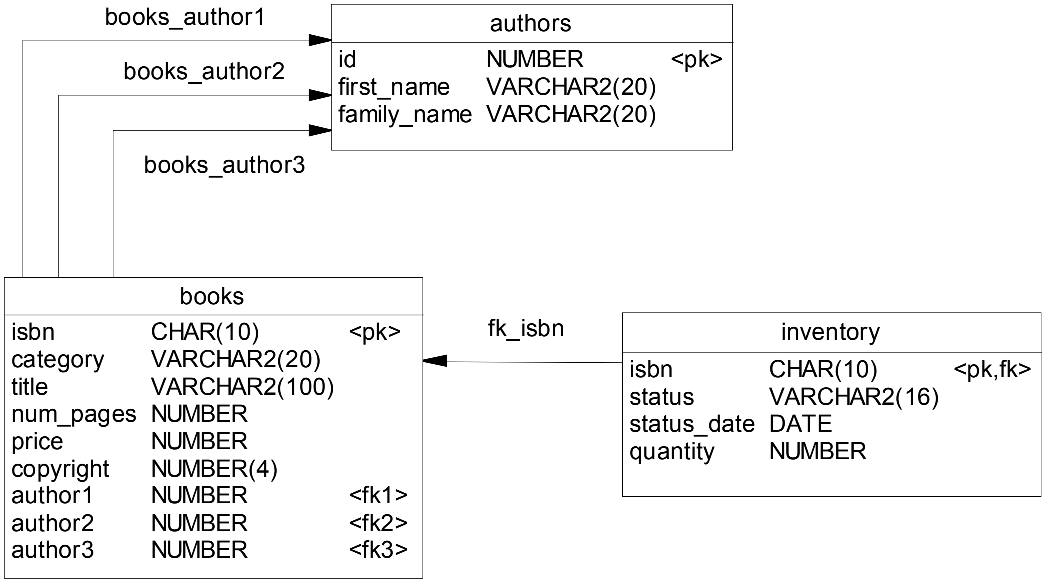
Database Systems
Lab 7
PL/SQL Programming 1
Version 1.1

The objectives of this practical session is to write simple PL/SQL code using SQL*Plus utility and run the code against Oracle Server.

Three tables will be used in this session and the script to create and populate them is provided on the shared folder. The file is called **'author.sql'**. Before completing any of these exercises you should run this script in iSQLPlus to create the sample database

The schema used for this session is as follows:



Task (1)

Type in the following PL/SQL Code and execute it-

```
SET SERVEROUTPUT ON
DECLARE
   max_author_no  Authors.id%type;
BEGIN
   SELECT MAX(id)  INTO max_author_no
   FROM authors;
   DBMS_OUTPUT.PUT_LINE('The maximum author is : ' ||
max_author_no);
 END;
/
```

What is the maximum author id?

Task (2)

Write an anonymous block to display the date on which status was entered in ‘inventory’ table for the book titled ‘Oracle DBA 101’

Hints:

Remember you need to set the **serveroutput on** to be able to print within a block. Your select statement will need to join the inventory and books tables together.

```
SET  SERVEROUTPUT ON
DECLARE
  Date_entered inventory.status_date%type;
  Book_Title  books.title%type := 'Oracle DBA 101';
BEGIN
  SELECT status_date INTO Date_entered
  FROM inventory, books
  WHERE books.isbn = inventory.isbn AND
  Title = Book_Title;
  DBMS_OUTPUT.PUT_LINE('Book : ' || Book_Title || ' entered on
'|| Date_entered);
END;
/
```

Task (3)

Using the same code developed for Task (2), change the title to ‘High-Performance SQL Tuning’ and display the status from inventory table.

```
SET  SERVEROUTPUT ON
DECLARE
  pstatus inventory.status%type;
  Book_Title  books.title%type := 'Oracle High-Performance SQL
Tuning';
BEGIN
  SELECT status INTO pstatus
  FROM inventory, books
  WHERE books.isbn = inventory.isbn AND
  Title = Book_Title;
  DBMS_OUTPUT.PUT_LINE('Book : ' || Book_Title || ' Status :
'|| pstatus);
END;
/
```

Task (4)

Write an anonymous block to calculate and display the full price of the book titled “Oracle Database 10g A Beginner's Guide”.

The code should then check and apply the discount using the guidelines below t o produce the discounted price which should also be outputted. You should ensure that your discounted price is round up to 2 decimal places e.g. £12.99.

- No discount if the price is less than £25,
- 25% discount if the price is less than £40,

- 40% discount if the price is less than £50.
- For any other price, the discount is 50%

a) What is the discount price does you program give for Oracle Database 10g A Beginner's Guide?

b) What is the discount price does you program give for Oracle 24x7 Tips and Techniques?

```
SET SERVEROUTPUT ON
DECLARE
    full_price      books.price%type;
    book_title  VARCHAR2(100);
    discount_price books.price%type;

BEGIN
    book_title := 'Oracle 24x7 Tips and Techniques';

    SELECT price  INTO full_price
    FROM books
    WHERE title like book_title ;

    IF full_price < 25  THEN
                discount_price := full_price;
    ELSIF full_price >= 25 and full_price <40   THEN
               discount_price :=  full_price -
(full_price*0.25);
    ELSIF full_price >= 40 and full_price <=50 THEN
       discount_price :=  full_price - (full_price*0.4);
   ELSE
     discount_price :=  full_price - (full_price*0.5);
    END IF;

DBMS_OUTPUT.PUT_LINE (book_title || 'Full Price:
'||full_price||' Disounted Pice: '|| ROUND(discount_price,2));
EXCEPTION
         WHEN others THEN
            DBMS_OUTPUT.PUT_LINE (SQLERRM);
END;
/
SHOW errors
```