

WSIL

INTEGRAZIONE DATI SUL WEB

ANNO 2016-2017

Dario Di Pasquale
Mattia Tomeo
Fabio Napoli

PROBLEMA

PROBLEMA(1)

- Cerchi una **guida rapida** che ti aiuti a scegliere un nuovo linguaggio di programmazione, un suo framework o una particolare tecnologia, da apprendere?
- Ogni anno vengono sviluppati nuovi linguaggi di programmazione o framework!



PROBLEMA(2)

- Spesso alcuni linguaggi nascono e muoiono in poco tempo
- I programmatori devono mantenersi costantemente aggiornati per rispondere alle esigenze di mercato





PERCHÉ PERDERSI IN UN DEDALO DI TECNOLOGIE???

SOLUZIONE



WHAT SHOULD I LEARN

WSIL

COS'È

COS'È (1)

- Recuperare informazioni su un linguaggio di programmazione può essere tedioso, dovendo consultare molti siti e piattaforme differenti



COS'È(2)

‘’ **WSIL** è una piattaforma che permette di ricavare informazioni e statistiche utili relativamente alla popolarità dei linguaggi di programmazione e dei framework’’



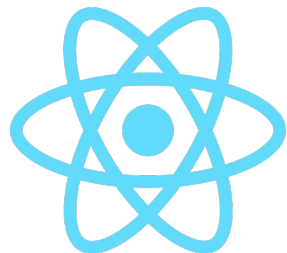
COS'È(3)

‘Permette inoltre di trovare offerte di lavoro e corsi online per essi, attingendo da più fonti’



TECNOLOGIE
UTILIZZATE

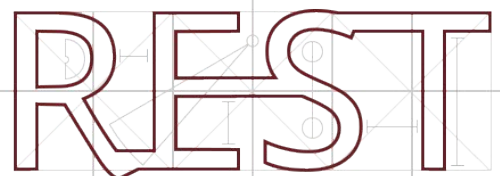
TECNOLOGIE UTILIZZATE



django



django



framework

DEMO

LE FONTI

LE FONTI(1)

GitHub



StackOverflow



LE FONTI(2)

Wikipedia



WIKIPEDIA
The Free Encyclopedia

Google Trends



LE FONTI(3)

Udacity



Coursera



LE FONTI(4)

Authentic Jobs



Indeed



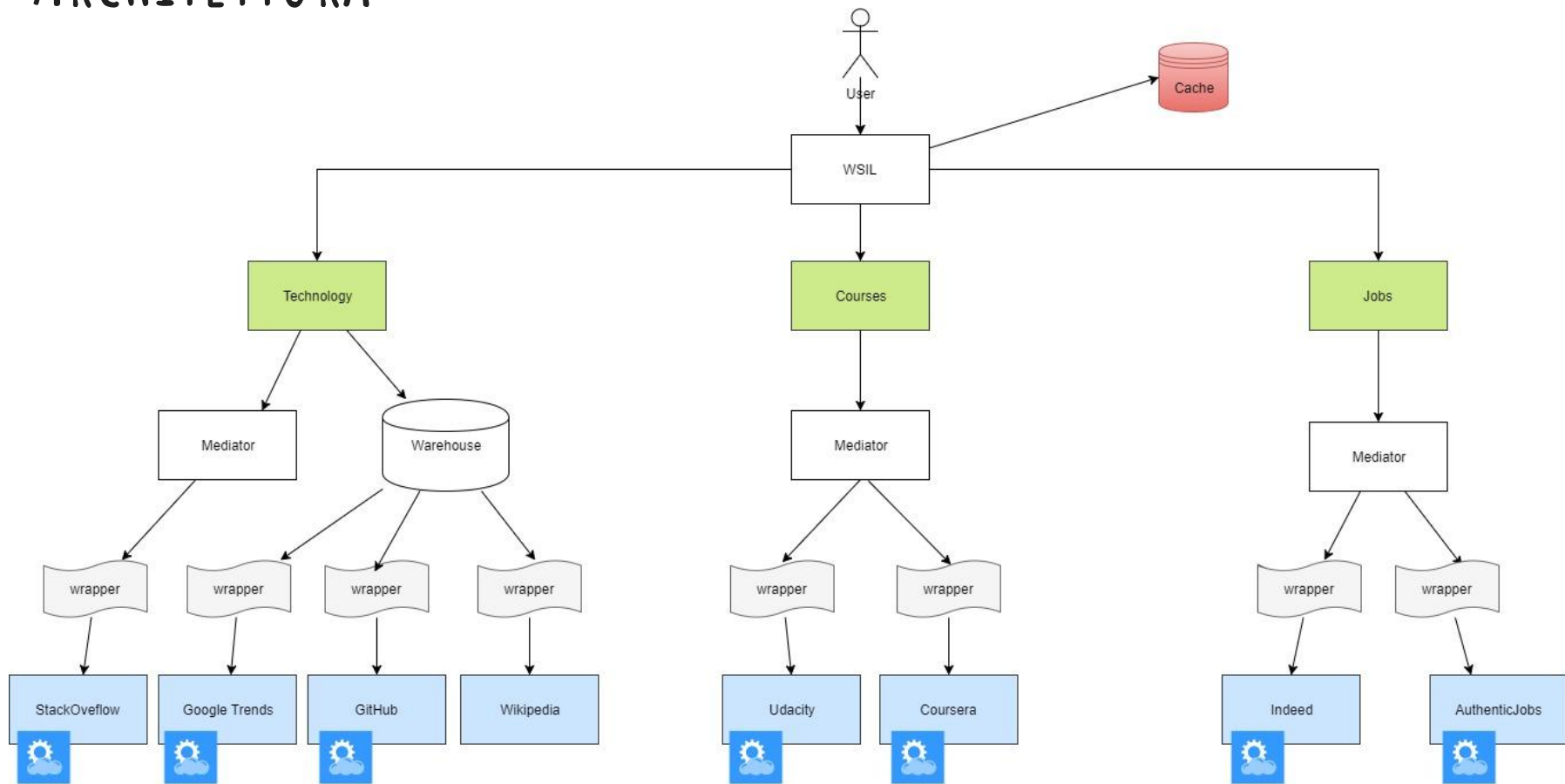
LE FONTI(5)

LA VOLATILITÀ

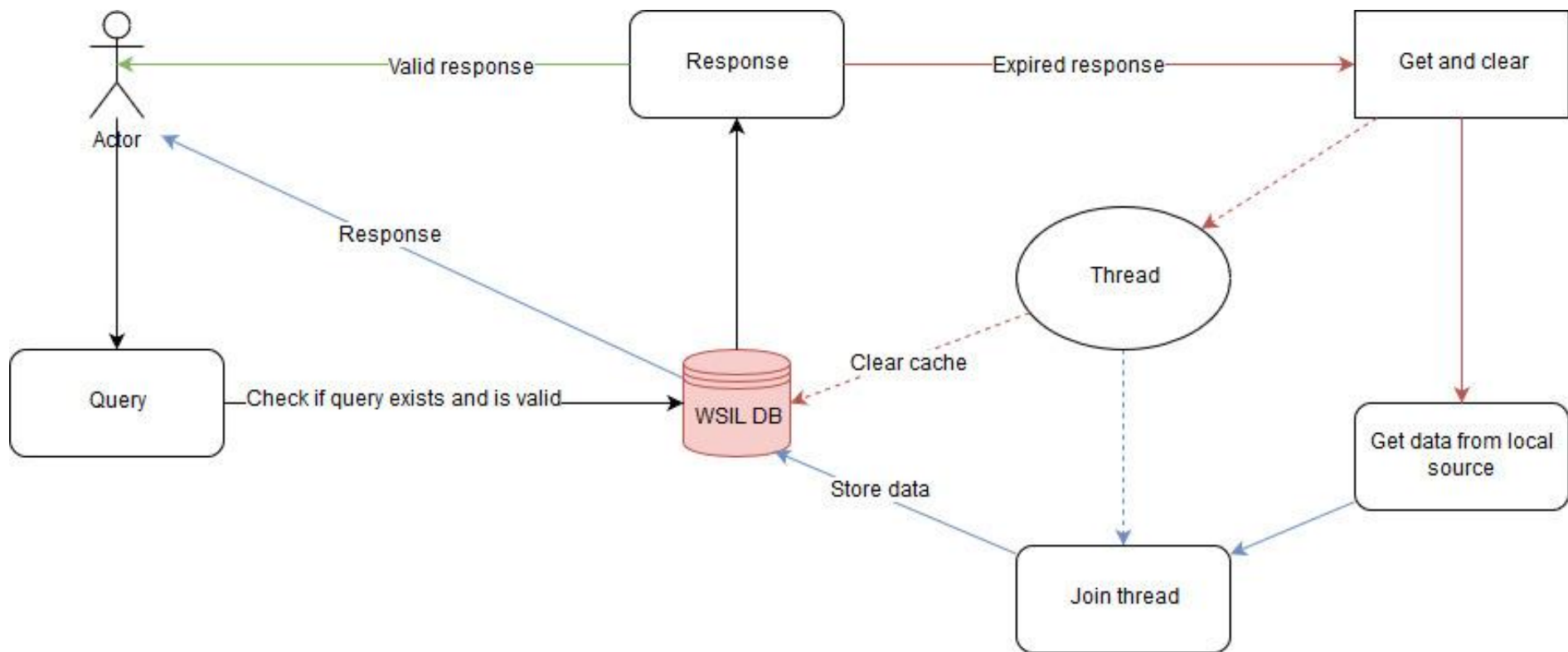
	GitHub	Stack Overflow	Wikipedia	Google Trends	Udacity	Coursera	Authentic Jobs	Indeed
Tipologia	Database	API	WEB PAGE	API	API	API	API	API
Volatilità	MEDIA	ALTA	BASSA	MEDIA	BASSA	BASSA	ALTA	ALTA

ARCHITETTURA

ARCHITETTURA



CACHE



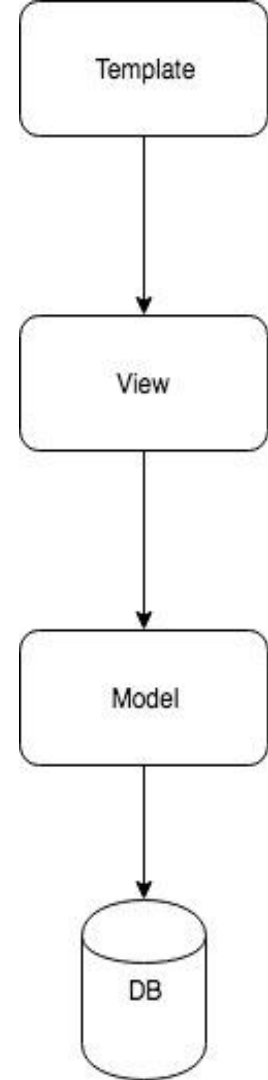
QUERYING

I **model** di Django rappresentano lo schema globale;

L'accesso allo schema locale avviene tramite opportuni metodi codificati nei model;

Abbiamo due tipi di query:

- query per keyword;
- query su tutto il database.



QUERY PER KEYWORD

Coursera

Indeed

Authentic Jobs

In questo caso viene generata la richiesta utilizzando il nome del linguaggio o del framework della quale si vogliono recuperare le informazioni



QUERY DI TUTTO IL DB

Udacity

Vengono scaricati tutti i corsi e inseriti nella cache. Quelli aventi la parola chiave nella descrizione vengono restituiti dalla query.



JOINT FONTE

JOINT FONTI(1)



- RICERCA DELL'INTERESSE NEL TEMPO DI UN LINGUAGGIO
- RICERCA DELL'INTERESSE NEL TEMPO DI UN FRAMEWORK
- RICERCA DELL'INTERESSE PER REGIONE DI UN LINGUAGGIO
- RICERCA DELL'INTERESSE PER REGIONE DI UN FRAMEWORK
- RICERCA DEI PARTNER DEI CORSI

JOINT FONTI(2)

- RICERCA DELL'INTERESSE NEL TEMPO DI UN LINGUAGGIO

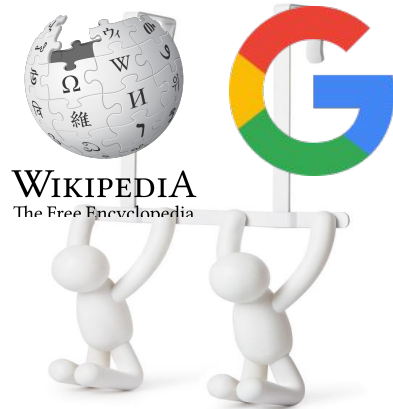
Si cerca prima il nome del linguaggio all'interno della fonte S1(GitHub) per poi ricavare il grado di interesse nella fonte S4(Google Trends)



JOINT FONTI(3)

- RICERCA DELL'INTERESSE NEL TEMPO DI UN FRAMEWORK

Si cerca prima il nome del framework all'interno della fonte S3(Wikipedia) per poi ricavare il grado di interesse dalla fonte S4(Google Trends)



JOINT FONTI(4)

- RICERCA DELL'INTERESSE PER REGIONE DI UN LINGUAGGIO

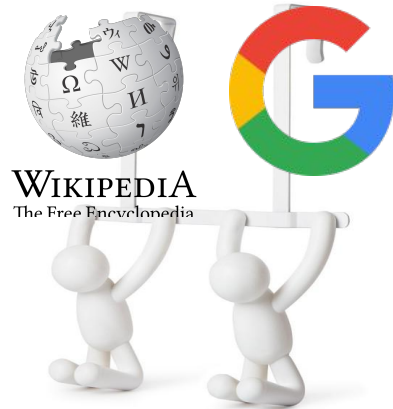
Si cerca prima il nome del linguaggio all'interno della fonte S1(GitHub) per poi ricavare il grado di interesse dalla fonte S4(Google Trends)



JOINT FONTI(5)

- RICERCA DELL'INTERESSE PER REGIONE DI UN FRAMEWORK

Si cerca prima il nome del framework all'interno della fonte S3(Wikipedia) per poi ricavare il grado di interesse dalla fonte S4(Google Trends)



JOINT FONTI(6)

- RICERCA DEI PARTNER DEI CORSI

Dato il nome di un partner prelevato dalla fonte S5(Udacity), viene verificata la sua presenza nella fonte S6(Coursera) in modo da non avere due occorrenze replicate.



SCHEMI LOCALI

SCHEMI LOCALI (S1, S2)



- **S1.language(?language_name, ?numberRepo)**
- **S2.tag(?tag_name, count)**

SCHEMI LOCALI (S3)



WIKIPEDIA
The Free Encyclopedia

- **S3.library_framework(name,type,origin_author,developers,initial_release,stable_release,repository,development_status,language_used,license,website,description)**
- **S3.feature(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks)**

SCHEMI LOCALI (S4)



- **S4.interest_over_time(?keyword, ?date, ?interest_rate)**
- **S4.interest_by_region(?keyword, ?region, ?interest_rate)**

SCHEMI LOCALI (S5, S6)



- `S5.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url)`
- `S5.affiliates(?course_key, ?name, ?image)`
- `S6.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url)`
- `s6.partner(?partner_id, ?partner_name)`
- `S6.partnership(?course_id, ?partner_id)`

SCHEMI LOCALI (S7, S8)

Authentic®



- `S7.job(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lng, ?keywords)`
- `S8.result(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query).`

WRAPPERS

WRAPPER (UDACITY)

Endpoint: <https://www.udacity.com/public-api/v0/courses>

Ts=(key, title, homepage, subtitle, level, starter, image, banner_image, teaser_video, summary, short_summary, required_knowledge, expected_learning, featured, syllabus, faq, full_course_available, expected_duration, expected_duration_unit, new_release, tracks, affiliates, instructors, affiliatesName, affiliatesImage)

Tw=course(?course_id, ?slug, ?course_type, ?course_name, ?logo, ?photo_url, ?description, ?workload, ?url), affiliates(?course_key, ?name, ?image)

WRAPPER (AUTHENTIC JOBS)

Endpoint:

`https://authenticjobs.com/api?api_key=<API_KEY>&method=aj.jobs.search&keywords=<SEARCH_STRING>`

Ts=(id, title, description, perks, howToApply, postDate, relocation_assistance, telecommuting, keywords, applyurl, url, categoryID, category,Name,categoryType,categoryLogo,categoryTagline,category, locationID, locationName, locationCity, locationCountry, locationLatitude, locationLongitude, locationState)

Tw=job(?job_title, ?description, ?post_date, ?company_id, ?location_id, ?keywords), company(?company_id, ?company_name, ?company_url), location(?location_id, ?location_nam, ?lat, ?lng)

WRAPPER (INDEED)

Endpoint:

`http://api.indeed.com/ads/apisearch?publisher=<PUBLISHER_ID>&q=<SEARCH_STRING>&v=<VERSION_NUMBER_API>&latlong=1`

Ts=(query,jobTitle, company, city, state, country, language, formattedLocation, source, date, snippet, url, jobKey, sponsored, station, expired, indeedApply,latitude, longitude)

Tw=(jobTitle, company, description, location, postDate,company_name, ?company_url url, lat, lon,query)

WRAPPER (COURSEERA)

Endpoint:

`https://api.coursera.org/api/courses.v1?q=search&query=<SEARCH_STRING>`

Ts=(id, slug, name, primaryLanguages, subtitleLanguages, partnerLogo, instructors, partnerIds, photoUrl, certificates, description, startDate, workload, previewLink, specializations)

Tw=(course_id, slug, course_type, course_name, logo, photo_url, description, workload, url)

WRAPPER (COURSEERA PARTNER)

Endpoint:

`https://api.coursera.org/api/partners.v1/:course_id/includes=partnerIds`

Ts=(id, name, shortName, description, banner, courseIds, instructorIds, links, location)

Tw=partner(partner_id, partner_name), partnership(partner_id, course_id)

WRAPPER (STACKOVERFLOW)

Endpoint:

`https://api.stackexchange.com/2.2/tags?order=desc&sort=popular&inname=<keyword>&site=stackoverflow`

`Ts=(has_synonyms, is_moderator_only, is_required, count, name)`

`Tw=tag(name, count)`

WRAPPER (GOOGLE TRENDS)

Usiamo la libreria **Pytrends** grazie ai seguenti metodi:

```
TrendReq(GOOGLE_USR, GOOGLE_PWD), build_payload(kw_list),  
pytrends.interest_over_time(), pytrends.interest_by_region()
```

Interest over time

```
Ts = Tw=(keyword, interval, interest_rate)
```

Interest by region

```
Ts = Tw = (keyword, region, interest_rate)
```

WRAPPER (GITHUB)

- Si utilizza **GitArchive**
- I dati messi a disposizione da GitHub Archive sono accessibili scaricando il file
<http://data.githubarchive.org/YYYY-MM-DD-HH.json.gz>;
- I dati vengono filtrati e da questi vengono ricavate delle statistiche;
- I risultati dell'elaborazione vengono salvati nel database.

WRAPPER (GITHUB)(2)

- **Ts**=language(language_name, repo_count)
- **Tw** = RepositoryUsingIt(language_name, repo_count),
Language(language_name).

WRAPPER (WIKIPEDIA)

Link: https://en.wikipedia.org/wiki/Comparison_of_web_frameworks

- **library_framework:**

XPATH: `//*[@id="mw-content-text"]/div/table[position() < 18 and position() > 2]/tr[position() > 1]`

XPATH: `./../preceding-sibling::h3[1]/span/text()`

Ts1 = (name, stable_release, release_date, license, language_used)

WRAPPER (WIKIPEDIA)

link pagina del framework/libreria

XPATH: `//*[@id="mw-content-text"]/div/table[position() <18]/tr/th/a@href`

XPATH: `//*[@id="mw-content-text"]/div/table[position()<=3]/tr`

Ts2 = (initial_release, repository, development_status, type, license, website)

XPATH: `//*[@id="mw-content-text"]/div/p[1] |//*[@id="mw-content-text"]/p[1]`

Ts3 = (description).

Tw = library_framework(name, type, initial_release, stable_release, repository, development_status, language_used, license, website, description)

WRAPPER (WIKIPEDIA)

- **features**

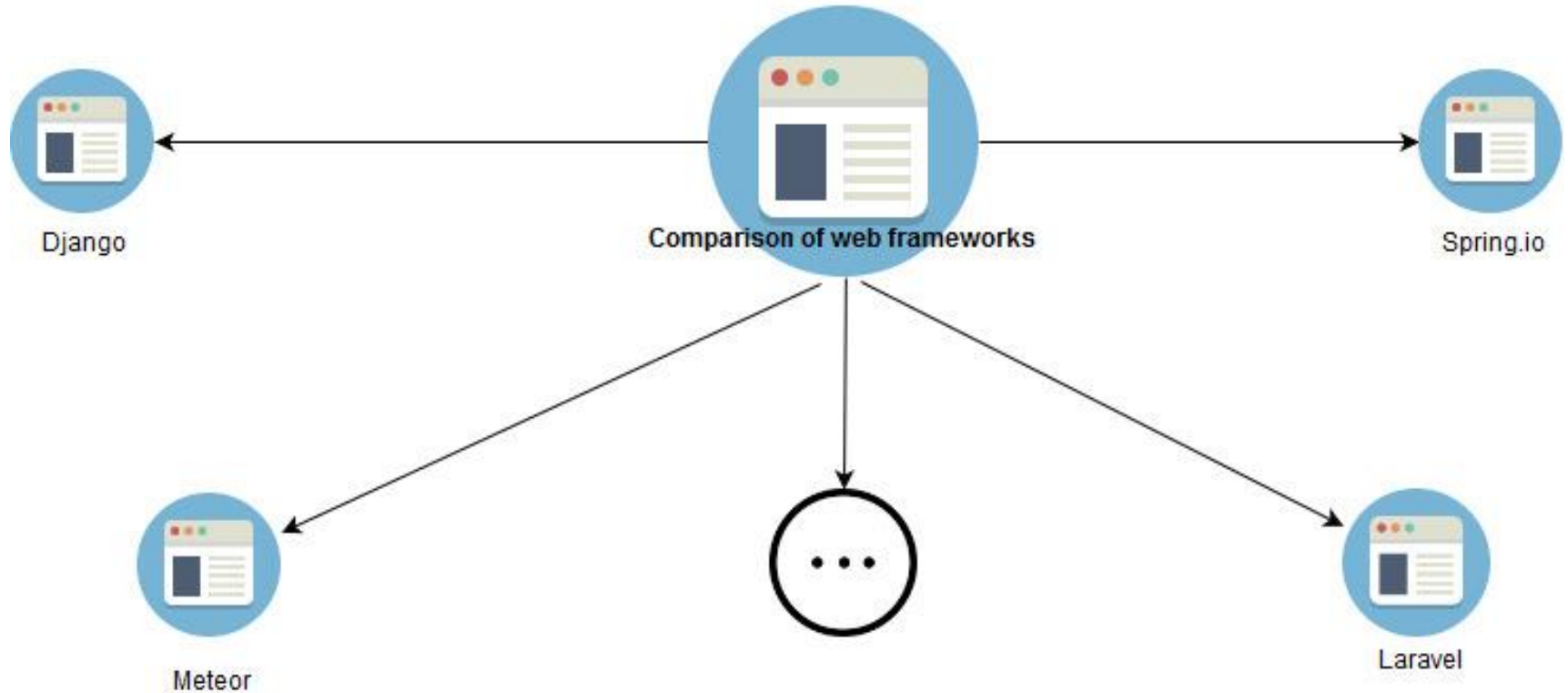
Ts = (library_framework_name, ajax, mvc_framework, mvc_push_pull, localization, orm, testing_framework, db_migration_framework, security_framework, template_framework, caching_framework, form_validation_frameworks).

XPATH: `//*[@id="mw-content-text"]/div/table[position() >= 18]/tr`

Tw = Ts

WRAPPER (WIKIPEDIA)

```
//*[@id="mw-content-text"]/div/table[position() < 18]/tbody/tr/th/a@href
```



SCHEMA GLOBALE

SCHEMA GLOBALE(1)

- **Language**(?name)
- **LibraryOrFramework**(?name,?type,?initial_release,?stable_release,?repository,?development_status,?language,?license,?website,?description)
- **Features**(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework,?security_framework,?template_framework, ?caching_framework, ?form_validation_frameworks)

SCHEMA GLOBALE(2)

- **RepositoriesUsingIt**(?language_name, ?repo_count)
- **QuestionOnIt**(?tag, ?count)
- **InterestOverTimeLanguage**(?language_name, ?date, ?interest_rate)
- **InterestOverTimeFrameworkLibrary**(?fw_or_lib, ?date, ?interest_rate)
- **InterestByRegionLanguage**(?language, ?region, ?interest_rate)

SCHEMA GLOBALE(3)

- **InterestByRegionFrameworkLibrary**(?fw_or_lib, ?region, ?interest_rate)
- **Course**(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url)
- **CoursePartner**(?course_id, ?partner_id, ?partner_name)
- **Job**(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query)

MAPPING GAV

MAPPING GAV(1)

- **Language**(?name) :- S1.language(?name, _).
- **LibraryOrFramework**(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description) :- S3.library_framework(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description).

MAPPING GAV(2)

- **Features**(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks) :- S3.feature(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks).

MAPPING GAV(3)

- **RepositoriesUsingIt**(?language_name, ?repo_count) :-
S1.language(?language_name, ?repo_count).`
- **QuestionOnIt**(?tag, ?count) :- S2.tag(?tag, ?count).
- **InterestOverTimeLanguage**(?language_name, ?date, ?interest_rate) :-
S4.interest_over_time(?language_name, ?date, ?interest_rate),
S1.language(?language_name, _).

MAPPING GAV(4)

- **InterestOverTimeFrameworkLibrary**(?fw_or_lib, ?date, ?interest_rate) :-
S4.interest_over_time(?fw_or_lib, ?date, ?interest_rate),
S3.library_framework(?fw_or_lib, _, _, _, _, _, _, _, _).
- **InterestByRegionLanguage**(?language, ?region, ?interest_rate):-
S4.interest_by_region(?language, ?region, ?interest_rate),
S1.language(?language, _).
- **InterestByRegionFrameworkLibrary**(?fw_or_lib, ?region, ?interest_rate):-
S4.interest_by_region(?fw_or_lib, ?region, ?interest_rate),
S3.library_framework(?fw_or_lib, _, _, _, _, _, _, _, _).

MAPPING GAV(5)

- **Course**(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url) :- S5.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url).
- **Course**(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url) :- S6.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url).
- **CoursePartner**(?course_id, ?partner_id, ?partner_name) :- S5.affiliates(?course_id, ?partner_name), S6.partner(?partner_id, ?partner_name).

MAPPING GAV(6)

- **CoursePartner**(?course_id, ?partner_id, ?partner_name) :- S6.partnership(?course_id, ?partner_id), S6.partner(?partner_id, ?partner_name).
- **Job**(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query) :- S7.job(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query).
- **Job**(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query) :- S8.result(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query).

MAPPING LAV

MAPPING LAV(1)

- **S1.language**(?language_name, ?numberRepo) :- Language(?language_name), RepositoriesUsingIt(?language_name,?numberRepo),InterestOverTimeLanguage(?language_name, _, _),InterestByRegionLanguage(?language_name, _, _).
- **S2.tag**(?tag, ?count) :- QuestionOnIt(?tag, ?count).
- **S3.library_framework**(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description) :- LibraryOrFramework(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description), InterestOverTimeFrameworkLibrary(?name, _, _), InterestByRegionFrameworkLibrary(?name, _, _).

MAPPING LAV(2)

- **S3.feature**(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks) :-
Features(?library_framework_name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks).

MAPPING LAV(3)

- **S4.interest_over_time**(?keyword, ?date, ?interest_rate) :-
InterestOverTimeLanguage(?keyword, ?date, ?interest_rate).
- **S4.interest_over_time**(?keyword, ?date, ?interest_rate):-
InterestOverTimeFrameworkLibrary(?keyword, ?date, ?interest_rate).
- **S4.interest_by_region**(?keyword, ?region, ?interest_rate) :-
InterestByRegionLanguage(?keyword, ?region, ?interest_rate).
- **S4.interest_by_region**(?keyword, ?region, ?interest_rate) :-
InterestByRegionFrameworkLibrary(?keyword, ?region, ?interest_rate).

MAPPING LAV(4)

S5.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url) :- Course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url).

S5.affiliates(?course_id, ?partner_name) :- CoursePartner(?course_id, ?partner_name).

S6.course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url) :- Course(?course_id, ?slug, ?course_type, ?logo, ?photo_url, ?description, ?workload, ?url).

MAPPING LAV(5)

- **S6.partnership**(?course_id, ?partner_id) :- CoursePartner(?course_id, ?partner_id, _, _).`
- **S6.partner**(?partner_id, ?partner_name) :- CoursePartner(_, ?partner_id, ?partner_name).
- **S7.job**(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query) :- Job(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query).
- **S8.result**(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query) :- Job(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?query).

QUERY

QUERY(1)

ELENCO DEI LINGUAGGI

- **DATALOG** : `q(?name) :- Language(?name).`
- **SQL** : `select name
from Language`



QUERY(2)

ELENCO DEI NOMI DELLE LIBRERIE E DEI FRAMEWORK DISPONIBILI

- **DATALOG** : `q(?name) :- LibraryOrFramework(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description).`
- **SQL** : `select name`
`from LibraryOrFramework`



QUERY(3)

NOMI DELLE LIBRERIE E DEI FRAMEWORK DISPONIBILI CON RELATIVO LINGUAGGIO

- **DATALOG** : `q(?name,?language_name):-LibraryOrFramework(?name, ?type, ?initial_release, ?stable_release, ?repository, ?development_status, ?language, ?license, ?website, ?description), Language(?language_name).`
- **SQL**: `select LibraryOrFramework.name, Language.name
from LibraryOrFramework join Language
on LibraryOrFramework.language_used = Language.name`



QUERY(4)

NOMI DELLE LIBRERIE E DEI FRAMEWORK DISPONIBILI DI UN DATO LINGUAGGIO

- **DATALOG** : `q(?fw_name, 'LINGUAGGIO') :-`
`LibraryOrFramework(?fw_name, ?type, ?initial_release,`
`?stable_release, ?repository, ?development_status, 'LINGUAGGIO',`
`?license, ?website, ?description), Language('LINGUAGGIO').`
- **SQL**:`select LibraryOrFramework.name`
`from LibraryOrFramework join Language`
`on LibraryOrFramework.used_language = Language.name`
`where Language.name = 'LINGUAGGIO'`



QUERY(5)

NOMI DELLE LIBRERIE E DEI FRAMEWORK CON LE CARATTERISTICHE PRINCIPALI

- **DATALOG** : `q(?name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks) :- LibraryOrFramework(?name, _, _, _, _, _, _, _, _, _), Features(?name, ?ajax, ?mvc_framework, ?mvc_push_pull, ?localization, ?orm, ?testing_framework, ?db_migration_framework, ?security_framework, ?template_framework, ?caching_framework, ?form_validation_frameworks).`



QUERY(5)

NOMI DELLE LIBRERIE E DEI FRAMEWORK CON LE CARATTERISTICHE PRINCIPALI

- **SQL** : `select Features.*`
 `from LibraryOrFramework join Features`
 `on LibraryOrFramework.name=Features.library_framework_name`



QUERY(6)

MOSTRA PER OGNI LINGUAGGIO L'UTILIZZO SU GITHUB

- **DATALOG** : `q(?language_name, ?repo_count) :-
RepositoriesUsingIt(?language_name, ?repo_count),
Language(?language_name).``
- **SQL**: `select repository_count, language_name
from RepositoriesUsingIt join Language
where RepositoriesUsingIt.language_name=Language.name`



QUERY(7)

MOSTRA PER OGNI LINGUAGGIO IL NUMERO DI DOMANDE SU STACKOVERFLOW

- **DATALOG** : `q(?language_name, ?question_count) :-
QuestionOnIt(?language_name, ?question_count),
Language(?language_name).`
- **SQL** : `select name, count
from QuestionOnIt join Language
on QuestionOnIt.tag = Language.name`



QUERY(8)

MOSTRA PER OGNI LINGUAGGIO, IL GRADO DI INTERESSE NEL TEMPO SU GOOGLE TRENDS

- **DATALOG** : `q(?language_name, ?date, ?interest_rate) :- InterestOverTimeLanguage(?language_name, ?date, ?interest_rate), Language(?language_name).`
- **SQL**: `select interest_rate, date, language_name from InterestOverTimeLanguage join Language on InterestOverTimeLanguage.language_name = Language.name`



QUERY(9)

MOSTRA PER OGNI FRAMEWORK,IL GRADO DI INTERESSE NEL TEMPO SU GOOGLE TRENDS

- **DATALOG** : `q(?fw_name, ?date, ?interest_rate) :- InterestOverTimeFrameworkLibrary(?fw_name, ?date, ?interest_rate), LibraryOrFramework(?fw_name, _, _, _, _, _, _, _, _)`.
- **SQL**: `select interest_rate, date, fw_or_lib from InterestOverTimeFrameworkLibrary join LibraryOrFramework on InterestOverTimeFrameworkLibrary.fw_or_lib = LibraryOrFramework.name``



QUERY(10)

MOSTRA PER OGNI LINGUAGGIO, GRADO DI INTERESSE PER REGIONE SU GOOGLE TRENDS

- **DATALOG** : `q(?language_name, ?region, ?interest_rate) :- InterestByRegionLanguage(?language_name, ?region, ?interest_rate), Language(?language_name).`
- **SQL**: `select interest_rate, region, language from InterestByRegionLanguage join Language on InterestByRegionLanguage.language = Language.name`



QUERY(11)

MOSTRA PER OGNI FRAMEWORK, GRADO DI INTERESSE PER REGIONE SU GOOGLE TRENDS

- **DATALOG** : `q(?fw_or_lib, ?region, ?interest_rate) :- InterestByRegionFrameworkLibrary(?fw_or_lib, ?region, ?interest_rate), LibraryOrFramework(?fw_or_lib, _, _, _, _, _, _, _, _)`.

- **SQL**: `select interest_rate, region, fw_or_lib from InterestByRegionFrameworkLibrary join Language on InterestByRegionFrameworkLibrary.fw_or_lib = LibraryOrFramework.name`



QUERY(12)

MOSTRA TUTTI I CORSI PER UN DETERMINATO LINGUAGGIO

- **DATALOG** : `q(?slug, ?course_type, ?language_name, ?logo, ?photo_url, ?description, ?workload, ?url) :- Course(_, ?slug, ?course_type, ?language_name, ?logo, ?photo_url, ?description, ?workload, ?url), Language(?language_name).`

`q2(?slug, ?course_type, ?logo, ?photo_url, ?language_name, ?workload, ?url) :- Course(_, ?slug, ?course_type, ?logo, ?photo_url, ?language_name, ?workload, ?url), Language(?language_name).`



QUERY(12)

MOSTRA TUTTI I CORSI PER UN DETERMINATO LINGUAGGIO

- **SQL:**

```
select slug, course_type, Language.name, logo, photo_url,  
description, workload, url  
from Course join Language  
where Course.description like %Language.name% OR Course.slug  
like %Language.name%
```



QUERY(13)



MOSTRA TUTTI I CORSI PER UN DETERMINATO FRAMEWORK O LIBRERIA

- **DATALOG:** `q(?slug, ?course_type, ?fw_name, ?logo, ?photo_url, ?description, ?workload, ?url) :- Course(_, ?slug, ?course_type, ?fw_name, ?logo, ?photo_url, ?description, ?workload, ?url), LibraryOrFramework(?fw_name, _, _, _, _, _, _ _ , _).`

`q2(?slug, ?course_type, ?logo, ?photo_url, ?fw_name, ?workload, ?url) :- Course(_, ?slug, ?course_type, ?logo, ?photo_url, ?fw_name, ?workload, ?url), LibraryOrFramework(?fw_name, _, _, _, _, _, _ _ , _).`

QUERY(13)

MOSTRA TUTTI I CORSI PER UN DETERMINATO FRAMEWORK O LIBRERIA

- **SQL:** `select slug, course_type, LibraryOrFramework.name, logo, photo_url, description, workload, url
from Course join LibraryOrFramework
where Course.description like %LibraryOrFramework.name% OR
Course.slug like %LibraryOrFramework.name%`



QUERY(14)

PARTNER PER OGNI CORSO

- **DATALOG:** `q(?partner_id, ?partner_name) :- Course(_?course_id, _, _, _, _, _, _, _), CoursePartner(?course_id, ?partner_name).`
- **SQL:** `select partner_name, slug
from Course join CoursePartner
on Course.course_id = CoursePartner.course_id`



QUERY(15)

TUTTI I LAVORI PER UN LINGUAGGIO

- **DATALOG:** `q(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- Job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name), Language(?language_name).`
- **SQL:** `select * from Job join Language where Job.description like %Language.name%`



QUERY(16)

TUTTI I LAVORI PER UN FRAMEWORK

- **DATALOG:** `q(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?fw_name) :- Job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?fw_name), LibraryOrFramework(?fw_name, _, _, _, _, _, _)`
- **SQL:** `select * from Job join LibraryOrFramework where Job.description like %LibraryOrFramework.name%`



QUERY
REFORMULATION GAV

QUERY REFORMULATION GAV(1)

QUERY:

```
q(?job_title, ?description, ?post_date, ?company_name,  
?company_url, ?location_name, ?lat, ?lon, ?language_name) :-  
Job(?job_title, ?description, ?post_date, ?company_name,  
?company_url, ?location_name, ?lat, ?lon, ?language_name),  
Language(?language_name).
```

QUERY REFORMULATION GAV(2)

RULES UNFOLDING GAV:

- `q(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- S7.job(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?language_name), S1.language(?language_name, _).`
- `q(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- S8.result(?job_title, ?description, ?post_date, ?company_name, ?company_url, ?location_name, ?lat, ?lon, ?language_name), S1.language(?language_name, _).`

QUERY
REFORMULATION LAV

QUERY REFORMULATION LAV(1)

QUERY:

```
q(?job_title, ?description, ?post_date, ?company_name,  
?company_url, ?location_name, ?lat, ?lon, ?language_name) :-  
Job(?job_title, ?description, ?post_date, ?company_name,  
?company_url, ?location_name, ?lat, ?lon, ?language_name),  
Language(?language_name).
```

QUERY REFORMULATION LAV(2)

BUCKET CREATION LAV:

ATOM:

- **Job**(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name)
- **Language**(?language_name).

QUERY REFORMULATION LAV(3)

BUCKET CREATION LAV:

BUCKET:

- **BucketG1** : `S8.result(?job_title, ?company_name, ?location_name, ?post_date, ?description, ?url, ?lat, ?lon, ?language_name) , S7.job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name)`
- **BucketG2** : `S1.language(?language_name, ?v1)`

QUERY REFORMULATION LAV(4)

CANDIDATE REWRITES CREATION LAV:

- **q1(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- S7.job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name), S1.language(?language_name, ?v1).**
- **q2(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- S8.result(?job_title, ?company_name, ?location_name, ?post_date, ?description, ?url, ?lat, ?lon, ?language_name), S1.language(?language_name, ?v1).**

QUERY REFORMULATION LAV(5)

REWRITES VERIFICATION LAV:

- **exp_q1(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- Job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name), Language(?language_name), RepositoriesUsingIt(?language_name, ?v1), InterestOverTimeLanguage(?language_name, ?v2, ?v3), InterestByRegionLanguage(?language_name, ?v4, ?v5)**
- **exp_q2(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- Job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name), Language(?language_name), RepositoriesUsingIt(?language_name, ?v1), InterestOverTimeLanguage(?language_name, ?v2, ?v3), InterestByRegionLanguage(?language_name, ?v4, ?v5)**

QUERY REFORMULATION LAV(6)

CHECKING CONTAINMENT

exp_q1 = exp_q2 VERIFY:

- `exp_q1(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :-
Job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name),
Language(?language_name), RepositoriesUsingIt(?language_name, ?v1), InterestOverTimeLanguage(?language_name, ?v2, ?v3),
InterestByRegionLanguage(?language_name, ?v4, ?v5).`

QUERY REFORMULATION LAV(7)

CHECKING CONTAINMENT

- **CREATE CANONICAL DATABASE FOR q1: for all variables v, v is the constant**

Job	("job_title", "description", "post_date", "company_name", "url", "location_name", "lat", "lon", "language_name")
Language	("language_name")
RepositoriesUsingIt	("language_name", "v1")
InterestOverTimeLanguage	("language_name", "v2", "v3")
InterestByRegionLanguage	("language_name", "v4", "v5")

- **FrozenHead** = q1("job_title", "description", "post_date", "company_name", "url", "location_name", "lat", "lon", "language_name")

QUERY REFORMULATION LAV(8)

CHECKING CONTAINMENT

- **query sul db canonico:**

le variabili contenute in **Job**(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) possono essere unificate con i valori contenuti nella tupla di **Job** del database canonico, ottenendo il fatto `Job("job_title", "description", "post_date", "company_name", "url", "location_name", "lat", "lon", "language_name");`

QUERY REFORMULATION LAV(9)

CHECKING CONTAINMENT

- **query sul db canonico:**

dall'unificazione della fase precedente otteniamo il fatto **Language**("language_name"), che è contenuto nel database canonico. Da questo otteniamo il risultato finale della query **q("job_title", "description", "post_date", "company_name", "url", "location_name", "lat", "lon", "language_name").**

QUERY REFORMULATION LAV(10)

CHECKING CONTAINMENT

```
q("job_title", "description", "post_date", "company_name",  
"url", "location_name", "lat", "lon", "language_name").
```

=

```
q1("job_title", "description", "post_date", "company_name",  
"url", "location_name", "lat", "lon", "language_name")
```

$q1 \subseteq q$

QUERY REFORMULATION LAV(11)

CHECKING CONTAINMENT

- `q1(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name) :- S7.job(?job_title, ?description, ?post_date, ?company_name, ?url, ?location_name, ?lat, ?lon, ?language_name), S1.language(?language_name, ?v1)`;`
- `q2(?job_title,?description,?post_date,?company_name,?url,?location_name,?lat,?lon,?language_name) :- S8.result(?job_title, ?company_name, ?location_name, ?post_date, ?description, ?url, ?lat, ?lon, ?language_name), S1.language(?language_name, ?v1).`

q1 OR q2.

SVILUPPI FUTURI

SVILUPPI FUTURI

- Integrazione di altre fonti, come videocorsi (es. Udemy);
- Possibilità di votare i linguaggi e i framework preferiti



GRAZIE PER
L'ATTENZIONE