DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Databases Project – Spring 2018

Team No: 17

Names: Cho Hyun Jii, Poopalasingam Kirusanth, Reetz Florian

Contents

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

What has changed

- Updated DDL:
    - Added index creation
    - Increased length of VARCHAR to meet the import files requirement (e.g. clip.clip_title was increased from 255 to 1024)
    - Changed all CHAR to VARCHAR
- Updated ERM diagram
- Added additional assumptions regarding the spouses and language files have been added
- Explained database selection
- Fixed all queries from deliverable 2, added first five rows.
- Added Deliverable 3 section

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 1

Assumptions

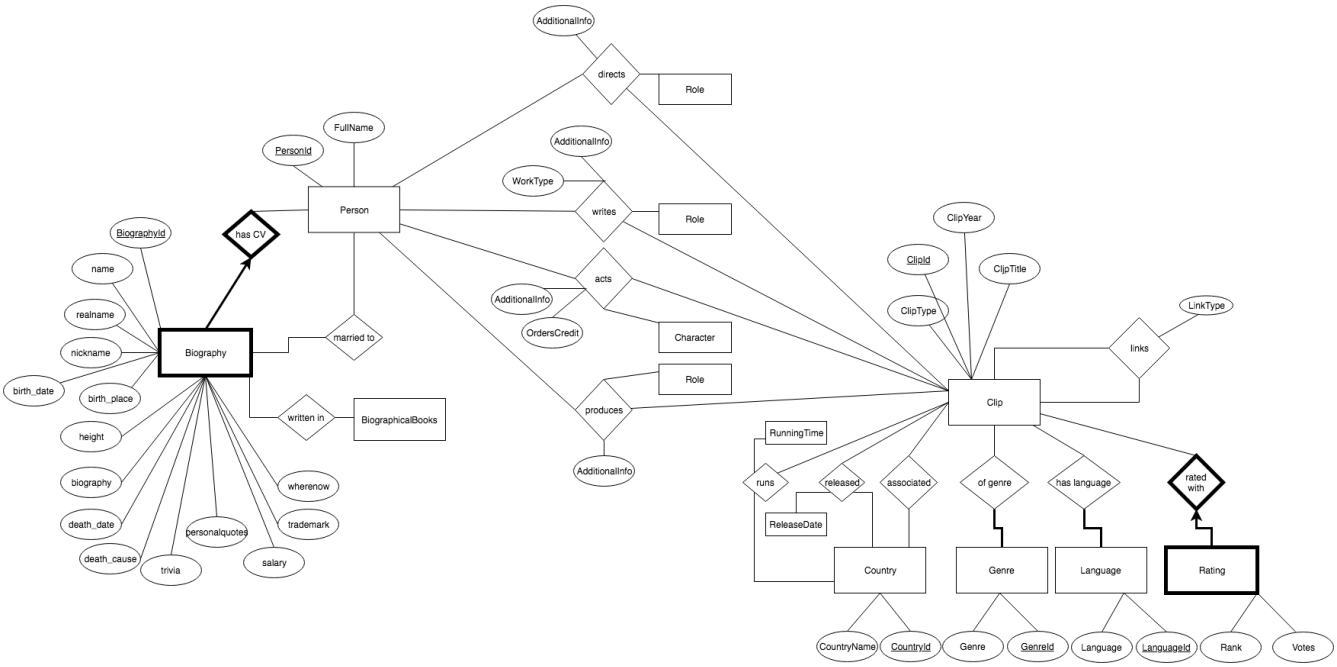
We make the listed design decisions based on the following assumptions:

- *Actors, Writers, Producers, Directors* are not modelled as entities but as relations between entity *Person* and entity *Clip* because one *Person* can perform multiple jobs and we assume that queries become easier if the jobs are relations.
- *ReleaseDate* and *RunningTime* have been moved to a single entity since both describe additional information about a clip released in a country.
- *Biography* has been modelled as one entity since we don't know the exact data yet which could justify a subdivision.
- We model *Country* as a separate entity to be able to describe other relations to *Country* like we do with the *released* relation.
- *Rating* is modelled as a weak entity since it is only associated to a single *Clip* (a *Rating* without an existing *Clip* does not make sense)
- *Biography* is also modelled as a weak entity (we cannot have a *Biography* for a non-existing *Person*)
- We consider the attribute "language" of entity *Language* a primary key because it is a unique and necessary attribute which cannot change.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Entity Relationship Schema

Schema



Description

We apply the following constraints

Key Constraints:

- One *Biography* is only associated with one *Person*. A *Person* can have many *Biographies*. (One-to-Many)
- One *Rating* is associated to one *Clip*. A *Clip* can have many *Ratings*. (One-to-Many)
- Every other relation is modelled as many-to-many (e.g. every *Person* can direct, write, act or produce in many *Clips*. A *Clip* can have many producer, director, actor or writer. Every *Clip* can have many *Countries* associated. A *Country* can be associated with many *Clips*.)
- In principle, a *Clip* should have at least one *Country* and at least one *Language* but we do not want to enforce this because we do not know the data yet.

Participation Constraints:

- Each *Rating* is assigned to exactly one *Clip*.
- Each *Biography* is assigned to exactly one *Person*
- Each *Language* and *Genres* must have at least one *Clip* to be relevant to this database.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

- Every other relation participates partially (e.g. *Country*: A *Country* does not need to be associated with a *Clip* if it is a *Country* where a *Clip* is released. A *Clip* does not need to have a *Country* association)

Relational Schema

ER schema to Relational schema

The following describes how we translate the constraints

- Translating the many-to-many relationships "directed, acted, produced, wrote" we did not use a superkey as primary key (i.e. primary key(person_id,clip_id)) because this key does not allow for one person to act two different roles in the same clip.
- In contrast, translating the many-to-many relationships "clip_country, clip_genre, clip_language" we use a superkey as primary key (e.g. primary key(clip_id,genre_id))
- The key constraints in "clip_rating" and "biography" are enforced by setting the foreign key NOT NULL.

The following attributes are defined as unique:

- Country(country_name)
- Genre(Genre)

DDL

```
CREATE TABLE Person
(
  person_id INTEGER,
  fullname  VARCHAR(1024) UNIQUE,
  PRIMARY KEY (person_id)
);


CREATE TABLE Clip
(
  clip_id    INTEGER,
  clip_type  VARCHAR(20),
  clip_year  DATE,
  clip_title VARCHAR(1024),
  PRIMARY KEY (clip_id)
);

CREATE TABLE Directs
(
  person_id       INTEGER,
  clip_id         INTEGER,
  additional_info VARCHAR(1024),
  role            VARCHAR(1024),
  PRIMARY KEY (person_id, clip_id, role),
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

```
  FOREIGN KEY (person_id) REFERENCES Person (person_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id)
);

CREATE TABLE Acts
(
  person_id       INTEGER,
  clip_id         INTEGER,
  additional_info VARCHAR(1024),
  orders_credit   INTEGER,
  character       VARCHAR(1024),
  PRIMARY KEY (person_id, clip_id, character),
  FOREIGN KEY (person_id) REFERENCES Person (person_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id)
);


CREATE TABLE Produces
(
  person_id       INTEGER,
  clip_id         INTEGER,
  additional_info VARCHAR(300),
  role            VARCHAR(200),
  PRIMARY KEY (person_id, clip_id, role),
  FOREIGN KEY (person_id) REFERENCES Person (person_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id)
);

CREATE TABLE Writes
(
  person_id       INTEGER,
  clip_id         INTEGER,
  additional_info VARCHAR(400),
  work_type       VARCHAR(50),
  role            VARCHAR(100),
  PRIMARY KEY (person_id, clip_id, role),
  FOREIGN KEY (person_id) REFERENCES Person (person_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id)
);

CREATE TABLE ClipLinks
(
  cliplink_id  INTEGER,
  clip_from_id INTEGER,
  clip_to_id   INTEGER,
  link_type    VARCHAR(255),
  PRIMARY KEY (cliplink_id),
  FOREIGN KEY (clip_from_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (clip_to_id) REFERENCES Clip (clip_id)
);

CREATE TABLE Country
(
  country_id  INTEGER,
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

```
  countryname VARCHAR(100),
  PRIMARY KEY (country_id),
  constraint ux_country unique (countryname)
);

CREATE TABLE Genre
(
  genre_id INTEGER,
  genre    VARCHAR(20),
  PRIMARY KEY (genre_id),
  constraint ux_genre unique (genre)
);

CREATE TABLE Language
(
  language_id INTEGER,
  language    VARCHAR(70),
  PRIMARY KEY (language_id),
  constraint ux_language unique (language)
);

CREATE TABLE Clip_country
(
  clip_id    INTEGER,
  country_id INTEGER,
  PRIMARY KEY (clip_id, country_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (country_id) REFERENCES Country (country_id)
);

CREATE TABLE Clip_genre
(
  clip_id  INTEGER,
  genre_id INTEGER,
  PRIMARY KEY (clip_id, genre_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (genre_id) REFERENCES Genre (genre_id)
);

CREATE TABLE Clip_language
(
  clip_id     INTEGER,
  language_id INTEGER,
  PRIMARY KEY (clip_id, language_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (language_id) REFERENCES Language (language_id)
);

CREATE TABLE Clip_rating
(
  clip_id   INTEGER NOT NULL,
  rating_id INTEGER,
  rank      NUMBER(10),
  votes     NUMBER(10),
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
  PRIMARY KEY (rating_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id)
    ON DELETE CASCADE
);

CREATE TABLE Released
(
  clip_id      INTEGER,
  country_id   INTEGER,
  release_date DATE,
  PRIMARY KEY (clip_id, country_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (country_id) REFERENCES Country (country_id)
);

CREATE TABLE Runs
(
  clip_id      INTEGER,
  country_id   INTEGER,
  running_time NUMBER(10),
  PRIMARY KEY (clip_id, country_id),
  FOREIGN KEY (clip_id) REFERENCES Clip (clip_id),
  FOREIGN KEY (country_id) REFERENCES Country (country_id)
);

CREATE TABLE Biography
(
  biography_id      INTEGER,
  name              VARCHAR(256),
  realname          VARCHAR(256),
  nickname          VARCHAR(512),
  birth_date        DATE,
  birth_place       VARCHAR(128),
  height            FLOAT4,
  biography         VARCHAR(80000),
  biographer        VARCHAR(128),
  death_date        DATE,
  death_place       VARCHAR(128),
  death_cause       VARCHAR(256),
  trivia            VARCHAR(59000),
  personalquotes    VARCHAR(36000),
  salary            VARCHAR(3000),
  trademark         VARCHAR(6400),
  wherenow          VARCHAR(10000),
  person_id         INTEGER NOT NULL,
  FOREIGN KEY (person_id) REFERENCES Person (person_id)
    ON DELETE CASCADE,
  PRIMARY KEY (biography_id)
);

CREATE TABLE BiographicalBooks
(
  book_id      INTEGER,
  title        VARCHAR(100),
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

```
  biography_id INTEGER NOT NULL,
  FOREIGN KEY (biography_id) REFERENCES Biography (biography_id),
  PRIMARY KEY (book_id)
);

CREATE TABLE Married_to
(
  married_id   INTEGER NOT NULL,
  biography_id INTEGER NOT NULL,
  person_id    INTEGER NOT NULL,
  marrie_date  VARCHAR(50),
  state        VARCHAR(20),
  children     VARCHAR(20),
  PRIMARY KEY (married_id),
  FOREIGN KEY (biography_id) REFERENCES Biography (biography_id),
  FOREIGN KEY (person_id) REFERENCES person (person_id)
);


-- creating indexes:
CREATE EXTENSION IF NOT EXISTS pg_trgm;
create index ix_person_name ON person using gin (fullname gin_trgm_ops);
create index ix_clip_title ON clip using gin (clip_title gin_trgm_ops);
create index ix_language ON language using gin (language gin_trgm_ops);
create index ix_country ON country using gin (countryname gin_trgm_ops);

create index ix_act_character on acts(person_id, character);
create index ix_rating on clip_rating(clip_id, votes, rank);

create index ix_acts_order_credit on acts(orders_credit);

create index ix_directs_role ON directs using gin (role gin_trgm_ops);
create index ix_clip_year on clip(clip_id, clip_year);

create index ix_role on directs(role);
create index ix_person_fullname on person(person_id, fullname);
create index ix_married on married_to(person_id);
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

General Comments

Work allocation
Cho : DDL commands
Poopalasingam : ER model
Reetz : revision and comments

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 2

Assumptions

- All strings are properly represented by encoding them as 'utf-8'.
- There is a pair of ClipId's (719315,719344) which are equal in all attributes. We do not treat it as duplicate because we take the ClipId's as given.
- Languages with/without brackets and languages with additional descriptions like 'version' are considered one language. This is not assumed for 'subtitles' or unfamiliar languages.
- Dates given in 'release_date' and 'biography' are assumed to have the format '01 January 1999' and converted into the oracle date format. If days or months are not given, we assume first day of the month and the first month of the year.
- References to many "clips" in actors, directors, etc. is given in the format '[info1 | info2]'. Each bracket entry is written into a separate row.
- Persons are not given with a unique identifier. Thus, we assume the FullName to be unique. Unique names are taken from the combined tables of biographies, actors, producers, writers, directors and they are associated with an integer ID ordered along the alphabetical order of names.
- Biographies provides information with string characters. We convert this information into numerical information for birth/death date (see above) and for height. The latter is given in either >># cm<< or >>#' # ½"<< which are converted into "cm" units.
- Spouses are given with names (enclosed by ' ') and additional information. The names are added to the person relation.

Data Loading

We ended up using PostgreSQL as DBMS. The path went from Oracle (too slow) to SQLite (types are not enforced[1]) to PostgreSQL.

The data has been cleaned and sorted using the python library "pandas". The data is exported to CSV and imported into the DB directly using the "COPY FROM" command. This has shown to have the best performance.

Query Implementation

Query a:

Description of logic:

Print the name and length of the 10 longest clips that were released in France.

[1] https://www.sqlite.org/datatype3.html

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SQL statement

```
SELECT
  C.CLIP_ID,
  C.CLIP_TITLE,
  max(R.RUNNING_TIME) AS runtime
FROM CLIP c
  JOIN RUNS R ON c.CLIP_ID = R.CLIP_ID
  JOIN COUNTRY C2 ON r.COUNTRY_ID = C2.COUNTRY_ID
WHERE C2.COUNTRYNAME = 'France'
GROUP BY c.CLIP_ID, c.CLIP_TITLE
ORDER BY runtime DESC, c.CLIP_ID, C.CLIP_TITLE
FETCH FIRST 10 ROWS ONLY;
```

First 5 results

	clip_id	clip_title	runtime
1	1771887	Éveil	1967
2	2153849	Cinéastes de notre temps {La nouvelle vague par elle-même}	1964
3	1151471	Pop up	1230
4	149351	Bad Lieutenant: Ed Pressman Interview	630
5	2144795	Chroniques de l'Afrique sauvage	624

Query b:

Description of logic:

Compute the number of clips released per country in 2001

SQL statement

```
SELECT
  c2.COUNTRYNAME,
  count(*) AS nb_of_clips
FROM CLIP c
  JOIN RELEASED R ON c.CLIP_ID = R.CLIP_ID
  JOIN COUNTRY C2 ON R.COUNTRY_ID = C2.COUNTRY_ID
WHERE extract(YEAR FROM r.RELEASE_DATE) = 2001
GROUP BY c2.COUNTRYNAME
ORDER BY c2.COUNTRYNAME;
```

First 5 results

	countryname	nb_of_clips
1	Afghanistan	1
2	Angola	1
3	Argentina	887
4	Armenia	2
5	Australia	1903

Query c:

Description of logic:

Compute the numbers of clips per genre released in the USA after 2013.

SQL statement

```
SELECT
  G.GENRE,
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
  count(*) AS nb_of_clips
FROM CLIP c
  JOIN CLIP_GENRE CG ON CG.CLIP_ID = C.CLIP_ID
  JOIN RELEASED R ON c.CLIP_ID = R.CLIP_ID
  JOIN COUNTRY C2 ON R.COUNTRY_ID = C2.COUNTRY_ID
  JOIN GENRE G ON CG.GENRE_ID = G.GENRE_ID
WHERE extract(YEAR FROM r.RELEASE_DATE) > 2013
      AND C2.COUNTRYNAME = 'USA'
GROUP BY G.GENRE
ORDER BY g.GENRE;
```

First 5 results

	genre	nb_of_clips
1	Action	14
2	Adventure	10
3	Animation	1
4	Biography	3
5	Comedy	13

Query d:

Description of logic:

Print the name of actor/actress who has acted in more clips than anyone else

SQL statement

```
SELECT
  d.FULLNAME
FROM (
  SELECT
     P.FULLNAME,
     count(*) AS nb_acts
   FROM PERSON P
     JOIN ACTS A2 ON P.PERSON_ID = A2.PERSON_ID
   GROUP BY P.person_id
   ORDER BY nb_acts DESC
   FETCH FIRST 1 ROW ONLY
) d;
```

First 5 results

	fullname
1	Vorderman Carol

Query e:

Description of logic:

Print the maximum number of clips any director has directed.

SQL statement

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

```
SELECT
  P.FULLNAME,
  count(*) AS nb_acts
FROM PERSON P
  JOIN DIRECTS D2 ON P.PERSON_ID = D2.PERSON_ID
  JOIN CLIP C2 ON D2.CLIP_ID = C2.CLIP_ID
GROUP BY P.FULLNAME
ORDER BY nb_acts DESC
FETCH FIRST 1 ROWS ONLY;
```

First 5 results

	fullname	nb_acts
1	Brown Ellen (I)	2252

Query f:

Description of logic:

Print the names of people that had at least 2 different jobs in a single clip. For example, if X has both acted, directed and written movie Y, his/her name should be printed out. On the other hand, if X has acted as 4 different personas in the same clip, but done nothing else, he/she should not be printed.

SQL statement

```
SELECT
  distinct d.FULLNAME
FROM (
   SELECT
     c.CLIP_ID,
     P.PERSON_ID,
     p.fullname,
     count(a.CLIP_ID) AS acts,
     count(d.CLIP_ID) AS directs,
     count(w.CLIP_ID) AS writes,
     count(pr.CLIP_ID) as produces
   FROM CLIP C, PERSON P
     LEFT JOIN ACTS a ON a.PERSON_ID = p.PERSON_ID
     LEFT JOIN DIRECTS d ON d.PERSON_ID = p.PERSON_ID
     LEFT JOIN WRITES w ON w.PERSON_ID = p.PERSON_ID
     LEFT JOIN produces pr ON pr.person_id = P.person_id
   WHERE
     a.CLIP_ID = c.CLIP_ID AND
     d.CLIP_ID = c.CLIP_ID AND
     w.CLIP_ID = c.CLIP_ID AND
     pr.clip_id = c.clip_id
   GROUP BY C.CLIP_ID, P.PERSON_ID
   HAVING ((count(a.CLIP_ID) > 0 AND count(d.CLIP_ID) > 0) OR
          (count(d.CLIP_ID) > 0 AND count(w.CLIP_ID) > 0) OR
          (count(a.CLIP_ID) > 0 AND count(w.CLIP_ID) > 0) OR
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
          (count(a.clip_id) > 0 and count(pr.clip_id) > 0) OR
          (count(d.clip_id) > 0 and count(pr.clip_id) > 0) OR
          (count(w.clip_id) > 0 and count(pr.clip_id) > 0)
 )
 )d
order by d.fullname;
```

First 5 results

	fullname
1	Aalam Steve
2	Aames Willie
3	Aanenson Quentin
4	Aaron Aash
5	Aaron II Tony

Query g:

Description of logic:

Print the 10 most common clip languages

SQL statement

```
SELECT
  L.LANGUAGE
FROM CLIP_LANGUAGE
  JOIN LANGUAGE L ON CLIP_LANGUAGE.LANGUAGE_ID = L.LANGUAGE_ID
GROUP BY L.LANGUAGE
ORDER BY count(*) DESC
FETCH FIRST 10 ROWS ONLY;
```

First 5 results

	language
1	english
2	spanish
3	german
4	french
5	japanese

Query h:

Description of logic:

Print the full name of the actor who has performed in the highest number of clips with a user-specified type.

SQL statement

```
with person_act_count as (
 SELECT
   a.PERSON_ID,
   count(*) AS count
 FROM ACTS A
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

```
   JOIN CLIP C2 ON A.CLIP_ID = C2.CLIP_ID
 WHERE C2.CLIP_TYPE = :clip_type
 GROUP BY a.PERSON_ID
)
SELECT
  p.FULLNAME
FROM person_act_count b
  JOIN PERSON p ON p.PERSON_ID = b.PERSON_ID
ORDER BY b.count DESC
FETCH FIRST 1 ROWS ONLY;
```

First 5 results

	fullname
1	Ozokwor Patience

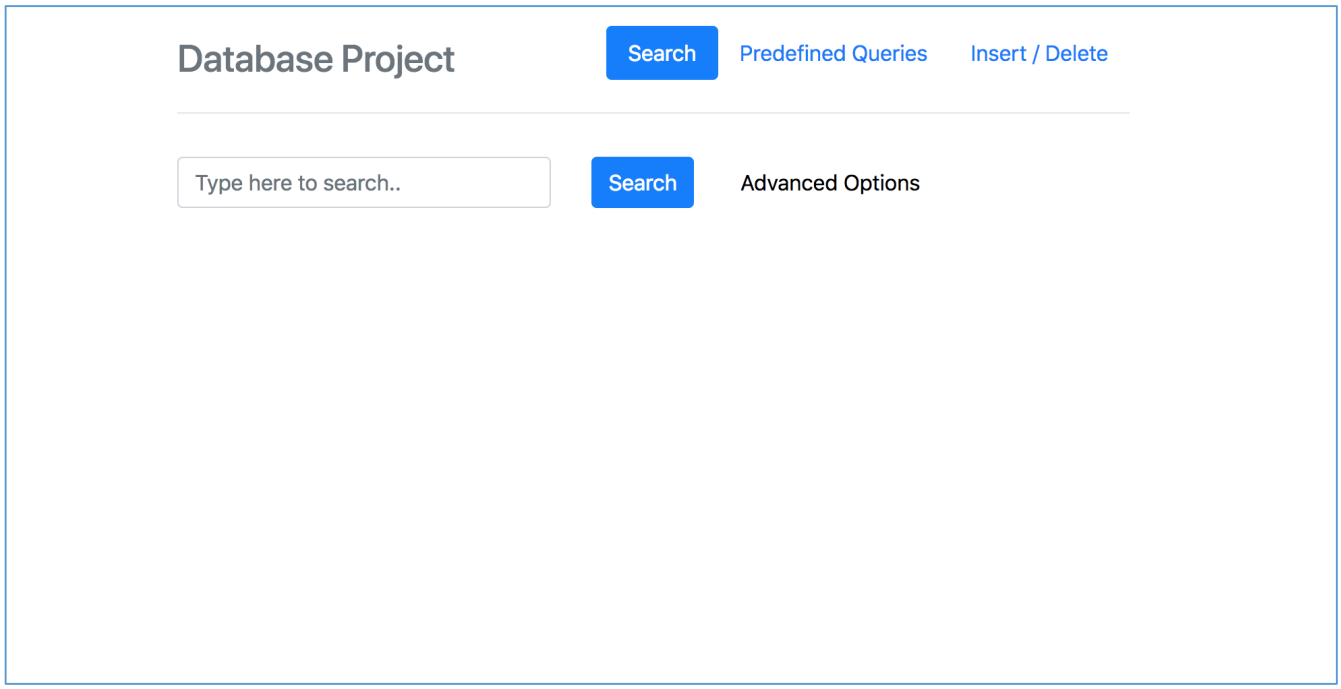
Interface

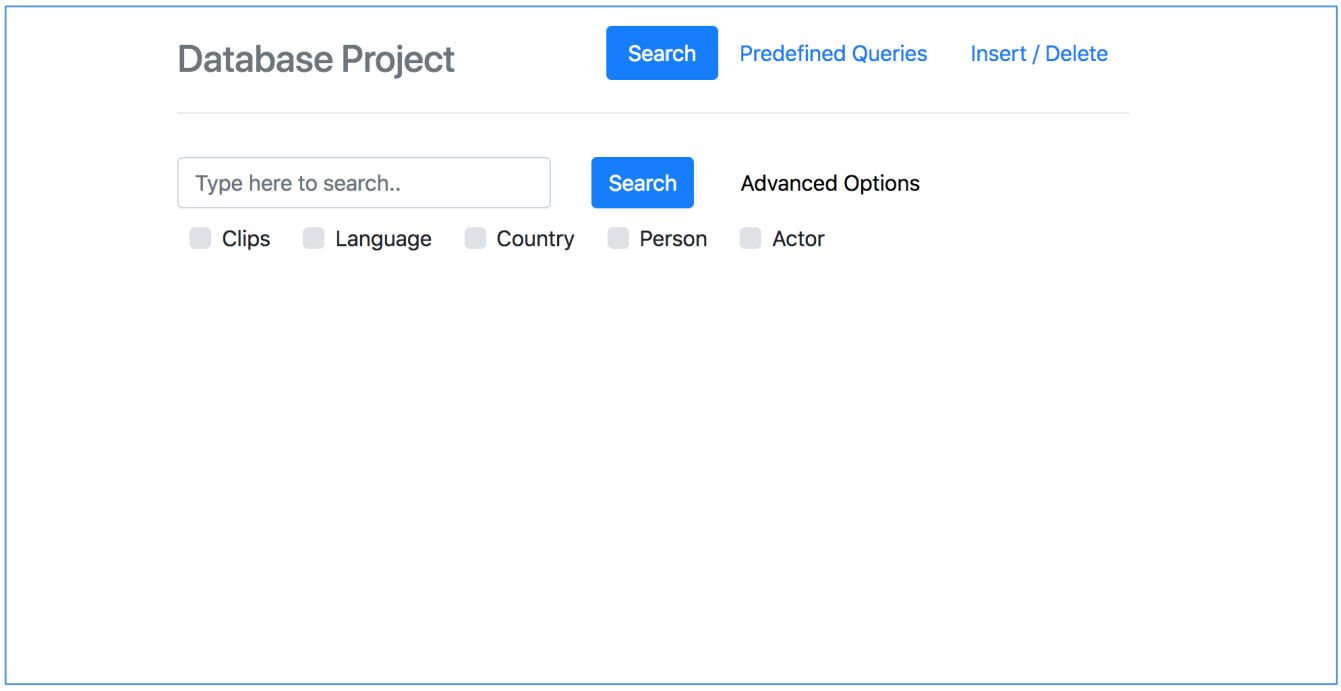
Design logic Description

The user interface is a simple web page, where user can search for relevant entities, and see the result. The backend is developed in Python with Flask library. The web page is based on Bootstrap 4 CSS Framework.

Screenshots

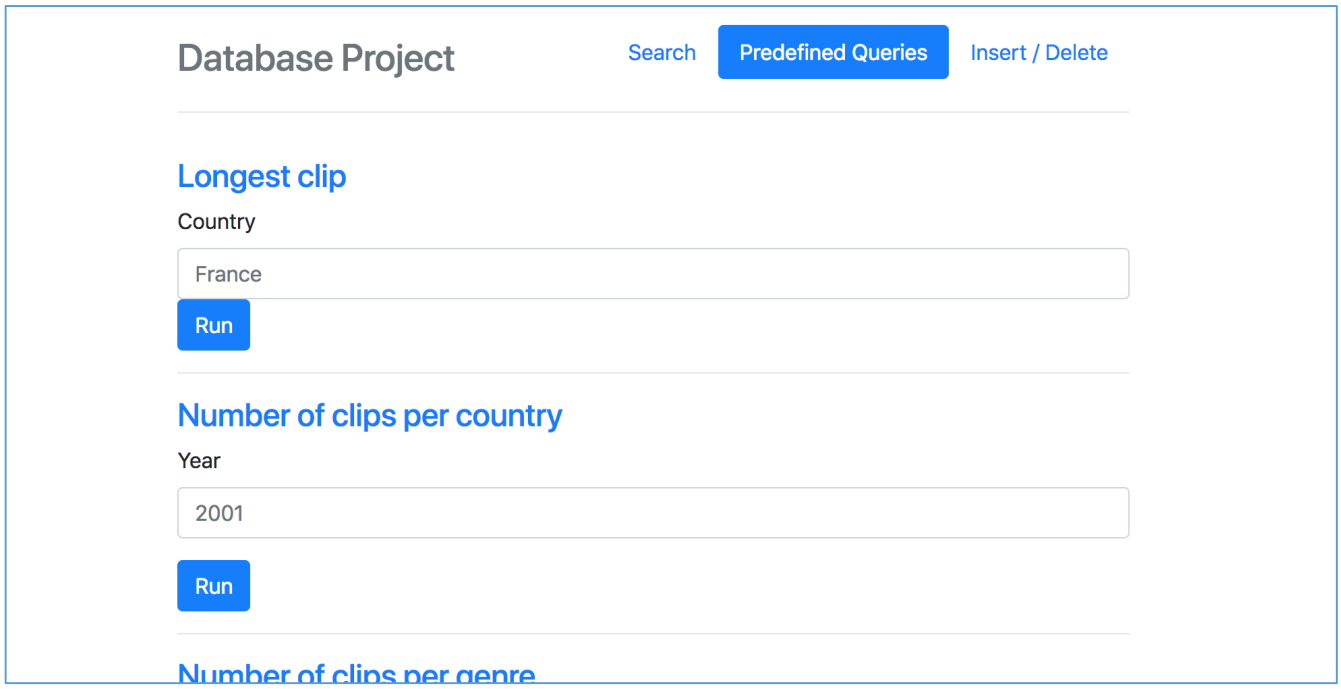
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

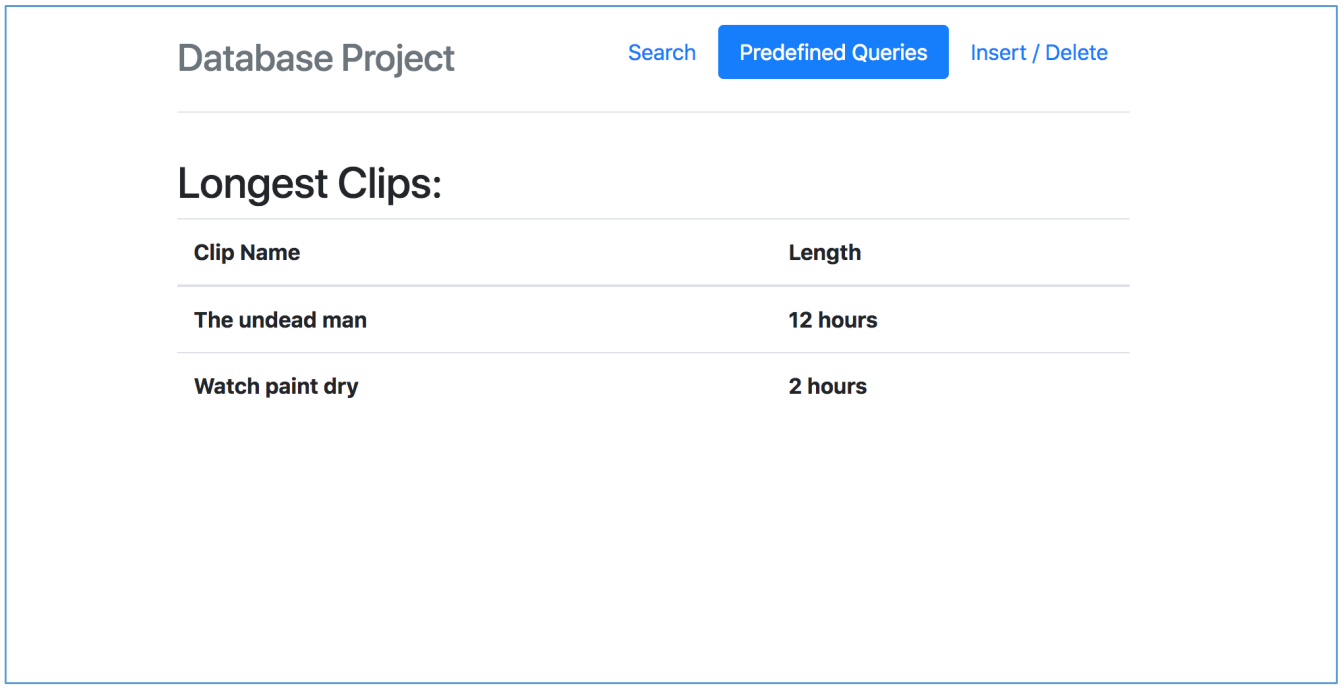




DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

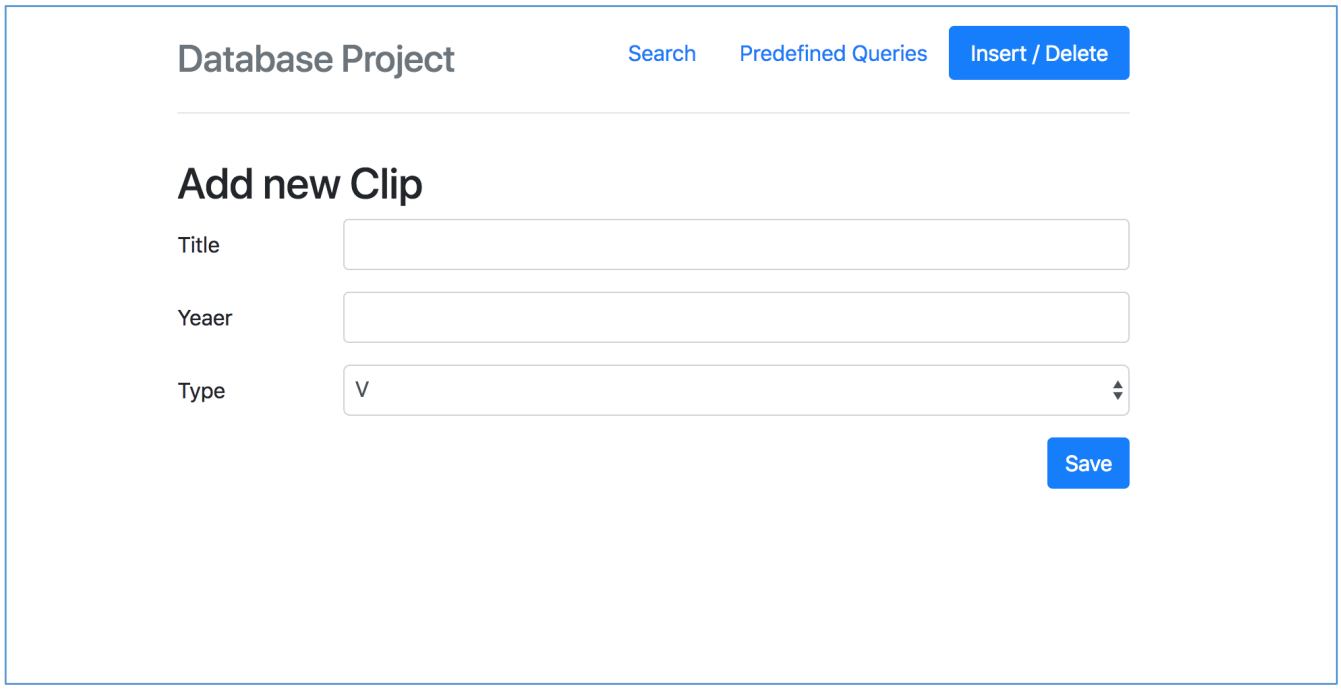
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE





DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 3

Assumptions

- For spouses under biographies, we created a relation from biography to person in a "married_to" relation. The married_to table has biography_id, person_id, date (of marriage), marital status and children (number). We assume that spouses with no marital status stated are married or unknown, so we mark them as "married/unknown". When there is no children number stated, we assume that it is "0/unknown".
- For languages, we lowercase all languages and remove all parentheses. In addition, we assume that "English", "English version", and "English version only" are all clips in English so we clean the data to remove "version" and "only", for all applicable languages
- Missing characters and roles have been replaced by "NA"
- Made person.fullname unique in the database, since we use it that way in pandas imports.
- Missing spouse information is considered as unmarried

Query Implementation

Query a:

Description of logic:

Print the names of the top 10 actors ranked by the average rating of their 3 highest-rated clips that where voted by at least 100 people. The actors must have had a role in at least 5 clips (not necessarily rated).

SQL statement

```
with actors_with_more_than_five_roles as (
    SELECT
      person_id as person_id,
      count(character) as number_of_clips_he_played
    FROM acts
    GROUP BY person_id
    HAVING count(character) >= 5
),
actor_with_avg_rating as (
  SELECT
  actor.*,
  (
    -- gives the average of the top rated clips
    SELECT
      avg(f.rank)
    FROM (
      -- find 3 highest-rated clips
      SELECT
        r.rank
      FROM clip_rating r
      JOIN acts a ON a.clip_id = r.clip_id
      WHERE votes >= 100 AND a.person_id = actor.person_id AND r.rank IS NOT NULL
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

```
      ORDER BY r.rank DESC
      FETCH FIRST 3 ROWS ONLY
    )f
  ) as avg_rating
  FROM actors_with_more_than_five_roles actor
)
select
  p.fullname,
  b.number_of_clips_he_played,
  b.avg_rating
FROM actor_with_avg_rating b
  join person p on p.person_id = b.person_id
  where b.avg_rating is not NULL
order by b.avg_rating DESC
fetch first 10 rows ONLY;
```

First 5 results

10 rows

	fullname	number_of_clips_he_played	avg_rating
1	Meeks Johnny Ray	13	10
2	Bowden Blesst	5	10
3	Culina Arijana	60	10
4	Hillis Ali	45	10
5	Jenkins Ken (I)	324	10

Query b:

Description of logic:

Compute the average rating of the top-100 rated clips per decade in decreasing order.

SQL statement

```
select
  b.decade,
  (
    select
      avg(d.rank)
    FROM (
      SELECT cr.rank
      FROM clip_rating cr
      JOIN clip c ON c.clip_id = cr.clip_id
      WHERE extract(DECADE FROM c.clip_year) * 10 = b.decade
      ORDER BY cr.rank DESC
      FETCH FIRST 100 ROWS ONLY
    ) d
  ) as avg_rating
from (
    SELECT
      DISTINCT extract(DECADE FROM clip_year) * 10 as decade
    FROM clip
) b
where b.decade is not null
order by b.decade desc;
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

First 5 results

	decade	avg_rating
1	2010	10
2	2000	10
3	1990	10
4	1980	10
5	1970	10

Query c:

Description of logic:

For any video game director, print the first year he/she directed a game, his/her name and all his/her game titles from that year.

SQL statement

```
with director_first_year_he_directed as (
  select
    d.person_id as person_id,
    min(extract(year from clip_year)) as first_year
  from directs d
    join clip c on c.clip_id = d.clip_id
  where d.role like '%game%'
  group by d.person_id
)
select
  distinct p.fullname,
  df.first_year,
  c.clip_title
from director_first_year_he_directed df
  join directs d on d.person_id = df.person_id
  join person p on p.person_id = d.person_id
  join clip c on c.clip_id = d.clip_id
where extract(year from c.clip_year ) = df.first_year;
```

First 5 results

	fullname	first_year	clip_title
1	Asmussen Stig	2010	God of War III
2	Chadani Osamu	2009	0 Day: Attack on Earth
3	Hoshino Jin	2005	Raiden III
4	Inamoto Noboru	2009	0 Day: Attack on Earth
5	Itô Hiroyuki	2006	Fainaru fantajî XII

Query d:

Description of logic:

For each year, print the title, year and rank-in-year of top 3 clips, based on their ranking.

SQL statement

```
WITH clip_year_rank_in_year AS (
  select
    C.CLIP_YEAR,
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
    C.clip_title,
    CR.RANK,
    ROW_NUMBER() OVER(PARTITION BY C.CLIP_YEAR ORDER BY CR.RANK DESC, CR.VOTES DESC) AS
rowind
  FROM CLIP C
  JOIN CLIP_RATING CR ON C.CLIP_ID = CR.CLIP_ID
)
SELECT
  extract(YEAR FROM s.CLIP_YEAR) as year,
  s.CLIP_TITLE,
  s.RANK
FROM clip_year_rank_in_year s
WHERE s.rowind <=3 and s.CLIP_YEAR IS NOT NULL
ORDER BY s.clip_year desc;
```

First 5 results

	year	clip_title	rank
1	2011	Bubblegum & Broken Fingers	10
2	2011	Dreams Awake	10
3	2011	Game of Thrones	10
4	2010	Mass Effect 2	10
5	2010	Spartacus: Blood and Sand {Kill Them All (#1.13)}	10

Query e:

Description of logic:

Print the names of all directors who have also written scripts for clips, in all of which they were additionally actors (but not necessarily directors) and every clip they directed has at least two more points in ranking than any clip they wrote.

SQL statement

```
with directed as (
  select
    distinct p.person_id,
    cr.rank
  from person p
    join directs d on p.person_id = d.person_id
    join clip c on d.clip_id = c.clip_id
    join clip_rating cr on c.clip_id = cr.clip_id
),
wrote as (
  select distinct --join to get writers with their clip_ratings
    p.person_id,
    cr.rank
from person p
  join writes w on p.person_id = w.person_id
  join clip c on w.clip_id = c.clip_id
  join clip_rating cr on c.clip_id = cr.clip_id
)
select
 aw.fullname,
 min(d.rank) as minpoints_directed,
 max(w.rank) as minpoints_written
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
from (
  select
      distinct --join to get writers who acted in their written clips
      p.person_id,
      p.fullname
  from person p
    join writes w on p.person_id = w.person_id
    join acts a on p.person_id = a.person_id
    join clip c on w.clip_id = c.clip_id and a.clip_id = c.clip_id
  ) as aw
join directed d on d.person_id=aw.person_id --final join of above tables
join wrote w on w.person_id=aw.person_id
group by aw.fullname
having( min(d.rank) >= max(w.rank) +2) --data selection criterion on ranking
order by aw.fullname
```

First 5 results

	fullname	minpoints_directed	minpoints_written
1	Abrams Abiola	8	5
2	Avella Joe (II)	8	5
3	Baring Zam	8	4
4	Beech Andrew	6	2
5	Blakeway Denys	9	7

Query f:

Description of logic:

Print the names of the actors that are not married and have participated in more than 2 clips that they both acted in and co-directed it.

SQL statement

```
with unmarried_person as (
    select
      distinct p.person_id
    from person p
      left join married_to m on m.person_id = p.person_id
    where m.person_id is null or m.marital_status not like 'married%'
),
acting_codirectors as (
  select
    p.person_id
  from person p
    join acts a on p.person_id = a.person_id
    join directs d on p.person_id = d.person_id
    join clip c on d.clip_id = c.clip_id and a.clip_id = c.clip_id
  where d.role like 'co-director%'
  group by p.person_id
  having count(p.person_id) > 2
)
select
```

```
  p.fullname
from person p
  join unmarried_person a on a.person_id = p.person_id
  join acting_codirectors d on d.person_id = p.person_id
order by p.fullname;
```

First 5 results

	fullname
1	Bagans Zak (I)
2	Barack Jesse
3	Beard Tanner
4	Bettinelli-Olpin Matt
5	Blass Jorge

Query g:

Description of logic:

Print the names of screenplay story writers who have worked with more than 2 producers.

SQL statement

```
with screenplay_writer as (
  select
    distinct w.person_id
  from writes w
 where work_type like '%screenplay story%'
)
select
  person.fullname
from clip c
  join produces p on c.clip_id = p.clip_id
  join writes w on c.clip_id = w.clip_id
  join person person on person.person_id = w.person_id
where w.person_id in ( select * from screenplay_writer )
group by w.person_id, person.fullname
having count(distinct p.person_id) > 2
```

First 5 results

	fullname
1	Alford Theron
2	Aniekwe Ikenna
3	Anozie Joy Stephen
4	Armand Deddy
5	Ayinde Abibat S.

Query h:

Description of logic:

Compute the average rating of an actor's clips (for each actor) when she/he has a leading role (first 3 credits in the clip).

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SQL statement

```
with person_leading as (
    select
      DISTINCT aa.person_id,
      fullname,
      orders_credit
    from acts aa
      join clip_rating cr on aa.clip_id = cr.clip_id
      join person p on aa.person_id = p.person_id
      where orders_credit <= 3
)
select
  b.person_id,
  b.fullname,
  (
    Select round(avg(cr.rank), 2)
    from acts a
      join clip_rating cr on a.clip_id = cr.clip_id
      where orders_credit <= 3
            AND b.person_id = a.person_id
  ) as avg_rating
from person_leading b
  order by b.person_id DESC
;
```

First 5 results

	person_id	fullname	avg_rating
1	2588641	Þórðardóttir Guðrún	6
2	2588628	Þórisdóttir Lilja	7
3	2588618	Þórhallsdóttir Hólmfríður	6
4	2588612	Þórarinsdóttir Lilja Nótt	7
5	2588599	Þorvaldsson Guðmundur Ingi	5

Query i:

Description of logic:

Compute the average rating for the clips whose genre is the most popular genre.

SQL statement

```
with most_popular_genre as (
  select
    cg.genre_id
  from clip_genre cg
  group by cg.genre_id
  order by count(*)  desc
  fetch first 1 rows only
)
select
  round(avg(cr.rank), 2)
from clip_rating  cr
  join clip_genre cg on cg.clip_id = cr.clip_id
where cg.genre_id = ( select * from most_popular_genre )
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

First 5 results

	round
1	6.5

Query j:

Description of logic:

Print the names of the actors that have participated in more than 100 clips, of which at least 60% where short but not comedies nor dramas, and have played in more comedies than double the dramas. Print also the number of comedies and dramas each of them participated in.

SQL statement

```
with actor_more_than_100_clips as (
    select
      a.person_id                 as person_id,
      count(distinct a.clip_id) as number_of_clips
    from acts a
    group by a.person_id
    having count(distinct a.clip_id) > 100
),
short_not_comedy_nor_drama as (
    -- clips which are not short but not comedies nor dramas
      select c.clip_id
      from clip c
        join clip_genre cg on c.clip_id = cg.clip_id
        join genre g on cg.genre_id = g.genre_id
      group by c.clip_id
      -- is short but not comedy nor drama
      having array_agg(g.genre) && ARRAY ['Short' :: varchar]
             and not (array_agg(g.genre) && ARRAY ['Comedy' :: varchar])
             and not (array_agg(g.genre) && ARRAY ['Drama' :: varchar])
),
actor_of_interest as (
  -- select actors, where 60% where short but not comedy nor drama
  select
    a.person_id,
    aiq.number_of_clips,
    count(distinct a.clip_id) as short_not_comedies
  from acts a
    join actor_more_than_100_clips aiq on aiq.person_id = a.person_id
    join short_not_comedy_nor_drama s on a.clip_id = s.clip_id
    join person p on a.person_id = p.person_id
  group by a.person_id, aiq.number_of_clips
  having count(distinct a.clip_id) >= 0.6 * aiq.number_of_clips
),
actor_with_nr_comedy_drama as (
  select
```

```
    p.fullname,
    (
      select count(*)
      from acts a
        join clip c on a.clip_id = c.clip_id
        join clip_genre cg on c.clip_id = cg.clip_id
        join genre g on cg.genre_id = g.genre_id
      where g.genre = 'Drama' and a.person_id = p.person_id
    ) as nr_drama,
    (
      select count(*)
      from acts a
        join clip c on a.clip_id = c.clip_id
        join clip_genre cg on c.clip_id = cg.clip_id
        join genre g on cg.genre_id = g.genre_id
      where g.genre = 'Comedy' and a.person_id = p.person_id
    ) as nr_comedy
  from actor_of_interest aoi
    join person p on p.person_id = aoi.person_id
)
select
  *
from actor_with_nr_comedy_drama a
where a.nr_comedy > 2*a.nr_drama;
```

First 5 results

	fullname	nr_drama	nr_comedy
1	Allen Dayton	1	10
2	FitzPatrick James A.	0	1
3	Smith Pete (I)	9	102

Query k:

Description of logic:

Print the number of Dutch movies whose genre is the second most popular one.

SQL statement

```
Select
  cg.genre_id,
  count(cg.genre_id)
from clip_country cc
    join country c on cc.country_id = c.country_id
    join clip_genre cg on cc.clip_id = cg.clip_id
where c.countryname = 'Netherlands'
GROUP BY cg.genre_id
order by count(cg.genre_id) desc
offset 1
fetch first 1 rows only;
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

First 5 results

	genre_id	count
1	1	1449

Query l:

Description of logic:

Print the name of the producer whose role is coord

SQL statement

```
select
  distinct p.fullname
From produces
join person p on produces.person_id = p.person_id
where role like 'coordinating producer: %'
order by p.fullname;
```

First 5 results

	fullname
1	Austin Ray (I)
2	Bertelli Gian Carlo
3	Boucher Ping

Query Analysis

Selected Queries (and why)

Query a

Initial Running time: >1h

Optimized Running time: **44s 661ms** (execution: 41s 546ms, fetching: 115ms)

Explain the improvement:
Added the following two index in order to avoid the full table scan:

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
create index ix_act_character on acts(person_id, character);
create index ix_rating on clip_rating(clip_id, votes, rank);
```

The first index improves the select on “actors_with_more_than_five_roles” while the second one improves the subselect performance.

Initial plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		10	7.17849039E8	7.17849039E8
Transformation (Limit)		10	7.17849039E8	7.17849039E8
Union (Aggregate)		65748	1093023.0	1031102.0
Unknown (Gather Merge)		131496	1091380.0	1031102.0
Union (Aggregate)		65748	1075202.0	1030102.0
Sort (Sort)		5925565	1044916.0	1030102.0
Full scan (Seq Scan)	table: acts;	5925565	160982.0	0.0
Access (CTE Scan)		65748	7.16753287E8	0.0
Union (Aggregate)		1	10901.0	10901.0
Transformation (Limit)		3	10901.0	10901.0
Sort (Sort)		8	10901.0	10901.0
Hash join (Hash Join)		8	10901.0	9815.0
Index scan (Index Only Scan)	table: acts; index: acts_pkey;	216	741.0	0.0
Transformation (Hash)		69963	8598.0	8598.0
Full scan (Seq Scan)	table: clip_rating;	69963	8598.0	0.0
Sort (Sort)		65419	2892.0	2728.0
Access (CTE Scan)		65419	1314.0	0.0

Improved plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		10	1.35977997E8	1.35977997E8
Transformation (Limit)		10	1.35977997E8	1.35977997E8
Union (Aggregate)		65748	780957.0	1000.0
Unknown (Gather Merge)		131496	779313.0	1000.0
Union (Aggregate)		65748	763135.0	0.0
Index scan (Index Only Scan)	table: acts; index: ix_act_character;	5925565	732850.0	0.0
Access (CTE Scan)		65748	1.3519431E8	0.0
Union (Aggregate)		1	2056.0	2056.0
Transformation (Limit)		3	2056.0	2056.0
Sort (Sort)		8	2056.0	2056.0
Nested loops (Nested Loop)		8	2056.0	0.0
Index scan (Index Scan)	table: acts; index: ix_act_character;	216	729.0	0.0
Index scan (Index Only Scan)	table: clip_rating; index: ix_rating;	1	6.0	0.0
Sort (Sort)		65419	2892.0	2728.0
Access (CTE Scan)		65419	1314.0	0.0

Query c

Initial Running time: 1m 41s 65 ms (execution: 1 m 41s 53 ms, fetching: 12 ms)

Optimized Running time: in 82 ms (execution: 72 ms, fetching: 10 ms)

Explain the improvement:

Created the following index:

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
CREATE INDEX ix_directs_role ON directs using gin (role gin_trgm_ops);
create index ix_clip_year on clip(clip_id, clip_year);
```

The first one improves the "%role%" search query and the second one is used for the year selection.

Initial plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		21	17758.0	17757.0
Unique (Unique)		21	17758.0	17757.0
Union (Aggregate)		127	14261.0	14246.0
Unknown (Gather Merge)		106	14259.0	14246.0
Union (Aggregate)		53	13247.0	13246.0
Sort (Sort)		53	13246.0	13246.0
Nested loops (Nested Loop)		53	13244.0	0.0
Full scan (Seq Scan)	table: directs;	53	12798.0	0.0
Index scan (Index Scan)	table: clip; index: clip_pkey;	1	8.0	0.0
Sort (Sort)		21	3496.0	3496.0
Nested loops (Nested Loop)		21	3495.0	1.0
Nested loops (Nested Loop)		4214	1258.0	0.0
Nested loops (Nested Loop)		127	1075.0	0.0
Access (CTE Scan)		127	2.0	0.0
Index scan (Index Scan)	table: person; index: person_pkey;	1	8.0	0.0
Index scan (Index Only Scan)	table: directs; index: directs_pkey;	33	1.0	0.0
Index scan (Index Scan)	table: clip; index: clip_pkey;	1	0.0	0.0

Improved plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		21	5046.0	5046.0
Unique (Unique)		21	5046.0	5046.0
Union (Aggregate)		127	1550.0	1547.0
Sort (Sort)		127	1547.0	1547.0
Nested loops (Nested Loop)		127	1542.0	21.0
Bitmap index scan (Bitmap Heap Scan)	table: directs;	127	474.0	20.0
Bitmap index scan (Bitmap Index Scan)	index: ix_directs_role;	127	20.0	0.0
Index scan (Index Only Scan)	table: clip; index: ix_clip_year;	1	8.0	0.0
Sort (Sort)		21	3496.0	3496.0
Nested loops (Nested Loop)		21	3495.0	1.0
Nested loops (Nested Loop)		4214	1258.0	0.0
Nested loops (Nested Loop)		127	1075.0	0.0
Access (CTE Scan)		127	2.0	0.0
Index scan (Index Scan)	table: person; index: person_pkey;	1	8.0	0.0
Index scan (Index Only Scan)	table: directs; index: directs_pkey;	33	1.0	0.0
Index scan (Index Scan)	table: clip; index: ix_clip_year;	1	0.0	0.0

Query f

Initial Running time: 1s 779ms (execution: 1s 767ms, fetching: 12ms)

Optimized Running time: 621ms (execution: 613ms, fetching: 8ms)

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Explain the improvement:

```
create index ix_role on directs(role);
create index ix_person_fullname on person(person_id, fullname);
create index ix_married on married_to(person_id);
```

Initial plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		12544	231663.0	231631.0
Sort (Sort)		12544	231663.0	231631.0
Merge join (Merge Join)		2508838	142742.0	10654.0
Index scan (Index Only Scan)	table: person; index: person_pkey;	2588664	124086.0	0.0
Sort (Sort)		102098	10908.0	10653.0
Full scan (Seq Scan)	table: married_to;	102098	2158.0	0.0
Union (Aggregate)		1	28441.0	28441.0
Sort (Sort)		1	28441.0	28441.0
Nested loops (Nested Loop)		1	28441.0	1193.0
Unknown (Gather)		1	28441.0	1192.0
Nested loops (Nested Loop)		1	27441.0	192.0
Nested loops (Nested Loop)		2	27440.0	192.0
Bitmap index scan (Bitmap Heap Scan)	table: directs;	4512	7116.0	191.0
Bitmap index scan (Bitmap Index Scan)	index: ix_directs_role;	10830	189.0	0.0
Index scan (Index Only Scan)	table: acts; index: acts_pkey;	1	4.0	0.0
Index scan (Index Only Scan)	table: person; index: person_pkey;	1	0.0	0.0
Index scan (Index Only Scan)	table: clip; index: ix_clip_year;	1	0.0	0.0
Hash join (Hash Join)		12544	59593.0	8.0
Access (CTE Scan)		2508838	50176.0	0.0
Transformation (Hash)		1	8.0	8.0
Nested loops (Nested Loop)		1	8.0	0.0
Access (CTE Scan)		1	0.0	0.0
Index scan (Index Scan)	table: person; index: person_pkey;	1	8.0	0.0

Improved plan

Operation	Params	Rows	Total Cost	Startup Cost
Select		12544	192917.0	192886.0
Sort (Sort)		12544	192917.0	192886.0
Merge join (Merge Join)		2508838	104001.0	6.0
Index scan (Index Only Scan)	table: person; index: ix_person_fullname;	2588664	89046.0	0.0
Index scan (Index Only Scan)	table: married_to; index: ix_married;	102098	7207.0	0.0
Union (Aggregate)		1	28441.0	28441.0
Sort (Sort)		1	28441.0	28441.0
Nested loops (Nested Loop)		1	28441.0	1193.0
Unknown (Gather)		1	28441.0	1192.0
Nested loops (Nested Loop)		1	27441.0	192.0
Nested loops (Nested Loop)		2	27440.0	192.0
Bitmap index scan (Bitmap Heap Scan)	table: directs;	4512	7116.0	191.0
Bitmap index scan (Bitmap Index Scan)	index: ix_directs_role;	10830	189.0	0.0
Index scan (Index Only Scan)	table: acts; index: acts_pkey;	1	4.0	0.0
Index scan (Index Only Scan)	table: person; index: ix_person_fullname;	1	0.0	0.0
Index scan (Index Only Scan)	table: clip; index: ix_clip_year;	1	0.0	0.0
Hash join (Hash Join)		12544	59589.0	4.0
Access (CTE Scan)		2508838	50176.0	0.0
Transformation (Hash)		1	4.0	4.0
Nested loops (Nested Loop)		1	4.0	0.0
Access (CTE Scan)		1	0.0	0.0
Index scan (Index Only Scan)	table: person; index: ix_person_fullname;	1	4.0	0.0

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Search Query User Interface

The following type of query is used in the user interface to search for person:

```
select
  person.fullname
from acts, person
where acts.person_id = person.person_id
      AND (person.fullname ilike :search)
limit 1
```

Initial Running time:

- For input "Brad Pitt": 2s 528ms
- For input "Kidman": 1s 318ms
- For input "mike": 74ms

Initial plan:

Operation	Params	Rows	Total Cost	Startup Cost
Select		1	329.0	0.0
Transformation (Limit)		1	329.0	0.0
Nested loops (Nested Loop)		420	138298.0	0.0
Full scan (Seq Scan)	table: person;	259	82082.0	0.0
Index scan (Index Only Scan)	table: acts; index: acts_pkey;	64	216.0	0.0

The following index has been added in order to improve the performance:

```
CREATE INDEX ix_person_name ON person using gin (fullname gin_trgm_ops);
```

This adds a PostgreSQL GIN (Generalized Inverted Index) trigram index, which creates a trigram data structure which is used for the "like" search.

Optimized Running time:

- For input "Brad Pitt": 94ms
- For input "Kidman": 71ms
- For input "mike": 86ms

Optimized plan:

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Select		1	166.0	30.0
Transformation (Limit)		1	166.0	30.0
Nested loops (Nested Loop)		420	57229.0	30.0
Bitmap index scan (Bitmap Heap Scan)	table: person;	259	1013.0	30.0
Bitmap index scan (Bitmap Index Scan)	index: ix_person_name;	259	29.0	0.0
Index scan (Index Only Scan)	table: acts; index: acts_pkey;	64	216.0	0.0

The above optimization has also been made for the following tables:

- Clip
  ```
  CREATE INDEX ix_clip_title ON clip using gin (clip_title gin_trgm_ops);
  ```
- Language
  ```
  CREATE INDEX ix_language ON language using gin (language gin_trgm_ops);
  ```
- Country
  ```
  CREATE INDEX ix_country ON country using gin (countryname gin_trgm_ops);
  ```

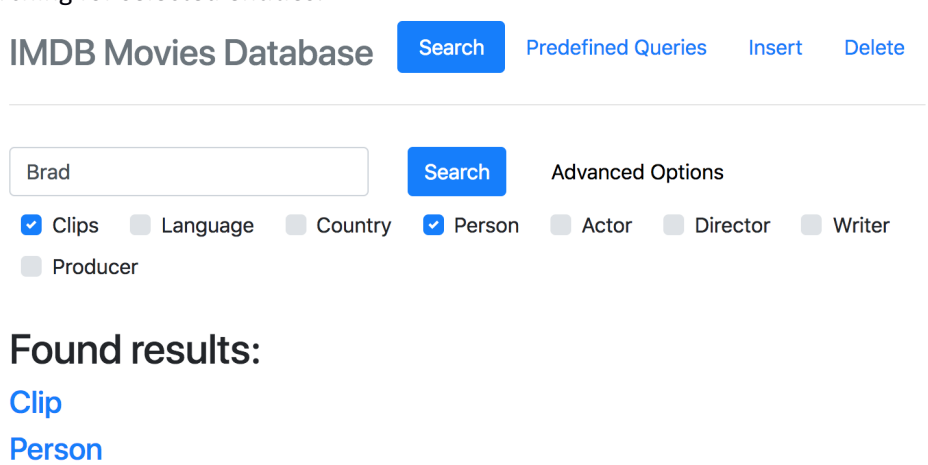
Interface

Design logic Description

For the insert and deletes, only the simple cases have been added.

Screenshots

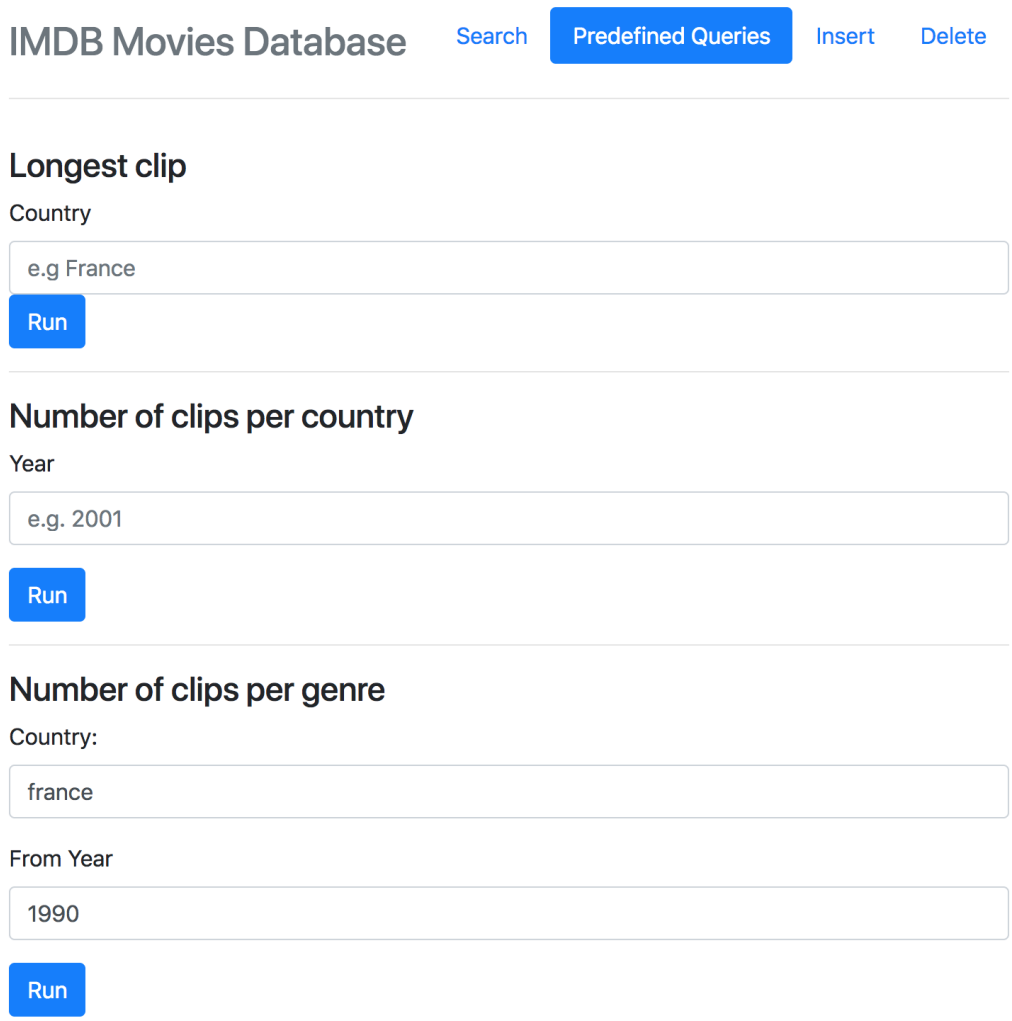
Searching for selected entities:



DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

User can search for certain predefined queries:



With results showing as:

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

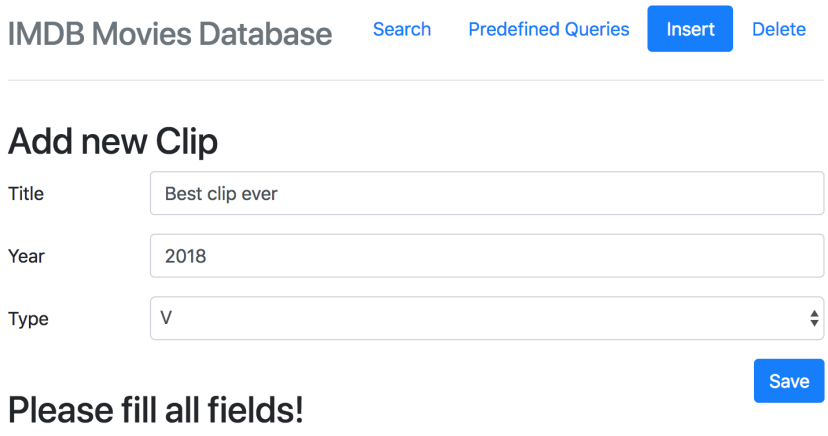
IMDB Movies Database Search Predefined Queries Insert Delete

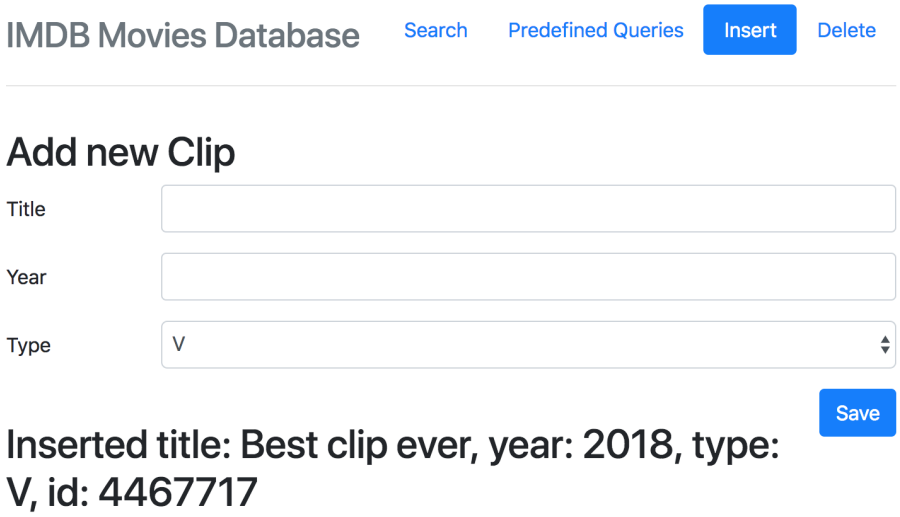
Number of clips per genre

Genre	Number of clips
Action	2169
Adventure	1530
Animation	1548
Biography	642
Comedy	6606
Crime	2343
Documentary	4499
Drama	12390
Family	1210
Fantasy	1367
Film-Noir	10

Add new clips:

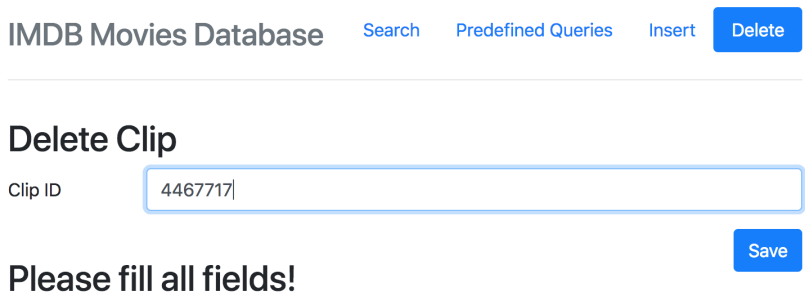
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/



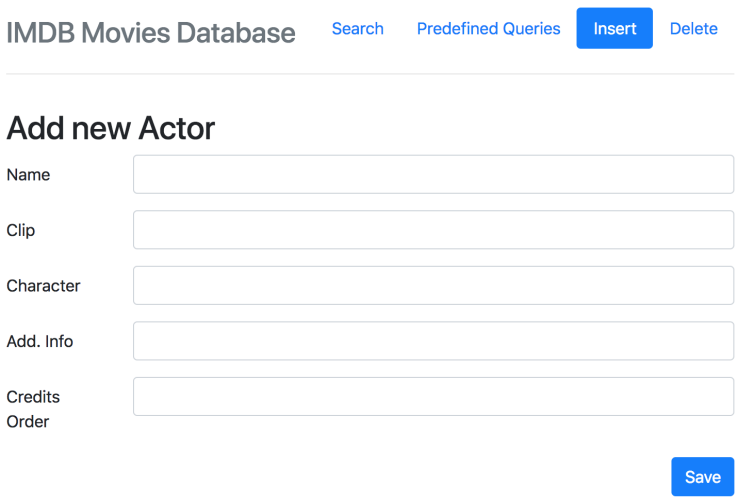


And delete existing clips:

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/



Similarly, adding new actors:



General Comments

Work allocation
Cho : User interface / Query
Poopalasingam : User interface / Query
Reetz : Imports / Query