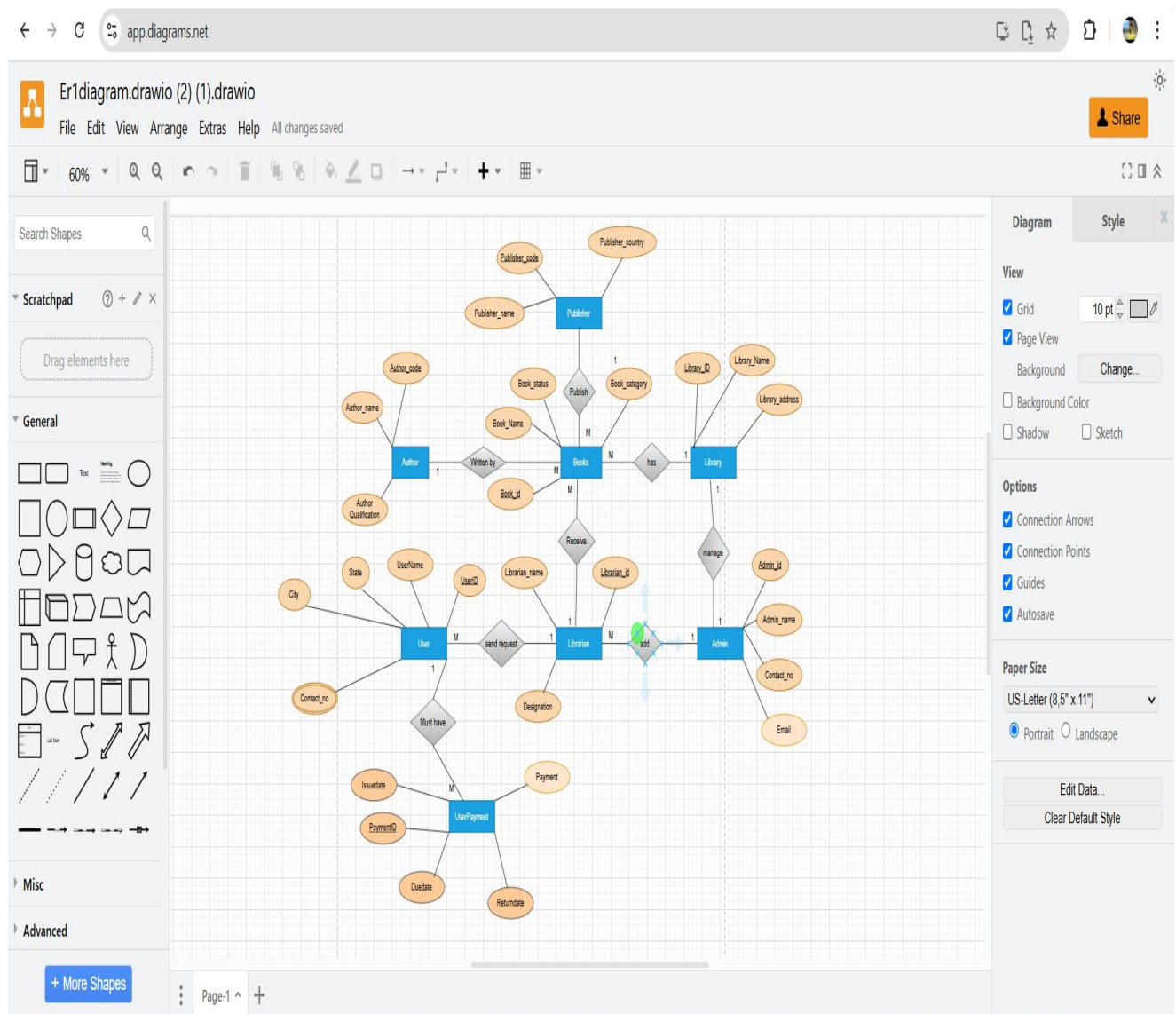
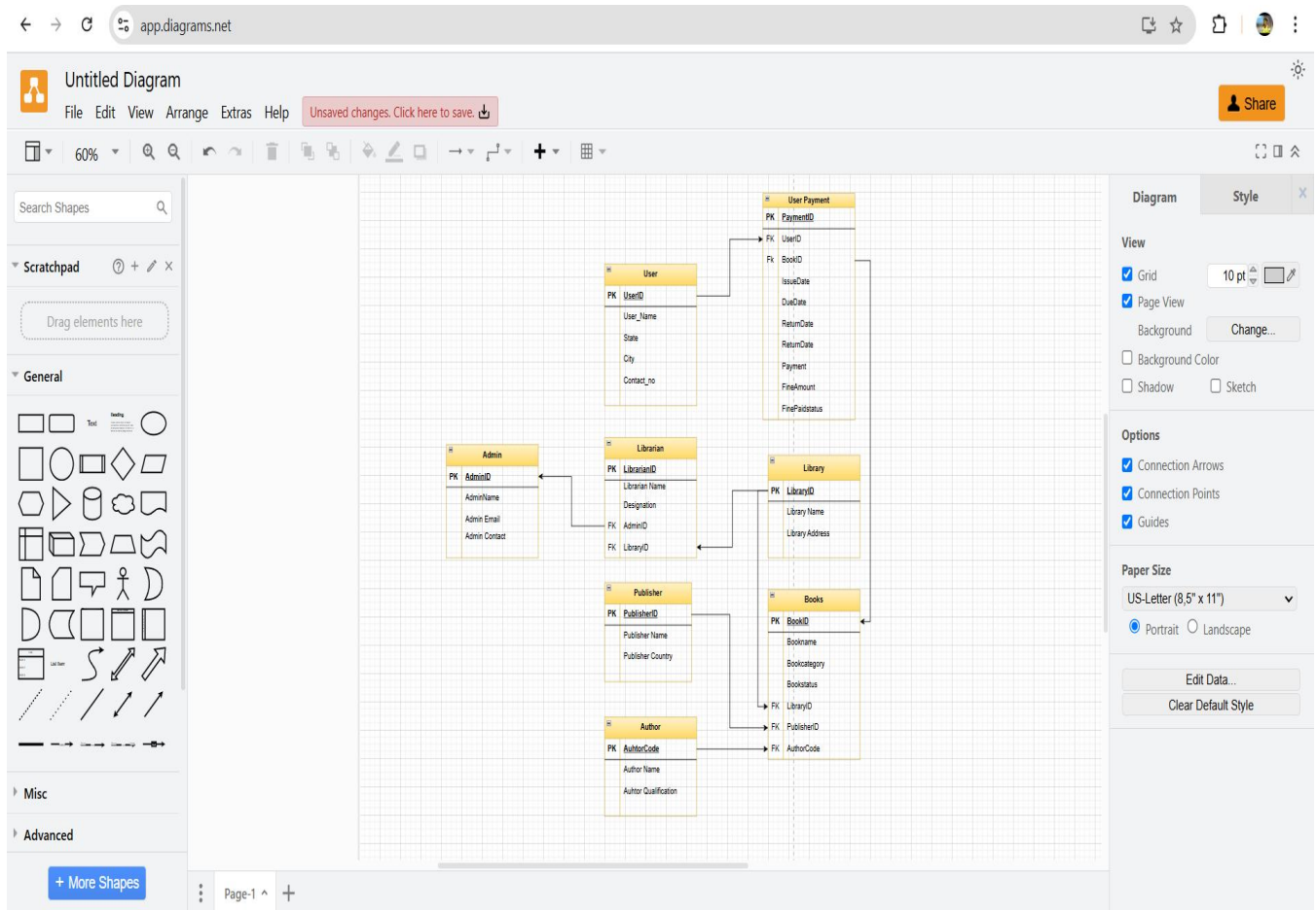


Library Management System





Live SQL

Feedback Help sairajkumar2002@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
259 UPDATE Books
260 SET BookStatus = 'Available'
261 WHERE BookID = :NEW.BookID;
262 END IF;
263 END;
264 /
265 -- View to show user payment history and fine status
266 CREATE OR REPLACE VIEW UserBookHistory AS
267 SELECT u.UserID, u.UserName, b.BookID, b.BookName, up.IssueDate, up.DueDate, up.ReturnDate,
268        NVL(up.FineAmount, 0) AS FineAmount, up.FinePaidStatus
269 FROM UserTable u
```

Package created.

Package Body created.

Trigger created.

Trigger created.

View created.

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym

Built with using Oracle APEX - Privacy - Terms of Use

Live SQL

Feedback Help sairajkumar2002@gmail.com

SQL Worksheet

Clear Find Actions Save Run


```
275 -- Add a new book
276 BEGIN
277     LibraryPackage.AddBook(
278         BookID => BookSeq.NEXTVAL,
279         BookName => 'The Catcher in the Rye',
280         BookCategory => 'Fiction',
281         BookStatus => 'Available',
282         LibraryID => 1,
283         PublisherID => 201,
284         AuthorCode => 1
285     );
```

BOOKID	BOOKNAME	BOOKCATEGORY	BOOKSTATUS	LIBRARYID	PUBLISHERID	AUTHORCODE
104	The Catcher in the Rye	Fiction	Available	1	201	1
101	Adventures of Huckleberry Finn	Fiction	Available	1	201	1
102	Pride and Prejudice	Romance	Issued	2	202	2
103	Bahubali	Historic	Issued	2	204	3

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym

Built with using Oracle APEX - Privacy - Terms of Use

← → ↻ 🌐 livesqloracle.com/ords/f?p=590:1:8947270341619::RP::

≡  Live SQL

Feedback Help 🧑🏻 sairajkumar2002@gmail.com ⚙️

SQL Worksheet

Clear Find Actions Save Run


```
289 -- Update book status to 'Available'
290 BEGIN
291     LibraryPackage.UpdateBookStatus(
292         BookID => 101,
293         NewStatus => 'Available'
294     );
295 END;
```

BOOKID	BOOKNAME	BOOKCATEGORY	BOOKSTATUS	LIBRARYID	PUBLISHERID	AUTHORCODE
104	The Catcher in the Rye	Fiction	Available	1	201	1
101	Adventures of Huckleberry Finn	Fiction	Available	1	201	1
102	Pride and Prejudice	Romance	Available	2	202	2
103	Bahubali	Historic	Available	2	204	3

Download CSV

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↻ 🌐 livesqloracle.com/ords/f?p=590:1:8947270341619::RP::

≡  Live SQL

Feedback Help 🧑🏻 sairajkumar2002@gmail.com ⚙️

SQL Worksheet

Clear Find Actions Save Run

```
297 -- Add a new user payment record
298 BEGIN
299     LibraryPackage.AddUserPayment(
300         UserID => 2,
301         BookID => 101,
302         IssueDate => TO_DATE('2024-11-01', 'YYYY-MM-DD'),
303         DueDate => TO_DATE('2024-11-15', 'YYYY-MM-DD')
304     );
305 END;
```

Statement processed.

PAYMENTID	USERID	BOOKID	ISSUEDATE	DUEDATE	RETURNDATE	PAYMENT	FINEAMOUNT	FINEPAIDSTATUS
1001	2	101	01-NOV-24	15-NOV-24	-	0	0	Unpaid


Download CSV

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↻

livesqloracle.com/ords/?p=590:1:8947270341619::RP::

🔍 ⭐ 🗑️ 🌐

≡  Live SQL

Feedback ⓘ Help 👤 sairajkumar2002@gmail.com ⌵ 🌙

SQL Worksheet

🗑️ Clear 🔍 Find ⌵ Actions 💾 Save ▶️ Run

```
308 -- Calculate the fine for a user and book
309 DECLARE
310     FineAmount NUMBER;
311 BEGIN
312     FineAmount := LibraryPackage.CalculateFine(
313         UserID => 2,
314         BookID => 101
315     );
316     DBMS_OUTPUT.PUT_LINE('Fine Amount: ' || FineAmount);
317 END;
318 /
```


Statement processed.
Fine Amount: 90

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation · Ask Tom · Dev Gym
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

← → ↻

livesqloracle.com/ords/?p=590:1:8947270341619::RP::

🔍 ⭐ 🗑️ 🌐

≡  Live SQL

Feedback ⓘ Help 👤 sairajkumar2002@gmail.com ⌵ 🌙

SQL Worksheet

🗑️ Clear 🔍 Find ⌵ Actions 💾 Save ▶️ Run

```
319 -- Return a book and calculate fine
320 DECLARE
321     FineAmount NUMBER;
322 BEGIN
323     FineAmount := LibraryPackage.ReturnBook(
324         UserID => 2,
325         BookID => 101,
326         ReturnDate => TO_DATE('2024-12-05', 'YYYY-MM-DD')
327     );
328     DBMS_OUTPUT.PUT_LINE('Fine Amount on Return: ' || FineAmount);
329 END;
330 /
331 select * from userpaymenttable where userid=2
```


Statement processed.
Fine Amount on Return: 90

PAYMENTID	USERID	BOOKID	ISSUEDATE	DUEDATE	RETURNDATE	PAYMENT	FINEAMOUNT	FINEPAIDSTATUS
1001	2	101	01-NOV-24	15-NOV-24	-	0	0	Paid

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation · Ask Tom · Dev Gym
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

← → ↺ 🌐 livesqloracle.com/ords/f?p=590:1:8947270341619::RP::

🔍 ⚙️ 🌙

≡  Live SQL

💬 Feedback ⓘ Help 👤 sairajkumar2002@gmail.com ⌵ ⚙️

SQL Worksheet

🗑️ Clear 🔍 Find ⌵ Actions 💾 Save ▶️ Run

343 -- Query to display user book history with fine and payment status
344 SELECT * FROM UserBookHistory;
345
346 -- Insert a new payment record to trigger the BookIssueTrigger
347 BEGIN
348 INSERT INTO UserPaymentTable (PaymentID, UserID, BookID, IssueDate, DueDate, Payment)
349 VALUES (PaymentSeq.NEXTVAL, 4, 103, SYSDATE, SYSDATE + 10, 0);
350 END;
351 /
352 -- Query the Books table to see if the BookStatus has been updated to 'Issued' by the trigger
353 SELECT * FROM Books WHERE BookID = 103;
354
355 -- Execute a block to update the ReturnDate for a book

Statement processed.

BOOKID	BOOKNAME	BOOKCATEGORY	BOOKSTATUS	LIBRARYID	PUBLISHERID	AUTHORCODE
103	Bahubali	Historic	Issued	2	204	3


Download CSV

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym

Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↺ 🌐 livesqlorade.com/ords/f?p=590:1.8947270341619::RP::

🔍 ⚙️ 🌙 🧑🏿

≡  Live SQL

💬 Feedback ⓘ Help 👤 sairajkumar2002@gmail.com ⌵ 🌙

SQL Worksheet

🗑️ Clear 🔍 Find ⌵ Actions 📄 Save ▶️ Run

```
355 -- Execute a block to update the ReturnDate for a book
356 DECLARE
357     FineAmount NUMBER;
358 BEGIN
359
```

Statement processed.
Fine Amount: 0

BOOKID	BOOKNAME	BOOKCATEGORY	BOOKSTATUS	LIBRARYID	PUBLISHERID	AUTHORCODE
101	Adventures of Huckleberry Finn	Fiction	Available	1	201	1


Download CSV

PAYMENTID	USERID	BOOKID	ISSUEDATE	DUEDATE	RETURNDATE	PAYMENT	FINEAMOUNT	FINEPAIDSTATUS
1001	2	101	01-NOV-24	15-NOV-24	05-DEC-24	0	0	Paid

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation · Ask Tom · Dev Gym
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

← → ↺ 🌐 livesqlorade.com/ords/f?p=590:1.8947270341619::RP::

🔍 ⚙️ 🌙 🧑🏿

≡  Live SQL

💬 Feedback ⓘ Help 👤 sairajkumar2002@gmail.com ⌵ 🌙

SQL Worksheet

🗑️ Clear 🔍 Find ⌵ Actions 📄 Save ▶️ Run

```
369 END;
370 /
371
372 -- Query to verify if the trigger updated the BookStatus to 'Available'
373 SELECT * FROM Books WHERE BookID = 101;
374 select * from userpaymenttable
375 -- Query to display user book history with fine and payment status
376 SELECT * FROM UserBookHistory;
377
378
```

USERID	USERNAME	BOOKID	BOOKNAME	ISSUEDATE	DUEDATE	RETURNDATE	FINEAMOUNT	FINEPAIDSTATUS
2	Chris Evans	101	Adventures of Huckleberry Finn	01-NOV-24	15-NOV-24	05-DEC-24	0	Paid
4	John snow	103	Bahubali	24-NOV-24	04-DEC-24	-	0	Paid

Download CSV

2 rows selected.

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation · Ask Tom · Dev Gym
Built with ❤️ using Oracle APEX · Privacy · Terms of Use

← → ↻ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 👤 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⚙️ Actions 💾 Save 🏃 Run

```
1 -- Library Table
2 CREATE TABLE Library (
3     LibraryID INT PRIMARY KEY,
4     LibraryName VARCHAR(100),
5     LibraryAddress VARCHAR(200)
6 );
7
8 -- Admin Table
9 CREATE TABLE Admin (
10     AdminID INT PRIMARY KEY,
11     AdminName VARCHAR(100),
12     AdminEmail VARCHAR(100),
13     AdminContact VARCHAR(15)
14 );
15
16 -- Librarian Table
17 CREATE TABLE Librarian (
18     LibrarianID INT PRIMARY KEY,
19     LibrarianName VARCHAR(100),
20     Designation VARCHAR(50),
21     AdminID INT,
22     LibraryID INT,
23     FOREIGN KEY (AdminID) REFERENCES Admin(AdminID),
24     FOREIGN KEY (LibraryID) REFERENCES Library(LibraryID)
25 );
26
27 -- Publisher Table
28 CREATE TABLE Publisher (
29     PublisherID INT PRIMARY KEY,
30     PublisherName VARCHAR(100),
31     PublisherCountry VARCHAR(50)
32 );
33
34 -- Author Table
35 CREATE TABLE Author (
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↻ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 👤 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⚙️ Actions 💾 Save 🏃 Run

```
34 -- Author Table
35 CREATE TABLE Author (
36     AuthorCode INT PRIMARY KEY,
37     AuthorName VARCHAR(100),
38     AuthorQualification VARCHAR(50)
39 );
40
41 -- Books Table
42 CREATE TABLE Books (
43     BookID INT PRIMARY KEY,
44     BookName VARCHAR(100),
45     BookCategory VARCHAR(50),
46     BookStatus VARCHAR(20),
47     LibraryID INT,
48     PublisherID INT,
49     AuthorCode INT,
50     FOREIGN KEY (LibraryID) REFERENCES Library(LibraryID),
51     FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID),
52     FOREIGN KEY (AuthorCode) REFERENCES Author(AuthorCode)
53 );
54
55 -- User Table
56 CREATE TABLE UserTable (
57     UserID INT PRIMARY KEY,
58     UserName VARCHAR(100),
59     State VARCHAR(50),
60     City VARCHAR(50),
61     ContactNo VARCHAR(15)
62 );
63
64 -- UserPayment Table (with added FineAmount and FinePaidStatus attributes)
65 CREATE TABLE UserPaymentTable (
66     PaymentID INT PRIMARY KEY,
67     UserID INT,
68     BookID INT,
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↺ 🌐 livesql.oracle.com/ords/?p=590:1:1267845407880::RP::

🔍 🔖 ⌂ 👤

🔌 Live SQL

🗉 Feedback 🛠️ Help 👤 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⌵ Actions 📄 Save 🏃 Run

```
132 -- Sequence for UserPaymentTable
134 CREATE SEQUENCE PaymentSeq START WITH 1001 INCREMENT BY 1;
135
136 -- Sequence for Publisher
137 CREATE SEQUENCE PublisherSeq START WITH 205 INCREMENT BY 1;
138
139 -- Package creation for Library operations
140 CREATE OR REPLACE PACKAGE libraryPackage IS
141     PROCEDURE AddBook(BookID IN INT, BookName IN VARCHAR2, BookCategory IN VARCHAR2,
142         BookStatus IN VARCHAR2, LibraryID IN INT,
143         PublisherID IN INT, AuthorCode IN INT);
144
145     PROCEDURE UpdateBookStatus(BookID IN INT, NewStatus IN VARCHAR2);
146
147     PROCEDURE AddUserPayment(UserID IN INT, BookID IN INT, IssueDate IN DATE, DueDate IN DATE);
148
149     FUNCTION CalculateFine(UserID IN INT, BookID IN INT) RETURN NUMBER;
150
151     FUNCTION ReturnBook(UserID IN INT, BookID IN INT, ReturnDate IN DATE) RETURN NUMBER;
152 END libraryPackage;
153 /
154 CREATE OR REPLACE PACKAGE BODY libraryPackage IS
155     PROCEDURE AddBook(BookID IN INT, BookName IN VARCHAR2, BookCategory IN VARCHAR2,
156         BookStatus IN VARCHAR2, LibraryID IN INT,
157         PublisherID IN INT, AuthorCode IN INT) IS
158     BEGIN
159         INSERT INTO Books (BookID, BookName, BookCategory, BookStatus, LibraryID, PublisherID, AuthorCode)
160         VALUES (BookID, BookName, BookCategory, BookStatus, LibraryID, PublisherID, AuthorCode);
161     END AddBook;
162
163     PROCEDURE UpdateBookStatus(BookID IN INT, NewStatus IN VARCHAR2) IS
164     BEGIN
165         UPDATE Books
166         SET BookStatus = NewStatus
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↺ 🌐 livesql.oracle.com/ords/?p=590:1:1267845407880::RP::

🔍 🔖 ⌂ 👤

🔌 Live SQL

🗉 Feedback 🛠️ Help 👤 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⌵ Actions 📄 Save 🏃 Run

```
155 PROCEDURE AddBook(BookID IN INT, BookName IN VARCHAR2, BookCategory IN VARCHAR2,
156     BookStatus IN VARCHAR2, LibraryID IN INT,
157     PublisherID IN INT, AuthorCode IN INT) IS
158 BEGIN
159     INSERT INTO Books (BookID, BookName, BookCategory, BookStatus, LibraryID, PublisherID, AuthorCode)
160     VALUES (BookID, BookName, BookCategory, BookStatus, LibraryID, PublisherID, AuthorCode);
161 END AddBook;
162
163 PROCEDURE UpdateBookStatus(BookID IN INT, NewStatus IN VARCHAR2) IS
164 BEGIN
165     UPDATE Books
166     SET BookStatus = NewStatus
167     WHERE BookID = BookID;
168 END UpdateBookStatus;
169
170 PROCEDURE AddUserPayment(UserID IN INT, BookID IN INT, IssueDate IN DATE, DueDate IN DATE) IS
171 BEGIN
172     INSERT INTO UserPaymentTable (PaymentID, UserID, BookID, IssueDate, DueDate, Payment, FineAmount, FinePaidStatus)
173     VALUES (PaymentSeq.NEXTVAL, UserID, BookID, IssueDate, DueDate, 0.00, 0.00, 'Unpaid');
174 END AddUserPayment;
175
176 FUNCTION CalculateFine(UserID INT, BookID INT) RETURN NUMBER IS
177 fine NUMBER := 0;
178 dueDate DATE;
179 fineRate NUMBER := 10; -- Fine per day is now 10 rupees
180 daysLate NUMBER;
181 BEGIN
182 -- Fetch the first due date for the given UserID and BookID where ReturnDate is NULL
183 SELECT DueDate
184 INTO dueDate
185 FROM (SELECT DueDate
186     FROM UserPaymentTable
187     WHERE UserID = UserID
188     AND BookID = BookID
189     AND ReturnDate IS NULL
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↺ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 🧑 sarajkumar2002@gmail.com 🌙

SQL Worksheet

Clear Find Actions Save Run

```
188 AND BookID = BookID
189 AND ReturnDate IS NULL
190 ORDER BY PaymentID DESC)
191 WHERE ROWNUM = 1;
192
193 -- Calculate fine if the provided return date exceeds the due date
194 IF SYSDATE > dueDate THEN
195   dayslate := TRUNC(SYSDATE) - TRUNC(dueDate);
196   fine := dayslate * fineRate;
197 END IF;
198
199 RETURN fine;
200
201 EXCEPTION
202 WHEN NO_DATA_FOUND THEN
203   RETURN 0; -- No fine if no record found
204 END CalculateFine;
205
206 FUNCTION ReturnBook(userID IN INT, BookID IN INT, ReturnDate IN DATE) RETURN NUMBER IS
207   fineAmount NUMBER := 0;
208 BEGIN
209   -- Calculate the fine
210   fineAmount := CalculateFine(userID, BookID);
211
212   -- Update ReturnDate, Payment, and PaymentStatus
213   UPDATE UserPaymentTable
214   SET ReturnDate = ReturnDate,
215       Payment = fineAmount,
216       finepaidstatus = CASE
217         WHEN fineAmount = 0 THEN 'Paid'
218         ELSE 'Unpaid'
219       END
220   WHERE UserID = userID
221     AND BookID = BookID
222     AND ReturnDate IS NULL;
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↺ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 🧑 sarajkumar2002@gmail.com 🌙

SQL Worksheet

Clear Find Actions Save Run

```
218 ELSE 'Unpaid'
219 END
220 WHERE UserID = UserID
221 AND BookID = BookID
222 AND ReturnDate IS NULL;
223
224 -- If fine > 0, mark it as paid
225 IF fineAmount > 0 THEN
226   UPDATE UserPaymentTable
227   SET finepaidstatus = 'Paid'
228   WHERE UserID = UserID
229     AND BookID = BookID;
230 END IF;
231
232 RETURN fineAmount;
233 EXCEPTION
234 WHEN OTHERS THEN
235   DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
236 RETURN 0;
237 END ReturnBook;
238 END LibraryPackage;
239 /
240
241
242 -- Trigger to update Book Status when a book is issued
243 CREATE OR REPLACE TRIGGER BookIssueTrigger
244 AFTER INSERT ON UserPaymentTable
245 FOR EACH ROW
246 BEGIN
247   UPDATE Books
248   SET BookStatus = 'Issued'
249   WHERE BookID = :NEW.BookID;
250 END;
251 /
252
253 -- Trigger to update Book Status when a book is returned
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↻ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 🧑🏻 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⚙️ Actions 📄 Save 🏃 Run

```
249 WHERE BookID = :NEW.BookID;
250 END;
251 /
252
253 -- Trigger to update Book Status when a book is returned
254 CREATE OR REPLACE TRIGGER BookReturnTrigger
255 AFTER UPDATE ON UserPaymentTable
256 FOR EACH ROW
257 BEGIN
258 IF :NEW.ReturnDate IS NOT NULL THEN
259 UPDATE Books
260 SET BookStatus = 'Available'
261 WHERE BookID = :NEW.BookID;
262 END IF;
263 END;
264 /
265
266 -- View to show user payment history and fine status
267 CREATE OR REPLACE VIEW UserBookHistory AS
268 SELECT u.UserID, u.UserName, b.BookID, b.BookName, up.IssueDate, up.DueDate, up.ReturnDate,
269 NVL(up.FineAmount, 0) AS FineAmount, up.FinePaidStatus
270 FROM UserTable u
271 JOIN UserPaymentTable up ON u.UserID = up.UserID
272 JOIN Books b ON up.BookID = b.BookID;
273 /
274
275 -----Test example
276
277 -- Add a new book
278 BEGIN
279 LibraryPackage.AddBook(
280 BookID => BookSeq.NEXTVAL,
281 BookName => 'The Catcher in the Rye',
282 BookCategory => 'Fiction',
283 BookStatus => 'Available',
284 LibraryID => 1,
285 PublisherID => 201,
286 );
287 /
288
289 select * from books
290 -- Update book status to 'Available'
291 BEGIN
292 LibraryPackage.UpdateBookStatus(
293 BookID => 101,
294 NewStatus => 'Available'
295 );
296 /
297
298 -- Add a new user payment record
299 BEGIN
300 LibraryPackage.AddUserPayment(
301 UserID => 2,
302 BookID => 101,
303 IssueDate => TO_DATE('2024-11-01', 'YYYY-MM-DD'),
304 DueDate => TO_DATE('2024-11-15', 'YYYY-MM-DD')
305 );
306 /
307
308 select * from userpaymenttable
309 -- Calculate the fine for a user and book
310 DECLARE
311 FineAmount NUMBER;
312 BEGIN
313 FineAmount := LibraryPackage.CalculateFine(
314 UserID => 2,
315 BookID => 101
316 );
317 /
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

← → ↻ 🌐 livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help 🧑🏻 sairajkumar2002@gmail.com 🌙

SQL Worksheet

🧹 Clear 🔍 Find ⚙️ Actions 📄 Save 🏃 Run

```
281 BookStatus => 'Available',
282 LibraryID => 1,
283 PublisherID => 201,
284 AuthorCode => 1
285 );
286 END;
287 /
288 select * from books
289 -- Update book status to 'Available'
290 BEGIN
291 LibraryPackage.UpdateBookStatus(
292 BookID => 101,
293 NewStatus => 'Available'
294 );
295 END;
296 /
297
298 -- Add a new user payment record
299 BEGIN
300 LibraryPackage.AddUserPayment(
301 UserID => 2,
302 BookID => 101,
303 IssueDate => TO_DATE('2024-11-01', 'YYYY-MM-DD'),
304 DueDate => TO_DATE('2024-11-15', 'YYYY-MM-DD')
305 );
306 /
307
308 select * from userpaymenttable
309 -- Calculate the fine for a user and book
310 DECLARE
311 FineAmount NUMBER;
312 BEGIN
313 FineAmount := LibraryPackage.CalculateFine(
314 UserID => 2,
315 BookID => 101
316 );
317 /
```

SQL Statement Output

2024 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help sarajkumar2002@gmail.com

SQL Worksheet Clear Find Actions Save Run

```
314      BookID => 101;
315  };
316  DBMS_OUTPUT.PUT_LINE('Fine Amount: ' || FineAmount);
317  END;
318  /
319  -- Return a book and calculate fine
320  DECLARE
321    FineAmount NUMBER;
322  BEGIN
323    FineAmount := LibraryPackage.ReturnBook(
324      UserID => 2,
325      BookID => 101,
326      ReturnDate => TO_DATE('2024-12-05', 'YYYY-MM-DD')
327    );
328    DBMS_OUTPUT.PUT_LINE('Fine Amount on Return: ' || FineAmount);
329  END;
330  /
331  select * from userpaymenttable where userid=2
332  -- Query the UserPaymentTable to see if the return date and payment were updated
333  SELECT * FROM UserPaymentTable WHERE BookID = 101 AND UserID = 2;
334
335  -- Query the Books table to verify the book status change
336  SELECT * FROM Books WHERE BookID = 101;
337
338  -- Query the Books table to verify the book is added
339  SELECT * FROM Books WHERE BookName = 'The Catcher in the Rye';
340
341
342
343  -- Query to display user book history with fine and payment status
344  SELECT * FROM UserBookHistory;
345
346  -- Insert a new payment record to trigger the BookIssueTrigger
347  BEGIN
348    INSERT INTO UserPaymentTable (PaymentID, UserID, BookID, IssueDate, DueDate, Payment)
349    VALUES (PaymentSeq.NEXTVAL, 4, 101, SYSDATE, SYSDATE + 10, 0);
350  END;
351  /
352  -- Query the Books table to see if the BookStatus has been updated to 'Issued' by the trigger
353  SELECT * FROM Books WHERE BookID = 101;
354
355  -- Execute a block to update the ReturnDate for a book
356  DECLARE
357    FineAmount NUMBER;
358  BEGIN
359    -- returning a book by updating the ReturnDate in UserPaymentTable-----
360    UPDATE UserPaymentTable
361    SET ReturnDate = TO_DATE('2024-12-05', 'YYYY-MM-DD')
362    WHERE UserID = 2 AND BookID = 101 AND ReturnDate IS NULL;
363
364    -- Optionally, if you're also calculating the fine, do that too-----
365    FineAmount := LibraryPackage.ReturnBook(2, 101, TO_DATE('2024-12-05', 'YYYY-MM-DD'));
366
367    -- Output fine information (if needed)
368    DBMS_OUTPUT.PUT_LINE('Fine Amount: ' || FineAmount);
369  END;
370  /
371  -- Query to verify if the trigger updated the BookStatus to 'Available'
372  SELECT * FROM Books WHERE BookID = 101;
373  select * from userpaymenttable
374  -- Query to display user book history with fine and payment status
375  SELECT * FROM UserBookHistory;
376
377
```

SQL Statement Output

2024 Oracle - Live SQL 24A.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use

livesql.oracle.com/ords/f?p=590:1:1267845407880::RP::

Live SQL

Feedback Help sarajkumar2002@gmail.com

SQL Worksheet Clear Find Actions Save Run

```
343  -- Query to display user book history with fine and payment status
344  SELECT * FROM UserBookHistory;
345
346  -- Insert a new payment record to trigger the BookIssueTrigger
347  BEGIN
348    INSERT INTO UserPaymentTable (PaymentID, UserID, BookID, IssueDate, DueDate, Payment)
349    VALUES (PaymentSeq.NEXTVAL, 4, 101, SYSDATE, SYSDATE + 10, 0);
350  END;
351  /
352  -- Query the Books table to see if the BookStatus has been updated to 'Issued' by the trigger
353  SELECT * FROM Books WHERE BookID = 101;
354
355  -- Execute a block to update the ReturnDate for a book
356  DECLARE
357    FineAmount NUMBER;
358  BEGIN
359    -- returning a book by updating the ReturnDate in UserPaymentTable-----
360    UPDATE UserPaymentTable
361    SET ReturnDate = TO_DATE('2024-12-05', 'YYYY-MM-DD')
362    WHERE UserID = 2 AND BookID = 101 AND ReturnDate IS NULL;
363
364    -- Optionally, if you're also calculating the fine, do that too-----
365    FineAmount := LibraryPackage.ReturnBook(2, 101, TO_DATE('2024-12-05', 'YYYY-MM-DD'));
366
367    -- Output fine information (if needed)
368    DBMS_OUTPUT.PUT_LINE('Fine Amount: ' || FineAmount);
369  END;
370  /
371  -- Query to verify if the trigger updated the BookStatus to 'Available'
372  SELECT * FROM Books WHERE BookID = 101;
373  select * from userpaymenttable
374  -- Query to display user book history with fine and payment status
375  SELECT * FROM UserBookHistory;
376
377
```

SQL Statement Output

2024 Oracle - Live SQL 24A.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use