## Bookstore ER Diagram

**Company** — address, city, country, companyName, companyID, zipcode, State

**Employee** — firstName, lastName, employeeID, position, sID

**Customer** — firstName, lastName, customerID, paymentMethod

**Store** — Country, City, Zipcode, State, Store_ID, Address, companyID

**Transaction Details bs_tranD** — tID, title, author, pID

**Transaction bs_tran** — tranID, cID, eID

Company 1 — Own — M Store
Store 1 — has — M Employee
Employee M — has — M Customer
Employee 1 — has — M Transaction bs_tran
Customer 1 — has — M Transaction bs_tran
Transaction Details bs_tranD M — has — 1 Transaction bs_tran

## Bookstore Schema

**bs_company**
- PK companyID
- companyName
- address
- city
- state
- zipcode
- country

**bs_store**
- PK storeID
- FK companyID
- country
- address
- city
- state
- zipcode

**bs_employee**
- PK employeeID
- firstName
- lastName
- FK sID
- position

**bs_tran**
- PK tranID
- FK cID
- FK eID

**bs_tranD**
- PK pID
- FK tID
- author
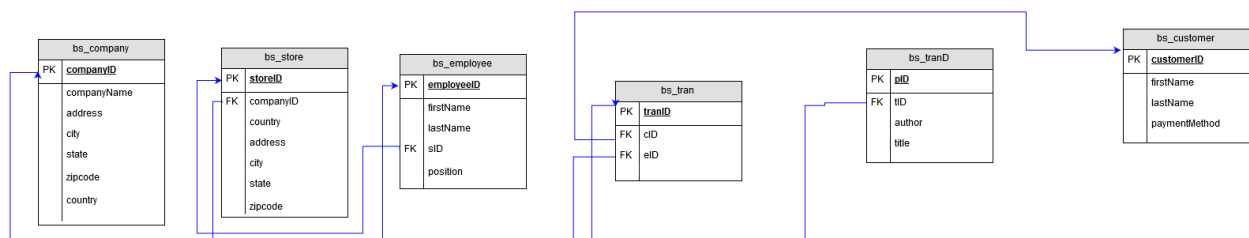- title

**bs_customer**
- PK customerID
- firstName
- lastName
- paymentMethod

# Data Definition Queries

DROP TABLE IF EXISTS `bs_company`;

CREATE TABLE `bs_company` (
 `companyID` int(11) NOT NULL AUTO_INCREMENT,
 `companyName` varchar(255) NOT NULL,
 `address` varchar(255) NOT NULL,
 `city` varchar(255) NOT NULL,
 `state` varchar(255) NOT NULL,
 `zipcode` int(11) NOT NULL,

```sql
  `country` varchar(255) NOT NULL,
  PRIMARY KEY (`companyID`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;

INSERT INTO `bs_company` VALUES (1, 'Big Bookstore', '555 Ocean Street', 'Santa Cruz', 'CA', 95064,
'U.S.' );

DROP TABLE IF EXISTS `bs_store`;

CREATE TABLE `bs_store` (
  `storeID` int(11) NOT NULL AUTO_INCREMENT,
  `address` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `state` varchar(255) NOT NULL,
  `zipcode` int(11) NOT NULL,
  `country` varchar(255) NOT NULL,
  `companyID` int(11),
  PRIMARY KEY (`storeID`),
  FOREIGN KEY (`companyID`) REFERENCES `bs_company` (`companyID`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;


INSERT INTO `bs_store` VALUES (1, '3232 Whistle Lane', 'Felton', 'CA', 95066, 'U.S.', 1 ), (2, '9501 Smiley
Street', 'Boulder', 'CO', 45555, 'U.S.', 1 ), (3, '1000 Spring Street', 'Nowheresville', 'Maine', 01212, 'U.S.',
1), (4, '5454 Wetwilly Way', 'Lansings', 'MI', 64646, 'U.S.', 1 ), (5, '45 Main Street', 'Lansings', 'MI', 64646,
'U.S.', 1 );


DROP TABLE IF EXISTS `bs_employee`;
CREATE TABLE `bs_employee` (
  `employeeID` int(11) NOT NULL AUTO_INCREMENT,
  `firstName` varchar(255) NOT NULL,
  `lastName` varchar(255) NOT NULL,
  `sID` int(11),
  `position` varchar(255) NOT NULL,
  PRIMARY KEY (`employeeID`),
  FOREIGN KEY (`storeID`) REFERENCES `bs_store` (`storeID`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

INSERT INTO `bs_employee` VALUES (1,'Mick','Faber', 3,
'Cashier'),(2,'Josh','Sagan',3,'Stocker'),(3,'Barbara','McCullers',3, 'Barista'),(4,'Tony','Huxley',4,
'Cashier'),(5,'Larry','Irving',3, 'Cashier'),(6,'Lorainne','Dickinson',4, 'Supervisor II'),(7,'Darla','Oates',1,
'Supervisor III'),(8,'Patricia','Conroy',1,'Barista'),(9,'Milly','Melville', 2, 'Stocker'), (10,'Donna','Zamyatin',
1, 'Cashier'),(11,'Tina','Hemingway',1,'Cashier'),(12,'Wanda','Faulkner',5,
'Cashier'),(13,'Amanda','Asimov',2, 'Supervisor II'),(14,'Vince','Woolf',2, 'Cashier'),(15,'Flo','Fitzgerald',1,
```

'Supervisor II'), (16,'Darla','Kesey',1, 'Cashier '),(17,'Freddy','Gaiman',1,'Cashier'),(18,'Josh','Kafka', 2, 'Director');


DROP TABLE IF EXISTS `bs_customer`;
CREATE TABLE `bs_customer` (
 `customerID` int(11) NOT NULL AUTO_INCREMENT,
 `firstName` varchar (255) NOT NULL,
 `lastName` varchar (255) NOT NULL,
 `paymentMethod` varchar (255) NOT NULL,
 PRIMARY KEY (`customerID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


INSERT INTO `bs_customer` VALUES (1, 'Jane', 'Eyre', 'Visa'),(2, 'Isserley', 'Vodsel', 'Visa' ), (3, 'Phoebe', 'Caulfield', 'Cash'), (4, 'Gregor', 'Samsa', 'Visa'), (5, 'John', 'Singer', 'Cash'), (6, 'Mick', 'Kelly', 'Cash'), (7, 'Joseph', 'K', 'check') ;

DROP TABLE IF EXISTS `bs_tran`;

CREATE TABLE `bs_tran` (
 `tranID` int(11) AUTO_INCREMENT,
 `cID` int(11),
 `eID` int(11),
 PRIMARY KEY (`tranID`),
 FOREIGN KEY (`cID`) REFERENCES `bs_customer` (`customerID`) ON DELETE CASCADE,
 FOREIGN KEY (`eID`) REFERENCES `bs_employee` (`employeeID`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

INSERT INTO `bs_tran` VALUES (1, 2, 4), (2, 1, 3), (3, 1, 1);

DROP TABLE IF EXISTS `bs_tranD`;
CREATE TABLE `bs_tranD` (
 `pID` int(11) AUTO_INCREMENT,
 `tID` int(11) NOT NULL,
 `author` VARCHAR(255) NOT NULL,
 `title` VARCHAR(255) NOT NULL,
 PRIMARY KEY (`pID`),
 FOREIGN KEY (`tID`) REFERENCES `bs_tran` (`tranID`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

INSERT INTO `bs_tranD` VALUES (1, 1, 2), (2, 1, 2), (3, 2, 1);

## Data Manipulation Queries

"INSERT INTO bs_employee (`firstName`,`lastName`,`storeID`,`position`) VALUES (?,?,?,?)"

DELETE FROM [table] WHERE employeeID = [form specified]

SELECT * FROM [table]

DELETE FROM [table] WHERE employeeID = [form specified]

INSERT INTO bs_employee (`firstName`,`lastName`,`storeID`,`position`) VALUES (?,?,?,?)

UPDATE bs_employee SET firstName=?, lastName=?, storeID=?, position=? WHERE employeeID=?

SELECT * FROM  [user specified table] WHERE [user specified table].[attribute] = [user input name]

SELECT [table1].[column1],[table2].[column2]
FROM [table1]
FULL JOIN [table2]
ON [table1].[userdefined] = [table2].[userdefined];


///MANY TO MANY RELATIONSHIPs///
SELECT * FROM `bs_customer` AS cust
INNER JOIN bs_tran AS tran ON tran.cID = cust.customerID
INNER JOIN bs_employee AS emp ON emp.employeeID = tran.eID
WHERE emp.[user defined] = '[user entry]';

SELECT * FROM bs_employee AS emp
INNER JOIN bs_tran AS tran ON tran.eID = emp.employeeID
INNER JOIN bs_customer AS cust ON cust.customerID = tran.cID
WHERE cust.[user defined] = 'Eyre';


## Outline


The Bookstore Database holds information for a chain of bookstores.  The 6 entities are company, store, employee, customer, transaction, and transaction details.  This project examines the relationships between these entities.


## Database Outline

The first entity to examine is the bookstore company (represented as bs_company). While the company has many stores, a store only has this one company. In order for this company to exist, it must have stores. Basic location information (mailing address parts) are included as attributes.

The next entity, the bookstore store (represented as bs_store), has a many to one relationship with the company. A store must have a company to exist, and only one company. Basic location information and a storeID number make up its attributes.

The next entity, bookstore employee(represented as bs_employee), has a many to one relationship with the store. While a store has many employees, an employee only works at one store. A store must have employees, and employees must have a store. Its attributes include the name of the individual broken up into first and last attributes. Employee also has ID, position, and sID attributes. sID represents the employees store and is a foreign key to the store table's storeID attribute.

The employee entity has two relationships. The first to examine is the relationship with the customer entity. The second relationship with a transaction will be examined at the end of this outline. The customer employee relationship is many to many, as an employee may have many customers and a customer can be helped by many employees. While a customer must have an employee (as being a customer means something is bought), an employee, however, does not need to have a customer (e.g. stockers, store directors, accountants). The customer entity is comprised of four attributes. The first two comprise the first and last names of the customer. The next is a customer ID. The last is the payment method. Were we to examine this attribute further, it could be an entity itself and serve to uniquely identify the customer through payment information. As this is a bookstore database, this was omitted for need of simplicity.

The next entity, the transaction (represented as bs_tran) has a many to one relationship with the customer. While a customer may make many transactions, one transaction may be made by only one customer. This entity represents a receipt that links the employee to the customer. The transaction has three attributes, an auto incremented transaction ID number and two foreign keys. The first foreign key, cID, represents the customer making the transaction and is the customer entities customerID number. Because customers may have anonymity, the customer ID links the customer to transactions. The third attribute, the employee ID, eID, links the employee to the transaction through a foreign key that references the employee ID number.

The final entity we will examine is transaction details (bs_tranD). This entity has four attributes. The first is the pID number. This autoincremented integer represents products. While this could have been an entity in and of itself, for the sake of simplicity is was omitted. Instead, as what commonly appears on a receipt from a bookstore, the next 3 attributes briefly mention a description of the items purchased, namely the title and author of the book. The last attribute is the tID, which is a foreign key that references the transaction ID number, thus linking various books to one transaction. While a transaction may have many details (various books and products), the transaction detail has only one transaction. Both are necessary for one another to exist.