

# **BOOK STORE MANAGEMENT DBMS-PROJECT**

```
CREATE TABLE manager (  
    manager_id INTEGER PRIMARY KEY,  
    manager_name VARCHAR(100) NOT NULL  
);  
CREATE TABLE books (  
    book_id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    manager_id INTEGER NOT NULL,  
    title VARCHAR(100) NOT NULL,  
    author VARCHAR(100) NOT NULL,  
    publisher VARCHAR(100) NOT NULL,  
    price NUMBER(10,2) NOT NULL,  
    stock INTEGER NOT NULL,  
    FOREIGN KEY (manager_id) REFERENCES manager (manager_id)  
);  
CREATE TABLE customers (  
    customer_id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    phone VARCHAR(20) NOT NULL  
);  
  
CREATE TABLE orders (  
    order_id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    customer_id INTEGER NOT NULL,  
    order_date DATE DEFAULT SYSDATE,  
    FOREIGN KEY (customer_id) REFERENCES customers (customer_id)  
);  
  
CREATE TABLE order_items (  
    order_id INTEGER NOT NULL,  
    book_id INTEGER NOT NULL,  
    quantity INTEGER NOT NULL,  
    PRIMARY KEY (order_id, book_id),  
    FOREIGN KEY (order_id) REFERENCES orders (order_id),  
    FOREIGN KEY (book_id) REFERENCES books (book_id)  
);  
  
INSERT INTO manager VALUES('24','Ram');  
INSERT INTO manager VALUES('27','Sham');  
INSERT INTO manager VALUES('10','Balram');
```

MANAGER_ID	MANAGER_NAME
24	Ram
27	Sham
10	Balram

```
INSERT INTO books VALUES('237','24','Magic Tree','Abhinav','Sushant','1500','20');
INSERT INTO books VALUES('452','27','Winter Fairy','Shubham','Manoj','2000','10');
INSERT INTO books VALUES('480','10','Enchanted','Aryan','Vikas','2500','8');
```

```
INSERT INTO customers VALUES('1','Pranjali','pranjali15@gmail.com','9834652376');
INSERT INTO customers VALUES('2','Jasmine','jasmine17@gmail.com','7640271548');
INSERT INTO customers VALUES('3','Nandini','nandini5@gmail.com','9843725476');
```

```
INSERT INTO orders VALUES('39','1','15-JAN-2022');
INSERT INTO orders VALUES('72','2','27-APR-2022');
INSERT INTO orders VALUES('58','3','23-MAY-2022');
```

```
INSERT INTO order_items VALUES('39','237','7');
INSERT INTO order_items VALUES('72','452','5');
INSERT INTO order_items VALUES('58','480','3');
```

```
select * from manager;
select * from books;
select * from customers;
select * from orders;
select * from order_items;
```

```
Table created.
1 row(s) inserted.
Table created.
1 row(s) inserted.
Table created.
1 row(s) inserted.
Table created.
1 row(s) inserted.
Table created.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
```

BOOK_ID	MANAGER_ID	TITLE	AUTHOR	PUBLISHER	PRICE	STOCK
237	24	Magic Tree	Abhinav	Sushant	1500	20
452	27	Winter Fairy	Shubham	Manoj	2000	10
480	10	Enchanted	Aryan	Vikas	2500	8

ORDER_ID	BOOK_ID	QUANTITY
39	237	7
72	452	5
58	480	3

```
CREATE OR REPLACE FUNCTION get_order_total (
  p_order_id IN INTEGER
```

```

) RETURN NUMBER IS
    v_total NUMBER := 0;
BEGIN
    SELECT SUM(quantity * price)
    INTO v_total
    FROM order_items oi
    JOIN books b ON oi.book_id = b.book_id
    WHERE oi.order_id = p_order_id;
    RETURN v_total;
END;
/

```

```

CREATE OR REPLACE TRIGGER update_stock
AFTER INSERT ON order_items
FOR EACH ROW
DECLARE
    v_stock books.stock%TYPE;
BEGIN

    SELECT stock INTO v_stock FROM books WHERE book_id = :NEW.book_id;

    UPDATE books SET stock = v_stock - :NEW.quantity WHERE book_id = :NEW.book_id;
END;
/

```

```

CREATE OR REPLACE TRIGGER prevent_negative_price
BEFORE INSERT,UPDATE ON books
FOR EACH ROW
BEGIN
    IF :NEW.price < 0 THEN
        RAISE_APPLICATION_ERROR(-20007, 'Price cannot be negative.');
```

```

    END IF;

```

```

END;
/

```

```

CREATE OR REPLACE PROCEDURE update_book_stock (
    p_book_id IN INTEGER,
    p_stock IN INTEGER
) IS
BEGIN
    UPDATE books
    SET stock = p_stock
    WHERE book_id = p_book_id;

```

```
    COMMIT;
END;
/
```

```
INSERT INTO orders (customer_id)
VALUES (1);
INSERT INTO order_items (order_id, book_id, quantity)
VALUES (1, 1, 2);
```

```
SELECT get_order_total(39) FROM dual;
```

```
BEGIN
    update_book_stock(237, 150);
END;
/
```

```
select * from books;
```

```
DECLARE
```

```
    CURSOR c1 IS
```

```
        SELECT title, author
```

```
        FROM books;
```

```
    v_title books.title%TYPE;
```

```
    v_author books.author%TYPE;
```

```
BEGIN
```

```
    OPEN c1;
```

```
    LOOP
```

```
        FETCH c1 INTO v_title, v_author;
```

```
        EXIT WHEN c1%NOTFOUND;
```

```
        DBMS_OUTPUT.PUT_LINE('Title: ' || v_title || ', Author: ' || v_author);
```

```
    END LOOP;
```

```
    CLOSE c1;
```

```
END;
```

```
/
```

```
*/
```

Function created.

Trigger created.

Trigger created.

Procedure created.

1 row(s) inserted.

Statement processed.

Title: Magic Tree, Author: Abhinav

Title: Winter Fairy, Author: Shubham

Title: Enchanted, Author: Aryan

GET\_ORDER\_TOTAL(39)

10500

Download CSV

Statement processed.

BOOK_ID	MANAGER_ID	TITLE	AUTHOR	PUBLISHER	PRICE	STOCK
237	24	Magic Tree	Abhinav	Sushant	1500	150
452	27	Winter Fairy	Shubham	Manoj	2000	10
480	10	Enchanted	Aryan	Vikas	2500	8