

WolfPubDb

for the *WolfCity* publishing house

CSC540 Database Management Systems
Project Report 2

Vamsi Varada, Aswin Itha, Mohan Kumar, Ishwari Zare
March 11, 2021

1. Global Relational database Schema

- **StaffMember(ID, SSN, Name, Address, PhoneNumber, Gender, Age, JobTitle, DOJ)**
 - **ID \rightarrow ID, SSN, Name, Address, PhoneNumber, Gender, Age, JobTitle, DOJ**
holds because each staff member with all of these attributes have a unique (Employee)ID
 - **SSN \rightarrow ID, SSN, Name, Address, PhoneNumber, Gender, Age, JobTitle, DOJ**
holds because SSN, if present, always maps to one staff member. Thus, it determines all other attributes.

Because the left hand side of each of these functional dependencies is a superkey, this relation is in BCNF and therefore in 3NF. There are no other functional dependencies because no combination of the other attributes is sufficient to determine a unique staff member. It's possible for two different staff members to have identical values for each of the other attributes.

- **Author(AuthorID, EmploymentType)**
 - **AuthorID \rightarrow AuthorID, EmploymentType**
holds since this relation has only two attributes. It is in BCNF and therefore in 3NF.
EmploymentType need not be a key because multiple authors have the same EmploymentType ie., it cannot identify an author.
- **Editor (EditorID, EmploymentType)**
 - **EditorID \rightarrow EditorID, EmploymentType**
holds since this relation has only two attributes. It is in BCNF and therefore in 3NF.
EmploymentType need not be a key because multiple editors have the same EmploymentType ie., it cannot identify an editor.
- **Publication(PID, Title, Type, Topics, CostofEachPiece)**
 - **PID \rightarrow PID, Title, type, Topics, CostofEachPiece**
holds because each publication has a PID that can determine all the other attributes. As the LHS is a superkey, it is in BCNF and hence in 3NF. There are no other functional dependencies because any other attribute or a combination of attributes can determine a unique publication. It's possible for two publications to have identical values for each of the other attributes.

- **Issues(PID, IssueNumber, Type, DateofIssue, Periodicity, TableofContents)**
 - **PID → PID, IssueNumber, Type, DateofIssue, Periodicity, TableofContents**
holds because each issue with all of the respective attributes has a unique PID.
 - **IssueNumber → IssueNumber, PID, Type, DateofIssue, Periodicity, TableofContents**
holds because each issue with all of the respective attributes has a unique IssueNumber.

As LHS of each of these FDs is a superkey, this relation is in BCNF and hence in 3NF. There are no other functional dependencies because any other attribute or a combination of attributes can determine a unique book . It's possible for two issues to have identical values for each of the other attributes.

- **Articles(PID, ArtNum, ArticleName, ArticleText, DateofCreation)**
 - **PID, ArtNum → PID, ArtNum, ArticleName, ArticleText, DateofCreation**
holds because an article is a weak entity set which is unique within a book. PID and IssueNumber together is a key. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.
- **Books(PID, ISBN, Edition, PublicationDate, DateofCreation, TableofContents)**
 - **PID → PID, ISBN, Edition, PublicationDate, DateofCreation, TableofContents**
holds because each book with all of the respective attributes has a unique PID.
 - **ISBN → ISBN, PID, Edition, PublicationDate, DateofCreation, TableofContents**
holds because each book with all of the respective attributes has a unique ISBN.

As LHS of each of these FDs is a superkey, this relation is in BCNF and hence in 3NF. There are no other functional dependencies because any other attribute or a combination of attributes can determine a unique book . It's possible for two books to have identical values for each of the other attributes.

- **Chapter(PID, ChapNum, ChapterName, Contents)**
 - **PID, ChapNum → PID, ChapNum, ChapterName, Contents**
holds because chapter is a weak entity set which is unique within a book. PID and ChapNum together is a key. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.

- **Pens(AuthorID, PID)**
 - **AuthorID, PID → AuthorID, PID**
holds because there are only two attributes and both of them together is a superkey. Hence, it is in BCNF and therefore 3NF.
- **Writes(AuthorID, ArtNum, PID)**
 - **AuthorID, ArtNum, PID → AuthorID, ArtNum, PID**
holds because there are only three attributes and all of them together is a superkey. Hence, it is in BCNF and therefore in 3NF.
- **Edits(EditorID, PID)**
 - **EditorID, PID → EditorID, PID**
holds because there are only two attributes and both of them together is the superkey. Hence, it is in BCNF and therefore in 3NF.
- **Transactions(TID, TransactionDate, Amount, DebitCredit, PaymentMode, TransactionType)**
 - **TID → TID, TransactionDate, Amount, DebitCredit, PaymentMode, TransactionType**
holds because each transaction can be identified by a unique TID. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.
- **Distributor(DID, EIN, DName, OutstandingBalance, Type, Address, City, Location, POC, PhoneNumber)**
 - **DID → DID, EIN, DName, OutstandingBalance, Type, Address, City, Location, POC, PhoneNumber**
holds because each distributor with all of these attributes has a unique DID.
 - **EIN → DID, EIN, D_Name, OutstandingBalance, Type, Address, City, Location, POC, PhoneNumber**
holds because EIN always maps to one distributor. Thus, it determines all other attributes.
 - **PhoneNumber → DID, EIN, D_Name, OutstandingBalance, Type, Address, City, Location, POC, PhoneNumber**
holds because each distributor has its own phone number and no two distributors have the same phone number.

Because the left hand side of each of these functional dependencies is a superkey, this relation is in BCNF and therefore in 3NF. There are no other functional dependencies because no combination of the other attributes is sufficient to determine a unique distributor. It's possible for two different distributors to have identical values for each of the other attributes.

- **Orders(OrderID, DID, OrderDate, PriceOfOrder, DeliveryDate, ShippingCost)**
 - **OrderID \rightarrow OrderID, DID, OrderDate, PriceofOrder, DeliveryDate, ShippingCost**
holds because each order with all of these attributes has a unique OrderID. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.
- **GetsPaid(TID, ID, WorkType)**
 - **TID \rightarrow TID, ID, WorkType**
holds because every transaction to a staff member for a work type can be uniquely identified by a TID. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.
- **MadeBy(TID, DID)**
 - **TID \rightarrow TID, DID**
holds because there are only two attributes and LHS is the superkey. Hence, it is in BCNF and therefore in 3NF.
- **MadeFor(TID, OrderID)**
 - **TID \rightarrow TID, OrderID**
holds because there are only two attributes and LHS is the superkey. Hence, it is in BCNF and therefore in 3NF.
- **Has(PID, OrderID, NumberofCopies, PurchaseCost)**
 - **PID, OrderId \rightarrow PID, OrderID, NumberofCopies, PurchaseCost**
FD holds because PID and OrderID can determine all the attributes. Hence together they form the key for the relationship. As LHS of this FD is a superkey, this relation is in BCNF and hence in 3NF.

2. Design for Global Schema

The entity sets in our diagram were made into relations with the corresponding attributes for Staff Members, Publication, Distributors, Orders and Transactions.

We combined many to one relationships into attributes by making the key of one as an attribute of many example Order and Distributor relations. This reduces redundancy and decreases overhead.

Relationships have been turned into schemas which are defined by the attributes that are keys of the entities they represent. For example MadeBy, MadeFor and GetsPaid are relationships defined by the attributes of their corresponding entities which are Staffmember, Orders and Distributors.

- **StaffMember(ID, SSN, Name, Address, PhoneNumber, Gender, Age, JobTitle, DOJ)**
 - ID is the primary key.
 - SSN has a UNIQUE constraint and SSN is allowed to have NULL values, in case an employee doesn't have an SSN.
 - Name, Address, PhoneNumber, Gender, Age, JobTitle, DOJ cannot have null values.
- **Author(AuthorID, EmploymentType)**
 - AuthorID is the Primary Key.
 - AuthorID is also the Foreign Key referencing StaffMember(ID).
 - EmploymentType has a NOT NULL constraint.
- **Editor (EditorID, EmploymentType)**
 - EditorID is the Primary Key.
 - EditorID is also the Foreign Key referencing StaffMember(ID).
 - EmploymentType has a NOT NULL constraint.
- **Publication(PID, Title, Type, Topics, CostofEachPiece)**
 - PID is the Primary Key.
 - Title, Type, Topics are NOT NULL.
 - CostOfEachPiece is NOT NULL and has a DEFAULT 0
- **Issues(PID, IssueNumber, Type, DateofIssue, Periodicity, TableofContents)**
 - PID is a Primary Key.
 - PID is also the Foreign Key referencing Publication(PID).
 - IssueNumber has a UNIQUE and NOT NULL constraint.
 - Type, DateofIssue, Periodicity, TableofContent is NOT NULL.

- **Articles(PID, ArtNum, ArticleName, ArticleText, DateofCreation)**
 - PID and ArtNum is a Primary Key.
 - PID is the Foreign Key referencing Issues(PID) and it has a DELETE ON CASCADE constraint.
 - ArticleName, ArticleText and DateofCreation is NOT NULL.
- **Books(PID, ISBN, Edition, PublicationDate, DateofCreation, TableofContents)**
 - PID is a Primary Key.
 - PID is also the Foreign Key referencing Publication(PID).
 - ISBN has a UNIQUE and NOT NULL Constraints.
 - Edition, PublicationDate, DateofCreation, TableofContents cannot have NULL values.
- **Chapter(PID, ChapNum, ChapterName, Contents)**
 - PID and ChapNum together is a Primary Key.
 - PID is the Foreign Key referencing Books(PID) and it has a DELETE ON CASCADE constraint.
 - ChapterName and Contents is NOT NULL.
- **Pens(AuthorID, PID)**
 - AuthorID and PID together are the Primary Key
 - AuthorID is the Foreign Key referencing Author(AuthorID).
 - PID is the Foreign Key referencing Books(PID) and it has a DELETE ON CASCADE constraint.
- **Writes(AuthorID, articleNumber, PID)**
 - AuthorID, articleNumber and issueNumber together is a Primary Key.
 - AuthorID is the Foreign Key referencing Author(AuthorID). articleNumber and PID are the Foreign Keys referencing Articles(ArtNum,PID).
 - The articleNumber and PID is also on the DELETE ON CASCADE constraint.
- **Edits(EditorID, PID)**
 - EditorID and PID together is a Primary Key.
 - They are also the Foreign Keys referencing Editor(EditorID) and Publication(PID).
- **Transactions(TID, TransactionDate, Amount, DebitCredit, PaymentMode, TransactionType)**
 - TID is a Primary Key.
 - TransactionDate, Amount, DebitCredit, PaymentMode, TransactionType is NOT NULL.
 - Amount has a DEFAULT 0

- **Distributor(DID, EIN, DName, OutstandingBalance, Type, Address, City, Location, POC, PhoneNumber)**
 - DID is the primary Key.
 - OutstandingBalance, DName, Type, Address, City, Location, POC is NOT NULL.
 - EIN, PhoneNumber should have a UNIQUE and a NOT NULL Constraint.
 - OutstandingBalance has a DEFAULT 0.
- **Orders(OrderID, DID, OrderDate, PriceOfOrder, DeliveryDate, ShippingCost)**
 - OrderID is the primary key.
 - DID is Foreign Key referencing Distributor(DID).
 - OrderDate, DeliveryDate are NOT NULL.
 - Default value of PriceOfOrder, ShippingCost has a DEFAULT 0.
- **GetsPaid(TID, ID, WorkType)**
 - TID is a Primary Key and also a Foreign Key referencing Transactions(TID).
 - ID is a Foreign Key referencing StaffMember(ID).
 - Work Type is NOT NULL.
- **MadeBy(TID, DID)**
 - TID is a Primary Key and is also a Foreign Key referencing Transactions(TID).
 - DID is a Foreign Key referencing Distributor(DID).
- **MadeFor(TID, OrderID)**
 - TID is a Primary Key and also a Foreign Key referencing Transactions(TID).
 - OrderID is a Foreign Key referencing Orders(OrderID).
- **Has(PID, OrderID, NumberofCopies, PurchaseCost)**
 - PID and OrderID together form a Primary Key.
 - PID is a Foreign Key referencing Publication(PID) and OrderID is a Foreign Key referencing Orders(OrderID).
 - NumberofCopies has a DEFAULT 1 and PurchaseCost has a DEFAULT 0.

3. Base Relations

```
CREATE TABLE StaffMember (  
    ID INT,  
    SSN VARCHAR(128) UNIQUE,  
    Name VARCHAR(128) NOT NULL,  
    Address VARCHAR(128) NOT NULL,  
    PhoneNumber VARCHAR(16) NOT NULL,  
    Gender VARCHAR(128) NOT NULL,  
    Age INT NOT NULL,  
    JobTitle VARCHAR(128) NOT NULL,  
    DOJ DATE NOT NULL,  
    PRIMARY KEY(ID) );
```

```
CREATE TABLE Author (  
    AuthorID INT,  
    EmploymentType VARCHAR(128) NOT NULL,  
    PRIMARY KEY(AuthorID),  
    FOREIGN KEY (AuthorID) REFERENCES StaffMember(ID) );
```

```
CREATE TABLE Editor (  
    EditorID INT,  
    EmploymentType VARCHAR(128) NOT NULL,  
    PRIMARY KEY(EditorID),  
    FOREIGN KEY (EditorID) REFERENCES StaffMember(ID) );
```

```
CREATE TABLE Publication (  
    PID INT,  
    Title VARCHAR(128) NOT NULL,  
    Type VARCHAR(128) NOT NULL,  
    Topics VARCHAR(128) NOT NULL,  
    CostOfEachPiece FLOAT DEFAULT 0,  
    PRIMARY KEY(PID) );
```

```
CREATE TABLE Issues (  
    PID INT,  
    IssueNumber INT NOT NULL UNIQUE,  
    Type VARCHAR(128) NOT NULL,  
    DateOfIssue DATE NOT NULL,  
    Periodicity VARCHAR(128) NOT NULL,  
    TableOfContents VARCHAR(128) NOT NULL,  
    PRIMARY KEY(PID),  
    FOREIGN KEY(PID) REFERENCES Publication(PID) );
```

```
CREATE TABLE Articles (  
    PID INT,  
    ArtNum INT,  
    ArticleName VARCHAR(128) NOT NULL,  
    ArticleText VARCHAR(128) NOT NULL,  
    DateOfCreation DATE NOT NULL,  
    PRIMARY KEY(PID,ArtNum),  
    FOREIGN KEY(PID) REFERENCES Issues(PID)  
    ON DELETE CASCADE);
```

```
CREATE TABLE Books (  
    PID INT,  
    ISBN INT NOT NULL UNIQUE,  
    Edition VARCHAR(128) NOT NULL,  
    PublicationDate DATE NOT NULL,  
    DateOfCreation DATE NOT NULL,  
    TableOfContents VARCHAR(128) NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY(PID) REFERENCES Publication(PID) );
```

```
CREATE TABLE Chapter (  
    PID INT,  
    ChapNum INT NOT NULL,  
    ChapterName VARCHAR(128) NOT NULL,  
    Contents VARCHAR(128) NOT NULL,  
    PRIMARY KEY(PID,ChapNum),  
    FOREIGN KEY(PID) REFERENCES Books(PID)  
    ON DELETE CASCADE );
```

```
CREATE TABLE Pens (  
    AuthorID INT,  
    PID INT,  
    PRIMARY KEY(AuthorID,PID),  
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),  
    FOREIGN KEY (PID) REFERENCES Books(PID)  
    ON DELETE CASCADE );
```

```
CREATE TABLE Writes (  
    AuthorID INT,  
    ArtNum INT,  
    PID INT,  
    PRIMARY KEY(AuthorID,ArtNum,PID),  
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),  
    FOREIGN KEY (PID, ArtNum) REFERENCES Articles(PID, ArtNum)  
    ON DELETE CASCADE );
```

```
CREATE TABLE Edits (  
    EditorID INT,  
    PID INT,  
    PRIMARY KEY(EditorID,PID),  
    FOREIGN KEY (PID) REFERENCES Publication(PID),  
    FOREIGN KEY (EditorID) REFERENCES Editor(EditorID) );
```

```
CREATE TABLE Transactions (  
    TID INT,  
    TransactionDate DATE NOT NULL,  
    Amount FLOAT DEFAULT 0,  
    DebitCredit VARCHAR(128) NOT NULL,  
    PaymentMode VARCHAR(128) NOT NULL,  
    TransactionType VARCHAR(128) NOT NULL,  
    PRIMARY KEY(TID) );
```

```
CREATE TABLE Distributor (  
    DID INT,  
    EIN INT NOT NULL UNIQUE,  
    DName VARCHAR(128) NOT NULL,  
    OutstandingBalance FLOAT DEFAULT 0,  
    Type VARCHAR(128) NOT NULL,  
    Address VARCHAR(128) NOT NULL,  
    City VARCHAR(128) NOT NULL,  
    Location VARCHAR(128) NOT NULL,  
    POC VARCHAR(128) NOT NULL,  
    PhoneNumber VARCHAR(10) NOT NULL UNIQUE,  
    PRIMARY KEY(DID) );
```

```
CREATE TABLE Orders (  
    OrderID INT,  
    DID INT,  
    OrderDate DATE NOT NULL,  
    PriceOfOrder FLOAT DEFAULT 0,  
    DeliveryDate DATE NOT NULL,  
    ShippingCost FLOAT DEFAULT 0,  
    PRIMARY KEY(OrderID),  
    FOREIGN KEY (DID) REFERENCES Distributor(DID) );
```

```
CREATE TABLE GetsPaid (  
    TID INT,  
    ID INT,  
    WorkType VARCHAR(128) NOT NULL,  
    PRIMARY KEY(TID),  
    FOREIGN KEY (TID) REFERENCES Transactions(TID),  
    FOREIGN KEY (ID) REFERENCES StaffMember(ID) );
```

```
CREATE TABLE MadeBy (  
    TID INT,  
    DID INT,  
    PRIMARY KEY(TID),  
    FOREIGN KEY (TID) REFERENCES Transactions(TID),  
    FOREIGN KEY (DID) REFERENCES Distributor(DID) );
```

```
CREATE TABLE MadeFor (  
    TID INT,  
    OrderID INT,  
    PRIMARY KEY(TID),  
    FOREIGN KEY (TID) REFERENCES Transactions(TID),  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) );
```

```
CREATE TABLE Has (  
    PID INT,  
    OrderID INT,  
    NumberOfCopies INT DEFAULT 1,  
    PurchaseCost FLOAT DEFAULT 0,  
    PRIMARY KEY(PID,OrderID),  
    FOREIGN KEY (PID) REFERENCES Publication(PID),  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) );
```

- show tables;

```

+-----+
| Tables_in_aitha |
+-----+
| Articles        |
| Author          |
| Books           |
| Chapter         |
| Distributor     |
| Editor          |
| Edits           |
| GetsPaid        |
| Has             |
| Issues          |
| MadeBy          |
| MadeFor         |
| Orders          |
| Pens            |
| Publication     |
| StaffMember     |
| Transactions    |
| Writes          |
+-----+
18 rows in set (0.00 sec)

```

- SELECT * FROM StaffMember;

ID	SSN	Name	Address	PhoneNumber	Gender	Age	JobTitle	DOJ
1	1	Aswin	4206 Whistler Ct, Raleigh	9842708890	Male	16	Author	2020-11-02
2	2	Ishwari	4206 Whistler Ct, Raleigh	9842708891	Female	16	Editor	2020-12-02
3	3	Mohan	4206 Whistler Ct, Raleigh	9842708892	Male	16	Editor	2020-11-11
4	4	Vamsi	4206 Whistler Ct, Raleigh	9842708893	Male	16	Admin	2020-11-02
5	5	Kumar	4206 Whistler Ct, Raleigh	9842708894	Male	25	Billing	2020-11-02
6	6	Zare	4206 Whistler Ct, Raleigh	9842708895	Female	16	Admin	2020-11-30
7	7	Madhur	2510 Avent Ferry Rd, Cary	9842708896	Male	16	Sales	2021-11-02
8	8	Itha	4206 Whistler Ct, Raleigh	9842708897	Male	16	HR	2020-10-02
9	9	Varada	4206 Whistler Ct, Raleigh	9842708898	Male	16	Editor	2019-11-02
10	10	Kanakapura	4206 Whistler Ct, Raleigh	9842708899	Male	16	Admin	2016-11-02
11	11	Shekhar	4206 Whistler Ct, Raleigh	9842448890	Male	16	Author	2020-03-02

- **SELECT * FROM Author;**

AuthorID	EmploymentType
1	Permanent
11	Invited

- **SELECT * FROM Editor;**

EditorID	EmploymentType
2	Invited
3	Permanent
9	Permanent

- **SELECT * FROM Publication;**

PID	Title	Type	Topics	CostOfEachPiece
101	Gone With Wind	Book	Love Story	10
102	IEEE	Journal	Science and Technology	15
103	Angels and Demons	Book	Fiction	100
104	AutoCar	Magazine	Automotives	26
105	Family Circle	Magazine	Lifestyle	79
106	Wolf of WallStreet	Journal	Stock Market	12
107	Immortals of Meluha	Book	Mythological	33

- **SELECT * FROM Issues;**

PID	IssueNumber	Type	DateOfIssue	Periodicity	TableOfContents
102	1211	Journal	2011-05-02	Monthly	First Line Second Line
104	1212	Magazine	2020-07-23	Annual	One Line Two Line
105	1213	Magazine	2000-05-14	Monthly	1 Line 2Line
106	1214	Journal	1990-01-01	Annual	I Line II Line

- **SELECT * FROM Articles;**

PID	ArtNum	ArticleName	ArticleText	DateOfCreation
102	1501	Snake Species	Lorem Ipsum	2020-03-09
102	1502	Radio Activity effect on snakes	Valor	2020-03-09
102	1503	Life of Snakes	Death of snake	2020-03-09
104	1501	Speed thrills	But Kills	2012-07-15
104	1502	Road Rage	Be soft on my Curves	2012-07-15
104	1503	Fast and Furious	Tokyo Drift	2012-07-15
105	1502	Fat and Curious	Obesity	2019-08-25
105	1503	Hit the Gym	Be healthy	2019-08-25
105	1504	Spa	Sona	2019-08-25
106	1502	Robinhood	Get rich or die anyway	2009-02-12
106	1503	Buy the Dip	Taste the dip	2009-02-12
106	1504	Dalal Street	Ka dariya	2009-02-12

- **SELECT * FROM Books;**

PID	ISBN	Edition	PublicationDate	DateOfCreation	TableOfContents
101	501	Vol-1	2002-07-04	2000-03-19	Line-1Line-2
103	502	Vol-7	2012-09-24	2007-10-31	Line-1 Line-2 Line-3
107	503	Part-2	2005-01-12	1999-12-01	Line-1Line-2

- **SELECT * FROM Chapter;**

PID	ChapNum	ChapterName	Contents
101	1	Into the woods	Lorem ipsum valor
101	2	Into the Forest	ipsum valor
101	3	Into the Mountains	Fell and broken
103	1	Mason Power	The secret people
103	2	What happened that day	They are everywhere
107	1	The Nagas	Who are they?
107	2	the oath	Promise made is made
107	3	Vayuputras	Friends in need

- **SELECT * FROM Pens;**

AuthorID	PID
1	101
1	103
1	107
11	101
11	103

- **SELECT * FROM Writes;**

AuthorID	ArtNum	PID
1	1501	102
1	1502	104
1	1502	105
1	1503	104
1	1503	105
1	1503	106
1	1504	106
11	1501	104
11	1502	102
11	1502	106
11	1503	102
11	1504	105

- **SELECT * FROM Edits;**

EditorID	PID
2	101
2	103
2	107
3	107
9	103
9	107

- **SELECT * FROM Transactions;**

TID	TransactionDate	Amount	DebitCredit	PaymentMode	TransactionType
5001	2022-01-01	20	Credit	Cash	Distributor
5002	2022-02-07	500	Credit	DebitCard	Distributor
5003	2022-02-11	25.5	Credit	CreditCard	Distributor
5004	2022-02-27	330	Credit	Cash	Distributor
5005	2022-01-15	12	Debit	Cheque	Shipping Cost
5006	2022-01-15	2.9	Debit	Cheque	Shipping Cost
5007	2022-02-15	3.33	Debit	Cheque	Shipping Cost
5008	2022-02-15	0	Debit	Cheque	Shipping Cost
5009	2022-03-01	5000	Debit	ACH	Salary
5010	2022-03-01	6000	Debit	ACH	Salary
5011	2022-03-01	4501	Debit	ACH	Salary
5012	2022-03-01	3784	Debit	ACH	Salary
5013	2022-03-01	2878	Debit	ACH	Salary
5014	2022-03-01	2344	Debit	ACH	Salary
5015	2022-03-01	5028	Debit	ACH	Salary
5016	2022-03-01	2829	Debit	ACH	Salary
5017	2022-03-01	2834	Debit	ACH	Salary
5018	2022-03-01	1123	Debit	ACH	Salary
5019	2022-03-01	2311	Debit	ACH	Salary

- **SELECT * FROM Distributor;**

DID	EIN	DName	OutstandingBalance	Type	Address	City	Location	POC	PhoneNumber
7001	8001	Higgin Bothams	50	Wholesale	2842 Avent Ferry	New York	Wake	Larry James	5134789920
7002	8002	Wolfpack Outfitters	110	Book Store	4400 Gorman Street	Raleigh	Wake	Dan Green	5134789921
7003	8003	Hunt Library	89.5	Library	3210 Falcon Ridge	Raleigh	Wake	John Paul	5134789922
7004	8004	Hill Lib	55	Library	3420 Kings Ct	Cary	Wake	James Bond :)	5134789923

- **SELECT * FROM Orders;**

OrderID	DID	OrderDate	PriceOfOrder	DeliveryDate	ShippingCost
90005	7001	2022-01-01	60	2022-01-10	12
90006	7001	2021-01-07	210	2022-02-10	2.9
90007	7002	2022-02-11	115	2022-02-15	3.33
90008	7002	2022-02-01	85	2022-02-10	0

- **SELECT * FROM GetsPaid;**

TID	ID	WorkType
5009	1	Permanent
5010	2	Invited
5011	3	Permanent
5012	4	Permanent
5013	5	Permanent
5014	6	Permanent
5015	7	Permanent
5016	8	Permanent
5017	9	Permanent
5018	10	Permanent
5019	11	Invited

- **SELECT * FROM MadeBy;**

TID	DID
5001	7001
5002	7002
5003	7003
5004	7004

- `SELECT * FROM MadeFor;`

TID	OrderID
5005	90001
5006	90002
5007	90003
5008	90004

- `SELECT * FROM Has;`

PID	OrderID	NumberOfCopies
101	90001	4
103	90002	10
104	90003	2
107	90004	10

4. SQL Queries

• 4.1 Information Processing

- **Enter basic info on a new publication**

```
INSERT INTO Publication VALUES(108,"Wings of Fire","Book","Motivation",69);
```

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

- **Update publication info**

```
UPDATE Publication SET Topics='Encouragement' WHERE PID=108;
```

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

```
Rows matched: 1 Changed: 1 Warnings: 0
```

- **Assign editor to a publication**

```
INSERT INTO Edits VALUES(9,101);
```

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

- **Editor view on his/her own publication**

```
SELECT * FROM Publication WHERE PID IN (SELECT PID FROM EDITS WHERE  
EditorID= 2);
```

```
+-----+-----+-----+-----+-----+  
| PID | Title                | Type | Topics          | CostOfEachPiece |  
+-----+-----+-----+-----+-----+  
| 101 | Gone With Wind       | Book | Love Story      | 10               |  
| 103 | Angels and Demons    | Book | Fiction         | 100              |  
| 107 | Immortals of Meluha  | Book | Mythological    | 33               |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

- **Edit table of contents of a Publication**

```
UPDATE Books SET TableOfContents = "Line-1Line-2Line-3" WHERE PID = 101;
```

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

```
Rows matched: 1 Changed: 1 Warnings: 0
```

- **Add Articles**

```
INSERT INTO Articles VALUES(106,1505,"Market is Bullish","BULLY","2022-03-09");
```

- **Delete Articles**

DELETE FROM Articles WHERE PID=106 AND ArtNum =1505;

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

- **Add Chapter**

INSERT INTO Chapter VALUES(101,4,"Into the Wild","ipsum valor");

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

a

- **Delete chapter**

DELETE FROM Chapter WHERE PID = 101 AND ChapNum=4;

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

- **Enter new book edition**

INSERT INTO Publication VALUES(109,"Gone With Wind","Book","Love Story",10);

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

INSERT INTO Books VALUES(109,504,"Vol-2","2005-06-03","2003-02-1","Line-1Line-2Line-3");

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

- **Get Book Details**

SELECT * FROM Books where ISBN=501;

```
+-----+-----+-----+-----+-----+-----+
| PID | ISBN | Edition | PublicationDate | DateOfCreation | TableOfContents |
+-----+-----+-----+-----+-----+-----+
| 101 | 501 | Vol-1 | 2002-07-04 | 2000-03-19 | Line-1Line-2 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- **Enter new issue**

INSERT INTO Publication VALUES(110,"AutoCar","Magazine","Automotives",25.5);

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

INSERT INTO Issues VALUES(110,1215,"Journal","2012-12-05","Annual","First Line
Second Line th");

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

- **Update book edition**

UPDATE Books SET Edition='Volume-7' WHERE PID=103;

```
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- **Update an Issue**

UPDATE Issues SET DateOfIssue='2011-07-12' WHERE PID=102;

```
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- **Delete book edition**

DELETE FROM Books WHERE PID=108;

```
MariaDB [vvarada3]> DELETE FROM Books where PID=108;
Query OK, 1 row affected (0.00 sec)
```

- **Delete issue**

DELETE FROM Issues WHERE PID=109;

```
[MariaDB [vvarada3]> DELETE FROM Issues where PID=109;
Query OK, 1 row affected (0.01 sec)
```

- **Enter an Article**

INSERT INTO Articles VALUES(106,1505,"Market is Bullish","BULLY","2022-03-09");

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

- **Enter a Chapter**

INSERT INTO Chapter VALUES(101,4,"Into the Sea","Ser Loras");

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

- **Update an article**

UPDATE Articles SET ArticleName="Snake and its Life" WHERE PID=102 AND ArtNum=1503;

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

- **Update a Chapter**

UPDATE Chapter SET ChapterName="Into the Plains" WHERE PID=101 AND ChapNum=1;

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

- **Update text of an article**

UPDATE Articles SET ArticleText="What is it to be a snake." WHERE PID=102 AND ArtNum=1503;

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

- **Enter author payment**

INSERT INTO Transactions VALUES(5020,"2022-03-01",20,"Debit","ACH","Salary");

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

INSERT INTO GetsPaid VALUES (5020,1,'Permanent');

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

- **Enter editor payment**

INSERT INTO Transactions VALUES(5021,"2022-02-05",50,"Debit","ACH","Salary");

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

INSERT INTO GetsPaid VALUES (5021,3,'Permanent');

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

- **Payment Claimed by addressee**

SELECT TransactionDate FROM Transactions WHERE TID IN (SELECT TID FROM GetsPaid WHERE ID=1);

```
+-----+
| TransactionDate |
+-----+
| 2022-03-01      |
| 2022-03-01      |
+-----+
2 rows in set (0.01 sec)
```


- **Enter new distributor**

INSERT INTO Distributor VALUES(7101,8101,"Higgin Bothams",24,"Wholesale","New York","Larry James",5134789420);

Query OK, 1 row affected (0.01 sec)

- **Update Distributor info**

UPDATE Distributor SET City="Durham" WHERE DID=7101;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

- **Delete Distributor**

DELETE FROM Distributor WHERE DID = 7005 ;

Query OK, 1 row affected (0.01 sec)

- **Input Order from Distributor for a certain date**

INSERT INTO Orders VALUES(90005,7004,"2022-02-28",55,"2022-03-15",0);

Query OK, 1 row affected (0.01 sec)

- **Change Outstanding Balance on Receiving Payment**

update Distributor SET OutstandingBalance = 0 where DID =7004;

Query OK, 1 row affected (0.01 sec)

insert into Transactions values(50023,'2022-02-13',55,'Credit','ACH','Distributor');

insert into Transactions values(50023,'2022-02-13',55,'Credit','ACH','Distributor');

- **Number and Total Price of Copies per publication per distributor per month**

```

SELECT      Year(O.OrderDate) As Year, Month(O.OrderDate) AS Month,
            PID, SUM(NumberOfCopies) AS "No of Copies",
            O.DID, SUM(H.PurchaseCost) AS "Purchase Cost"

FROM Has H
JOIN      Orders O      ON      H.OrderID=O.OrderID
GROUP BY  Year,Month,PID, O.DID
ORDER BY  Year DESC, MONTH DESC;

```

Year	Month	PID	No of Copies	DID	Purchase Cost
2022	2	101	4	7002	40
2022	2	102	4	7002	60
2022	2	103	1	7002	100
2022	1	101	3	7001	30
2022	1	102	2	7001	30
2021	1	101	1	7001	10
2021	1	103	2	7001	200

7 rows in set (0.00 sec)

- **Total Revenue**

```

SELECT SUM(Amount) AS "Revenue" FROM Transactions WHERE
DebitCredit="Credit";

```

```

+-----+
| Revenue |
+-----+
| 745 |
+-----+
1 row in set (0.00 sec)

```

- **Total Expense**

```

SELECT SUM(Amount) AS "Expense" FROM Transactions WHERE
DebitCredit="Debit";

```

```

+-----+
| Expense |
+-----+
| 38650.23000001907 |
+-----+
1 row in set (0.00 sec)

```

- **Total Current number of Distributors**

SELECT COUNT(1) AS "No of Distributors" FROM Distributor;

```
+-----+
| No of Distributors |
+-----+
|                    4 |
+-----+
1 row in set (0.01 sec)
```

- **Revenue per City,**

```
SELECT      City, SUM(Amount) AS Revenue
from        Transactions T
JOIN        MadeBy M    on    T.TID = M.TID
JOIN        Distributor D on    M.DID = D.DID GROUP BY City;
```

```
MariaDB [vvarada3]> SELECT City, SUM(Amount) from Transactions T JOIN MadeBy M on T.TID = M.TID JOIN Distributor D on M.DID = D.DID GROUP BY City;
+-----+-----+
| City | SUM(Amount) |
+-----+-----+
| Cary |          30 |
| New York |       10 |
| Raleigh |      125.5 |
+-----+-----+
3 rows in set (0.01 sec)
```

- **Revenue per Location,**

```
SELECT      Location, SUM(Amount) AS Revenue
from        Transactions T
JOIN        MadeBy M    on    T.TID = M.TID
JOIN        Distributor D on    M.DID = D.DID GROUP BY Location;
```

```
+-----+-----+
| Location | Revenue |
+-----+-----+
| Wake    |      745 |
+-----+-----+
1 row in set (0.00 sec)
```

- **Revenue per Distributor**

```
SELECT      M.DID, sum(T.Amount) AS Revenue
from        Transactions T
JOIN        MadeBy M    on    T.TID=M.TID
GROUP BY    M.DID;
```

DID	SUM
7001	10
7002	100
7003	25.5
7004	30

4 rows in set (0.00 sec)

- **Total Payment to editor or author per time period**

```
SELECT      A.AuthorID, Year(T.TransactionDate) As Year,
            Month(T.TransactionDate) AS Month,
            SUM(T.Amount) AS "Amount"
FROM Transactions T
JOIN        GetsPaid G    ON    G.TID = T.TID
JOIN        Author A      ON    A.AuthorID = G.ID
GROUP BY    Year, Month, A.AuthorID
ORDER BY    Year DESC, Month Desc;
```

AuthorID	Year	Month	"Amount"
1	2022	3	5000
11	2022	3	4622
11	2021	5	5511

3 rows in set (0.00 sec)

```
SELECT      E.EditorID, Year(T.TransactionDate) As Year,
            Month(T.TransactionDate) AS Month,
            SUM(T.Amount) AS "Amount"
FROM Transactions T
JOIN        GetsPaid G    ON    G.TID = T.TID
JOIN        Editor E      ON    E.EditorID = G.ID
GROUP BY    Year, Month, E.EditorID
ORDER BY    Year DESC, Month Desc;
```

EditorID	Year	Month	"Amount"
2	2022	3	6000
3	2022	3	4501
9	2022	3	2834

- **Total payment per work type**

```
SELECT      G.WorkType, SUM(T.Amount) AS Amount
FROM Transactions T
JOIN      GetsPaid G    ON    T.TID = G.TID
GROUP BY    G.WorkType;
```

```
+-----+-----+
| WorkType | Amount |
+-----+-----+
| Invited  |  16133 |
| Permanent |  30321 |
+-----+-----+
2 rows in set (0.00 sec)
```

- 4.2

Total Expense

1. EXPLAIN SELECT SUM(Amount) AS "Expense" FROM Transactions WHERE DebitCredit="Debit";

2.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	Transactions	ALL	NULL	NULL	NULL	NULL	22	Using where

1 row in set (0.00 sec)

3. CREATE INDEX DebitCreditIndex on Transactions(DebitCredit);

4.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	Transactions	ref	DebitCreditIndex	DebitCreditIndex	130	const	17	Using index condition

1 row in set (0.01 sec)

Get book details

1. EXPLAIN SELECT * FROM Books where ISBN=501;

2.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	Books	const	ISBN	ISBN	4	const	1	

1 row in set (0.00 sec)

3. CREATE INDEX isbnIndex ON Books(ISBN);

4.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	Books	ref	isbnIndex	isbnIndex	4	const	1	

1 row in set (0.00 sec)

3. CREATE INDEX isbnIndex ON Books(ISBN);

- **4.3 Query Correctness**

- a. **Total Payment to author per time period**

```

SELECT      A.AuthorID,
            Year(T.TransactionDate) AS Year,
            Month(T.TransactionDate) AS Month,
            SUM(T.Amount) AS Amount

FROM Transactions T
JOIN      GetsPaid G    ON    G.TID = T.TID
JOIN      Author A      ON    A.AuthorID = G.ID

GROUP BY   Year, Month, A.AuthorID
ORDER BY   Year DESC, Month Desc;

```

Relational Algebra

$$\pi_{(\text{AuthorID}, \text{Year}, \text{Month}, \text{Amount})} (\gamma_{\text{A.AuthorID}, \text{Year}(\text{T.TransactionDate}), \text{Month}(\text{T.TransactionDate}), \text{SUM}(\text{T.Amount}), \{\text{A.AuthorID}, \text{Year}, \text{Month}, \text{Amount}\}} (\rho_{\text{T}}(\text{Transactions}) \bowtie_{\text{G.TID} = \text{T.TID}} \rho_{\text{G}}(\text{GetsPaid}) \bowtie_{\text{A.AuthorID} = \text{G.ID}} \rho_{\text{A}}(\text{Author})))$$

Suppose T is a tuple in Transactions relations and G is a tuple in GetsPaid relation such that G.TID is same as T.TID. A join on these two relations with these common tuples provides us information about the transactions made as payment to the Staff members. Further, suppose A is a tuple in the relation Author indicating AuthorID and G is a tuple in the relation GetsPaid indicating AuthorID as ID such that A.AuthorID and G.ID is the same. A join on these would give us information about which transactions to the staff members are made to the author. Grouping together the tuples based on the AuthorID and applying the aggregate function of sum on the Amount attribute for these tuples and also selecting the TransactionDate attribute for grouping keeping a track of time period, we can get the amount paid to the author per time period. We select the attributes of AuthorId, Year, Month and Amount.

- b. **Total payment per work type**

```

SELECT      G.WorkType, SUM(T.Amount) AS Amount
FROM Transactions T
JOIN      GetsPaid G ON T.TID = G.TID
GROUP BY   G.WorkType;

```

Relational Algebra

$$\pi_{(\text{WorkType}, \text{Amount})} (\gamma_{\text{G.WorkType}, \text{Sum}(\text{T.Amount})} (\rho_{\text{T}}(\text{Transactions}) \bowtie_{\text{T.TID} = \text{G.TID}} \rho_{\text{G}}(\text{GetsPaid})))$$

Suppose T is a tuple in Transactions relation and G is a tuple in the GetsPaid relation such that T.TID is same as G.TID. The join of these two relations gives us information about The transactions made for each work type such as book authorship , article authorship or editorial work to the employees. By grouping together the transactions made as a payment to the employees based on the work type, we apply the aggregate function of sum to calculate the total payment made per work type. The query thus returns the worktype and total payment made for the same.