

INSTITUTO SUPERIOR TÉCNICO - UL



TÉCNICO
LISBOA

PROJECT3-SIBD

DATABASE MODELING - GROUP 30

Authors:

Afonso COSTA

Bárbara SILVA

Gonçalo MESTRE

IST ID:

83986

86950

87005

Professor Bruno MARTINS

Professor Francisco REGATEIRO

6 of December of 2019

Chapter 1

A Web Application Using the Database

1.1 Client Search

At first it was asked to develop a page to search for a client, so it can be used to search for a client when the person comes to a clinic. Here the page was developed assuming that the information introduced is just exclusive, which means that if no information is introduced all the results are showed. As an example when only part of the address is filled only the clients in which the address has no part the resembles the filled address are excluded.

The address of the developed page to search for the clients is

<http://web.ist.utl.pt/ist187005/searchclient.php> and in Figure 1.1 is shown the page.



The screenshot shows a web browser window with the address bar displaying "Inseguro | web.ist.utl.pt/ist187005/searchclient.php". Below the address bar, the page title is "Search for a client". The form contains three input fields: "VAT number:", "Name:", and "Address:". Each field is followed by a text input box. At the bottom of the form is a "Submit" button.

Figure 1.1: Web Form for Searching Clients

For this two tests were made, a search for "City" on the address which has, as a result, the information shown in Figure 1.2. and a search with no introduced information, which has, as result, the information shown in Figures 1.3 and 1.4. The URL of this page with the results is **http:**

//web.ist.utl.pt/ist187005/showclient.php

← → ↻ ⓘ Inseguro | web.ist.utl.pt/ist187005/showclient.php

Clients found:

VAT	name	birth_date	street	city	zip	gender	age	Make an Appointment
977444855	Samuel Mcneil	1938-12-15	Melville Quadrant	Salt Lake City	8962-501	M	81	Insert New Appointment
977444905	Dustin Odom	1903-09-22	Smithy Retreat	Salt Lake City	4300-824	M	116	Insert New Appointment
977444927	Vivien Fraizer	1950-12-27	Old Rectory Hollies	Salt Lake City	5511-574	F	69	Insert New Appointment

[Insert New Client](#)
[Go to the Previous Page](#)

Figure 1.2: Results of client search with City as the address

← → ↻ ⓘ Inseguro | web.ist.utl.pt/ist187005/showclient.php

Clients found:

VAT	name	birth_date	street	city	zip	gender	age	Make an Appointment
977444858	Jaida Chase	1902-10-01	Durham Cross	Temecula	8295-433	F	117	Insert New Appointment
977444859	Mckenna Medina	1973-02-01	Durham Cross	Temecula	9241-985	M	46	Insert New Appointment
977444860	Jaylynn Nichols	2006-01-27	Greenacres Vale	Silver Spring	5663-789	M	13	Insert New Appointment
977444861	Karter Kennedy	2010-06-05	Broom Banks	Swan Island	5588-899	F	9	Insert New Appointment
977444862	Ayaan Leon	1992-09-13	Mill Race Fold	Swan Island	3059-353	M	27	Insert New Appointment
977444863	Andy Peck	1946-12-17	Mill Race Fold	Oakland	6454-132	M	73	Insert New Appointment
977444864	Demarcus Day	1909-01-14	Spinney Brambles	Nashville	4259-196	F	110	Insert New Appointment
977444865	Samuel Mcneil	1938-12-15	Melville Quadrant	Salt Lake City	8962-501	M	81	Insert New Appointment
977444866	Rylan Reid	2000-02-26	Henley Reach	Canton	8257-786	M	19	Insert New Appointment
977444867	Drake Hoover	1980-11-03	Redhill Hollow	Cape Town	6497-706	F	39	Insert New Appointment
977444868	Zachery Howard	1913-01-20	Dove Heath	Lawrence	4972-460	M	106	Insert New Appointment
977444869	Desirae Hill	1987-06-09	Redhill Hollow	Lafayette	6063-434	M	32	Insert New Appointment
977444870	Layton Mcguire	1936-09-18	Magdalen Gardens	Metairie	1607-316	F	83	Insert New Appointment
977444871	Kassandra Travis	1998-07-09	Broom Banks	Moreno Valley	8464-986	M	21	Insert New Appointment
977444872	Brenton Mejia	1924-04-15	Wesley Circle	Wilmington	8063-471	M	95	Insert New Appointment
977444873	Lilian Farmer	2016-07-22	Arden Strand	Lawrence	9821-710	F	3	Insert New Appointment
977444874	Christian Lindsey	1951-10-12	Alexander Lane	Lafayette	3138-454	M	68	Insert New Appointment
977444875	Audrina Santiago	1995-06-18	Christine Avenue	London	9892-834	M	24	Insert New Appointment
977444876	Madalynn Walker	1959-01-10	Greenacres Vale	Cape Town	9039-815	F	60	Insert New Appointment
977444877	Harmony Kaufman	1970-05-05	Anglesey Bank	Oakland	3121-335	M	49	Insert New Appointment
977444878	Cheyenne Yang	2017-03-26	Common Glade	Beijing	4841-871	M	2	Insert New Appointment
977444879	Ashley Murray	1920-07-20	Mill Race Fold	Verdi	8800-183	F	99	Insert New Appointment
977444880	Melvin Ibarra	1930-10-22	Chartwell Garth	Oakland	4404-374	M	89	Insert New Appointment
977444881	Isiah Kemp	1900-08-28	Magdalen Gardens	Thousand Oaks	9030-747	M	119	Insert New Appointment
977444882	Caylee Ashley	1900-09-03	Fleming Oval	Canton	7261-341	F	119	Insert New Appointment
977444883	Jay McKenzie	1956-11-09	Alexander Lane	Cape Town	1333-407	M	63	Insert New Appointment
977444884	Priscilla Hendricks	1956-10-22	Chartwell Garth	Sydney	3751-133	M	63	Insert New Appointment
977444885	Lina Huynh	1914-07-07	Bedford Grove	Belmopan	9099-280	F	105	Insert New Appointment

Figure 1.3: Results of client search with no parameters

977444886	Dorian Carter	1958-02-14	Fleming Oval	Black Mtn	2508-424	M	61	Insert New Appointment
977444887	Brandon Castillo	1990-05-21	Melville Quadrant	London	4318-366	M	29	Insert New Appointment
977444888	Frances Parrish	2011-11-22	Mill Race Fold	Thousand Oaks	2064-559	F	8	Insert New Appointment
977444889	Yadira Weaver	1922-06-25	Tennyson Moor	Beijing	7683-436	M	97	Insert New Appointment
977444890	Elle Sandoval	1914-11-16	Smithy Retreat	Oakland	9915-236	M	105	Insert New Appointment
977444891	Mauricio Guerra	1925-07-26	Magdalen Gardens	Swan Island	6578-475	F	94	Insert New Appointment
977444892	Dashawn Neal	1940-08-15	Beechcroft Corner	Vientiane	1404-358	M	79	Insert New Appointment
977444893	Izayah Hayden	1916-09-06	Old Rectory Hollies	Evansville	1097-933	M	103	Insert New Appointment
977444894	Dante Fitzpatrick	1913-08-01	Arden Strand	Black Mtn	2983-124	F	106	Insert New Appointment
977444895	Nathalie Bender	1980-06-25	Spinney Brambles	Black Mtn	3171-482	M	39	Insert New Appointment
977444896	Darius Prince	1974-06-13	Feltham Road	Amarillo	1465-403	M	45	Insert New Appointment
977444897	Gina Bradley	1997-08-19	Durham Cross	Beijing	8187-486	F	22	Insert New Appointment
977444898	Selina Higgins	1925-01-06	Dove Heath	Vienna	9918-360	M	94	Insert New Appointment
977444899	Joe Hinton	1939-10-26	Broughton Green	Amarillo	9820-831	M	80	Insert New Appointment
977444900	Giselle Conner	1961-06-25	Old Rectory Hollies	Beijing	1971-634	F	58	Insert New Appointment
977444901	Julia Petty	1985-06-10	Broughton Green	Wilmington	3377-367	M	34	Insert New Appointment
977444902	Alec Bird	1900-08-26	Beechcroft Corner	Dorchester	9554-537	M	119	Insert New Appointment
977444903	Brielle Young	2008-12-21	Redhill Hollow	Cape Town	6376-278	F	11	Insert New Appointment
977444904	Malakai Larson	1988-04-19	Beechcroft Corner	Black Mtn	2337-747	M	31	Insert New Appointment
977444905	Dustin Odom	1903-09-22	Smithy Retreat	Salt Lake City	4300-824	M	116	Insert New Appointment
977444906	Leland Carey	1954-01-15	Smithy Retreat	Pocatello	3659-504	F	65	Insert New Appointment
977444907	Davis Moon	1904-02-14	Chartwell Garth	Peterson Afb	9737-744	M	115	Insert New Appointment
977444908	Luciana Robbins	1914-07-24	Dickens Las	Wilmington	6383-265	M	105	Insert New Appointment
977444909	Yamilet Avila	1911-10-21	Chartwell Garth	Pocatello	7756-701	F	108	Insert New Appointment
977444910	Kendra Chase	2007-05-19	Alexander Lane	Murfreesboro	2191-922	M	12	Insert New Appointment
977444911	Braeden Brock	1902-07-19	Durham Cross	Dorchester	3585-744	M	117	Insert New Appointment
977444912	Luis Nunez	1903-08-26	Anglesey Bank	London	3085-974	F	116	Insert New Appointment
977444913	Lydia Cain	1973-09-05	Old Rectory Hollies	Estersnow	8872-338	M	46	Insert New Appointment
977444914	Dangelo Dennis	2006-06-12	Melville Quadrant	Cape Town	4559-248	M	13	Insert New Appointment
977444915	Brayden Figueroa	1972-10-03	Melville Quadrant	Murfreesboro	2083-761	F	47	Insert New Appointment
977444916	Lailah Delacruz	2002-10-17	Smithy Retreat	East Orange	3413-853	M	17	Insert New Appointment
977444917	Brisa Ball	1947-09-01	Severn Circus	East Orange	5387-670	M	72	Insert New Appointment
977444918	Chanel Stout	1973-11-19	Dickens Las	Swan Island	2793-225	F	46	Insert New Appointment
977444919	Peter Jaillet	1953-12-11	Broughton Green	Amarillo	1638-257	M	66	Insert New Appointment
977444920	Jane Sweettooth	1953-03-23	Bedford Grove	Basseterre	8751-808	M	66	Insert New Appointment
977444921	Latasha Kaiser	1944-06-01	Fleming Oval	Black Mtn	4311-446	F	75	Insert New Appointment
977444922	Lisabeth Stutzman	1911-05-16	Greenacres Vale	Murfreesboro	6966-544	M	108	Insert New Appointment
977444923	Marianela Berning	1938-07-15	Willowhayne Gardens	Dorchester	8427-417	M	81	Insert New Appointment
977444924	Monet Todd	1932-09-28	Willowhayne Gardens	London	5436-391	F	87	Insert New Appointment
977444925	Laura Creager	2000-06-03	Dove Heath	Murfreesboro	4654-574	M	19	Insert New Appointment
977444926	Melaine Kuchera	1921-10-08	Magdalen Gardens	Moscow	6406-647	M	98	Insert New Appointment
977444927	Vivien Fraizer	1950-12-27	Old Rectory Hollies	Salt Lake City	5511-574	F	69	Insert New Appointment
977444928	Goncalo Mestre	1998-10-03	IST	Lisbon	2355-563	M	21	Insert New Appointment
977444929	Afonso Costa	1997-09-01	IST	Lisbon	2742-456	M	22	Insert New Appointment

[Insert New Client](#)

[Go to the Previous Page](#)

Figure 1.4: Rest of the results of client search with no parameters

Using Figure, 1.2, as an example to analyse this web page, here it is possible to see that this allows an user of this web pages to access multiple functionalities from here. So, starting with, one can see

that it is possible to select the option to insert a new client and go to a web page to do it. It is also possible to select the option to insert a new appointment for the client in the same row, which takes the user to the corresponding web page with a form. As other functionalities there is the possibility of selecting the VAT of a client, which takes the user to a page with the appointments of that client, and as a last one it is also possible to select at the end of the page the option that takes the user the first web page so it is possible to search for clients again.

1.2 Client Insertion

As a functionality the set of web pages also needed to implement client insertion on the database, so as following the page shown on Figure 1.2 and on selecting "Insert New Client" the user is taken to a web page with the form to add a new client as shown in Figure 1.5. This web page has the URL `http://web.ist.utl.pt/ist187005/newclient.php`.

← → ↻ ⓘ Inseguro | web.ist.utl.pt/ist187005/newclient.php

Insert a new client

VAT number:

Name:

Birth Date:

Street:

City:

Zip Code Part1:

Zip Code Part2:

Gender: ☒ Male ☐ Female ☐ Other

Age:

[Go to the First Page](#)

Figure 1.5: Web page with the form to introduce the data to insert a new client

It is to note that this page has the option to go back to the first page and search for clients so it is possible to check before introducing a client.

After filling the form with the information shown in Figure 1.5, the user is redirected to another page as shown in Figure 1.6. This new page has the URL `http://web.ist.utl.pt/ist187005/insertclient.php`.

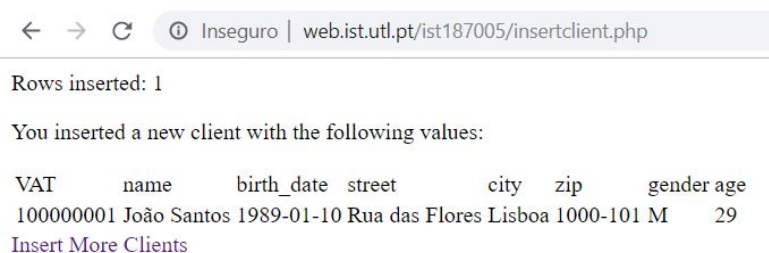


Figure 1.6: Web page with the results of the insertion of a client

It is to note that on the end of this page there is an option to "Insert More Clients" and go to the form used before.

After this insertion it was confirmed using the web application, as shown in Figures 1.7 and 1.8.



Figure 1.7: Web page with the search for the inserted client

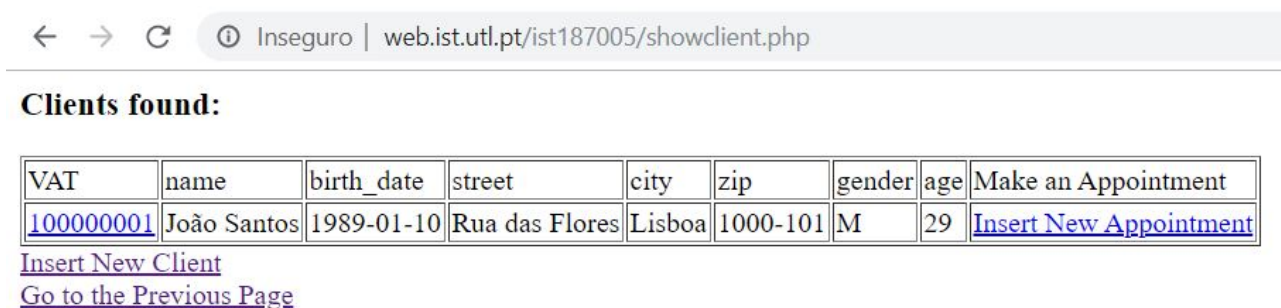


Figure 1.8: Web page with the results of the search for the inserted client

1.3 Appointment Insertion

On the first set of web pages it was asked to also include pages for the client to introduce an appointment for each client. So, following the page where the results for a search are shown like in Figure 1.8 and selecting the only client on this figure the user is redirected to a page with a form to choose the time for the appointment, as shown in Figure 1.9.

The URL for this web page is `http://web.ist.utl.pt/ist187005/newappointment.php?cvat=100000001`.

It is of relevance that in the URL there is the VAT of the client for which an appointment is being scheduled, being this the way that the variable is passed to this page. It is also important that in the end of the page there is an option to go to the first page and search for clients again.

Here it was a point where it was needed to take a non-trivial decision and was decided to have this first page for the user to select the time, and then have a second page where the available doctors are listed and then the desired one is selected by the user



← → ↻ ▲ Inseguro | web.ist.utl.pt/ist187005/newappointment.php?cvat=100000001

Select an hour and day for the appointment

Date:

Hour:

[Go to the First Page](#)

Figure 1.9: Web page with the form to select the time for a new appointment

On Figures 1.10 and 1.11 is shown the page where the doctors are listed and available for selection, being the client vat and the time for the appointment sent via the form on the last page. It is to note that to send the client vat it was used the input type "hidden".

The screenshot shows a web browser window with the address bar displaying 'Inseguro | web.ist.utl.pt/ist187005/showdoctors.php'. The page title is 'The available doctors for that time are'. Below the title, there is a dropdown menu for 'Name and Vat:' with the selected option '977442106 Vivien Fraizer'. To the right of the dropdown is a text box for 'Description:'. Below the text box is a 'Submit' button. At the bottom of the form is a link 'Go to the Previous Page'.

Figure 1.10: Web page with the listed doctors for selection and the text box to introduce the description for the appointment

This screenshot shows the same web page as Figure 1.10, but the dropdown menu for 'Name and Vat:' is open, displaying a list of 15 doctors. The first option, '977442106 Vivien Fraizer', is highlighted in blue. The list of doctors is as follows:

- 977442106 Vivien Fraizer
- 977442107 Cristen Aultman
- 977442108 Jane Sweettooth
- 977442109 Milo Chagolla
- 977442110 Nestor Dinkel
- 977442111 Waylon Glisson
- 977442112 Luke Ureno
- 977442113 Stuart Mcquire
- 977442114 Tyree Reaper
- 977442115 Chery Zullo
- 977442116 Caryn Kingsberry
- 977442117 Cierra Worth
- 977442118 Ada Carmody
- 977442119 Rosio Stefanski
- 977442120 Mario Mancino
- 977442121 Esther Spiker

The 'Description:' text box and the 'Submit' button are still visible to the left of the dropdown menu. The 'Go to the Previous Page' link is also present at the bottom left.

Figure 1.11: Same web Page as in Figure 1.10, but now showing the listed doctors

In the end of the page there is also an option to go to the previous page so that another time can be selected if the desired doctor is not available. After submitting the information the user is redirected to a page showing the results of this insertion. This page shown in Figure 1.12 has an option in the end to insert more appointments that redirects the user to the web page to select the time of the appointment. It also has the URL, <http://web.ist.utl.pt/ist187005/insertappointment.php>.



Figure 1.12: Web page with the results of the appointment insertion

To test if the list of the doctors was correct, it was then checked using the same time for the appointment the listed doctors, and was obtained the Figure 1.13 which shows that the listing is well done because the doctor Vivien Frazer with VAT 977442106 is no longer available for that time.

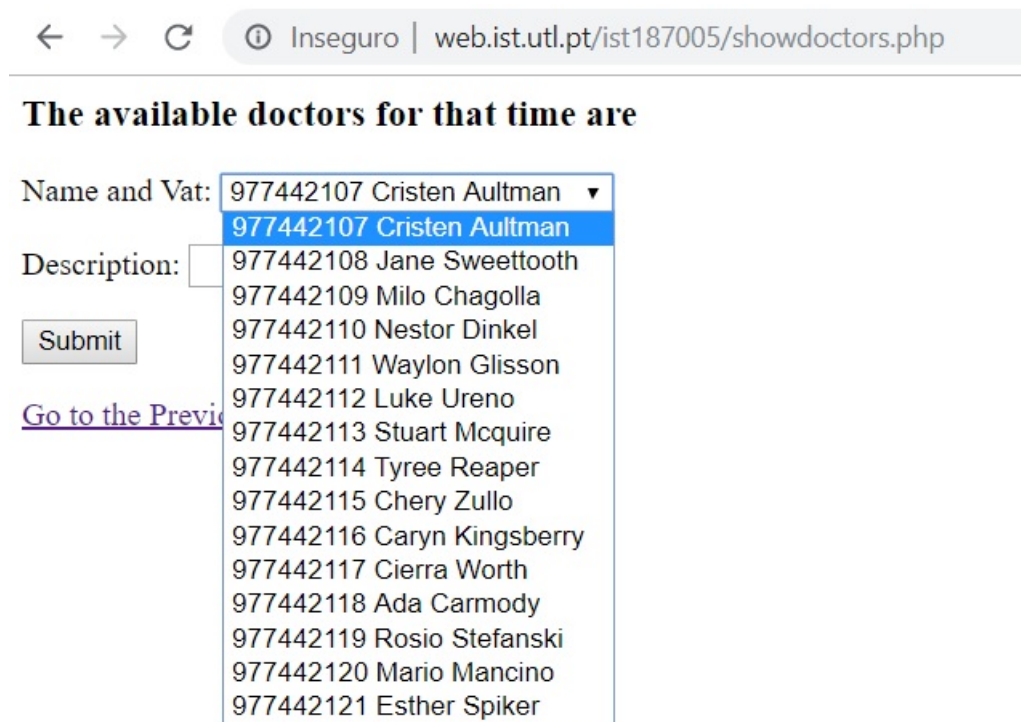


Figure 1.13: Web page with the listed doctors after the insertion

1.4 Search Appointments

Here it was also asked to make available the search for appointments, so as an example it was searched for the introduced client "João Santos" with VAT "100000001". The result of this search is shown in the web page in Figure 1.14, then pressing the VAT the user is redirected to the page in Figure 1.15 which has the URL <http://web.ist.utl.pt/ist187005/showappointment.php?cvat=100000001>.

Note here that the vat of the client for which the appointments are being listed is being passed by the URL.

Clients found:

VAT	name	birth_date	street	city	zip	gender	age	Make an Appointment
100000001	João Santos	1989-01-10	Rua das Flores	Lisboa	1000-101	M	29	Insert New Appointment

[Insert New Client](#)
[Go to the Previous Page](#)

Figure 1.14: Web page with the result of the search for the referred client

Appointments found for 100000001:

VAT_doctor	date_timestamp	description	Check this Appointment
977442106	2019-12-10 10:00:00	He wants to clean left side of mouth	Check this Appointment

[Go to the First Page](#)

Figure 1.15: Web page with appointments found for client with VAT 100000001

In this page it is possible to see the appointment that was introduced earlier and it's information. It is possible to check that there is an option to redirect the user to the first page and search for clients.

Here there is also the option to "Check this Appointment" for every appointment, which allows the user to check the information for the consultation corresponding to a specific appointment. Pressing this option redirects the user to a web page shown in Figure 1.16 which has the information for the corresponding consultation. There are also options to insert a dental charting procedure, which will be talked in depth later, an option to insert more data for the consultation and an option to go to the previous page to see the listed appointments and check for any other consultation. This web page URL is <http://web.ist.utl.pt/ist187005/showconsultation.php?apccvat=100000001&appdvat=977442106&appdate=2019-12-10%2010:00:00>.

Here it is to note that the vat of the client is still being passed by the URL so that the user can go to the previous page and that the primary keys of the consultation (the Vat of the doctor and the date

of it), are also passed by the URL. It is also of notice that the information shows that there was no consultation for the selected appointment, because only the appointment was inserted, being it still necessary to insert the consultation with it's information.



Figure 1.16: Web page with the consultation information for the appointment introduced on client 100000001

1.5 Insert Data for Consultation

It was also asked to have a set of web pages to allow the insertion of information for a consultation, so to implement that, following the option to "Insert More Data for this Consultation" on the web page in Figure 1.16, the user is redirected to the web page shown in Figure 1.17. This web page has the URL `http://web.ist.utl.pt/ist187005/newdata.php?appcvat=100000001&appdvat=977442106&appdate=2019-12-10%2010:00:00`.

Note that the here the information regarding the client and the primary key of the consultation are being passed through the URL. There is also an option to go to the previous page.

[←](#) [→](#) [↺](#) [Inseguro](#) | [web.ist.utl.pt/ist187005/newdata.php?appcvat=100000001&appdvat=977442106&appdate=2019-12-10%2010:00:00](#)

Insert a nurse for this consultation

Name and Vat:

Introduce data for the consultation corresponding to this appointment

Soap Notes - Subjective:

Soap Notes - Objective:

Soap Notes - Assessment:

Soap Notes - Plan:

Insert a diagnostic code for this consultation

Diagnostic Code:

Insert a prescription for this consultation

Diagnostic code to cure:

Medication Name:

Dosage:

Description:

[Go to the Previous Page](#)

Figure 1.17: Web page with the form to introduce information to a consultation

This page has 4 forms, one for each type of data there can introduced regarding a consultation, each form redirects the user to a different web page with the result of the information insertion.

Here it was decided that the insertion would not be allowed on any other camp before there is at least one nurse, because it was mandatory that every consultation had a nurse. It was not allowed also the insertion of a prescription if there was no associated diagnostic code with the referred consultation.

It will be shown now the output of the introduction for each type of data. Note that all of this 4 web pages have an option in the end to go back and introduce more data to the consultation. So, starting with nurses, the output of an introduction is the web page shown in Figure 1.18 and has the URL <http://web.ist.utl.pt/ist187005/insertnurse.php> Note that it is this web page that implements the insertion of the consultation when there was only an appointment, this was the chosen page to do that, because it is mandatory that every consultation had a nurse. So, there could not be a consultation with a nurse associated and there could also not be a nurse associated without a real consultation.

For the soap notes the output is the web page shown in Figure 1.19 and has the URL `http://web.ist.utl.pt/ist187005/insertsoap.php`

As for the diagnostic code the output is the web page shown in Figure 1.20 and has the URL `http://web.ist.utl.pt/ist187005/insertdiagnostic.php`

For last there is the prescription with the output being the web page shown in Figure 1.21 with the URL `http://web.ist.utl.pt/ist187005/insertprescription.php`



Figure 1.18: Web page with the result of the insertion of a nurse

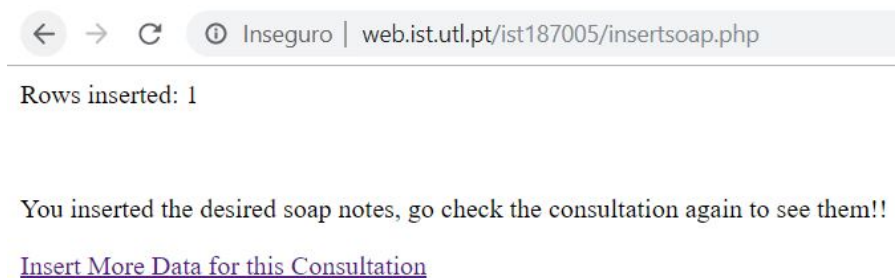


Figure 1.19: Web page with the result of the insertion of soap notes

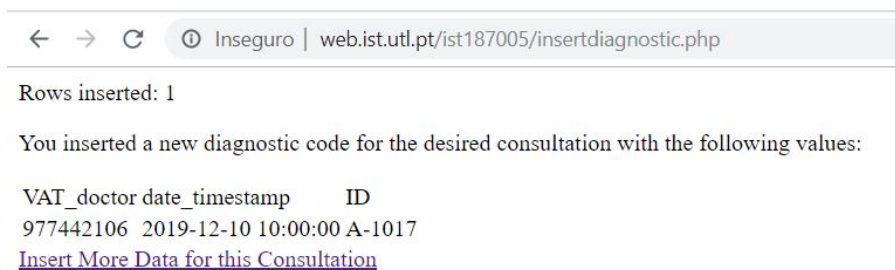


Figure 1.20: Web page with the result of the insertion of a diagnostic code

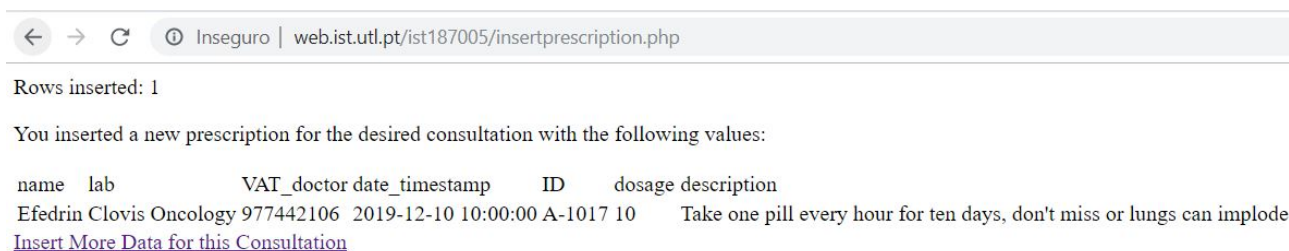


Figure 1.21: Web page with the result of the insertion of a prescription

To confirm this all, it is checked the consultation again with the result shown in Figure 1.22 that shows all the information added in the form to this consultation and confirms what was expected.

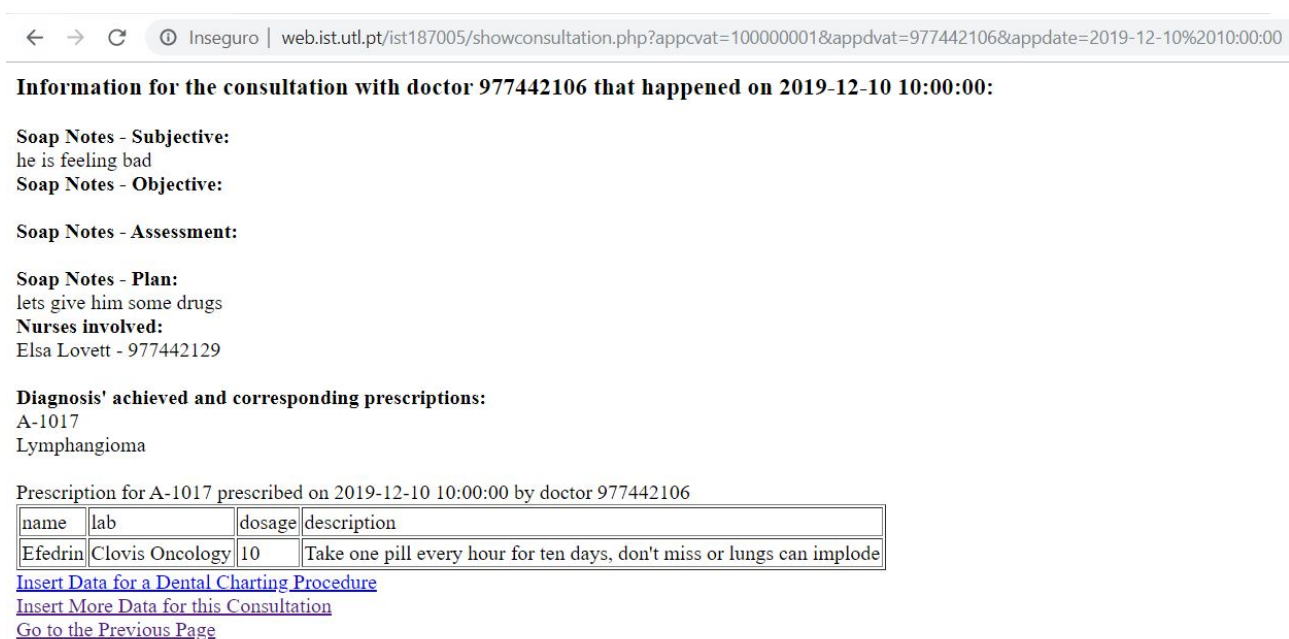


Figure 1.22: Web page with the confirmation of the insertion of data in the consultation

1.6 Insert Dental Charting

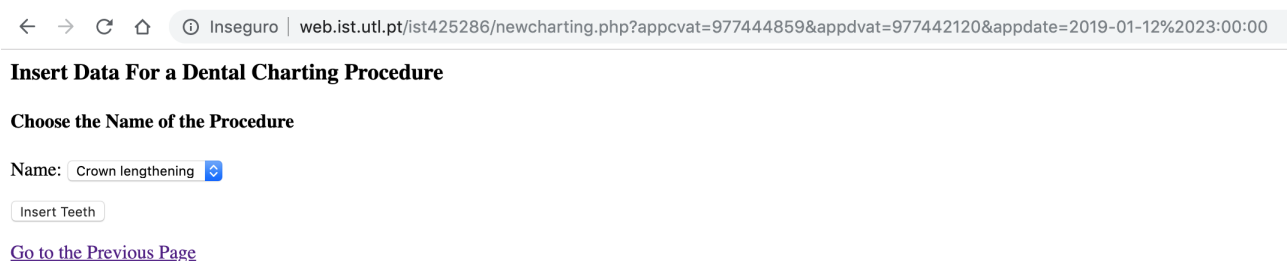
As it is showed in Figure 1.16, there is an option that allows the user to insert a dental charting procedure, and for that propose there were created tree new pages:

- *newcharting.php*: Allows the user to choose the name of the dental charting procedure, between those that are available in proceduretable table (fig.1.23);
 - URL: `http://web.ist.utl.pt/ist425286/newcharting.php?appcvat=977444859&appdvat=977442120&appdate=2019-01-12%2023:00:00`
- *newteeth.php*: Allows the user: to choose the teeth of the procedure according to *Palmer Notation*

Numbering System, to insert a measure for the gap between the tooth and the gums and to insert a description about the procedure (fig.1.24);

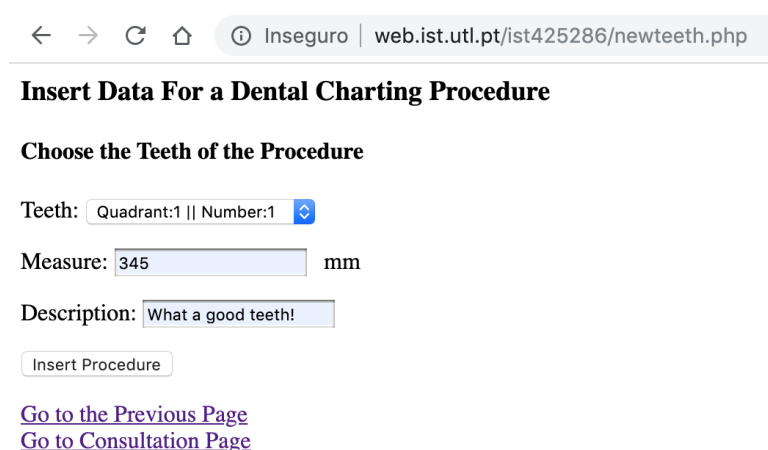
- URL: <http://web.ist.utl.pt/ist425286/newteeth.php>
- *insertcharting.php*: Gives information to user about the new dental charting procedure inserted (fig.1.25).
 - URL: <http://web.ist.utl.pt/ist425286/insertcharting.php>

The first two pages (fig.1.23 and fig.1.24) were made to collect the data to insert (fig.1.25), and both of them have some restrictions. The *newcharting.php* offers a picklist with all the possible names for the Dental Charting Procedure available in the proceduretable, which is a table that stores all the kind of procedures that can be done at the clinic. The *newteeth.php* allows the user to pick a teeth between all the available, that hasn't had done a procedure with the same name (chosen in the previous page), in the same consultation.



The screenshot shows a web browser window with the URL web.ist.utl.pt/ist425286/newcharting.php?appcvat=977444859&appdvat=977442120&appdate=2019-01-12%2023:00:00. The page title is "Insert Data For a Dental Charting Procedure". Below the title, there is a section "Choose the Name of the Procedure". A label "Name:" is followed by a dropdown menu showing "Crown lengthening". Below the dropdown is a button labeled "Insert Teeth". At the bottom, there is a link "Go to the Previous Page".

Figure 1.23: Web page with the form to pick the name of the dental charting procedure



The screenshot shows a web browser window with the URL web.ist.utl.pt/ist425286/newteeth.php. The page title is "Insert Data For a Dental Charting Procedure". Below the title, there is a section "Choose the Teeth of the Procedure". A label "Teeth:" is followed by a dropdown menu showing "Quadrant:1 || Number:1". Below the dropdown is a text input field labeled "Measure:" containing the value "345", followed by the unit "mm". Below the input field is a label "Description:" followed by a text input field containing the value "What a good teeth!". Below the input field is a button labeled "Insert Procedure". At the bottom, there are two links: "Go to the Previous Page" and "Go to Consultation Page".

Figure 1.24: Web page with the form to introduce information about the teeth for the dental charting procedure

This last page is where the all the previously inserted information about the dental charting procedure, will be stored in the database. Starting by checking. If there is any consultation created, then it checks if there is any procedure in consultation for that consultation, if there isn't it will be automatically created, and then checking if there was already created a dental charting procedure for that consultation, that teeth and that name the user is redirected to the page it will be shown a table with all the inserted values, and an option to go back and insert more data for a Dental Charting procedure.

← → ↻ 🏠 ⓘ Inseguro | web.ist.utl.pt/ist425286/insertcharting.php

You inserted a new dental charting procedure for the desired consultation with the following values:

name	VAT_doctor	date_timestamp	quadrant	number_	description	measure
Crown lengthening	977442120	2019-01-12 23:00:00	1	1	What a good teeth!	345

[Insert More Data for a Dental Charting Procedure](#)

Figure 1.25: Web page with the result of the insertion of a dental charting procedure

Chapter 2

Functions, Triggers and Stored Procedures

2.1 Trigger 1

Here it was asked to create a trigger that updates the age of the clients according to the birth date and the current date, whenever a new appointment for that client is introduced on the database.

For each new appointment arriving to the data base, the client's age is set to the difference (in years) between the current date and is birth date where. The VAT_Client on the new appointment and the VAT from every client are compared to choose the one that the appointment belongs to and do the update.

```
1 CREATE TRIGGER Update_Age AFTER INSERT ON appointment
2     FOR EACH ROW
3     BEGIN
4     UPDATE client
5     SET AGE = TIMESTAMPDIFF(YEAR, birth_date, CURDATE())
6     WHERE client.VAT = NEW.VAT_client;
7 END$$
```

Listing 2.1: *Update_Age* Trigger

The trigger was previously tested as we can see in Figure 2.1, where was inserted a new client with the wrong age, then it was also inserted a new appointment for the client, and after it the age of the client was already correct.

```

MySQL [ist425286]>
MySQL [ist425286]> INSERT INTO client VALUES (988444861,'Trigger_1', '1902-10-1','Durham Cross', 'Temecula', '8295-433', 'F', 9);
Query OK, 1 row affected (0.00 sec)

MySQL [ist425286]> SELECT * FROM client WHERE VAT=988444861;
+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT      | name      | birth_date | street      | city      | zip      | gender | age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 988444861 | Trigger_1 | 1902-10-01 | Durham Cross | Temecula  | 8295-433 | F      | 9  |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

MySQL [ist425286]> INSERT INTO appointment VALUES (977442106, '2020-1-4 7:00:00', 'Teeth hurt a lot', 988444861);
Query OK, 1 row affected (0.01 sec)

MySQL [ist425286]> SELECT * FROM client WHERE VAT=988444861;
+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT      | name      | birth_date | street      | city      | zip      | gender | age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 988444861 | Trigger_1 | 1902-10-01 | Durham Cross | Temecula  | 8295-433 | F      | 117 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

```

Figure 2.1: Test made to Trigger 1

2.2 Trigger 2

It was asked to write triggers to ensure that:

- (a) An individual that is a receptionist or a nurse at the clinic cannot simultaneously be a doctor;
- (b) Doctors cannot simultaneously be trainees and permanent doctors.

2.2.1 Trigger 2 - a)

For the propose of the first instruction there were created 6 triggers:

- *Doctor_Cannot_Be_Receptionist_or_Nurse_I*: To ensure that when is inserted a Doctor, there isn't any Receptionist or Nurse with the same VAT;
- *Doctor_Cannot_Be_Receptionist_or_Nurse_U*: To ensure that doctors can't update their VAT to one that corresponds already a to a Nurse or a Receptionist;
- *Nurse_Cannot_Be_Doctor_I*: To ensure that when a nurse is inserted, there isn't any Doctor with that VAT;
- *Nurse_Cannot_Be_Doctor_U*: To ensure that nurses can't update their VAT to one that is already being used by a Doctor;
- *Receptionist_Cannot_Be_Doctor_I*: To ensure that when a receptionist is inserted, there isn't any Doctor with that VAT;
- *Receptionist_Cannot_Be_Doctor_U*: To ensure that receptionists can't update their VAT to one that is already being used by a Doctor;

As the triggers that fire with an insert are really similar, it is only explained the one that is more complex, *Doctor_Cannot_Be_Receptionist_or_Nurse_I*, the same occurs with the triggers that fire with an update, so is also compared with the *Doctor_Cannot_Be_Receptionist_or_Nurse_U* trigger.

The insert triggers fire before a insert so, for each insertion, it verifies if there is any equal VAT to the inserted VAT, if there is and the equal VAT that corresponds in any way to doctor and nurse, or doctor and receptionist, it will stop the insertion and will send a message with the correspondent employee job.

```

1 DROP TRIGGER IF EXISTS Doctor_Cannot_Be_receptionist_or_Nurse_I;
2 DELIMITER $$
3 CREATE TRIGGER Doctor_Cannot_Be_receptionist_or_Nurse_I BEFORE INSERT ON doctor
4     FOR EACH ROW
5     BEGIN
6         IF (EXISTS (SELECT nurse.VAT FROM nurse WHERE nurse.VAT = NEW.VAT))
7             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
            is an Nurse';
8         END IF;
9         IF (EXISTS (SELECT receptionist.VAT FROM receptionist WHERE receptionist
            .VAT = NEW.VAT))
10            THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
            is an Receptionist';
11        END IF;
12 END$$
13 DELIMITER ;

```

Listing 2.2: *Doctor_Cannot_Be_Receptionist_or_Nurse_I* trigger

As we can see in Figure 2.2, when was inserted a new doctor with a VAT that corresponded to a nurse/receptionist occurred an error that said: "This employee is a Nurse/Receptionist", then when was inserted an employee without a job in the database the trigger didn't fire, in the end of the test it's possible to see that only the employee that wasn't a nurse or a receptionist, was inserted into the Doctor.

```

MySQL [ist425286]> INSERT INTO doctor VALUES (998442200, 'Orthodontist', 'Description.', 'email20@denesbrilhantes.com.br');
ERROR 1644 (45000): This employee is an Nurse
MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN nurse WHERE VAT=998442200;
+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT | name | birth_date | street | city | zip | IBAN | salary |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 998442200 | Trigger_2_nurse | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 20 | 2000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

MySQL [ist425286]> INSERT INTO doctor VALUES (998442201, 'Orthodontist', 'Description.', 'email21@denesbrilhantes.com.br');
ERROR 1644 (45000): This employee is an Receptionist
MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN receptionist WHERE VAT=998442201;
+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT | name | birth_date | street | city | zip | IBAN | salary |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 998442201 | Trigger_2_recepcionist | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 21 | 2000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

MySQL [ist425286]> INSERT INTO doctor VALUES (998442202, 'Orthodontist', 'Description.', 'email22@denesbrilhantes.com.br');
Query OK, 1 row affected (0.00 sec)

MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202 OR VAT=998442201 OR VAT=998442200;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT | name | birth_date | street | city | zip | IBAN | salary | specialization | biography |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 998442202 | Trigger_2_doctor | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 22 | 2000 | Orthodontist | Description. |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Figure 2.2: Test made to *Doctor_Cannot_Be_Receptionist_or_Nurse_I* Trigger

As it is possible to see in *Doctor_Cannot_Be_Receptionist_or_Nurse_U*, the only difference between the insert triggers and the update ones is the error message which, in this case, says "Cannot UPDATE VAT from a Doctor. Should first DELETE or UPDATE the Nurse then INSERT the NEW VAT INTO Doctor", which presents an alternative way to accomplish the intended.

```

1 DROP TRIGGER IF EXISTS Doctor_Cannot_Be_receptionist_or_Nurse_U;
2 DELIMITER $$
3 CREATE TRIGGER Doctor_Cannot_Be_receptionist_or_Nurse_U BEFORE UPDATE ON doctor
4     FOR EACH ROW
5     BEGIN
6         IF( EXISTS (SELECT nurse.VAT FROM nurse WHERE nurse.VAT = NEW.VAT))
7             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
8                 VAT from a Doctor. Should first DELETE or UPDATE the Nurse then INSERT the
9                 NEW VAT INTO Doctor.';
10            END IF;
11            IF( EXISTS (SELECT receptionist.VAT FROM receptionist WHERE receptionist
12                .VAT = NEW.VAT))
13                THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
14                    VAT from a Doctor. Should first DELETE or UPDATE the Nurse then INSERT the
15                    NEW VAT INTO Doctor.';
16            END IF;
17 END$$

```


13 DELIMITER ;

Listing 2.3: *Doctor_Cannot_Be_Receptionist_or_Nurse_U* trigger

2.2.2 Trigger 2 - b)

To accomplish this second instruction, were created 4 triggers:

- *PermanentDoctor_Cannot_Be_TraineeDoctor_I*: To ensure that when is inserted a permanent doctor and there is already a trainee doctor with the inserted VAT, then the trainee becomes a permanent doctor;
- *PermanentDoctor_Cannot_Be_TraineeDoctor_U*: To ensure that permanent doctors can't update their VAT to one that corresponds already to a trainee;
- *TraineeDoctor_Cannot_Be_PermanentDoctor_I*: To ensure that when a trainee doctor is inserted, there isn't any permanent doctor with that VAT;
- *TraineeDoctor_Cannot_Be_PermanentDoctor_U*: To ensure that trainees can't update their VAT if it is equal to one that is already being used by a permanent doctor.

The last two triggers, (*PermanentDoctor_Cannot_Be_TraineeDoctor_I* and *PermanentDoctor_Cannot_Be_TraineeDoctor_U*) work exactly the same way as the triggers made in trigger 2-a) section, with the only difference that instead of being about nurse and doctor is about trainee doctor and permanent doctor. For this reason both of this triggers aren't shown here.

The trigger *PermanentDoctor_Cannot_Be_TraineeDoctor_I* fires before a permanent doctor is inserted, and, for each one it searches for a trainee doctor with the same VAT, if it finds one, then the trainee (VAT) is erased from the trainee_doctor table so it becomes possible to insert a permanent doctor with the pertained VAT.

```

1 DROP TRIGGER IF EXISTS PermanentDoctor_Cannot_Be_TraineeDoctor_I;
2 DELIMITER $$
3 CREATE TRIGGER PermanentDoctor_Cannot_Be_TraineeDoctor_I BEFORE INSERT ON
  permanent_doctor
4     FOR EACH ROW
5     BEGIN
6         IF (EXISTS (SELECT trainee_doctor.VAT FROM trainee_doctor WHERE
  trainee_doctor.VAT = NEW.VAT))
7             THEN DELETE
8                 FROM trainee_doctor
9                 WHERE trainee_doctor.VAT = NEW.VAT;

```

```

10         END IF;
11 END$$
12 DELIMITER ;

```

Listing 2.4: *PermanentDoctor_Cannot_Be_TraineeDoctor_I* trigger

All the triggers were previously tested, but to avoid repetitions we will only show the test made to *PermanentDoctor_Cannot_Be_TraineeDoctor_I*. As we can see in figure 2.3, initially there weren't any permanent doctors but there was one trainee. Then when inserted two doctors in the permanent_doctor table (one of them was the one that was in the trainee_doctor table), it is also possible to see that the one that was in the trainee_doctor table is no longer there, but both of them appear in the permanent_doctor table.

```

MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor WHERE VAT=998442303 OR VAT =998442304;
Empty set (0.01 sec)

MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor WHERE VAT=998442303 OR VAT =998442304;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT | name | birth_date | street | city | zip | IBAN | salary | specialization | biography | email | supervisor |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 998442304 | Trigger_2_trainee | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 234 | 2000 | Orthodontist | Description. | email1234@denesbrilhantes.com.br | 977442106 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

MySQL [ist425286]> INSERT INTO permanent_doctor VALUES (998442303, 14);
Query OK, 1 row affected (0.01 sec)

MySQL [ist425286]> INSERT INTO permanent_doctor VALUES (998442304, 3);
Query OK, 1 row affected (0.00 sec)

MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor WHERE VAT=998442303 OR VAT =998442304;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| VAT | name | birth_date | street | city | zip | IBAN | salary | specialization | biography | email | years |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 998442303 | Trigger_2_permanent | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 233 | 2000 | Orthodontist | Description. | email1233@denesbrilhantes.com.br | 14 |
| 998442304 | Trigger_2_trainee | 1991-10-07 | Broom Banks | Estersnow | 4429-551 | 234 | 2000 | Orthodontist | Description. | email1234@denesbrilhantes.com.br | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [ist425286]> SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor WHERE VAT=998442303 OR VAT =998442304;
Empty set (0.01 sec)

```

Figure 2.3: Test made to *PermanentDoctor_Cannot_Be_TraineeDoctor_I* Trigger

2.3 Trigger 3

In this section, triggers were created in order to ensure that different individuals can not have the same phone number. To impose this integrity constraint it is necessary to create 2 triggers **before** the change for each table (*phone_number_employee* and *phone_number_client*): insert and update. If a change is invalid (when the phone number already exists), these triggers cancel it using a signal. The insert and update triggers are very similar to each other, but the update doesn't consider the phone number being changed as a restriction. This means that it is possible to change the phone number "987654321", associated to an employee or client, to the same number "987654321", but it is impossible to insert that number until it is deleted from the database.

```

1 DROP TRIGGER IF EXISTS Phone_employee_I;
2

```

```

3 DELIMITER $$
4
5 CREATE TRIGGER Phone_employee_I BEFORE INSERT ON phone_number_employee
6 FOR EACH ROW
7 BEGIN
8     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
9         phone = NEW.phone))
10         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
11             number is already associated to an employee.';
12     END IF;
13     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
14         phone = NEW.phone))
15         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
16             number is already associated to a client.';
17     END IF;
18 END$$

```

Listing 2.5: Trigger *Phone_employee_I*

To test the trigger for inserting an employee's phone number (*Phone_employee_I*), we tried adding the phone numbers "955831140" and "955831782" to the employee with VAT = "977442115". These numbers were associated to an employee and a client, respectively. As expected, these queries were not accepted. Then, we tried inserting the phone number "111111111", that wasn't in the database yet, to the same employee and the query was successfully executed, as it can be seen in figure 2.4.

```

MySQL [ist186950]> SELECT * FROM phone_number_employee WHERE VAT = 977442115;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977442115 | 955831285 |
| 977442115 | 955831289 |
+-----+-----+
2 rows in set (0.00 sec)

MySQL [ist186950]> INSERT INTO phone_number_employee VALUES(977442115, 955831140);
ERROR 1644 (45000): This phone number is already associated to an employee.
MySQL [ist186950]> INSERT INTO phone_number_employee VALUES(977442115, 955831782);
ERROR 1644 (45000): This phone number is already associated to a client.
MySQL [ist186950]> INSERT INTO phone_number_employee VALUES(977442115, 111111111);
Query OK, 1 row affected (0.01 sec)

MySQL [ist186950]> SELECT * FROM phone_number_employee WHERE VAT = 977442115;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977442115 | 111111111 |
| 977442115 | 955831285 |
| 977442115 | 955831289 |
+-----+-----+
3 rows in set (0.00 sec)

```

Figure 2.4: Test made to *Phone_employee_I* trigger

```

1 DROP TRIGGER IF EXISTS Phone_employee_U;
2
3 DELIMITER $$
4
5 CREATE TRIGGER Phone_employee_U BEFORE UPDATE ON phone_number_employee
6 FOR EACH ROW
7 BEGIN
8     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
9         phone = NEW.phone AND pne.VAT <> OLD.VAT))
10         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
11             number is already associated to an employee.';
12     END IF;
13     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
14         phone = NEW.phone))
15         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
16             number is already associated to a client.';
17     END IF;
18 END$$

```

Listing 2.6: Trigger *Phone_employee_U*

To test the trigger for updating an employee's phone number (*Phone_employee_U*), we tried updating the phone number previously added ("11111111") to the phone numbers "955831140" and "955831782". Once again, these queries were not accepted. Then, we tried updating it to the same phone number ("11111111"). This query was successfully executed, as expected and explained before (phone number being changed is not considered as a restriction). Next, we tried updating it to "22222222". Since this number wasn't associated yet, this query was successfully executed, as it can be seen in figure 2.5 (a). Finally, we tried changing the phone number "955831145" from a person (employee with VAT = "977442108", in this case) to another (employee with VAT = "977442115"). This test is represented in figure 2.5 (b) and it was successfully executed, since changing a phone number from one person to another is not against the integrity constraint.

For inserting and updating a client's phone number similar tests were made and similar results were obtained for the triggers *Phone_client_I* and *Phone_client_U*. This results are represented in figures 2.6 and 2.7.

```

1 DROP TRIGGER IF EXISTS Phone_client_I;
2
3 DELIMITER $$
4

```



```

MySQL [ist186950]> SELECT * FROM phone_number_employee WHERE VAT = 977442115;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977442115 | 1111111111 |
| 977442115 | 955831285  |
| 977442115 | 955831289  |
+-----+-----+
3 rows in set (0.01 sec)

MySQL [ist186950]> UPDATE phone_number_employee SET phone = 955831140 WHERE VAT = 977442115 AND phone = 1111111111;
ERROR 1644 (45000): This phone number is already associated to an employee.
MySQL [ist186950]> UPDATE phone_number_employee SET phone = 955831782 WHERE VAT = 977442115 AND phone = 1111111111;
ERROR 1644 (45000): This phone number is already associated to a client.
MySQL [ist186950]> UPDATE phone_number_employee SET phone = 1111111111 WHERE VAT = 977442115 AND phone = 1111111111;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 Warnings: 0

MySQL [ist186950]> UPDATE phone_number_employee SET phone = 222222222 WHERE VAT = 977442115 AND phone = 1111111111;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MySQL [ist186950]> SELECT * FROM phone_number_employee WHERE VAT = 977442115;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977442115 | 222222222  |
| 977442115 | 955831285  |
| 977442115 | 955831289  |
+-----+-----+
3 rows in set (0.00 sec)

```

(a)

```

MySQL [ist186950]> UPDATE phone_number_employee SET VAT = 977442115 WHERE phone = 955831145;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MySQL [ist186950]> SELECT * FROM phone_number_employee WHERE VAT = 977442115;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977442115 | 222222222  |
| 977442115 | 955831145  |
| 977442115 | 955831285  |
| 977442115 | 955831289  |
+-----+-----+
4 rows in set (0.00 sec)

```

(b)

Figure 2.5: Tests made to *Phone_employee_I* and *Phone_employee_U* triggers

```

5 CREATE TRIGGER Phone_client_I BEFORE INSERT ON phone_number_client
6 FOR EACH ROW
7 BEGIN
8     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
9                 phone = NEW.phone))
10        THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
11           number is already associated to an employee.';
12     END IF;
13     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
14                 phone = NEW.phone))
15        THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
16           number is already associated to a client.';
17     END IF;

```

14 END\$\$

Listing 2.7: Trigger *Phone_client_I*

```

1 DROP TRIGGER IF EXISTS Phone_client_U;
2
3 DELIMITER $$
4
5 CREATE TRIGGER Phone_client_U BEFORE UPDATE ON phone_number_client
6 FOR EACH ROW
7 BEGIN
8     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
9         phone = NEW.phone))
10         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
11             number is already associated to an employee.';
12     END IF;
13     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
14         phone = NEW.phone AND pnc.VAT <> OLD.VAT))
15         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
16             number is already associated to a client.';
17     END IF;
18 END$$

```

Listing 2.8: Trigger *Phone_client_U*

```

MySQL [ist186950]> SELECT * FROM phone_number_client WHERE VAT = 977444907;
+-----+-----+
| VAT   | phone |
+-----+-----+
| 977444907 | 955832127 |
| 977444907 | 955832131 |
+-----+-----+
2 rows in set (0.00 sec)

MySQL [ist186950]> INSERT INTO phone_number_client VALUES(977444907, 955831140);
ERROR 1644 (45000): This phone number is already associated to an employee.
MySQL [ist186950]> INSERT INTO phone_number_client VALUES(977444907, 955831782);
ERROR 1644 (45000): This phone number is already associated to a client.
MySQL [ist186950]> INSERT INTO phone_number_client VALUES(977444907, 333333333);
Query OK, 1 row affected (0.00 sec)

MySQL [ist186950]> SELECT * FROM phone_number_client WHERE VAT = 977444907;
+-----+-----+
| VAT   | phone |
+-----+-----+
| 977444907 | 333333333 |
| 977444907 | 955832127 |
| 977444907 | 955832131 |
+-----+-----+
3 rows in set (0.00 sec)

```

Figure 2.6: Test made to *Phone_client_I* trigger

```

MySQL [ist186950]> SELECT * FROM phone_number_client WHERE VAT = 977444907;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977444907 | 333333333 |
| 977444907 | 955832127 |
| 977444907 | 955832131 |
+-----+-----+
3 rows in set (0.00 sec)

MySQL [ist186950]> UPDATE phone_number_client SET phone = 955831140 WHERE VAT = 977444907 AND phone = 333333333;
ERROR 1644 (45000): This phone number is already associated to an employee.
MySQL [ist186950]> UPDATE phone_number_client SET phone = 955831782 WHERE VAT = 977444907 AND phone = 333333333;
ERROR 1644 (45000): This phone number is already associated to a client.
MySQL [ist186950]> UPDATE phone_number_client SET phone = 333333333 WHERE VAT = 977444907 AND phone = 333333333;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

MySQL [ist186950]> UPDATE phone_number_client SET phone = 444444444 WHERE VAT = 977444907 AND phone = 333333333;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MySQL [ist186950]> SELECT * FROM phone_number_client WHERE VAT = 977444907;
+-----+-----+
| VAT      | phone      |
+-----+-----+
| 977444907 | 444444444 |
| 977444907 | 955832127 |
| 977444907 | 955832131 |
+-----+-----+
3 rows in set (0.00 sec)

```

Figure 2.7: Test made to *Phone_client_U* trigger

2.4 Function

In this section, a function was created in order to compute the total number of no-shows (appointments that didn't become consultations, because the client missed them) for clients of a given gender, within a given age group and within a given year (year of the appointment). This function performs two queries, returns one integer (ns_count) and receives as arguments the gender, the lower and upper age limits and the year of the appointments/consultations. The first query counts the total number of appointments for this group of clients within the given year ("count"), while the second query counts the number of appointments that became consultations for this group of clients within the given year ("s_count"). This way it is possible to calculate the number of appointments that didn't become consultations by doing: "ns_count = count - s_count".

```

1 DELIMITER ;
2
3 DROP FUNCTION IF EXISTS no_shows_count;
4
5 DELIMITER $$
6
7 CREATE FUNCTION no_shows_count(gender VARCHAR(15), lower_age INTEGER, upper_age
    INTEGER, year INTEGER)
8     RETURNS INTEGER
9 BEGIN
10     DECLARE count INTEGER;
11     DECLARE s_count INTEGER;

```

```

12     SELECT count(*) INTO count
13     FROM appointment a1, client cl
14     WHERE a1.VAT_client = cl.VAT
15     AND cl.gender = gender
16     AND cl.age BETWEEN lower_age AND upper_age
17     AND year(a1.date_timestamp) = year;
18     SELECT count(*) INTO s_count
19     FROM client cl, consultation c LEFT JOIN appointment a
20     ON (a.VAT_doctor = c.VAT_doctor AND a.date_timestamp = c.date_timestamp)
21     WHERE a.VAT_client = cl.VAT
22     AND cl.gender = gender
23     AND cl.age BETWEEN lower_age AND upper_age
24     AND year(a.date_timestamp) = year;
25     RETURN count - s_count;
26 END $$
27
28 DELIMITER ;
29
30 SELECT no_shows_count('F', 80, 100, 2019) AS no_shows_count;

```

Listing 2.9: Function *no_shows_count*

We tested this function for gender = 'F', lower age limit = 80, upper age limit = 100 and year of the appointments = 2019. In figure 2.8, we can see the count of no-shows obtained, which is 6, as well as the corresponding information about the appointments. This information was obtained by running the following query.

```

1 SELECT a1.VAT_client, cl.name, cl.gender, cl.age, a1.date_timestamp, a1.
   VAT_doctor
2 FROM client cl, appointment a1
3 WHERE a1.VAT_client = cl.VAT
4 AND cl.gender = 'F'
5 AND cl.age BETWEEN 80 AND 100
6 AND year(a1.date_timestamp) = 2019
7 AND (a1.VAT_doctor, a1.date_timestamp) NOT IN (
8     SELECT c.VAT_doctor, c.date_timestamp
9     FROM consultation c LEFT JOIN appointment a2
10    ON (a2.VAT_doctor = c.VAT_doctor AND a2.date_timestamp = c.
       date_timestamp));

```

Listing 2.10: Query of function *no_shows_count*

As expected, the count (6) corresponds to the number of appointments that didn't become consultations obtained for clients of gender 'F' and within the age group 80-100, in 2019.

```
+-----+
| no_shows_count |
+-----+
|          6 |
+-----+
1 row in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| VAT_client | name          | gender | age | date_timestamp | VAT_doctor |
+-----+-----+-----+-----+-----+-----+
| 977444870 | Layton Mcguire | F      | 83 | 2019-01-14 10:00:00 | 977442106 |
| 977444870 | Layton Mcguire | F      | 83 | 2019-01-13 20:00:00 | 977442108 |
| 977444870 | Layton Mcguire | F      | 83 | 2019-01-08 20:00:00 | 977442114 |
| 977444879 | Ashley Murray  | F      | 99 | 2019-01-11 07:00:00 | 977442106 |
| 977444879 | Ashley Murray  | F      | 99 | 2019-01-04 11:00:00 | 977442114 |
| 977444891 | Mauricio Guerra | F      | 94 | 2019-01-04 10:00:00 | 977442116 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Figure 2.8: Test made to *no_shows_count* function

2.5 Procedure

In this section, a stored procedure was written in order to change the salary of all doctors that have been practicing for more than x years, where x corresponds to `n_years` (number of years) and is an input parameter. The new salary should correspond to a raise of 5 percent over the original salary if the doctor had more than 100 consultation in the current year, and a raise of 10 percent otherwise. The implemented stored procedure has two updates very similar to each other. Each update counts the number of consultations that each doctor had in the current year and verifies if this doctor has more than `n_years` of experience. The first one corresponds to doctors that had 100 or less consultations and it sets the salary to 1.05 times the previous salary. The second update corresponds to doctors that had more than 100 consultations and it sets the salary to 1.1 times the previous salary.

```

1 DELIMITER ;
2
3 DROP PROCEDURE IF EXISTS change_salary;
4
5 DELIMITER $$
6
7 CREATE PROCEDURE change_salary(IN n_years INTEGER)
8 BEGIN
9     UPDATE employee e
10    SET e.salary = 1.05 * e.salary
11    WHERE (SELECT count(*) FROM consultation c WHERE e.VAT = c.VAT_doctor
12    AND year(c.date_timestamp) = year(CURDATE()) GROUP BY c.VAT_doctor) <= 100
13    AND (SELECT pd.years FROM permanent_doctor pd WHERE e.VAT = pd.VAT) >
14    n_years;
15    UPDATE employee e
16    SET e.salary = 1.1 * e.salary
17    WHERE (SELECT count(*) FROM consultation c WHERE e.VAT = c.VAT_doctor
18    AND year(c.date_timestamp) = year(CURDATE()) GROUP BY c.VAT_doctor) > 100
19    AND (SELECT pd.years FROM permanent_doctor pd WHERE e.VAT = pd.VAT) >
20    n_years;
21 END$$
22
23 DELIMITER ;
24
25 CALL change_salary(10);

```

Listing 2.11: Procedure *change_salary*

To verify this procedure, a test was implemented by calling the procedure with an input value of 10 (10 years of practising). The results before and after calling this procedure are represented in figure 2.9. As it can be seen, the salaries of the doctors with VAT = "977442106" and VAT = "977442108" suffered a raise of 10 percent, while the salaries of the doctors with VAT = "977442107" and VAT = "977442109" suffered a raise of 5 percent. As expected, the salaries of the doctors with VAT = "977442110" and VAT = "977442111" didn't change, since this doctors have less that 10 years of practising.

```

+-----+-----+-----+-----+
| VAT_doctor | number_years | count_consultations | salary |
+-----+-----+-----+-----+
| 977442106 | 14 | 131 | 9322 |
| 977442107 | 18 | 18 | 7021 |
| 977442108 | 14 | 119 | 3400 |
| 977442109 | 23 | 54 | 8435 |
| 977442110 | 5 | 75 | 3143 |
| 977442111 | 3 | 112 | 4638 |
+-----+-----+-----+-----+
6 rows in set (0.01 sec)

```

(a) Before procedure

```

+-----+-----+-----+-----+
| VAT_doctor | number_years | count_consultations | salary |
+-----+-----+-----+-----+
| 977442106 | 14 | 131 | 10254 |
| 977442107 | 18 | 18 | 7372 |
| 977442108 | 14 | 119 | 3740 |
| 977442109 | 23 | 54 | 8857 |
| 977442110 | 5 | 75 | 3143 |
| 977442111 | 3 | 112 | 4638 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

(b) After procedure

Figure 2.9: Tests made to *change_salary* procedure

Appendix A - Code files used in the projection

3.1 PHP/HTML code - Part 1

3.1.1 searchclient.php

```
1 <html>
2   <body>
3     <form action="showclient.php" method="post">
4       <h3> Search for a client </h3>
5       <p> VAT number: <input type="text" name="cvat"/></p>
6       <p> Name: <input type="text" name="cname"/></p>
7       <p> Address: <input type="text" name="caddress"/></p>
8       <p> <input type="submit" value="Submit"/></p>
9     </form>
10  </body>
11 </html>
```

3.1.2 showclient.php

```
1 <?php
2   require_once('sql_funcs.php');
3
4
5   $connection = null;
6   new_connection($connection);
7
8   $sql = "SELECT * FROM client
9         WHERE (client.VAT = :cvat OR :cvat = '')
10        AND (client.name LIKE :cname OR :cname = '')
11        AND (client.city LIKE :caddress
```

```

12         OR client.street LIKE :caddress
13         OR client.zip LIKE :caddress
14         OR :caddress = ''");
15     $result = sql_secure_query($connection, $sql, Array(":cvat" => $_REQUEST['
cvat'],
16                                     ":cname" => '%'.
$_REQUEST['cname'].'%',
17                                     ":caddress" => '%'.
$_REQUEST['caddress'].'%'));
18     $connection=null;
19     $nrows = $result->rowCount();
20     if($nrows == 0){
21         echo("<h3> There were found no clients meeting those conditions!</h3>");
22     }
23     else{
24         echo("<h3> Clients found: </h3>\n");
25         echo("<table border=\"1\">");
26         echo("<tr><td>VAT</td><td>name</td><td>birth_date</td><td>street</td><td>
>city</td><td>zip</td><td>
27             gender</td><td>age</td><td>Make an Appointment</td></tr>");
28         foreach($result as $row){
29             echo("<tr><td>");
30             echo("<a href=\"showappointment.php?cvat=");
31             echo($row['VAT']);
32             echo(">{$row['VAT']}</a>");
33             echo("</td><td>");
34             echo($row['name']);
35             echo("</td><td>");
36             echo($row['birth_date']);
37             echo("</td><td>");
38             echo($row['street']);
39             echo("</td><td>");
40             echo($row['city']);
41             echo("</td><td>");
42             echo($row['zip']);
43             echo("</td><td>");
44             echo($row['gender']);
45             echo("</td><td>");
46             echo($row['age']);

```

```

47         echo("</td><td>");
48         echo("<a href=\"newappointment.php?cvat=\"");
49         echo($row['VAT']);
50         echo(">Insert New Appointment</a>");
51         echo("</td></tr>\n");
52     }
53     echo("</table>\n");
54 }
55 ?>
56 <html>
57     <body>
58         <a href="newclient.php"> Insert New Client </a> <br>
59         <a href="searchclient.php"> Go to the Previous Page </a>
60     </body>
61 </html>

```

3.1.3 newclient.php

```

1 <html>
2     <body>
3         <form action="insertclient.php" method="post">
4             <h3>Insert a new client </h3>
5             <p> VAT number: <input type="number" name="cvat" min="100000000" max
6             ="999999999"/></p>
7             <p> Name: <input type="text" name="cname"/></p>
8             <p> Birth Date: <input type="date" name="cbday" max="<?=date("Y-m-d"
9             )?>"/></p>
10            <p> Street: <input type="text" name="cstreet"/></p>
11            <p> City: <input type="text" name="ccity"/></p>
12            <p> Zip Code Part1: <input type="number" name="czip1" min="1000" max
13            ="9999"/></p>
14            <p> Zip Code Part2: <input type="number" name="czip2" min="100" max=
15            "999"/></p>
16            <p> Gender: <input type="radio" name="cgender" value="M" checked>
17            Male
18                <input type="radio" name="cgender" value="F"> Female
19                <input type="radio" name="cgender" value="other"> Other
20            </p>
21            <p> Age: <input type="number" name="cage" min="0" max ="<?200?>"
22            <p> <input type="submit" value="Submit"/></p>

```



```

17         <a href="searchclient.php"> Go to the First Page </a>
18     </form>
19 </body>
20 </html>

```

3.1.4 insertclient.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4
5     $connection = null;
6     new_connection($connection);
7
8     $dateage = strtotime($_REQUEST['cbday']);
9     if($_REQUEST['cvat'] == 0 || strlen($_REQUEST['cname']) < 3 || date("Y",
10     $dateage) < 1850 ){
11         echo("<p> Fill in correctly all the camps in the form!</p><br>");
12     }
13     else if(date("Y", $dateage) + $_REQUEST['cage'] > date("Y") ||
14         date("Y", $dateage) + $_REQUEST['cage'] < date("Y") - 1){
15         echo("<p> Fill the age correctly pls! </p><br>");
16     }
17     else{
18         $zip = $_REQUEST['czip1'].'-'. $_REQUEST['czip2'];
19
20         $sql = "INSERT INTO client VALUES (:cvat, :cname, :cbday, :cstreet, :
21         ccity, :czip, :cgender, :cage)";
22         $result = sql_secure_query($connection, $sql, Array(":cvat" => $_REQUEST
23         ['cvat'],
24                                     ":cname" =>
25         $_REQUEST['cname'],
26                                     ":cbday" =>
27         $_REQUEST['cbday'],
28                                     ":cstreet" =>
29         $_REQUEST['cstreet'],
30                                     ":ccity" =>
31         $_REQUEST['ccity'],
32                                     ":czip" => $zip,
33                                     ":cgender" =>

```

```

$_REQUEST['cgender'],
27                                     " :cage" => $_REQUEST
['cage']));
28
29     $nrows = $result->rowCount();
30     echo("<p>Rows inserted: $nrows</p>");
31     $sql = "SELECT * FROM client WHERE client.VAT = :cvat";
32     $result = sql_secure_query($connection, $sql, Array(" :cvat" => $_REQUEST
['cvat']));
33     echo("<p>You inserted a new client with the following values:</p>");
34     echo("<table>");
35     echo("<tr><td>VAT</td><td>name</td><td>birth_date</td><td>street</td><td>
>city</td><td>zip</td><td>
36         gender</td><td>age</td></tr>");
37     foreach($result as $row){
38         echo("<tr><td>");
39         echo($row['VAT']);
40         echo("</td><td>");
41         echo($row['name']);
42         echo("</td><td>");
43         echo($row['birth_date']);
44         echo("</td><td>");
45         echo($row['street']);
46         echo("</td><td>");
47         echo($row['city']);
48         echo("</td><td>");
49         echo($row['zip']);
50         echo("</td><td>");
51         echo($row['gender']);
52         echo("</td><td>");
53         echo($row['age']);
54         echo("</td></tr>\n");
55     }
56     echo("</table>\n");
57
58 }
59 $connection = null;
60 ?>
61

```

```

62 <html>
63     <body>
64         <a href="newclient.php"> Insert More Clients </a>
65     </body>
66 </html>

```

3.1.5 newappointment.php

```

1 <html>
2     <body>
3         <form action="showdoctors.php" method="post">
4             <h3>Select an hour and day for the appointment</h3>
5             <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
$_REQUEST['cvat'])?>" /> </p>
6             <p> Date: <input type="date" name="appdate" min="<?=date("Y-m-d")?>"
/></p>
7             <p> Hour: <input type="number" name="apphour" min="09" max="17"/></p>
8             <p> <input type="submit" value="Submit"/></p>
9             <a href="searchclient.php"> Go to the First Page </a>
10        </form>
11    </body>
12 </html>

```

3.1.6 showdoctors.php

```

1 <html>
2     <body>
3         <form action="insertappointment.php" method="post">
4             <h3>The available doctors for that time are</h3>
5             <p>Name and Vat:
6                 <select name="appdoctor">
7
8 <?php
9     require_once('sql_funcs.php');
10
11     $connection = null;
12     new_connection($connection);
13
14     $date = $_REQUEST['appdate'].' '.$_REQUEST['apphour'].':00:00';
15     $sql = "SELECT DISTINCT e.VAT, name

```

```

16         FROM (employee AS e LEFT OUTER JOIN appointment AS a
17             ON e.VAT = a.VAT_doctor) RIGHT OUTER JOIN doctor
18             ON doctor.VAT = e.VAT
19             WHERE e.VAT NOT IN(SELECT VAT_doctor
20                 FROM appointment AS a2
21                 WHERE a2.date_timestamp = :appdate)");
22
23     $result = sql_secure_query($connection, $sql, Array(":appdate" => $date));
24     $connection = null;
25     $nrows = $result->rowCount();
26     if($nrows == 0){
27         echo("<h3> There are no available doctors at the given hour </h3>");
28     }
29     else{
30         foreach($result as $row){
31             $vat = $row['VAT'];
32             $name = $row['name'];
33             echo("<option value=\"\$vat\">$vat $name</option>");
34         }
35     }
36 ?>
37         </select>
38     </p>
39     <p> Description: <input type="text" name="desc"/></p>
40     <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
41         $_REQUEST['appcvat'])?>" /> </p>
42     <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
43         $date)?>" /> </p>
44     <p><input type="submit" value="Submit"/></p>
45 </form>
46 </body>
47 </html>
48 <?php
49     echo("<a href=\"newappointment.php?cvat=");
50     echo($_REQUEST['appcvat']);
51     echo(">Go to the Previous Page</a>");
52 ?>

```

3.1.7 insertappointment.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4     $connection = null;
5     new_connection($connection);
6
7     $sql = "INSERT INTO appointment VALUES(:dvat, :date, :desc, :cvat)";
8     $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
9     appdoctor'],
10                                     ":date" => $_REQUEST['
11     appdate'],
12                                     ":desc" => $_REQUEST['
13     desc'],
14                                     ":cvat" => $_REQUEST['
15     appcvat']));
16
17     $nrows = $result->rowCount();
18     echo("<p>Rows inserted: $nrows</p>");
19     $sql = "SELECT * FROM appointment WHERE appointment.VAT_doctor = :dvat AND
20     appointment.date_timestamp = :date";
21     $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
22     appdoctor'],
23                                     ":date" => $_REQUEST['
24     appdate']));
25
26     echo("<p>You inserted a new appointment with the following values:</p>");
27     echo("<table>");
28     echo("<tr><td>VAT_doctor</td><td>date_timestamp</td><td>description</td><td>
29     VAT_client</td></tr>");
30     foreach($result as $row){
31         echo("<tr><td>");
32         echo($row['VAT_doctor']);
33         echo("</td><td>");
34         echo($row['date_timestamp']);
35         echo("</td><td>");
36         echo($row['description']);
37         echo("</td><td>");
38     }

```



```

30     echo($row['VAT_client']);
31     echo("</td></tr>\n");
32 }
33 echo("</table>\n");
34 $connection = null;
35 echo("<a href=\"newappointment.php?cvat=\"");
36 echo($_REQUEST['appcvat']);
37 echo("\>Insert More Appointments</a>");
38 ?>

```

3.1.8 showappointment.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4     $connection = null;
5     new_connection($connection);
6
7     $sql = "SELECT * FROM appointment
8           WHERE appointment.VAT_client = :cvat";
9
10    $result = sql_secure_query($connection, $sql, Array(":cvat" => $_REQUEST['
11    cvat']));
12    $connection = null;
13    $nrows = $result->rowCount();
14    if($nrows == 0){
15        echo("<h3> There were found no appointments for the given client </h3>");
16    };
17    }
18    else{
19        echo("<h3> Appointments found for ");
20        echo($_REQUEST['cvat']);
21        echo(": </h3>");
22        echo("<table border=\"1\">");
23        echo("<tr><td>VAT_doctor</td><td>date_timestamp</td><td>description");
24        echo("</td><td>Check this Appointment</td></tr>");
25        foreach($result as $row){
26            echo("<tr><td>");
27            echo($row['VAT_doctor']);
28            echo("</td><td>");

```

```

27         echo($row['date_timestamp']);
28         echo("</td><td>");
29         echo($row['description']);
30         echo("</td><td>");
31         echo("<a href=\"showconsultation.php?appcvat=");
32         echo($row['VAT_client']);
33         echo("&appdvat=");
34         echo($row['VAT_doctor']);
35         echo("&appdate=");
36         echo($row['date_timestamp']);
37         echo("\>Check this Appointment</a>");
38         echo("</td></tr>\n");
39     }
40     echo("</table>\n");
41 }
42 ?>
43
44 <html>
45     <body>
46         <a href="searchclient.php"> Go to the First Page </a>
47     </body>
48 </html>

```

3.1.9 showconsultation.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4     $connection = null;
5     new_connection($connection);
6
7     $sql = "SELECT * FROM consultation
8           WHERE consultation.VAT_doctor = :conductor
9           AND consultation.date_timestamp = :condate";
10    $result = sql_secure_query($connection, $sql, Array(":conductor" =>
11    $_REQUEST['appdvat'],
12                                     ":condate" => $_REQUEST[
13    'appdate']));
14    $nrows = $result->rowCount();
15    if($nrows == 0){

```

```

14         echo("<h3> There was no consultation for the selected appointment </h3>"
15     );
16     $connection = null;
17 }
18 else{
19     echo("<h3> Information for the consultation with doctor ");
20     echo($_REQUEST['appdvat']);
21     echo(" that happened on ");
22     echo($_REQUEST['appdate']);
23     echo(": </h3>\n");
24     foreach($result as $row){
25         echo("<strong>Soap Notes - Subjective:</strong><br>");
26         echo($row['SOAP_S']);
27         echo("<br>");
28         echo("<strong>Soap Notes - Objective:</strong><br>");
29         echo($row['SOAP_O']);
30         echo("<br>");
31         echo("<strong>Soap Notes - Assessment:</strong><br>");
32         echo($row['SOAP_A']);
33         echo("<br>");
34         echo("<strong>Soap Notes - Plan:</strong><br>");
35         echo($row['SOAP_P']);
36         echo("<br>");
37     }
38
39     $sql = "SELECT ca.VAT_nurse AS vat, e.name AS name
40           FROM employee AS e, consultation_assistant AS ca
41           WHERE ca.VAT_nurse = e.VAT
42                 AND ca.VAT_doctor = :conductor
43                 AND ca.date_timestamp = :condate";
44     $result = sql_secure_query($connection, $sql, Array(":conductor" =>
45     $_REQUEST['appdvat'],
46                                     ":condate" =>
47     $_REQUEST['appdate']));
48     $nrows = $result->rowCount();
49     if($nrows != 0){
50         echo("<strong>Nurses involved: </strong><br>");
51         foreach($result as $row){
52             echo($row['name']);

```

```

50         echo(" - ");
51         echo($row['vat']);
52         echo("<br>");
53     }
54     echo("<br>");
55 }
56 $sql = "SELECT cd.ID AS ID, description
57         FROM diagnostic_code AS d, consultation_diagnostic AS cd
58         WHERE d.ID = cd.ID
59         AND cd.VAT_doctor = :conductor
60         AND cd.date_timestamp = :condate";
61 $result = sql_secure_query($connection, $sql, Array(":conductor" =>
62 $_REQUEST['appdvat'],
63                                     ":condate" =>
64 $_REQUEST['appdate']));
65 $nrows = $result->rowCount();
66 if($nrows != 0){
67     echo("<strong> Diagnosis' achieved and corresponding prescriptions:
68 </strong> <br>") ;
69     foreach($result as $row){
70         echo($row['ID']);
71         echo("<br>");
72         echo($row['description']);
73         echo("<br>");
74         echo("<br>");
75         $sql = "SELECT name, lab, dosage, description
76                 FROM prescription AS p
77                 WHERE p.VAT_doctor = :conductor
78                 AND p.date_timestamp = :condate
79                 AND p.ID = :pid";
80         $result2 = sql_secure_query($connection, $sql, Array(":conductor"
81 => $_REQUEST['appdvat'],
82                                     ":condate"
83 => $_REQUEST['appdate'],
84                                     ":pid" =>
85 $row['ID']));
86         $nrows2 = $result2->rowCount();

```

```

83         if($nrows2 != 0){
84             echo("Prescription for ");
85             echo($row['ID']);
86             echo(" prescribed on ");
87             echo($_REQUEST['appdate']);
88             echo(" by doctor ");
89             echo($_REQUEST['appdvat']);
90             echo("<table border=\"1\">");
91             echo("<tr><td>name</td><td>lab</td><td>dosage</td><td>
description</td></tr>");
92             foreach($result2 as $row2){
93                 echo("<tr><td>");
94                 echo($row2['name']);
95                 echo("</td><td>");
96                 echo($row2['lab']);
97                 echo("</td><td>");
98                 echo($row2['dosage']);
99                 echo("</td><td>");
100                echo($row2['description']);
101                echo("</td></tr>\n");
102            }
103            echo("</table>\n");
104        }
105    }
106 }
107 }
108 echo("<a href=\"newcharting.php?appcvat=");
109 echo($_REQUEST['appcvat']);
110 echo("&appdvat=");
111 echo($_REQUEST['appdvat']);
112 echo("&appdate=");
113 echo($_REQUEST['appdate']);
114 echo(">Insert Data for a Dental Charting Procedure</a><br>");
115 echo("<a href=\"newdata.php?appcvat=");
116 echo($_REQUEST['appcvat']);
117 echo("&appdvat=");
118 echo($_REQUEST['appdvat']);
119 echo("&appdate=");
120 echo($_REQUEST['appdate']);

```

```

121     echo("\>Insert More Data for this Consultation</a><br>");
122     echo("<a href=\"showappointment.php?cvat=");
123     echo($_REQUEST['appcvat']);
124     echo("\>Go to the Previous Page</a>");
125 ?>

```

3.1.10 newdata.php

```

1 <html>
2     <body>
3         <form action="insertnurse.php" method="post">
4             <h3>Insert a nurse for this consultation </h3>
5             <p>Name and Vat:
6                 <select name="connurse">
7 <?php
8     require_once('sql_funcs.php');
9
10    $connection = null;
11    new_connection($connection);
12
13    $sql = "SELECT DISTINCT e.VAT, name
14            FROM employee AS e, nurse AS n
15            WHERE e.VAT = n.VAT
16            AND e.VAT NOT IN(SELECT ca.VAT_nurse
17                               FROM consultation_assistant AS ca
18                               WHERE ca.VAT_doctor = :condoc
19                               AND ca.date_timestamp = :condate)";
20
21    $result = sql_secure_query($connection, $sql, Array(":condoc" => $_REQUEST['
22    appdvat'],
23    ":condate" => $_REQUEST[
24    'appdate']));
25    foreach($result as $row){
26        $vat = $row['VAT'];
27        $name = $row['name'];
28        echo("<option value=\"\$vat\">\$vat \$name</option>");
29    }
30 ?>
31
32     </select>
33 </p>
34
35 </body>
36 </html>

```



```

31         <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
$_REQUEST['appcvat'])?>" /> </p>
32         <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
$_REQUEST['appdvat'])?>" /> </p>
33         <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
$_REQUEST['appdate'])?>" /> </p>
34         <p><input type="submit" value="Submit"/></p>
35     </form>
36 </body>
37 <html>
38
39 <html>
40     <body>
41         <form action="insertsoap.php" method="post">
42             <h3>Introduce data for the consultation corresponding to this
appointment</h3>
43             <p> Soap Notes - Subjective: <input type="text" name="soaps"/></p>
44             <p> Soap Notes - Objective: <input type="text" name="soapo"/></p>
45             <p> Soap Notes - Assessment: <input type="text" name="soapa"/></p>
46             <p> Soap Notes - Plan: <input type="text" name="soapp"/></p>
47             <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
$_REQUEST['appcvat'])?>" /> </p>
48             <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
$_REQUEST['appdvat'])?>" /> </p>
49             <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
$_REQUEST['appdate'])?>" /> </p>
50             <p><input type="submit" value="Submit"/></p>
51         </form>
52     </body>
53 </html>
54
55 <html>
56     <body>
57         <form action="insertdiagnostic.php" method="post">
58             <h3>Insert a diagnostic code for this consultation </h3>
59             <p>Diagnostic Code:
60                 <select name="diacode">
61 <?php
62

```

```

63     $sql = "SELECT DISTINCT ID
64           FROM diagnostic_code
65           WHERE ID NOT IN(SELECT ID
66                           FROM consultation_diagnostic AS cd
67                           WHERE cd.VAT_doctor = :condoc
68                           AND cd.date_timestamp = :condate)";
69     $result = sql_secure_query($connection, $sql, Array(":condoc" => $_REQUEST['
70     appdvat'],
71                                     ":condate" => $_REQUEST[
72     'appdate']));
73     $nrows = $result->rowCount();
74     foreach($result as $row){
75         $ID = $row['ID'];
76         echo("<option value=\"\$ID\">$ID</option>");
77     }
78 ?>
79     </select>
80     </p>
81     <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
82     $_REQUEST['appcvat'])?>" /> </p>
83     <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
84     $_REQUEST['appdvat'])?>" /> </p>
85     <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
86     $_REQUEST['appdate'])?>" /> </p>
87     <p><input type="submit" value="Submit"/></p>
88     </form>
89     </body>
90 </html>
91 <html>
92     <body>
93         <form action="insertprescription.php" method="post">
94         <h3>Insert a prescription for this consultation</h3>
95         <p>Diagnostic code to cure:
96         <select name="prescid">
97     <?php
98     $sql = "SELECT DISTINCT ID
99           FROM consultation_diagnostic AS cd
100          WHERE cd.VAT_doctor = :condoc

```

```

97         AND cd.date_timestamp = :condate";
98     $result = sql_secure_query($connection, $sql, Array(":condoc" => $_REQUEST['
appdvat'],
99                                     ":condate" => $_REQUEST[
'update']));
100     $nrows = $result->rowCount();
101     foreach($result as $row){
102         $ID = $row['ID'];
103         echo("<option value=\"\$ID\">$ID</option>");
104     }
105 ?>
106         </select>
107     </p>
108     <p>Medication Name:
109         <select name="med">
110 <?php
111     $sql = "SELECT DISTINCT name, lab
112           FROM medication AS m
113           WHERE NOT EXISTS(SELECT 1
114                           FROM prescription AS p
115                           WHERE p.VAT_doctor = :condoc
116                           AND p.date_timestamp = :condate
117                           AND m.name = p.name
118                           AND m.lab = p.lab)";
119     $result = sql_secure_query($connection, $sql, Array(":condoc" => $_REQUEST['
appdvat'],
120                                     ":condate" => $_REQUEST[
'update']));
121     $nrows = $result->rowCount();
122     foreach($result as $row){
123         $name = $row['name'];
124         $lab = $row['lab'];
125         $append = $name."!".$lab;
126         echo("<option value=\"\$append\">$name $lab</option>");
127     }
128 ?>
129         </select>
130     </p>
131     <p>Dosage: <input type="number" name="dosage"/></p>

```

```

132     <p>Description: <input type="text" name="description"/></p>
133     <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
$_REQUEST['appcvat'])?>" /> </p>
134     <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
$_REQUEST['appdvat'])?>" /> </p>
135     <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
$_REQUEST['appdate'])?>" /> </p>
136     <p><input type="submit" value="Submit"/></p>
137 </form>
138 </body>
139 </html>
140
141 <?php
142     echo("<a href=\"showconsultation.php?appcvat=");
143     echo($_REQUEST['appcvat']);
144     echo("&appdvat=");
145     echo($_REQUEST['appdvat']);
146     echo("&appdate=");
147     echo($_REQUEST['appdate']);
148     echo("\>Go to the Previous Page</a><br>");
149 ?>

```

3.1.11 insertnurse.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4
5     $connection = null;
6     new_connection($connection);
7
8     $sql = "SELECT * FROM consultation AS c WHERE c.VAT_doctor = :dvat
9             AND c.date_timestamp = :date";
10    $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
appdvat'],
11                                                         ":date" => $_REQUEST['
appdate']));
12    $nrows = $result->rowCount();
13    if($nrows == 0){
14        $sql = "INSERT INTO consultation VALUES (:dvat, :date, '', '', '', '')";

```

```

15     $result = sql_secure_query($connection, $sql, Array("dvat" => $_REQUEST[
16         'appdvat'],
17                                     ":date" => $_REQUEST
18         ['appdate']));
19
20     $nrows = $result->rowCount();
21     echo("<p>Rows inserted: $nrows</p>");
22 }
23 $sql = "INSERT INTO consultation_assistant VALUES (:dvat, :date, :nvat)";
24 $result = sql_secure_query($connection, $sql, Array("dvat" => $_REQUEST['
25     appdvat'],
26                                     ":date" => $_REQUEST['
27     appdate'],
28                                     ":nvat" => $_REQUEST['
29     connurse']));
30
31 $nrows = $result->rowCount();
32 echo("<p>Rows inserted: $nrows</p>");
33 $sql = "SELECT * FROM consultation_assistant AS ca WHERE ca.VAT_doctor = :
34     dvat
35                                     AND ca.date_timestamp =
36     :date
37                                     AND ca.VAT_nurse = :nvat
38 ";
39 $result = sql_secure_query($connection, $sql, Array("dvat" => $_REQUEST['
40     appdvat'],
41                                     ":date" => $_REQUEST['
42     appdate'],
43                                     ":nvat" => $_REQUEST['
44     connurse']));
45
46 echo("<p>You inserted a new assistant for the desired consultation with the
47 following values:</p>");
48 echo("<table>");
49 echo("<tr><td>VAT_doctor</td><td>date_timestamp</td><td>VAT_nurse</td></tr>"
50 );
51 foreach($result as $row){
52     echo("<tr><td>");
53     echo($row['VAT_doctor']);
54     echo("</td><td>");

```

```

41     echo($row['date_timestamp']);
42     echo("</td><td>");
43     echo($row['VAT_nurse']);
44     echo("</td></tr>\n");
45 }
46 echo("</table>\n");
47
48
49 $connection = null;
50 echo("<a href=\"newdata.php?appcvat=\"");
51 echo($_REQUEST['appcvat']);
52 echo("&appdvat=\"");
53 echo($_REQUEST['appdvat']);
54 echo("&appdate=\"");
55 echo($_REQUEST['appdate']);
56 echo("\n">Insert More Data for this Consultation</a><br>");
57 ?>

```

3.1.12 insertsoap.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4
5     $connection = null;
6     new_connection($connection);
7
8     $sql = "SELECT * FROM consultation AS c WHERE c.VAT_doctor = :dvat
9           AND c.date_timestamp = :date";
10    $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['appdvat'],
11                                                       ":date" => $_REQUEST['appdate']));
12    $nrows = $result->rowCount();
13    if($nrows == 0){
14        echo("<strong>Insert a nurse before inserting soap notes.</strong><br>");
15        ;
16        echo("<strong>Its mandatory that every consultation has a nurse.</strong><br>");
17    }

```

```

17     else{
18         $sql = "UPDATE consultation
19             SET VAT_doctor = :condoc, date_timestamp = :condate, SOAP_S = :
20             soaps, SOAP_O = :soapo, SOAP_A = :soapa, SOAP_P = :soapp
21             WHERE consultation.VAT_doctor = :condoc
22             AND consultation.date_timestamp = :condate";
23         $result = sql_secure_query($connection, $sql, Array(":condoc" =>
24             $_REQUEST['appdvat'],
25             ":condate" =>
26             $_REQUEST['appdate'],
27             ":soaps" =>
28             $_REQUEST['soaps'],
29             ":soapo" =>
30             $_REQUEST['soapo'],
31             ":soapa" =>
32             $_REQUEST['soapa'],
33             ":soapp" =>
34             $_REQUEST['soapp']));
35         $nrows = $result->rowCount();
36         echo("<p>Rows inserted: $nrows</p><br>");
37         echo("<p>You inserted the desired soap notes, go check the consultation
38         again to see them!! </p>");
39     }
40
41     $connection = null;
42     echo("<a href='\"newdata.php?appcvat='\"");
43     echo($_REQUEST['appcvat']);
44     echo("&appdvat='\"");
45     echo($_REQUEST['appdvat']);
46     echo("&appdate='\"");
47     echo($_REQUEST['appdate']);
48     echo(">Insert More Data for this Consultation</a><br>");
49 ?>

```

3.1.13 insertdiagnostic.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4

```

```

5     $connection = null;
6     new_connection($connection);
7
8     $sql = "SELECT * FROM consultation AS c WHERE c.VAT_doctor = :dvat
9             AND c.date_timestamp = :date";
10    $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
11    appdvat'],
12                                     ":date" => $_REQUEST['
13    appdate']));
14    $nrows = $result->rowCount();
15    if($nrows == 0){
16        echo("<strong>Insert a nurse before inserting a diagnostic code.</strong>
17        ><br>");
18        echo("<strong>Its mandatory that every consultation has a nurse.</strong>
19        ><br>");
20    }
21    else{
22        $sql = "INSERT INTO consultation_diagnostic VALUES (:dvat, :date, :
23        diacode)";
24        $result = sql_secure_query($connection, $sql, Array(":dvat" =>
25        $_REQUEST['appdvat'],
26                                     ":date" => $_REQUEST
27        ['appdate'],
28                                     ":diacode" =>
29        $_REQUEST['diacode']));
30
31        $nrows = $result->rowCount();
32        echo("<p>Rows inserted: $nrows</p>");
33        $sql = "SELECT * FROM consultation_diagnostic AS cd WHERE cd.VAT_doctor
34        = :dvat
35                                     AND cd.
36        date_timestamp = :date
37                                     AND cd.ID = :diacode
38        ";
39        $result = sql_secure_query($connection, $sql, Array("dvat" => $_REQUEST[
40        'appdvat'],
41                                     ":date" => $_REQUEST
42        ['appdate'],
43                                     ":diacode" =>

```



```

$_REQUEST['diacode']));
31     echo("<p>You inserted a new diagnostic code for the desired consultation
    with the following values:</p>");
32     echo("<table>");
33     echo("<tr><td>VAT_doctor</td><td>date_timestamp</td><td>ID</td></tr>");
34     foreach($result as $row){
35         echo("<tr><td>");
36         echo($row['VAT_doctor']);
37         echo("</td><td>");
38         echo($row['date_timestamp']);
39         echo("</td><td>");
40         echo($row['ID']);
41         echo("</td></tr>\n");
42     }
43     echo("</table>\n");
44 }
45
46 $connection = null;
47 echo("<a href=\"newdata.php?appcvat=");
48 echo($_REQUEST['appcvat']);
49 echo("&appdvat=");
50 echo($_REQUEST['appdvat']);
51 echo("&appdate=");
52 echo($_REQUEST['appdate']);
53 echo("\>Insert More Data for this Consultation</a><br>");
54 ?>

```

3.1.14 insertprescription.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4
5     $connection = null;
6     new_connection($connection);
7
8     $sql = "SELECT * FROM consultation AS c WHERE c.VAT_doctor = :dvat
9           AND c.date_timestamp = :date";
10    $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
    appdvat'],

```

```

11                                     ":date" => $_REQUEST['
appdate']));
12     $nrows = $result->rowCount();
13     if($nrows == 0){
14         echo("<strong>Insert a nurse before inserting a prescription.</strong><
br>");
15         echo("<strong>Its mandatory that every consultation has a nurse.</strong>
<br>");
16     }
17     else{
18         $sql = "SELECT * FROM consultation_diagnostic AS cd WHERE cd.VAT_doctor
= :dvat
19                                     AND cd.
date_timestamp = :date";
20         $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST
['appdvat'],
21                                     ":date" => $_REQUEST
['appdate']));
22
23         $nrows = $result->rowCount();
24         if($nrows == 0){
25             echo("<strong>Insert a diagnostic code before inserting a
prescription.</strong><br>");
26             echo("<strong>There can't be a prescription without a diagnostic
code.</strong><br>");
27         }
28         else{
29             $med = explode("!", $_REQUEST['med']);
30             $sql = "INSERT INTO prescription VALUES (:name, :lab, :dvat, :date,
:prescid, :dosage, :desc)";
31             $result = sql_secure_query($connection, $sql, Array(":name" => $med
[0],
32                                     ":lab" => $med
[1],
33                                     ":dvat" =>
$_REQUEST['appdvat'],
34                                     ":date" =>
$_REQUEST['appdate'],
35                                     ":prescid" =>

```

```

$_REQUEST['prescid'],
36                                     ":dosage" =>
$_REQUEST['dosage'],
37                                     ":desc" =>
$_REQUEST['description']));
38
39     $nrows = $result->rowCount();
40     echo("<p>Rows inserted: $nrows</p>");
41     $sql = "SELECT * FROM prescription AS p WHERE p.name = :name
42                                     AND p.lab = :lab
43                                     AND p.VAT_doctor = :dvat
44                                     AND p.date_timestamp = :date
45                                     AND p.ID = :prescid";
46     $result = sql_secure_query($connection, $sql, Array(":name" => $med
[0],
47                                     ":lab" => $med
[1],
48                                     ":dvat" =>
$_REQUEST['appdvat'],
49                                     ":date" =>
$_REQUEST['appdate'],
50                                     ":prescid" =>
$_REQUEST['prescid']));
51     echo("<p>You inserted a new prescription for the desired
consultation with the following values:</p>");
52     echo("<table>");
53     echo("<tr><td>name</td><td>lab</td><td>VAT_doctor</td><td>
date_timestamp</td><td>ID</td><td>dosage</td><td>description</td></tr>");
54     foreach($result as $row){
55         echo("<tr><td>");
56         echo($row['name']);
57         echo("</td><td>");
58         echo($row['lab']);
59         echo("</td><td>");
60         echo($row['VAT_doctor']);
61         echo("</td><td>");
62         echo($row['date_timestamp']);
63         echo("</td><td>");
64         echo($row['ID']);

```

```

65         echo("</td><td>");
66         echo($row['dosage']);
67         echo("</td><td>");
68         echo($row['description']);
69         echo("</td></tr>\n");
70     }
71     echo("</table>\n");
72 }
73 }
74 $connection = null;
75 echo("<a href=\"newdata.php?appcvat=");
76 echo($_REQUEST['appcvat']);
77 echo("&appdvat=");
78 echo($_REQUEST['appdvat']);
79 echo("&appdate=");
80 echo($_REQUEST['appdate']);
81 echo("\n">Insert More Data for this Consultation</a><br>");
82
83 ?>

```

3.1.15 newcharting.php

```

1 <html>
2     <body>
3         <form action="newteeth.php" method="post">
4             <h3>Insert Data For a Dental Charting Procedure </h3>
5             <h4>Choose the Name of the Procedure </h4>
6             <p> Name:
7                 <select name="pro_name">
8                     <?php
9                         require_once('sql_funcs.php');
10                        $connection = null;
11                        new_connection($connection);
12
13                        $sql = "SELECT DISTINCT name
14                               FROM procedurette
15                               WHERE type_='dental charting
16                        '";
17
18                        $result = sql_secure_query(
19                            $connection, $sql, Array());

```

```

17         foreach($result as $row){
18             $pname = $row['name'];
19             echo("<option value=\" $pname\">
    $pname</option>");
20         }
21     ?>
22     </select>
23     </p>
24     <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
    $_REQUEST['appcvat'])?>" /> </p>
25     <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
    $_REQUEST['appdvat'])?>" /> </p>
26     <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
    $_REQUEST['appdate'])?>" /> </p>
27     <p><input type="submit" value="Insert Teeth"/></p>
28 </form>
29 </body>
30 <html>
31
32 <?php
33     echo("<a href=\"showconsultation.php?appcvat=");
34     echo($_REQUEST['appcvat']);
35     echo("&appdvat=");
36     echo($_REQUEST['appdvat']);
37     echo("&appdate=");
38     echo($_REQUEST['appdate']);
39     echo("\>Go to the Previous Page</a><br>");
40 ?>

```

3.1.16 newteeth.php

```

1 <html>
2     <body>
3         <form action="insertcharting.php" method="post">
4             <h3>Insert Data For a Dental Charting Procedure </h3>
5             <h4>Choose the Teeth of the Procedure </h4>
6             <p> Teeth:
7                 <select name="teeth">
8                     <?php
9

```

```

10         require_once('sql_funcs.php');
11         $connection = null;
12         new_connection($connection);
13
14         $sql = "SELECT DISTINCT quadrant, number_
15                 FROM teeth AS t
16                 WHERE NOT EXISTS(SELECT 1
17                                     FROM procedure_charting AS p
18                                     WHERE VAT = :condoc
19                                     AND date_timestamp = :condate
20                                     AND t.quadrant = p.quadrant
21                                     AND t.number_ = p.number_
22                                     AND p.name = :pname)";
23
24         $result = sql_secure_query($connection, $sql, Array(":
condoc" => $_REQUEST['appdvat'],
25                                                         ":
condate" => $_REQUEST['appdate'],
26                                                         ":
pname" => $_REQUEST['pro_name']));
27         foreach($result as $row){
28             $quad = $row['quadrant'];
29             $numb = $row['number_'];
30             $append = $quad."!". $numb;
31             echo("<option value=\"\$append\"> Quadrant:$quad ||
Number:$numb </option> ");
32         }
33         ?>
34         </select>
35     </p>
36     <p> Measure: <input type="text" name="measure"/> <span style="margin
-left:8px;">mm</span></p>
37     <p> Description: <input type="text" name="descr"/></p>
38     <p><input type="hidden" id="pro_name" name="pro_name" value="<?echo(
$_REQUEST['pro_name'])?>" /> </p>
39     <p><input type="hidden" id="appcvat" name="appcvat" value="<?echo(
$_REQUEST['appcvat'])?>" /> </p>
40     <p><input type="hidden" id="appdvat" name="appdvat" value="<?echo(
$_REQUEST['appdvat'])?>" /> </p>

```

```

41         <p><input type="hidden" id="appdate" name="appdate" value="<?echo(
$_REQUEST['appdate'])?>" /> </p>
42         <p><input type="submit" value="Insert Procedure"/></p>
43     </form>
44 </body>
45 <html>
46
47 <?php
48     echo("<a href=\"newcharting.php?appcvat=\"");
49     echo($_REQUEST['appcvat']);
50     echo("&appdvat="");
51     echo($_REQUEST['appdvat']);
52     echo("&appdate="");
53     echo($_REQUEST['appdate']);
54     echo("\>Go to the Previous Page</a><br>");
55     echo("<a href=\"showconsultation.php?appcvat=\"");
56     echo($_REQUEST['appcvat']);
57     echo("&appdvat="");
58     echo($_REQUEST['appdvat']);
59     echo("&appdate="");
60     echo($_REQUEST['appdate']);
61     echo("\>Go to Consultation Page</a><br>");
62
63
64 ?>

```

3.1.17 insertcharting.php

```

1 <?php
2     require_once('sql_funcs.php');
3
4     $connection = null;
5     new_connection($connection);
6
7     $sql = "SELECT * FROM consultation AS c WHERE c.VAT_doctor = :dvat
8             AND c.date_timestamp = :ddate";
9     $result = sql_secure_query($connection, $sql, Array(":dvat" => $_REQUEST['
appdvat'],
10                                     ":ddate" => $_REQUEST['
appdate']));

```

```

$rows = $result->rowCount();

if($rows == 0){

    echo("<strong>Insert a nurse before inserting a Dental Charting
    Procedure.</strong><br>");
    echo("<strong>Its mandatory that every consultation has a nurse.</strong>
    ><br>");

    $connection = null;
    echo("<a href=\"newdata.php?appcvat=\"");
    echo($_REQUEST['appcvat']);
    echo("&appdvat="");
    echo($_REQUEST['appdvat']);
        echo("&appdate="");
    echo($_REQUEST['appdate']);
    echo("<\>Insert More Data for this Consultation</a><br>");
    echo("<a href=\"newcharting.php?appcvat=\"");
    echo($_REQUEST['appcvat']);
    echo("&appdvat="");
    echo($_REQUEST['appdvat']);
    echo("&appdate="");
    echo($_REQUEST['appdate']);
    echo("<\>Go to the Previous Page</a>");

}

else{

    $sql = "SELECT * FROM procedure_in_consultation AS c WHERE c.name = :
pname

                                AND          c.VAT_doctor = :dvat

AND c.date_timestamp = :ddate";
    $result = sql_secure_query($connection, $sql, Array(":pname" =>
$_REQUEST['pro_name'],

                                ":dvat" => $_REQUEST['appdvat'],

                                ":ddate" =>

```



```

$_REQUEST['appdate']));
43     $nrows = $result->rowCount();
44     $sql = "USE ist187005";
45     sql_secure_query($connection, $sql);
46     $connection->beginTransaction();
47
48     if($nrows == 0){
49         $sql = "INSERT INTO procedure_in_consultation VALUES (:pname, :
50         dvat, :ddate, '')";
51         $result = sql_secure_query($connection, $sql, Array(":pname" =>
52         $_REQUEST['pro_name'],
53         ":dvat" => $_REQUEST['appdvat'],
54         ":ddate" =>
55         $_REQUEST['appdate']));
56         if ($result == FALSE){
57             $connection -> rollback();
58             $info = $connection->errorInfo();
59             echo("<p>Error: {$info[2]}</p>");
60             exit();
61         }
62     }
63     $teeth = explode("!", $_REQUEST['teeth']);
64     $sql = "SELECT * FROM procedure_charting AS p WHERE p.name = :pname
65     AND p.
66     VAT = :dvat
67     AND p.
68     date_timestamp = :ddate
69     AND p.
70     quadrant = :dquad
71     AND p.
72     number_ = :dnumb";
73     $result = sql_secure_query($connection, $sql, Array(":pname" =>
74     $_REQUEST['pro_name'],
75     ":dvat" => $_REQUEST['appdvat'],
76     ":ddate" =>
77     $_REQUEST['appdate'],

```

```

70         ":dquad" => $teeth[0],
                                                    ":dnumb" =>
71     $teeth[1]));
72     $nrows = $result->rowCount();
73     if($nrows == 0){
74         $teeth = explode("!", $_REQUEST['teeth']);
75         $sql = "INSERT INTO procedure_charting VALUES (:pname, :dvat, :
76         ddate, :dquad, :dnumb, :ddesc, :dmeasure)";
77         $result = sql_secure_query($connection, $sql, Array(":pname" =>
78         $_REQUEST['pro_name'],
79
80         ":dvat" => $_REQUEST['appdvat'],
81         ":ddate" =>
82         $_REQUEST['appdate'],
83
84         ":dquad" => $teeth[0],
85         ":dnumb" =>
86         $teeth[1],
87
88         ":ddesc" =>
89         $_REQUEST['descr'],
90
91         ":dmeasure" =>
92         $_REQUEST['measure']));
93     if ($result == FALSE){
94         $connection -> rollback();
95         $info = $connection->errorInfo();
96         echo("<p>Error: {$info[2]}</p>");
97         exit();
98     }
99     $sql = "SELECT * FROM procedure_charting AS p WHERE p.name = :
100     pname
101
102     AND p.VAT = :dvat
103
104     AND p.date_timestamp = :ddate
105
106     AND p.quadrant = :dquad
107
108     AND p.number_ = :dnumb";
109     $result = sql_secure_query($connection, $sql, Array(":pname" =>

```

```

$_REQUEST['pro_name'],
94
                                ":dvat" => $_REQUEST['appdvat'],
95
                                ":ddate" =>
$_REQUEST['appdate'],
96
                                ":dquad" => $teeth[0],
97
                                ":dnumb" =>
$teeth[1]));
98
    $nrows = $result->rowCount();
99
    echo("<p>You inserted a new dental charting procedure for the
desired consultation with the following values:</p>");
100
    }else{
101
        echo("<p>You already registered that dental charting procedure
:</p>");
102
    }
103
    $connection -> commit();
104
    echo("<table border=\"1\">");
105
    echo("<tr><td>name</td><td>VAT_doctor</td><td>date_timestamp</td>
<td>quadrant</td><td>number_</td><td>description</td><td>measure</td></tr>")
;
106
    foreach($result as $row){
107
        echo("<tr><td>");
108
        echo($row['name']);
109
        echo("</td><td>");
110
        echo($row['VAT']);
111
        echo("</td><td>");
112
        echo($row['date_timestamp']);
113
        echo("</td><td>");
114
        echo($row['quadrant']);
115
        echo("</td><td>");
116
        echo($row['number_']);
117
        echo("</td><td>");
118
        echo($row['desc_']);
119
        echo("</td><td>");
120
        echo($row['measure']);
121
        echo("</td></tr>\n");
122
    }
123
    echo("</table>\n");

```

```

124         $connection = null;
125         echo("<br>");
126         echo("<a href=\"newcharting.php?appcvat=\"");
127         echo($_REQUEST['appcvat']);
128         echo("&appdvat=");
129         echo($_REQUEST['appdvat']);
130         echo("&appdate=");
131         echo($_REQUEST['appdate']);
132         echo(">Insert More Data for a Dental Charting Procedure</a><br>");
133     }
134 ?>

```

3.1.18 sql_funcs.php

```

1 <?php
2
3 function new_connection(&$connection)
4 {
5     $host = "db.tecnico.ulisboa.pt";
6     $user = "ist187005";
7     $pass = "qesr8498";
8     $dsn = "mysql:host=$host;dbname=$user";
9     try
10    {
11        $connection = new PDO($dsn, $user, $pass);
12    }
13    catch(PDOException $exception)
14    {
15        echo("<p>ERROR: ");
16        echo($exception->getMessage());
17        echo("</p>");
18        exit();
19    }
20 }
21
22 function sql_secure_query($connection, $sql, Array $vals = [] )
23 {
24     $stmt = $connection->prepare($sql);
25
26     foreach($vals as $key => &$value)

```

```

27     $stmt->bindParam($key , $value);
28
29     if ($stmt->execute() == FALSE)
30     {
31         $info = $connection->errorInfo();
32         echo("<div class=\"alert alert-danger\"> <b>Error: </b>");
33         echo($stmt->errorInfo()[2]);
34         echo("</div>");
35         exit();
36     }
37
38     return $stmt ;
39 }
40
41 ?>

```

3.2 SQL code - Part 2

3.2.1 Trigger_1.sql

```

1 #-----< TRIGGER 1 >-----
2
3 /* Write a trigger to update the age of the clients according to the birth date
4 and the current date. The trigger should fire whenever a new appointment
5 for a client is inserted into the database.*/
6
7 DELIMITER $$
8
9 CREATE TRIGGER Update_Age AFTER INSERT ON appointment
10     FOR EACH ROW
11     BEGIN
12         UPDATE client
13         SET AGE = TIMESTAMPDIFF(YEAR, birth_date, CURDATE())
14         WHERE client.VAT = NEW.VAT_client;
15 END$$
16
17 DELIMITER ;
18
19
20 #-----< Tests done

```

```

>-----
21 INSERT INTO client VALUES (988444861, 'Trigger_1', '1902-10-1', 'Durham Cross', '
    Temecula', '8295-433', 'F', 9);
22 SELECT * FROM client WHERE VAT=988444861;
23 INSERT INTO appointment VALUES (977442106, '2020-1-4 7:00:00', 'Teeth hurt a lot
    ', 988444861);
24 SELECT * FROM client WHERE VAT=988444861;

```

3.2.2 Trigger_2.sql

```

1 #-----< TRIGGER 2
    >-----
2 #Write triggers to ensure that:
3
4 #------(a)
    -----

5 # an individual that is a receptionist or a nurse at the clinic cannot
    simultaneously be a doctor:
6
7 DROP TRIGGER IF EXISTS Doctor_Cannot_Be_receptionist_or_Nurse_I;
8 DELIMITER $$
9 CREATE TRIGGER Doctor_Cannot_Be_receptionist_or_Nurse_I BEFORE INSERT ON doctor
10     FOR EACH ROW
11     BEGIN
12         IF (EXISTS (SELECT nurse.VAT FROM nurse WHERE nurse.VAT = NEW.VAT))
13             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
                is an Nurse';
14         END IF;
15         IF (EXISTS (SELECT receptionist.VAT FROM receptionist WHERE receptionist
                .VAT = NEW.VAT))
16             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
                is an Receptionist';
17         END IF;
18 END$$
19 DELIMITER ;
20 DROP TRIGGER IF EXISTS Doctor_Cannot_Be_receptionist_or_Nurse_U;
21 DELIMITER $$
22 CREATE TRIGGER Doctor_Cannot_Be_receptionist_or_Nurse_U BEFORE UPDATE ON doctor
23     FOR EACH ROW

```

```

24 BEGIN
25 IF( EXISTS (SELECT nurse.VAT FROM nurse WHERE nurse.VAT = NEW.VAT))
26 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
VAT from a Doctor. Should first DELETE or UPDATE the Nurse then INSERT the
NEW VAT INTO Doctor.';
27 END IF;
28 IF( EXISTS (SELECT receptionist.VAT FROM receptionist WHERE receptionist
.VAT = NEW.VAT))
29 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
VAT from a Doctor. Should first DELETE or UPDATE the Nurse then INSERT the
NEW VAT INTO Doctor.';
30 END IF;
31 END$$
32 DELIMITER ;
33 #-----< Tests done
>-----
34 # Cria uma nurse, um receptionist e um doctor para n o alterar os
valores do populate
35 INSERT INTO employee VALUES (998442200, 'Trigger_2_nurse', '1991-10-7', '
Broom Banks', 'Estersnow', '4429-551', 20, 2000);
36 INSERT INTO nurse VALUES (998442200);
37 INSERT INTO employee VALUES (998442201, 'Trigger_2_receptionist', '
1991-10-7', 'Broom Banks', 'Estersnow', '4429-551', 21, 2000);
38 INSERT INTO receptionist VALUES (998442201);
39 INSERT INTO employee VALUES (998442202, 'Trigger_2_doctor', '1991-10-7', '
Broom Banks', 'Estersnow', '4429-551', 22, 2000);
40
41 # Insere um doctor que j uma nurse, uma receptionist, e um que n o
nada ainda
42 INSERT INTO doctor VALUES (998442200, 'Orthodontist', 'Description.', '
email20@dentesbrilhantes.com.br');
43 SELECT * FROM employee NATURAL JOIN nurse WHERE VAT=998442200;
44 INSERT INTO doctor VALUES (998442201, 'Orthodontist', 'Description.', '
email21@dentesbrilhantes.com.br');
45 SELECT * FROM employee NATURAL JOIN receptionist WHERE VAT=998442201;
46 INSERT INTO doctor VALUES (998442202, 'Orthodontist', 'Description.', '
email22@dentesbrilhantes.com.br');
47 SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
48

```

```

49      # Atualiza um doctor para um vat que      uma nurse ou uma receptionist
50      UPDATE doctor SET VAT = 998442201 WHERE VAT = 998442202;
51      UPDATE doctor SET VAT = 998442200 WHERE VAT = 998442202;
52      SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
53      SELECT * FROM employee NATURAL JOIN receptionist WHERE VAT=998442201;
54      SELECT * FROM employee NATURAL JOIN nurse WHERE VAT=998442200;
55 DROP TRIGGER IF EXISTS Nurse_Cannot_Be_Doctor_I;
56 DELIMITER $$
57 CREATE TRIGGER Nurse_Cannot_Be_Doctor_I BEFORE INSERT ON nurse
58     FOR EACH ROW
59     BEGIN
60         IF (EXISTS (SELECT doctor.VAT FROM doctor WHERE doctor.VAT = NEW.VAT))
61             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
        is an Doctor';
62         END IF;
63 END$$
64 DELIMITER ;
65 DROP TRIGGER IF EXISTS Nurse_Cannot_Be_Doctor_u;
66 DELIMITER $$
67 CREATE TRIGGER Nurse_Cannot_Be_Doctor_U BEFORE UPDATE ON nurse
68     FOR EACH ROW
69     BEGIN
70         IF (EXISTS (SELECT doctor.VAT FROM doctor WHERE doctor.VAT = NEW.VAT))
71             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
        VAT from a Nurse. Should first DELETE or UPDATE the Doctor then INSERT the
        NEW VAT INTO Nurse.';
72         END IF;
73 END$$
74 DELIMITER ;
75 #-----< Tests done
    >-----
76     INSERT INTO nurse VALUES (998442202);
77     SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
78     INSERT INTO nurse VALUES (998442200);
79     SELECT * FROM employee NATURAL JOIN nurse WHERE VAT=998442200;
80
81     UPDATE nurse SET VAT = 998442202 WHERE VAT = 998442200;
82     SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
83     SELECT * FROM employee NATURAL JOIN nurse WHERE VAT=998442200;

```



```

84 DROP TRIGGER IF EXISTS receptionist_Cannot_Be_Doctor_I;
85 DELIMITER $$
86 CREATE TRIGGER receptionist_Cannot_Be_Doctor_I BEFORE INSERT ON receptionist
87     FOR EACH ROW
88     BEGIN
89         IF (EXISTS (SELECT doctor.VAT FROM doctor WHERE doctor.VAT = NEW.VAT))
90             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This employee
          is an Doctor';
91         END IF;
92 END$$
93 DELIMITER ;
94 DROP TRIGGER IF EXISTS receptionist_Cannot_Be_Doctor_U;
95 DELIMITER $$
96 CREATE TRIGGER receptionist_Cannot_Be_Doctor_U BEFORE UPDATE ON receptionist
97     FOR EACH ROW
98     BEGIN
99         IF (EXISTS (SELECT doctor.VAT FROM doctor WHERE doctor.VAT = NEW.VAT))
100             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
          VAT from a receptionist. Should first DELETE or UPDATE the Doctor then INSERT
          the NEW VAT INTO Receptionist.';
101         END IF;
102 END$$
103 DELIMITER ;
104 #-----< Tests done
      >-----
105     INSERT INTO receptionist VALUES (998442202);
106     SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
107     INSERT INTO receptionist VALUES (998442201);
108     SELECT * FROM employee NATURAL JOIN receptionist WHERE VAT=998442201;
109
110     UPDATE receptionist SET VAT = 998442202 WHERE VAT = 998442201;
111     SELECT * FROM employee NATURAL JOIN doctor WHERE VAT=998442202;
112     SELECT * FROM employee NATURAL JOIN receptionist WHERE VAT=998442201;
113 #----- (b)
      -----

114 # doctors cannot simultaneously be trainees and permanent staff
115 DROP TRIGGER IF EXISTS PermanentDoctor_Cannot_Be_TraineeDoctor_I;
116 DELIMITER $$

```

```

117 CREATE TRIGGER PermanentDoctor_Cannot_Be_TraineeDoctor_I BEFORE INSERT ON
    permanent_doctor
118     FOR EACH ROW
119     BEGIN
120         IF (EXISTS (SELECT trainee_doctor.VAT FROM trainee_doctor WHERE
            trainee_doctor.VAT = NEW.VAT))
121             THEN DELETE
122                 FROM trainee_doctor
123                 WHERE trainee_doctor.VAT = NEW.VAT;
124         END IF;
125 END$$
126 DELIMITER ;
127 DROP TRIGGER IF EXISTS PermanentDoctor_Cannot_Be_TraineeDoctor_U;
128 DELIMITER $$
129 CREATE TRIGGER PermanentDoctor_Cannot_Be_TraineeDoctor_U BEFORE UPDATE ON
    permanent_doctor
130     FOR EACH ROW
131     BEGIN
132         IF (EXISTS (SELECT trainee_doctor.VAT FROM trainee_doctor WHERE
            trainee_doctor.VAT = NEW.VAT))
133             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
            VAT from a Permanent Doctor. Try INSERT INTO permanent_doctor (VAT, years)';
134         END IF;
135 END$$
136 DELIMITER ;
137 #-----< Tests done
    >-----
138     # Cria tr s doctor (1 permanent e 2 trainee) para n o alterar os
    valores do populate
139     INSERT INTO employee VALUES (998442203, 'Trigger_2_permanent', '1991-10-7',
    ', 'Broom Banks', 'Estersnow', '4429-551', 23, 2000);
140     INSERT INTO employee VALUES (998442204, 'Trigger_2_trainee', '1991-10-7',
    ', 'Broom Banks', 'Estersnow', '4429-551', 24, 2000);
141     INSERT INTO employee VALUES (998442205, 'Trigger_2_trainee', '1991-10-7',
    ', 'Broom Banks', 'Estersnow', '4429-551', 25, 2000);
142     INSERT INTO doctor VALUES (998442203, 'Orthodontist', 'Description.', '
    email23@dentesbrilhantes.com.br');
143     INSERT INTO doctor VALUES (998442204, 'Orthodontist', 'Description.', '
    email24@dentesbrilhantes.com.br');

```

```

144     INSERT INTO doctor VALUES (998442205, 'Orthodontist', 'Description.', '
email25@dentesbrilhantes.com.br');
145     INSERT INTO trainee_doctor VALUES (998442204, 977442106);
146     INSERT INTO trainee_doctor VALUES (998442205, 977442106);
147
148     # Insere um permanent doctor que j         um trainee, e um que n o
nada ainda
149     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor
WHERE VAT=998442203;
150     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor
WHERE VAT=998442204;
151     INSERT INTO permanent_doctor VALUES (998442203, 14);
152     INSERT INTO permanent_doctor VALUES (998442204, 3);
153     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor
WHERE VAT=998442203 OR VAT=998442204;
154
155     # Atualiza um permanent doctor que j         uma trainee, e um que n o
nada ainda
156     UPDATE permanent_doctor SET VAT = 998442205 WHERE VAT = 998442203;
157     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor
WHERE VAT=998442203;
158     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor
WHERE VAT=998442205;
159 DROP TRIGGER IF EXISTS TraineeDoctor_Cannot_Be_PermanentDoctor_I;
160 DELIMITER $$
161 CREATE TRIGGER TraineeDoctor_Cannot_Be_PermanentDoctor_I BEFORE INSERT ON
trainee_doctor
162     FOR EACH ROW
163     BEGIN
164         IF (EXISTS (SELECT permanent_doctor.VAT FROM permanent_doctor WHERE
permanent_doctor.VAT = NEW.VAT))
165             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This Doctor is
Permanent';
166         END IF;
167 END$$
168 DELIMITER ;
169 DROP TRIGGER IF EXISTS TraineeDoctor_Cannot_Be_PermanentDoctor_U;
170 DELIMITER $$
171 CREATE TRIGGER TraineeDoctor_Cannot_Be_PermanentDoctor_U BEFORE UPDATE ON

```

```

trainee_doctor
172     FOR EACH ROW
173     BEGIN
174         IF (EXISTS (SELECT permanent_doctor.VAT FROM permanent_doctor WHERE
permanent_doctor.VAT = NEW.VAT))
175             THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot UPDATE
VAT from a trainee_doctor. Should first DELETE the trainee_doctor then INSERT
the NEW VAT INTO trainee_doctor.';
176         END IF;
177 END$$
178 DELIMITER ;
179 #-----< Tests done
>-----
180     INSERT INTO client VALUES (988444861, 'Trigger_1', '1902-10-1', 'Durham
Cross', 'Temecula', '8295-433', 'F', 9);
181     SELECT * FROM client WHERE VAT=988444861;
182     INSERT INTO appointment VALUES (977442106, '2020-1-4 7:00:00', 'Teeth
hurt a lot', 988444861);
183     SELECT * FROM client WHERE VAT=988444861;
184     # Insere um permanent doctor que j      um trainee, e um que n o
nada ainda
185     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor
WHERE VAT=998442205;
186     INSERT INTO trainee_doctor VALUES (998442204, 3);
187     INSERT INTO trainee_doctor VALUES (998442205, 3);
188     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor
WHERE VAT=998442203 OR VAT=998442204 OR VAT=998442205;
189     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor
WHERE VAT=998442205;
190
191     # Atualiza um permanent doctor que j      uma trainee, e um que n o
nada ainda
192     UPDATE trainee_doctor SET VAT = 998442204 WHERE VAT = 998442205;
193     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN permanent_doctor
WHERE VAT=998442204;
194     SELECT * FROM employee NATURAL JOIN doctor NATURAL JOIN trainee_doctor
WHERE VAT=998442204 OR VAT=998442205;

```

3.2.3 Trigger_3.sql

```
1 -----
2 -----TRIGGER-----
3 -----
4
5 ----- TRIGGER -----
6 -- Checks when inserting new phone number for employee
7
8 DELIMITER ;
9
10 DROP TRIGGER IF EXISTS Phone_employee_I;
11
12 DELIMITER $$
13
14 CREATE TRIGGER Phone_employee_I BEFORE INSERT ON phone_number_employee
15 FOR EACH ROW
16 BEGIN
17     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
18         phone = NEW.phone))
19         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
20             number is already associated to an employee.';
21     END IF;
22     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
23         phone = NEW.phone))
24         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
25             number is already associated to a client.';
26     END IF;
27 END$$
28
29 DELIMITER ;
30
31 ----- TRIGGER -----
32 -- Checks when updating new phone number for employee (AND pne.VAT <> OLD.VAT)
33
34 DELIMITER ;
35
36 DROP TRIGGER IF EXISTS Phone_employee_U;
37
38 DELIMITER $$
39
```

```
36 CREATE TRIGGER Phone_employee_U BEFORE UPDATE ON phone_number_employee
37 FOR EACH ROW
38 BEGIN
39     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
40         phone = NEW.phone AND pne.VAT <> OLD.VAT))
41         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
42         number is already associated to an employee.';
43     END IF;
44     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
45         phone = NEW.phone))
46         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
47         number is already associated to a client.';
48     END IF;
49 END$$
50 DELIMITER ;
51
52 ----- TRIGGER -----
53 -- Checks when inserting new phone number for client
54
55 DELIMITER ;
56
57 DROP TRIGGER IF EXISTS Phone_client_I;
58
59 DELIMITER $$
60
61 CREATE TRIGGER Phone_client_I BEFORE INSERT ON phone_number_client
62 FOR EACH ROW
63 BEGIN
64     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
65         phone = NEW.phone))
66         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
67         number is already associated to an employee.';
68     END IF;
69     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
70         phone = NEW.phone))
71         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
72         number is already associated to a client.';
73     END IF;
```

```

67 END$$
68
69 DELIMITER ;
70
71 ----- TRIGGER -----
72 -- Checks when updating new phone number for client (AND pnc.VAT <> OLD.VAT)
73
74 DELIMITER ;
75
76 DROP TRIGGER IF EXISTS Phone_client_U;
77
78 DELIMITER $$
79
80 CREATE TRIGGER Phone_client_U BEFORE UPDATE ON phone_number_client
81 FOR EACH ROW
82 BEGIN
83     IF (EXISTS (SELECT pne.phone FROM phone_number_employee pne WHERE pne.
84         phone = NEW.phone))
85         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
86         number is already associated to an employee.';
87     END IF;
88     IF (EXISTS (SELECT pnc.phone FROM phone_number_client pnc WHERE pnc.
89         phone = NEW.phone AND pnc.VAT <> OLD.VAT))
90         THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'This phone
91         number is already associated to a client.';
92     END IF;
93 END$$
94
95 DELIMITER ;

```

3.2.4 Function1.sql

```

1 -----
2 -----FUNCTION-----
3 -----
4
5 -- variable count: count for clients of given gender, age and year
6 -- variable s_count: count for clients of given gender, age and year that showed
7 -- count - s_count: count for clients of given gender, age and year thar didn't
   showed

```

```
8
9 DELIMITER ;
10
11 DROP FUNCTION IF EXISTS no_shows_count;
12
13 DELIMITER $$
14
15 CREATE FUNCTION no_shows_count(gender VARCHAR(15), lower_age INTEGER, upper_age
    INTEGER, year INTEGER)
16     RETURNS INTEGER
17 BEGIN
18     DECLARE count INTEGER;
19     DECLARE s_count INTEGER;
20     SELECT count(*) INTO count
21     FROM appointment a1, client cl
22     WHERE a1.VAT_client = cl.VAT
23     AND cl.gender = gender
24     AND cl.age BETWEEN lower_age AND upper_age
25     AND year(a1.date_timestamp) = year;
26     SELECT count(*) INTO s_count
27     FROM client cl, consultation c LEFT JOIN appointment a
28     ON (a.VAT_doctor = c.VAT_doctor AND a.date_timestamp = c.date_timestamp)
29     WHERE a.VAT_client = cl.VAT
30     AND cl.gender = gender
31     AND cl.age BETWEEN lower_age AND upper_age
32     AND year(a.date_timestamp) = year;
33     RETURN count - s_count;
34 END $$
35
36 DELIMITER ;
37
38 SELECT no_shows_count('F', 80, 100, 2019) AS no_shows_count;
39
40 ----- QUERY TO CHECK RESULT: -----
41
42 -- Clients of given gender, age and year that missed the appointment
43
44 SELECT a1.VAT_client, cl.name, cl.gender, cl.age, a1.date_timestamp, a1.
    VAT_doctor
```



```

45 FROM client cl, appointment a1
46 WHERE a1.VAT_client = cl.VAT
47 AND cl.gender = 'F'
48 AND cl.age BETWEEN 80 AND 100
49 AND year(a1.date_timestamp) = 2019
50 AND (a1.VAT_doctor, a1.date_timestamp) NOT IN (
51     SELECT c.VAT_doctor, c.date_timestamp
52     FROM consultation c LEFT JOIN appointment a2
53     ON (a2.VAT_doctor = c.VAT_doctor AND a2.date_timestamp = c.
        date_timestamp));

```

3.2.5 Procedure1.sql

```

1  -----
2  -----PROCEDURE-----
3  -----
4
5  DELIMITER ;
6
7  DROP PROCEDURE IF EXISTS change_salary;
8
9  DELIMITER $$
10
11 CREATE PROCEDURE change_salary(IN n_years INTEGER)
12 BEGIN
13     UPDATE employee e
14     SET e.salary = 1.05 * e.salary
15     WHERE (SELECT count(*) FROM consultation c WHERE e.VAT = c.VAT_doctor
        AND year(c.date_timestamp) = year(CURDATE()) GROUP BY c.VAT_doctor) <= 100
16     AND (SELECT pd.years FROM permanent_doctor pd WHERE e.VAT = pd.VAT) >
        n_years;
17     UPDATE employee e
18     SET e.salary = 1.1 * e.salary
19     WHERE (SELECT count(*) FROM consultation c WHERE e.VAT = c.VAT_doctor
        AND year(c.date_timestamp) = year(CURDATE()) GROUP BY c.VAT_doctor) > 100
20     AND (SELECT pd.years FROM permanent_doctor pd WHERE e.VAT = pd.VAT) >
        n_years;
21 END$$
22
23 DELIMITER ;

```

24

25 `CALL change_salary(10);`