

The Comicbook Management System (CMS)

Andrew Wiebe, Minghuai Zheng, Nadir Khan INF1343H

Dr. Arik Senderovich

April 3, 2020

1. Queries

1. Return the ISBN of books published by Marvel that cost less than 8\$ to purchase.
2. Return the publisher of Silver 'Surfer.'
3. Return the total number of unique customers.
4. Return the addresses for the warehouse that ISBN 0671242256 came from.
5. Return names of Books from Warehouse 2 that have a purchase price of less than 8\$
6. Return names of Books create by Steve Ditko
7. Return name of Customer that has email with laser-ham@hotmail.ca
8. Return Artist name who created X-MEN
9. Return Book title and volume with ISBN number 671242257
10. Return city in which Silver Surfer was published.

2. Translate Queries to Relational Algebra

1. π ISBN (σ Publisher_Name = 'Marvel' \wedge Purchase_Price > 8)(Books)
2. π Publisher_Name (σ Title = 'Silver Surfer') (Books)
3. Count queries cannot be expressed in relational algebra.
4. π Warehouse_Code (Warehouse_Code = Purchase_ID.Warehouse_Code \wedge Purchase_ID.Purchase_ISBN = 671242256 Purchase_ID) (Warehouse \bowtie Warehouse)
5. π Title (σ Warehouse_Code = 2 \wedge Purchased_Price < 8) (Warehouse \bowtie Books)
6. π Title (σ Author_Name = 'Steve Ditko') (Books)
7. π Customer_Name (σ Email = 'laser-ham@hotmail.ca') (Customer)
8. π Artist_Name (σ Comments = 'X-MEN') (Artist)
9. π Title^Volume (σ ISBN = 671242257) (Books)
10. π City_of_Publication (Publisher_Name = Books.Publisher_Name \wedge Books.Title = 'Silver Surfer') (Books \bowtie Publisher)

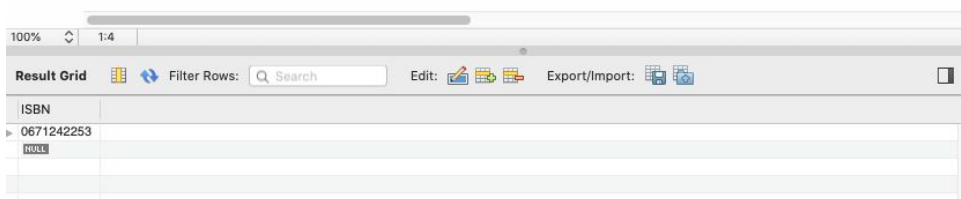
3. Queries in MYSQL

1. SELECT ISBN

FROM Books

Where Publisher_Name = 'Marvel' and Purchase_Price > 8

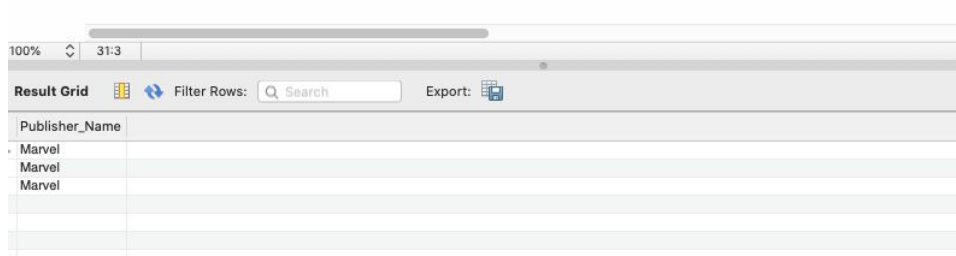
```
1 • SELECT ISBN
2 FROM Books
3 Where Publisher_Name = 'Marvel' and Purchase_Price > 8
4
5
```



The screenshot shows a SQL query result grid. The query is: `SELECT ISBN FROM Books Where Publisher_Name = 'Marvel' and Purchase_Price > 8`. The result grid has a column header 'ISBN' and one data row containing the value '0671242253'. Below the ISBN value, there is a 'NULL' value in another column. The interface includes a search bar, 'Filter Rows' button, and 'Export/Import' options.

2. SELECT Publisher_Name
FROM Books
Where Title = 'Silver Surfer'

```
1 • SELECT Publisher_Name
2 FROM Books
3 Where Title = 'Silver Surfer'
```

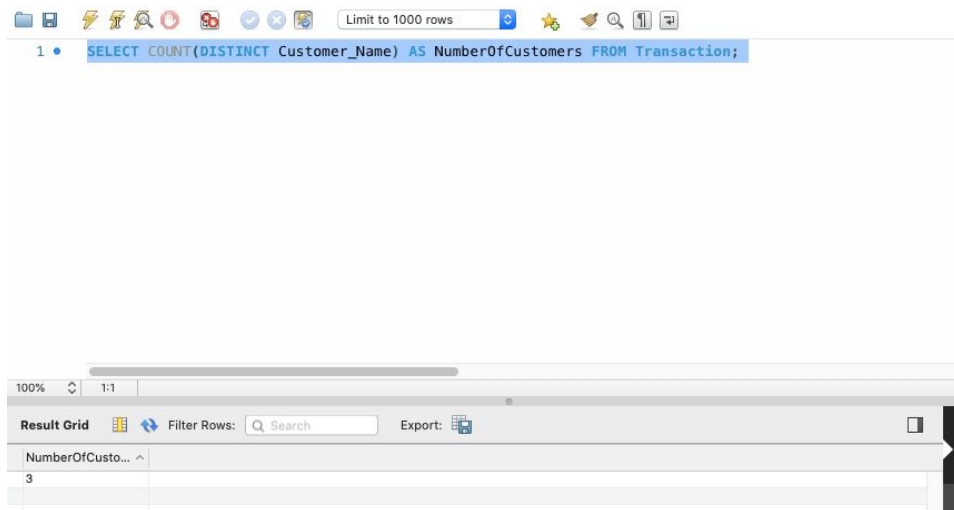


The screenshot shows a SQL query result grid. The query is: `SELECT Publisher_Name FROM Books Where Title = 'Silver Surfer'`. The result grid has a column header 'Publisher_Name' and three data rows, each containing the value 'Marvel'. The interface includes a search bar, 'Filter Rows' button, and 'Export' options.

3. SELECT COUNT(DISTINCT Customer_Name)

AS NumberOfCustomers

FROM Transaction;



4. SELECT Warehouse.Warehouse_Code

FROM Warehouse JOIN Purchase_ID

ON Warehouse.Warehouse_Code = Purchase_ID.Warehouse_Code

AND Purchase_ID.Purchase_ISBN = 0671242256

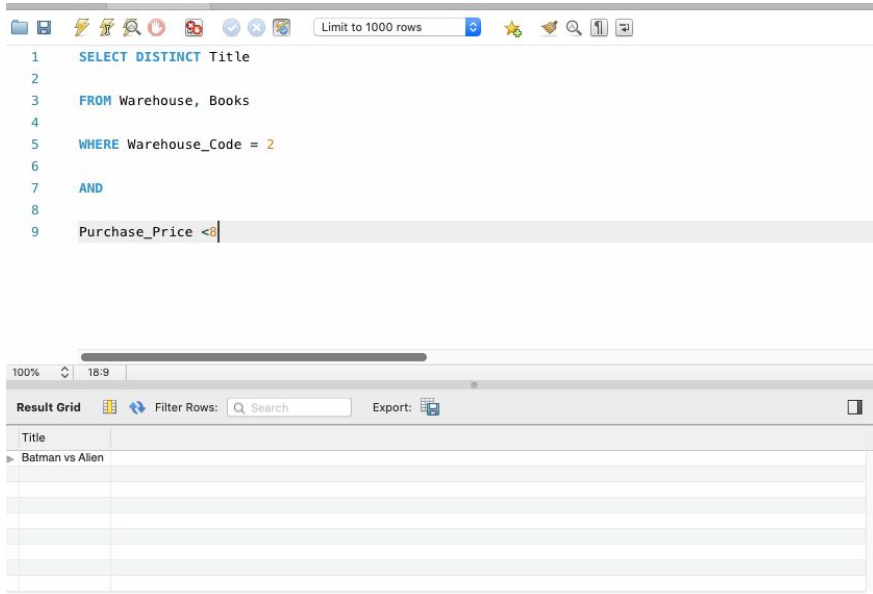
```
Query 2
Limit to 1000 rows
1 • SELECT Warehouse.Warehouse_Code
2 FROM Warehouse JOIN Purchase_ID
3 ON Warehouse.Warehouse_Code = Purchase_ID.Warehouse_Code
4 AND Purchase_ID.Purchase_ISBN = 0671242256
```

30% 43:4

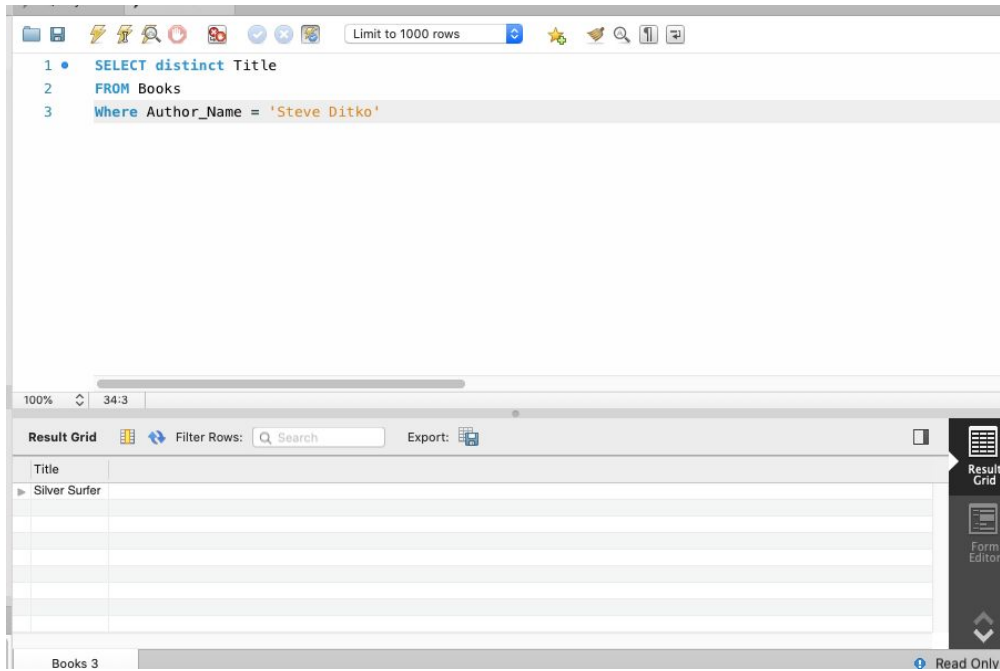
Result Grid Filter Rows: Search Export:

Warehouse_Code
1
2

5. SELECT distinct Title
FROM Warehouse, Books
WHERE Warehouse_Code = 2
AND
Purchase_Price <8



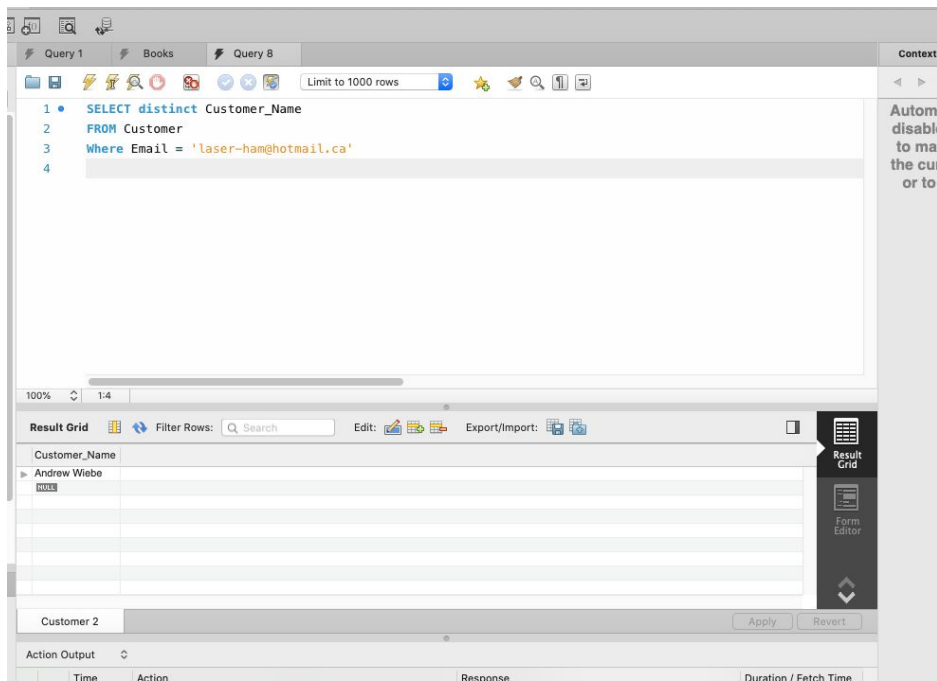
6. SELECT distinct Title
FROM Books
Where Author_Name = 'Steve Ditko'



7. SELECT distinct Customer_Name

FROM Customer

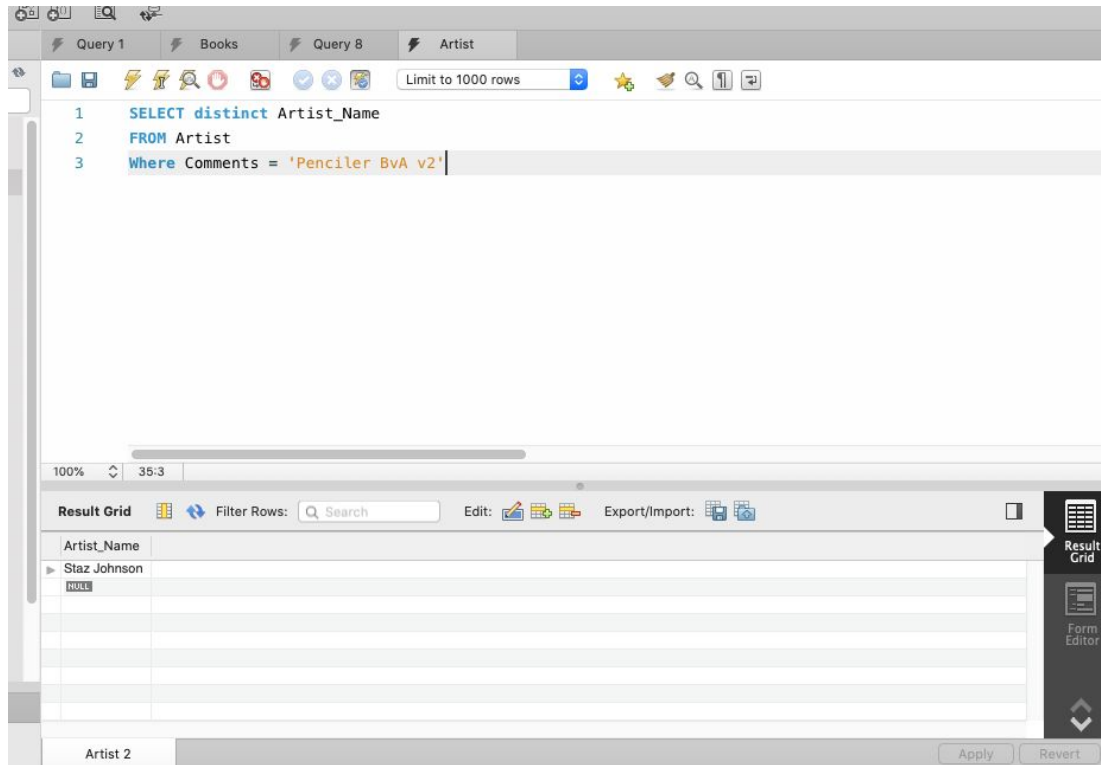
Where Email = 'laser-ham@hotmail.ca'



8. SELECT distinct Artist_Name

FROM Artist

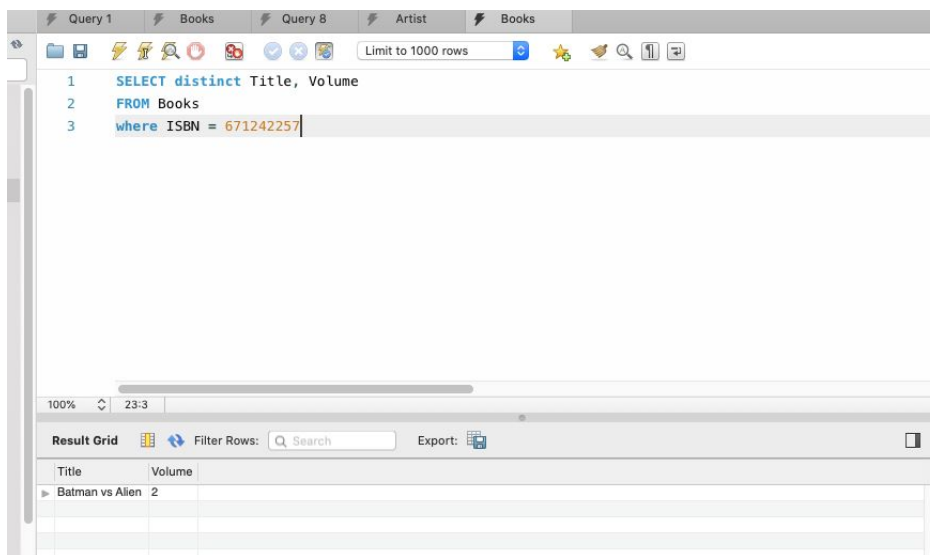
Where Comments = 'Penciler BvA v2'



9. SELECT distinct Title, Volume

FROM Books

where ISBN = 671242257

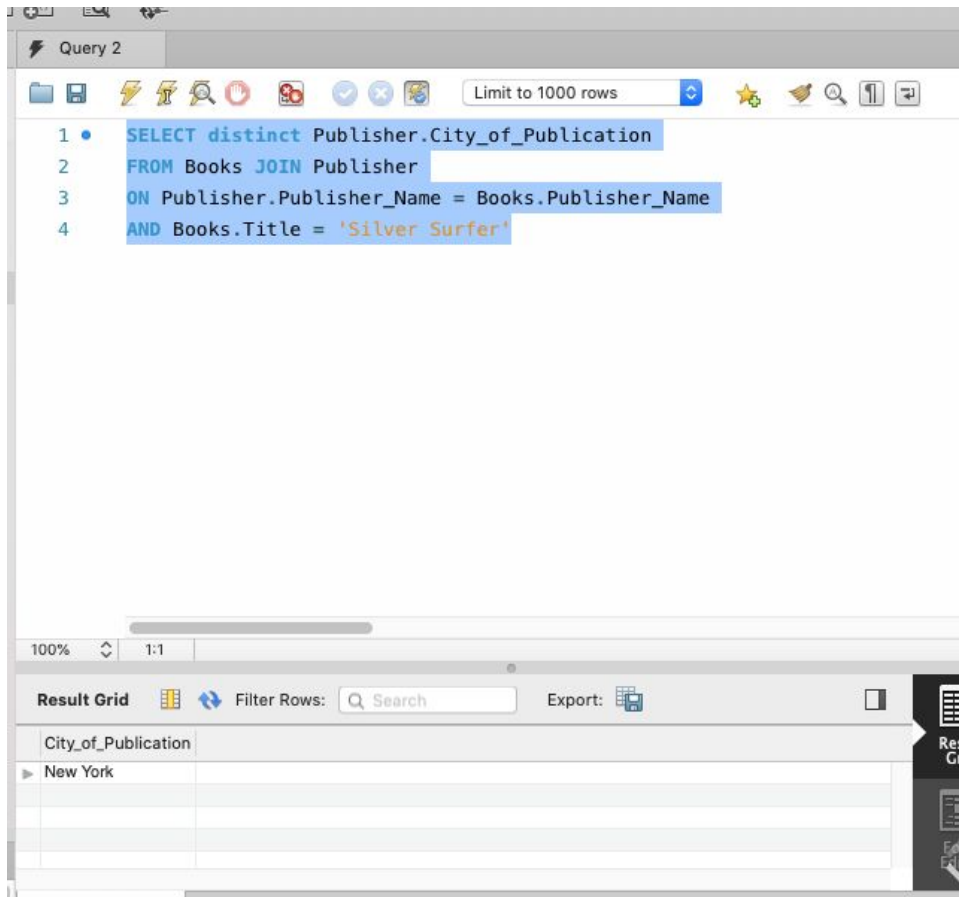


10. SELECT distinct Publisher.City_of_Publication


```
FROM Books JOIN Publisher
```

```
ON Publisher.Publisher_Name = Books.Publisher_Name
```

```
AND Books.Title = 'Silver Surfer'
```



4. Deletions/Insertions/Updates (2 of each)

```
DELETE FROM `Stan's Comics DB`.`Books` WHERE (`ISBN` = '671242258');
```

```
DELETE FROM `Stan's Comics DB`.`Inventory_Location` WHERE (`Publisher_Address` = '666 Antibes Rd');
```

```
INSERT INTO `Stan's Comics DB`.`Customer` (`Email`, `Customer_Name`, `Phone`, `Address`) VALUES ('pandahammer@mail.com', 'Paul Ruff', '(416) 555-2300', '131-2001 Box Ave');
```

```
INSERT INTO `Stan's Comics DB`.`Books` (`Volume`, `Title`, `Barcode`, `ISBN`, `Issue`, `Purchase_Price`, `Artist_Name`, `Author_Name`, `Inventory_Location_Address`,
```

```
`Publisher_Name`, `Count`, `Purchase_Year`, `Publication_Year`) VALUES ('2', 'xxxHolic', '1006', '671242259', '1', '10', 'Clamp', 'Clamp', '666 Antibes Rd', 'Clamp', '2', '2005', '2007');
```

```
UPDATE `Stan's Comics DB`.`Author` SET `Address` = '4005 Markup St, Tokyo' WHERE (`Name` = 'Clamp');
```

```
UPDATE `Stan's Comics DB`.`Books` SET `Inventory_Location_Address` = '666 Antibes Rd' WHERE (`ISBN` = '671242254');
```

5. DDL

```
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;  
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZE  
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----  
-- Schema mydb  
-----  
-----
```

```
-- Schema Stan's Comics DB  
-----  
-----
```

```
-----  
-- Schema Stan's Comics DB  
-----  
-----
```

```
CREATE SCHEMA IF NOT EXISTS `Stan's Comics DB` DEFAULT CHARACTER SET utf8 ;  
USE `Stan's Comics DB` ;
```

```
-----  
-- Table `Stan's Comics DB`.`Artist`  
-----  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Artist` (  
  `Artist_Name` VARCHAR(45) NOT NULL,  
  `Address` VARCHAR(45) NOT NULL,  
  `Comments` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`Artist_Name`))  
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Author`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Author` (  
  `Name` VARCHAR(255) NOT NULL,  
  `Address` VARCHAR(255) NULL DEFAULT NULL,  
  `URL` VARCHAR(255) NULL DEFAULT NULL,  
  PRIMARY KEY (`Name`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Inventory_Location`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Inventory_Location` (  
  `Publisher_Address` VARCHAR(45) NOT NULL,  
  `Phone` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`Publisher_Address`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Publisher`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Publisher` (  
  `Publisher_Name` VARCHAR(45) NOT NULL,  
  `City_of_Publication` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Publisher_Name`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Books`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Books` (  
  `Volume` INT NOT NULL,  
  `Title` VARCHAR(45) NOT NULL,
```

```

`Barcode` INT NOT NULL,
`ISBN` INT NOT NULL,
`Issue` VARCHAR(45) NOT NULL,
`Purchase_Price` INT NOT NULL,
`Artist_Name` VARCHAR(45) NOT NULL,
`Author_Name` VARCHAR(255) NOT NULL,
`Inventory_Location_Address` VARCHAR(45) NOT NULL,
`Publisher_Name` VARCHAR(45) NOT NULL,
`Count` INT NOT NULL,
`Purchase_Year` VARCHAR(45) NOT NULL,
`Publication_Year` VARCHAR(45) NOT NULL,
PRIMARY KEY (`ISBN`),
INDEX `fk_Books_Artist1_idx` (`Artist_Name` ASC) VISIBLE,
INDEX `fk_Books_Author1_idx` (`Author_Name` ASC) VISIBLE,
INDEX `fk_Books_Inventory_Location1_idx` (`Inventory_Location_Address` ASC) VISIBLE,
INDEX `fk_Books_Publisher1_idx` (`Publisher_Name` ASC) VISIBLE,
CONSTRAINT `fk_Books_Artist1`
    FOREIGN KEY (`Artist_Name`)
    REFERENCES `Stan's Comics DB`.`Artist` (`Artist_Name`),
CONSTRAINT `fk_Books_Author1`
    FOREIGN KEY (`Author_Name`)
    REFERENCES `Stan's Comics DB`.`Author` (`Name`),
CONSTRAINT `fk_Books_Inventory_Location1`
    FOREIGN KEY (`Inventory_Location_Address`)
    REFERENCES `Stan's Comics DB`.`Inventory_Location` (`Publisher_Address`),
CONSTRAINT `fk_Books_Publisher1`
    FOREIGN KEY (`Publisher_Name`)
    REFERENCES `Stan's Comics DB`.`Publisher` (`Publisher_Name`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `Stan's Comics DB`.`Business`
-----

```

```

CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Business` (
  `Business_ID` INT NOT NULL,
  `Address` VARCHAR(45) NULL DEFAULT NULL,
  `Phone` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`Business_ID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```
-----  
-- Table `Stan's Comics DB`.`Customer`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Customer` (  
  `Email` VARCHAR(45) NULL DEFAULT NULL,  
  `Customer_Name` VARCHAR(45) NOT NULL,  
  `Phone` VARCHAR(45) NULL DEFAULT NULL,  
  `Address` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`Customer_Name`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Warehouse`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Warehouse` (  
  `Warehouse_Code` INT NOT NULL,  
  `Phone` VARCHAR(45) NULL DEFAULT NULL,  
  `Address` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`Warehouse_Code`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-----  
-- Table `Stan's Comics DB`.`Purchase_ID`  
-----
```

```
CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Purchase_ID` (  
  `Order_ID` INT NOT NULL,  
  `Count` INT NOT NULL,  
  `Business_Business_ID` INT NOT NULL,  
  `Warehouse_Code` INT NOT NULL,  
  `Inventory_Location_Address` VARCHAR(45) NOT NULL,  
  `Purchase_ISBN` INT NOT NULL,  
  PRIMARY KEY (`Order_ID`),  
  INDEX `fk_Purchase_ID_Business1_idx` (`Business_Business_ID` ASC) VISIBLE,  
  INDEX `fk_Purchase_ID_Warehouse1_idx` (`Warehouse_Code` ASC) VISIBLE,  
  INDEX `fk_Purchase_ID_Inventory Location1_idx` (`Inventory_Location_Address` ASC) VISIBLE,  
  INDEX `fk_Purchase_ID_Book1_idx` (`Purchase_ISBN` ASC) VISIBLE,  
  CONSTRAINT `fk_Purchase_ID_Book1`  
    FOREIGN KEY (`Purchase_ISBN`)
```

```

REFERENCES `Stan's Comics DB`.`Books` (`ISBN`),
CONSTRAINT `fk_Purchase_ID_Business1`
  FOREIGN KEY (`Business_Business_ID`)
    REFERENCES `Stan's Comics DB`.`Business` (`Business_ID`),
CONSTRAINT `fk_Purchase_ID_Inventory Location1`
  FOREIGN KEY (`Inventory_Location_Address`)
    REFERENCES `Stan's Comics DB`.`Inventory_Location` (`Publisher_Address`),
CONSTRAINT `fk_Purchase_ID_Warehouse1`
  FOREIGN KEY (`Warehouse_Code`)
    REFERENCES `Stan's Comics DB`.`Warehouse` (`Warehouse_Code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `Stan's Comics DB`.`Transaction`
-----

```

```

CREATE TABLE IF NOT EXISTS `Stan's Comics DB`.`Transaction` (
  `Transaction_ID` INT NOT NULL,
  `Item_Count` INT NOT NULL,
  `Final_Sale_Price` INT NULL DEFAULT NULL,
  `Customer_Name` VARCHAR(45) NOT NULL,
  `Books_ISBN` INT NOT NULL,
  PRIMARY KEY (`Transaction_ID`),
  INDEX `fk_Transaction_Customer_idx` (`Customer_Name` ASC) VISIBLE,
  INDEX `fk_Transaction_Books1_idx` (`Books_ISBN` ASC) VISIBLE,
  CONSTRAINT `fk_Transaction_Books1`
    FOREIGN KEY (`Books_ISBN`)
      REFERENCES `Stan's Comics DB`.`Books` (`ISBN`),
  CONSTRAINT `fk_Transaction_Customer`
    FOREIGN KEY (`Customer_Name`)
      REFERENCES `Stan's Comics DB`.`Customer` (`Customer_Name`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```