

Birla Institute of Technology and Science

CS F212

Database Systems



**PORTFOLIO MANAGEMENT SYSTEM
DOCUMENTATION**

Submitted to: Dr. Amit Dua & Parth Patel

Submitted by:

Agrawal Vansh Anil (2021A7PS2215P)

Priyansh Patel (2021A7PS2682P)

Introduction-

This project aims towards creating a database system for the purpose of Portfolio management. The schema has several different tables. There are several functionalities like get holdings, delete holdings, update holdings, get riskLevel, get total returns many more. These functions can be directly performed by executing the SQL queries or through the user interface. Several assumptions are made for designing the schema mentioned in the document itself.

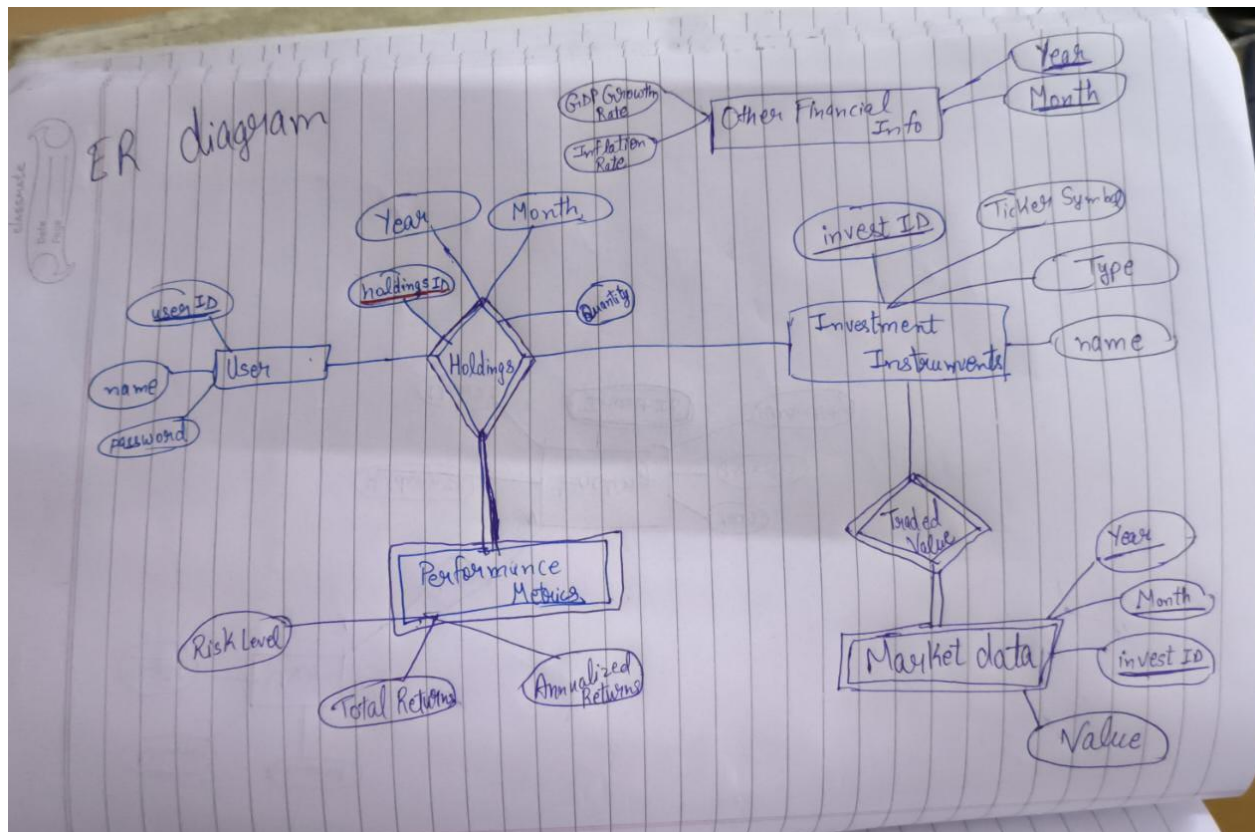
The snapshots of ER diagram and relational schema are pasted in the document. These can also be viewed in the figma file:

<https://www.figma.com/file/XYJzjRqeNMIz16db6uQck/Hospital-Management-System?node-id=0-1&t=iFQBpv6QzYzWrxX1-0>

The drive links for the video recordings are attached at the end of this document.

Note: There were certain instances in our ER diagram where we could have used weak entities and weak relations but we tried to avoid those and deal with strong entities/relations only.

EER Diagram:-



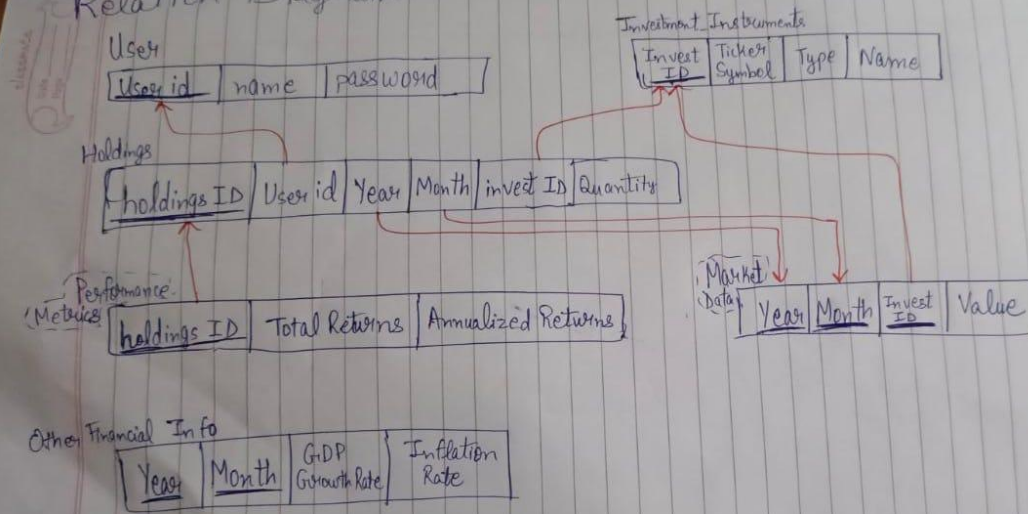
Entities with their attributes -

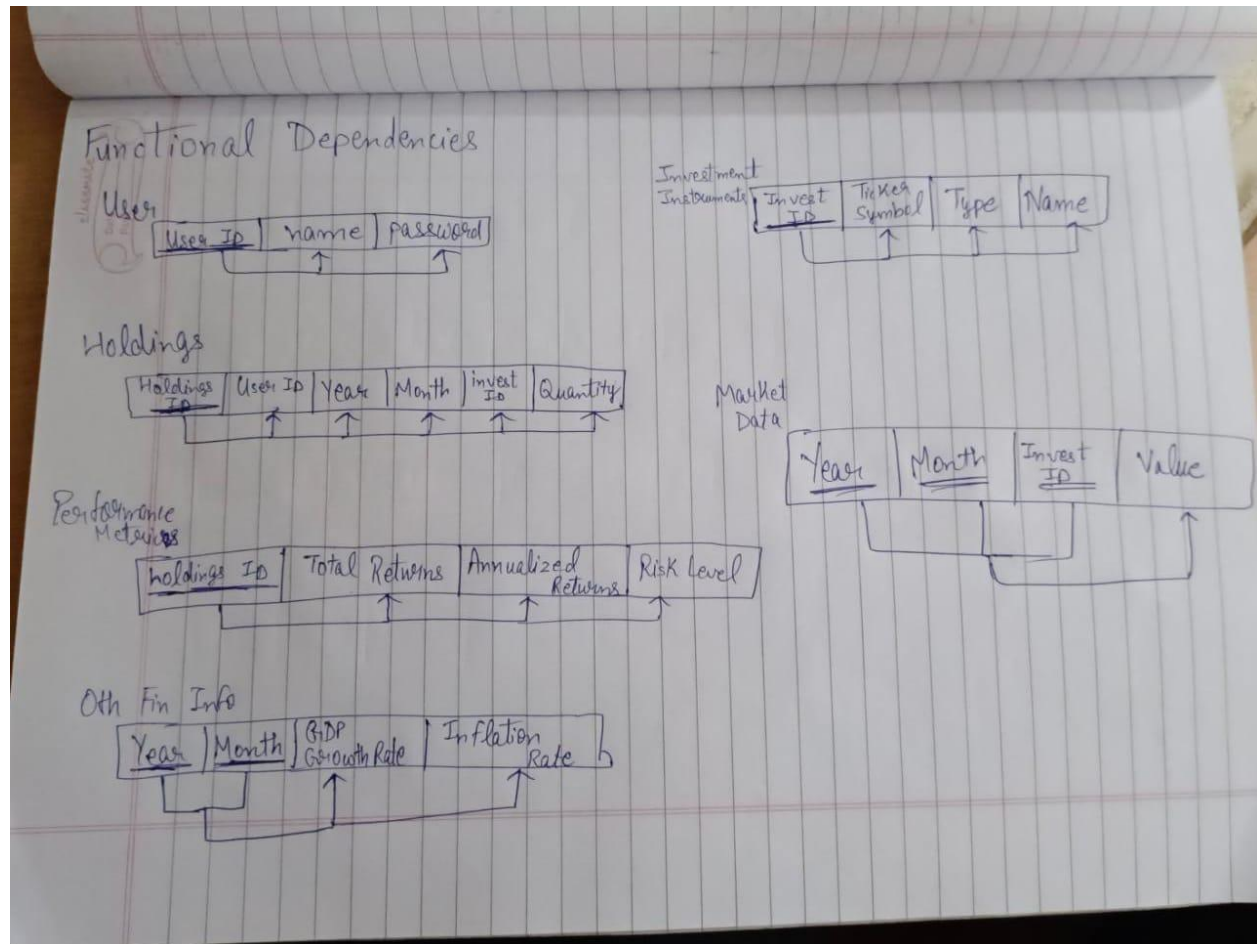
- 1) **User** - UserID(primary key-int), name(varchar), password(varchar)
- 2) **Investment Instruments** - invest id(int), ticker symbol(varchar), type(varchar), name(varchar).
- 3) **Market Data**- year(primary key-int), month(primary key-int), Invest id(primary key, int), value(double)
- 4) **Performance Metrics** - holdingsID(primary key-int), totalReturns(double), annualized returns(double), riskLevel(varchar)
- 5) **Other Financial info**- year(primary key-int), month(Primary key-int) and inflation rate(double),GDP growth rate(double)

Relationships-

- 1) **Holdings** - userID and holdingsID(Primary key-int). This is a relationship between 3 entities. The relation between *user* and *holdings* with 1:N cardinality which semantically means that a user can have multiple holdings, but a particular holding will be by only one user because user can have several holdings. The relation has total participation from holdings sides as each user can exist even without a holding but a holding should have a user always. It's a relation between *holdings* and *investment instruments* with N:1 cardinality which means that a holding is having a particular investment (stock or any type) but a particular stock can be in multiple holdings. *holdings* has total participation, while the *instrument* has partial participation.
- 2) **Traded Value** - investID(Primary key-int), year(int), month(int) - It's a relation between *investment instruments* and *Market data* with 1:N cardinality which means that an instrument can have multiple market data according to date(year,month), and one market data will be related to only one investment instrument.

Relation Diagram





RELATIONAL SCHEMA-

SQL Queries-

1) Create investment Tables

```

CREATE TABLE `holdings` (
  `holdingsID` int NOT NULL AUTO_INCREMENT,
  `userID` int NOT NULL,
  `year` int NOT NULL,
  `month` int NOT NULL,
  `investID` int NOT NULL,
  `quantity` int DEFAULT '0',
  PRIMARY KEY (`holdingsID`),
  KEY `has_idx` (`userID`),
  KEY `are type of_idx` (`investID`),
  CONSTRAINT `are type of` FOREIGN KEY (`investID`) REFERENCES `investment_instruments` (`investID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `has` FOREIGN KEY (`userID`) REFERENCES `user` (`userID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

2) Create Performance metrics Table

```
CREATE TABLE `performance_metrics` (  
  `holdingsID` int NOT NULL,  
  `totalReturns` decimal(10,2) NOT NULL,  
  `annualizedReturns` decimal(10,2) NOT NULL,  
  `riskLevel` varchar(20) NOT NULL,  
  PRIMARY KEY (`holdingsID`),  
  CONSTRAINT `have` FOREIGN KEY (`holdingsID`) REFERENCES `holdings` (`holdingsID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

3) Create market data table

```
CREATE TABLE `market_data` (  
  `year` int NOT NULL,  
  `month` int NOT NULL,  
  `investID` int NOT NULL,  
  `value` double NOT NULL,  
  PRIMARY KEY (`investID`,`year`,`month`),  
  CONSTRAINT `constitute market data` FOREIGN KEY (`investID`) REFERENCES `investment_instruments` (`investID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

4) Create other financial information table

```
CREATE TABLE `other_financial_info` (  
  `month` int NOT NULL,  
  `year` int NOT NULL,  
  `gdpGrowth` double NOT NULL,  
  `inflation` double NOT NULL,  
  PRIMARY KEY (`month`,`year`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

5) Insert a new investment into the investments table.

```
INSERT INTO `portfolio`.`holdings`  
(`holdingsID`,  
  `userID`,  
  `year`,  
  `month`,  
  `investID`,  
  `quantity`)  
VALUES  
(100,1,2023,2,1,1);
```


6) Update the number of shares held for an investment in the investments table.

```

21
22 • CALL `portfolio`.`update_holding`(1, 1, 2022, 1, 1, 100);
23 • select * from holdings


```


Result Grid




Filter Rows:


Edit:







Export/Import:



	holdingsID	userID	year	month	investID	quantity
	1	1	2022	1	1	100
	2	1	2022	2	1	8
	3	1	2022	3	1	3
	4	1	2022	4	1	4
	5	1	2022	5	1	1
	6	1	2022	6	1	7
	7	1	2022	7	1	5
	8	1	2022	8	1	4

We have made a procedure named update holdings and here it is being called.

7) Delete an investment from the investments table.

```

8 • CALL `portfolio`.`delete_holding`(1);
9 • select * from holdings
10

```

Result Grid

Filter Rows:

Edit:

	holdingsID	userID	year	month	investID	quantity
▶	2	1	2022	2	1	8
	3	1	2022	3	1	3
	4	1	2022	4	1	4
	5	1	2022	5	1	1
	6	1	2022	6	1	7

We have made a procedure for this named delete_holding.

In 6th query, we edited the quantity for holdingID = 1 and it is shown in the above figure.

Now, it got deleted because of this procedure call.

8)Join the investments table with the performance metrics table to retrieve the total return for each investment.


```

1 • SELECT *
2   FROM holdings
3   NATURAL JOIN performance_metrics;

```

Result Grid Filter Rows: Export: Wrap Cell Content:									
	holdingsID	userID	year	month	investID	quantity	totalReturns	annualizedReturns	riskLevel
▶	2	1	2022	2	1	8	8.37	8.42	Medium
	3	1	2022	3	1	3	13.85	13.94	High
	4	1	2022	4	1	4	20.60	20.73	High
	5	1	2022	5	1	1	16.76	16.93	High
	6	1	2022	6	1	7	15.26	15.48	High
	7	1	2022	7	1	5	3.86	4.11	Low
	8	1	2022	8	1	4	13.57	13.91	High
	9	1	2022	9	1	4	7.95	8.36	Medium
	10	1	2022	10	1	4	-4.73	-4.26	Negative
	11	1	2022	11	1	7	-6.84	-6.20	Negative
	13	1	2021	1	1	2	19.40	19.18	High
	14	1	2021	2	1	1	0.73	0.56	Low
	15	1	2021	3	1	8	3.46	3.30	Low
	16	1	2021	4	1	3	0.91	0.76	Low
	17	1	2021	5	1	5	3.52	3.39	Low

9) Join the investments table with the market data table to retrieve the stock prices for a particular date.

```

8 • CALL `portfolio`.`delete_holding` (1);
9 • select * from holdings
10

```

Result Grid Filter Rows: Edit:						
	holdingsID	userID	year	month	investID	quantity
▶	2	1	2022	2	1	8
	3	1	2022	3	1	3
	4	1	2022	4	1	4
	5	1	2022	5	1	1
	6	1	2022	6	1	7

10) Group the investments by type and retrieve the average annualized return for each type and retrieve the total number of shares held for each type.

```
1 • SELECT ii.type, AVG(pm.annualizedReturns) AS avg_annualized_return, SUM(h.quantity) AS total_quantity
2   FROM holdings h
3  JOIN investment_instruments ii ON h.investID = ii.investID
4  JOIN performance_metrics pm ON h.holdingsID = pm.holdingsID
5  GROUP BY ii.type;
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	type	avg_annualized_return	total_quantity			
B		14.135455	107			

11) Filter the investments by risk level and retrieve the top-performing investments.

```
2
3 • CALL `portfolio`.`get_top_performing_holdings`('low');
```

Result Grid								Filter Rows:	Export:	Wrap Cell Content:
	holdingsID	year	month	investID	name	quantity	totalReturns			
▶	7	2022	7	1	SnP Tech Index	5	3.86			
	17	2021	5	1	SnP Tech Index	5	3.52			
	15	2021	3	1	SnP Tech Index	8	3.46			
	16	2021	4	1	SnP Tech Index	3	0.91			
	14	2021	2	1	SnP Tech Index	1	0.73			

12) Calculate the total value of all investments based on the number of shares held and the current stock prices from the market data table.

```

1 • SELECT SUM(h.quantity * value) AS total_value
2 FROM holdings h
3 JOIN investment_instruments ii ON h.investID = ii.investID
4 JOIN (SELECT investID, MAX(year) AS latest_year, MAX(month) AS latest_month
5       FROM market_data
6       GROUP BY investID) latest_md
7 ON ii.investID = latest_md.investID
8 JOIN market_data md
9 ON latest_md.investID = md.investID
10    AND latest_md.latest_year = md.year
11    AND latest_md.latest_month = md.month;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_value			
	1687189.2			

13) Calculate the portfolio's overall annualized return based on the investments' individual returns and the number of shares held.

```

1 • SELECT SUM(h.quantity * pm.annualizedReturns) / SUM(h.quantity) AS portfolio_annualized_return
2 FROM holdings h
3 JOIN performance_metrics pm ON h.holdingsID = pm.holdingsID;
4

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	portfolio_annualized_return			
	14.256636			

14) Calculate the correlation between two investments' performance using the performance metrics table.




```

• SELECT
    (SUM(x.totalReturns * y.totalReturns) - SUM(x.totalReturns) * SUM(y.totalReturns) / COUNT(*)) /
    (SQRT((SUM(x.totalReturns * x.totalReturns) - SUM(x.totalReturns) * SUM(x.totalReturns) / COUNT(*)) *
    (SUM(y.totalReturns * y.totalReturns) - SUM(y.totalReturns) * SUM(y.totalReturns) / COUNT(*))))
AS correlation
FROM
    performance_metrics AS x
INNER JOIN
    performance_metrics AS y ON x.holdingsID < y.holdingsID
WHERE
    x.holdingsID = 1 AND y.holdingsID = 2;

```





15)Retrieve the most recent inflation rate from the other financial information table.

```
1 • SELECT inflation
2 FROM other_financial_info
3 ORDER BY year DESC, month DESC
4 LIMIT 1;
```

Result Grid			Filter Rows: <input type="text"/>	Export: 
	inflation			
	7			

16)Filter the market data table by date range and retrieve the stock prices for a particular investment.

```
1 • SELECT value, year, month
2 FROM market_data
3 WHERE investID = (SELECT investID FROM holdings WHERE holdingsID=3)
4 AND (year > 2021 OR (year = 2021 AND month >= 2))
5 AND (year < 2022 OR (year = 2022 AND month <= 6));
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	value	year	month		
▶	15366.54	2021	2		
	14961.08	2021	3		
	15339.8	2021	4		
	14952.38	2021	5		
	13480.34	2021	6		
	13229.54	2021	7		
	12261.64	2021	8		
	11802.88	2021	9		
	11919.91	2021	10		
	11232.82	2021	11		
	11380.96	2021	12		
	13413.34	2022	1		
	14283.64	2022	2		
	13505.68	2022	3		

17) Calculate the percentage change in stock prices for a particular investment between two dates.

```
1 • SELECT ((MAX(value) - MIN(value))/MIN(value))*100 AS percentage_change
2 FROM market_data
3 WHERE investID = 1
4 AND (year = 2021 AND month >= 1)
5 OR (year = 2021 AND month <= 4)
6 OR (year > 2021 AND year < 2023)
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
percentage_change				
▶	47.92509806086095			

Steps for front end:-

- 1) Download the portfolio.py file.
- 2) Import the database in mySql.
- 3) Find and change all the instance of "root" to your host name and "dbms@123" to pwd of your database.
- 4) Install tk and mysql-connector-python using -
pip install tk
Pip install mysql-connector-python
Perform the above 2 commands in terminal
- 5) userID = 1, password = 123
- 6) We have implemented frontend for the general things such as seeing holdings table, viewing market data, viewing investment instruments available.
- 7) To generate reports, sql queries are needed.

Video folder

https://drive.google.com/drive/folders/1s_JXETGezZ9f46-kXtG7VW7MHn2dBVEI?usp=share_link