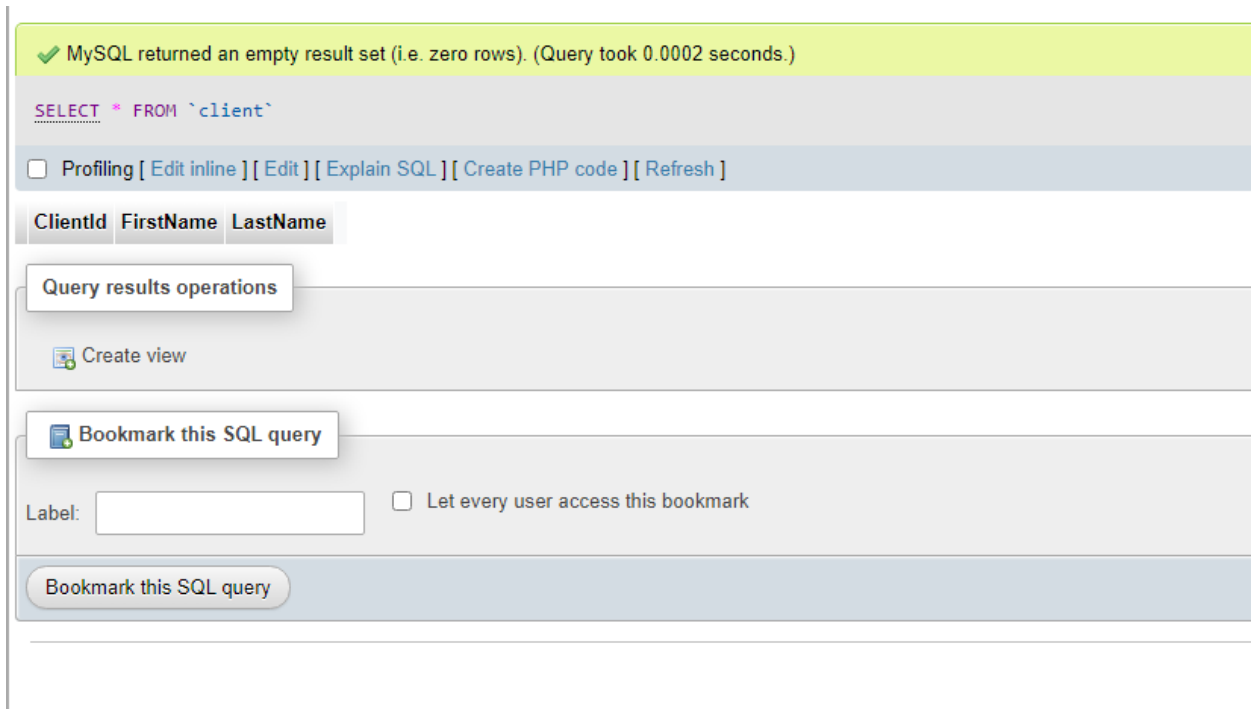


Before creating a client:




✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)


```
SELECT * FROM `client`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

ClientId	FirstName	LastName
----------	-----------	----------

Query results operations

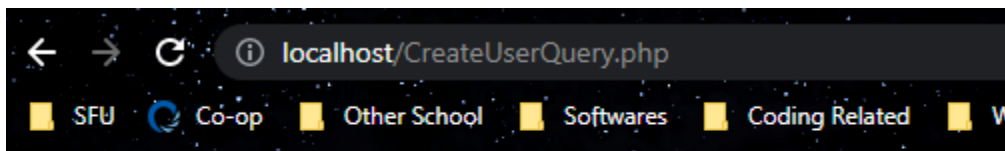
 Create view

 Bookmark this SQL query

Label: ☐ Let every user access this bookmark

[Bookmark this SQL query](#)

Created user:



Successfully created User Account with id number: 123

After creating user:

✓ Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

`SELECT * FROM `client``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 ▾ Filter rows:

Extra options

← T → ▾ ClientId FirstName LastName

☐ Edit Copy Delete 123 jac sekhon

↶ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 ▾ Filter rows:

Query results operations

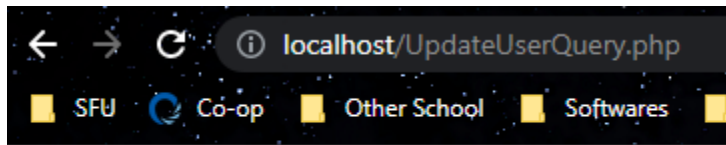
Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

Updating user information:



Successfully updated your first and last name

Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

```
SELECT * FROM `client`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows:

Extra options

	ClientId	FirstName	LastName
<input type="checkbox"/> Edit Copy Delete	123	green	purple

Before creating schedule:


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SELECT * FROM `creates_schedule`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

ScheduleId	ClientId	ScheduleStartDate	ScheduleEndDate
------------	----------	-------------------	-----------------

Query results operations

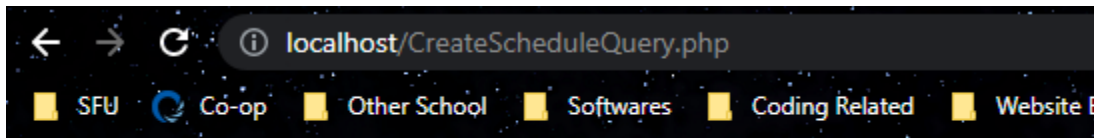
 Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

Creating schedule:



Schedule 1 successfully created for user 123 from 2022-08-02 to 2022-08-06

After creating schedule:

✓ Showing rows 0 - 0 (1 total, Query took 0.0001 seconds.)

`SELECT * FROM `creates_schedule``

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▾ Filter rows:

Extra options

			ScheduleId	ClientId	ScheduleStartDate	ScheduleEndDate	
<input type="checkbox"/>	Edit	Copy	Delete	1	123	2022-08-02	2022-08-06

☐ Check all

With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 ▾ Filter rows:

Before filling schedule 1 with cities:


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)


```
SELECT * FROM `filledwith`
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

ScheduleId	CityName
------------	----------

Query results operations

 Create view

 Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

Filling with city: Agra

← → ↻ ⓘ localhost/AddToScheduleQuery.php

SFU

Co-op

Other School

Softwares

Coding Related

Successfully added city 'Agra' to Schedule id '1'

After filling schedule 1 with Agra

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `filledwith`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▾ Filter rows:

Extra options

← T → ▾ ScheduleId CityName

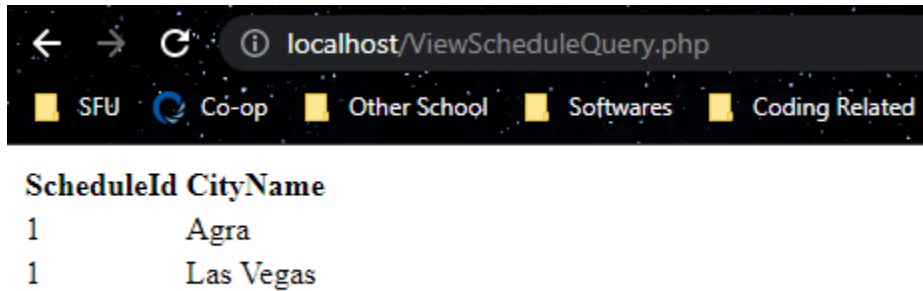
☐ [Edit](#) [Copy](#) [Delete](#) 1 Agra

↑ ☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 ▾ Filter rows:

Before viewing schedule: add in las vegas to schedule 1

View of schedule 1:

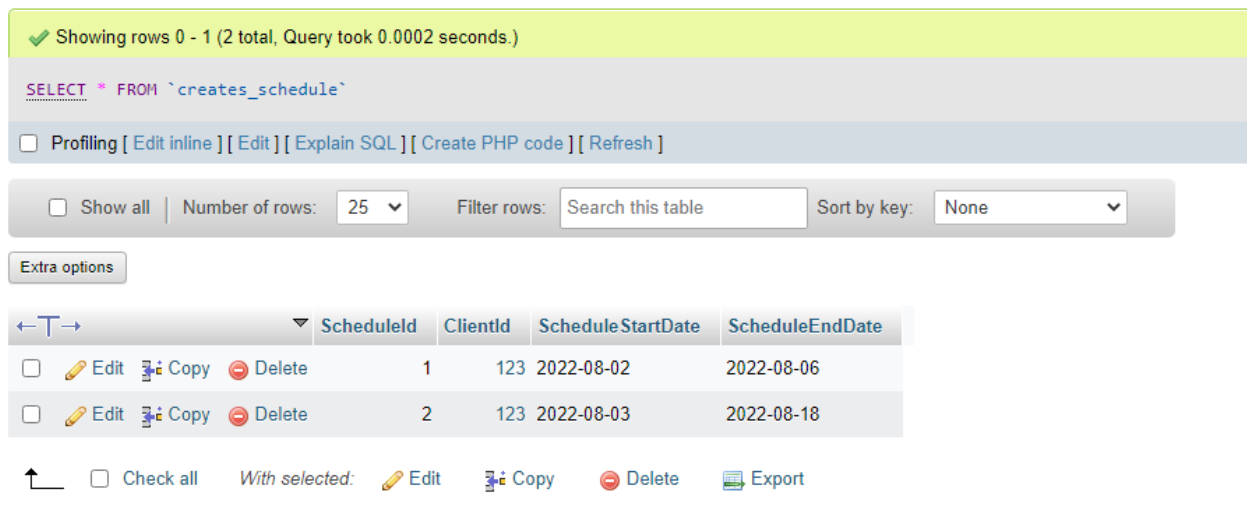


The screenshot shows a web browser at the URL `localhost/ViewScheduleQuery.php`. Below the address bar is a navigation bar with five items: SFU, Co-op, Other School, Softwares, and Coding Related. The main content area displays a table with two columns: **ScheduleId** and **CityName**. The table contains two rows: one with **ScheduleId** 1 and **CityName** Agra, and another with **ScheduleId** 1 and **CityName** Las Vegas.

ScheduleId	CityName
1	Agra
1	Las Vegas

Deleting Client

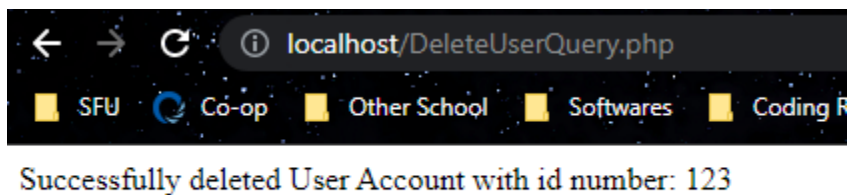
Before deleting add another scheduleID = 2



The screenshot shows a web browser displaying a query result for the `creates_schedule`` table. The query is `SELECT * FROM `creates_schedule``. The result shows two rows. The first row has **ScheduleId** 1, **ClientId** 123, **ScheduleStartDate** 2022-08-02, and **ScheduleEndDate** 2022-08-06. The second row has **ScheduleId** 2, **ClientId** 123, **ScheduleStartDate** 2022-08-03, and **ScheduleEndDate** 2022-08-18. The interface includes a status bar at the top indicating 2 rows shown, a toolbar with options like Edit, Copy, and Delete, and a table with columns for ScheduleId, ClientId, ScheduleStartDate, and ScheduleEndDate.

	ScheduleId	ClientId	ScheduleStartDate	ScheduleEndDate
<input type="checkbox"/> Edit Copy Delete	1	123	2022-08-02	2022-08-06
<input type="checkbox"/> Edit Copy Delete	2	123	2022-08-03	2022-08-18

Now delete the client and watch cascade occur:



The screenshot shows a web browser at the URL `localhost/DeleteUserQuery.php`. Below the address bar is a navigation bar with five items: SFU, Co-op, Other School, Softwares, and Coding R. The main content area displays a message: "Successfully deleted User Account with id number: 123".

Successfully deleted User Account with id number: 123


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SELECT * FROM `client`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

ClientId	FirstName	LastName
----------	-----------	----------

Query results operations

 [Create view](#)


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SELECT * FROM `creates_schedule`
```

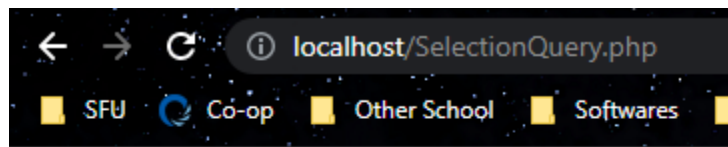
☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

ScheduleId	ClientId	ScheduleStartDate	ScheduleEndDate
------------	----------	-------------------	-----------------

Query results operations

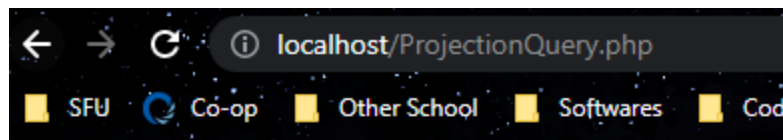
 [Create view](#)

Selection Query for showing Currency name and exchange rate where exchange rate < 1



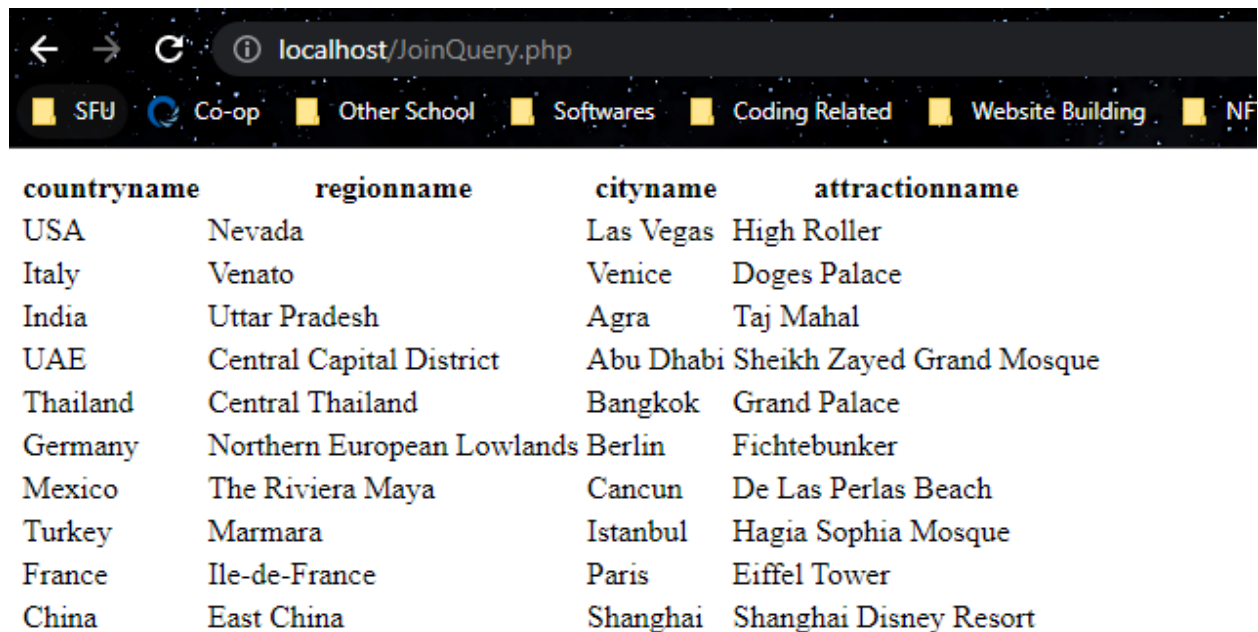
CurrencyName	exchangerate
Bitcoin	0.000034
Euro	0.76
United States Dollar	0.78

Projection Query for showing the list of all the countries available in the database



countryname
China
France
Germany
India
Italy
Mexico
Thailand
Turkey
UAE
USA

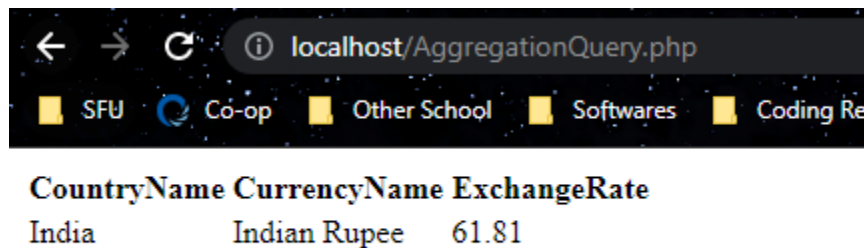
Join query for joining tables: countryhasaregion, city_hasa and attraction_hasa_event1:



The screenshot shows a web browser window with the address bar displaying 'localhost/JoinQuery.php'. Below the browser window, a table displays the results of a join query. The table has four columns: 'countryname', 'regionname', 'cityname', and 'attractionname'. The data is as follows:

countryname	regionname	cityname	attractionname
USA	Nevada	Las Vegas	High Roller
Italy	Veneto	Venice	Doges Palace
India	Uttar Pradesh	Agra	Taj Mahal
UAE	Central Capital District	Abu Dhabi	Sheikh Zayed Grand Mosque
Thailand	Central Thailand	Bangkok	Grand Palace
Germany	Northern European Lowlands	Berlin	Fichtebunker
Mexico	The Riviera Maya	Cancun	De Las Perlas Beach
Turkey	Marmara	Istanbul	Hagia Sophia Mosque
France	Ile-de-France	Paris	Eiffel Tower
China	East China	Shanghai	Shanghai Disney Resort

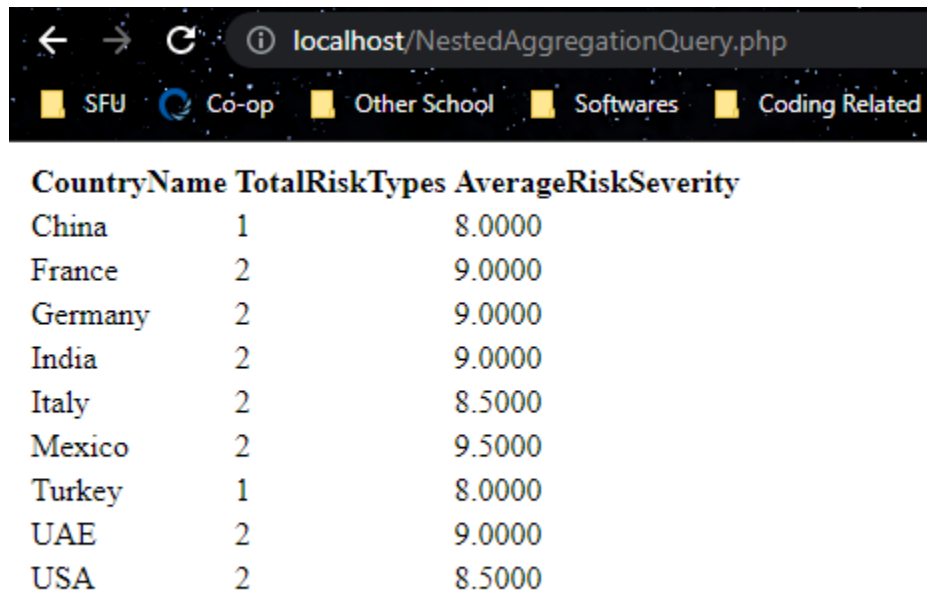
Aggregate query for showing the country and exchange rate of the country that has the highest cad->local currency exchange rate



The screenshot shows a web browser window with the address bar displaying 'localhost/AggregationQuery.php'. Below the browser window, a table displays the results of an aggregate query. The table has three columns: 'CountryName', 'CurrencyName', and 'ExchangeRate'. The data is as follows:

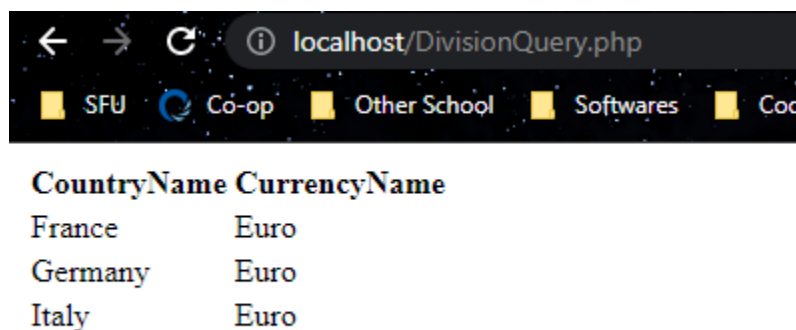
CountryName	CurrencyName	ExchangeRate
India	Indian Rupee	61.81

Nested aggregate query for showing the average risk type and number of risks that are above 7/10 severity rating



CountryName	TotalRiskTypes	AverageRiskSeverity
China	1	8.0000
France	2	9.0000
Germany	2	9.0000
India	2	9.0000
Italy	2	8.5000
Mexico	2	9.5000
Turkey	1	8.0000
UAE	2	9.0000
USA	2	8.5000

Division query to show all countries that use the Euro and filter out all that do not



CountryName	CurrencyName
France	Euro
Germany	Euro
Italy	Euro