento ai Pokémon degli ID peculiari usati per identificare i Pokémon solo all’interno dei bestiari presenti nei singoli videogiochi, chiamati **Pokédex**.

In particolare, ~~si fa riferimento a parte dei~~ **~~Pokémon permessi~~** ~~attraverso il loro ID nel Pokédex Nazionale (che contiene tutti i Pokémon mai presentati nella storia del franchise e ottenuto precedentemente dallo scraping di \textit{Pokémon Database}), per cui è nota una corrispondenza tra ID e nome del Pokémon, mentre la restante parte attraverso~~ ~~il loro ID in tre Pokédex detti~~ **~~regionali~~**~~, che descrivono le 3 diverse regioni geografiche del videogioco Pokémon Spada e Scudo~~. **Obiettivo** dello scraping su Serebii.net è stato quello di ottenere i **tre Pokédex regionali** che descrivono le 3 diverse regioni geografiche del videogioco Pokémon Spada e Scudo (~~chiamati~~ **~~Pokédex di Galar, di Isle of Armor e di The Crown Tundra~~**) al fine di permettere, in seguito, di associare ad ogni ID della lista corrispondente il nome completo di un Pokémon, che permette di identificarlo univocamente. ~~I tre dataset ottenuti associano ad ogni record relativo a un particolare Pokémon regionale l’ID regionale e una serie di attributi qualitativi e quantitativi che ne descrivono le caratteristiche~~. ~~Questo perché come poi vedremo si è scelto poi di utilizzare il nome completo del Pokémon al fine di identificarlo univocamente.~~

**Slide 13: Scraping Su Bulbapedia [0.45 minuti slide, 9:30 minuti totale]**

Come illustrato precedentemente, ~~una meccanica peculiare dell’attuale stagione di del gioco competitivo prevede che~~ alcune specie di Pokémon siano in grado di trasformarsi, temporaneamente in battaglia, nella propria **variante Gigantamax** una versione alternativa dalle dimensioni e dalle statistiche base maggiorate. Ogni Pokémon in forma Gigantamax possiede inoltre una mossa esclusiva che prende il nome di **G-Max Move**. **Obiettivo** di questa ultima fase di scraping su bulbapedia è stato ottenere una dataset dei Pokémon in grado di avere forma Gigantamax e uno delle G-Max Move a essi associate, ~~con relativi attributi qualitativi e quantitativi che ne descrivano effetti e statistiche~~, con lo scopo poi di **arricchire**, con i record ottenuti, i dataset principali dei Pokémon e delle mosse.

**PARTE DI GIAN Tempi: 9:00**

**Slide 14: Data Cleansing, Integration ed Enrichment [5 secondi slid]**

**A questo punto siamo passati alla fase di pulizia e integrazione dei dataset.**

**Slide 15: Obiettivi [25 secondi slide, 30 secondi totali]**

**L’obiettivo era quello di pulire i dati, integrando i 23 dataset acquisiti per creare i 6 dataset e le 13 tabelle ponte finali da importare su Neo4J, con un occhio di riguardo anche alla questione della consistenza delle diverse rappresentazioni delle stesse entità nei vari dataset.**

**Slide 16: Dataset API e scraping di Pokémon Database [40 secondi slide, 1:05 minuti totali]**

**Inizialmente siamo andati a pulire i dataset acquisiti tramite API, concentrandoci sulla gestione degli attributi di tipo lista o dizionario, da cui sono state estratte solo le informazioni utili ed eliminando quelle superflue. Contestualmente, sono stati corretti alcuni attributi, come ‘Is-Baby’, ‘Is\_legendary’ e ‘Is\_mythical’ che sono stati riassunti in ‘Rarity’ e ‘Generation’ che è stato tramutato in intero. Inoltre, sono state estratte le tabelle ponte Abilities-Pokémon e Moves-Pokémon, utili in seguito.**

**Slide 17: Dataset API e scraping di Pokémon Database [1 minuto slide, 2:05 minuti totali]**

**A questo punto è stata effettuata l’integrazione fra i dataset dell’API puliti e quelli di Pokémon Database. Per integrare le informazioni è stato effettuato un record linkage utilizzando la distanza di Levenshtein e basandosi su nome e ID per i Pokémon, che hanno un numero identificativo unico, e sul solo nome per le altre entità. In alcuni casi è stato necessario intervenire sui nomi per facilitare il matching, per esempio eliminando i – e rendendo maiuscola la prima lettera di ogni parola. I record esclusi dal matching sono stati controllati manualmente e poi aggiunti, se necessario. Dal dataset delle mosse, inoltre, è stato estratto un dataset riguardante le mosse G-Max.**

**Slide 18: Dataset Gigantamax Pokémon e G-Max Moves [1 minuto slide, 3:05 minuti totali]**

**La tabella appena estratta è stata poi integrata con G-Max Moves di Bulbapedia, andando poi a concatenare il risultato con il dataset principale delle mosse. Per quanto riguarda il dataset Gigantamax Pokémon di Bulbapedia, invece, è stata effettuata una fase di schema matching col dataset principale dei Pokémon, aggiungendo alcuni valori mancanti per specifici attributi e andandolo poi a concatenare. Va sottolineato che, a questo punto, è stato generato un nuovo attributo Name, costituito da Nome specie + Nome variante, rappresentazione che è stata mantenuta fino alla fine e che consente di identificare univocamente ciascuna entità Pokémon, a differenza dell’ID che è condiviso fra varianti diverse della stessa specie. È stata anche estratta la tabella ponte fra le varianti Gigantamax e le rispettive mosse G-Max.**

**Slide 19: Dataset Type e relative tabelle ponte [20 secondi slide, 3:25 minuti totali]**

**È stato pulito il dataset Types API, mantenendo solo le informazioni utili, mentre la Type Chart (18x18) dello scraping è stata convertita in una tabella ponte che riguarda l’efficacia di un tipo su un altro da 324 righe, una per coppia di tipi. Sono poi state generate anche le tabelle ponte fra Pokémon e mosse e i rispettivi tipi.**

**Slide 20: Tabelle ponte relative a Pokémon [45 secondi slide, 4:10 minuti totali]**

**Sono state poi create altre tabelle ponte: EVOLVES\_FROM e HAS\_VARIANT estraendo gli attributi necessari da Pokémon, MAY\_HAS e MAY\_LEARN sistemando le tabelle ponte ottenute in precedenza correggendo i nomi dei Pokémon per ricondurli al nuovo formato. La correzione è avvenuta sempre tramite record linkage, usando in questo caso l’algoritmo q-gram, più efficiente per gestire i casi di nomi identici ma costituiti da stringhe in ordine diverso.**

**Slide 21: Integrazione dati del competitivo [30 secondi slide, 4:40 minuti totali]**

**L’ultima parte ha riguardato l’integrazione dei dati del competitivo. Inizialmente le liste ottenute dal sito VictoryRoad sono state trattate in modo tale da associare a ciascun identificativo il nome di un Pokémon, ponendo attenzione ai riferimenti specifici di ciascuna lista, dal momento che alcune di esse fanno riferimento agli ID dei Pokédex regionali anziché a quello nazionale.**

**Slide 22: Integrazioni dati del competitivo [45 secondi slide, 5:25 minuti totali]**

**Le informazioni delle liste sono poi state integrate al dataset dei Pokémon. Lo stesso è stato fatto con i dati estratti da Pikalytics, integrati con i rispettivi dataset principali. Anche su questi ultimi è stata necessaria una fase di record linkage per ricondurre i nomi di Pokémon e mosse al formato precedentemente introdotto. A questo punto i dataset sono pronti al caricamento su Neo4J.**

**Slide 23: Data Quality [10 secondi slide, 5:35 minuti totali]**

**Non prima, però, di analizzare alcune dimensioni di qualità.**

**Slide 24: Currency [50 secondi slide, 6:25 minuti totali]**

**A livello di currency, una cosa importante da sottolineare è che nei casi in cui avessimo a disposizione le stesse informazioni provenienti da API e scraping è stata presa la decisione di mantenere quelle dello scraping, in quanto più aggiornate e, soprattutto per i dati dei giochi più recenti, anche più affidabili. Per il resto, va sottolineato che i nostri dati perdono currency mensilmente, con i dati del competitivo che, ogni primo del mese, vengono aggiornati con i dati delle competizioni del mese precedente.**

**Slide 25: Completezza [10 secondi slide, 6:35 minuti totali]**

**A livello di completezza, per tutti gli attributi la presenza di missing values è ampiamente giustificabile, ad eccezione dell’attributo Usage\_Attributes di Items che risultava particolarmente sparso sin dall’inizio.**

**Slide 26: Consistency [1 minuto slide, 7:35 minuti totali]**

**Molto importante è il discorso sulla consistenza. Avendo un gran numero di dataset di partenza, le rappresentazioni delle stesse entità fra i vari dataset era parecchio eterogenea. Per questo, in fase di integrazione, siamo intervenuti a livello di nomi per cercare di uniformare i nomi delle stesse entità. In modo particolare, siamo dovuti intervenire a livello dei nomi dei Pokémon per cui, come ho detto in precedenza, è stato scelto il formato Nome specie + Nome variante, così da poter identificare univocamente ciascuna entità di tipo Pokémon. Tale rappresentazione è stata estesa a tutti i dataset, così da non avere problemi anche a livello di relazioni.**

**A questo punto lascio la parola a Remo, per la parte di importazione in Neo4J.**

**PARTE DI REMO Tempi: 7:40**

**Slide 27: Data Model [5 secondi slide, 5 secondi totali]**

**Una volta opportunamente puliti e integrati i dati e averne valutato la loro qualità, si è alla fase di modellazione dei dati, ovvero alla fase dove viene definito il modello più adatto a rappresentare le nostre informazioni.**

**Slide 28: Data Model [x secondi slide, x minuti totali]**

**Come già accennato in precedenza, la scelta è ricaduta su un database a grafo, principalmente per:**

* **La natura relazionale dei dati e dalla volontà di mettere in risalto la relazione fra gli elementi più rilevanti del mondo Pokémon competitivo**
* **Dal rapporto tra le operazioni di scrittura e lettura in quanto le operazioni di scrittura, computazionalmente più onerose, risultano essere molte meno rispetto a quelle di lettura, essendo che le statistiche del gioco competitivo vengono aggiornate mensilmente.**

**Slide 29: Neo4j [x secondi slide, x minuti totali]**

**La piattaforma scelta per lo storage dei dati è Neo4j, questo principalmente perché: è una piattaforma open source, è un Database a grafo nativo e totalmente transazionale.**

**Slide 30: Schema1 [x secondi slide, x minuti totali]**

**A questo punto, analizzando i dati a nostra disposizione siamo passati all’individuazione di tutte le entità e relazionali di nostro interesse. Nello specifico sono state individuati 6 tipi di nodi: il nodo Pokémon che descrive le principali caratteristiche di un Pokémon e le sue statistiche legate al gioco competitivo,aggiornate a Marzo 2022,il nodo Moves che descrive le principali caratteristiche di una mossa, ovvero di una skill che un Pokémon può usare, il nodo Items che descrive le principali caratteristiche di uno strumento, elemento fondamentale del gioco competitivo, il nodo Natures che descrive le principali caratteristiche di una specifica natura, ossia una feature che indica la ”personalità” di un Pokémon e che ne influenza alcune statistiche, il nodo Type che descrive le caratteristiche di un tipo, ovvero una caratteristica peculiare di Pokémon e mosse e infine il nodo Abilities che descrive le caratteristiche di una specifica abilità, ovvero attributi speciali dei Pokémon che possono agire come potenziamento e/o introdurre effetti particolari.**

**Per quanto riguarda invece le relazioni sono state individuate 13 tipi di relazioni utili a descrivere sia la parte informativa ed enciclopedica del mondo Pokèmon, sia la parte del gioco competitivo.**

**Slide 31: Schema2 [x secondi slide, x minuti totali]**

**Una volta definite tutte le componenti del nostro grafo, si è deciso di inserire alcune restrizioni nello schema generale, ai fini di mantenere un certo grado di qualità e consistenza dei dati. Nello specifico è stato aggiunto, per ogni tipo di Nodo, un vincolo di unicità per la proprietà Name**: ciò impedisce quindi di inserire all’interno del database due nodi, dello stesso tipo, aventi lo stesso valore per la proprietà Name. Inoltre, in Neo4j l’aggiunta di un vincolo implica l’automatica aggiunta di un indice per la stessa proprietà. Ciò permette di aumentare le performance in fase di interrogazione, in quanto il sistema riuscirà ad individuare più rapidamente il punto di partenza del grafo. Al momento, non sono stati aggiunti altre indicizzazioni, questo perché la proprietà maggiormente interrogata, per ogni tipo di nodo, è la proprietà Name. Ciò non toglie che, in un secondo momento, si possano aggiungere altri indici su altre proprietà; è comunque doveroso sottolineare che l’aggiunta di indici comporta un aumento dello spazio di archiviazione in quanto il sistema produce delle duplicazioni di alcuni dati presenti nel database, oltre che un rallentare le operazioni di scrittura.

**Slide 32: Data Storage [x secondi slide, x minuti totali]**

**Passiamo ora a descrivere come i dati sono stati caricati in Neo4j. Per questo progetto è stato utilizzata la versione di Community Server di Neo4j e le operazioni di CREATE, UPDATE e DELETE sono state realizzate tramite Python, attraverso il driver ufficiale.**

**Slide 33: Storage 1 [x secondi slide, x minuti totali]**

**La creazione dei nodi e delle relazioni ha necessitato di una breve fase di pre-processing dei dati al fine di uniformare i datasets con le convezioni di Neo4j. Per la creazione dei nodi è stata implementata una funzione, integrata con la procedura di LOAD CSV, che controlla per ogni riga del dataset, se il valore dell’i-esima colonna è presente o meno se il dato non è presente non inserisco la proprietà e passo alla colonna successiva, mentre se il dato è presente inserisco la proprietà e il relativo valore. In questo modo quindi, una proprietà non viene inserita qualora il valore fosse mancante, dal momento che non è garantito che il dato di una specifica proprietà sia disponibile per tutti i nodi. La fase di import dei nodi ha prodotto un totale 3071 nodi.**

**Slide 34: Storage 2 [x secondi slide, x minuti totali]**

**L’import delle relazioni invece, effettuato sempre tramite Python e procedura LOAD CSV, è avvenuto sfruttando le tabelle ponte create nella fasi di acquisizione e integrazione dei dati. Ciò è possibile grazie al vincolo imposto sulla proprietà Name che garantisce un MATCH preciso tra il nodo di partenza e quello di arrivo. La fase di import delle relazioni ha prodotto un totale di 177720 relazioni.**

**Slide 35: Summary [5 secondi slide, x minuti totali]**

**Riportiamo quindi la distribuzione sia dei nodi che delle relazioni. Notiamo come per i nodi il tipo di nodo più frequente è caratterizzato ovviamente dai Pokemon, mentre per le relazioni, la relazione che presenta la più alta frequenza è MAY\_LEARN**

**Slide 36: Graph Exploration [5 secondi slide, x minuti totali]**

**Passiamo quindi all’ultima sezione, dedicata alla presentazione di alcune possibili interrogazioni e esplorazioni del grafo creato. Di seguito verranno riportati due esempi di query in linguaggio Cypher e un esempio di visualizzazione di un sottoinsieme del grafo creato tramite Gephi.**

**Slide 37: Query1 [x secondi slide, x minuti totali]**

**Ad esempio: Quali sono i Pokemon di tipo fuoco che vengono utilizzati con l’abilità “Solar Power”?**

**La query riporta i seguenti risultati: Charizard, Charmender, Charmeleon e Mega Houndoom.**

**Slide 38: Query2 [x secondi slide, x minuti totali]**

**Oppure:** Quali, tra le mosse più utilizzate dal Pokémon Charizard, sono super-efficaci sul Pokémon Arcanine?

La query, o meglio le query in quanto in questo caso è necessario prima sapere il tipo del Pokémon difendente, ovvero Arcanine, riporta il sequente risultato: Ancient Power

**Slide 39: Competitive Graph Data Visualization [x secondi slide, x minuti totali]**

**Infine per concludere, oltre all’usuale interrogazione è possibile effettuare delle analisi sui dati provenienti dal gioco competitivo ai fini di valutare il comportamento della communitiy. La visualizzazione è stata ottenuta tramite il tool Gephi, con l’obiettivo di mettere in risalto, ad esempio, quali sono i Pokémon maggiormente usati come compagni di squadra. Tali Pokémon sono evidenziati aumentando la dimensione della Label in base al grado entrate, ovvero dal numero di relazione che un nodo ha in entrata. E’ possibile quindi notare come i Pokemon maggiormente utilizzati come compagni di squadra sono: Zacian Crowned Sword, Incineroar, Rillaboom ecc..**

**Grazie mille per l’attenzione.**