Coursework

DATA MANAGEMENT SYSTEMS (7086CEM)

Name: Jusel Justin
SID: 12646627
Mail ID: juseljustj@uni.coventry.ac.uk
Module Leader: Dr Rachid Anane

Table of Contents

Part A: Coventry Medical Group

Q1: Entity relationship diagram for Coventry Medical group

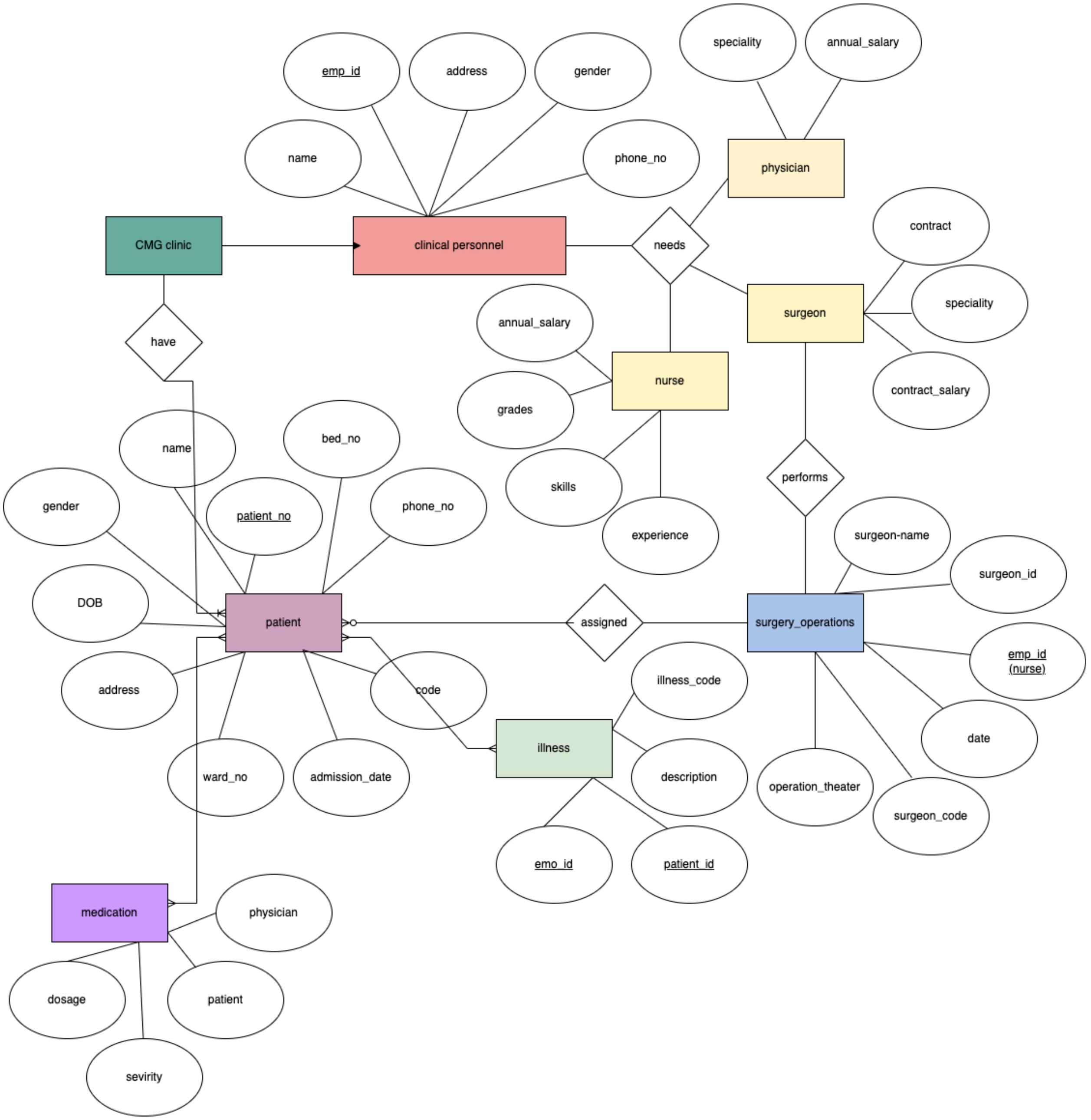


Figure 0.1: Coventry ER-diagram of attributes and entities

The major characteristic of an entity relation diagram (ER diagram) is to make a pictorial representation of an actual relationship between entities. Here in the above image an ER model have been designed in the Draw.io platform for the medical department. Basically, ER diagram helps to dictate the logical pattern of a database (Dièneet *al*. 2020). Here three symbols have been used to make the diagram such as rectangle, ovals and diamonds. ER model can analyse the necessities of the data to build appropriate database.

Q2: Coventry Relational tables

cliinic_personnel (**emp_id,** name, gender, address, phone_no)
physician (speciality, annual_salary)
surgeon (speciality, contract, contract_salary)
nurse (annual_salary, grades, skills, experience)
surgeory_operation (emp_id*, surgeon_code,operation_theater, patient, surgeon_name, date)
patient (**patient_id**, address, phone_no, DOB, gender, admission_date, ward_no, bed_no)
Illness (emp_id*, patientID*, illness_code, description)
medication (patient_id*, physician, dosage, severity)

The employees at CMG are physicians, nurses and surgeons. Under table clinic_personnel we have attributes emp_id (primary key) as it is identified as a unique value. Under table surgery_operations we have noticed that emp_id is the foreign key as it has a unique value and is already a primary key in another table, hence it is a foreign key. Taking about the table patient, we have included patient_id (primary key) to fetch the data related to the patient from the database. In the illness table emp_id and patient_id are foreign keys. In medication table, patient_id is the foreign key to understand which patient needs which medication a prescribed.

Part B: Table creating SQL programming

Q1: Table creation

First table

```
CREATE TABLE USER_Information (
user_IdVARCHAR(300) primary key,

  name    VARCHAR(500),
email_AddressVARCHAR(300)
);

INSERTINTO  USER_Information VALUES ('kendj3kenderine, Jkendj3@hotmail.co.uk');

INSERTINTO  USER_Information VALUES ('Patel11PatelFpatel11@ntl.co.uk');

INSERTINTO  USER_Information VALUES ('flak05FlavelKflak05@freeserve.co.uk');

INSERTINTO  USER_Information VALUES ('johnsj9JohnsonJjohnsj9@msn.co.uk');

INSERTINTO  USER_Information VALUES ('keita77KeitaRkeita77@hotmail.co.uk');

INSERTINTO  USER_Information VALUES ('Simpb91SimpsonBSimpb91@tesco.co.uk');

SELECT * FROM USER_Information
```

Second table

```
CREATE TABLE Music_Details (
music_IdVARCHAR(300) primary key,
title VARCHAR(500),
category_CodeVARCHAR(300),
publisher_IdVARCHAR(300),
cost_Per_DownloadFLOAT(125)
);
```

```
INSERTINTO  Music_Details VALUES ('M001James Bond: Golden EyesC13P0010.99');

INSERTINTO  Music_Details VALUES ('M002Lake HouseC13P0021.99');

INSERTINTO  Music_Details VALUES ('M003DvorakSymphony No 9C11P0031.49');

INSERTINTO  Music_Details VALUES ('M004HandelWater MusicC11P0041.79');

INSERTINTO  Music_Details VALUES ('M005Sense and SensibilityC13P0051.50');

INSERTINTO  Music_Details VALUES ('M006BeatlesYesterdayC12P0051.10');

INSERTINTO  Music_Details VALUES ('M007Elton JohnYour SongC12P0060.89');

SELECT * FROM Music_Details
```

Third table

```
CREATE TABLE Music_Download (
user_IdVARCHAR(300) primary key,
music_IdVARCHAR(300),
download_Date DATE
);

INSERTINTO  Music_Download VALUES ('kendj3M00203-May-18');

INSERTINTO  Music_Download VALUES ('johnsj9M00501-May-19');

INSERTINTO  Music_Download VALUES ('patel11M00206-May-18');

INSERTINTO  Music_Download VALUES ('johnsj9M00106-May-19');
INSERTINTO  Music_Download VALUES ('kendj3M00301-Aug-19');

INSERTINTO  Music_Download VALUES ('Keita77M00402-Aug-19');

INSERTINTO  Music_Download VALUES ('Simpb91M00705-Sep-18');

SELECT * FROM Music_Download
```

Fourth table

```
CREATE TABLE Music_Genre (
category_CodeVARCHAR(300) primary key,
title VARCHAR(500)
);

INSERTINTO  Music_Genre VALUES ('C11', 'Classics');

INSERTINTO  Music_Genre VALUES ('C12', 'Pop-Rock');

INSERTINTO  Music_Genre VALUES ('C13', 'Movie Soundtrack');

SELECT * FROM Music_Genre
```

Fifth table

```
CREATE TABLE Publishers (
publisher_IdVARCHAR(300) primary key,
publisher_NameVARCHAR(500)
);
```

```
INSERTINTO  Publishers VALUES ('P001Arista Records');

INSERTINTO  Publishers VALUES ('P002Lakeshore Records');

INSERTINTO  Publishers VALUES ('P003EMI');

INSERTINTO  Publishers VALUES ('P004DECCA');

INSERTINTO  Publishers VALUES ('P005Sony Music');

INSERTINTO  Publishers VALUES ('P006DJM Records');

SELECT * FROM Publishers
```

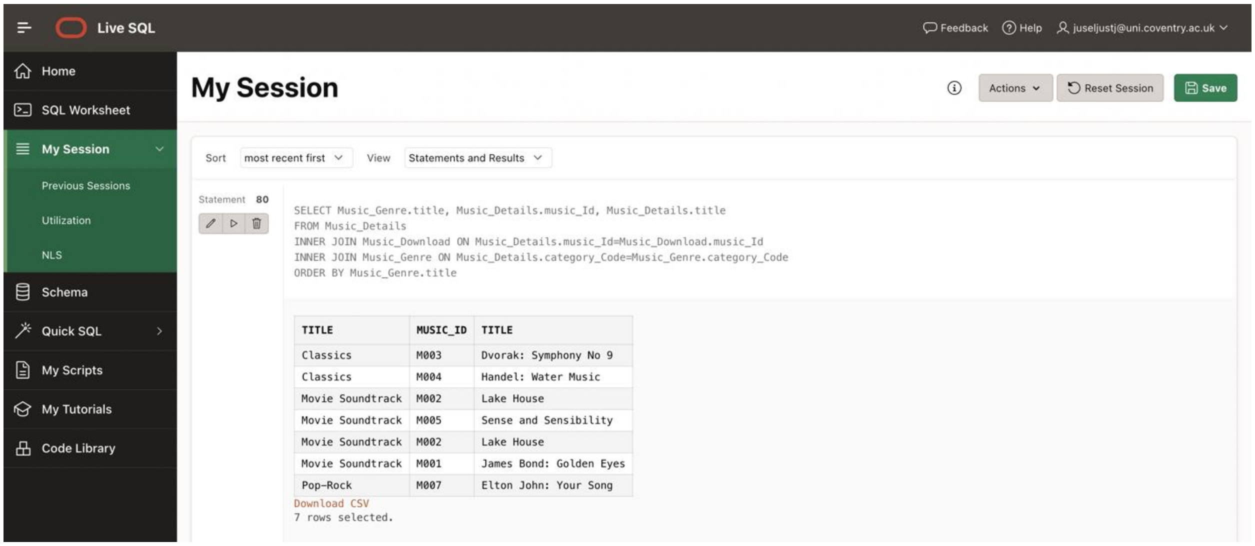
Q2: SQL statements

a) Statement number 1

SQL Query

```
SELECT Music_Details.music_Id, Music_Details.title, Publishers.publisher_Name
FROM Music_Details
INNER JOIN Publishers
ON Music_Details.publisher_Id=Publishers.publisher_Id
order by Music_Details.title;
```

Output solutions basic



Sort: most recent first View: Statements and Results

Statement 80

```
SELECT Music_Genre.title, Music_Details.music_Id, Music_Details.title
FROM Music_Details
INNER JOIN Music_Download ON Music_Details.music_Id=Music_Download.music_Id
INNER JOIN Music_Genre ON Music_Details.category_Code=Music_Genre.category_Code
ORDER BY Music_Genre.title
```

TITLE	MUSIC_ID	TITLE
Classics	M003	Dvorak: Symphony No 9
Classics	M004	Handel: Water Music
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M005	Sense and Sensibility
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M001	James Bond: Golden Eyes
Pop-Rock	M007	Elton John: Your Song

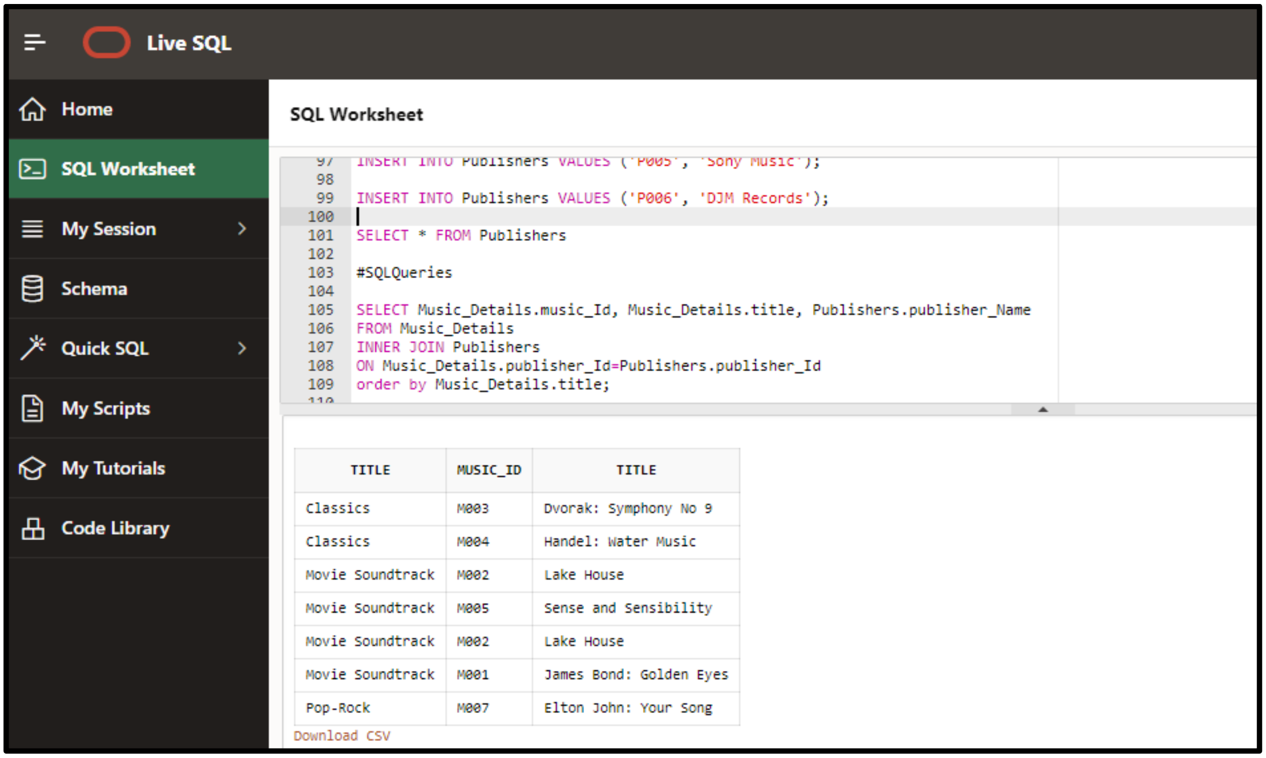
Download CSV
7 rows selected.

Figure 2: Outcome of the problem

(Source: Acquired from SQL)

The result has been successfully developed by applying the required query statement.

Evidence of the resultss



```
INSERT INTO Publishers VALUES ('P005', 'Sony Music');

INSERT INTO Publishers VALUES ('P006', 'DJM Records');

SELECT * FROM Publishers

#SQLQueries

SELECT Music_Details.music_Id, Music_Details.title, Publishers.publisher_Name
FROM Music_Details
INNER JOIN Publishers
ON Music_Details.publisher_Id=Publishers.publisher_Id
order by Music_Details.title;
```

TITLE	MUSIC_ID	TITLE
Classics	M003	Dvorak: Symphony No 9
Classics	M004	Handel: Water Music
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M005	Sense and Sensibility
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M001	James Bond: Golden Eyes
Pop-Rock	M007	Elton John: Your Song

Download CSV

Figure 3: Result

(Proper Source: Acquired from SQL)

The image shows that the software work has been properly done in Oracle live SQL server.

b) Statement number 2

SQL query

SELECT Music_Download.user_Id,Music_Details.music_Id,Music_Genre.title

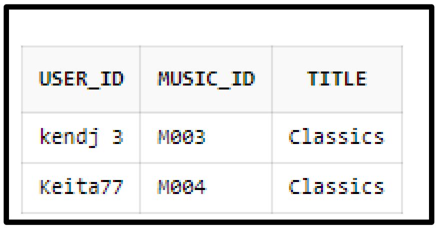
FROM Music_Download

INNER JOIN Music_Details ON Music_Download.music_Id=Music_Details.music_Id

INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code

WHERE Music_Genre.title='Classics';

Output solution



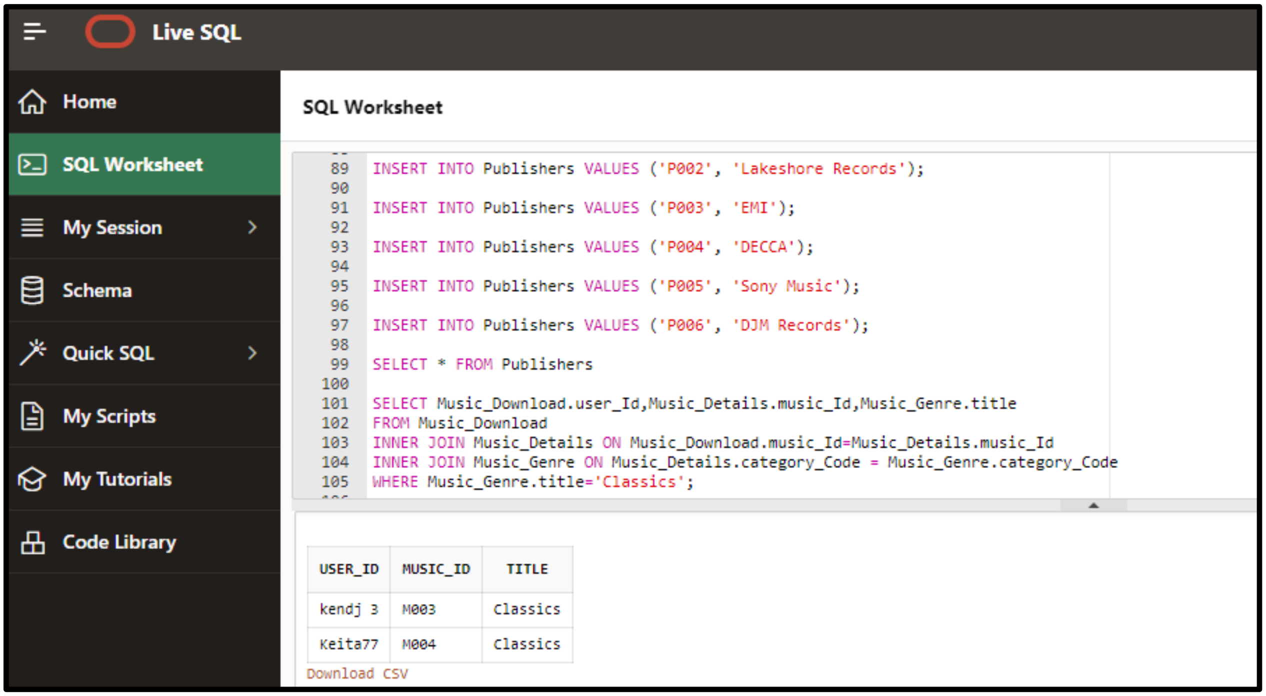
USER_ID	MUSIC_ID	TITLE
kendj 3	M003	Classics
Keita77	M004	Classics

Figure 4: Outcome

(Proper Source: Acquired from SQL)

Figure 4 shows the users with the classic music genre has been implemented.

Main Evidence of result



```
INSERT INTO Publishers VALUES ('P002', 'Lakeshore Records');

INSERT INTO Publishers VALUES ('P003', 'EMI');

INSERT INTO Publishers VALUES ('P004', 'DECCA');

INSERT INTO Publishers VALUES ('P005', 'Sony Music');

INSERT INTO Publishers VALUES ('P006', 'DJM Records');

SELECT * FROM Publishers

SELECT Music_Download.user_Id,Music_Details.music_Id,Music_Genre.title
FROM Music_Download
INNER JOIN Music_Details ON Music_Download.music_Id=Music_Details.music_Id
INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code
WHERE Music_Genre.title='Classics';
```

USER_ID	MUSIC_ID	TITLE
kendj 3	M003	Classics
Keita77	M004	Classics

Figure 5: Result of the SQL query statement

(Source: Acquired from SQL)

The SQL worksheet stage displays how the task has been performed.

c) Statement number 3

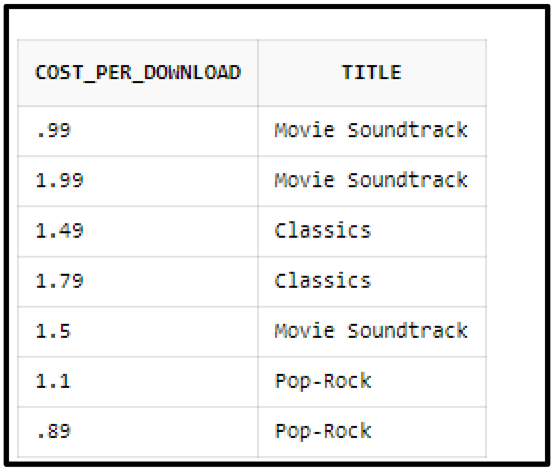
SQL query

SELECT Music_Details.cost_Per_Download, Music_Genre.title

FROM Music_Details

INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code

Output of the solution



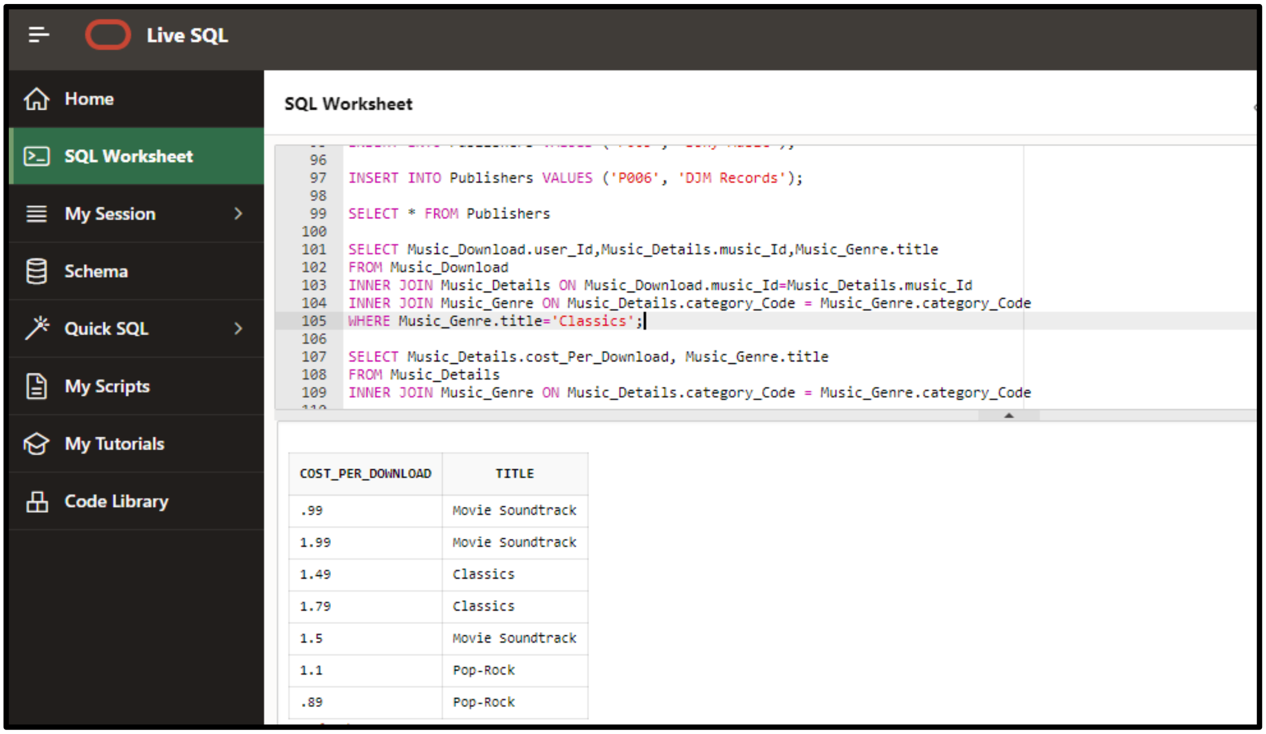
COST_PER_DOWNLOAD	TITLE
.99	Movie Soundtrack
1.99	Movie Soundtrack
1.49	Classics
1.79	Classics
1.5	Movie Soundtrack
1.1	Pop-Rock
.89	Pop-Rock

Figure 6: Outcome of the SQL query

(Source: Acquired from SQL)

The number of downloaded movies with different genres has been successfully developed.

Evidence of result



SQL Worksheet

```

INSERT INTO Publishers VALUES ('P006', 'DJM Records');

SELECT * FROM Publishers

SELECT Music_Download.user_Id,Music_Details.music_Id,Music_Genre.title
FROM Music_Download
INNER JOIN Music_Details ON Music_Download.music_Id=Music_Details.music_Id
INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code
WHERE Music_Genre.title='Classics';

SELECT Music_Details.cost_Per_Download, Music_Genre.title
FROM Music_Details
INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code
```

COST_PER_DOWNLOAD	TITLE
.99	Movie Soundtrack
1.99	Movie Soundtrack
1.49	Classics
1.79	Classics
1.5	Movie Soundtrack
1.1	Pop-Rock
.89	Pop-Rock

Figure 7: SQL worksheet

(Source: Acquired from SQL)

The entire SQL programming has been done in the worksheet section accurately.

d) Statement number 4

SQL query

SELECT Music_Genre.title, Music_Details.music_Id, Music_Details.title

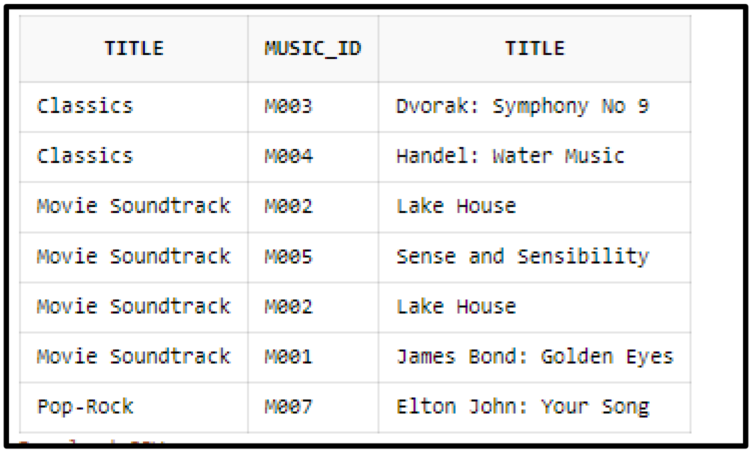
FROM Music_Details

INNER JOIN Music_Download ON Music_Details.music_Id=Music_Download.music_Id

INNER JOIN Music_Genre ON Music_Details.category_Code=Music_Genre.category_Code

ORDER BY Music_Genre.title

Output of the SQL query



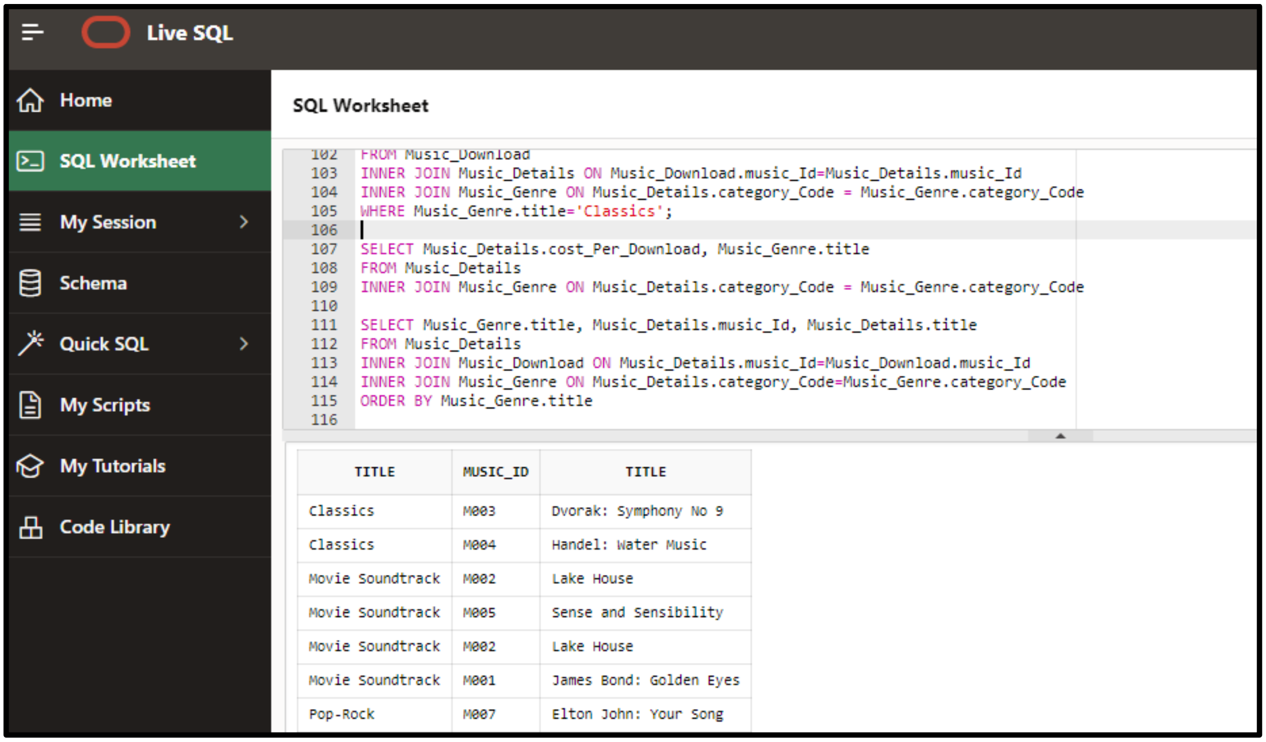
TITLE	MUSIC_ID	TITLE
Classics	M003	Dvorak: Symphony No 9
Classics	M004	Handel: Water Music
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M005	Sense and Sensibility
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M001	James Bond: Golden Eyes
Pop-Rock	M007	Elton John: Your Song

The Figure 8: SQL query solution

(Source: Acquired from SQL)

The movies of particular music id have been developed with the title of the music categories.

SQL worksheet



SQL Worksheet

```
FROM Music_Download
INNER JOIN Music_Details ON Music_Download.music_Id=Music_Details.music_Id
INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code
WHERE Music_Genre.title='Classics';

SELECT Music_Details.cost_Per_Download, Music_Genre.title
FROM Music_Details
INNER JOIN Music_Genre ON Music_Details.category_Code = Music_Genre.category_Code

SELECT Music_Genre.title, Music_Details.music_Id, Music_Details.title
FROM Music_Details
INNER JOIN Music_Download ON Music_Details.music_Id=Music_Download.music_Id
INNER JOIN Music_Genre ON Music_Details.category_Code=Music_Genre.category_Code
ORDER BY Music_Genre.title

```

TITLE	MUSIC_ID	TITLE
Classics	M003	Dvorak: Symphony No 9
Classics	M004	Handel: Water Music
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M005	Sense and Sensibility
Movie Soundtrack	M002	Lake House
Movie Soundtrack	M001	James Bond: Golden Eyes
Pop-Rock	M007	Elton John: Your Song

Figure 9: SQL statements in the worksheet

(Source: Acquired from SQL)

The SQL query has been successfully run in the software platform and the output is given above.

Part C: Sequential-Distributed processing

Q1:

a) The flight logical table

SQL statement to design table

```
CREATE TABLE  FLIGHT_DETAILS (

Year  VARCHAR(100) ,

Carrier  numeric(50) primary key,

Month VARCHAR(100),

Day_of_MonthVARCHAR(100),

Flight_Number  VARCHAR(100),

Departure_DelayVARCHAR(100),

Day_of_the_WeekVARCHAR(100),

Departure_TimeVARCHAR(100),

Actual_Departure_timeVARCHAR(100),

Arrival_TimeVARCHAR(100),

Arrival_DelayVARCHAR(100),

Cancellation VARCHAR(100),

Weather_DelayVARCHAR(100)

);
```

```
INSERTINTO  FLIGHT_DETAILS VALUES (‘2001’, ‘002’, '5', '4', 'F125', '15', '1', '2000',
'2025', '2330', '15', 'No', '0');

INSERTINTO  FLIGHT_DETAILS VALUES (‘2002’, '004', '4', '3', 'F145', '15', '2', '1200',
'1215', '1630', '15', 'yes', '1');

INSERTINTO  FLIGHT_DETAILS VALUES (‘2003’, ‘005’, '5”', '2', 'F631', '16', '4', '1830',
'1846', '2230', '16', 'yes', '3');

INSERTINTO  FLIGHT_DETAILS VALUES (‘2004’, ‘007’, '4', '10', 'F293', '20', '3', '1600',
'1620', '2000', '20', 'No', '5');

INSERTINTO  FLIGHT_DETAILS VALUES (‘2005’, ‘1203’, '5', '29', 'F146', '15', '5', '1330',
'1345', '1600', '15', 'No', '8');

INSERTINTO  FLIGHT_DETAILS VALUES (‘2006’, ‘00120’, '6', '24', 'F124', '25', '7', '1000',
'1025', '1300', '25', 'yes', '0');

SELECT * FROM FLIGHT_DETAILS
```

The SQL***create table*** statement has been used to establish a table of Flight details for relational databases. It is shown that there are a total 13 attributes or columns available in the table with proper primary key and data types. The values in the attributes have been loaded by using the ***INSERTINTO*** statement in the research study. The select statement has been used to display the table in the output section of the software platform.

b) The needful query

```
SELECT Flight_Number FROM FLIGHT_DETAILS

Where DATE_TRUNC('month', CAST(scheduled_departure AS DATE)) AS month,

AVG(CASE WHEN departure_delay> 15 THEN 1

        ELSE 0 END) AS perc_delayed
```

WHERE departure_delay IS NOT NULL

GROUP BY 1,2;

Q2:Decentralization and justification

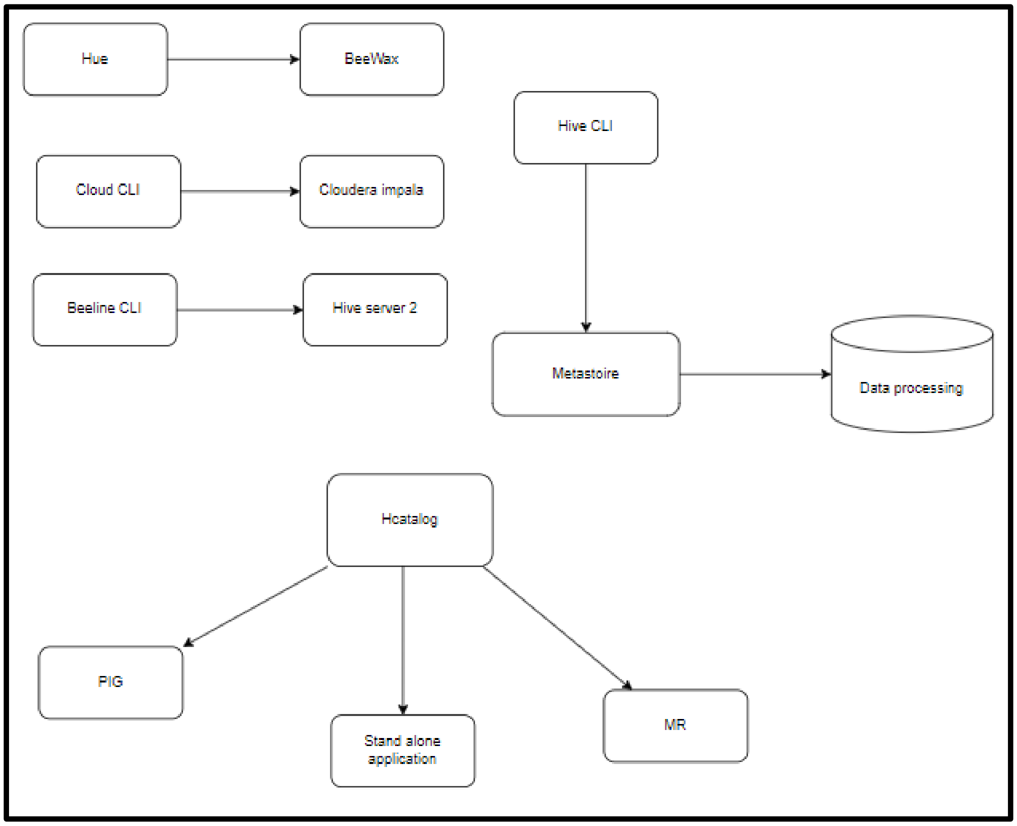


Figure 10: Decentralized diagram of data processing

(Source: Achieved from draw.io)

The above picture shows a decentralized architect diagram of data processing for block chain-based cloud solutions. Map reducing is a kind of programming model to process large volume of data. The data is extremely large, and it is not possible to process the data in centralized way. Therefore, the decentralized technology has been used to process the big data to improve strength and accessibility of the data (Li*et al*. 2019). The data can be more accurate and usable for every user with the help of decentralized data processing systems. The main objective of decentralization is to provide physical location of the data and to develop growth of the database.

Part D: Report

Introduction

NoSQL is a method of specialized database management systems that can modify a wide range of the information which standards, such as crucial value, composition, column, and graphic designs. An exclusive non-relational, disseminated, accommodating, and flawlessly scalable database designated as NoSQL-specific of database will normally be all of these things. A distributed archive is what a block chain essentially is. It can preserve truths such as who is in possession of a certain plot of land or how a commitment is made. The technology can be used to keep a record of requests that cannot be modified and to promote the investment while disguising certain facts.

Blockchain technology

The sectors of banks are looking for delivering best quality services to their customers by applying block chain technology (Vo*et al*. 2018). The unknown parties can be able to deal with an agreement regarding databases with the help of a block chain system for the banks. This system can provide financial benefits like payments by creating decentralization techniques. This task can be helpful for the banks to take attention on other useful activities besides calculating transactions made by payments.

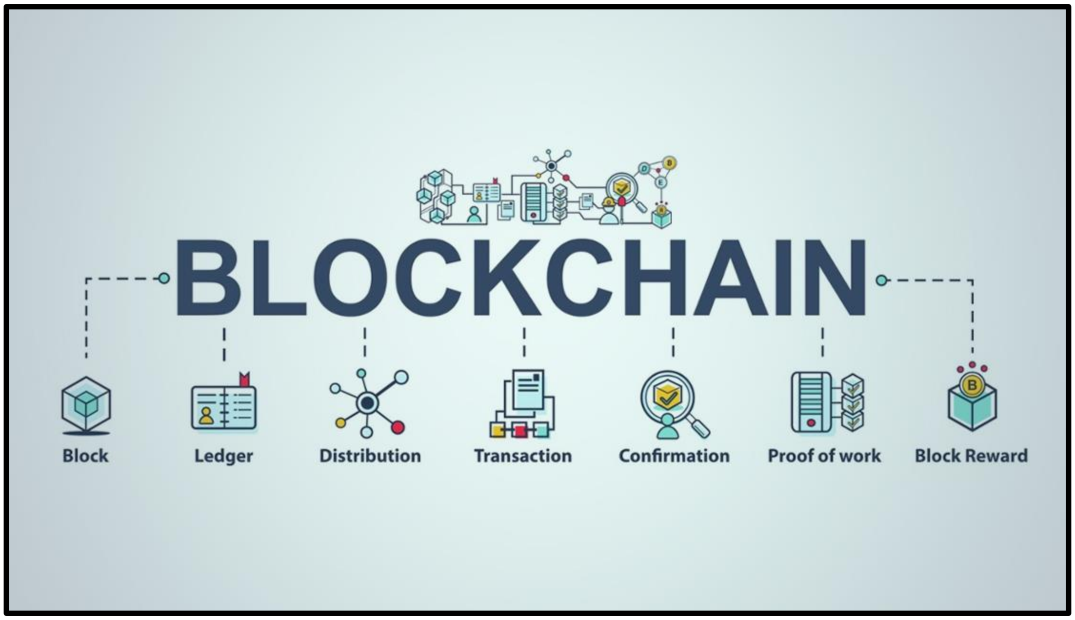


Figure 11: Block chain technology of banking sector

(Source: Kaur*et al.* 2018)

It is noted that by eradicating the need for administrators in the lending and credit industries as well, blockchain has transformed the financial sector. It has lowered interest rates and made financing money more secure and by replacing the paper-intensive process with blockchain, trade finance has altered. Throughout, it has improved trade parties' confidence, security, and transparency.

The banking sector has huge amounts of transactions regularly and relational databases are not suitable in that case. This is the biggest challenge of the banks and so they have finalized to use NoSQL databases rather than relational databases. The limitations of utilizing the NoSQL databases are relatively lower than the rational databases. Blockchain has an impact on clearing and settlement systems, as distributed ledgers can deliver more real-time payments between financial institutions while lowering operational costs. With the advent of initial coin offerings, fundraising has changed(Morais*et al*. 2019). Unbundling access to finance from financing economic services and businesses is possible with a new financing model.

Strength and limitation of relational database

There are so many key points of creating relational databases and these are

- ***Simplicity***
- ***Accessibility***
- ***High accuracy of the data***
- ***Flexibility***
- ***Top level security***
- ***Normalization***
- ***Data integrity***

The banking sector can achieve high quality of services by using relational databases with help of SQL programming. The ability to quickly divide data into several groupings and store them effectively is one of the main advantages of utilizing a relational database. You can use searches and filters to get more information about this arrangement. Without making any changes to the current system, the new database can be augmented to contain any set of data ranging into different categories.

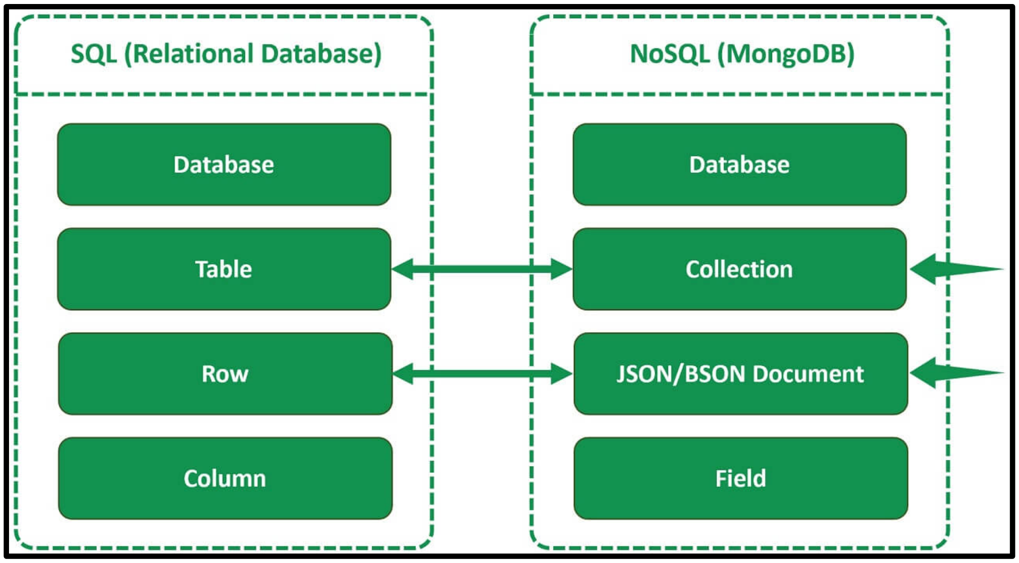


Figure 12: Retinol vs NoSQL database

(Source: Banane and Belangour2020)

The limitations of the relational databases are

- ***Physical storage issue***
- ***Deficiency of scalability***
- ***Maintenance issue***
- ***Toughness in making structure***
- ***Low level olf performance in time***

Strength and limitation of NoSQL databases

The benefits of using NoSQL databases are

- ***Elastic scalability***
- ***High level of flexibility***
- ***Open source database***
- ***Great performance***
- ***Data upgradation***

The disadvantages of the NoSQL databases are considered

- ***Less backup***
- ***Low level of regularity***
- ***Steadiness***

Conclusion

Blockchain technology offers a wide range of advantages. Financial institutions and lenders are now able to provide clients greater service and more security thanks to these advantages. Many financial companies have been able to better their operations and up their game in the banking sector thanks to blockchain technology and banking software solutions. It is concluded that NoSQL databases are more appropriate than relational databases in the banking usage circumstances.

References

Banane, M. and Belangour, A., 2020. A new system for massive RDF data management using Big Data query languages Pig, Hive, and Spark. *International Journal of Computing and Digital Systems*, *9*(2), pp.259-270.

Diène, B., Rodrigues, J.J., Diallo, O., Ndoye, E.H.M. and Korotaev, V.V., 2020. Data management techniques for Internet of Things. *Mechanical Systems and Signal Processing*, *138*, p.106564.

Li, X., Huang, X., Li, C., Yu, R. and Shu, L., 2019. EdgeCare: Leveraging edge computing for collaborative data management in mobile healthcare systems. *IEEE Access*, *7*, pp.22011-22025.

Morais, R., Silva, N., Mendes, J., Adão, T., Pádua, L., López-Riquelme, J.A., Pavón-Pulido, N., Sousa, J.J. and Peres, E., 2019. Mysense: A comprehensive data management environment to improve precision agriculture practices. *Computers and Electronics in Agriculture*, *162*, pp.882-894.

Vo, H.T., Kundu, A. and Mohania, M.K., 2018, March. Research Directions in Blockchain Data Management and Analytics. In *EDBT* (pp. 445-448).