



We are going to expand the HR system. The company has a small technical library where employees can check out books. In the past, it was based on an honor system, unfortunately, this did not work too well. We have to design a small database that tracks the books, the employee who currently has books checked out, and the check in transaction.

The company wants to empower its employees by giving them the best resources to do their jobs. Therefore, books can be checked out a maximum of 30 to 32 days (depending on the weekday, see below). For every day a book is past due, a fee of \$2 is assessed.

Here are the steps to create the database tables:

Books table (also review ERD diagram on next page):

ISBN	ISBN number of the book (primary key)
Title	Title of the book
No_of_Copies	Number of copies on hand
Author	The main author

Book Transaction (book_trans) table (also review ERD diagram on next page):

Book_trans_id	Integer column (primary key)
ISBN	Foreign key to books table
Employee_ID	Foreign key to employees table
Checkout_dte	Date a book is checked out
Due_dte	Date a book is due
Return_dte	Date a book is actually returned
PastDue_fees	Past due fees (\$2 per day)

Business rule for due date:

In general, the due date is 30 days from the checkout date. If the due date falls on a Saturday or Sunday then the loan period is 32 days. To test for a weekday, use the to_char function in Oracle with the format of 'D'. For example, to test which weekday November 12, 2007 falls on, use the expression to_char('12-NOV-2007','D'). This function returns a number between 1 and 7. 1 represents Sunday, 2 Monday, ...,7 Saturday.

Business rule for Past Due fees:

The number of days past due (Return date – Due date) times \$2, if not late then 0.

Business rule for checking out books:

An employee is only allowed to check out the same book once per day.

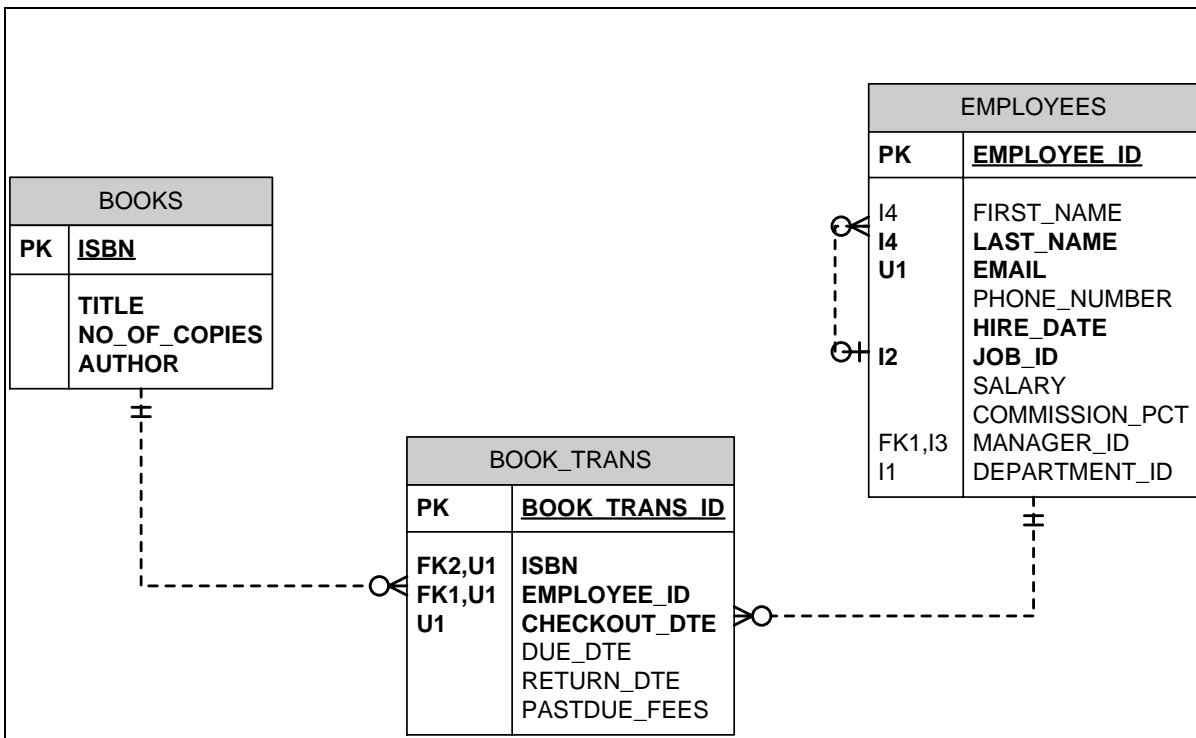


A selection of technical books is shown below:

ISBN	Title	Number of Copies	Main Author
0-07-225790-3	OCP Certification All-in-One Exam Guide	5	John Watson
0-201-43336-2	SQL Queries for Mere Mortals	3	Michael Hernandez
0-201-69471-9	Database Design for Mere Mortals	2	Michael Hernandez
0-07-225364-9	Databases Demystified	7	Andy Oppel
0-12-369379-9	SQL for Smarties	1	Joe Celko

Database ERD diagram

(Columns in bold indicate NOT NULL constraints)



A checkout transaction consists of two DML statements:

1. Add a record into the book_trans table.
2. Decrease the No_of_Copies in the books table by one.

A return transaction consists of two transactions:

1. Update the return date in the book_trans table.
2. Increase the No_of_Copies column in the books table by one.

On the next page, you will find the individual tasks necessary to complete this final. For each task or sub task, provide the SQL statement and a screenshot, some procedure as with homework assignments.



1. **Create the tables**

A. **CREATE TABLE books:**

Based on the ERD diagram create the Books table. Make sure to add the primary key constraint. In addition to the DDL statement, issue a describe command to show the created table.

B. **CREATE TABLE book_trans:**

Based on the ERD diagram create the Book_trans table. Choose a number data type for the primary key Book_trans_ID. This table has four constraints:

- Primary Key constraint
- One unique constraint on ISBN, Employee_ID, Checkout_dte (implementation of business rule for checking out books, see first page)
- Foreign key constraint Employee_ID
- Foreign key constraint ISBN

In addition to the DDL statement, issue a describe command to show the created table.

2. **Populate the books table:**

Write five DML statements to create the records in the book table (see previous page for book data). After inserting the book records, issue a SELECT statement to show the book records.

3. **Create the following checkout transactions:**

ISBN 0-07-225790-3 checked out on 15-NOV-2007 by employee 101(book_trans_id=1)
ISBN 0-07-225790-3 checked out on 12-NOV-2007 by employee 151(book_trans_id=2)
ISBN 0-201-69471-9 checked out on 14-NOV-2007 by employee 175(book_trans_id=3)
ISBN 0-12-369379-9 checked out on 16-NOV-2007 by employee 201(book_trans_id=4)

You need to populate the following column values(book_trans): Book_trans_id, ISBN, Employee_id, Checkout_dte
You need to update the No_of_Copies(books table) values.
Show book_trans and book records by issuing a SELECT statement.



4. **Update the due date in table book_trans:**

Write one UPDATE statement using an expression with the Decode function or a CASE WHEN statement and the to_Char function:

```
UPDATE book_trans SET due_dte = expression
```

Issue a SELECT statement to show all records and columns from table book_trans.

If you were not able to create the due date expression, simply manually calculate the due date for each book transaction record and hardcode the due date into an UPDATE statement (will cause grade reduction). In this case you need to update the four book transaction records individually. You also need to use a WHERE clause to update the records individually.

```
UPDATE book_trans SET due_dte = date WHERE book_trans_id = value
```

5. **Create the following check-in transactions:**

Employee 151 returned book 0-07-225790-3 on 18-DEC-2007

Employee 201 returned book 0-12-369379-9 on 16-DEC-2007

You need to update the following columns (book_trans): Return_dte

You need to update the No_of_Copies(books table)

Issue two SELECT statements to show all records and columns from table books and book_trans.

6. **Calculate the Past Due fees using SQL:**

Create first a SELECT statement with an expression that calculates the Past Due fees using the book_trans table. Use the Decode function and the sign function or CASE WHEN to test whether:

- sign(return_dte - due_dte) is -1 or 0 (no fees due, update to 0)
- sign(return_dte - due_dte) is 1 (fees due, update to (return_dte - due_dte)*2).

If the book is not returned yet, the whole expression will be null and no update will be performed. Then write an UPDATE statement to update the Past Due fee column in the book_trans table:

```
UPDATE book_trans SET PastDue_fees = decode( expression) or CASE WHEN statement
```

Issue a SELECT statement to show all records and columns from table book_trans.

7. **Write a SQL statement displaying the following columns:**

- Employee's Last_name concatenated with a comma and blank space concatenated with Employee's first_name. Use Name as an alias, limit this column to the first 30 characters
- Title of the book, use Book as an alias and limit this column to the first 40 characters
- Checkout_dte, Due_dte, Return_dte, PastDue_fees
- Order this SQL statement by last_name