Laboratory of Data Science

Report finale - Gruppo 17

Anno Accademico 2022/2023

Francesco Daquino 646706
f.daquino@studenti.unipi.it

1 Creazione del datawarehouse

In questa prima parte di progetto verranno illustrate le operazioni effettuate per costruire e popolare il database a partire dai file .csv forniti *answerdatacorrect* e *subject_metadata*.

1.1 Assignment 0: Definizione dello schema del database

Il primo passo del processo di BI è stata la costruzione dello schema del database (Figura 1), tramite software SQL Server Management Studio. In particolare, le singole tabelle dello schema sono state create tramite l'esecuzione della query sql relativa al file *create_tables.sql* allegato, con cui sono stati definiti attributi, tipo di dato e primary key per ciascuna tabella. Le chiavi esterne, invece, sono state definite solo successivamente alla creazione delle tabelle, e direttamente tramite software.

Per le dimensioni *organization*, *date* e *geography* sono stati definiti (in fase di split in sezione 1.2) nuovi numeri di identificazione, rispettivamente organizationid, dateid e geoid, che fungono da chiavi primarie delle tabelle. Invece, per le altre tabelle presenti nello schema è stato possibile selezionare un attributo già presente nel file .csv di partenza e impostarlo come chiave primaria.

Per la definizione dei tipi dei diversi attributi sono state effettuate le seguenti scelte:

- per gli attributi di tipo stringa con un numero variabile di caratteri è stato utilizzato il tipo "nvarchar";
- per gli attributi di tipo intero è stato definito il tipo "int";
- per gli attributi di tipo intero, che assumono al massimo un ristretto numero di valori non negativi è stato definito il tipo "tinyint";
- per l'attributo **date** della tabella *date* si è scelto di cambiare il formato da stringa a "date", utile per le fasi successive del progetto.

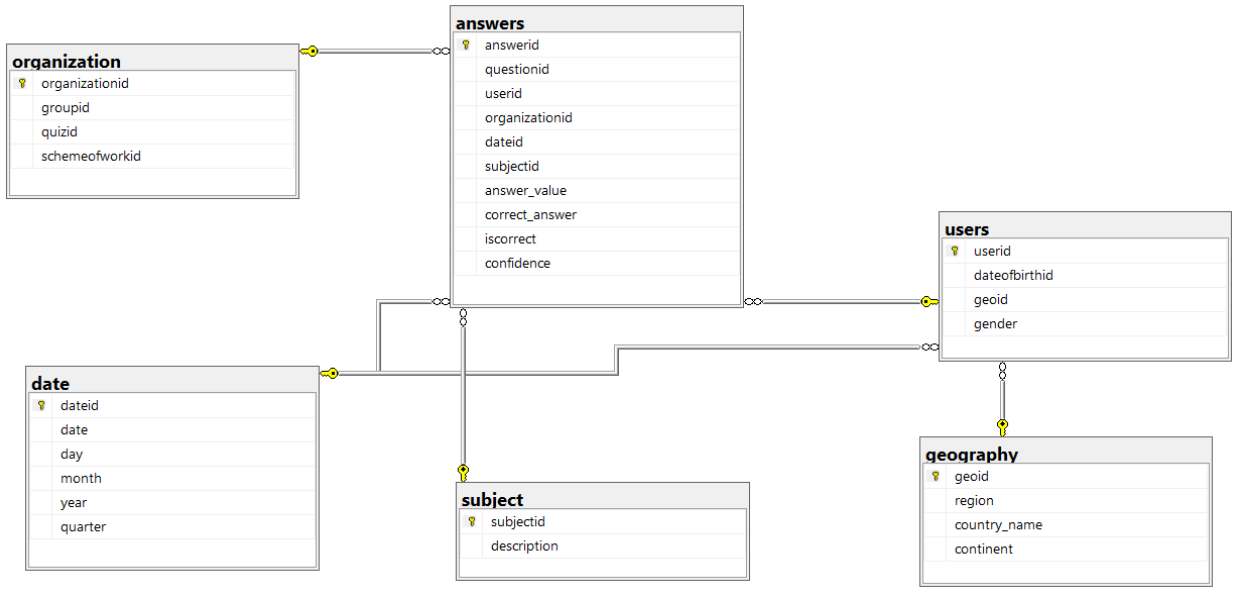

Figura 1: Database schema

1.2 Assignment 1: Generazione tabelle

Delle sei tabelle da dover generare tramite le informazioni contenute in *answerdatacorrect*:

- *organization*, *date* e *geography* sono state create nel file *split_answerdatacorrect.py*;
- la tabella *users* è stata creata nel file *create_users*.py;
- la tabella *answers* è stata creata nel file *create_answers.py*;
- la tabella *subject* è stata creata nel file *create_subject.py* tramite esplorazione del file *subject_metadata* fornito.

I dati prodotti sono stati inseriti nella cartella *project_data*. La scelta di tale suddivisione e ordine di esecuzione di file per la creazione delle tabelle è stata presa sulla base della possibilità di creazione di dizionari utilizzati per garantire il corretto accoppiamento e utilizzo delle chiavi esterne tra tabelle.

1.2.1 Split Answerdatacorrect

La divisione dei dati nelle tabelle è stata effettuata scorrendo con un ciclo for i record del file originale *answerdatacorrect*, seguendo il seguente schema di operazioni:

- scrittura dell' header delle tabelle tramite funzione *create_header*;

- scritture delle variabili “country_name” e “continent” della tabella geography tramite la definizione di due funzioni *countrycode_to_countryname* e *country_to_continent*, utilizzate ricorrendo alla libreria “pycountry-convert” la quale, utilizzando i dati sui paesi derivati da Wikipedia, fornisce funzioni di conversione tra nomi di paesi ISO, codici paese e nomi di continenti.
- Uso della funzione *prepare_date*, per scrivere i dati relativi alla data, inserendo come valore del trimestre uno tra Q1, Q2, Q3, Q4 in base al valore del mese, secondo le seguenti associazioni: [01, 03] → Q1; [04, 06] → Q2; [07, 09] → Q3; [10, 12] → Q4. Dal momento che la tabella *date* deve contenere sia le date di nascita degli utenti sia le date delle risposte, i due tipi di date sono state uniformate scartando le informazioni relative alle ore e ai minuti.
- Scrittura e preparazione dei file finali tramite metodo writerow(), effettuando di volta in volta un controllo sulla correttezza e univocità delle chiavi primarie tramite set, e aggiungendo il nuovo id progressivo laddove richiesto.

1.2.2 Creazione tabelle users, answers e subject

Per la costruzione della tabella *users* sono stati aperti in lettura i file .csv *geography* e *date* precedentemente generati, per mezzo dei quali sono stati creati due dizionari utilizzati per recuperare il valore corretto di geoid e dateofbirthid per ogni iterazione del file originale answerdatacorrect.

Allo stesso modo, per la costruzione della tabella *answers* sono stati aperti in lettura i file .csv organization e date, per mezzo dei quali sono stati creati due dizionari utilizzati per recuperare il valore corretto di organizationid e dateanswerd per ogni iterazione del file originale answerdatacorrect. Inoltre, confrontando riga per riga i valori delle variabili answer_value e correct_answer, è stato ottenuto l’attributo **iscorrect**, usato come misura di valutazione delle risposte.

Per la costruzione della tabella *subject* è stato aperto in lettura il file *subject_metadata* contenente le informazioni necessarie per la definizione della variabile description, che mostra la lista di materie relative agli id, opportunamente ordinate per livello tramite definizione della funzione *ordina_subjectid*.

Le scritture dei file finali sono state effettuate con gli stessi metodi descritti all’ultimo punto della sezione 1.2.1.

1.3 Assignment 2: Inserimento dei dati nello schema

Per popolare il database è stato scritto il programma “upload_data.py”, con la seguente struttura:

- apertura della connessione al database Group_17_DB tramite la libreria pyodbc e creazione del cursore;
- lettura dei file .csv, utilizzando il metodo csv.reader;
- creazione di un dizionario contenente le associazioni <nome_tabella>→ <reader_tabella>.
- Per ogni elemento del dizionario è stata generata una query “INSERT INTO”, dinamicamente in base al numero di elementi presenti nell’header del file corrispondente, eseguita tramite cursor.execute(). Contemporaneamente viene effettuato un controllo del tipo di dato, effettuando un cast ad intero dove necessario.
- Chiusura dei file, cursore e connessione.

2 Analisi dati tramite SSIS

L’obiettivo in questa seconda fase del progetto è quello di rispondere ad alcune business questions relative al database creato nella prima parte, utilizzando Sequel Server Integration Services (SSIS) e sviluppando delle soluzioni che calcolino i risultati dal lato client. In particolare, è stata sviluppata un’ unica soluzione contenente tre packages relativi alle tre business questions assegnate.

2.1 Assignment 0

- Business question: *For every subject, the number of correct answers of male and female students.*

Tramite il nodo "Origine OLE DB" sono state lette solo le colonne necessarie della tabella *Answers* (subjectid, userid e iscorrect). Con il nodo "Ricerca" è stata operata la giunzione (tramite userid) con la colonna "gender" della tabella *Users*. Tramite nodo “Suddivisione condizionale” è stato possibile filtrare e indirizzare in output le sole righe di dati contenenti risposte corrette, specificando la condizione =1 per l’attributo binario iscorrect. E’ stata poi applicata, tramite nodo ”Aggregazione” una funzione di aggregazione per contare il numero

di risposte corrette per genere, raggruppando i dati per gli attributi "subjectid" e "gender". Il risultato finale, scritto su un file di testo tramite nodo ”Destinazione file flat”, mostra per ogni subject id il numero di risposte corrette di studenti di genere maschile e femminile.

2.2 Assignment 1

- Business question*: A subject is said to be easy if it has more than 90% correct answers, while it is said to be hard if it has less than 20% correct answers. List every easy and hard subject, considering only subjects with more than 10 total answers.*

Analogamente a quanto visto per il primo assignment , è stato usato il nodo “Origine OLE DB” per il recupero delle colonne subjectid e iscorrect dalla tabella *Answers*. Successivamente si è scelto di utilizzare un nodo “Multicast” per indirizzare tutte le righe in input verso due output diversi, così da poter calcolare separatamente il numero di risposte corrette e il numero di risposte totali per ogni subject id.

Il primo risultato è stato ottenuto tramite un nodo di “Aggregazione” , contando il numero di righe raggruppate per subjectid, preventivamente filtrate per sole risposte corrette tramite nodo di “suddivisione condizionale”.

il secondo risultato è stato raggiunto eseguendo anche qui un raggruppamento subjectid e contando le righe totali tramite nodo “aggregazione”. E’ stato utilizzato un nodo di “suddivisione condizionale” per selezionare solo le subjectid che avessero un numero di risposte totali date maggiore di dieci.

Infine, i due flussi di dati, ordinati per subjectid tramite nodi di “ordinamento”, sono stati uniti tramite nodo di trasformazione “Merge join”, che fornisce un output unendo in join due set di dati ordinati.

Successivamente si è scelto di utilizzare un nodo di “Conversione dati” per avere i dati relativi al numero di risposte totali e corrette in formato numerico e, in base a quanto richiesto dal testo dell’assignment, di eseguire il calcolo della percentuale di risposte corrette in una nuova colonna, tramite l’inserimento della seguente espressione in un nodo di “Colonna derivata”: ***(n_risposte_corrette * 100)/ n_risposte_totali.***

Infine, si è scelto di utilizzare un ulteriore nodo di “Colonna derivata” per creare una nuova colonna che, tramite operatore condizionale, etichetti come ***easy*** le subject id con percentuale di risposte corrette maggiore di 90, ***hard*** le subjectid con percentuale di risposte corrette minore di 20 e ***normal*** le righe rimanenti

(expression: **perc < 20 ? "hard" : (perc > 90 ? "easy" : "normal"**)).
Il risultato finale, prima della scrittura su file di testo tramite nodo “Destinazione file flat”, è stato filtrato per selezionare le sole subjectid etichettate come easy e hard.

2.3 Assignment 2

- Business question: *For each country, the student or students that answered the most questions correctly for that country*.

Per definire una soluzione relativa alla business question è stato necessario accedere alle tre tabelle *Answers*, *Users* e *Geography*. Analogamente ai casi visti fin qui, dalla Fact Table *Answers* si ottengono gli attributi subjectid, userid e iscorrect. Geoid e country_name si recuperano invece mediante due consecutivi nodi di ricerca, rispettivamente dalle tabelle *Users* e *Geography*. A questo punto sono necessari un nodo di “Suddivisione condizionale” per selezionare solo le righe relative alle risposte corrette e uno di “Aggregazione” per calcolare il numero di risposte corrette totali raggruppate per userid e country_name.
Si è scelto di duplicare il risultato con un nodo di "Multicast". In questo modo è possibile, da un lato, applicare una funzione di aggregazione, con cui si ottiene il massimo valore del numero di risposte corrette per ogni country_name (output ordinato per tale valore), e dall’altro un semplice ordinamento di valori per numero di risposte corrette. Le due tabelle vengono poi ricongiunte con un nodo "Merge join" su uguaglianza dei valori appena descritti, quindi max_correct_answer e n_correct_answer. La tabella risultante conterrà quindi i distinti valori di country_name e per ognuno di essi lo studente che ha risposto al maggior numero di domande per quel paese. Il risultato viene scritto su file di testo tramite nodo “destinazione file flat”.

3 Analisi dati tramite SSAS

3.1 Assignment 0: costruzione datacube

Nell'ultima fase del progetto è stato costruito il datacube a partire dai dati contenuti nelle tabelle del database creato precedentemente, tramite SSAS (SQL Server Analysis Services).
In particolare, dopo aver selezionato una nuova origine dati per la soluzione denominata "SSASproject-group17" e aver impostato la connessione al DB, sono state definite le dimensioni *time subject users e organization.*
Come richiesto dall'esercizio, sono state definite le gerarchie appropriate per *time* e *geography*, e le misure necessarie per rispondere alle query.
La soluzione presenta dunque due gerarchie non flat costruite al fine di gestire sia le dimensioni temporali che spaziali dei dati, ovvero *DayMonthQuarterYear* e *RegionCountryContinent*, come visibile in Figura 2.

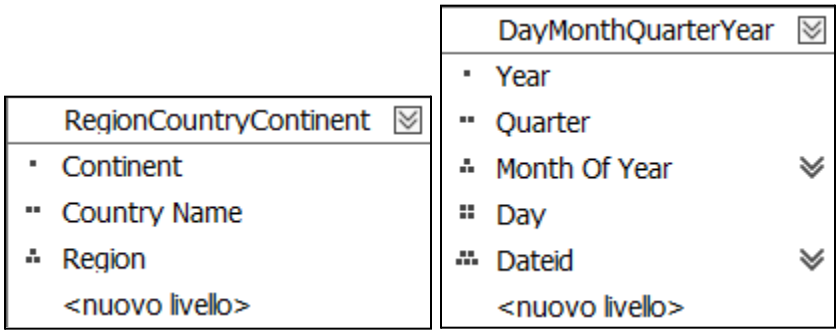

Figura 2: Gerarchie per Time e Geography

Per garantire una migliore visualizzazione e un corretto ordinamento dei dati, è stato creato tramite calcolo denominato la colonna "MonthOfYear" (inserita all'interno della gerarchia) contenente il nome dei mesi invece del singolo numero presente nella tabella originale.
Come ultimo step è stata definita una nuova colonna "Incorrect" tramite calcolo denominato della fact table *answers* ed infine, è stato creato il datacube "Group_17_DB", definendo come misure *Correct*, *Incorrect* e *Conteggio_di_Answers*, necessarie per rispondere alle successive query. In particolare, le prime due misure, essendo binarie con valori 0 e 1, sono stata definite selezionando AggregateFunction→Sum e FormastString→Standard e usate per misurare rispettivamente il numero di risposte corrette e sbagliate degli studenti;

Conteggio_di_Answers è stata invece lasciata con AggregateFunction→Count e FormastString→Standard.
Infine, è stata effettuata l'elaborazione del cubo lato server.

3.2 Assignment 1, 2 e 3: query MDX

Le soluzioni alle query mdx richieste sono state implementate nel file *Assignment1-3.mdx.*

- *Show the student that made the most mistakes for each country.*

Per mostrare il risultato richiesto, è stata utilizzata la misura *Incorrect* sulle colonne, e la combinazione delle funzioni *generate* e *topcount* sulle righe, grazie alle quali è stato possibile selezionare lo studente col più alto numero di errori per ciascun paese.

- *For each subject, show the student with the highest total correct answers.*

Con la stessa struttura della query 1, è stata utilizzata la misura *Correct* sulle colonne e la combinazione delle funzioni *generate* e *topcount* sulle righe per mostrare lo studente col più alto numero di risposte corrette per ogni materia, ed eliminando i valori nulli con la funzione *nonempty*.

- *For each continent, show the student with the highest ratio between his total correct answers and the average correct answers of that continent.*

E' stata scritta una query MDX che calcola come membro derivato "avg_correctanswers" tramite la funzione *AVG* a cui vengono passati due argomenti: il primo è l'espressione che definisce gli studenti per ciascun continente, e il secondo è la misura "Correct" di cui si vuole calcolare la media. Dopodiché viene calcolata la misura "ratio" come rapporto tra le risposte corrette e la loro media calcolata precedentemente, e utilizzata sulle colonne per selezionare lo studente che ha il valore di ratio più alto per ogni continente (anche qui con combinazione di funzioni *generate* e *topcount*). E' stata anche inserita una condizione *iff* per evitare di avere divisioni per valori nulli.

3.3 Assignment 4 e 5: creazione dashboard

- *Create a dashboard that shows the geographical distribution of correct answers and incorrect answers.*

Per mostrare la distribuzione geografica delle risposte sono state usate due mappe, una per *Correct* e una per *Incorrect*, ognuna delle quali legata ad uno specifico grafico ad anello, che mostra la distribuzione delle misure a diversa granularità geografica (tramite drilldown). La variazione di dimensione delle bolle sulle mappe dipende dal valore totale dell'attributo esplorato per quella regione, paese o continente.

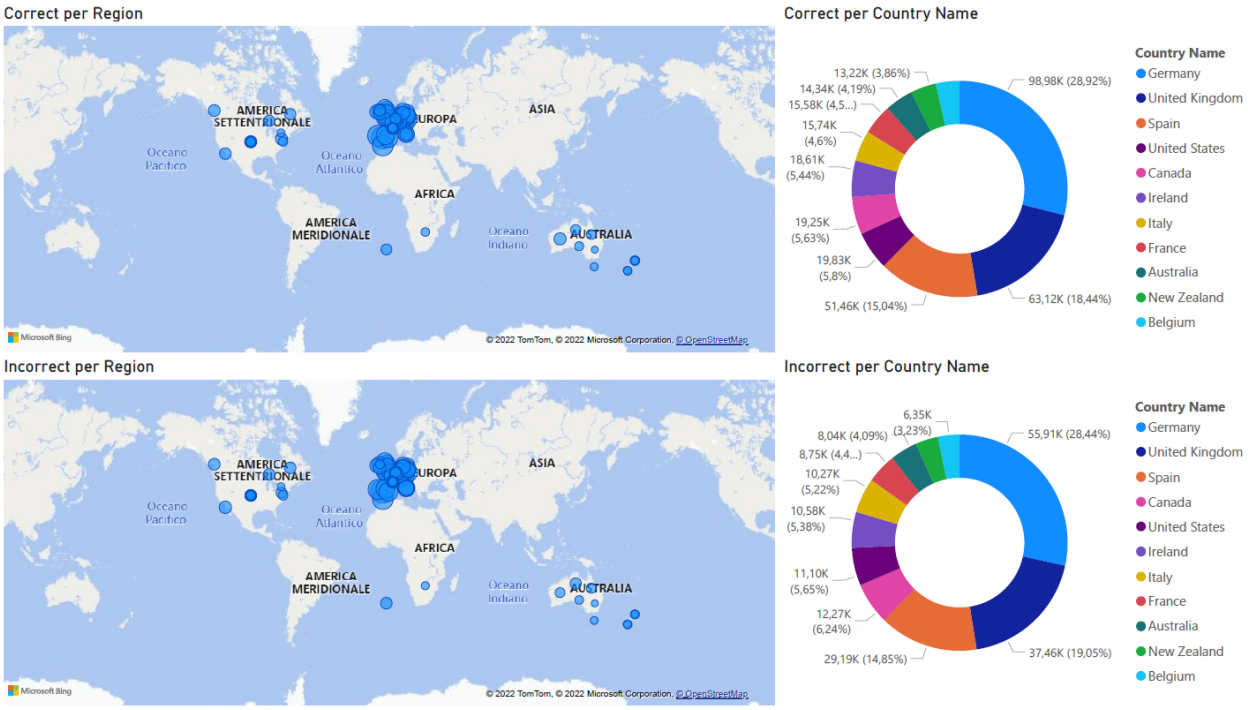

Figura 3: distribuzione geografica delle risposte corrette ed errate

- *Create a plot/dashboard of your choosing, that you deem interesting w.r.t. the data available in your cube.*

Per l'ultimo assignment si è deciso di utilizzare un istogramma a colonne in pila che, per gender diverso, mostra il numero di risposte totali date da studenti per diversa regione, paese o continente (anche qui visualizzabile con diversa granularità geografica). Si è scelto inoltre di applicare una serie di filtri per mostrare come cambia il grafico qualora l'utente abbia interesse nel selezionare una specifica subject o uno specifico periodo temporale, visualizzando sempre il numero totale di risposte tramite oggetto visivo "scheda".

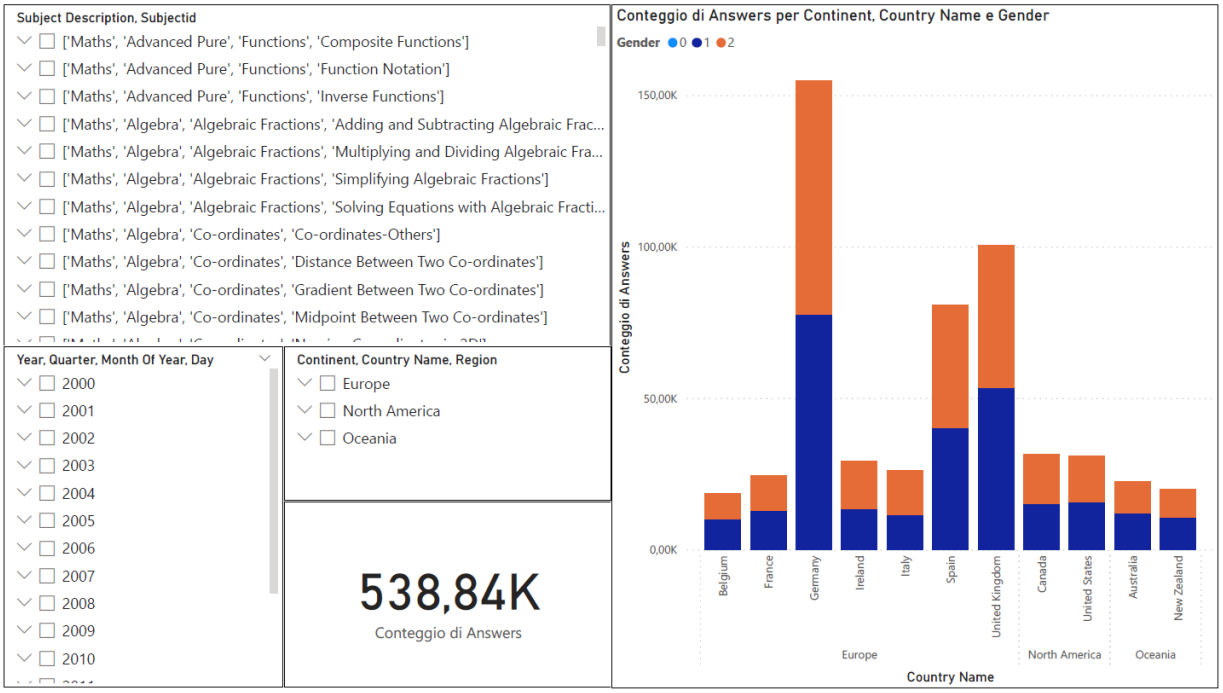

Figura 4: grafico a linee risposte corrette - istogramma a colonne in pila risposte totali