

Integrazione dati sul web

Prof. G. Costagliola



IDWork

Better Job. Better life.

Flavio Esposito
Antonio Vivone

A.A. 2018/2019

1.

Introduzione al problema

Introduzione al problema e sistema proposto

Al giorno d'oggi esistono molte piattaforme che hanno dato un notevole slancio al **mondo del lavoro** e aperto ulteriormente il **mercato** dietro ad esso



1. Introduzione al problema

3

Risulta infatti molto più semplice sia per le aziende trovare il **candidato giusto**, sia per i professionisti imbattersi in opportunità di lavoro adatte ai propri **interessi** e alle proprie **competenze**



L'aumento delle opportunità relative al mondo del lavoro ha generato la possibilità di **rispondere a domande a cui prima difficilmente si poteva trovare una risposta**

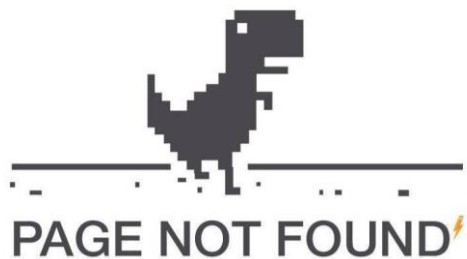


Del tipo...

- » È l'**azienda** che fa per me?
- » Il **compenso** è giusto?
- » Sono disposto a **trasferirmi**?
- » Quali sono le mie **aspettative future**?
- » Come posso prepararmi al meglio per l'**interview**?



Ma non ci sono al momento servizi che integrano tutte queste informazioni in una **singola piattaforma**



IDWork vuole essere la piattaforma di riferimento per chi **cerca lavoro** sul territorio statunitense e vuole farlo nel modo più sereno possibile, tenendo conto di tutti gli **aspetti di rilievo della scelta**





L'ambito di operatività del sistema è quello **USA**

Questa scelta è stata presa per due principali motivi:

- » **Qualità delle fonti** (completezza e autorevolezza)
- » **Disponibilità fonti** (validità e disponibilità)

Inoltre, si è scelto di considerare i lavori relativi all'**ambito IT** in considerazione di voler:

- » Rendere la piattaforma **meno generica**
- » Servirsi delle **risorse qualitativamente migliori**

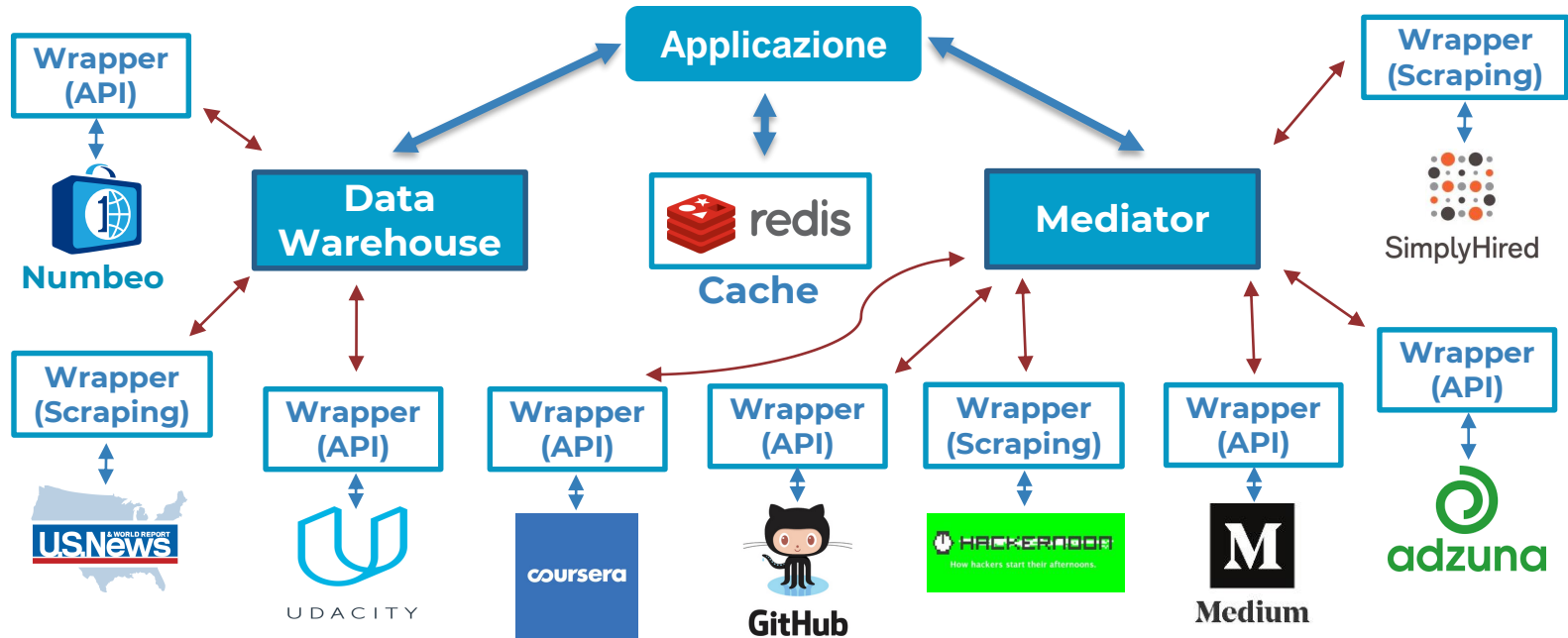


2.

Architettura e fonti

2. Architettura del sistema

10



- **Job:**

(**Job_Name, Category, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL**)



- **City:**
(City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Median_Age, Home_Price, Rent_Price, Commute_Time)



- **Article:**

(Title, Category, Author, Score, Img_URL, Article_URL)



Medium



- **Course:**
**(Course_Title, Topic, Description, Difficulty, Skills,
Preview, Course_URL)**



- **Repository:**

**(Repo_Name, Category, Author, Description,
LE_Date, Stars, Repo_URL)**



2. Fonti

16

<i>Nome</i>	<i>Metodo</i>	<i>Grado di volatilità</i>
Adzuna (AD)	API	Aggiornamento continuo
SimplyHired (SH)	Web scraping	Aggiornamento continuo
Numbeo (NU)	API	Validità mensile
U.S.News (USN)	Web scraping	Validità mensile

2. Fonti

17

<i>Nome</i>	<i>Metodo</i>	<i>Grado di volatilità</i>
Medium (ME)	API	Aggiornamento continuo
Hacker Noon (HN)	Web scraping	Aggiornamento continuo
Coursera (CO)	API	Validità giornaliera
Udacity (UD)	API	Validità giornaliera
Github (GH)	API	Aggiornamento continuo

Adzuna è un aggregatore di offerte di lavoro operante in tutto il mondo. Esso combina **migliaia di fonti** diverse su un unico portale offrendo così una vasta scelta per quando si ricerca un lavoro.



2. Fonti – Adzuna (AD)

19

Da Adzuna verranno prelevate le seguenti informazioni:

- Titolo del lavoro
- Azienda
- Descrizione del lavoro
- Luogo del lavoro
(città, nazione, latitudine e longitudine)
- Compenso (se presente)
- URL inserzione



AD (Job_Name, Field, Category, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Job_URL)

SimplyHired è una piattaforma che permette di consultare le statistiche sui valori dei **salari medi** per ogni professione sul territorio statunitense, in base al proprio grado di **esperienza**



2. Fonti – SimplyHired (SH)

21

Da SimplyHired verranno prelevate le seguenti informazioni:

- Salario minimo
- Salario medio
- Salario massimo



SH (Category, Min_Salary, Avg_Salary, Max_Salary, City, Country)

Numbeo è il più grande database al mondo di informazioni su città e nazioni. Fornisce informazioni come il **costo della vita**, la situazione sanitaria e gli indicatori del mercato immobiliare.



2. Fonti – Numbeo (NU)

23

Da Numbeo verranno prelevate le seguenti informazioni:

- Indice quality of life
- Indice affitto
- Indice criminalità
- Indice sanità



NU (City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index)

2. Fonti – U.S.News (USN)

24

U.S. News & World Report è una società americana che pubblica notizie, opinioni, **ranking** e **analisi** su tutte le principali località negli USA.

Offre un'overview completa delle città evidenziandone i pro e i contro
Inoltre non sono poche le **statistiche** di interesse per chi sta valutando un trasferimento



2. Fonti – U.S.News (USN)

25

Da U.S.News verranno prelevate le seguenti informazioni:

- Descrizione città
- Popolazione
- Salario medio annuo
- Età media città
- Prezzo medio abitazioni
- Prezzo medio affitti
- Tempo medio nei trasporti pubblici



USN (City, Country, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time, City_Description)

Medium è una piattaforma di pubblicazioni online di recente creazione. Al suo interno è possibile trovare risorse e tutorial sugli ultimi trend e tecnologie emergenti.



Medium

Da Medium verranno prelevate le seguenti informazioni:

- Titolo dell'articolo
- Autore
- Valutazione
- URL all'articolo
- URL immagine articolo

ME (Title, Author, Article_URL, Img_URL, Score, Category)



Medium

2. Fonti – Hacker Noon (HN)

28

Hacker Noon è un **tech blog** indipendente che permette a **professionisti** del settore IT e correlati di scrivere opinioni e articoli senza alcuna restrizione. È uno dei blog relativi al mondo dell'informatica più seguito globalmente e mette a disposizione uno staff di editor per controllare la qualità del materiale sul sito.



2. Fonti – Hackernoon (HN)

29

Da Hacker Noon verranno prelevate le seguenti informazioni:

- Titolo dell'articolo
- Autore
- URL all'articolo
- URL immagine articolo



HN (Title, Author, Article_URL, Img_URL, Category)

Coursera è una piattaforma statunitense che opera nel campo delle **tecnologie didattiche** fondata dai professori della Stanford **Andrew Ng** e **Daphne Koller**, che offre corsi online (**MOOC**), specializzazioni e corsi di studio
Coursera collabora con varie università e i servizi disponibili sono **certificati**



Da **Coursera** verranno prelevate le seguenti informazioni:

- Titolo del corso
- Descrizione breve del corso
- Difficoltà corso
- Skills risultato del corso
- Preview grafica del corso
- URL



CO (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

Udacity è un'organizzazione educativa che offre corsi online aperti (**MOOC**)

Offre lezioni video con sottotitoli e quiz integrati, promuovendo la metodologia del '**learning by doing**'



U D A C I T Y

Da **Udacity** verranno prelevate le seguenti informazioni:

- Titolo del corso
- Descrizione breve del corso
- Difficoltà corso
- Skills risultato del corso
- Preview grafica del corso
- URL



UDACITY

UD (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

GitHub è la piattaforma leader per l'hosting di **progetti software** sottoforma di codice sorgente. Dispone di una enorme quantità di progetti da poter **consultare** e a cui è possibile **contribuire**. Inoltre sono numerosi i repositories contenente **materiale didattico e di studio** raccolto dagli utenti



2. Fonti – GitHub (GH)

35

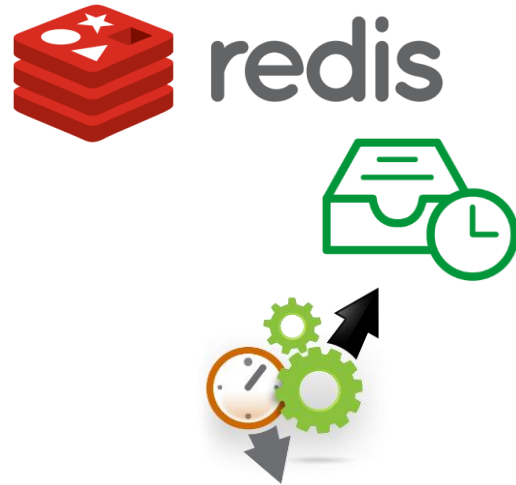
Da Github verranno prelevate le seguenti informazioni:

- Nome della repo
- Autore repo
- Descrizione repo
- Data ultima modifica
- Stelle repo
- Link alla repo



GH (Repo_Name, Category, Author, Description, LE_date, Stars, Repo_URL)

Per aumentare la reattività del sistema ed evitare di effettuare richieste alle fonti ad ogni ricerca da parte dell'utente, si è deciso di implementare un sistema di cache basato su Redis.



2. Cache

37

Global scheme table	Validità (ore)	Note
Job	1 ora	/
City	24 ore	/
Article	1 ora	/
Course	24 ore	/
Repository	1 ora	/

3.

Wrappers e descrizione funzionale

» **Adzuna** (API)

» **GET**

`http://api.adzuna.com/v1/api/jobs/us/search/1?app_id={APP_ID}&app_key={APP_KEY}&what={keyword}&where={location}&category=it-jobs`

- » **Simply Hired** (**Scraping**)
- » `Job_name` : `/html/body/div[4]/div/div[2]/div/div/h1/text()`
- » `min_salary`, `avg_salary`, `max_salary` :
`/html/body/div[4]/div/div[3]/div[1]/div[2]/div@data-points`
- » `City` : `/html/body/div[4]/div/div[2]/div/div/h1`

» **Numbeo** (API)

» **GET**

`https://www.numbeo.com/api/indices?api_key={APP_KEY}&query={city,country}`

» **U.S.News** (Scraping)

» Population :

```
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[1]/ul/li[1]/div[2]/h4[1]
```

» average_annual_salary :

```
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[1]/ul/li[2]/div[2]/h4[1]
```

- » Temps :
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[1]/ul/li[3]/div[2]/h4[1]
- » Median_age :
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[2]/ul/li[1]/div[2]/h4[1]
- » Median_home_price :
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[2]/ul/li[2]/div[2]/h4[1]

- » **Average_annual_rainfalls :**
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[2]/ul/li[3]/div[2]/h4[1]
- » **Unemployment_rate :**
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[3]/ul/li[1]/div[2]/h4[1]
- » **Median_monthly_rate :**
/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[3]/ul/li[2]/div[2]/h4[1]

- » `Average_commute_time :`
`/html/body/div[1]/div/div/div[3]/div/main/div[5]/div/div/div[3]/ul/li[3]/div[2]/h4[1]`

» **Medium (API)**

» **GET**

https://medium.com/_/api/tags/{keyword}/stream

- » **Hackernoon** (**Scraping**)
`https://hackernoon.com/tagged/{keyword}`
- » **Title** : `/html/body/div[1]/div[3]/div[3]/div[1]/div[1]/div/a`
- » **URL** :
`/html/body/div[1]/div[3]/div[3]/div[1]/div[1]/div/a@href`
- » **Author** :
`/html/body/div[1]/div[3]/div[3]/div[1]/div[3]/div/div[2]/a`

» **Coursera (API)**

» **POST**

`https://lua9b20g37-dsn.algolia.net/1/indexes/*/queries?x-algolia-application-id={APP_ID}&x-algolia-api-key={APP_KEY}&Host=lua9b20g37-dsn.algolia.net&....`

- » Body Request
- » **requests:** [{indexName: "prod_all_products",
- » **params:**
`query=\${keyword}&hitsPerPage=9223372036854775807&page=0&highlightPreTag=%3Cais-highlight-0000000000%3E&highlightPostTag=%3C%2Fais-highlight-0000000000%3E&optionalFilters=query%3A\${keyword}&facets=%5B%5D&tagFilters=` },

- » {indexName: "test_suggestions",
- » **params:**
`query=\${keyword}&hitsPerPage=9223372036854775807&page=0&highlightPreTag=%3Cais-highlight-0000000000%3E&highlightPostTag=%3C%2Fais-highlight-0000000000%3E&optionalFilters=query%3A\${keyword}&facets=%5B%5D&tagFilters=`]]

» **Udacity** (API)

» **GET**

<https://catalog-api.udacity.com/v1/catalog>

» **Github** (API)

» **GET**

`https://api.github.com/search/repositories?q={topic}`

3. Mapping GAV

53

Job (Job_Name, Category, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL) :-

AD(Job_Name, Field, **Category**, Company, Small_description, Small_description, **City**, **Country**, Lat, Lon, Salary, Job_URL),

SH(**Category**, Min_Salary, Avg_Salary, Max_Salary, **City**, **Country**) , Field = "it-jobs"

City (City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time) :-

NU(**City**, **Country**, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index),

USN (**City**, **Country**, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time, City_Description)

3. Mapping GAV

54

Article (Title, Category, Author, Score, Img_URL, Article_URL) :-

ME (Title, Category, Author, Score, Article_URL)

Article (Title, Category, Author, _ , Img_URL, Article_URL) :-

HN (Title, Category, Author, Img_URL, Article_URL)

Course (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL) :-

CO (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

Course (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL) :-

UD (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

Repository (Repo_Name, Category, Author, Description, LE_Date, Stars, Repo_URL) :-

GH (Repo_Name, Category, Author, Description, LE_Date, Stars, Repo_URL)

3. Mapping LAV

55

AD (Job_Name, Field, Category, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Job_URL) :-

Job (Job_Name, Category, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, , , , Job_URL)

SH (Category, Min_Salary, Avg_Salary, Max_Salary, City, Country) :-

Job (, Category , , , , , City, Country , , , , , Min_Salary, Avg_Salary, Max_Salary,)

NU (City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index) :-

City (City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, , , , , , , , , ,)

USN (City, Country, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time, City_Description) :-

City (City, Country, , , , , , , , City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time)

3. Mapping LAV

56

ME (Title, Category, Author, Score, Img_URL, Article_URL) :-
Article (Title, Category, Author, Score, Img_URL, Article_URL)

HN (Title, Category, Author, Img_URL, Article_URL) :-
Article (Title, Category, Author, _ , Img_URL, Article_URL)

CO (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL) :-
Course (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

UD (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL) :-
Course (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

GH (Repo_Name, Category, Author, Description, LE_Date, Stars, Repo_URL) :-
Repository (Repo_Name, Category, Author, Description, LE_Date, Stars, Repo_URL)

- **GetITJobsInPlace** (Job_Name, **Category**, Field, Company, Small_description, Full_description, **City**, **Country**, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL):



Job (Job_Name, **Category**, Field, Company, Small_description, Full_description, **City**, **Country**, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL)



SELECT *

FROM Job

WHERE Job.Category = "Javascript" AND Job.City = "Austin" AND Job.Country = "TX"

- **GetJobCityInfo** (**Job_Name**, **City**, **Country**, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time):



Job (**Job_Name**, Category, Field, Company, Small_description, Full_description, **City**, **Country**, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

City (**City**, **Country**, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time)



```
SELECT QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Median_Age, Home_Price, Rent_Price, Commute_Time
```

```
FROM Job, City
```

```
WHERE Job.Job_Name = "Full stack Javascript developer" AND Job.City = City.City  
AND Job.Country = City.Country
```

- **GetJobArticlesInfo** (**Category**, Title, Author, Score, Img_URL, Article_URL):



Job (Job_Name, **Category**, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

Article (Title, **Category**, Author, Score, Img_URL, Article_URL)



```
SELECT Title, Category, Author, Score, Img_URL, Article_URL
```

```
FROM Job, Article
```

```
WHERE Job.Category = Article.Category
```

- **GetJobCoursesInfo** (**Category**, Course_Title, Topic, Description, Difficulty, Preview, Course_URL):



Job (Job_Name, **Category**, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

Course (Course_Title, Topic, Description, Difficulty, **Category**, Preview, Course_URL)



```
SELECT Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL
```

```
FROM Job, Course
```

```
WHERE Job.Category = Course.Category
```

- **GetJobReposInfo** (**Category**, Repo_Name, Author, Description, LE_Date, Stars, Repo_URL):



Job (Job_Name, **Category**, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

Repository (Repo_Name, **Category**, Author, Description, LE_Date, Stars, Repo_URL)



```
SELECT Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL
```

```
FROM Job, Repository
```

```
WHERE Job.Category = Repository.Category
```

- **GetJobCityInfo** (**Job_Name**, **City**, **Country**, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time) :-

Job (**Job_Name**, Category, Field, Company, Small_description, Full_description, **City**, **Country**, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

City (**City**, **Country**, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time)

Query:

trova le informazioni delle città relative ai lavori

3. Unfolding Algorithm

63

1. Job (**Job_Name**, Category, Field, Company, Small_description, Full_description, **City**, **Country**, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),
2. City (**City**, **Country**, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Median_Age, Home_Price, Rent_Price, Commute_Time)

1. Job (Job_Name, Category, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL) :-
AD(**Job_Name**, Field, Category, Company, Small_description, Small_description, **City**, **Country**, Lat, Lon, Salary, Job_URL),
SH(Category, Min_Salary, Avg_Salary, Max_Salary, **City**, **Country**), Field = "it-jobs"

2. City (City, Country, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Median_Age, Home_Price, Rent_Price, Commute_Time) :-
NU(**City**, **Country**, Lat, Lon, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index),
USN (**City**, **Country**, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time, City_Description)

3. Unfolding Algorithm

64

GetJobCityInfo' (**Job_Name**, **City**, **Country**, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index, City_Description, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time) :-

Rule unfolding:

AD (**Job_Name**, V1, Field, V2, V3, V4, **City**, **Country**, V5, V6, V7, V8),

SH (V1, V9, V10, V11, **City**, **Country**), Field = "it-jobs",

NU (**City**, **Country**, V5, V6, QoL_Index, Rent_Index, Crime_Index, Healthcare_Index),

USN (**City**, **Country**, Population, Avg_Salary, Temperature, Median_Age, Home_Price, Rainfall_Level, Unemployment_Rate, Rent_Price, Commute_Time, City_Description)

Nulla da semplificare



- **GetJobCoursesInfo** (**Job_Name**, **Category**, Course_Title, Topic, Description, Difficulty, Preview, Course_URL):-

Job (**Job_Name**, **Category**, Field, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Min_Salary, Avg_Salary, Max_Salary, Job_URL),

Course (Course_Title, Topic, Description, Difficulty, **Category**, Preview, Course_URL)

Bucket Job

1. **AD** (Job_Name, Field', Category, Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Job_URL')
2. **SH** (Job_Name, Min_Salary', Avg_Salary', Max_Salary', City')

Bucket Course

1. **CO** (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)
2. **UD** (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

3. Bucket algorithm

66

Bucket Job

1. **AD** (Job_Name, Field', Category, Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', job_URL')
2. **SH** (Job_Name, Min_Salary', Avg_Salary', Max_Salary', City')

Bucket Course

1. **CO** (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)
2. **UD** (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)

Query candidate:

1. **AD** (Job_Name, Field', Category, Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Job_URL'), **CO** (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)
2. **AD** (Job_Name, Field', Category, Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Job_URL'), **UD** (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)

3. Bucket algorithm

67

Checking containment:

1. **AD** (Job_Name', Field', Category, Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Job_URL'), **CO** (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

Job (Job_Name, Category, Field', Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Min_Salary', Avg_Salary', Max_Salary', Job_URL'),
Course (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)

2. **AD** (Job_Name, Field, Category, Company, Small_description, Full_description, City, Country, Lat, Lon, Salary, Job_URL), **UD** (Course_Title, Topic, Description, Difficulty, Skills, Preview, Course_URL)

Job (Job_Name, Category, Field', Company', Small_description', Full_description', City', Country', Lat', Lon', Salary', Min_Salary', Avg_Salary', Max_Salary', Job_URL'),
Course (Course_Title, Topic, Description, Difficulty, Category, Preview, Course_URL)

3. Bucket algorithm

68

Checking containment:

- Ovviamente la query Q^1 non è inclusa in Q^2 , ed è ovvio anche il viceversa
- Q^1 e Q^2 sono query massimamente contenute in quanto la riscrittura delle query è la query iniziale stessa

Il risultato è dunque $Q^1 \cup Q^2$



3. Tecnologie utilizzate

69



mongoDB



redis

