# Tutorial: Stored Procedures

Students at the National University of Ngendipura (NUN) buy books for their studies. They also lend and borrow books to and from other students. Your company, Apasaja Private Limited, is commissioned by NUN Students Association (NUNStA) to implement an online book exchange system that records information about students, books that they own and books that they lend and borrow.

The database records the name, faculty, and department of each student. Each student is identified in the system by her email. The database also records the date at which the student joined the university (year attribute).

The database records the title, authors, publisher, year and edition and the ISBN-10 and ISBN-13 for each book. The International Standard Book Number, ISBN-10 or -13, is an industry standard for the unique identification of books. It is possible that the database records books that are not owned by any students (because the owners of a copy graduated or because the book was advised by a lecturer for a course but not yet purchased by any student.)

The database records the date at which a book copy is borrowed and the date at which it is returned. We refer to this information as a loan record.

For auditing purposes the database records information about the books, the copies and the owners of the copies as long as the owners are students or as there are loan records concerning the copies. For auditing purposes the database records information about graduated students as long as there are loan records concerning books that they owned.

**This tutorial uses the schema and data for the database created in "Tutorial: Creating and Populating Tables" including all the updates done during the tutorial.**

## Questions

*Not all questions will be discussed during tutorial. You are expected to attempt them before coming to the tutorial. You may be randomly called to present your answer during tutorial. You are encouraged to discuss them on Canvas Discussion.*

1. **Stored Functions and Procedures**.

   (a) Write a function `borrow_book_func` that, given the the email of a borrower (`VARCHAR(256)`), the ISBN13 of a book (`CHAR(14)`), and the borrow date (`DATE`), checks whether there is an available copy of the book, and, if that is the case, inserts a new loan record of the copy by the borrower. Return a message indicating success or failure of insertion.

Additionally, execute the following scenario using your function.

> Adeline Wong, with email awong007@msn.com, tries to borrow 3 copies of "Applied Calculus" by Deborah Hughes-Hallett, et al. with ISBN13 value of 978-0470170526.

(b) Write a procedure `borrow_book_proc` that, given the the email of a borrower (`VARCHAR(256)`), the ISBN13 of a book (`CHAR(14)`), and the borrow date (`DATE`), checks whether there is an available copy of the book, and, if that is the case, inserts a new loan record of the copy by the borrower. Raise a notice [3] indicating success or failure of insertion.

Additionally, execute the following scenario using your procedure.

> Adeline Wong, with email awong007@msn.com, tries to borrow 4 copies of "Calculus: Single Variable" by Deborah Hughes-Hallett, et al. with ISBN13 value of 978-0470089156.

2. **Cursor**.

(a) Write a function `borrow_diff` that takes in no parameter and does the following steps. First, sort the students in terms of the number of books borrowed in descending order. If there are multiple students with the same number of books borrowed, they can appear in any order. Then, find the difference in the number of books borrowed between a student $S$ and the previous student in the sorted order. For the first student, let the difference be `NULL`.

The output should be a table with the following columns.

| email | diff |
|---|---|

`email` is the student email in the given sorted order. `diff` is the difference amount as mentioned above.

## References

[1] S. Bressan and B. Catania. *Introduction to Database Systems*. McGraw-Hill Education, 2006. ISBN: 9780071246507.

[2] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems: The Complete Book*. 2nd ed. Prentice Hall Press, 2008. ISBN: 9780131873254.

[3] *PostgreSQL Docs: Errors and Messages*. https://www.postgresql.org/docs/current/plpgsql-errors-and-messages.html. [Online; last accessed 2025].

[4] *PostgreSQL Docs: Trapping Errors*. https://www.postgresql.org/docs/current/plpgsql-control-structures.html#PLPGSQL-ERROR-TRAPPING. [Online; last accessed 2025].

[5] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. 2nd. USA: McGraw-Hill, Inc., 2000. ISBN: 0072440422.

[6] *W3schools Online Web Tutorials*. https://www.w3schools.com/. [Online; last accessed 2025].