Mysql server

brew services start/restart/stop mysql

Import csv file and build staging table

Edit: *sudo nano /etc/my.cnf*

*[mysqld]*

*secure-file-priv=/Users/yannonghuang/Desktop*

* pointing to directory where csv file is located.
* Make csv files readable.

Set both mysql server and client to import csv file:

When launching mysql client, use: *mysql --local-infile=1 -u root -p*

After inside mysql client: *SET GLOBAL local\_infile=1;*

**BUILD schools**

load

normal case:

LOAD DATA LOCAL INFILE '/Users/yannonghuang/Desktop/schools.csv' into table staging FIELDS TERMINATED BY ',';

When transformation is needed:

LOAD DATA LOCAL INFILE '/Users/yannonghuang/Desktop/schools.csv'

into table staging

FIELDS TERMINATED BY ','

IGNORE 1 Lines

(

code,

name,

region,

city,

county,

community,

schoolBoardRegisteredName,

schoolboard,

principal,

startAt,

lastVisit,

stage,

request,

status,

description,

address,

phone,

email,

principalEmail,

principalCell,

principalWechat,

category,

studentsCount,

teachersCount,

kClassesCount,

g1ClassesCount ,

g2ClassesCount ,

g3ClassesCount ,

g4ClassesCount ,

g5ClassesCount ,

g6ClassesCount ,

kStudentsCount ,

g1StudentsCount ,

g2StudentsCount ,

g3StudentsCount ,

g4StudentsCount ,

g5StudentsCount ,

g6StudentsCount ,

mStudentsCount ,

boarderCount ,

stayBehindCount ,

@vInternetExists,

@vComputerRoomExists,

computerRoomCount ,

computersCount ,

multimediaSystemsCount ,

@vLibraryExists,

bookCornersCount ,

booksCount )

*SET* computerRoomExists*:=* if( @vComputerRoomExists= '', NULL, @vComputerRoomExists *=* '是'),

internetExists*:=* if( @vInternetExists= '', NULL, @vInternetExists *=* '是'),

libraryExists*:=* if( @vLibraryExists= '', NULL, @vLibraryExists *=* '是')

;

insert into surveys(

createdAt,

updatedAt,

status ,

request ,

address ,

region ,

city,

county,

community,

schoolBoardRegisteredName,

schoolboard,

phone,

email,

principal,

principalCell,

principalWechat,

description,

category,

studentsCount,

stayBehindCount,

boarderCount,

teachersCount,

kClassesCount,

g1ClassesCount,

g2ClassesCount,

g3ClassesCount,

g4ClassesCount,

g5ClassesCount,

g6ClassesCount,

kStudentsCount,

g1StudentsCount,

g2StudentsCount,

g3StudentsCount,

g4StudentsCount,

g5StudentsCount,

g6StudentsCount,

mStudentsCount,

computersCount,

computerRoomExists,

computerRoomCount,

internetExists,

multimediaSystemsCount,

libraryExists,

bookCornersCount,

booksCount,

code

)

select

now() as createdAt,

now() as updatedAt,

status ,

request ,

address ,

region ,

city,

county,

community,

schoolBoardRegisteredName,

schoolboard,

phone,

email,

principal,

principalCell,

principalWechat,

description,

category,

studentsCount,

stayBehindCount,

boarderCount,

teachersCount,

kClassesCount,

g1ClassesCount,

g2ClassesCount,

g3ClassesCount,

g4ClassesCount,

g5ClassesCount,

g6ClassesCount,

kStudentsCount,

g1StudentsCount,

g2StudentsCount,

g3StudentsCount,

g4StudentsCount,

g5StudentsCount,

g6StudentsCount,

mStudentsCount,

computersCount,

computerRoomExists,

computerRoomCount,

internetExists,

multimediaSystemsCount,

libraryExists,

bookCornersCount,

booksCount,

code

from staging;

insert into schools(

createdAt,

updatedAt,

code ,

name ,

description ,

principal ,

region ,

city,

county,

community,

address ,

phone ,

studentsCount ,

teachersCount ,

startAt,

stage ,

status ,

request ,

category ,

lastVisit

)

select

now() as createdAt,

now() as updatedAt,

code ,

name ,

description ,

principal ,

region ,

city,

county,

community,

address ,

phone ,

studentsCount ,

teachersCount ,

MAKEDATE(startAt,1) as startAt,

stage ,

status ,

request ,

category ,

MAKEDATE(lastVisit,1) as lastVisit

from staging;

Append donors

update nodedb.schools

set donor = (

select ssefSPM.donation.donor

from ssefSPM.donation, ssefSPM.schools

where ssefSPM.donation.school\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = nodedb.schools.code limit 1)

where id > 0

delete FROM nodedb.users where (emailverified is null) and id > 0;

insert into users (

createdAt,

updatedAt,

email ,

chineseName ,

phone ,

wechat ,

title ,

contactOnly,

code

)

select

now() as createdAt,

now() as updatedAt,

principalEmail as email,

principal as chineseName,

principalCell as phone,

principalWechat as wechat,

'校长' as title,

1 as contactOnly,

code

from staging;

initialize roles

INSERT INTO nodedb.user\_roles (

`createdAt`,

`updatedAt`,

userId, roleId)

select

now() as createdAt,

now() as updatedAt,

nodedb.users.id,

if (nodedb.users.schoolId is not null, 1,

if (nodedb.users.username = 'shinshin', 3, 2)

)

from nodedb.users

update surveys

set schoolId = (select id from schools where schools.code = surveys.code)

where id > 0 and

(select count(\*) from schools where schools.code = surveys.code) = 1

update users

set schoolId = (select id from schools where schools.code = users.code)

where id > 0 and (code is not null) and

(select count(\*) from schools where schools.code = users.code) = 1

update schools

set principalId = (select id from users where schools.code = users.code)

where id > 0 and

(select count(\*) from users where schools.code = users.code) = 1

update surveys

set principalId = (select id from users where surveys.code = users.code)

where id > 0 and

(select count(\*) from users where surveys.code = users.code) = 1

Normalize region names

update nodedb.schools

set region =

if (region = '湖南湘西', '湖南省湘西州',

if (region = '广西', '广西壮族自治区',

if (region = '内蒙古', '内蒙古自治区',

if (region = '宁夏', '宁夏回族自治区',

if (region = '新疆', '新疆维吾尔族自治区',

if (region = '西藏', '西藏自治区',

if (region = '重庆', '重庆市',

if (region regexp '.省', region,

concat(region, '省')

)

)

)

)

)

)

)

)

where id > 0

[¥¥¥¥](NULL) 2nd normalization: begin

update nodedb.schools

set region =

if (region = '湖南省湘西州省', '湖南省湘西州',

if (region = '广西壮族自治区省', '广西壮族自治区',

if (region = '内蒙古自治区省', '内蒙古自治区',

if (region = '宁夏回族自治区省', '宁夏回族自治区',

if (region = '新疆维吾尔族自治区省', '新疆维吾尔族自治区',

if (region = '西藏自治区省', '西藏自治区',

if (region = '重庆省', '重庆市', region

)

)

)

)

)

)

)

where id > 0

[¥¥¥¥](NULL) 2nd normalization: end

update nodedb.surveys

set region =

if (region = '湖南湘西', '湖南省湘西州',

if (region = '广西', '广西壮族自治区',

if (region = '内蒙古', '内蒙古自治区',

if (region = '宁夏', '宁夏回族自治区',

if (region = '新疆', '新疆维吾尔族自治区',

if (region = '西藏', '西藏自治区',

if (region = '重庆', '重庆市',

if (region regexp '.省', region,

concat(region, '省')

)

)

)

)

)

)

)

)

where id > 0

[¥¥¥¥](NULL) 2nd normalization: begin

update nodedb.surveys

set region =

if (region = '湖南省湘西州省', '湖南省湘西州',

if (region = '广西壮族自治区省', '广西壮族自治区',

if (region = '内蒙古自治区省', '内蒙古自治区',

if (region = '宁夏回族自治区省', '宁夏回族自治区',

if (region = '新疆维吾尔族自治区省', '新疆维吾尔族自治区',

if (region = '西藏自治区省', '西藏自治区',

if (region = '重庆省', '重庆市', region

)

)

)

)

)

)

)

where id > 0

[¥¥¥¥](NULL) 2nd normalization: end

**BUILD xr projects**

CREATE TABLE `staging\_xr` (

`startAt` year DEFAULT NULL,

`code` int NOT NULL,

`schoolName` varchar(255) DEFAULT NULL,

`name` varchar(255) DEFAULT NULL,

`description` varchar(255) DEFAULT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=512 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

load

LOAD DATA LOCAL INFILE '/Users/yannonghuang/Desktop/xr.csv' into table staging\_xr FIELDS TERMINATED BY ',';

insert into `projects` (

`createdAt`,

`updatedAt`,

`name`,

`description` ,

`schoolId`,

`startAt`,

`xr`

)

select

now() as createdAt,

now() as updatedAt,

staging\_xr.name,

staging\_xr.description,

schools.id,

MAKEDATE(staging\_xr. startAt, 2) as startAt,

1

from staging\_xr, schools

where staging\_xr.code = schools.code

update schools

set xr = 1

where exists (

select \*

from projects

where projects.schoolId = schools.id and projects.xr = 1

)

**Build DB instance**

Load backup\_ssefSPM\_0202\_2022.sql to build “ssefSPM” DB

delete from forms where id > 0

INSERT INTO `nodedb`.`forms`

(

`createdAt`,

`updatedAt`,

`title`,

`description`,

`deadline`,

`startAt`,

`form\_id`

)

select

now() as createdAt,

now() as updatedAt,

`title`,

`description`,

`due\_date` as `deadline`,

MAKEDATE(`year`,1) as startAt,

`id` as `form\_id`

from ssefSPM.forms

update `nodedb`.`forms`

set startAt = (

select MAKEDATE(ssefSPM.program\_instances.particular\_year,10)

from ssefSPM.program\_instances, ssefSPM.forms

where ssefSPM.program\_instances.id = ssefSPM.forms.instance\_id

and ssefSPM.forms.id = `nodedb`.`forms`.form\_id

)

where id > 0

delete from responses where id > 0

INSERT INTO `nodedb`.`responses`

(

`createdAt`,

`updatedAt`,

`startAt`,

`title`,

`schoolId`,

`result\_id`

)

select

now() as createdAt,

now() as updatedAt,

ssefSPM.program\_cases.`create\_on` as `startAt`,

ssefSPM.program\_cases.`title` as `title`,

`nodedb`.`schools`.`id` as `schoolId`,

ssefSPM.case\_application\_mapping.result\_id as `result\_id`

from ssefSPM.program\_cases, ssefSPM.schools, ssefSPM.case\_application\_mapping, `nodedb`.`schools`

where ssefSPM.program\_cases.s\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = `nodedb`.`schools`.`code`

and ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

UPDATE `nodedb`.`responses`

SET startAt = (

select MAKEDATE(ssefSPM.program\_instances.particular\_year,10)

from ssefSPM.program\_instances, ssefSPM.program\_cases, ssefSPM.case\_application\_mapping

where ssefSPM.program\_instances.id = ssefSPM.program\_cases.instance\_id

and ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

and `nodedb`.`responses`.`result\_id` = ssefSPM.case\_application\_mapping.result\_id

limit 1

)

where id > 0

-----

UPDATE `nodedb`.`responses`

SET

`formId` = (select `nodedb`.`forms`.id

from `nodedb`.`forms`, ssefSPM.form\_result

where

`nodedb`.`responses`.result\_id = ssefSPM.form\_result.id

And

`nodedb`.`forms`.form\_id = ssefSPM.form\_result.form\_id)

Where id > 0

delete FROM nodedb.projects where xr is null and id > 0;

INSERT INTO `nodedb`.`projects`

(

`createdAt`,

`updatedAt`,

`name`,

`budget`,

`status`,

`description`,

`startAt`,

`schoolId`

)

select

now() as createdAt,

now() as updatedAt,

ssefSPM.program\_cases.`title` as `title`,

ssefSPM.program\_cases. budget as budget,

'已完成' as `status`,

ssefSPM.program\_cases.`description` as `description`,

makedate(ssefSPM.program\_instances.particular\_year, 10),

`nodedb`.`schools`.`id` as `schoolId`

from ssefSPM.program\_instances, ssefSPM.program\_cases, ssefSPM.schools, `nodedb`.`schools`

where ssefSPM.program\_instances.id = ssefSPM.program\_cases.instance\_id

and ssefSPM.program\_cases.s\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = `nodedb`.`schools`.`code`

and not exists (select \*

from `nodedb`.`responses`, ssefSPM.case\_application\_mapping

where

ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

and ssefSPM.case\_application\_mapping.result\_id = `nodedb`.`responses`.result\_id

)

INSERT INTO `nodedb`.`projects`

(

`createdAt`,

`updatedAt`,

`name`,

`budget`,

`status`,

`description`,

`startAt`,

`schoolId`,

`responseId`

)

select

now() as createdAt,

now() as updatedAt,

ssefSPM.program\_cases.`title` as `title`,

ssefSPM.program\_cases. budget as budget,

'已完成' as `status`,

ssefSPM.program\_cases.`description` as `description`,

makedate(ssefSPM.program\_instances.particular\_year, 10),

`nodedb`.`schools`.`id` as `schoolId`,

`nodedb`.`responses`.id

from ssefSPM.program\_instances, ssefSPM.program\_cases, ssefSPM.schools, `nodedb`.`responses`, `nodedb`.`schools`, ssefSPM.case\_application\_mapping

where ssefSPM.program\_instances.id = ssefSPM.program\_cases.instance\_id

and ssefSPM.program\_cases.s\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = `nodedb`.`schools`.`code`

and ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

and ssefSPM.case\_application\_mapping.result\_id = `nodedb`.`responses`.result\_id

truncate staging\_forms;

INSERT INTO nodedb.staging\_forms

(

`createdAt`,

`updatedAt`,

`title`,

`q\_id`,

`q\_text`,

`input\_type`,

`data\_field\_name`,

`options`,

`form\_id`

)

SELECT

now() as createdAt,

now() as updatedAt,

ssefSPM.forms.title as title,

ssefSPM.form\_question.id as q\_id,

question as q\_text,

input\_type,

data\_field\_name,

`options` ,

ssefSPM.forms.id as `form\_id`

FROM ssefSPM.form\_question, ssefSPM.forms

where ssefSPM.form\_question.form\_id = ssefSPM.forms.id

truncate staging\_**responses**;

**staging\_responses**:

INSERT INTO nodedb.staging\_responses

(

`createdAt`,

`updatedAt`,

`title`,

`result\_id`,

`q\_id`,

`q\_value`

)

SELECT

now() as createdAt,

now() as updatedAt,

CONCAT(ssefSPM.forms.title, '-', ssefSPM.form\_result.school\_id) as title,

ssefSPM.form\_result.id as result\_id,

ssefSPM.form\_result\_content.question\_id as q\_id,

ssefSPM.form\_result\_content.question\_value as q\_value

from ssefSPM.forms, ssefSPM.form\_result, ssefSPM.form\_result\_content

where ssefSPM.forms.id = ssefSPM.form\_result.form\_id

and ssefSPM.form\_result.id = ssefSPM.form\_result\_content.result\_id

Launch fake action: 项目申请/新增项目征集, to load forms/responses into the FormBuilder templates from the old data formats.

INSERT INTO `nodedb`.`users`

(

`createdAt`,

`updatedAt`,

`username`,

`email`,

`chineseName`,

`phone`,

`wechat`,

`startAt`,

`lastLogin`,

`user\_id`

)

SELECT

now() as createdAt,

now() as updatedAt,

`users`.`login\_id`,

`users`.`email`,

`users`.`chinese\_name`,

`users`.`cellphone`,

`users`.`wechat`,

`users`.`start\_date`,

`users`.`last\_login`,

`users`.`user\_id`

FROM `ssefSPM`.`users`

update `nodedb`.`users`

set schoolId = (

select nodedb.schools.id

FROM ssefSPM.form\_result, ssefSPM.schools, nodedb.schools

where `nodedb`.`users`.user\_id = ssefSPM.form\_result.filled\_by

and ssefSPM.form\_result.school\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = nodedb.schools.code

limit 1

)

where (

select count(distinct nodedb.schools.id )

FROM ssefSPM.form\_result, ssefSPM.schools, nodedb.schools

where `nodedb`.`users`.user\_id = ssefSPM.form\_result.filled\_by

and ssefSPM.form\_result.school\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = nodedb.schools.code

group by `nodedb`.`users`.user\_id

) = 1 and id > 0

**OBSOLETE: BELOW BACKUP ONLY……………….**

update `nodedb`.`users`

set schoolId = null

where (

select count(distinct nodedb.schools.id )

FROM ssefSPM.form\_result, ssefSPM.schools, nodedb.schools

where `nodedb`.`users`.user\_id = ssefSPM.form\_result.filled\_by

and ssefSPM.form\_result.school\_id = ssefSPM.schools.id

and ssefSPM.schools.school\_number = nodedb.schools.code

group by `nodedb`.`users`.user\_id

) > 1 and id > 0

update responses

set userId = (

select users.id

from users, ssefSPM.form\_result

where ssefSPM.form\_result.id = responses.result\_id

and ssefSPM.form\_result.filled\_by = users.user\_id

)

where id > 0

**OBSOLETE: schoolBoardRegisteredName & schoolboard**

load

LOAD DATA LOCAL INFILE '/Users/yannonghuang/Desktop/2019Projects\_import.csv' into table staging\_project FIELDS TERMINATED BY ',';

update surveys

set

schoolBoardRegisteredName = (select schoolBoardRegisteredName

from staging\_project

where surveys.code = staging\_project.code limit 1),

schoolBoard = (select schoolBoard

from staging\_project

where surveys.code = staging\_project.code limit 1)

where id > 0

**Migrate school documents**

1. Move documents under <deploy directory>/backend/upload/;
2. Import the *documents* table dump.

(<deploy directory>: /home/bitnami/htdocs/projects/shinshin)

Here is how to populate the *documents* table:

INSERT INTO `nodedb`.`documents`

(

`createdAt`,

`updatedAt`,

`originalname`,

`filename`,

`path`,

`destination`,

`encoding`,

`mimetype`,

`docCategory`,

`schoolId`)

SELECT

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`file`,

`ssefSPM`.`document`.`file`,

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/School/', ssefSPM.schools.id, '/', `ssefSPM`.`document`.`file`),

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/School/', `ssefSPM`.`document`.`file`),

'7bit',

If (`ssefSPM`.`document`.`file` regexp '.docx', 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',

If (`ssefSPM`.`document`.`file` regexp '.jpg' OR `ssefSPM`.`document`.`file` regexp '.jpeg', 'image/jpeg',

If (`ssefSPM`.`document`.`file` regexp '.pdf', 'application/pdf',

If (`ssefSPM`.`document`.`file` regexp '.txt', 'text/plain',

If (`ssefSPM`.`document`.`file` regexp '.doc', 'application/msword',

If (`ssefSPM`.`document`.`file` regexp '.xls', 'application/vnd.ms-excel',

If (`ssefSPM`.`document`.`file` regexp '.pptx', 'application/vnd.openxmlformats-officedocument.presentationml.presentation',

If (`ssefSPM`.`document`.`file` regexp '.xlsx', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',

'application/octet-stream'

)

)

)

)

)

)

)

)

,

ssefSPM.translaters.name\_zh\_cn,

nodedb.schools.id

FROM ssefSPM.document, ssefSPM.schools, nodedb.schools, ssefSPM.document\_types, ssefSPM.translaters

where ssefSPM.document.related\_model = 'School'

and ssefSPM.document.related\_key = ssefSPM.schools.id

and nodedb.schools.code = ssefSPM.schools.school\_number

and ssefSPM.document.type\_id = ssefSPM.document\_types.id

and ssefSPM.document\_types.id\_translater = ssefSPM.translaters.id

**Migrate school profile photos**

INSERT INTO `nodedb`.`documents`

(

`createdAt`,

`updatedAt`,

`originalname`,

`filename`,

`path`,

`destination`,

`encoding`,

`mimetype`,

`docCategory`,

`schoolId`)

SELECT

now() as createdAt,

now() as updatedAt,

SUBSTRING\_INDEX(ssefSPM.schools.profile\_photo, '/', -1),

SUBSTRING\_INDEX(ssefSPM.schools.profile\_photo, '/', -1),

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/schoolphotos/',

SUBSTRING\_INDEX(ssefSPM.schools.profile\_photo, '/', -1)

),

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/schoolphotos/',

SUBSTRING\_INDEX(ssefSPM.schools.profile\_photo, '/', -1)

),

'7bit',

'image/jpeg',

'profile\_photo',

nodedb.schools.id

FROM ssefSPM.schools, nodedb.schools

where

nodedb.schools.code = ssefSPM.schools.school\_number

**Migrate project documents**

INSERT INTO `nodedb`.`dossiers`

(

`createdAt`,

`updatedAt`,

`originalname`,

`filename`,

`path`,

`destination`,

`encoding`,

`mimetype`,

`docCategory`,

`projectId`)

SELECT

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`file`,

`ssefSPM`.`document`.`file`,

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/ProgramCase/', ssefSPM. program\_cases.id, '/', `ssefSPM`.`document`.`file`),

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/ProgramCase/', `ssefSPM`.`document`.`file`),

'7bit',

If (`ssefSPM`.`document`.`file` regexp '.docx', 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',

If (`ssefSPM`.`document`.`file` regexp '.jpg' OR `ssefSPM`.`document`.`file` regexp '.jpeg', 'image/jpeg',

If (`ssefSPM`.`document`.`file` regexp '.pdf', 'application/pdf',

If (`ssefSPM`.`document`.`file` regexp '.txt', 'text/plain',

If (`ssefSPM`.`document`.`file` regexp '.doc', 'application/msword',

If (`ssefSPM`.`document`.`file` regexp '.xls', 'application/vnd.ms-excel',

If (`ssefSPM`.`document`.`file` regexp '.pptx', 'application/vnd.openxmlformats-officedocument.presentationml.presentation',

If (`ssefSPM`.`document`.`file` regexp '.xlsx', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',

'application/octet-stream'

)

)

)

)

)

)

)

)

,

ssefSPM.translaters.name\_zh\_cn,

nodedb.projects.id

FROM ssefSPM.document, ssefSPM.schools, nodedb.schools, ssefSPM.document\_types, ssefSPM.translaters, ssefSPM.program\_cases, nodedb.projects

where ssefSPM.document.related\_model = 'ProgramCase'

and ssefSPM.document.type\_id = ssefSPM.document\_types.id

and ssefSPM.document\_types.id\_translater = ssefSPM.translaters.id

and ssefSPM.document.related\_key = ssefSPM. program\_cases.id

and ssefSPM.program\_cases.s\_id = ssefSPM.schools.id

and nodedb.schools.code = ssefSPM.schools.school\_number

and nodedb.schools.id = nodedb.projects.schoolId

and ssefSPM.program\_cases.`title` = nodedb.projects.name

and ssefSPM.program\_cases.`description` = nodedb.projects.description

and not exists (select \*

from `nodedb`.`responses`, ssefSPM.case\_application\_mapping

where

ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

and ssefSPM.case\_application\_mapping.result\_id = `nodedb`.`responses`.result\_id

)

UNION

INSERT INTO `nodedb`.`dossiers`

(

`createdAt`,

`updatedAt`,

`originalname`,

`filename`,

`path`,

`destination`,

`encoding`,

`mimetype`,

`docCategory`,

`projectId`)

SELECT

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`create\_time`,

`ssefSPM`.`document`.`file`,

`ssefSPM`.`document`.`file`,

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/ProgramCase/', ssefSPM. program\_cases.id, '/', `ssefSPM`.`document`.`file`),

Concat('/home/bitnami/htdocs/projects/shinshin/backend/upload/ProgramCase/', `ssefSPM`.`document`.`file`),

'7bit',

If (`ssefSPM`.`document`.`file` regexp '.docx', 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',

If (`ssefSPM`.`document`.`file` regexp '.jpg' OR `ssefSPM`.`document`.`file` regexp '.jpeg', 'image/jpeg',

If (`ssefSPM`.`document`.`file` regexp '.pdf', 'application/pdf',

If (`ssefSPM`.`document`.`file` regexp '.txt', 'text/plain',

If (`ssefSPM`.`document`.`file` regexp '.doc', 'application/msword',

If (`ssefSPM`.`document`.`file` regexp '.xls', 'application/vnd.ms-excel',

If (`ssefSPM`.`document`.`file` regexp '.pptx', 'application/vnd.openxmlformats-officedocument.presentationml.presentation',

If (`ssefSPM`.`document`.`file` regexp '.xlsx', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',

'application/octet-stream'

)

)

)

)

)

)

)

)

,

ssefSPM.translaters.name\_zh\_cn,

nodedb.projects.id

FROM ssefSPM.document, ssefSPM.schools, nodedb.schools, ssefSPM.document\_types, ssefSPM.translaters, ssefSPM.program\_cases, nodedb.projects, `nodedb`.`responses`, ssefSPM.case\_application\_mapping

where ssefSPM.document.related\_model = 'ProgramCase'

and ssefSPM.document.type\_id = ssefSPM.document\_types.id

and ssefSPM.document\_types.id\_translater = ssefSPM.translaters.id

and ssefSPM.document.related\_key = ssefSPM. program\_cases.id

and ssefSPM.program\_cases.s\_id = ssefSPM.schools.id

and nodedb.schools.code = ssefSPM.schools.school\_number

and nodedb.schools.id = nodedb.projects.schoolId

and ssefSPM.program\_cases.`title` = nodedb.projects.name

and ssefSPM.program\_cases.`description` = nodedb.projects.description

and ssefSPM.case\_application\_mapping.case\_id = ssefSPM.program\_cases.id

and ssefSPM.case\_application\_mapping.result\_id = `nodedb`.`responses`.result\_id