INFO0009-2 Databases 2024-2025 *(project part 2)*

TRANSLATED WITH GOOGLE TRANSLATE, USE AT YOUR OWN RISK AND ASK FOR CLARIFICATIONS IF NECESSARY.

You have a simplified version of the database that you had to design:

- We only have one type of stop (stations)
- We gave an artificial key to the services
- We gave an artificial key to the agencies, **the names of the agencies remain unique**.

• AGENCE(ID,NOM,URL,*FUSEAU_HORAIRE*,TELEPHONE,SIEGE) • ARRET_DESSERVI(ITINERAIRE_ID,ARRET_ID,SEQUENCE) • ARRET(ID,NOM,LATITUDE,*LONGITUDE*) • EXCEPTION(SERVICE_ID,DATE,CODE) • HORRAIRE(TRAJET_ID,ITINERAIRE_ID,ARRET_ID,*HEURE_ARRIVEE*,HEURE_DEPART) • ITINERAIRE(ID,AGENCE_ID,TYPE,*NOM*) • LANGUEPRINCIPALE(AGENCE_ID,LANGUE) • SERVICE(ID,NOM,LUNDI,*MARDI*,MERCREDI,JEUDI,VENDREDI,SAMEDI,DIMANCHE,DATE_DEBUT,DATE_FIN) • TRAJET(TRAJET_ID,SERVICE_ID,ITINERAIRE_ID,*DIRECTION*)	• AGENCY(ID,NAME,URL,*TIME_ZONE*,TELEPHONE,SIEGE) • STOP_SERVICED(ITINERAIRE_ID,SCAR_ID,SEQUENCE) • STOP(ID,NAME,LATITUDE,*LONGITUDE*) • EXCEPTION(SERVICE_ID,DATE,CODE) • SCHEDULE(ROUTE_ID,ITINERAIRE_ID,SCAR_ID,*ARRIVAL_TIME*,DEPARTURE_TIME) • ITINERART(ID,AGENCE_ID,TYPE,*NAME*) • MAIN_LANGUAGE(AGENCE_ID,LANGUAGE) • SERVICE(ID,NAME,MONDAY,*TUESDAY*,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY,DATE_DEBUT,END_DATE) • ROUTE(ROUTE_ID,SERVICE_ID,ITINERAIRE_ID,*DIRECTION*)

Mission

Scores : 0) Non-existent - 1) Insufficient - 2) Less than adequate - 3) Adequate - 4) Good - 5) Excellent - 6) Exceeds expectations	Weighting	Score (out of 5):

Write a script to initialize the database (creating tables, views, integrity constraints, etc.) on a MySQL server using the attached CSV files. You can either fill the tables manually (INSERT) or by loading the files (LOAD). All lines end with a *linefeed*. Examine the contents of the file, as some values are NULL (read part 1 of the project for details). Documentation LOAD : https://dev.mysql.com/doc/refman/8.4/en/load-data.html Index : `IGNORE 1 ROWS;`	2	
For each table in the list below, create a form to select and display its tuples by limiting the value of one or more of their attributes: AGENCY, SCHEDULE, and EXCEPTION. These constraints are limited to the constraints of capacity (i.e., "Gaston" contains "on") for variables that are strings (name, for example) and equality constraints for fields that are numbers or dates. When multiple constraints are provided, tuples must satisfy them all.	2	

Create a page with a form for adding a new service. The service identifier must be based on an automatically incremented sequence. The user must be able to check the days of the week of the service. The form also contains a text box for indicating exceptions. Each line corresponds to an exception. For example :
 	
`2025-04-20 INCLUDED`
`2025-04-21 INCLUDED`
`2025-04-22 EXCLUDED`
 	
Indices :
- https://www.phptutorial.net/php-tutorial/php-checkbox/
- $parts = explode(" ", trim($line))
- Pay attention to exception and transaction management
	2	

SERVICE represents all days between a start date and an end date. For example, WEEKDAY contains all days Monday through Friday between 01/01/2025 and 12/31/2031. Create a page that, for each date, displays the services (names) available. The query must rely on views: a recursive view that calculates all the dates of a service between the start date and the end date (therefore ignoring exceptions and unchecked days) and a second view that combines the first and takes into account exceptions and unchecked days.[1] I.e., a recursive view should return all dates between the start date and the end date of a service. For example, DAY OF WEEK dates include Saturdays and Sundays. They will be excluded from the calculation of the second view.	2	

[1] You can try to solve this problem in a single view, but it's ~~A little~~ more difficult.

Indices :

- DATE_ADD("2017-06-15", INTERVAL 10 DAY) https://www.w3schools.com/sql/func_mysql_date_add.asp
- MySQL GROUP_CONCAT function

A page where you show the average stopping time per trip and per route. This page presents the routes classified by name and their averages. You should also include the average stopping time of all trips for each route, as well as the average stopping time of all trips recorded in our system.

ITINERARY	ROUTE	AVG_STOP_TIME
IT1	1	average stopping time of this trip
IT1	2	average stopping time of this trip
IT1		average stopping time of all T1 trips
IT2	1	average stopping time of this trip
IT2	2	average stopping time of this trip
IT2		average stopping time of all T2 trips
		average stopping time of all trips

These aggregations can be obtained with separate queries (which requires transaction management), but you can also use ROLLUP to create an additional row with totals in the result set.

2

A page with a form allowing me to search for station names containing a given string. The search should not be case-sensitive. The form also contains a field that takes a number, which is optional. When searching, the page displays all stations including this chain and the number of trains that stop, arrive and depart there by department. The results must be presented in descending order of the number of stops, arrivals and departures. If a (valid) number is given, the results should only contain lines where the number of stops, arrivals or departures is at least equal to this number.

There are 632 stations and 3 services. Be careful not to run this query with an empty string, because you will get 1896 rows. ;-)

Index : LOWER https://www.w3schools.com/sql/func_mysql_lower.asp

2

A page with two forms. The first allows me to select a route to delete from a drop-down list. If a route is deleted, all planned trips are deleted as well.

The second allows me to select a route and direction. This will generate a new form to add a new trip for that route and direction. The user will then be able to enter arrival and departure times depending on the direction. You can use a text box containing a “CSV” format:

3

STOP1,hdepart1 STOP2,harrivé2,hdepart2 STOP3,harrivé3,hdepart3 STOP4,harrivé4 YOU <u>can</u> Also use tables in forms. Hint : <input type="text" name="varname[0]" > <input type="text" name="varname[1]" > <input type="text" name="varname[1]" > Pay attention to exception and transaction management. And integrity constraints, of course!		
Create a page to select a stop and edit its information, including its ID. These changes must be reflected. Make sure that the contact details are indeed in Belgium. This constraint will be simulated using the *bounding box* of Belgium: upper left = (2.51357303225, 49.5294835476) and lower right = (6.15665815596, 51.4750237087).	3	
Document: A technical report containing a description of your website architecture, a description of the operations you need to perform to initialize the database from the scripts you submitted, and a description of your queries, transactions, and any informed decisions regarding your implementation.	2	
	Grade:	0

Important

You can divide the different pages between you. Keep in mind that some pages may rely on (very) similar (sub)queries. If so, you might consider using named views. If a query has multiple occurrences of the same subquery, then you should consider using WITH clauses. **<u>Warning: the Transaction management and avoiding SQL injection are important.</u>** If you have multiple operations when loading a page (GET or POST), you (most likely) need to consider transaction handling.

Participation in the project is mandatory. Students who do not submit anything for the project, including a blank report and/or little or no source code, will receive a grade of absence (A) for the course. This project must be carried out as a group. As a reminder, plagiarism is severely punished. I also remind students to consult the "Charter for the use of artificial intelligence tools by students". You are encouraged to discuss ideas and approaches. with your peers, but you are not allowed to share your code.

Submission

This second part of the project must be carried out in groups of 3 students (the same groups from the first part). **The student who does not follow the instructions and who submits an incomplete project alone or in a group will receive an A for absence, because teamwork is one of the skills developed in this project.**

For this project, we ask you to submit, before **May 9, 2025 at 11:59 p.m**, 2 files: a ZIP archive, which will contain all the scripts you used for your project, as well as a report in **PDF**. **Warning: other compression formats inside a ZIP, compression formats renamed to ZIP, ... will all be considered invalid submissions. Make sure to download the PDF and ZIP as two separate files. This allows us to add comments to the PDF on eCampus.**

Regarding the ZIP, this means that you must submit: 1) The script(s) allowing you to initialize the database (initialization of the tables and, possibly, filling them), 2) all the scripts and data making up your website, and 3) the Docker configuration to launch your container.

The report will contain: 1) A description of the architecture of your website, 2) A description of the manipulations you need to perform to initialize the database from the scripts you submitted. 3) A description of the queries used to answer the questions. You can use an example of a query replacing variables with values. And 4), a brief description of the distributions of roles or tasks.

Use of `textarea`

```php
<form method="post" action="exemple-form.php">
    <textarea rows="5" name="text"></textarea>
    <br/>
    <input type="submit" value="Submit">
</form>
<?php
    if (isset($_POST['text'])) {
        $lines = explode("\n", $_POST['text']);
        // If the array is not empty
        if($lines){
            echo '<ul>';
            foreach ($lines as $line) {
                // Trim whitespaces, empty strings are considered false
                $x = trim($line);
                if ($x) echo '<that>'. $x .'</li>';
            }
            echo '</ul>';
        }
    }
?>
```