```
CREATE DATABASE IF NOT EXISTS LibraryDB_Final;
USE LibraryDB_Final;

DROP TABLE IF EXISTS Library;
CREATE TABLE Library (
    BookId INT PRIMARY KEY,
    BookName VARCHAR(100),
    Author VARCHAR(100),
    Price INT
);

INSERT INTO Library VALUES
(1,'C Programming','Dennis Ritchie',500),
(2,'DBMS Concepts','Korth',650),
(3,'Operating Systems','Galvin',700);

DROP TABLE IF EXISTS Library_Audit;
CREATE TABLE Library_Audit (
    AuditId INT AUTO_INCREMENT PRIMARY KEY,
    ActionType VARCHAR(30),
    Old_BookId INT,
    Old_BookName VARCHAR(100),
    Old_Author VARCHAR(100),
    Old_Price INT,
    TimeStampValue TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);


DELIMITER $$

DROP TRIGGER IF EXISTS before_library_insert $$
CREATE TRIGGER before_library_insert
BEFORE INSERT ON Library
FOR EACH ROW
BEGIN
   IF NEW.Price < 0 THEN
      SIGNAL SQLSTATE '45001'
      SET MESSAGE_TEXT = 'ERROR: Price cannot be negative!';
   END IF;
END $$


DROP TRIGGER IF EXISTS after_library_insert $$
CREATE TRIGGER after_library_insert
AFTER INSERT ON Library
FOR EACH ROW
BEGIN
   INSERT INTO Library_Audit(ActionType, Old_BookId, Old_BookName, Old_Author, Old_Price)
   VALUES ('AFTER INSERT', NULL, NULL, NULL, NULL);
END $$
```

```
DROP TRIGGER IF EXISTS before_library_update $$
CREATE TRIGGER before_library_update
BEFORE UPDATE ON Library
FOR EACH ROW
BEGIN
  IF NEW.Price < 0 THEN
    SIGNAL SQLSTATE '45002'
    SET MESSAGE_TEXT = 'ERROR: Updated price cannot be negative!';
  END IF;

  INSERT INTO Library_Audit(ActionType, Old_BookId, Old_BookName, Old_Author, Old_Price)
  VALUES ('BEFORE UPDATE', OLD.BookId, OLD.BookName, OLD.Author, OLD.Price);
END $$


DROP TRIGGER IF EXISTS after_library_update $$
CREATE TRIGGER after_library_update
AFTER UPDATE ON Library
FOR EACH ROW
BEGIN
  INSERT INTO Library_Audit(ActionType, Old_BookId, Old_BookName, Old_Author, Old_Price)
  VALUES ('AFTER UPDATE', OLD.BookId, OLD.BookName, OLD.Author, OLD.Price);
END $$
```

```
DROP TRIGGER IF EXISTS before_library_delete $$
CREATE TRIGGER before_library_delete
BEFORE DELETE ON Library
FOR EACH ROW
BEGIN
   IF OLD.BookId = 1 THEN
      SIGNAL SQLSTATE '45003'
      SET MESSAGE_TEXT = 'ERROR: BookId 1 cannot be deleted!';
   END IF;

   INSERT INTO Library_Audit(ActionType, Old_BookId, Old_BookName, Old_Author, Old_Price)
   VALUES ('BEFORE DELETE', OLD.BookId, OLD.BookName, OLD.Author, OLD.Price);
END $$


DROP TRIGGER IF EXISTS after_library_delete $$
CREATE TRIGGER after_library_delete
AFTER DELETE ON Library
FOR EACH ROW
BEGIN
   INSERT INTO Library_Audit(ActionType, Old_BookId, Old_BookName, Old_Author, Old_Price)
   VALUES ('AFTER DELETE', OLD.BookId, OLD.BookName, OLD.Author, OLD.Price);
END $$

DELIMITER ;
```