BS Degree in Applied Mathematics & Computer Science Academic year: 2024/25 - 3rd year, 2nd term Subject: File Structures and Databases First Assignment's Report: Relational DB Design and Population	uc3m | Universidad Carlos III de Madrid

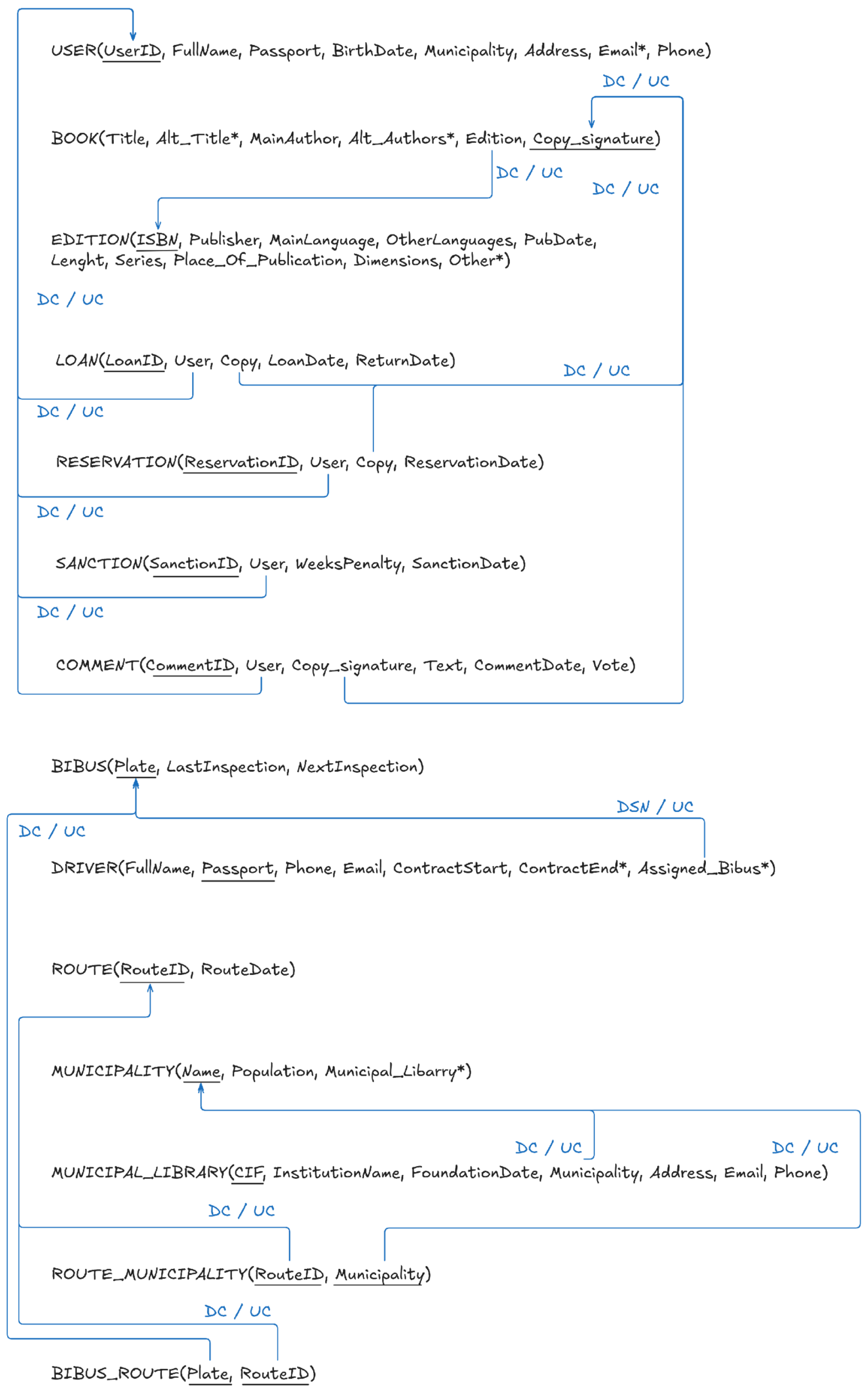
Lecturers:	**Francisco Javier Calle Gomez, Daniel Esteban Villamil Sierra**		
Group:	**283**	**Lab User**	**fsdb283**
Student:	**Antonio Nicolas Lemus Yeguas**	**NIA:**	**100522110**
Student:	**Denis Loren Moldovan**	**NIA:**	**100522240**
Student:	**Jorge Adrian Saghin Dudulea**	**NIA:**	**100522257**

1 Introduction

The Foundation for the Diffusion of Culture (Foundicu Org.) requires an updated database system to efficiently manage its bibliographic collections and mobile library services. The current database is inadequate, containing only three disjointed tables with limited constraints, leading to poor data integrity and insufficient semantic coverage. This project aims to design a new relational database that meets the foundation's operational needs, implements necessary constraints, and facilitates seamless data migration. The accompanying files, create-tables.sql and migration.sql, contain the implementation of the new schema and data migration procedures, respectively.

In this document will be present a diagram of the followed database structure, with the primary keys and the foreign keys, accompanied with tables that define the before mentioned structure.

2 Relational Design



USER(UserID, FullName, Passport, BirthDate, Municipality, Address, Email*, Phone)

BOOK(Title, Alt_Title*, MainAuthor, Alt_Authors*, Edition, Copy_signature)

EDITION(ISBN, Publisher, MainLanguage, OtherLanguages, PubDate, Lenght, Series, Place_Of_Publication, Dimensions, Other*)

LOAN(LoanID, User, Copy, LoanDate, ReturnDate)

RESERVATION(ReservationID, User, Copy, ReservationDate)

SANCTION(SanctionID, User, WeeksPenalty, SanctionDate)

COMMENT(CommentID, User, Copy_signature, Text, CommentDate, Vote)

BIBUS(Plate, LastInspection, NextInspection)

DRIVER(FullName, Passport, Phone, Email, ContractStart, ContractEnd*, Assigned_Bibus*)

ROUTE(RouteID, RouteDate)

MUNICIPALITY(Name, Population, Municipal_Libarry*)

MUNICIPAL_LIBRARY(CIF, InstitutionName, FoundationDate, Municipality, Address, Email, Phone)

ROUTE_MUNICIPALITY(RouteID, Municipality)

BIBUS_ROUTE(Plate, RouteID)

- Implicit semantics:

Presp_id	Stage	Mechanism	Description
I_1	Design	Primary key	Users are identified by their user ID.
I_2	Design	Primary key	Books are identified by their own copy signature
I_3	Design	Primary key	Each edition of a book is identified by its ISBN
I_4	Design	Primary key	Loans are identified by their ID
I_5	Design	Primary key	Each reservation of a book has its own ID
I_6	Design	Primary key	Sanctions are identified also by their ID
I_7	Design	Primary key	Each comment has its own ID
I_8	Design	Primary key	Bibuses are identified by their plates
I_9	Design	Primary key	Drivers are uniquely identified by their passports
I_{10}	Design	Primary key	Each route can be found by their ID
I_{11}	Design	Primary key	Each municipality has its own unique name
I_{12}	Design	Primary key	Each municipal library has its own CIF
I_{13}	Design	Primary key	Each route passing through a municipality is identified by its route ID and the municipality it goes to
I_{14}	Design	Primary key	The routes that a bibus will take its identified by the plate of the assigned bibus and the route ID.

Table 1: Implicit semantics incorporated into the relational graph

- Non-observed explicit semantics:

Presp_id	Description
S_1	Phone numbers have 9 digits (at least, at most)
S_2	Passports follow the format of the corresponding country standard, with checks that validate it before uploading to the database.
S_3	ISBNs have 13 digits
S_4	Plates have 4 digits, followed by 3 letters
S_5	The publishment date of the books is just the year
S_6	Books returned late are marked for condition review
S_7	Municipal libraries must have a valid email address
S_8	Editions must have a valid publication year
S_9	A user cannot reserve more than 5 books at a time
S_{10}	A bibus cannot operate if scheduled for maintenance
S_{11}	The foundation records user penalties for late returns
S_{12}	A municipal library must be associated with a valid municipality
S_{13}	A driver cannot be assigned to multiple mobile libraries on the same day
S_{14}	Book reservations are automatically canceled upon user penalization

Table 2: Non-observed explicit semantics

3 Relational Statics Implementation in SQL (DDL)

The script `create-tables.sql` is composed of two parts, the beginning where we drop all the previously created tables, and the rest where the new ones are created, in order to make the mass insertion from the previously stated ones. The new structure will have higher modularity, separating most of the information to their respective use.

Re-incorporated semantics:

Presp_id	Solution Description
S_1	Field size is 9 in the phone column, so the limit cannot be surpassed when adding numbers.
S_2	FOREIGN KEY constraint ensures that a book cannot exist if it doesn't have an ISBN (No edition exists for that book).
S_3	FOREIGN KEY constraints ensure that a loan cannot exist if no user nor book .
S_4	FOREIGN KEY constraint ensures that no reservation can exist without a user nor a book.
S_5	DEFAULT constraint on the vote of a comment ensures that even if the user doesn't input the vote, a score of 0 will be stored.
S_6	FOREIGN KEY constraint on municipal_library ensures that no library is stored if the municipality doesn't exist.

Table 3: re-incorporated explicit semantics

Incorporated implicit semantics:

Presp_id	Stage	Mechanism	Description
I_{15}	Not implem.	Check	There is no *age* greater than 120 years old
I_{16}	Implem.	Check	No user can have more than 2 active loans
I_{17}	Not implem.	Constraint	ISBN format is validated before insertion
I_{18}	Not implem.	Check	Users under 18 require guardian approval
I_{19}	Not Implem.	Foreign key	Reservations must link to scheduled bibus stops

Table 1(cont.): implicit semantics incorporated in the definition of each table

Excluded semantics:

Presp_id	Description	Cause	Explicit/ Implicit
E_1	Contracts are automatically updated with the company's update (integrity option UC on the FK referencing *Companies*).	PL/SQL does not observe this integrity option	Implicit
E_2	When book loans are overdue, a notification is sent to the user in question.	Requires an external notification system	Explicit
E_3	Books availability is updated in real-time.	Performance constraints	Explicit
E_4	Automatic library inventory updates.	Requires external API integration	Explicit
E_5	Reservations are limited dynamically based on demands.	Requires additional heuristics.	Explicit

Table 5: explicit semantics excluded in the creation of each table

4 Workload (DML)

We started dumping data into our tables first with all those tables that do not contain any foreign key, i.e. users, edition, routes, bibus and municipality, then we inserted those that depend on those that we have already created and so on. We separated the insertion in two phases, the first will consist of the users and the books, with the massive insertion of: users, editions, books, loan and comments, and a second part consisting on the insertion of data into tables: bibus, driver, routes, municipality, municipal_library, route_municipality and bibus_route.

1. User

For the insertion of data into the users table, we have inserted the following code into sql:

```
insert into users
  select distinct
    user_id,
    surname1||surname2||name,
    passport,
    to_date(birthdate, 'DD-MM-YYYY'),
    null,
    substr(address, 1, instr(address, ',', 1, 2)),
    email,
    phone
  from fsdb.loans
  where
    birthdate != '29-02-1970'
    ;
```

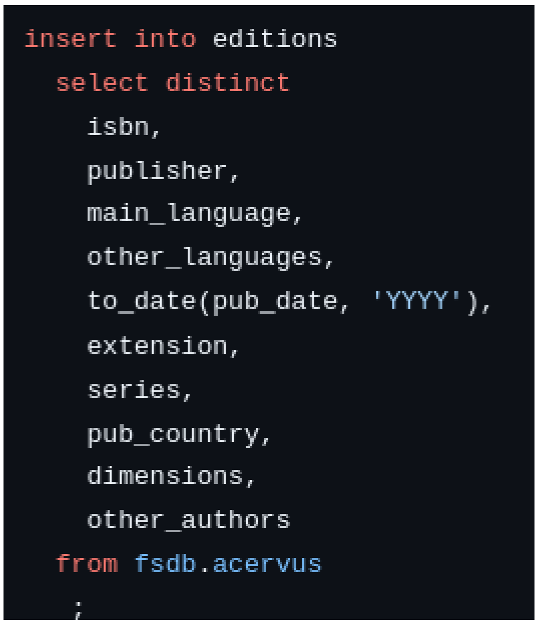
During the implementation of the code, we had several issues. The first one was a constraint regarding the violation of the primary key, this was due to several tuples containing the same user_id (our primary key) but having variations in some values. Then we noticed that some users had the same id and address, but this last feature had the same street but a different city/town, then we decided to insert into the table a substring of the address that will go from the first character until the number indicated by the instring function of where the second instance of the character ',' is located. That is because in the database fsdb.loans, address is: Name_of_street, post_code,Name_of_town. The substring that will be inserted will follow the pattern: Name_of_street,post_code.

Another error we noticed is that a user had a birthday 29-02-1970, which is not valid because 1970 was not a leap year. We then included in the code the condition that birthdate must have a value different that '29-02-1970'.

PD: This error happened with all the insertion of dates, we had to cast into a date all the dates used

2. Editions

For the insertion of data into the editions table, we have inserted the following code into sql:

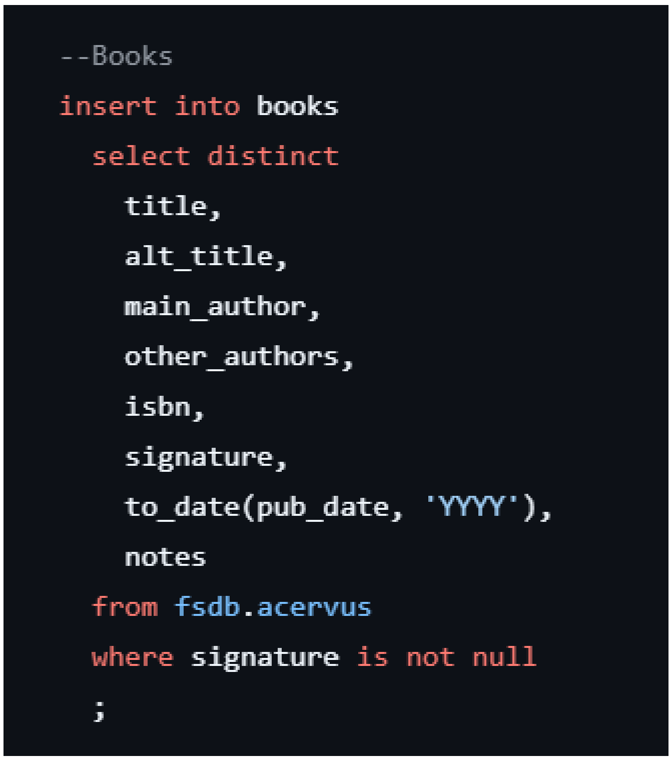


```sql
insert into editions
  select distinct
    isbn,
    publisher,
    main_language,
    other_languages,
    to_date(pub_date, 'YYYY'),
    extension,
    series,
    pub_country,
    dimensions,
    other_authors
  from fsdb.acervus
   ;
```

The only error we have was the casting of the dates. As the to_date values from the fsdb.acervus are just the year, for example, '1990', we had to cast them into a date with the format 'YYYY'.

3. Books

For the insertion of data into the books table, we have inserted the following code into sql:

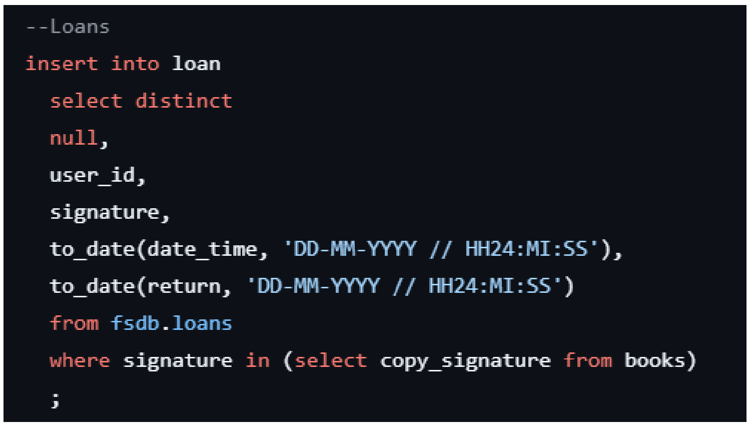


```
--Books
insert into books
  select distinct
    title,
    alt_title,
    main_author,
    other_authors,
    isbn,
    signature,
    to_date(pub_date, 'YYYY'),
    notes
  from fsdb.acervus
  where signature is not null
  ;
```

The only two errors during the import of the books data were that the year in which they were published (pub_date) had to be a string of four digits, and has been corrected as seen above. The second error related to this relation was what we hadn't declared the signature as null on the creation of the tables, and therefore we had to stablish is as not null.

4. Loan

For the insertion of data into the loans table, we have inserted the following code into sql:

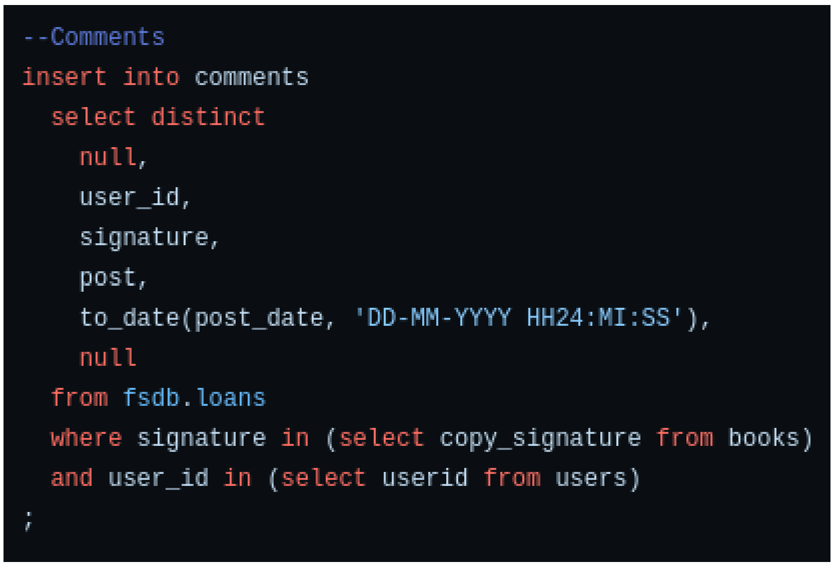


```
--Loans
insert into loan
  select distinct
  null,
  user_id,
  signature,
  to_date(date_time, 'DD-MM-YYYY // HH24:MI:SS'),
  to_date(return, 'DD-MM-YYYY // HH24:MI:SS')
  from fsdb.loans
  where signature in (select copy_signature from books)
  ;
```

The main issues related to the loans relation were basically that the to_date had to be in the format of 'Days-months-Years'. And for the time in 'Hours-Minutes-Seconds'. Then we also had to solve a problem regarding an infinite loop in the select copy_signature.

5. Comments

For the insertion of data into the comments table, we have inserted the following code into sql:

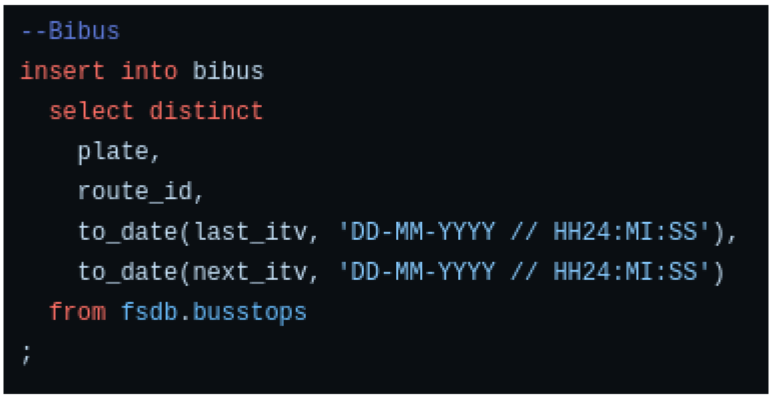


```
--Comments
insert into comments
  select distinct
    null,
    user_id,
    signature,
    post,
    to_date(post_date, 'DD-MM-YYYY HH24:MI:SS'),
    null
  from fsdb.loans
  where signature in (select copy_signature from books)
  and user_id in (select userid from users)
;
```

As the post_date from fsdb.loans follows the pattern 'Days-months-Years Hours-Minutes-Seconds' we had to cast them to date with that format. We also faced a problem when entering the value of copy_signature and userid in the select statements. After noticing and changing those values for the corresponding name of the column of books and users, the table created without issues.

6. Bibus

For the insertion of data into the bibuses table, we have inserted the following code into sql:

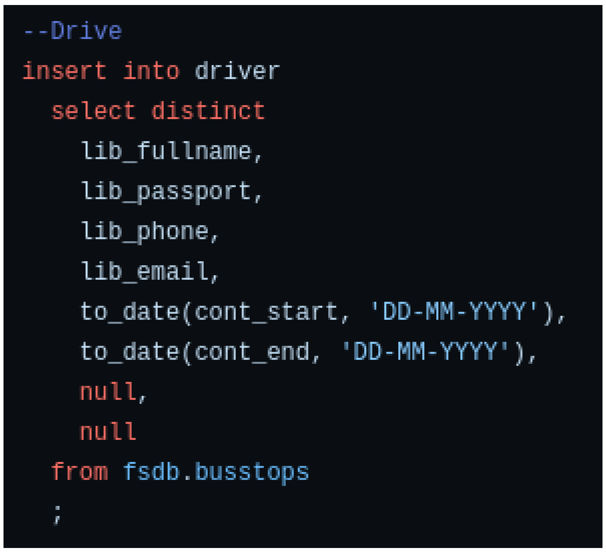


```
--Bibus
insert into bibus
  select distinct
    plate,
    route_id,
    to_date(last_itv, 'DD-MM-YYYY // HH24:MI:SS'),
    to_date(next_itv, 'DD-MM-YYYY // HH24:MI:SS')
  from fsdb.busstops
;
```

With the bibus relation we had a similar problem as with the previous one, we had to restrict the date and time to a format such as 'Days-months-Years Hours-Minutes-Seconds'

7. Driver

For the insertion of data into the drivers table, we have inserted the following code into sql:

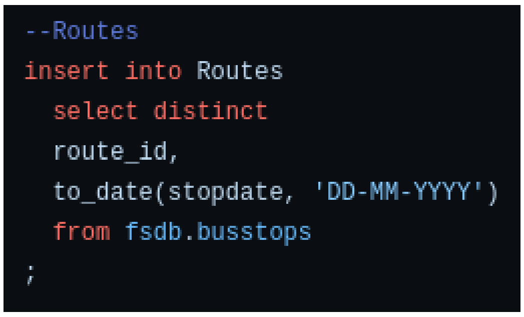


```
--Drive
insert into driver
  select distinct
    lib_fullname,
    lib_passport,
    lib_phone,
    lib_email,
    to_date(cont_start, 'DD-MM-YYYY'),
    to_date(cont_end, 'DD-MM-YYYY'),
    null,
    null
  from fsdb.busstops
  ;
```

Also, when importing into the driver relation we had the same problem as with the previous one, we had to restrict the date and time to a format such as 'Days-months-Years Hours-Minutes-Seconds'

8. Routes

For the insertion of data into the routes table, we have inserted the following code into sql:

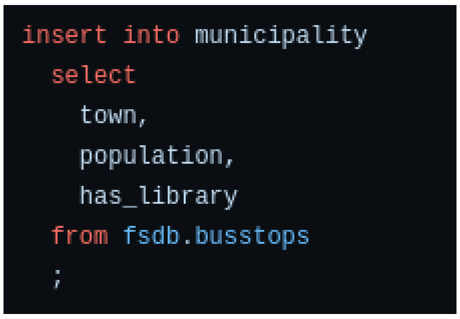


```
--Routes
insert into Routes
  select distinct
  route_id,
  to_date(stopdate, 'DD-MM-YYYY')
  from fsdb.busstops
;
```

As stated previously, we had to restrict to_date to follow 'Days-months-Years Hours-Minutes-Seconds' in order to ensure that the import was made properly.

9. Municipality

For the insertion of data into the municipalities table, we have inserted the following code into sql:

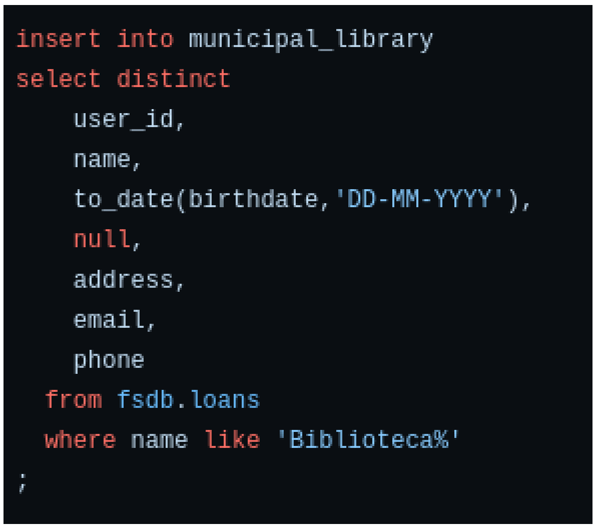


```
insert into municipality
  select
    town,
    population,
    has_library
  from fsdb.busstops
  ;
```

No issues were encountered during the importation of municipality. However, we considered municipal_library as if the municipality has a library through a 'Y' for yes and a 'N' for no

10. Municipal_library

For the insertion of data into the municipal libraries table, we have inserted the following code into sql:

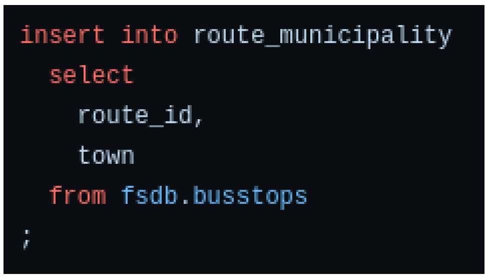


```
insert into municipal_library
select distinct
    user_id,
    name,
    to_date(birthdate,'DD-MM-YYYY'),
    null,
    address,
    email,
    phone
  from fsdb.loans
  where name like 'Biblioteca%'
;
```

We have noticed that there are users in fsdb.loans whose name started by 'Biblioteca…', so we decided to insert into the municipal libraries table all the data needed from fsdb.loans from rows where the name started with the word 'Biblioteca'.

11. Route_municipality

For the insertion of data into the municipal routes table, we have inserted the following code into sql:

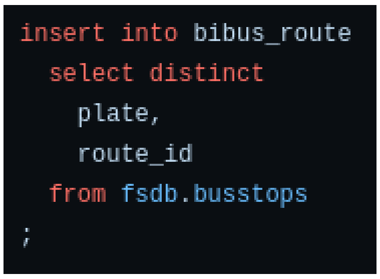


```
insert into route_municipality
  select
    route_id,
    town
  from fsdb.busstops
;
```

We didn't encounter any problem when developing the mass insertion into the route_municipality table, as it was straightforward.

12. Bibus_route

For the insertion of data into the bibuses routes table, we have inserted the following code into sql:



```
insert into bibus_route
  select distinct
    plate,
    route_id
  from fsdb.busstops
;
```

We didn't encounter any problem when developing the mass insertion into the bibus_route table either, just the same case as the last one.

PD: All the insert null that we have implemented were due to non-existence of any column with the data required, so we decided to insert null_value in them. Also, tables sanction and reservation have been left empty due to the same reason.