

CSC540 Database Management Systems

PROJECT REPORT 2

WolfPubDB Database Management System

By

Sneha Kumar, Shubham Dilip Pampattiwar, Sreeraksha Mavinhally Sreekantha
Team X

Assumptions:

1. Distributor's phone number will be a single value attribute
2. Contact person per distributor will be a single value attribute
3. The Price in Order table covers all copies.
4. PUB_NO is unique to a particular book edition or periodic publication issue.
5. Every edition of a book has a different ISBN.
6. If staff not an editor - he is an author.
7. Article authors are considered as Journalists.
8. The Date attribute in the Payment table gives the salary claimed date by the addressee.
9. Shipping cost is considered as expense by the publication house and not as income(revenue).

1. Global Relational Database Schema

Publication(PID, Topic, Title, Pub_No)

PID -> PID, Topic, Title, Pub_No

PID or the Publication ID uniquely identifies all other attributes - Title, Topic and Pub_No for a publication.

Pub_No -> Pub_No, PID, Topic, Title

Pub_No also uniquely identifies the combination of all other three attributes - PID, Topic and Title.

Since LHS for both the dependencies is a super key, the relation is in BCNF and therefore in 3NF.

No other dependencies will hold. For example: Attributes like topic/title can be the same for any 2 publications and thus will not functionally determine any other attributes.

Book(PID, Publication Date, ISBN, Edition)

PID -> PID, Publication Date, ISBN, Edition

Since Books is a subclass and Publication is the superclass, PID or the Publication ID uniquely identifies a book. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

No other dependencies will hold. Publication date/edition can be the same for any number books. Therefore we cannot have these attributes determining other attributes (For example 2 books can be published on the same day or can have the same edition).

Periodic publication(PID, Type, Issue date, periodicity)

PID -> PID, Type, Issue date, periodicity

Since periodic publication is a subclass and Publication is the superclass, PID or the Publication ID uniquely identifies a periodic publication. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

No other dependencies will hold. 2 or more periodic publications can be of the same type or can have the same issue date or periodicity. Therefore, these cannot uniquely identify any other attributes. The only attribute that can functionally determine all other attributes of a periodic publication is PID.

Chapter(PID, ChapterID, Title)

PID, ChapterID -> PID, ChapterID, Title

Every book has a unique PID and every chapter in a book will have a unique combination of PID and Chapter ID. The combination of these two will determine the chapter title. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

No other dependencies will hold. One publication can have many chapters and these values can be the same for other publications. Therefore, only a combination of these two can determine the title of a publication. Any two chapters can have the same title, this cannot determine any other attribute values.

Article(PID, ArticleID, text)

PID, ArticleID -> PID, ArticleID, text

Every periodic publication has a unique PID and every article in a periodic publication will have a unique combination of PID and Article ID. The combination of these two will determine the article text. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

This is similar to Chapter, where text can be the same for two articles but cannot determine other values. Therefore, no other dependencies will hold.

Orders(OrderID, Deadline date, Placement date, Shipping cost, price)

OrderID -> Deadline date, Placement date, Shipping cost, price

Every order will have a unique order ID and this determines all the other attributes associated with an order. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

Deadline date and Placement date can be the same for two or more orders. And orders can also have the same shipping cost and price. No two combinations of attributes can functionally determine other values. Therefore, no other dependencies will hold.

Distributor(DistributorID, Name, type, Phone, Balance, Contact Person, Street Address, City)

DistributorID -> DistributorID, Name, type, Phone, Balance, Contact Person, Street Address, City

Every distributor has a unique distributor ID. This ID determines all other attributes associated with a distributor, making it the super key. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

Name/type cannot determine other attributes of a distributor, 2 distributors can have the same name/type. Phone/Contact Person/Street/City - can also be the same for 2 or more distributors and cannot uniquely identify one particular distributor. Balance for 2/more distributors can be the same. Therefore, no other dependencies will hold.

Staff(StaffID, Name, Type)

StaffID -> StaffID, Name, Type

Every Publication House Staff has a unique staff ID. Staff ID determines the name of the staff and the type - if they are staff or invited(authors/editors). Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

Name -> does not hold, because two/more staffs can have the same name.

Type cannot uniquely identify a staff and any combination of the attributes will not uniquely identify staff. Therefore, no other dependencies will hold.

Editor(StaffID)

StaffID -> StaffID

Since it has only one attribute, it is in 3NF.

Author(StaffID)

StaffID -> StaffID

Since it has only one attribute, it is in 3NF.

Writes_Article(StaffID, PID, ArticleID)

StaffID, PID, ArticleID -> StaffID, PID, ArticleID

Writes_Article associates Authors and articles for a periodic publication. Since Articles is a weak entity set, periodic publication ID(PID) is needed to represent an article - Article is a unique combination of Article ID and PID. And A staff can write articles for any number of periodic publications. Hence, a combination of all three attributes will be unique.

Thus, Writes_Article has all three attributes as keys making LHS the superkey. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF. No attribute alone can identify all other attributes.

Writes_Book(StaffID, PID)

StaffID, PID -> StaffID, PID

This relation has only two attributes, hence it is in BCNF and therefore in 3NF.

ConsistsOf(PID, OrderID, number of copies)

PID, OrderID -> PID, OrderID, number of copies

ConsistsOf is a relationship which associates Publication and orders. Every order has a unique order ID and an associated PID which determines the number of copies ordered. Therefore, no other dependencies will hold (Two/more orders can have the same value for the number of copies attributes).

Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF.

Places(OrderID, DistributorID)

OrderID, DistributorID -> OrderID, DistributorID

This relation has only two attributes, hence it is in BCNF and therefore in 3NF.

Edits(StaffID, PID)

StaffID, PID -> StaffID, PID

This relation has only two attributes, hence it is in BCNF and therefore in 3NF.

Payment(StaffID, Date, Salary)

StaffID, Date -> StaffID, Date, Salary

Payment is a weak entity set. Every staff's payment is a unique combination of Staff ID and the date on which they receive their Salary. These two attributes determine the amount paid to the staff. Since LHS is super key, the functional dependency is in BCNF and therefore in 3NF. Staff ID alone cannot determine other attributes. Therefore, no other dependencies will hold.

2. Design for Global Schema

Design decision for global schema:

- ☐ All entity sets are converted to relations - Publication, Staff, Orders, Distributors. All of these entities have respective IDs as primary key.
- ☐ All subclasses are also converted to relations to avoid redundancy - Books, Periodic publication, editor, author.
- ☐ We have converted all weak entity sets to relations which are represented with the help of supporting entity set - Chapters, Articles Payments.
- ☐ All relationships in the E/R diagram are converted to relations - consistsOf, places, Writes_Article and Writes_Book. Their attributes are the keys of the entities they are associated with. Weak relationships are not converted to relations.

- Publication(PID, Topic, Title, Pub_No)
 - ☐ PID is the primary key
 - ☐ Topic, Title and Pub_No are not allowed to be null
- Books(PID, Publication Date, ISBN, Edition)
 - ☐ PID is the primary key
 - ☐ Publication_date, ISBN, Edition are not allowed to be null
- PeriodicPublication(PID, Type, Issue_date, Periodicity)
 - ☐ PID is the primary key
 - ☐ Type, Issue_date, Periodicity are not allowed to be null
- Chapter(PID, ChapterID, Title)
 - ☐ PID and ChapterID forms the primary key
 - ☐ Title is not allowed to be null
- Articles(PID, ArticleID, text)
 - ☐ PID and ArticleID forms the primary key
 - ☐ Text is not allowed to be null
- Orders(OrderID, Deadline date, Placement date, Shipping cost, price)
 - ☐ OrderID is the primary key

- ☐ Placement date, Shipping cost and price not allowed to be null
 - ☐ Deadline Date is allowed to be null if the deadline date for the order is not yet specified.
- Distributor(DistributorID, Name, type, Phone, Balance, Contact Person, Street Address, City)
 - ☐ DistributerID is the primary key
 - ☐ Name, type, Phone, Balance, Contact Person, Street Address and City are not allowed to be null
- Staff(StaffID, Name, Type)
 - ☐ StaffID is the primary key
 - ☐ Name, Type are not allowed to be null
- Editor(StaffID)
 - ☐ StaffID is the primary key
- Author(StaffID)
 - ☐ StaffID is the primary key
- Writes_Articles(StaffID, PID, ArticleID)
 - ☐ StaffID, PID and ArticleID forms the primary key
- Writes_Books(StaffID, PID)
 - ☐ StaffID and PID forms the primary key
- ConsistsOf(PID, OrderID, number of copies)
 - ☐ PID and OrderID forms the primary key
- Places(OrderID, DistributorID)
 - ☐ OrderID and DistributorID forms the primary key
- Edits(StaffID, PID)
 - ☐ StaffID and PID forms the primary key
- Payment(StaffID, Date, Salary)
 - ☐ StaffID and Date forms the primary key
 - ☐ Salary is not allowed to be null


```
MariaDB [skumar32]> show tables;
```

Tables_in_skumar32	
ARTICLE	
AUTHOR	
BOOK	
CHAPTER	
CONSISTSO	
DISTRIBUTOR	
EDITOR	
EDITS	
ORDERS	
PAYMENT	
PERIODICPUBLICATION	
PLACES	
PUBLICATION	
STAFF	
WRITES_ARTICLE	
WRITES_BOOK	

16 rows in set (0.00 sec)

3. Basic Relations

1. Publication(PID, Topic, Title, Pub_No)

```
CREATE TABLE PUBLICATION(  
  PID INT,  
  TOPIC INT,  
  TITLE VARCHAR(30) NOT NULL,  
  PUB_NO CHAR(5),  
  PRIMARY KEY (PID)  
);
```

Query OK, 0 rows affected (0.01 sec)

2. Book(PID, Publication Date, ISBN, Edition)

```
CREATE TABLE BOOK(  
  PID INT,  
  PUBLICATIONDATE DATE,  
  ISBN VARCHAR(30) NOT NULL UNIQUE,  
  EDITION INT, PRIMARY KEY (PID),  
  FOREIGN KEY ( PID ) REFERENCES PUBLICATION(PID)  
);
```

Query OK, 0 rows affected (0.01 sec)

3. Periodic publication(PID, Type, Issue date, periodicity)

```
CREATE TABLE PERIODICPUBLICATION(  
  PID INT,  
  TYPE VARCHAR(20) NOT NULL,  
  ISSUEDATE DATE NOT NULL,
```

```
PERIODICITY VARCHAR(30) NOT NULL,  
PRIMARY KEY (PID),  
FOREIGN KEY ( PID ) REFERENCES PUBLICATION(PID)  
);  
Query OK, 0 rows affected (0.00 sec)
```

4. Chapter(PID, ChapterID, Title)

```
CREATE TABLE CHAPTER(  
PID INT,  
ChapterID INT,  
TITLE VARCHAR(30) NOT NULL,  
PRIMARY KEY ( ChapterID),  
FOREIGN KEY ( PID ) REFERENCES PERIODICPUBLICATION(PID)  
ON UPDATE CASCADE ON DELETE CASCADE );  
Query OK, 0 rows affected (0.01 sec)
```

5. Article(PID, ArticleID, text)

```
CREATE TABLE ARTICLE(  
PID INT,  
ArticleID INT,  
Text VARCHAR(150) NOT NULL,  
PRIMARY KEY(ArticleID, PID),  
FOREIGN KEY(PID) REFERENCES BOOK(PID) ON UPDATE  
CASCADE ON DELETE CASCADE );  
Query OK, 0 rows affected (0.01 sec)
```

6. Orders(OrderID, Deadline date, Placement date, Shipping cost, price)

```
CREATE TABLE ORDERS(  
  OrderID INT,  
  Deadline_Date DATE,  
  Placement_Date DATE NOT NULL,  
  Shipping_Cost FLOAT NOT NULL,  
  Price FLOAT NOT NULL,  
  PRIMARY KEY(OrderID));  
  
Query OK, 0 rows affected (0.02 sec)
```

7. Distributor(DistributorID, Name, type, Phone, Balance, Contact Person, Street Address, City)

```
CREATE TABLE DISTRIBUTOR(DID INT,  
  -> NAME VARCHAR(30) NOT NULL,  
  -> TYPE VARCHAR(150) NOT NULL,  
  -> PHONE INT(10) NOT NULL,  
  -> BALANCE FLOAT NOT NULL,  
  -> CONTACT_PERSON VARCHAR(30) NOT NULL,  
  -> STREET VARCHAR(100) NOT NULL,  
  -> CITY VARCHAR(30) NOT NULL,  
  -> PRIMARY KEY(DID));  
  
Query OK, 0 rows affected (0.02 sec)
```

8. Staff(StaffID, Name, Type)

```
CREATE TABLE STAFF(  
  STAFFID INT,
```

NAME VARCHAR(30) NOT NULL,
TYPE VARCHAR(150) NOT NULL,
PRIMARY KEY(STAFFID));
Query OK, 0 rows affected (0.00 sec)

9. Editor(StaffID)

CREATE TABLE EDITOR(
STAFFID INT,
PRIMARY KEY(STAFFID),
FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
Query OK, 0 rows affected (0.00 sec)

10. Author(StaffID)

CREATE TABLE AUTHOR(
STAFFID INT,
PRIMARY KEY(STAFFID),
FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
Query OK, 0 rows affected (0.00 sec)

11. Writes_Article(StaffID, PID, ArticleID)

CREATE TABLE WRITES_ARTICLE(
-> STAFFID INT,
-> PID INT,
-> ARTICLEID INT,
-> PRIMARY KEY(STAFFID,PID,ARTICLEID),
-> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),
-> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID),

-> FOREIGN KEY(ARTICLEID) REFERENCES ARTICLE(ArticleID)
-> ON DELETE CASCADE ON UPDATE CASCADE
->);

Query OK, 0 rows affected (0.01 sec)

12. Writes_Book(StaffID, PID)

```
CREATE TABLE WRITES_BOOK(  
  -> STAFFID INT,  
  -> PID INT,  
  -> PRIMARY KEY(STAFFID,PID),  
  -> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),  
  -> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID)  
  -> );
```

Query OK, 0 rows affected (0.02 sec)

13. ConsistsOf(PID, OrderID, number of copies)

```
CREATE TABLE CONSISTSOOF(  
  ORDERID INT,  
  PID INT,  
  NO_OF_COPIES INT,  
  PRIMARY KEY(PID,ORDERID),  
  FOREIGN KEY(PID) REFERENCES PUBLICATION(PID),  
  FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID));
```

Query OK, 0 rows affected (0.01 sec)

14. Places(OrderID, DistributorID)

```
CREATE TABLE PLACES(  
  -> ORDERID INT UNIQUE,  
  -> DID INT,  
  -> PRIMARY KEY(ORDERID , DID),  
  -> FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID),  
  -> FOREIGN KEY(DID) REFERENCES DISTRIBUTOR(DID));
```

Query OK, 0 rows affected (0.02 sec)

15. Edits(StaffID, PID)

```
CREATE TABLE EDITS(  
  -> STAFFID INT,  
  -> PID INT,  
  -> PRIMARY KEY(STAFFID,PID),  
  -> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),  
  -> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID));
```

Query OK, 0 rows affected (0.02 sec)

16. Payment(StaffID, Date, Salary)

```
CREATE TABLE PAYMENT(  
  STAFFID INT,  
  PAYMENT_DATE DATE,  
  SALARY FLOAT NOT NULL,  
  PRIMARY KEY(STAFFID,PAYMENT_DATE),  
  FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
```

Query OK, 0 rows affected (0.01 sec)

CREATE QUERIES:

```
MariaDB [skumar32]>CREATE TABLE PUBLICATION(PID INT, TOPIC VARCHAR(30) NOT NULL, TITLE VARCHAR(30) NOT NULL, PUB_NO CHAR(5), PRIMARY KEY (PID));
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [skumar32]> CREATE TABLE BOOK( PID INT, PUBLICATIONDATE DATE, ISBN VARCHAR(30) NOT NULL UNIQUE, EDITION INT, PRIMARY KEY (PID), FOREIGN KEY ( PID ) REFERENCES PUBLICATION(PID) );
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [skumar32]> CREATE TABLE PERIODICPUBLICATION( PID INT, TYPE VARCHAR(20) NOT NULL, ISSUEDATE DATE NOT NULL, PERIODICITY VARCHAR(30) NOT NULL, PRIMARY KEY (PID), FOREIGN KEY ( PID ) REFERENCES PUBLICATION(PID) );
```

Query OK, 0 rows affected (0.00 sec)

```
MariaDB [skumar32]> CREATE TABLE CHAPTER( PID INT, ChapterID INT, TITLE VARCHAR(30) NOT NULL, PRIMARY KEY ( PID, ChapterID), FOREIGN KEY ( PID ) REFERENCES PERIODICPUBLICATION (PID) ON UPDATE CASCADE ON DELETE CASCADE );
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [skumar32]> CREATE TABLE ARTICLE( PID INT, ArticleID INT, Text VARCHAR(150) NOT NULL, PRIMARY KEY(PID, ArticleID), FOREIGN KEY(PID) REFERENCES BOOK(PID) ON UPDATE CASCADE ON DELETE CASCADE );
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [skumar32]> CREATE TABLE ORDERS(OrderID INT, Deadline Date DATE, Placement Date DATE NOT NULL, Shipping Cost FLOAT NOT NULL, Price FLOAT NOT NULL, PRIMARY KEY(OrderID));
```

Query OK, 0 rows affected (0.02 sec)

```
MariaDB [skumar32]> CREATE TABLE DISTRIBUTOR( DISTRIBUTORID INT, NAME VARCHAR(30) NOT NULL, TYPE VARCHAR(150) NOT NULL, PHONE INT(10) NOT NULL, BALANCE FLOAT NOT NULL, CONTACT_PERSON VARCHAR(30) NOT NULL, STREET VARCHAR(100) NOT NULL, CITY VARCHAR(30) NOT NULL );
```

Query OK, 0 rows affected (0.01 sec)

```
MariaDB [skumar32]> CREATE TABLE STAFF( STAFFID INT, NAME VARCHAR(30)
NOT NULL, TYPE VARCHAR(150) NOT NULL, PRIMARY KEY(STAFFID) );
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [skumar32]> CREATE TABLE AUTHOR( STAFFID INT, PRIMARY
KEY(STAFFID), FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [skumar32]> CREATE TABLE EDITOR(STAFFID INT, PRIMARY
KEY(STAFFID), FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [skumar32]> CREATE TABLE DISTRIBUTOR(DID INT,
-> NAME VARCHAR(30) NOT NULL,
-> TYPE VARCHAR(150) NOT NULL,
-> PHONE INT(10) NOT NULL,
-> BALANCE FLOAT NOT NULL,
-> CONTACT_PERSON VARCHAR(30) NOT NULL,
-> STREET VARCHAR(100) NOT NULL,
-> CITY VARCHAR(30) NOT NULL,
-> PRIMARY KEY(DID));
Query OK, 0 rows affected (0.02 sec)
```

```
MariaDB [skumar32]> CREATE TABLE WRITES_ARTICLE(
-> STAFFID INT,
-> PID INT,
-> ARTICLEID INT,
-> PRIMARY KEY(STAFFID,PID,ARTICLEID),
-> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),
-> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID),
-> FOREIGN KEY(ARTICLEID) REFERENCES ARTICLE(ARTICLEID)
-> );
Query OK, 0 rows affected (0.01 sec)
```

```
|
MariaDB [skumar32]> CREATE TABLE WRITES_BOOK(
    -> STAFFID INT,
    -> PID INT,
    -> PRIMARY KEY(STAFFID,PID),
    -> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),
    -> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID)
    -> );
```

Query OK, 0 rows affected (0.02 sec)

```
MariaDB [skumar32]> CREATE TABLE PLACES(
    -> ORDERID INT UNIQUE,
    -> DID INT,
    -> PRIMARY KEY(ORDERID , DID),
    -> FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID),
    -> FOREIGN KEY(DID) REFERENCES DISTRIBUTOR(DID));
```

Query OK, 0 rows affected (0.02 sec)

```
MariaDB [skumar32]> CREATE TABLE EDITS(
    -> STAFFID INT,
    -> PID INT,
    -> PRIMARY KEY(STAFFID,PID),
    -> FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID),
    -> FOREIGN KEY(PID) REFERENCES PUBLICATION(PID));
```

Query OK, 0 rows affected (0.02 sec)

```
MariaDB [skumar32]> CREATE TABLE PAYMENT( STAFFID INT, PAYMENT_DATE
DATE, SALARY FLOAT NOT NULL, PRIMARY KEY(STAFFID,PAYMENT_DATE),
FOREIGN KEY(STAFFID) REFERENCES STAFF(STAFFID));
```

Query OK, 0 rows affected (0.01 sec)

SELECT QUERIES:

```
MariaDB [skumar32]> select * from ARTICLE;
```

PID	ArticleID	Text
2	1	Curabitur pretium tincidunt lacus. Nulla gravida orci a odio. Nullam varius, turpis et commodo pharetra, est eros bibendum elit, nec luctus magna feli
5	1	Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
6	1	Now is the winter of our discontent
Made glorious summer by this sun of York		
7	1	And all the clouds that lour'd upon our house In the deep bosom of the ocean buried
2	2	Our stern alarums changed to merry meetings,
Our dreadful marches to delightful measures.		
5	2	To fright the souls of fearful adversaries,
He capers nimbly in a lady's chamber To the lascivious pleasing of a lute.		
6	2	But I, that am not shaped for sportive tricks,
Nor made to court an amorous looking-glass		

7 rows in set (0.00 sec)

```
MariaDB [skumar32]> select * from AUTHOR;
```

STAFFID
1
3
4
7
8

5 rows in set (0.00 sec)

```
MariaDB [skumar32]> select * from BOOK;
```

PID	PUBLICATIONDATE	ISBN	EDITION
1	2009-03-21	456-3-16-148410-0	2
3	2009-02-19	123-3-86-147810-0	1
4	2009-01-11	623-3-96-784880-1	3
8	2009-03-25	456-3-16-148560-01	3

4 rows in set (0.00 sec)

```
MariaDB [skumar32]> select * from CHAPTER;
```

PID	ChapterID	TITLE
1	1	Introduction
1	2	Introduction
3	1	Prologue
3	2	Apple Barrel
4	1	The Sunset
4	2	Sound off

6 rows in set (0.00 sec)

MariaDB [skumar32]> select * from CONSISTSOF;

ORDERID	PID	NO_OF_COPIES
6	2	1
7	2	1
1	3	7
2	4	1
3	5	3
4	6	2
5	7	4

7 rows in set (0.01 sec)

MariaDB [skumar32]> select * from DISTRIBUTOR;

DID	NAME	TYPE	PHONE	BALANCE	CONTACT_PERSON	STREET	CITY
1	Pearson	Retail	2147483647	700	Ed Woodward	Richmond Avenue	Portland
2	Baker & Taylor	Book Store	2147483647	1050	Matthew Perry	Brent Street	Durham
3	Hunt	Library	2147483647	5125.5	Sneha Kumar	Kaplan Drive	Orlando
4	Barnes and Nobles	Book Store	2147483647	0	Kin Hyun Jung	Richmond Avenue	Austin
5	Penguin	wholesale	2147483647	50	Steven Lim	Richmond Avenue	Huoston
6	D. H. Hill	library	2147483647	650	Andrew Ng	Richmond Avenue	Denver

6 rows in set (0.01 sec)

MariaDB [skumar32]> select * from EDITOR;

STAFFID
2
5
6
9
10

5 rows in set (0.01 sec)

MariaDB [skumar32]> select * from EDITS;

STAFFID	PID
2	4
5	3
6	2
9	5

4 rows in set (0.01 sec)

MariaDB [skumar32]> select * from ORDERS;

OrderID	Deadline_Date	Placement_Date	Shipping_Cost	Price
1	2010-11-11	2010-07-07	50	100
2	2010-12-10	2010-12-06	30	150
3	2010-05-03	2010-05-01	20	170
4	0000-00-00	2010-07-05	50	200
5	2011-01-11	2011-01-09	40	350
6	2011-02-02	2011-02-01	80	400
7	2011-03-19	2011-03-12	10	178
8	2011-04-15	2011-04-04	30	760

8 rows in set (0.00 sec)

MariaDB [skumar32]> select * from PAYMENT;

STAFFID	PAYMENT_DATE	SALARY
1	2020-05-07	115000
1	2020-06-07	10000
1	2020-07-07	5000
2	2020-05-06	125000
3	2020-05-08	125060
4	2020-05-09	122060
5	2020-05-02	102060
6	2020-05-12	92060
7	2020-05-02	95060
8	2020-05-01	95020
9	2020-05-06	90270
10	2020-05-04	90770

12 rows in set (0.01 sec)

MariaDB [skumar32]> select * from PERIODICPUBLICATION;

PID	TYPE	ISSUEDATE	PERIODICITY
2	Magazine	2009-02-22	Monthly
5	Magazine	2009-03-22	Monthly
6	Magazine	2009-03-12	Weekly
7	Journal	2009-03-10	Monthly

4 rows in set (0.00 sec)

MariaDB [skumar32]> select * from PLACES;

ORDERID	DID
1	3
2	3
3	4
4	5
5	6
6	1
7	2

7 rows in set (0.01 sec)

[MariaDB [skumar32]> select * from PUBLICATION;

PID	TOPIC	TITLE	PUB_NO
1	Fiction	The Digital Fortress	B1
2	Current Affairs	The Digit Magazine	PP1
3	Mystery	Gone girl	B2
4	Sci-Fi	Martian	B3
5	Fashion	Vogue	PP2
6	News	Time	PP3
7	DBMS	Journal of Database Management	PP4
8	Fiction	The Digital Fortress	B1-2

8 rows in set (0.00 sec)

[MariaDB [skumar32]> select * from STAFF;

STAFFID	NAME	TYPE
1	Jack Ryan	Full-Time
2	Robert Kaufmann	Full-Time
3	Brad Pitt	Full-Time
4	Stephen king	Invited
5	James Patterson	Full-Time
6	Dean Koontz	Invited
7	J. K. Rowling	Full-Time
8	John Grisham	Invited
9	Nora Roberts	Full-Time
10	Dr. Seuss	Invited

10 rows in set (0.01 sec)

[MariaDB [skumar32]> select * from WRITES_ARTICLE;

STAFFID	PID	ARTICLEID
4	2	1
4	2	2
7	5	1
7	5	2
8	6	1
8	6	2
8	7	1

7 rows in set (0.00 sec)

[MariaDB [skumar32]> select * from WRITES_BOOK;

STAFFID	PID
1	3
3	1
4	4

3 rows in set (0.00 sec)

4. SQL queries

4.1

Information processing

Maintaining publication, staff, distributor accounts and order details

- **Enter Publication Information:**

- If Publication type is a 'Book':

```
MariaDB [skumar32]> INSERT INTO PUBLICATION VALUES(1,  
'Fiction', 'The Digital Fortress', 'B1');
```

Query OK, 1 row affected (0.00 sec)

```
MariaDB [skumar32]> INSERT INTO BOOK VALUES(1, '2020-03-21',  
'978-3-16-148410-0', 2);
```

Query OK, 1 row affected, 1 warning (0.01 sec)

- If Publication type is a 'Periodic Publication':

```
MariaDB [skumar32]> INSERT INTO PUBLICATION VALUES(2,  
'Current Affairs', 'The Digit Magazine');
```

Query OK, 1 row affected (0.00 sec)

```
MariaDB [skumar32]> INSERT INTO PERIODICPUBLICATION  
VALUES(2, 'Magazine', '2020-03-22', 'Weekly');
```

Query OK, 1 row affected, 1 warning (0.00 sec)

- **Update Publication(Book/Periodic Publication) information:**

```
UPDATE BOOK SET ISBN = '456-3-16-148410-0' WHERE PID = 1;
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
UPDATE PERIODICPUBLICATION SET PERIODICITY = 'Monthly' WHERE  
PID = 2;
```

Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Publication information:**

DELETE FROM PUBLICATION WHERE PID = 11;

Query OK, 1 row affected (0.00 sec)

- **Enter Staff information:**

- If the Staff is an Author:

INSERT INTO STAFF VALUES(1, 'Jack Ryan', 'Full-Time');

Query OK, 1 row affected (0.00 sec)

INSERT INTO AUTHOR VALUES(1);

Query OK, 1 row affected (0.00 sec)

- If the Staff is an Editor:

INSERT INTO STAFF VALUES(2, 'Robert Kaufmann', 'Invited');

Query OK, 1 row affected (0.00 sec)

INSERT INTO EDITOR VALUES(2);

Query OK, 1 row affected (0.00 sec)

- **Update Staff information:**

UPDATE STAFF SET TYPE = 'Full-Time' WHERE STAFFID=2;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Staff Information:**

DELETE FROM STAFF WHERE STAFFID=1;

Query OK, 1 row affected (0.00 sec)

- **Enter Distributor Information:**

INSERT INTO DISTRIBUTOR VALUES(1, 'Pearson', 'Retail', 9879554765,
700.00, 'Kevin Friend', 'Richmond Avenue', 'Portland');

Query OK, 1 row affected, 1 warning (0.00 sec)

- **Update Distributor Information:**

UPDATE DISTRIBUTOR SET CONTACT_PERSON = 'Ed Woodward'
WHERE NAME = 'Pearson';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Distributor Information:**

DELETE FROM DISTRIBUTOR WHERE NAME = 'Pearson';

Query OK, 1 row affected (0.00 sec)

- **Enter Order Information:**

INSERT INTO ORDERS VALUES(1, '2010-10-10', '2010-07-07', 50, 100);

Query OK, 1 row affected (0.00 sec)

- **Update Order Information:**

```
UPDATE ORDERS SET DEADLINE_DATE = '2010-11-11' WHERE  
ORDERID = 1;
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Order Information:**

```
DELETE FROM ORDERS WHERE PLACEMENT_DATE = '2010-07-07';
```

Query OK, 1 row affected (0.00 sec)

- **Input orders from distributors, for a book edition or an issue of a publication per distributor, for a certain date.**

```
INSERT INTO ORDERS VALUES(9, '2011-02-18', '2011-02-07', 55, 180);
```

Query OK, 1 row affected (0.00 sec)

```
MariaDB [skumar32]> INSERT INTO PLACES VALUES(9,1);
```

Query OK, 1 row affected (0.00 sec)

```
MariaDB [skumar32]> INSERT INTO CONSISTSOOF VALUES(9,7,2);
```

Query OK, 1 row affected (0.00 sec)

- **Bill distributor for an order**

```
INSERT INTO ORDERS VALUES(10, '2011-03-19', '2011-02-08', 20, 270);
```

Query OK, 1 row affected (0.01 sec)

```
MariaDB [skumar32]> INSERT INTO PLACES VALUES(10,2);
```

Query OK, 1 row affected (0.01 sec)

- **Update balance for Distributor:**

```
UPDATE DISTRIBUTOR SET BALANCE = 50 WHERE NAME = 'Penguin';
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Enter Chapter Information:**

```
INSERT INTO CHAPTER VALUES(1, 1, 'Prologue');
```

Query OK, 1 row affected (0.00 sec)

- **Update Chapter Information:**

```
UPDATE CHAPTER SET TITLE = 'Introduction' WHERE ChapterID = 1;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Chapter Information:**

```
DELETE FROM CHAPTER WHERE ChapterID = 1;
```

Query OK, 3 row affected (0.00 sec)

- **Add Article Information:**

INSERT INTO ARTICLE VALUES(2, 1, 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation');

Query OK, 1 row affected, 1 warning (0.00 sec)

- **Update Article Information:**

UPDATE ARTICLE SET TEXT = 'Curabitur pretium tincidunt lacus. Nulla gravida orci a odio. Nullam varius, turpis et commodo pharetra, est eros bibendum elit, nec luctus magna felis sollicitudin' WHERE ArticleID = 1;

Query OK, 1 row affected, 1 warning (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 1

- **Delete Article Information:**

DELETE FROM ARTICLE WHERE ARTICLEID =1

Query OK, 4 row affected (0.00 sec)

- **Enter Payment Information:**

INSERT INTO PAYMENT VALUES(1, '2020-05-07', 115000.00);

Query OK, 1 row affected (0.00 sec)

- **Update Payment Information:**

UPDATE PAYMENT SET SALARY = 135000 WHERE STAFFID = 3;

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0 Changed: 0 Warnings: 0

- **Delete Payment Information:**

DELETE FROM PAYMENT WHERE SALARY > 95020;
Query OK, 6 row affected (0.00 sec)

Maintaining Orders:

- **Ordes and Publications mapping (CONSISTSOF):**

INSERT INTO CONSISTSOF values(1,3,7);
Query OK, 1 row affected (0.00 sec)

- **Orders and Distributors mapping (PLACES):**

INSERT INTO PLACES values(2, 3);
Query OK, 1 row affected (0.00 sec)

Maintaining Publications:

- **Staff(Editor/Author) and Publications mapping(EDITS):**

INSERT INTO EDITS VALUES(2, 4);
Query OK, 1 row affected (0.00 sec)

RetrieveInformation:

- **Find Publication based on topic:**

- For Books:

SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE
PUBLICATION.TOPIC = "Mystery";

```
MariaDB [skumar32]> SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";
```

PID	TOPIC	TITLE	PUB_NO	PUBLICATIONDATE	ISBN	EDITION
3	Mystery	Gone girl	B2	2009-02-19	123-3-86-147810-0	1

1 row in set (0.01 sec)

- For Periodic Publications:

```
SELECT * FROM PUBLICATION NATURAL JOIN
PERIODICPUBLICATION WHERE PUBLICATION.TOPIC =
<INPUT_TOPIC>;
```

```
MariaDB [skumar32]> SELECT * FROM PUBLICATION NATURAL JOIN PERIODICPUBLICATION WHERE PUBLICATION.TOPIC = "Fashion";
```

PID	TOPIC	TITLE	PUB_NO	TYPE	ISSUEDATE	PERIODICITY
5	Fashion	Vogue	PP2	Magazine	2009-03-22	Monthly

1 row in set (0.00 sec)

- **Find Publication based on Publication date(Publication date/issue date):**

- For Books:

```
SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE
PUBLICATIONDATE = <INPUT_DATE>;
```

```
[MariaDB [skumar32]> SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATIONDATE = '2009-03-21';
```

PID	TOPIC	TITLE	PUB_NO	PUBLICATIONDATE	ISBN	EDITION
1	Fiction	The Digital Fortress	B1	2009-03-21	456-3-16-148410-0	2

1 row in set (0.01 sec)

- For Periodic Publication:

```
SELECT * FROM PUBLICATION NATURAL JOIN
PERIODICPUBLICATION WHERE ISSUE DATE = <INPUT_DATE>;
```

```
MariaDB [skumar32]> SELECT * FROM PUBLICATION NATURAL JOIN PERIODICPUBLICATION WHERE ISSUEDATE = '2009-02-22';
```

PID	TOPIC	TITLE	PUB_NO	TYPE	ISSUEDATE	PERIODICITY
2	Current Affairs	The Digit Magazine	PP1	Magazine	2009-02-22	Monthly

1 row in set (0.01 sec)

- **Find Publication based on Author's Name:**

- For Books:

```
SELECT * FROM WRITES_BOOK W NATURAL JOIN PUBLICATION P
NATURAL JOIN BOOK B WHERE W.STAFFID IN (SELECT A.STAFFID
```

FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME =
<STAFF_NAME>);

SELECT * FROM WRITES_BOOK W NATURAL JOIN PUBLICATION P
NATURAL JOIN BOOK B WHERE W.STAFFID IN (SELECT A.STAFFID
FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Jack
Ryan");

```

MariaDB [skumar32]> SELECT * FROM WRITES_BOOK W NATURAL JOIN PUBLICATION P NATURAL JOIN BOOK B WHERE W.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Jack Ryan");
+-----+-----+-----+-----+-----+-----+-----+-----+
| PID | STAFFID | TOPIC | TITLE | PUB_NO | PUBLICATIONDATE | ISBN | EDITION |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 1 | Mystery | Gone girl | B2 | 2009-02-19 | 123-3-86-147810-0 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

- For Periodic Publication:

SELECT * FROM WRITES_ARTICE WA NATURAL JOIN PUBLICATION P
NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT
A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME
= <STAFF_NAME>);

SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION
P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT
A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME
= "Stephen King");

```

MariaDB [skumar32]> SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
+-----+-----+-----+-----+-----+-----+-----+-----+
| PID | ARTICLEID | STAFFID | TOPIC | TITLE | PUB_NO | Text |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 1 | 4 | Current Affairs | The Digit Magazine | PP1 | Curabitur pretium tincidunt lacus. Nulla gravida orci a odio. Nullam varius, turpis et commodo pharetra, est eros bibendum elit, nec luctus magna feli |
| 2 | 2 | 4 | Current Affairs | The Digit Magazine | PP1 | Our stern alarums changed to merry meetings, |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

Reports:

- **Get the number of copies sold per distributor per month:**

```
SELECT SUM(C.NO_OF_COPIES),SUM(O.PRICE), P.DID,
MONTH(PLACEMENT_DATE), YEAR(PLACEMENT_DATE)
FROM CONSISTSOFC C NATURAL JOIN PLACES P NATURAL JOIN ORDERS O
group by P.DID, MONTH(PLACEMENT_DATE), YEAR(PLACEMENT_DATE);
```

```
MariaDB [skumar32]> SELECT SUM(C.NO_OF_COPIES),SUM(O.PRICE), P.DID, MONTH(PLACEMENT_DATE), YEAR(PLACEMENT_DATE) FROM CONSISTSOFC C NATURAL JOIN PLACES P NATURAL JOIN ORDERS O group by P.DID, MONTH(PLACEMENT_DATE), YEAR(PLACEMENT_DATE);
```

SUM(C.NO_OF_COPIES)	SUM(O.PRICE)	DID	MONTH(PLACEMENT_DATE)	YEAR(PLACEMENT_DATE)
1	400	1	2	2011
1	170	2	3	2011
7	100	3	7	2010
1	150	3	12	2010
3	170	4	5	2010
2	200	5	7	2010
4	350	6	1	2011

7 rows in set (0.01 sec)

- **Get the total Revenue:**

```
SELECT SUM(PRICE) AS REVENUE FROM ORDERS;
```

```
[MariaDB [skumar32]> SELECT SUM(PRICE) AS REVENUE FROM ORDERS;
```

REVENUE
2308

1 row in set (0.00 sec)

- **Get Total Expense:**

```
SELECT (SELECT SUM(SALARY) FROM PAYMENT) + (SELECT
SUM(Shipping_Cost) FROM ORDERS) as TOTAL_EXPENSE;
```

```
[MariaDB [skumar32]> SELECT (SELECT SUM(SALARY) FROM PAYMENT) + (SELECT SUM(Shipping_Cost) FROM ORDERS) as TOTAL_EXPENSE;
```

TOTAL_EXPENSE
1067670

1 row in set (0.00 sec)

- **Get the Total number of Distributors:**

SELECT COUNT(*) AS TOTAL_NO_DISTRIBUTORS FROM DISTRIBUTOR;

```
[MariaDB [skumar32]> SELECT COUNT(*) FROM DISTRIBUTOR;
```

COUNT(*)
6

1 row in set (0.00 sec)

- **Get the Total Revenue generated per city:**

SELECT SUM(O.PRICE) AS TOTAL_REVENUE, D.CITY FROM DISTRIBUTOR D
NATURAL JOIN PLACES P NATURAL JOIN ORDERS O GROUP BY D.CITY;

```
[MariaDB [skumar32]> SELECT SUM(O.PRICE) AS TOTAL_REVENUE, D.CITY FROM DISTRIBUTOR D NATURAL JOIN PLACES P NATURAL JOIN ORDERS O GROUP BY D.CITY;
```

TOTAL_REVENUE	CITY
170	Austin
350	Denver
448	Durham
200	Houston
250	Orlando
580	Portland

6 rows in set (0.00 sec)

- **Get the Total revenue generated per Distributor:**

SELECT SUM(O.PRICE) AS TOTAL_REVENUE,D.DID FROM DISTRIBUTOR D
NATURAL JOIN PLACES P NATURAL JOIN ORDERS O GROUP BY D.DID;

```
[MariaDB [skumar32]> SELECT SUM(O.PRICE) AS TOTAL_REVENUE,D.DID FROM DISTRIBUTOR D NATURAL JOIN PLACES P NATURAL JOIN ORDERS O GROUP BY D.DID;
```

TOTAL_REVENUE	DID
580	1
448	2
250	3
170	4
200	5
350	6

6 rows in set (0.00 sec)

- **Get the Total Revenue generated per Location:**

SELECT D.STREET,D.CITY,SUM(O.Price) AS TOTAL_REVENUE FROM
DISTRIBUTOR D NATURAL JOIN PLACES P NATURAL JOIN ORDERS O
GROUP BY D.STREET,D.CITY;

```
[MariaDB [skumar32]> SELECT D.STREET,D.CITY,SUM(O.Price) AS TOTAL_REVENUE FROM DISTRIBUTOR D NATURAL JOIN PLACES P NATURAL JOIN ORDERS O GROUP BY D.STREET,D.CITY;
```

STREET	CITY	TOTAL_REVENUE
Brent Street	Durham	448
Kaplan Drive	Orlando	250
Richmond Avenue	Austin	170
Richmond Avenue	Denver	350
Richmond Avenue	Houston	200
Richmond Avenue	Portland	580

```
6 rows in set (0.00 sec)
```

Payments:

- **Get the total Staff Payments (Salary):**

```
SELECT SUM(SALARY) FROM PAYMENT P NATURAL JOIN STAFF S WHERE
PAYMENT_DATE BETWEEN <START_DATE> AND <END_DATE>;
```

```
SELECT SUM(SALARY) FROM PAYMENT P WHERE PAYMENT_DATE
BETWEEN '2020-05-02' AND '2020-05-07';
```

```
[MariaDB [skumar32]> SELECT SUM(SALARY) FROM PAYMENT P WHERE PAYMENT_DATE BETWEEN '2020-05-02' AND '2020-05-07';
```

SUM(SALARY)
618160

```
1 row in set (0.00 sec)
```

- **Get the total Staff Payments based on work type:**

- For Editors:

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN EDITOR
WHERE PAYMENT_DATE BETWEEN <START_DATE> AND
<END_DATE>;
```

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN EDITOR
WHERE PAYMENT_DATE BETWEEN '2020-05-01' AND '2020-05-09';
```

```
[MariaDB [skumar32]> SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN EDITOR WHERE PAYMENT_DATE BETWEEN '2020-05-01' AND '2020-05-09';
```

SUM(SALARY)
408100

```
1 row in set (0.01 sec)
```

- For Book Authors:

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR  
NATURAL JOIN WRITES_BOOK WHERE PAYMENT_DATE BETWEEN  
<START_DATE> AND <END_DATE>;
```

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR  
NATURAL JOIN WRITES_BOOK WHERE PAYMENT_DATE BETWEEN  
'2020-05-07' AND '2020-05-08';
```

```
MariaDB [skumar32]> SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR NATURAL JOIN WRITES_BOOK WHERE PAYMENT_DATE BETWEEN '2020-05-07' AND '2020-05-08';  
+-----+  
| SUM(SALARY) |  
+-----+  
|      240060 |  
+-----+  
1 row in set (0.01 sec)
```

- For Article Authors:

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR  
NATURAL JOIN WRITES_ARTICLE WHERE PAYMENT_DATE BETWEEN  
<START_DATE> AND <END_DATE>;
```

```
SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR  
NATURAL JOIN WRITES_ARTICLE WHERE PAYMENT_DATE BETWEEN  
'2020-05-01' AND '2020-05-08';
```

```
MariaDB [skumar32]> SELECT SUM(SALARY) FROM PAYMENT NATURAL JOIN AUTHOR NATURAL JOIN WRITES_ARTICLE WHERE PAYMENT_DATE BETWEEN '2020-05-01' AND '2020-05-08';  
+-----+  
| SUM(SALARY) |  
+-----+  
|      475180 |  
+-----+  
1 row in set (0.00 sec)
```

4.2

Query 1 Find Publication based on Publication date - For Books:

```
SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL  
JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S  
NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
```

1. SQL Query:

```
SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL  
JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S  
NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
```

```
MariaDB [skumar32]> SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
```

PID	ARTICLEID	STAFFID	TOPIC	TITLE	PUB_NO	Text
2	1	4	Current Affairs	The Digit Magazine	PP1	Curabitur pretium tincidunt lacus. Nulla gravida orci a odio. Nullam varius, turpis et commodo pharetra, est eros bibendum elit, nec luctus magna feli
2	2	4	Current Affairs	The Digit Magazine	PP1	Our stern alarums changed to merry meetings, Our dreadful marches to delightful measures.

```
2 rows in set (0.00 sec)
```

2. SQL Query with Explain directive:

```
EXPLAIN SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P  
NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM  
STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
```

```
MariaDB [skumar32]> EXPLAIN SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	WA	index	PRIMARY,PID,ARTICLEID	PRIMARY	12	NULL	7	Using index
1	PRIMARY	A	eq_ref	PRIMARY	PRIMARY	4	skumar32.WA.STAFFID	1	Using index
1	PRIMARY	A	eq_ref	PRIMARY,PID	PRIMARY	8	skumar32.WA.ARTICLEID,skumar32.WA.PID	1	
1	PRIMARY	P	eq_ref	PRIMARY	PRIMARY	4	skumar32.WA.PID	1	
1	PRIMARY	S	eq_ref	PRIMARY	PRIMARY	4	skumar32.WA.STAFFID	1	Using where

```
5 rows in set (0.00 sec)
```

3. Index Creation Query

CREATE INDEX name_INDEX ON STAFF(NAME);

```

MariaDB [skumar32]> CREATE INDEX name_INDEX ON STAFF(NAME);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
MariaDB [skumar32]>

```

4. SQL Query Output post creation of index:

EXPLAIN SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S.NAME = "Stephen King");

```

MariaDB [skumar32]> EXPLAIN SELECT * FROM WRITES_ARTICLE WA NATURAL JOIN PUBLICATION P NATURAL JOIN ARTICLE A WHERE WA.STAFFID IN (SELECT A.STAFFID FROM STAFF S NATURAL JOIN AUTHOR A WHERE S
.NAME = "Stephen King");

```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	S	ref	PRIMARY,name_INDEX	name_INDEX	32	const	1	Using where; Using index
1	PRIMARY	A	eq_ref	PRIMARY	PRIMARY	4	skumar32.S.STAFFID	1	Using index
1	PRIMARY	WA	ref	PRIMARY,PID,ARTICLEID	PRIMARY	4	skumar32.S.STAFFID	3	Using index
1	PRIMARY	A	eq_ref	PRIMARY,PID	PRIMARY	8	skumar32.WA.ARTICLEID,skumar32.WA.PID	1	
1	PRIMARY	P	eq_ref	PRIMARY	PRIMARY	4	skumar32.WA.PID	1	

5 rows in set (0.00 sec)

Query 2 [Find Publication based on Topic]:

SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

1. SQL Query:

SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

```

MariaDB [skumar32]> SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

```

PID	TOPIC	TITLE	PUB_NO	PUBLICATIONDATE	ISBN	EDITION
3	Mystery	Gone girl	B2	2009-02-19	123-3-86-147810-0	1

1 row in set (0.01 sec)

2. SQL Query with Explain directive:

EXPLAIN SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

```
MariaDB [skumar32]> EXPLAIN SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	BOOK	ALL	PRIMARY	NULL	NULL	NULL	4	
1	SIMPLE	PUBLICATION	eq_ref	PRIMARY	PRIMARY	4	skumar32.BOOK.PID	1	Using where

2 rows in set (0.01 sec)

3. Index Creation Query:

CREATE INDEX TOPIC_INDEX ON PUBLICATION(TOPIC);

```
MariaDB [skumar32]> CREATE INDEX TOPIC_INDEX ON PUBLICATION(TOPIC);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

4. SQL Query Output post creation of index:

EXPLAIN SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

```
MariaDB [skumar32]> EXPLAIN SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	PUBLICATION	ref	PRIMARY,TOPIC_INDEX	TOPIC_INDEX	32	const	1	Using index condition
1	SIMPLE	BOOK	eq_ref	PRIMARY	PRIMARY	4	skumar32.PUBLICATION.PID	1	

2 rows in set (0.00 sec)

4.3

1. FIND PUBLICATION BASED ON TOPIC:

SELECT * FROM PUBLICATION NATURAL JOIN BOOK WHERE PUBLICATION.TOPIC = "Mystery";

$\Pi_{*}(\sigma_{\text{PUBLICATION.TOPIC} = \text{"Mystery"}}(\text{PUBLICATION} \bowtie \text{BOOK}))$

Suppose p is any tuple in PUBLICATION and b is any tuple in BOOK such that the value p.PID is the same as the value b.PID. Each such combination of tuples (p, b) gives all details about the publication (PID, title, topic, Pub_no) and the book (ISBN and Edition). For each such combination (p,b), the query returns all those tuples whose topic is given as input. These values

are the publication ID, title(given as input), topic, publication number, ISBN and edition. This is exactly what the query should return.

2. Find Publication based on Publication date(Publication date/issue date):

```
SELECT * FROM PUBLICATION NATURAL JOIN PERIODICPUBLICATION  
WHERE ISSUEDATE = '2009-02-22';
```

$$\Pi * (\sigma_{\text{ISSUEDATE} = '2009-03-21'}(\text{PUBLICATION} \bowtie \text{PERIODICPUBLICATION}))$$

Suppose p is any tuple in PUBLICATION and q is any tuple in PERIODICPUBLICATION such that the value p.PID is the same as the value q.PID. Each such combination of tuples (p, q) gives all details about the publication (PID, title, topic, Pub_no) and the book (Issue Date, periodicity). For each such combination (p,q), the query returns all those tuples whose IssueDate is given as input. These values are the publication ID, title, topic, publication number, Issue Date and periodicity. This is exactly what the query should return.