

# Cryptocurrency mining business

Nouha Lamrani

DATABASES COURSWORK 3

## Table of contents

1.Business system summary:	2
2. An overview of the business environment and my work objectives:	2
3. Justification how to develop the database to meet the objectives, and how it will support the business as a working DBMS:	3
4.ERD of my idea:	4
4.Database relationship diagram:	4
5.Screenshots of sample data:	5
6.Demonstration of the scenario	20
7.Demonstration of the scenario	36

## 1. Business system summary:

Since 2009, the substitute world of economy has been progressively increasing due to the major role of cryptocurrencies. Which is the reason why I chose to pick a cryptocurrency mining business, regardless of the difficulties in understanding the system that should be used. As it is a popular subject that many people talk about from one side, and others that totally ignore it but want to know more about it. In my report, I'll present the cryptocurrency data processing structure, transactions, blocks, blockchain network, social media and exchange market, to demonstrate the relationship between all of them and how all the system works.

## 2. An overview of the business environment and my work objectives:

This business idea needs a database management system to determine the different types of cryptocurrencies (Bitcoin (BTC) - Ethereum (ETH) - Solana (SOL) ...), that have different prices depending on the worldwide Crypto Exchange. All this Altcoins got popularized through social media such as Twitter. They were sold and bought by different users all over the world producing different and unique transactions added to the blockchain that empower the existence of cryptocurrency.

My work objectives are:

1. Confirm the impact of the world currency exchange on the cryptocurrencies 's prices.
2. Determine the major role of blockchain for cryptocurrency existence.
3. The influence of networking and social media on the growth of digital currency.

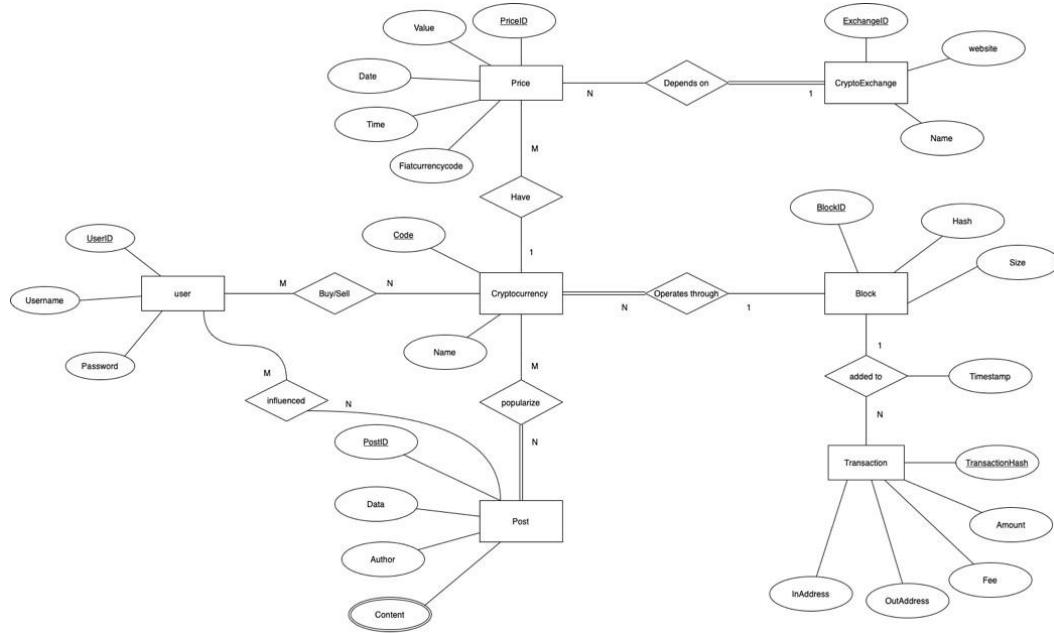
### 3. Justification how to develop the database to meet the objectives, and how it will support the business as a working DBMS:

To justify my objectives, and in order to confirm the impact of the world currency exchange on the cryptocurrencies 's prices, this digital currency is sold or bought by a user that has a unique username and password, at a specific price that varies depending on the date, time and the crypto market exchange available on different websites. In the database, relationships between the cryptocurrency with user, price and market exchange need to be created to support this confirmation.

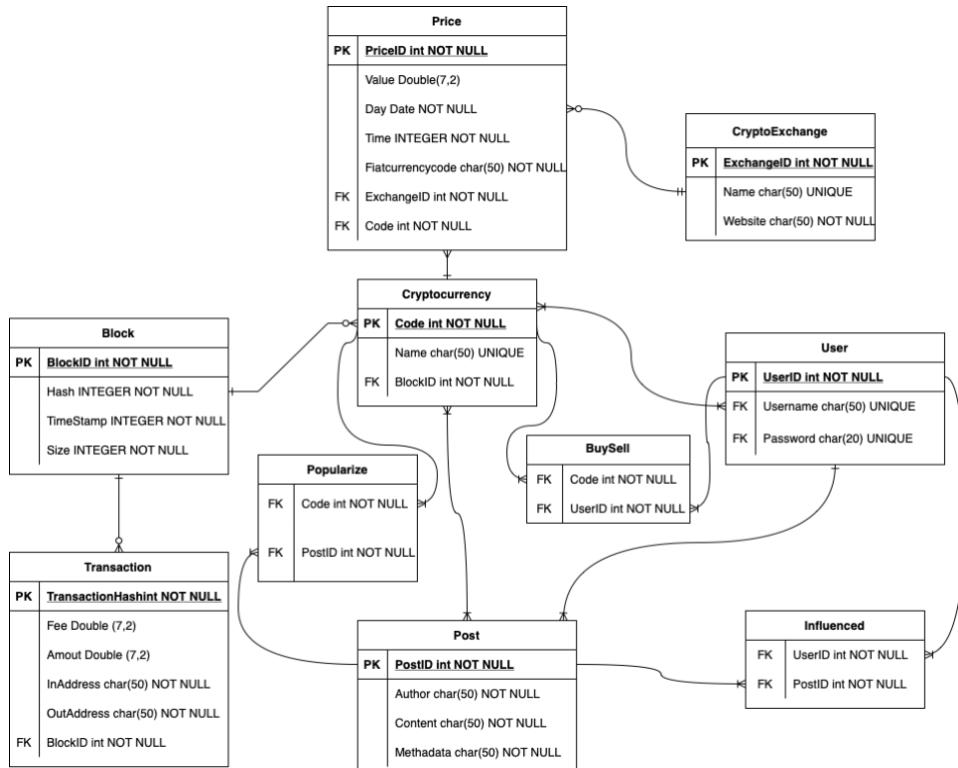
When it comes to determine the major role of blockchain for cryptocurrency existence, different transactions are made that have unique transaction hash (unique string of characters associated to every verified transaction) and added to the blockchain at a timestamp creating a decentralized ledger which is a fundamental part of the cryptocurrency. In the database, the relationship between the blockchain with transactions and digital currency needs to be created to determine this important role.

Finally, the influence of networking and social media on the growth of digital currency. Nowadays, social media plays a massive role in influencing online users. Once an author publishes content related to cryptocurrency, it makes it more famous and get people to invest their money on it. In the database, relationships between the cryptocurrency with social media and users need to be created to show how much networking influences people to buy crypto currencies.

#### 4.ERD of my idea:



#### 4.Database relationship diagram:



## 5. Screenshots of sample data:

- Block Table:

The screenshot shows the phpMyAdmin interface for the 'Block' table in the 'Database coursework 2' database. The table has four columns: BlockID, Hash, TimeStamp, and Size. There are 7 rows of data.

	BlockID	Hash	TimeStamp	Size
<input type="checkbox"/>	0022533	0055787	0063378	0088427
<input type="checkbox"/>	0053857	0066545	0044242	0089872
<input type="checkbox"/>	0065576	0099622	0044587	0057553
<input type="checkbox"/>	0068889	0028232	0088269	0073427
<input type="checkbox"/>	0077588	0098667	0042624	0053555
<input type="checkbox"/>	0099637	0093355	0034959	0082338
<input type="checkbox"/>	1506287	8572439	2081364	5210796

- Block table structure:

The screenshot shows the phpMyAdmin interface for the 'Block' table structure. The table has four columns: BlockID, Hash, TimeStamp, and Size. A primary key constraint is defined on the BlockID column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	BlockID	int(7)		UNSIGNED ZEROFILL	No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More
2	Hash	int(7)		UNSIGNED ZEROFILL	No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More
3	TimeStamp	int(7)		UNSIGNED ZEROFILL	No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More
4	Size	int(7)		UNSIGNED ZEROFILL	No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="button"/> Edit	PRIMARY	BTREE	Yes	No	BlockID	7	A	No	

Partitions:

No partitioning defined!

Partition table:

- BuySell table:

The screenshot shows the phpMyAdmin interface for the 'BuySell' table in the 'Database coursework 2' database. The table has two columns: 'Code' and 'UserID'. The data is as follows:

Code	UserID
0029874	0032258
0048865	0058437
0052530	0059839
0078935	0053239
0292746	0059869
0936444	0078474
3448997	0058433

- BuySell table structure:

The screenshot shows the phpMyAdmin interface for the 'BuySell' table structure. The table has two columns: 'Code' and 'UserID'. The structure is defined as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Code	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	UserID	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

## Cryptocurrency table:

The screenshot shows the phpMyAdmin interface for the 'Cryptocurrency' table in the 'Database coursework\_2' database. The table has 7 rows and 4 columns: Code, Name, and BlockID. The data includes entries for Monero, Stellar, Dogecoin, Cardano, Polkadot, Litecoin, and Bitcoin.

	Code	Name	BlockID
<a href="#">Edit</a>	0029874	Monero	0022533
<a href="#">Edit</a>	0048864	Stellar	0063857
<a href="#">Edit</a>	0052538	Dogecoin	0066576
<a href="#">Edit</a>	0078935	Cardano	0068889
<a href="#">Edit</a>	0292746	Polkadot	0077588
<a href="#">Edit</a>	0936444	Litecoin	0096367
<a href="#">Edit</a>	3448997	Bitcoin	1506287

- Cryptocurrency structure:

The screenshot shows the 'Table structure' tab for the 'Cryptocurrency' table. It displays the columns: Code (int(7)), Name (varchar(50)), and BlockID (int(7)). The 'Indexes' section shows three indexes: Name, Name\_3, and Name\_2. The 'Partitions' section is empty.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Code	int(7)	UNICODE_CI_AS	UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	Name	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	BlockID	int(7)	UNICODE_CI_AS	UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a>	1	PRIMARY	BTREE	Yes	Code	7	A	No	
<a href="#">Edit</a>	2	BTREE	Yes	No	Name	7	A	No	
<a href="#">Edit</a>	3	BTREE	Yes	No	Name_3	7	A	No	
<a href="#">Edit</a>	4	BTREE	No	No	Name_2	7	A	No	
<a href="#">Edit</a>	5	BTREE	No	No	BlockID	7	A	No	

## CryptoExchange table:

The screenshot shows the phpMyAdmin interface for the 'CryptoExchange' table. The table structure is as follows:

	ExchangeID	Name	Website
<input type="checkbox"/>	0003799	Bitcoin	<a href="https://bitcoinira.com">https://bitcoinira.com</a>
<input type="checkbox"/>	0006885	Dogecoin	<a href="https://blockfi.com">https://blockfi.com</a>
<input type="checkbox"/>	0292746	Cardano	<a href="https://www.kraken.com">https://www.kraken.com</a>
<input type="checkbox"/>	0524228	Monero	<a href="https://crypto.com/vea/">https://crypto.com/vea/</a>
<input type="checkbox"/>	0579329	Stellar	<a href="https://www.coinbase.com/en">https://www.coinbase.com/en</a>
<input type="checkbox"/>	0936444	Litecoin	<a href="https://www.binance.com/en">https://www.binance.com/en</a>
<input type="checkbox"/>	0977294	Polkadot	<a href="https://www.etoro.com">https://www.etoro.com</a>

- CryptoExchange structure:

The screenshot shows the phpMyAdmin interface for the 'CryptoExchange' table structure. The columns are defined as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ExchangeID	int(7)	utf8mb4_general_ci	UNSIGNED ZEROFILL	No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More
2	Name	varchar(50)	utf8mb4_general_ci		No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More
3	Website	varchar(50)	utf8mb4_general_ci		No	None			<input type="button"/> Change <input type="button"/> Drop <input type="button"/> More

Indexes section:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="button"/> Edit <input type="button"/> Rename <input type="button"/> Drop		PRIMARY	BTREE	Yes	ExchangeID	7	A	No	
<input type="button"/> Edit <input type="button"/> Rename <input type="button"/> Drop		Name	BTREE	Yes	Name	7	A	No	

- Influenced table:

The screenshot shows the phpMyAdmin interface for the 'Influenced' table in the 'Database coursework 2' database. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	UserID	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	PostID	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the table structure, there is a section for 'Indexes' which lists two indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a> <a href="#">Rename</a> <a href="#">Drop</a>	UserID	BTREE	No	No	UserID	7	A	No	
<a href="#">Edit</a> <a href="#">Rename</a> <a href="#">Drop</a>	PostID	BTREE	No	No	PostID	7	A	No	

At the bottom of the page, there is a console window with the following SQL query:

```
>SELECT * FROM `CryptoExchange`;
```

- Influenced structure table:

The screenshot shows the phpMyAdmin interface for the 'Influenced' table in the 'Database coursework 2' database. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	UserID	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	PostID	int(7)		UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the table structure, there is a section for 'Indexes' which lists two indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a> <a href="#">Rename</a> <a href="#">Drop</a>	UserID	BTREE	No	No	UserID	7	A	No	
<a href="#">Edit</a> <a href="#">Rename</a> <a href="#">Drop</a>	PostID	BTREE	No	No	PostID	7	A	No	

At the bottom of the page, there is a console window with the following SQL query:

```
>SELECT * FROM `CryptoExchange`;
```

- Popularize Table:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User, information\_schema, mysql, performance\_schema, phpmyadmin, test, and week1. The 'Popularize' table is selected. The main area displays the table's data with columns 'PostID' and 'Code'. The data rows are:

PostID	Code
0396954	0052538
0554930	0078935
0577838	0252746
0588438	0936444
0727237	3448997
0967857	0029874
8366986	0048864

Below the table, there are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. A 'Bookmark this SQL query' section allows labeling the query and setting access permissions.

- Popularize structure table:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User, information\_schema, mysql, performance\_schema, phpmyadmin, test, and week1. The 'Popularize' table structure is selected. The main area shows the table structure with two columns: 'PostID' and 'Code'. The 'Indexes' section contains one index named 'PostID' with a type of BTREE and cardinality of 7. The 'Information' section shows the table's creation query:

```
>SELECT * FROM 'CryptoExchange'
```

- Post table:

The screenshot shows the phpMyAdmin interface for the 'Post' table in the 'Crypto' database. The table has 7 rows of data:

	PostID	Author	Content	Metadata
<input type="checkbox"/>	0396954	Nouha Lamrani	Post Facebook	Posts effectiveness
<input type="checkbox"/>	0554939	Kas Khok	Post Twitter	Analysing social media performance
<input type="checkbox"/>	0577838	Ion Pruteanu	Newspapers	Creation date and time
<input type="checkbox"/>	0588438	Krystian Mowinski	Post Telegram	Social media engagement
<input type="checkbox"/>	0727237	Abbos Abdullaev	Advertisment	Digital management
<input type="checkbox"/>	0967857	Bayram Tosun	Magazines	Competitor data
<input type="checkbox"/>	8366986	Jad Korayani	Instagram Post	custom taxonomies

- Post structure table:

The screenshot shows the phpMyAdmin interface for the 'Post' table in the 'Database coursework 2' database. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PostID	int(7)	UNSERIALIZED	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
2	Author	varchar(50)	utf8mb4_general_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
3	Content	varchar(50)	utf8mb4_general_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More
4	Metadata	varchar(50)	utf8mb4_general_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment	
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	PostID	7	A	No

Partition table:

No partitioning defined!

Console:

```
>SELECT * FROM `CryptoExchange`;
```

- Price table:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists various tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User. The 'Price' table is selected. The main area displays the 'Price' table data with the following columns: PriceID, Value, Day, Time, Fiatcurrencycode, Code, ExchangeID. The data shows 7 rows of currency exchange rates. Below the table, there are buttons for 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'. A 'Query results operations' section includes 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. A 'Bookmark this SQL query' section allows labeling the query. The bottom console shows the command: >SELECT \* FROM `CryptoExchange`.

- Price structure table:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists various tables. The 'Price' table is selected. The main area shows the 'Table structure' tab, displaying the following columns: PriceID, Value, Day, Time, Fiatcurrencycode, Code, ExchangeID. The 'Attributes' column shows details like unsigned zero fill, no null, and specific collations. The 'Indexes' tab shows two indexes: 'PriceID' (PRIMARY, BTREE) and 'Code' (BTREE). Below the table structure, there are buttons for 'Add', 'Propose table structure', 'Track table', 'Move columns', 'Normalize', and 'Create an index on'. The bottom console shows the command: >SELECT \* FROM `CryptoExchange`.

- Transaction table:

The screenshot shows the phpMyAdmin interface for the 'Transaction' table in the 'Database coursework 2' database. The table has columns: TransactionHash, Fee, Amount, InAddress, OutAddress, and BlockID. The data grid displays 7 rows of transaction records. Below the table, there are buttons for operations like Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' section is also present.

	TransactionHash	Fee	Amount	InAddress	OutAddress	BlockID
1	0022567	0348.56	5667.50	1YuqNMd5gMP8sZbEjBfoubVAcI	3YuqNMd5gMP8sZbEjBfoubVAcI	0022533
2	0026846	2345.21	0355.22	15NFYsxJxRVGceEZ24RkX2RmHRD	35NFYsxJxRVGceEZ24RkX2RmHRD	0066587
3	0027883	0678.50	4677.78	1FWQ9elc4QT9R4SdfFxBFFF5NM	3FWQ9elc4QT9R4SdfFxBFFF5NM	0066576
4	0058787	45346.50	0665.70	1qRNRIrwzsAEEptEKDBGJYUjhB	3qRNRIrwzsAEEptEKDBGJYUjhB	0068889
5	0074868	8475.40	0567.50	1ywUAUyHwChRe4Q4Ubc8d94jX	3ywUAUyHwChRe4Q4Ubc8d94jX	0077588
6	0087755	0123.12	1234.34	1X6TE7duJlJLsFMrooYY2sMuQ	3X6TE7duJlJLsFMrooYY2sMuQ	0099637
7	0099524	3578.43	6678.90	1B9DHfajt9BTG7BXUneBsuePb	3B9DHfajt9BTG7BXUneBsuePb	1506287

- Transaction structure table:

The screenshot shows the phpMyAdmin interface for the 'Transaction' table structure. It lists columns: TransactionHash, Fee, Amount, InAddress, OutAddress, and BlockID. The 'Indexes' section shows a primary key 'PRIMARY' on the 'TransactionHash' column. Below the table, there are buttons for operations like Print, Propose table structure, Track table, Move columns, Normalize, Add, and Create index.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	TransactionHash	int(7)		UNSIGNED ZEROFILL	No	None			Change  Drop  More
2	Fee	double(7,2)		UNSIGNED ZEROFILL	No	None			Change  Drop  More
3	Amount	double(7,2)		UNSIGNED ZEROFILL	No	None			Change  Drop  More
4	InAddress	varchar(35)	utf8mb4_general_ci		No	None			Change  Drop  More
5	OutAddress	varchar(35)	utf8mb4_general_ci		No	None			Change  Drop  More
6	BlockID	int(7)		UNSIGNED ZEROFILL	No	None			Change  Drop  More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit	Rename	Drop	PRIMARY	BTREE	Yes	No	TransactionHash	7	A No
Edit	Rename	Drop	BlockID	BTREE	No	No	BlockID	7	A No

- User table:

The screenshot shows the phpMyAdmin interface for the 'User' table in the 'Database coursework 2' database. The table structure is as follows:

	User ID	Username	Password
<a href="#">Edit</a>	0032258	Nouha_45	ZMwLprtdAH
<a href="#">Edit</a>	0038437	Bayram_321	rTuMEukjCU
<a href="#">Edit</a>	0039839	kas_78	nunGeEDc5u
<a href="#">Edit</a>	0053239	Krystian98	2r6YXDSMFu
<a href="#">Edit</a>	0059869	Jad123	Y8gJDwXnR
<a href="#">Edit</a>	0078474	abbos167	Fps8v3mica
<a href="#">Edit</a>	0098433	Ion569	1Yzh8CcjSd

Below the table, there are buttons for 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'. The 'Console' section at the bottom contains the query: `>SELECT * FROM `User``.

- User structure table:

The screenshot shows the phpMyAdmin interface for the 'User' table structure in the 'Database coursework 2' database. The structure is defined by the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	User ID	int(7)	utf8mb4_general_ci	UNSIGNED ZEROFILL	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	Username	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	Password	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the columns, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', and 'Add to central columns'. The 'Indexes' section shows three primary keys:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a>	<a href="#">Rename</a>	<a href="#">Drop</a>	PRIMARY	BTREE	Yes	No	User ID	7	A No
<a href="#">Edit</a>	<a href="#">Rename</a>	<a href="#">Drop</a>	Username	BTREE	Yes	No	Username	7	A No
<a href="#">Edit</a>	<a href="#">Rename</a>	<a href="#">Drop</a>	Password	BTREE	Yes	No	Password	7	A No

Below the indexes, there is a button for 'Create an index on' and a note: 'No partitioning defined!'. The 'Console' section at the bottom contains the query: `>SELECT * FROM `User``.

- Foreign key for N:1 relation:

- CryptoExchange to price:

```

1 -- CryptoExchange to Price
2 ALTER TABLE Price
3 ADD CONSTRAINT ExchangeID
4 FOREIGN KEY (ExchangeID)
5 REFERENCES CryptoExchange (ExchangeID);

```

The screenshot shows the phpMyAdmin interface with the database 'Database coursework 2' selected. In the SQL tab, the above SQL code is entered. Below the code, a green message indicates that MySQL returned an empty result set. At the bottom, the status bar shows the query was executed successfully.

- Cryptocurrency to price:

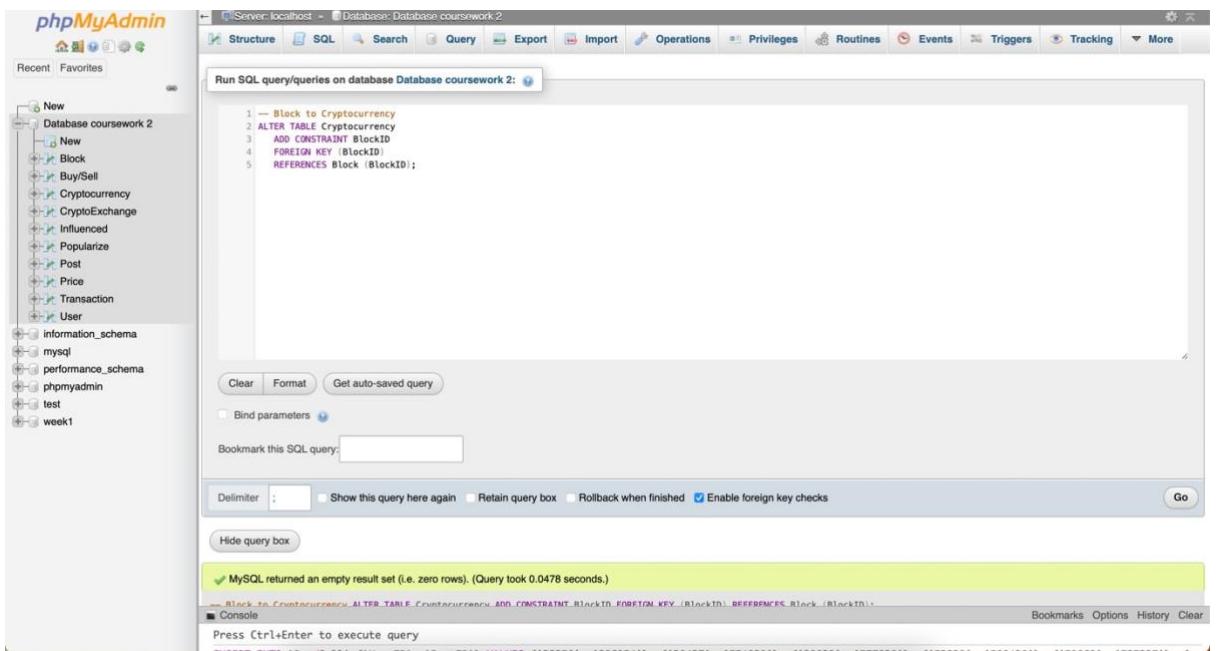
```

1 -- Cryptocurrency to Price
2 ALTER TABLE Price
3 ADD CONSTRAINT Code
4 FOREIGN KEY (Code)
5 REFERENCES Cryptocurrency (Code);

```

The screenshot shows the phpMyAdmin interface with the database 'Database coursework 2' selected. In the SQL tab, the above SQL code is entered. Below the code, a green message indicates that MySQL returned an empty result set. At the bottom, the status bar shows the query was executed successfully.

- Block to cryptocurrency:



The screenshot shows the phpMyAdmin interface for a database named 'Database coursework 2'. In the left sidebar, there is a tree view of tables: New, Database coursework 2 (which contains New, Block, Buy/Sell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User), information\_schema, mysql, performance\_schema, phpmypadmin, test, and week1. The 'Cryptocurrency' table is selected. In the main panel, the 'SQL' tab is active. A SQL query is entered in the text area:

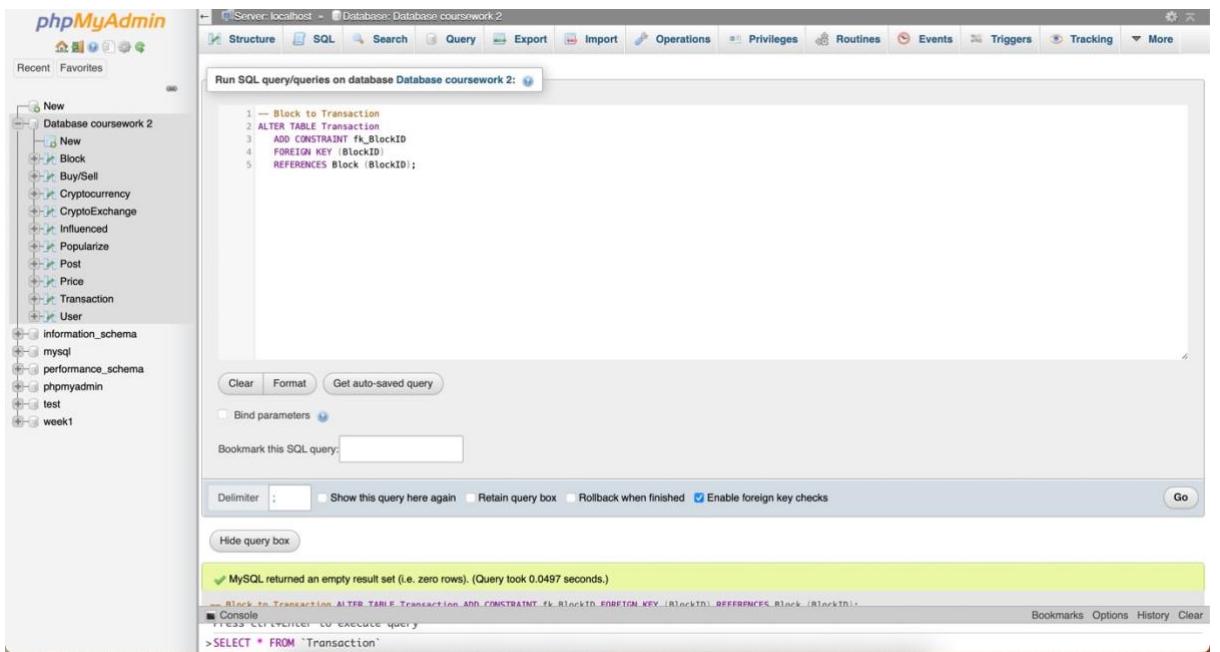
```

1 -- Block to Cryptocurrency
2 ALTER TABLE Cryptocurrency
3   ADD CONSTRAINT BlockID
4     FOREIGN KEY (BlockID)
5       REFERENCES Block (BlockID);

```

Below the query, the status bar shows: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0478 seconds.)

- Block to transaction:



The screenshot shows the phpMyAdmin interface for the same database. The 'Transaction' table is selected in the tree view. In the main panel, the 'SQL' tab is active. A SQL query is entered in the text area:

```

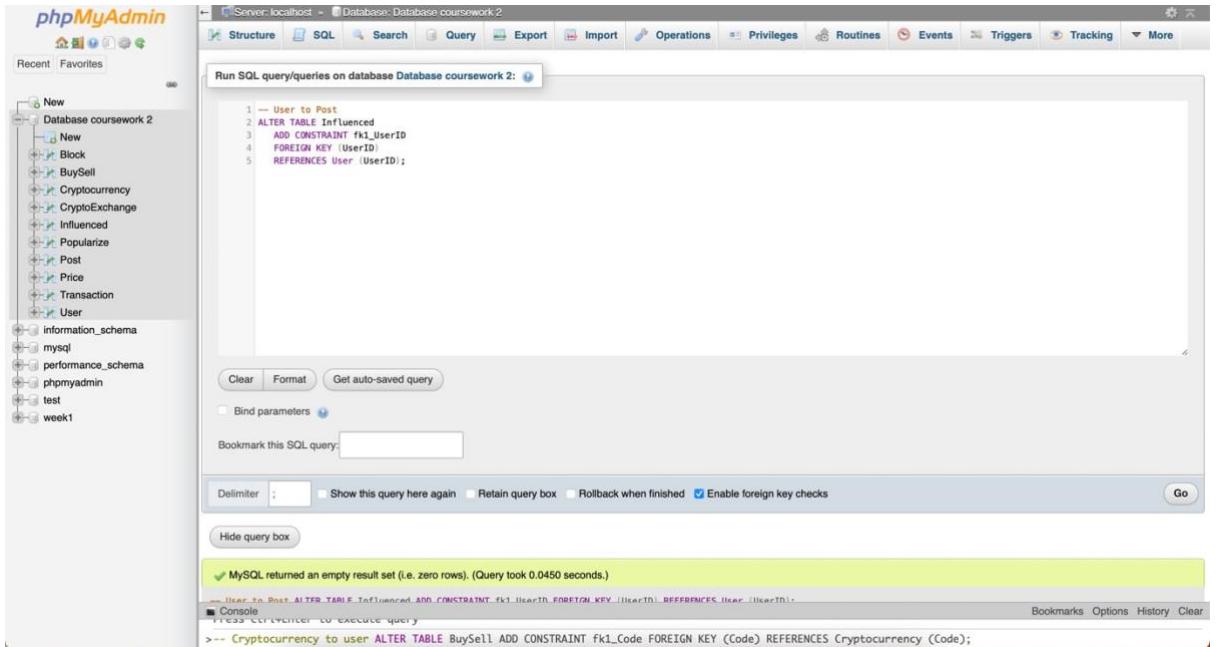
1 -- Block to Transaction
2 ALTER TABLE Transaction
3   ADD CONSTRAINT fk_BlockID
4     FOREIGN KEY (BlockID)
5       REFERENCES Block (BlockID);

```

Below the query, the status bar shows: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0497 seconds.)

- Foreign Key for M:N relations:

- User to post:



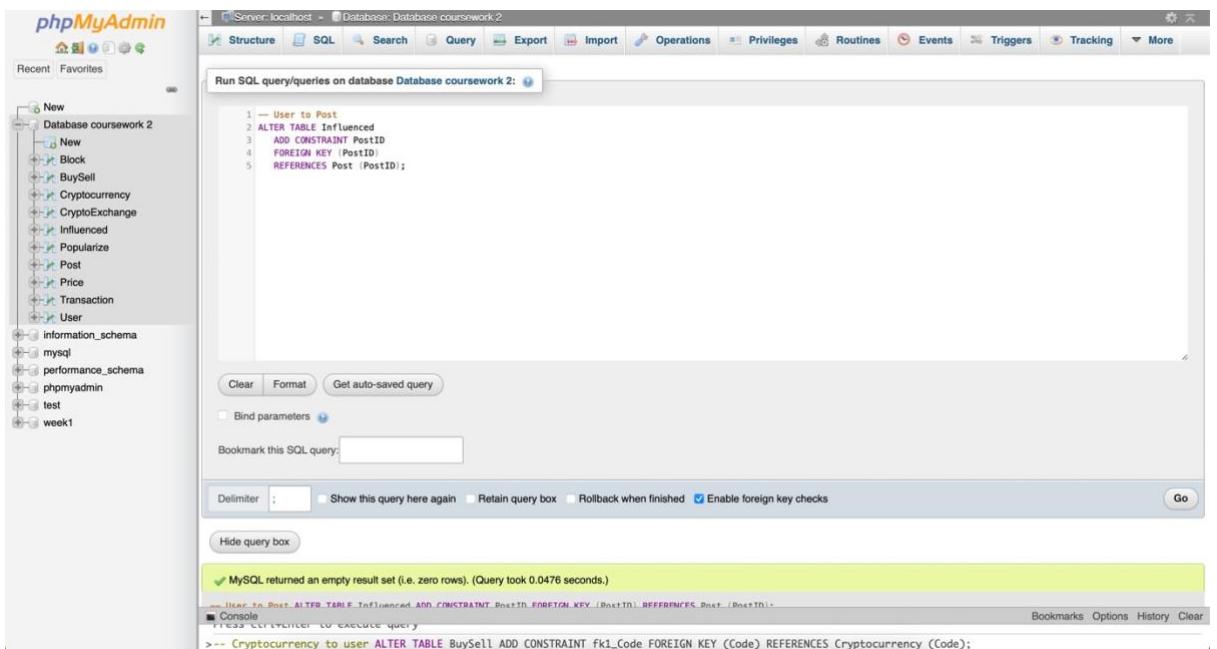
The screenshot shows the phpMyAdmin interface for a database named 'Database coursework 2'. The left sidebar lists tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, and User. The 'Influenced' table is selected. The main area contains a SQL query window with the following code:

```

1 -- User to Post
2 ALTER TABLE Influenced
3 ADD CONSTRAINT fk1_UserID
4 FOREIGN KEY (UserID)
5 REFERENCES User (UserID);

```

Below the code, there are buttons for 'Clear', 'Format', 'Get auto-saved query', 'Bind parameters', and a 'Bookmark this SQL query' input field. At the bottom, there are options for 'Delimiter', 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks'. A green status bar at the bottom indicates: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0450 seconds.)'.



This screenshot is identical to the one above, showing the same phpMyAdmin interface and SQL query for creating a foreign key constraint 'fk1\_UserID' on the 'Influenced' table. The code is identical:

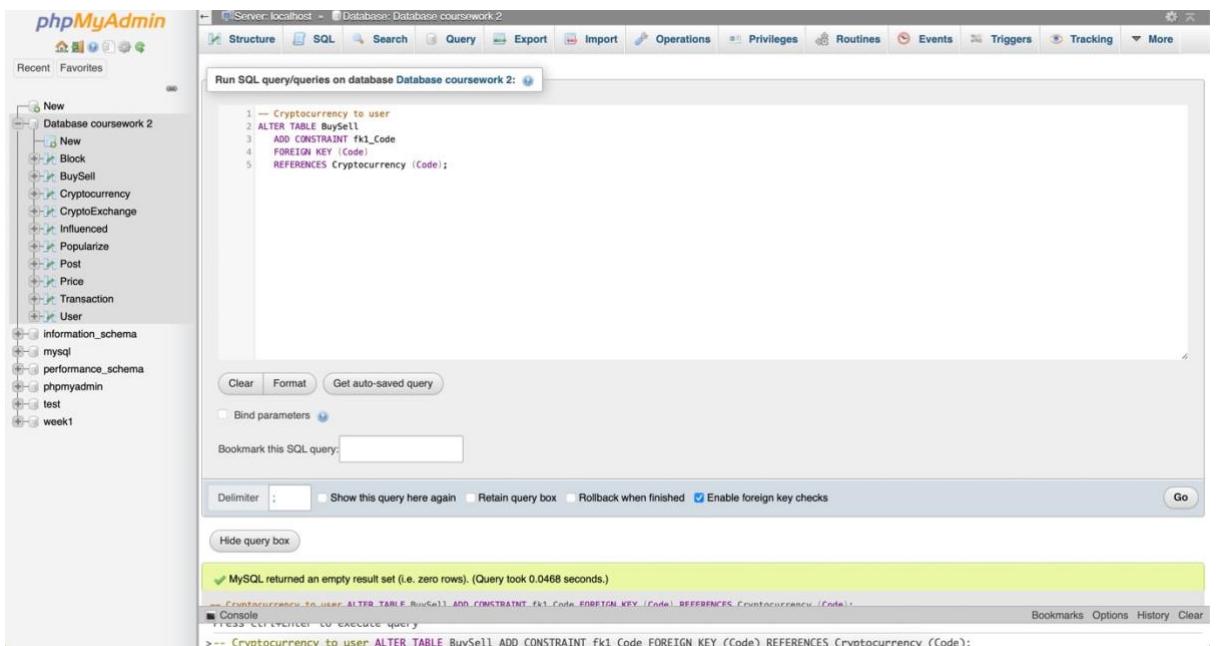
```

1 -- User to Post
2 ALTER TABLE Influenced
3 ADD CONSTRAINT fk1_UserID
4 FOREIGN KEY (UserID)
5 REFERENCES User (UserID);

```

The status bar at the bottom also indicates: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0476 seconds.)'.

- User to cryptocurrency:



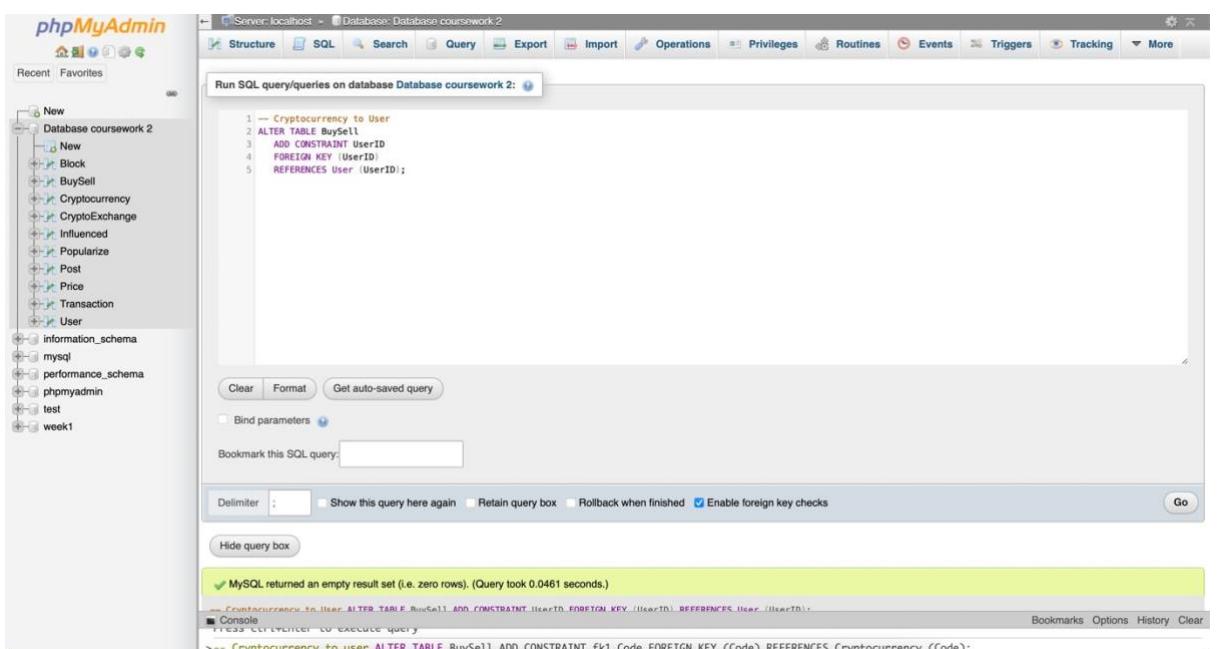
The screenshot shows the phpMyAdmin interface for a database named 'coursework\_2'. The left sidebar lists tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, and User. The 'BuySell' table is selected. The main area contains the following SQL code:

```

1 -- Cryptocurrency to user
2 ALTER TABLE BuySell
3 ADD CONSTRAINT fk1_Code
4 FOREIGN KEY (Code)
5 REFERENCES Cryptocurrency (Code);

```

Below the code, the status bar indicates: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0468 seconds.)



The screenshot shows the phpMyAdmin interface for the same database 'coursework\_2'. The 'BuySell' table is selected. The main area contains the following SQL code:

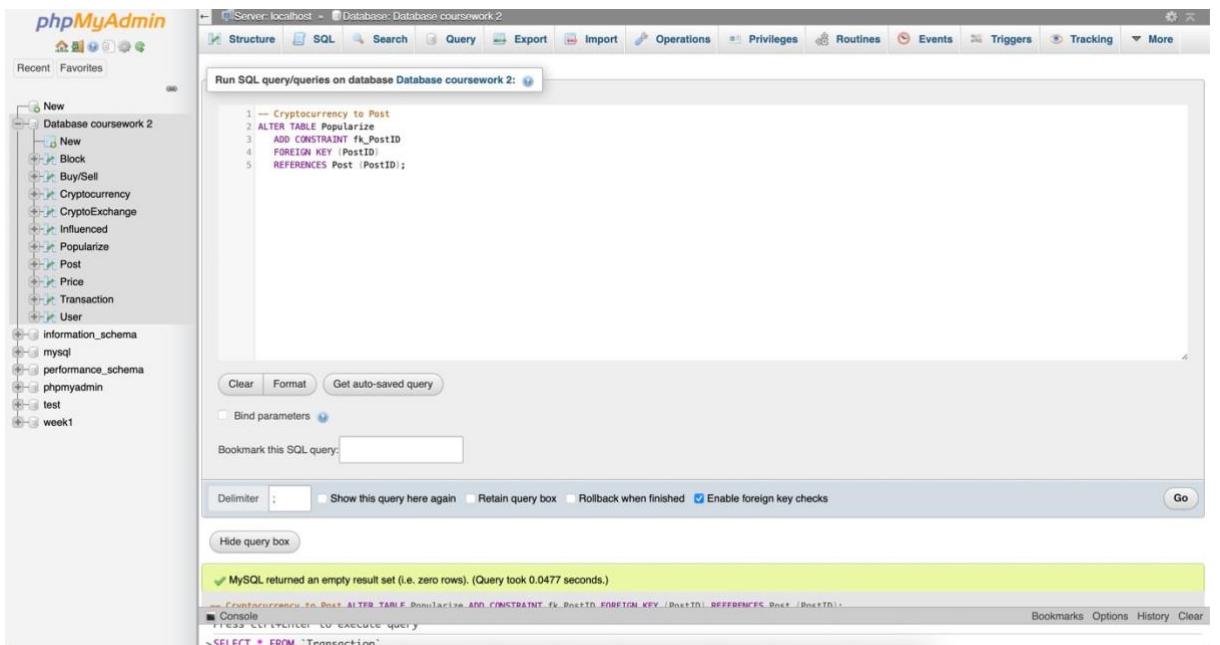
```

1 -- Cryptocurrency to User
2 ALTER TABLE BuySell
3 ADD CONSTRAINT UserID
4 FOREIGN KEY (UserID)
5 REFERENCES User (UserID);

```

Below the code, the status bar indicates: MySQL returned an empty result set (i.e. zero rows). (Query took 0.0461 seconds.)

- Cryptocurrency to post:



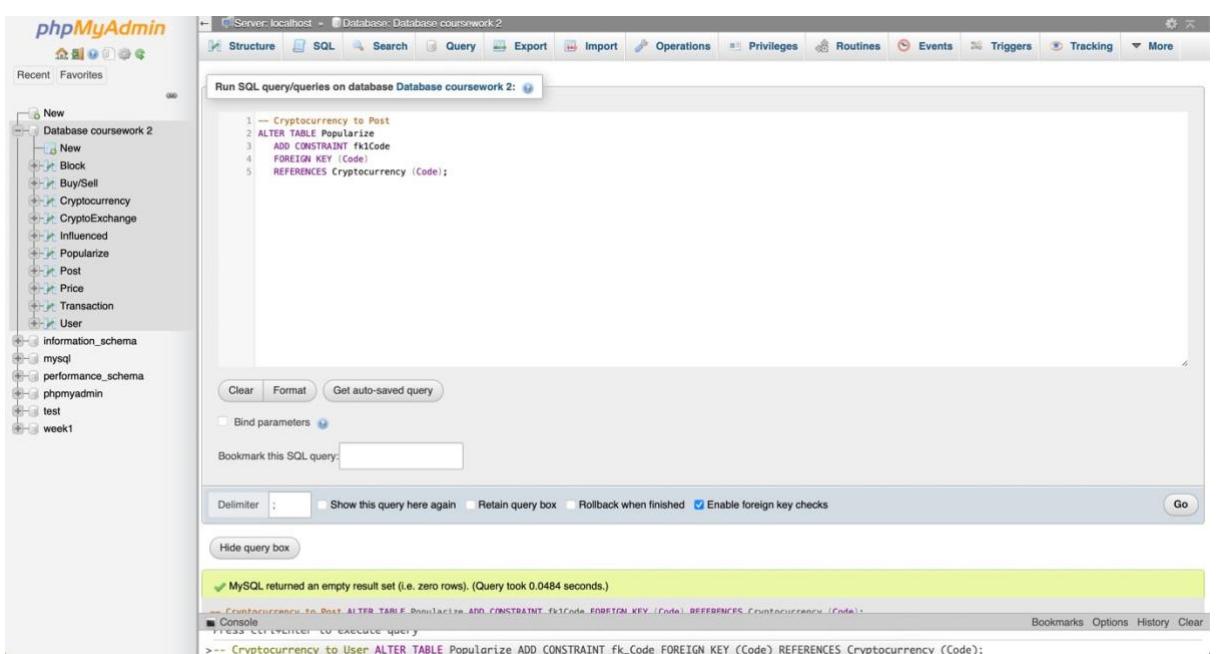
The screenshot shows the phpMyAdmin interface for the database 'Database coursework 2'. In the left sidebar, under the 'New' category, there is a table named 'Popularize'. In the main SQL query window, the following SQL code is entered:

```

1 -- Cryptocurrency to Post
2 ALTER TABLE Popularize
3   ADD CONSTRAINT fk_PostID
4     FOREIGN KEY (PostID)
5       REFERENCES Post (PostID);

```

Below the code, the status message indicates: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0477 seconds.)". The command history at the bottom shows: "Cryptocurrency to Post ALTER TABLE Popularize ADD CONSTRAINT fk\_PostID FOREIGN KEY (PostID) REFERENCES Post (PostID);".



The screenshot shows the phpMyAdmin interface for the database 'Database coursework 2'. In the left sidebar, under the 'New' category, there is a table named 'Popularize'. In the main SQL query window, the following SQL code is entered:

```

1 -- Cryptocurrency to Post
2 ALTER TABLE Popularize
3   ADD CONSTRAINT fkCode
4     FOREIGN KEY (Code)
5       REFERENCES Cryptocurrency (Code);

```

Below the code, the status message indicates: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0484 seconds.)". The command history at the bottom shows: "Cryptocurrency to Post ALTER TABLE Popularize ADD CONSTRAINT fkCode FOREIGN KEY (Code) REFERENCES Cryptocurrency (Code);".

## 6.Demonstration of the scenario:

- 1) Find PriceID and Code from Price table where the Value is higher than 100 with being in a descending order.
  - 2) Count each Block ID, Find Hash from Block table and group them by Hash.
  - 3) Find the average and sum value from the Price table.
  - 4) Find the PriceID from Price table that was displayed between '2022-04-10' and '202204-25'.
  - 5) Combine both columns: Username from User and Author from Post.
  - 6) Create a new table called NewTable that contains Id,Name, Phone and add it to your database.
  - 7) Find the maximum value in Price table by adding a column called 'Max\_value' to Price table.
  - 8) Find the TransactionHash which Hash is lower than 90000.00.
  - 9) Find PriceID, Value, Day,Time, Fiatcurrencycode from price Table where the day is not 2022-04-26 and the PriceID is higher than 0008922, or the value is lower than 565.67.
  - 10) Make the InAddress equal to the OutAddress in the Transaction Table.
  - 11) Find Fiat currency code and Exchange ID who is available in <https://www.binance.com/en>.
  - 12) Find the cryptoExchange name which value is higher than 1000.00.
  - 13) Create a view 'Price\_1' that contains all the Values > 1500, then drop it.
  - 14) Give an example of a nested subquery using Price, Transaction and User tables.
  - 15) Find all Transaction's information whose blockID is not 0022533 and earns less than it.
- (Note: all underlined questions are for complex queries)

o Solution and outputs:

1) Query 1:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists various tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User, and several system schemas. The 'Price' table is selected. The main area displays the results of a query: "Find PriceID and Code from Price table where the Value is higher than 100 with being in a descending order". The results show 7 rows with columns 'PriceID' and 'Code'. The data is as follows:

PriceID	Code
0938486	0029874
0595834	0078935
0489386	0936444
0682453	0292746
0008922	0052538
0822433	3448997
0274347	0048864

2) Query 2:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar lists various tables: New, Block, BuySell, Cryptocurrency, CryptoExchange, Influenced, Popularize, Post, Price, Transaction, User, and several system schemas. The 'Block' table is selected. The main area displays the results of a query: "Count each Block ID, Find Hash from Block table and group them by Hash". The results show 7 rows with columns 'COUNT(BlockID)' and 'BlockID, Hash'. The data is as follows:

COUNT(BlockID)	BlockID	Hash
1	0068888	0028232
1	0025353	0055787
1	0063857	0066545
1	0099637	0093355
1	0077588	0098667
1	0066576	0099622
1	1506287	8572439

### 3) Query 3:

The screenshot shows the phpMyAdmin interface for a database named 'Database coursework 2'. The left sidebar lists tables: New, Database coursework 2, and various other tables like Block, BuySell, Cryptocurrency, etc. The main area is titled 'Table: Price' with tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers. A message at the top says 'Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)'. Below it is the SQL query: `-- Find the average and sum value from the Price table SELECT AVG(Value),SUM(Value) FROM Price;`. The results show one row: **AVG(Value)** 2593.085714 and **SUM(Value)** 18151.60. There are buttons for Show all, Number of rows (set to 25), Filter rows, and a search bar. Below the results is a 'Query results operations' section with Print, Copy to clipboard, Export, Display chart, and Create view options. At the bottom is a 'Bookmark this SQL query' section with a label input and a checkbox for 'Let every user access this bookmark'. The console at the bottom shows the command `>SELECT * FROM 'Price'`.

### 4) Query 4:

This screenshot is similar to the previous one but shows a different SQL query. The main message at the top says 'Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)'. The SQL query is: `-- Find the PriceID from Price table that was displayed between '2022-04-10' and '2022-04-25' SELECT PriceID FROM Price WHERE Day BETWEEN '2022-04-10' AND '2022-04-25';`. The results show four rows of PriceID values: 0008922, 0274347, 0682453, and 0838486. The interface includes 'Edit', 'Copy', and 'Delete' buttons for each row. The 'Sort by key' dropdown is set to 'None'. The rest of the interface is identical to the first screenshot, including the 'Query results operations' and 'Bookmark this SQL query' sections.

### 5) Query 5:

phpMyAdmin

Server: localhost - Database: Database coursework 2 - Table: Price

Recent Favorites

New Database coursework 2

- New
- Block
- BuySell
- Cryptocurrency
- CryptoExchange
- Influenced
- Popularize
- Post
- Price
- Transaction
- User

information\_schema

mysql

performance\_schema

phpmyadmin

test

week1

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

Showing rows 0 - 13 (14 total, Query took 0.0010 seconds.)

-- Combine both columns: Username from User and Author from Post. `SELECT Username FROM User UNION SELECT Author FROM Post;`

[Edit inline] [Edit] [Create PHP code]

Show all Number of rows: 25 Filter rows: Search this table

+ Options

Username

abbos167
Bayram_321
Ion569
Jad123
kas_78
Krystian98
Nouha_45
Nouha Lamrani
Kas Khok
Ion Pruteanu
Krystian Mowinski
Abbos Abdullaev
Bayram Tosun
Jad Korayani

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Console

SQL Editor > EXECUTE query

>`SELECT * FROM 'Price'`

Bookmarks Options History Clear

The screenshot shows the phpMyAdmin interface for a database named 'Database coursework 2'. The left sidebar shows various databases and tables. The main area displays the 'Price' table with 14 rows. The table contains a single column 'Username' which is a combination of 'User.Username' and 'Post.Author'. A warning message at the top indicates that grid edit, checkbox, and other features are not available because there is no unique column selected. The SQL query used is: 'SELECT Username FROM User UNION SELECT Author FROM Post;'. The bottom of the screen shows the query editor with the same query.

6) Query 6:

The screenshot shows the phpMyAdmin interface for a database named 'Database coursework.2'. In the left sidebar, under the 'Database coursework.2' section, there is a new table named 'NewTable' with columns 'ID', 'Name', and 'Phone'. The main panel displays a success message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0232 seconds.)' Below this, it says '— Create a new table called NewTable that contains Id,Name,Phone and add it to your database: CREATE TABLE NewTable (Id varchar(50), Name varchar(50), Phone varchar(20));' and provides options to 'Edit inline' or 'Create PHP code'. At the bottom of the main panel, the SQL query 'SELECT \* FROM `Price`' is shown.

This screenshot shows the phpMyAdmin interface for the same database. The table 'NewTable' is selected in the sidebar. The main panel displays the results of the query 'SELECT \* FROM `NewTable`', which is empty. It shows a single row with three columns: 'ID', 'Name', and 'Phone'. Below the results, there are sections for 'Query results operations' (with 'Create view') and 'Bookmark this SQL query' (with a 'Label:' input field and a checkbox for 'Let every user access this bookmark'). The SQL query 'SELECT \* FROM `NewTable`' is also visible at the bottom of the main panel.

## 7) Query 7:

Showing rows 0 - 6 (7 total, Query took 0.0010 seconds.)

```
-- Find the maximum value in Price table by adding a row called 'Max_value' to Price table: SELECT PriceID, Value, Day, Time, Fiatcurrencycode, code, ExchangeID, MAX(Value) OVER () AS max_value FROM Price;
```

PriceID	Value	Day	Time	Fiatcurrencycode	code	ExchangeID	max_value
0274347	0456.45	2022-04-24	2022-04-08 17:25:36	BZR	0048864	0524228	8675.56
0822433	0554.59	2022-04-26	2022-04-08 17:25:36	ZAR	3448997	0579329	8675.56
0008922	0565.67	2022-04-14	2022-04-08 15:57:02	CNY	0052538	0006885	8675.56
0682453	0987.45	2022-04-11	2022-04-08 17:25:36	GBP	0292746	0292746	8675.56
0489384	2345.43	2022-04-29	2022-04-08 17:25:36	HKD	0936444	0936444	8675.56
0595834	4566.45	2022-04-28	2022-04-08 17:25:36	EUR	0078935	0977294	8675.56
0838484	8675.56	2022-04-13	2022-04-08 17:25:36	AUD	0029874	0003799	8675.56

## 8) Query 8:

Showing rows 0 - 2 (3 total, Query took 0.0013 seconds.)

```
-- Find the TransactionHash which Hash is lower than 90000.00. SELECT ce.TransactionHash FROM Transaction ce JOIN Block p ON ce.BlockID= p.BlockID WHERE hash < 90000;
```

TransactionHash
0022567
0026846
0058787

## 9) Query 9:

phpMyAdmin

Server: localhost - Database: Database coursework 2 - Table: Price

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

Showing rows 0 - 4 (total, Query took 0.0012 seconds.)

```
-- Find PriceID, Value, Day, Time, Fiatcurrencycode from price Table where the day is not 2022-04-26 and the PriceID is higher than 0008922, or the value is lower than 0565.67
SELECT PriceID, Value, Day, Time, Fiatcurrencycode FROM Price WHERE NOT ((day = '2022-04-26' AND PriceID > 0008922) OR Value < 0565.67);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	PriceID	Value	Day	Time	Fiatcurrencycode
<input type="checkbox"/>	Edit Copy Delete	0008922	0565.67	2022-04-14	2022-04-08 15:57:02 CNY
<input type="checkbox"/>	Edit Copy Delete	0489388	2345.43	2022-04-29	2022-04-08 17:25:36 HKD
<input type="checkbox"/>	Edit Copy Delete	0595834	4566.45	2022-04-28	2022-04-08 17:25:36 EUR
<input type="checkbox"/>	Edit Copy Delete	0682453	0987.45	2022-04-11	2022-04-08 17:25:36 GBP
<input type="checkbox"/>	Edit Copy Delete	0838484	8675.56	2022-04-13	2022-04-08 17:25:36 AUD

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

Console

```
>Describe Price_1;
```

Bookmarks Options History Clear

## 10) Query 10:

The screenshot shows the phpMyAdmin interface for the 'Crypto' database. The left sidebar shows the schema structure with 'New' and 'Crypto' expanded, containing 'Block', 'BuySell', 'Cryptocurrency', 'CryptoExchange', 'Influenced', 'NewTable', 'Popularize', 'Post', 'Price', 'Transaction', and 'User'. The 'Transaction' table is selected in the main area. The table has columns: TransactionHash, Fee, Amount, InAddress, OutAddress, and BlockID. There are 6 rows of data:

TransactionHash	Fee	Amount	InAddress	OutAddress	BlockID
0022567	0346.56	5667.50	3YuqNMd5gMP8sZbEjBloubVAc	3YuqNMd5gMP8sZbEjBloubVAc	0022533
0026846	2345.21	0355.22	35NFYSXJRVGcxZ24Rlx2RmlRD	35NFYSXJRVGcxZ24Rlx2RmlRD	0063857
0027883	0678.50	4677.78	3FWQ9etc4QT9R4SdfFxBF5NM	3FWQ9etc4QT9R4SdfFxBF5NM	0066576
0058787	45346.50	0665.70	3qQRNrlmwsAEpIEKDBGJYUhB	3qQRNrlmwsAEpIEKDBGJYUhB	0068889
0074868	8475.40	0567.50	3ywUBAUjHwChRe4Q4Ubc8d94jX	3ywUBAUjHwChRe4Q4Ubc8d94jX	0077588
0087755	0123.12	1234.34	3X6TE7duJVLsFmrooYY2sMuQ	3X6TE7duJVLsFmrooYY2sMuQ	0099637
0099524	3578.43	6678.90	3B9DHFalj9BTG7BXUneBsuePb	3B9DHFalj9BTG7BXUneBsuePb	1506287

The screenshot shows the phpMyAdmin interface for the 'Crypto' database. The left sidebar shows the schema structure with 'New' and 'Crypto' expanded. The 'Query' tab is selected in the top menu. The query results show '0 rows affected. (Query took 0.0014 seconds.)' and the SQL command: `-- Make the InAddress equal to the OutAddress in the Transaction Table UPDATE Transaction SET InAddress=OutAddress, OutAddress=InAddress;`. The message at the bottom states: `>-- Assuming there was a mistake when applying the data in the transaction table, where we mixed the InAddress with OutAddress, swap the Values of the two Colu..`

## 11) Query 11:

phpMyAdmin

Server: localhost - Database: Database coursework 2 - Table: Price

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0028 seconds.)

-- Find Fiat currency code and Exchange ID who is available in https://www.binance.com/en Select c.Fiatcurrencycode, c.ExchangeID From Price c JOIN CryptoExchange acr ON c.ExchangeID=acr.ExchangeID WHERE Website ='https://www.binance.com/en';

[Edit inline] [Edit] [Create PHP code]

Show all Number of rows: 25 Filter rows: Search this table

+ Options

Fiatcurrencycode	ExchangeID
HKD	0936444

Show all Number of rows: 25 Filter rows: Search this table

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

Console

PRESS ENTER TO EXECUTE QUERY

>SELECT \* FROM 'Price'

Bookmarks Options History Clear

## 12) Query 12:

phpMyAdmin

Server: localhost - Database: Crypto - Table: cryptoExchange

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total, Query took 0.0028 seconds.)

-- Find the cryptoechange name which value is higher than 1000.00. SELECT ce.name FROM cryptoExchange ce join Price p ON ce.ExchangeID = p.ExchangeID WHERE value > 1000;

[Edit inline] [Edit] [Create PHP code]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

name
Litecoin
Polkadot
Bitcoin

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

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

Console

### 13) Query 13:

The screenshot shows two instances of the phpMyAdmin interface. In the top instance, a SQL query is run to create a view named 'Price\_1' which selects all rows where the 'Value' column is greater than 1500 from the 'Price' table. The message indicates the query was successful and took 0.0238 seconds.

```
-- MySQL returned an empty result set (i.e. zero rows). (Query took 0.0238 seconds.)
-- create a view 'Price_1' that contains all the Values > 1500 , CREATE OR REPLACE VIEW Price_1 AS SELECT * FROM Price WHERE Value > 1500;
[Edit inline] [Edit] [Create PHP code]
```

The bottom instance shows the results of the 'Price\_1' view. It displays three rows of data with columns: PriceID, Value, Day, Time, Fiatcurrencycode, Code, and Exchangecode. The data is as follows:

PriceID	Value	Day	Time	Fiatcurrencycode	Code	Exchangecode
0489386	2345.43	2022-04-29	2022-04-08 17:25:36	HKD	0936444	0936444
0595834	4566.45	2022-04-28	2022-04-08 17:25:36	EUR	0078935	0977294
0838486	8675.56	2022-04-13	2022-04-08 17:25:36	AUD	0029874	0003799

The screenshot shows the phpMyAdmin interface for a database named 'Database coursework 2'. In the left sidebar, under the 'New' category, there is a newly created view named 'Price\_1'. The main panel displays a message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0012 seconds.)' followed by the SQL command: '-- Drop the view you just created Drop View price\_1;'. Below this, there are buttons for 'Edit inline', 'Edit', and 'Create PHP code'.

#### 14) Query 14:

The screenshot shows the phpMyAdmin interface for the 'Database coursework 2' database. The left sidebar shows the 'Price' table under the 'New' category. The main panel displays a message: 'Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)' followed by an example of a nested subquery. Below this, the 'Price' table is listed with one row: PriceID: 0274347, Value: 0456.45, Day: 2022-04-24, Flatcurrencycode: B2R. There are buttons for 'Edit', 'Copy', 'Delete', 'Check all', 'With selected:', and 'Export'.

#### 15) Query 15:

Showing rows 0 - 4 (total, Query took 0.0013 seconds.)

```
-- Find all Transaction's information whose blockID is not 0022533 and earns less than it. SELECT TransactionHash, BlockID, Fee FROM Transaction WHERE Amount < ANY( SELECT Amount FROM Transaction WHERE BlockID = 0022533) AND BlockID <> '0022533';
```

	TransactionHash	BlockID	Fee			
<input type="checkbox"/>	Edit	Copy	Delete	0026846	0063857	2345.21
<input type="checkbox"/>	Edit	Copy	Delete	0027883	0066576	0678.50
<input type="checkbox"/>	Edit	Copy	Delete	0058787	0068889	45346.50
<input type="checkbox"/>	Edit	Copy	Delete	0074868	0077588	8475.40
<input type="checkbox"/>	Edit	Copy	Delete	0087755	0099537	0123.12

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

Query results operations

Bookmark this SQL query

Console

## o Screenshot of the whole code:

```

1  [phpMyAdmin SQL Dump]
2   -- version 5.1.1
3   -- https://www.phpmyadmin.net/
4   --
5   -- Host: localhost
6   -- Generation Time: Apr 15, 2022 at 04:23 PM
7   -- Server version: 10.4.21-MariaDB
8   -- PHP Version: 8.1.1
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*140101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*140101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*140101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;
18 /*140101 SET NAMES utf8mb4 */;
19
20
21 -- Database: `Crypto`
22 --
23
24
25
26 -- Table structure for table `Block`
27 --
28
29
30 CREATE TABLE `Block` (
31   `BlockID` int(7) UNSIGNED ZEROFILL NOT NULL,
32   `Hash` int(7) UNSIGNED ZEROFILL NOT NULL,
33   `TimeStamp` int(7) UNSIGNED ZEROFILL NOT NULL,
34   `Size` int(7) UNSIGNED ZEROFILL NOT NULL,
35 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
36
37
38 -- Dumping data for table `Block`
39
40
41 INSERT INTO `Block` (`BlockID`, `Hash`, `TimeStamp`, `Size`) VALUES
42   (0022533, 0055878, 0063378, 0088427),
43   (0063857, 0066545, 0044242, 0089872),
44   (0066576, 0099622, 0044587, 0057553),
45   (0068889, 0028232, 0088269, 0073427),

```

```

    46  (0077588, 0098667, 0042624, 0053555),
    47  (0099637, 0093355, 0034059, 0082338),
    48  (1586287, 8572439, 2681364, 5210796);
    49
    50  -----
    51  --
    52  --
    53  -- Table structure for table 'BuySell'
    54  --
    55
    56  CREATE TABLE `BuySell` (
    57  | `Code` int(7) UNSIGNED ZEROFILL NOT NULL,
    58  | `UserID` int(7) UNSIGNED ZEROFILL NOT NULL
    59  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
    60
    61  --
    62  -- Dumping data for table 'BuySell'
    63  --
    64
    65  INSERT INTO `BuySell` (`Code`, `UserID`) VALUES
    66  (0029874, 0032580),
    67  (0048864, 0034437),
    68  (0052538, 0039839),
    69  (0078935, 0053239),
    70  (0292746, 0059869),
    71  (0936444, 0078474),
    72  (3448997, 0090433);
    73
    74  -----
    75  --
    76  --
    77  -- Table structure for table 'Cryptocurrency'
    78  --
    79
    80  CREATE TABLE `Cryptocurrency` (
    81  | `Code` int(7) UNSIGNED ZEROFILL NOT NULL,
    82  | `Name` varchar(50) NOT NULL,
    83  | `BlockID` int(7) UNSIGNED ZEROFILL NOT NULL
    84  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
    85
    86  --
    87  -- Dumping data for table 'Cryptocurrency'
    88  --
    89
    90
    91  INSERT INTO `Cryptocurrency` (`Code`, `Name`, `BlockID`) VALUES
    92  (0029874, 'Monero', 0022533),
    93  (0048864, 'Stellar', 0063857),
    94  (0052538, 'Dogecoin', 0066576),
    95  (0078935, 'Cardano', 0068889),
    96  (0292746, 'Polkadot', 0077588),
    97  (0936444, 'Litecoin', 0099637),
    98  (3448997, 'Bitcoin', 1506287);
    99
    100  --
    101  --
    102  -- Table structure for table 'CryptoExchange'
    103  --
    104
    105  CREATE TABLE `CryptoExchange` (
    106  | `ExchangeID` int(7) UNSIGNED ZEROFILL NOT NULL,
    107  | `Name` varchar(50) NOT NULL,
    108  | `Website` varchar(50) NOT NULL
    109  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
    110
    111  --
    112  -- Dumping data for table 'CryptoExchange'
    113  --
    114
    115  INSERT INTO `CryptoExchange` (`ExchangeID`, `Name`, `Website`) VALUES
    116  (0003799, 'Bitcoin', 'https://bitcoinafrica.com'),
    117  (0006805, 'Dogecoin', 'https://blockfi.com'),
    118  (0292746, 'Cardano', 'https://www.kraken.com'),
    119  (0574228, 'Monero', 'https://crypto.com/eua/'),
    120  (0579329, 'Stellar', 'https://www.coinbase.com'),
    121  (0936444, 'Litecoin', 'https://www.binance.com/en'),
    122  (0977294, 'Polkadot', 'https://www.etoro.com');
    123
    124  --
    125  --
    126  --
    127  -- Table structure for table 'Influenced'
    128  --
    129
    130  CREATE TABLE `Influenced` (
    131  | `UserID` int(7) UNSIGNED ZEROFILL NOT NULL,
    132  | `PostID` int(7) UNSIGNED ZEROFILL NOT NULL
    133  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='--';

```

```

145 -- Dumping data for table `Influenced`
146 --
147 INSERT INTO `Influenced` (`UserID`, `PostID`) VALUES
148 (0832258, 0396954),
149 (0838437, 0554939),
150 (0839839, 0577838),
151 (0853233, 0588438),
152 (0859869, 0727237),
153 (0878474, 0961857),
154 (0908433, 0866986);
155 --
156 --
157 -- Table structure for table `NewTable`
158 --
159 CREATE TABLE `NewTable` (
160   `ID` varchar(50) DEFAULT NULL,
161   `Name` varchar(50) DEFAULT NULL,
162   `Phone` varchar(20) DEFAULT NULL
163 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
164 --
165 --
166 -- Table structure for table `Popularize`
167 --
168 CREATE TABLE `Popularize` (
169   `PostID` int(7) UNSIGNED ZEROFILL NOT NULL,
170   `Code` int(7) UNSIGNED ZEROFILL NOT NULL
171 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
172 --
173 -- Dumping data for table `Popularize`
174 --
175 INSERT INTO `Popularize` (`PostID`, `Code`) VALUES
176 (0396954, 0052538),
177 (0554939, 0078935),
178 (0577838, 0292746),
179 (0588438, 0936444),
180 (0727237, 3448997),
181 (0961857, 0029874),
182 (0866986, 0048864);
183 --
184 --
185 -- Table structure for table `Post`
186 --
187 CREATE TABLE `Post` (
188   `PostID` int(7) UNSIGNED ZEROFILL NOT NULL,
189   `Author` varchar(50) NOT NULL,
190   `Content` varchar(50) NOT NULL,
191   `Metadata` varchar(50) NOT NULL
192 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
193 --
194 -- Dumping data for table `Post`
195 --
196 INSERT INTO `Post` (`PostID`, `Author`, `Content`, `Metadata`) VALUES
197 (0396954, 'Nouha Lamrani', 'Post Facebook', 'Posts effectiveness '),
198 (0554939, 'Kas Khok', 'Post Twitter', 'Analysing social media performance'),
199 (0577838, 'Ion Pruteanu', 'Newspapers', 'Creation date and time'),
200 (0588438, 'Krystian Mowinski', 'Post Telegram', 'Social media engagement'),
201 (0727237, 'Abbos Abdullaev', 'Advertisement', 'Digital management'),
202 (0961857, 'Bayram Tosun', 'Magazines', 'Competitor data'),
203 (0866986, 'Jad Korayani', 'Instagram Post', 'custom taxonomies');
204 --
205 --
206 -- Table structure for table `Price`
207 --
208 CREATE TABLE `Price` (
209   `PriceID` int(7) UNSIGNED ZEROFILL NOT NULL,
210   `Value` double(7,2) UNSIGNED ZEROFILL NOT NULL,
211   `Day` date NOT NULL,
212   `Time` timestamp NOT NULL DEFAULT current_timestamp(),
213   `Fiatcurrencycode` varchar(35) NOT NULL,
214   `Code` int(7) UNSIGNED ZEROFILL NOT NULL,
215   `ExchangeID` int(7) UNSIGNED ZEROFILL NOT NULL
216 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

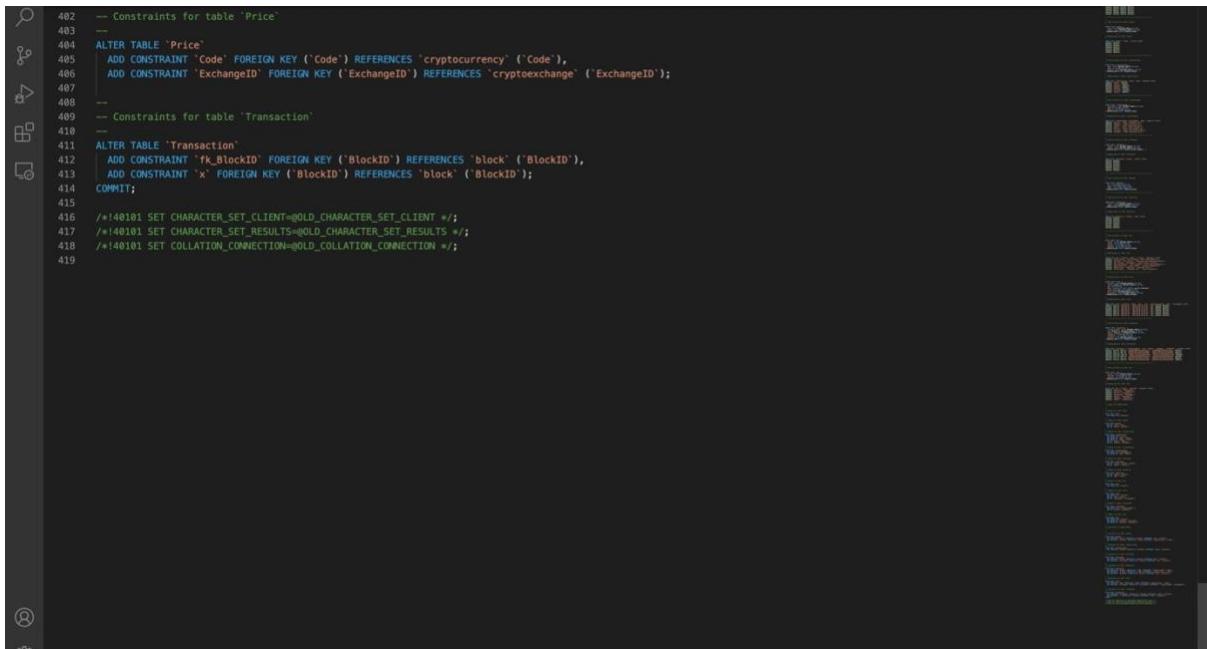
225
226  -- Dumping data for table 'Price'
227
228
229
230  INSERT INTO `Price` (`PriceID`, `Value`, `Day`, `Time`, `Fiatcurrencycode`, `Code`, `ExchangeID`) VALUES
231  (008922, 0565.67, '2022-04-14', '2022-04-08 14:57:02', 'CNY', 0052538, 0006885),
232  (0274347, 0456.45, '2022-04-24', '2022-04-08 16:25:36', 'BZR', 0048864, 0524228),
233  (0489388, 2345.43, '2022-04-29', '2022-04-08 16:25:36', 'HKD', 0936444, 0936444),
234  (0595834, 4566.45, '2022-04-28', '2022-04-08 16:25:36', 'EUR', 0078935, 0977294),
235  (0682453, 0987.45, '2022-04-11', '2022-04-08 16:25:36', 'GBP', 0292746, 0292746),
236  (0822433, 0554.59, '2022-04-26', '2022-04-08 16:25:36', 'ZAR', 3448997, 0579329),
237  (0838486, 8675.56, '2022-04-13', '2022-04-08 16:25:36', 'AUD', 0029874, 0003799);
238
239  -----
240
241  --
242  -- Table structure for table 'Transaction'
243
244
245  CREATE TABLE `Transaction` (
246    `TransactionHash` int(7) UNSIGNED ZEROFILL NOT NULL,
247    `Fee` double(7,2) UNSIGNED ZEROFILL NOT NULL,
248    `Amount` double(7,2) UNSIGNED ZEROFILL NOT NULL,
249    `InAddress` varchar(35) NOT NULL,
250    `OutAddress` varchar(35) NOT NULL,
251    `BlockID` int(7) UNSIGNED ZEROFILL NOT NULL
252 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
253
254
255  --
256  -- Dumping data for table 'Transaction'
257
258
259  INSERT INTO `Transaction` (`TransactionHash`, `Fee`, `Amount`, `InAddress`, `OutAddress`, `BlockID`) VALUES
260  (0022567, 0346.56, 5667.56, '3YugWd5P9s2bcj0fYouDVAcI', '3YugWd5P9s2bcj0fYouDVAcI', 0022533),
261  (0026846, 2345.21, 0355.22, '35MFYSXJN9VgcxEZ24R0cGRMHD', '35MFYSXJN9VgcxEZ24R0cGRMHD', 0038577),
262  (0027803, 0575.50, 4677.78, 'JWQ9etc401JH4sdfFFBF7FSM', '3WQ9etc401JH4sdfFFBF7FSM', 0065576),
263  (0078647, 45346.58, 0665.70, '3qNRNr7wszAEptEN0BCJYh1B', '3qNRNr7wszAEptEN0BCJYh1B', 0065899),
264  (00874868, 8475.48, 0567.56, '3yuw8AUyWcHr4o40u0c8d94jX', '3yuw8AUyWcHr4o40u0c8d94jX', 0077588),
265  (00887755, 0123.12, 1234.34, '3X6TE7duJfVLsPMrooy72MuQ', '3X6TE7duJfVLsPMrooy72MuQ', 0095637),
266  (009524, 3578.43, 6678.96, '3B90Hrafj9BTG78X0ne8uPb', '3B90Hrafj9BTG78X0ne8uPb', 1562577);
267
268  -----
269
270  -- Table structure for table 'User'
271
272
273  CREATE TABLE `User` (
274    `UserID` int(7) UNSIGNED ZEROFILL NOT NULL,
275    `Username` varchar(50) NOT NULL,
276    `Password` varchar(50) NOT NULL
277 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
278
279
280  --
281  -- Dumping data for table 'User'
282
283
284  INSERT INTO `User` (`UserID`, `Username`, `Password`) VALUES
285  (0032258, 'Nouha_45', '2hKlp1t4H'),
286  (0038437, 'Bayram_321', 'TfUEUKjCU'),
287  (0039839, 'kas_78', 'numedE05u'),
288  (0053239, 'Krystian98', '2r6YXxSMFu'),
289  (0059869, 'Jed123', 'Ybf10eXn!'),
290  (0078474, 'abbos167', 'Fp8v9mica'),
291  (0058433, 'Ion569', 'ty2h8Ccjsel');
292
293
294  --
295  -- Indexes for dumped tables
296
297
298  -- Indexes for table 'Block'
299
300  ALTER TABLE `Block`
301  ADD PRIMARY KEY (`BlockID`);
302
303  --
304  -- Indexes for table 'BuySell'
305
306  ALTER TABLE `BuySell`
307  ADD KEY `Code` (`Code`),
308  ADD KEY `UserID` (`UserID`);
309
310  --
311  -- Indexes for table 'Cryptocurrency'
312
313  ALTER TABLE `Cryptocurrency`

```

```

313 ADD PRIMARY KEY ('Code'),
314 ADD UNIQUE KEY 'Name' ('Name'),
315 ADD UNIQUE KEY 'Name_3' ('Name'),
316 ADD KEY 'Name_2' ('Name'),
317 ADD KEY 'BlockID' ('BlockID');
318 --
319 -- Indexes for table 'CryptoExchange'
320 --
321 ALTER TABLE `CryptoExchange`
322 ADD PRIMARY KEY ('ExchangeID'),
323 ADD UNIQUE KEY 'Name' ('Name');
324 --
325 -- Indexes for table 'Influenced'
326 --
327 ALTER TABLE `Influenced`
328 ADD KEY 'UserID' ('UserID','PostID'),
329 ADD KEY 'PostID' ('PostID');
330 --
331 -- Indexes for table 'Popularize'
332 --
333 ALTER TABLE `Popularize`
334 ADD KEY 'PostID' ('PostID'),
335 ADD KEY 'Code' ('Code');
336 --
337 -- Indexes for table 'Post'
338 --
339 ALTER TABLE `Post`
340 ADD PRIMARY KEY ('PostID');
341 --
342 -- Indexes for table 'Price'
343 --
344 ALTER TABLE `Price`
345 ADD PRIMARY KEY ('PriceID'),
346 ADD KEY 'Code' ('Code'),
347 ADD KEY 'ExchangeID' ('ExchangeID');
348 --
349 -- Indexes for table 'Transaction'
350 --
351 ALTER TABLE `Transaction`
352 ADD PRIMARY KEY ('TransactionHash'),
353 ADD KEY 'BlockID' ('BlockID');
354 --
355 -- Indexes for table 'User'
356 --
357 ALTER TABLE `user`
358 ADD PRIMARY KEY ('UserID'),
359 ADD UNIQUE KEY 'Username' ('Username'),
360 ADD UNIQUE KEY 'Password' ('Password');
361 --
362 -- Constraints for dumped tables
363 --
364 -- Constraints for table 'BuySell'
365 --
366 ALTER TABLE `BuySell`
367 ADD CONSTRAINT 'UserId' FOREIGN KEY ('UserID') REFERENCES `user` ('UserID'),
368 ADD CONSTRAINT 'fk1_Code' FOREIGN KEY ('Code') REFERENCES `cryptocurrency` ('Code');
369 --
370 -- Constraints for table 'Cryptocurrency'
371 --
372 ALTER TABLE `Cryptocurrency`
373 ADD CONSTRAINT 'BlockID' FOREIGN KEY ('BlockID') REFERENCES `block` ('BlockID');
374 --
375 -- Constraints for table 'Influenced'
376 --
377 ALTER TABLE `Influenced`
378 ADD CONSTRAINT 'PostID' FOREIGN KEY ('PostID') REFERENCES `post` ('PostID'),
379 ADD CONSTRAINT 'fk1_UserID' FOREIGN KEY ('UserID') REFERENCES `user` ('UserID');
380 --
381 -- Constraints for table 'Popularize'
382 --
383 ALTER TABLE `Popularize`
384 ADD CONSTRAINT 'fk1Code' FOREIGN KEY ('Code') REFERENCES `cryptocurrency` ('Code'),
385 ADD CONSTRAINT 'fk_Code' FOREIGN KEY ('Code') REFERENCES `cryptocurrency` ('Code'),
386 ADD CONSTRAINT 'fk_PostID' FOREIGN KEY ('PostID') REFERENCES `post` ('PostID');
387 --
388

```



```
402 -- Constraints for table 'Price'
403 --
404 ALTER TABLE `Price`
405 | ADD CONSTRAINT `Code` FOREIGN KEY (`Code`) REFERENCES `cryptocurrency` (`Code`),
406 | ADD CONSTRAINT `ExchangeID` FOREIGN KEY (`ExchangeID`) REFERENCES `cryptoexchange` (`ExchangeID`);
407 |
408 --
409 -- Constraints for table 'Transaction'
410 --
411 ALTER TABLE `Transaction`
412 | ADD CONSTRAINT `fk_BlockID` FOREIGN KEY (`BlockID`) REFERENCES `block` (`BlockID`),
413 | ADD CONSTRAINT `x` FOREIGN KEY (`BlockID`) REFERENCES `block` (`BlockID`);
414 |
415 COMMIT;
416 /*+140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
417 /*+140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
418 /*+140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
419
```

## 7.Demonstration of the scenario:

SQL is a declarative language that allows the user specifying a certain expected result and it is considered as the standard language for commercial relational DBMSs. In this report I was tasked with creating a database system for the smooth and efficient operation of a crypto mining business. I found in the beginning this business complicated, and that its relationships between entities should be carefully considered. In part one of the coursework, I gave an overview of my idea of the DBMS I chose to design and develop, my work objectives, and the purpose of having a DBMS I have proposed. Alongside with a logical relational database representation (ERD) and a Formal Data Modelling with detailed features of my system. I made sure to give detailed description in my report as the relationship might be hard to understand by readers, my next challenge was to identify all the foreign keys. While part two, was a successful implementation of the system from Part One using SQL by following the formal data model. I created database tables, database constraints, dropped existing tables and inserted appropriate values into tables. In addition to a performance of SELECT operations

using: (special operators, comparison/mathematical operators, logical operators...). Several screenshots were taken of every single table, both structurally and populated with proper and meaningful data in them. I also designed 15 questions/tasks in my application domain that are needed to query the database, including five complex queries. Some problem I faced during this second part was, finding complex queries and using new Select operations that were not working sometimes. I learned from this coursework, to evaluate a set of queries processing strategies, using SQL to create tables and retrieve (SELECT) information from a database and creating a relational database schema in SQL. From one side my database has multiple strengths such as simple data structure, support of different type of queries and an independence in the physical and logical data. From the other side the weaknesses, for example difficulty in representing some of the complex queries, limited as SQL supports only a restricted number of operations and the use of some relations which might not exist in the real world. If I was given more time to improve it, I would consider adding more tables and relations to extend it and try make it clearer and more understandable for everyone, as an example ‘pool’ table. A pool in cryptocurrency a group of users who share their resources to reinforce the probability of finding a block, in this case this table will be related to ‘Block’ table and ‘User’ table. The relations would be [group] between ‘user’ and ‘pool’ and [find] between ‘block’ and ‘pool’. I would also try more complex queries and try to make sure to avoid using MySQL keywords as attributes name if there is any. In conclusion, as future development, I would consider working on different types of businesses to be able to discover new challenges and practice more SQL.