Visual Novel Collection

Xin Guan

Northeastern University, Boston, MA, USA

Abstract

Visual novel, originated from RPG (role-playing game), is a distinct way of telling stories. Most of the visual novels depend on voluntary translators to produce releases of different languages. In order to help gather and publish the releases visual novels, this project is going to do a wiki-like database to store information on visual novels and its releases in multi-language. All source code is accessible at https://github.com/imbaguanxin/CS3200/tree/master/vndb_project

Introduction

A visual novel is an interactive literary genre, originated from JRPG (Japanese role-playing game) [1]. Visual novels have been popular within the Japanese style game and animation community and make up nearly 70% of the PC game titles released in 2006 [2].

Since this style of literature is originated from Japan, most of the contents are in and only in Japanese. Fans around the world speaking various languages need translators to cross the barrier of language. Most of the translation work is done by voluntary translators and programmers. Therefore, these voluntary community members tend to distribute their releases of translation on their personal websites. As a result, it is difficult for users to find translations releases and may lead to reparative translation work.

Visual Novel Collection is aimed at being a comprehensive database for all information related to visual novels. This project tends to be the underlying database of a wiki-like website as well as a platform for translations to release. This database includes the information of visual novels, releases in multi-language, producer company, art and design staff, and characters appear in the novels. Additionally, for a better search result, this project tags the visual novels and the characters.

Database Design

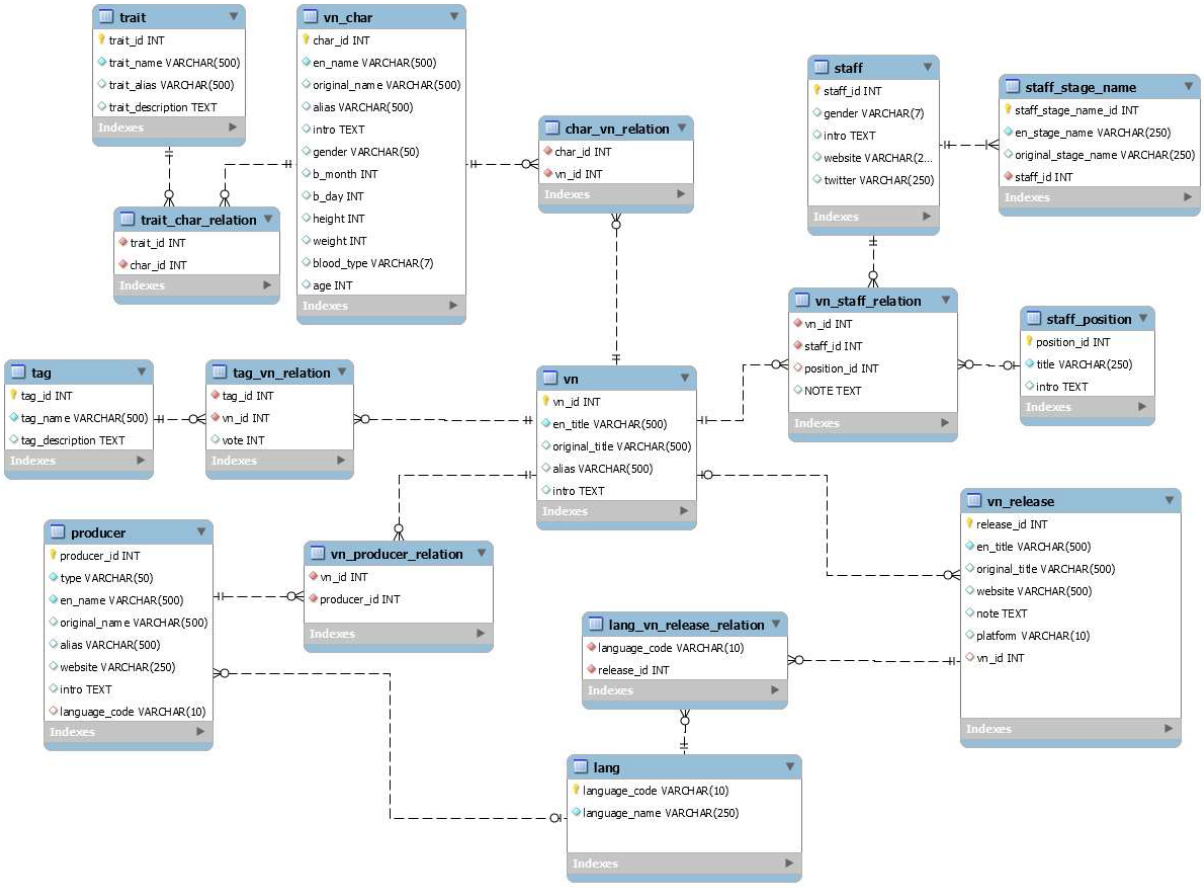


Figure 1 EER diagram of visual novel collection database

There are five key parts in this model: visual novel (*vn*), release (*vn_release*), original producer (*producer*), staff that create the visual novels (*staff*) and character appearing in the novels (*vn_char*). In this model, except the releases, which is an n:1 relation to visual novel, all other parts are related to visual novels in an n:m relation.

1. Visual novel:

Visual novel (*vn*) is the core of this database. This table has a "*vn_id*" as primary a key and an English title ("*en_title*") as a not-null attribute. Since a large proportion of visual novels are not originally in English, this database stores its original title in "*original_title*" column. Additionally, many of the visual novels have abbreviations or a shortened name within the fan community; the nullable "*alias*" stores this information. The last column is a wiki-like introduction ("*intro*").

For better search result, there is a separate *tag* table for visual novels. Like tags on twitter, there is an n:m relation between tags and visual novels with a vote number. This project collected the vote number by counting some visual novel forums' user's tags (anidb.org).

2. Release:

Releases are the translated versions published by voluntary translators. The primary key "*release_id*" identifies the releases uniquely. Since many the translations are not a complete version, the database needs to store the name of the translation in column "*en_title*". Like visual novel, the original title in the language the release use is also stored in attribute "*original_title*". "*website*" is the original site that publish the releases or the download website. "*note*" provides some flexibility on introducing the release. "*platform*" is the release platform, including windows, mac, Linux, play station, etc. The foreign key "*vn_id*" links to the visual novel table (*vn*). To deal with the multilanguage problem mentioned in the introduction, there is also an n:m relation between releases and language where the table *lang* uses an ISO639-1 standard [3].

3. producer:

The producer table is the original producer of a visual novel, including the producer and distributor. Each producer has a unique "*producer_id*" as a primary key. Since game makers are independent game designers or groups that are not companies, there is a "*type*" attribute to show the category of the producer. Similar to table *vn*, there is a non-null "*en_name*" attribute and nullable columns: "*original_name*" and "*alias*". Since still many of the producers are companies, they have their official website. Therefore, "*website*" attribute is the official site of the producer. An "*intro*" is the introduction of the corresponding producer. Lastly, the major language that the producer uses to create visual novels is listed as an attribute: "*language_code*", which is a foreign that links to the *lang* table.

4. staff:

Staff table stores the basic information, including gender, introduction, website ,and twitter. Since many of the staff may not have the following information, all these fields are nullable. Staffs use stage name for different positions or even visual novels in reality; therefore, we need a separate table to store the stage names of staff. The table *staff_stage_name*, an INT *staff_stage_name_id* is the primary key, and a non-nullable field "*en_stage_name*" stores the English stage name. Similar to other tables, "*original_stage_name*" is the staff's name in his native language. "*staff_id*" is a foreign key that links to the *staff* table.

There is not a stable relation between producer and staff in the visual novel industry, i.e., many of the jobs are outsourced, and one can work for multiple companies simultaneously. Therefore, this project separate staffs and producers. Since each staff work as a different role in each visual novel. There is an n:m relation between visual novels and staffs. Each relation has a position or role attribute. Therefore, the *vn_staff_relation* stores such information with three foreign keys linking to *vn*, *staff*, and *staff_position*.

5. character:

vn_char table stores the characters appear in the visual novels. Similar to other tables, characters have an English name ("*en_name*"), an original name ("*original_name*") and other names ("*alias*"). Also, characters have an introduction ("*intro*") attribute. Other columns including gender, birth date, height, weight, blood type and the age as columns. Moreover, for better search results, I added a trait table of characters, which is similar to the visual novel's tags.

Data Sources and Methods

There are some existing data on the web. Anidb [4] is a wiki-like database for all animations. Since many animations are originally visual novels, they have many visual novel related data. Also, I loaded the user votes for visual novels from this site. Additionally, I made use of the database dump or query interface from bangumi.tv [6] and vndb.org [5] to get metadata.

After collecting the metadata (in the form of tsv), we used Python Pandas [7] to load files and select the needed attributes. Then, we send the data to a SQL database to check the data consistency. Since data are collected from different platforms, we need to delete some repeated data and clear some rows of n:m relation tables since they might refer to none existing foreign keys. Finally, there are 27401 visual novels, 89552 characters, 9877 producers, 66397 releases, and 20381 staffs stored in the database.

The metadata is too big to upload; therefore, it is not provided. The data cleaning script can be accessed at https://github.com/imbaguanxin/CS3200/tree/master/vndb_project/data_cleaning

User Cases

Question 0: basic usages

The basic user case is finding visual novels, including finding visual novels by names, producers, and staffs. Also, finding releases according to visual novel is another essential problem that this database wants to solve.

The procedure that finds a visual novel's releases by name

```
drop procedure if exists find_release_by_vn_name;

delimiter //
create procedure find_release_by_vn_name
(
    in vn_title_param varchar(500)
)
begin

    select * from vn_release
    where vn_id in (select vn_id from vn where en_title like concat('%', vn_title_param, '%'));

end //
delimiter ;
```

The procedure that find the visual novels produced producer name

```
drop procedure if exists find_vn_by_producer_name;

delimiter //
create procedure find_vn_by_producer_name
(
    in producer_name_param varchar(500)
)
begin

    select en_title, original_title
    from vn_producer_relation
    left join vn using (vn_id)
    where producer_id in (
        select producer_id from producer where en_name like concat("%", producer_name_param, "%")
    );

end //
delimiter ;
```

The procedure that find the characters by visual novel's name.

```
drop procedure if exists find_chars_by_vn_name;
delimiter //
create procedure find_chars_by_vn_name
(
    in vn_name_param varchar(500)
)
begin

    select distinct vn_char.*, r.role
    from (select * from vn where en_title like concat('%', vn_name_param, '%')) as v
    left join char_vn_relation r using (vn_id)
    left join vn_char using (char_id);

end //
delimiter ;
```

The results of these procedures are rather long, so they are not included in this report. Test cases are provided in the queries file. I believe these procedures are the most common question that a user might use so I made them into procedures.

Question 1: How to find related traits

When a user is searching for traits of characters like blond hair, what is the most commonly shared traits on those characters who have blond hair? This result can help people search for characters. The following query finds out the most appeared five traits on characters with blond hair.

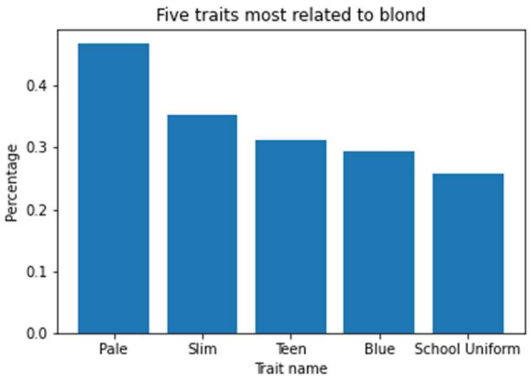
Query:

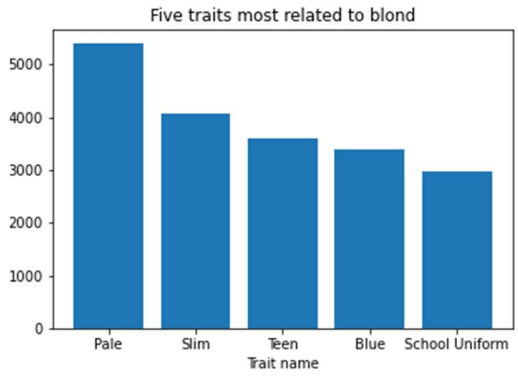
```sql
select trait_name, char_count
from (
    select r.trait_id, count(*) as char_count from(
        select trait_id, char_id from trait_char_relation
        join trait using (trait_id)
        where trait_name = 'Blond' and trait_description like "%hair%") as c
    left join trait_char_relation r on (c.char_id = r.char_id and c.trait_id != r.trait_id)
    group by r.trait_id
    order by char_count desc) as t_c
left join trait using (trait_id)
limit 5;
```

Result:

trait_name	char_count
Pale	5404
Slim	4059
Teen	3605
Blue	3390
School Uniform	2962

Since we can count the total characters with blond hair using query: "*select count(distinct char_id) from trait_char_relation where trait_id = (select trait_id from trait where trait_name = 'Blond');*", we can calculate the percentage as well:





Question 2: What is the most popular visual novel

Which visual novel is the most popular all over the world?

Firstly, I am deciding its popularity by counting the releases. The following query finds the top 10 visual novels with the most releases.

Query:

```
select vn.en_title, vn.original_title, count(release_id) as release_count
from vn_release
left join vn using (vn_id)
group by vn_id
order by release_count desc
limit 10;
```

Result:

en_title	original_title	release_count
Steins;Gate		73
Higurashi no Naku Koro ni	ひぐらしのなく頃に	60
Hakuouki ~Shinsengumi Kitan~	薄桜鬼 新選組奇譚	51
Planetarian ~Chiisana Hoshi no Yume~	Planetarian ～ちいさなほしのゆめ～	45
BlazBlue: Continuum Shift	ブレイブルー コンティニュアム シフト,	44
Higurashi no Naku Koro ni Kai	ひぐらしのなく頃に解	42
Clannad		40
Fate/Stay Night		37
Saya no Uta	沙耶の唄	36
Chaos;Child		36

If we change the definition of popularity, we may have a different result. Now we find out the top 10 visual novels that is translated to the greatest number of different languages.
Query:

```
select vn.en_title, vn.original_title, count(distinct language_code) as language_count
from vn
left join vn_release using (vn_id)
left join lang_vn_release_relation using (release_id)
group by vn_id
order by language_count desc
limit 10;
```

Result:

en_title	original_title	language_count
Knite		24
Doki Doki Literature Club!		13
Necrobarista		12
Carpe Diem		12
Amour Sucré		12
Sleepless Night		12
Katawa Shoujo		12
One Night, Hot Springs		12
Steins;Gate		12
Saya no Uta	沙耶の唄	12

The results are different. We can see that Japanese visual novels dominate the top 10 in terms of releases, while a large proportion of the top 10 visual novels in terms of language consists of

English visual novels. The overlapping visual novels are “Steins;Gate” and “Saya no Uta”, which are truly popular within the community.

Question 3: Who is the most productive staff and producer

In this question, we would like to find the most productive staff and producer.
Following query find the top 10 productive companies:

```
select en_name, original_name, website, language_code, count(vn_id) as vn_count
from vn_producer_relation
join producer using (producer_id)
where type = 'co'
group by producer_id
order by vn_count desc
limit 10;
```

Result:

en_name	original_name	website	language_code	vn_count
onomatope*Raspberry		http://onomatope.jp/	ja	11
Taito Corporation	株式会社タイトー	http://www.taito.co.jp/	ja	9
Blue Sky Co., Ltd.	有限会社ブルースカイ		ja	8
Mother Goose	マザーグース	http://www.milkcrown.co.jp/mothergoose/inde...	ja	6
Aquarium	アクアリウム	http://replay.waybackmachine.org/200904050...	ja	6
Accela, Inc		http://www.accelainc.com/service/	ja	5
Triplethreat	とりぷる・すれっと	http://crossover-game.jp/index2.html	ja	4
NekoNyan		https://nekonyansoft.com/	en	4
Softgarage Co., Ltd.	株式会社ソフトガレージ		ja	4
PSK	ＰＳＫ（パソコンショップ高知）		ja	4

Following query find the top 10 productive staff:
Since a staff may have many stage names, I pick the name with the lowest stage name id.
Query:

```
select en_stage_name, original_stage_name, vn_count
from (
    select staff_id, count(vn_id) as vn_count
    from vn_staff_relation
    group by staff_id
    order by vn_count desc
    limit 10) as sv
left join (
    select en_stage_name, original_stage_name, s.staff_id
    from (
        select min(staff_stage_name_id) as id, staff_id
        from staff_stage_name
        group by staff_id) as n
    left join staff_stage_name s on (n.id = s.staff_stage_name_id)
) as sn using (staff_id);
```

Result:

en_stage_name	original_stage_name	vn_count
Imai Yuka	今井 由香	481
Miyazaki Kyouichi	宮崎 京一	275
Sarah Anne Williams		241
Miyamori Yuu	宮森 ゆう	241
La Mancha	らまんちゃ	239
SEA		238
Ashiro Megu	亜城 めぐ	210
Maeno Tomoaki	前野 智昭	190
Kamiya Suguru	神夜 優	173
Factory Noise & AG	有限会社Factory Noise&AG	162

From the result, we find that many of the most productive staffs are character voice actors and BGM composers (conclude after google search).

Question 4: What languages do visual novel reader speak?

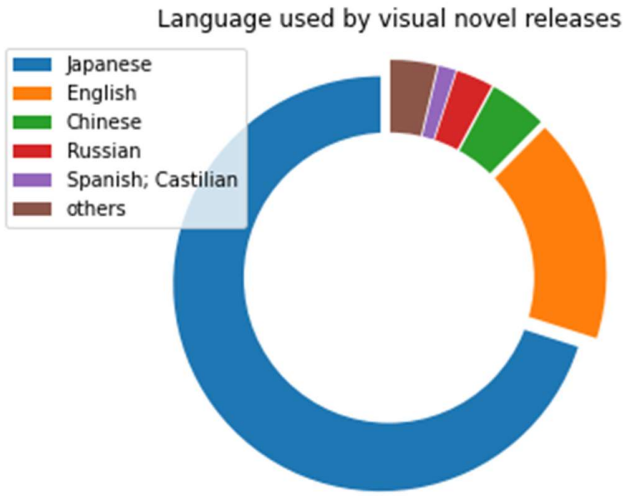
Find the top 5 languages that appear the most frequently in releases.

Query and Result:

```
select language_name, count(release_id) as lang_count
from lang_vn_release_relation
left join lang using (language_code)
group by language_code
order by lang_count desc
limit 5;
```

language_name	lang_count
Japanese	49054
English	12248
Chinese	3155
Russian	2029
Spanish; Castilian	947

Japanese and English are the most common release language. People speaking Chinese, Russian and Spanish also seem to like visual novels. We can get the total number by: "*select count(*) from lang_vn_release_relation;*". Therefore, we can draw the following pie chart:



Question 5: What kinds of visual novels are popular?

We are analyzing the tags of the top 50 popular visual novels (in terms of release number). If a tag is shared by half of the visual novels, then we consider it popular.

Query:

```
select tag_name, count(vn_id) as vn_count, tag_description
from
    (select vn.vn_id, count(release_id) as release_count
    from vn_release
    left join vn using (vn_id)
    group by vn_id
    order by release_count desc
    limit 50) as p
left join tag_vn_relation using (vn_id)
left join tag using (tag_id)
group by tag_id
having vn_count >= 25
order by vn_count desc;
```

Result:

tag_name	vn_count	tag_description
Male Protagonist	42	These games have a male protagonist.
ADV	39	In these games, the box with the text occupies only part of the screen, usually at the bottom. This is by far the most c...
Protagonist with a Face	36	This game's protagonist has a face that is visible in CGs and/or [URL=https://vndb.org/g1903]on their sprite(s)[/URL].n...
Multiple Endings	28	These games have several different endings.
One True End	28	Even though this game has multiple endings, one of them is considered to be the true ending to the story.nnSometimes ...
Unlockable Routes	28	This tag is for games with new unlockable routes. nnThey can be made available by getting to certain points in a game o...
Romance	27	Romance plays a large role in these games.
Branching Plot	27	This is the anti-thesis of [url=http://vndb.org/g145]Linear Plot[/url]. This story, at some point, will divide into multiple su...
Bad Endings with Story	27	The bad endings in this game have their own stories compared to the games that "Game Over" shortly after a bad choic...
No Sexual Content	26	This game has no sexual and real erotic content at all. It may have some small fan services here and there but those w...
Multiple Route Mystery	26	In order to understand the actual story of this game the player needs to play multiple routes. Each route reveals some ...
Mystery	25	This game features elements and themes related to the [url=http://en.wikipedia.org/wiki/Mystery_fiction]mystery[/url] ...

Conclusions

This project has provided a place for users to find more information about visual novels in multiple languages. The most useful function is collecting and presenting data of releases of visual novels for people all over the world. Thus, they don't need to dig into the internet to find translations in their language. Furthermore, this database might be developed to a web application in the future where visual novel lovers can discuss and share their loved visual novels.

In terms of limitations and future works, this database is not enough for a web application. Many of the components, including tables for users, logins, and publishers, are not included in the design. Also, visual novels involve not only plain text but also pictures, animations, and audio content, which are not included in this project. Collecting related visual and audio data and store them in the database consumes much more time and effort than my expectation, so I gave up the process. Furthermore, the relational model might not be the best way to tackle the problem of storing wiki-like websites' data with flexible attributes.

Author Contributions

Xin Guan is the only member in this project thus he did all the work.

References

1. Cavallaro, Dani. *Anime and the visual novel: narrative structure, design and play at the crossroads of animation and computer games*. McFarland & Company. 2010: p. 8..

2. *AMN and Anime Advanced Announce Anime Game Demo Downloads*. Hirameki International Group Inc. 8 February 2006. Retrieved 1 December 2006.

3. Organization That Made the Standard. *Codes for the representation of names of languages — Part 1: Alpha-2 code (2002).* (Standard No. 639-1) Retrieved from https://www.iso.org/iso-639-language-codes.html.

4. anidb.net softwares, Retrieved from https://anidb.net/software

5. Yorhel, *vndb API*, Retrieved from https://vndb.org/d11

6. bangumi, *bangumi API*, Retrieved from https://bangumi.tv/dev/app

7. McKinney, W., & others. (2010). *Data structures for statistical computing in python*. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).