**CP5503 Enterprise Database Systems Oracle**

**Assignment Part3 - APEX (10% due by 5 pm Friday Week10)**

**Demo Pages**

You can view the demo at:

<https://apex.oracle.com/pls/apex/f?p=80582>

**Username:** jc165984\_usr

**Password:** Apex\_nov\_15

**Task 1 Setup your development environment (10 marks)**

**Request a workspace**

Go to the hosted demo environment <http://apex.oracle.com>/en and request a workspace:



- What type of workspace would you like to request? Select **Application Development**.

+ Identify **the administrator** for the service:

- First Name: Your real first name

- Last Name: Your real last name

- Email: Must be your jcu student email

- Workspace: Must be your jcnumber, for example, jc111222

- New Schema to Create: Must be LIBDB\_JCNUMBER, for example, LIBDB\_JC111222

- Space Location: 25 MB

- Answers to survey questions: Other, New to APEX, No, No.

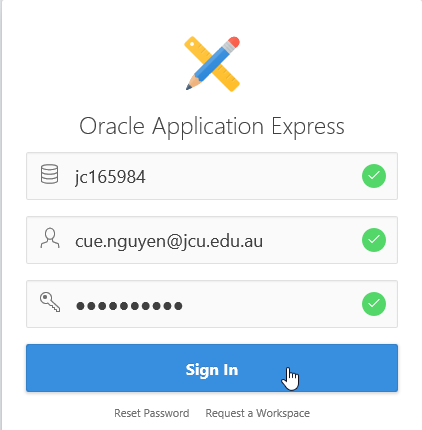
+ The system generates an email with your login credentials and sends it to the email address you entered.

- You should receive an email with the email’s subject as “Action required: account request for …”

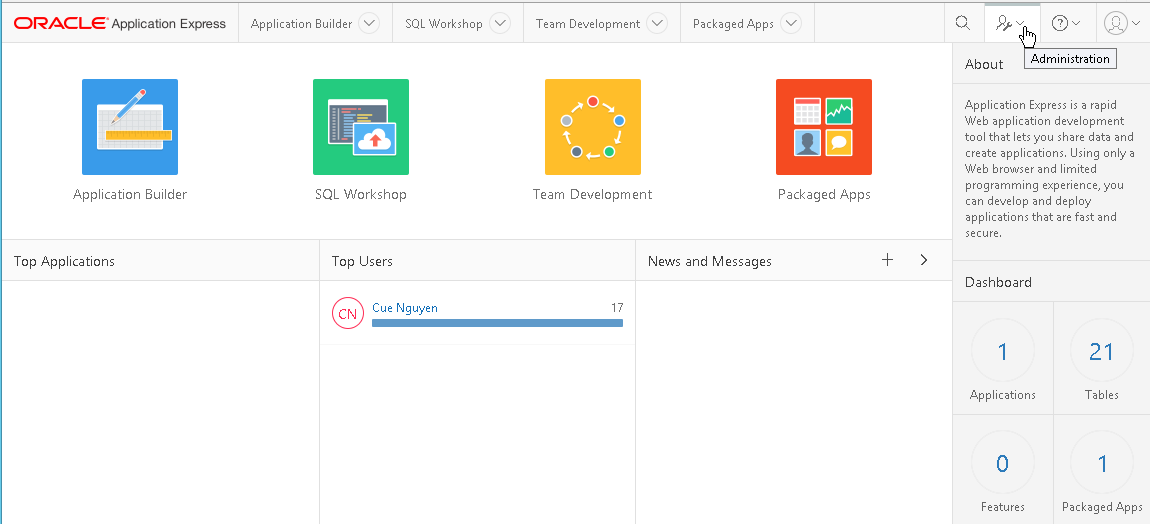
- You have to click on the link to complete the approval process and receive your credentials.

- It will take a number of minutes and then you will be asked to change your password (strong password required).

- You can now sign-in.



If successfully, you will see this screen:



**Creating Additional User Accounts**

Please sign in if you haven’t done so. You are now to create two additional user accounts.

On the right top corner, click **Administrator**  → **Manage Users and Groups** → **Create User**:

**+ One developer account:**

User Identification

- Username: jcnumber\_dev, for example, jc111222\_dev

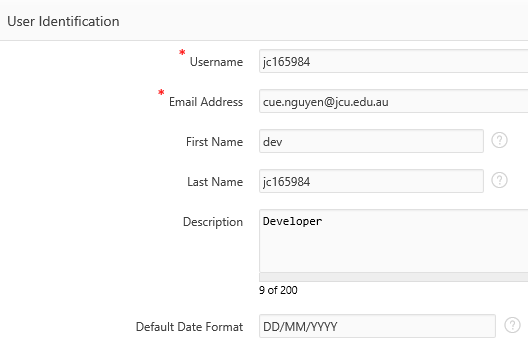
- Email Address: your jcu student email

- First Name: dev

- Last Name: jcnumber

- Description: Developer

- Default Date Format: DD/MM/YYYY



Account Privileges

- Default Schema: LIBDB\_JCNUMBER, for example, LIBDB\_JC111222

- Accessible Schemas (null for all):

- User is a workspace administrator: No

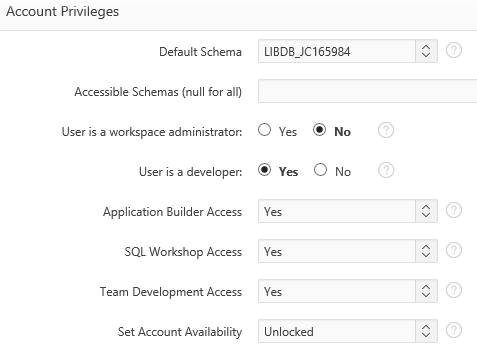
- User is a developer: Yes

- Application Builder Access: Yes

- SQL Workshop Access: Yes

- Team Development Access: Yes

- Set Account Availability: Unlocked



Password

- Password: \*\*\*\*\*\*\*\* (Passwords are case sensitive)

- Confirm Password: \*\*\*\*\*\*\*\*

- Require Change of Password on First Use: No

Group Assignments

- Group Assignments: SQL Developer

Click 

**+ One end user account:**

User Identification

- Username: jcnumber\_usr, for example, jc111222\_usr

- Email Address: your jcu student email

- First Name: usr

- Last Name: jcnumber

- Description: End user

- Default Date Format: DD/MM/YYYY

Account Privileges

- Default Schema: LIBDB\_JCNUMBER, for example, LIBDB\_JC111222

- Accessible Schemas (null for all):

- User is a workspace administrator: No

- User is a developer: No

- Application Builder Access: No

- SQL Workshop Access: No

- Team Development Access: Yes

- Set Account Availability: Unlocked

Password

- Password: \*\*\*\*\*\*\*\* (Passwords are case sensitive)

- Confirm Password: \*\*\*\*\*\*\*\*

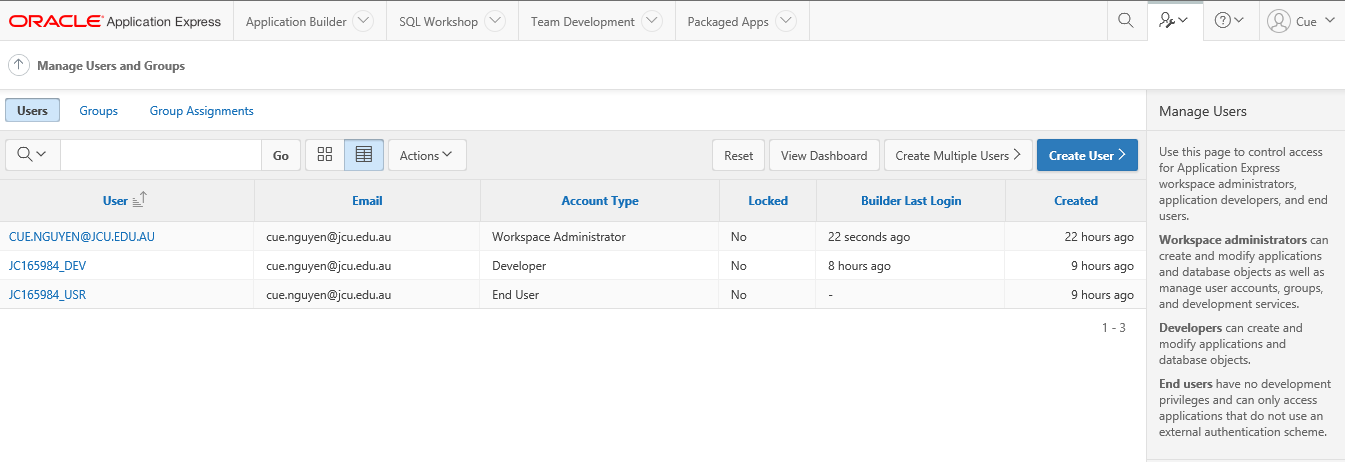
- Require Change of Password on First Use: No

Group Assignments

- Group Assignments: SQL Developer

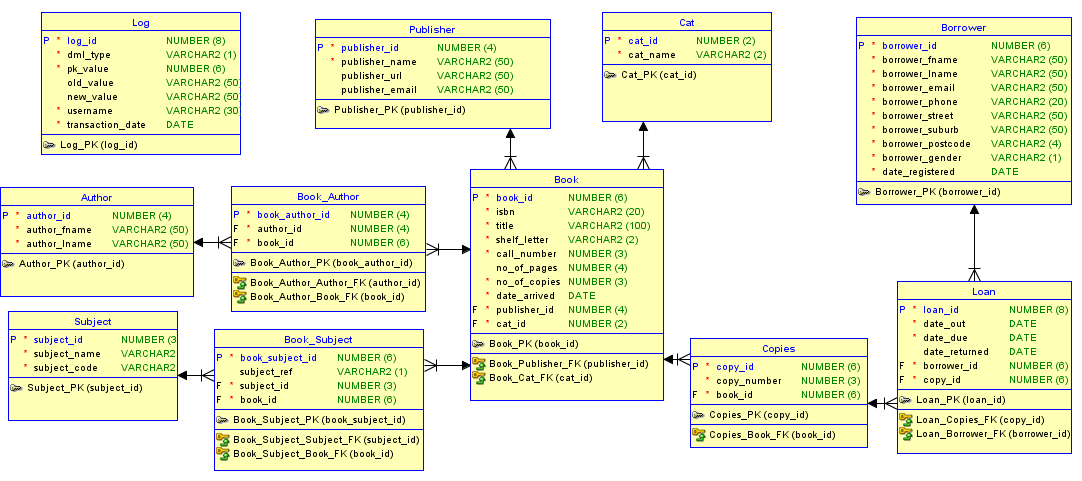
Click 

When you have done, Your **Manage Users and Groups** window will be similar to this:

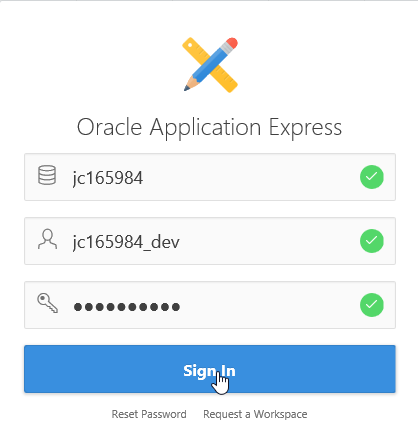


**Task 2 Installing a Simple Library Database Objects (10 marks)**

In this section, you are going to create a simple library database. The database consists of 11 tables which you can see on the following E-R diagram (next page).



Firstly, login as **the developer**:



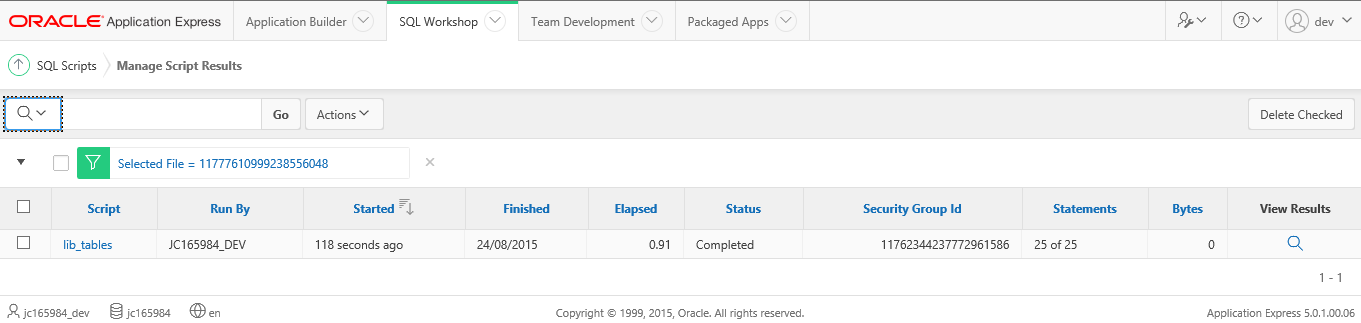
Select **SQL Workshop** on the top navigation bar:



You then click **SQL Scripts** → **Upload** to upload and run **lib\_tables.sql** script. It will create 11 tables:



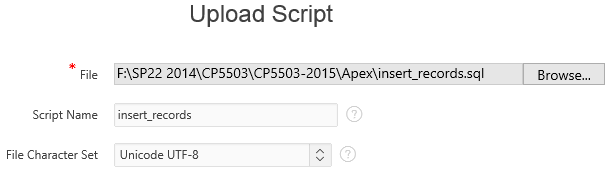
If things go right, you should see this window:



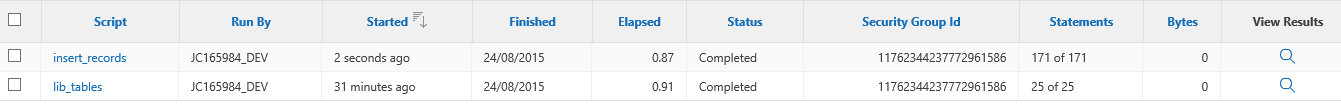
You can click **View Results** to see whether each statement was successfully executed or not (strongly recommended).

We now continue to upload and run the second script **insert\_records.sql** to insert records into the tables. Please note that the script also helps us to create 11 sequences as well.

To upload the second script, click **SQL Scripts**  to go back and upload and run the script **insert\_records.sql**.

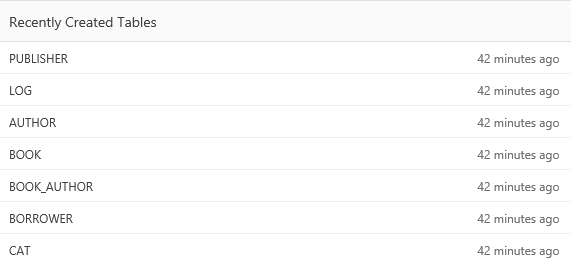


If things go right, you should see this:



You can click on any of the 2 scripts to see its content and even to modify it.

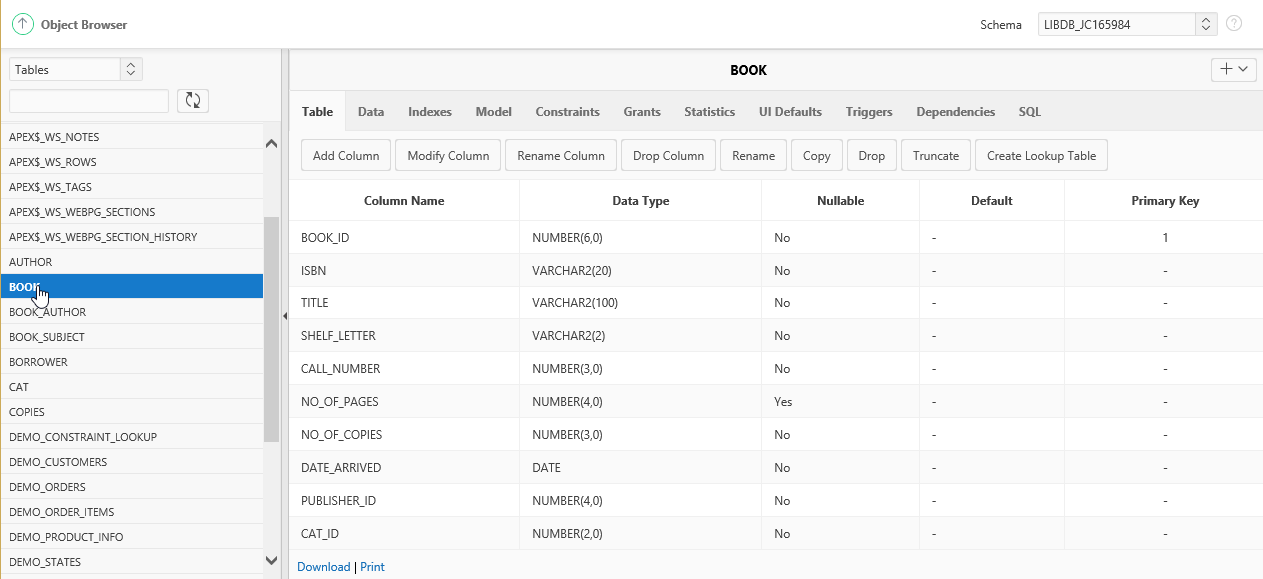
Click **SQL Workshop** on the top navigation bar to go back to the top level. You can see **APEX** notifies you the recent created tables.



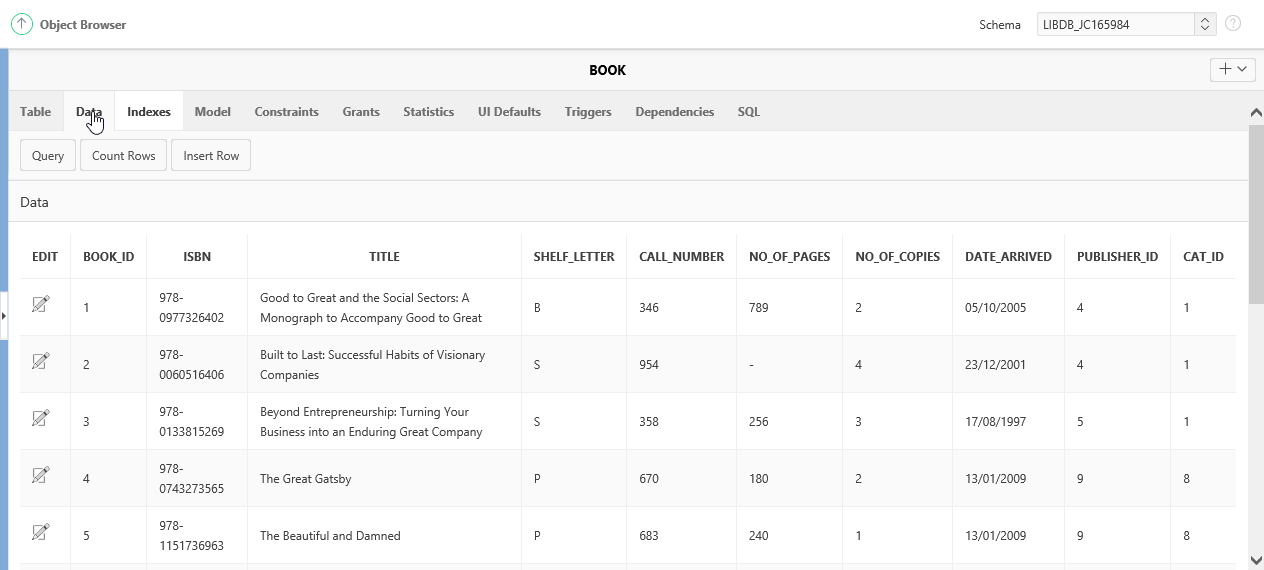
In addition, pay attention to the right hand side of the window; there is a panel named **Create Object** which allows you to manually create objects such as table, sequence, trigger etc.

You now click **Object Browser** to see what tables, sequences that you have had. You should see 11 new created tables and 11 new created sequences.

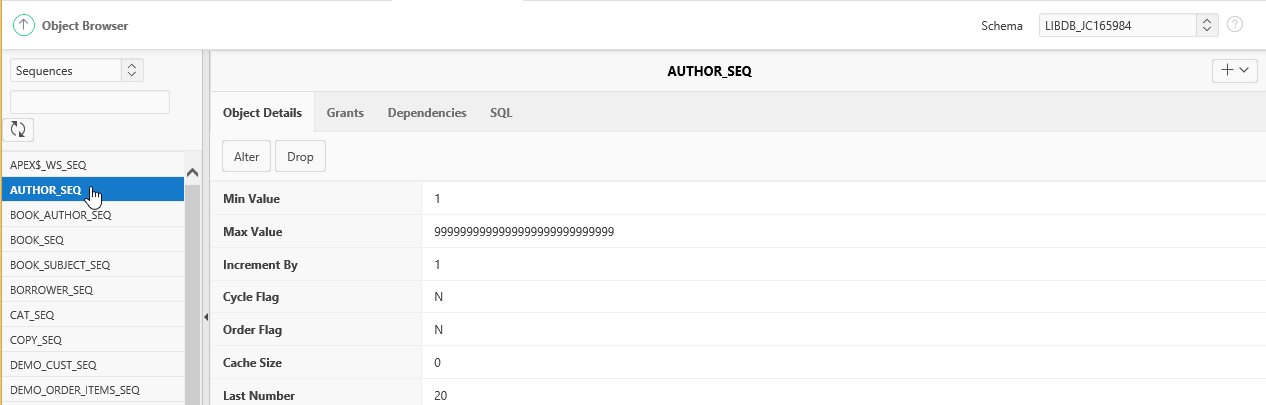
For example, in **Object Browser**, select the table **Book**, APEX will show you the structure of **Book**.



You can see other information such as data, constraints etc. by clicking on the supplied links. For example, select **Data**, you can see the content of **Book**.



To see the sequences, in **Object Browser**, change the category from **Tables** to **Sequences**, you should see the 11 new sequences. If you select the sequence **AUTHOR\_SEQ**, APEX will show you the details of **AUTHOR\_SEQ**.



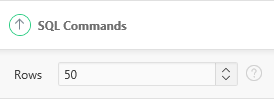
Do you know why **Last Numbe**r is 20?

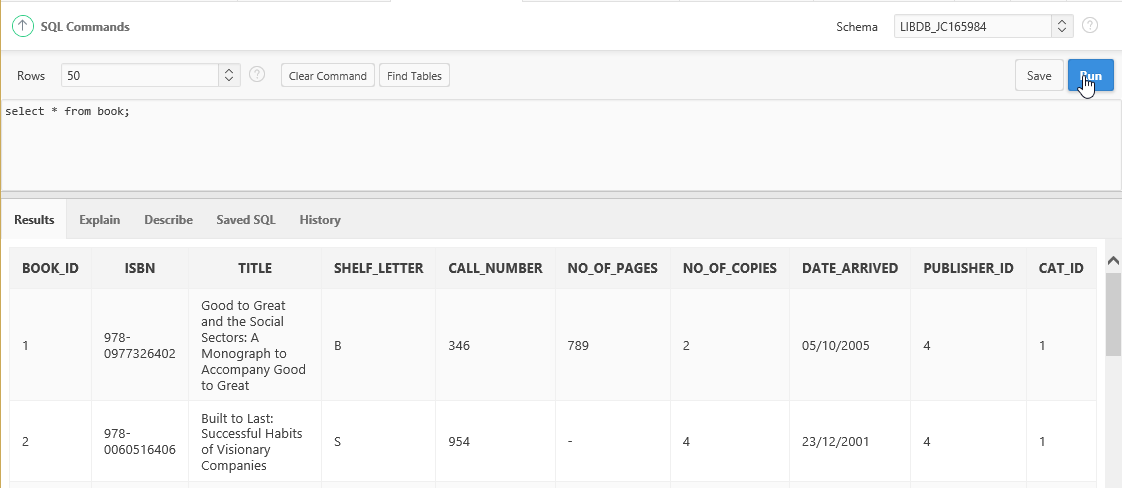
Go back to the top level of **SQL Workshop** and select **SQL Commands**. APEX provides you a simple text editor to write and run a SQL statement.

Try it by entering and running the following statement:

select \* from book;

By default, the number of rows is set to 10. You can change it, for example, if results are more than 10 rows change it to 50.





You now continue uploading and running the following scripts to create 5 functions:

1. FUNC\_GET\_AVAILABLE\_COPY.sql

2. FUNC\_COPY\_AVAILABLE\_STR.sql

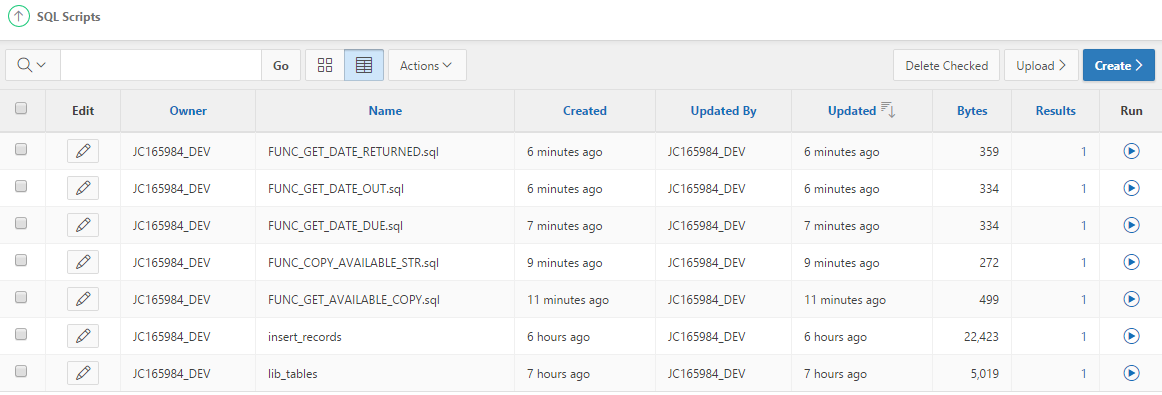
3. FUNC\_GET\_DATE\_DUE.sql

4. FUNC\_GET\_DATE\_OUT.sql

5. FUNC\_GET\_DATE\_RETURNED.sql

Note: Must run the script 1 first and then 2 because the second script calls a function in the first script.

When completed, the **SQL Scripts** looks similar to this:



**Building a Simple Loan Web-Based Application (80 marks)**

You are going to build a simple loan web based application to provide the following 2 main tasks: “Loan a book” and “Overdue Report”.

**Task 3 Loan a book (40 marks)**

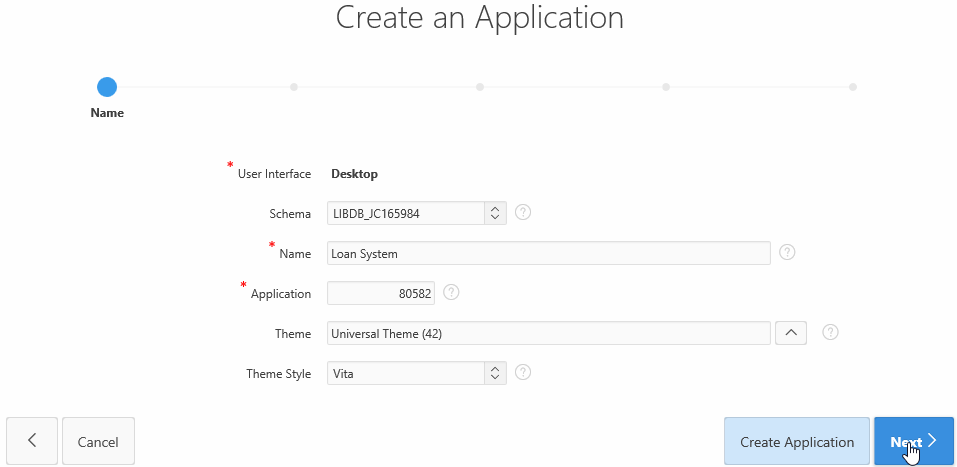
The end user (librarian staff) is able to browse/search for an available copy in order to insert a new loan record.

**Instructions for “Loan a book”**

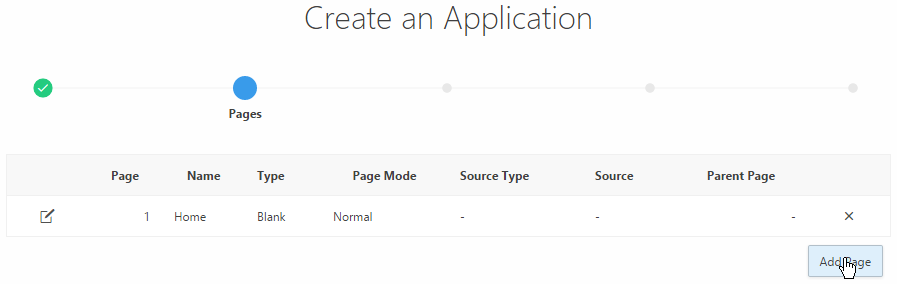
**Create Home and Books pages**

Select **Application Builder** to create a **desktop** database application, for example, “Loan System”.





Click **Next**.



Click **Add Page**. And then do the following settings:

**Select Page Type**: Report

**Parent Page**: Home

**Page Source**: SQL Query

**Report Type**: Interactive

Scroll it down to find the **Query** editor and enter the following query:

select

title,

no\_of\_pages "Pages",

cat\_name || ' ' || shelf\_letter || to\_char(call\_number) || ' ' ||

copy\_number "Call Number",

copy\_number "Copy Number",

copy\_id,

c.book\_id,

FUNC\_GET\_DATE\_OUT(copy\_id) "Date Out",

FUNC\_GET\_DATE\_DUE(copy\_id) "Date Due",

FUNC\_GET\_DATE\_RETURNED(copy\_id) "Date Returned",

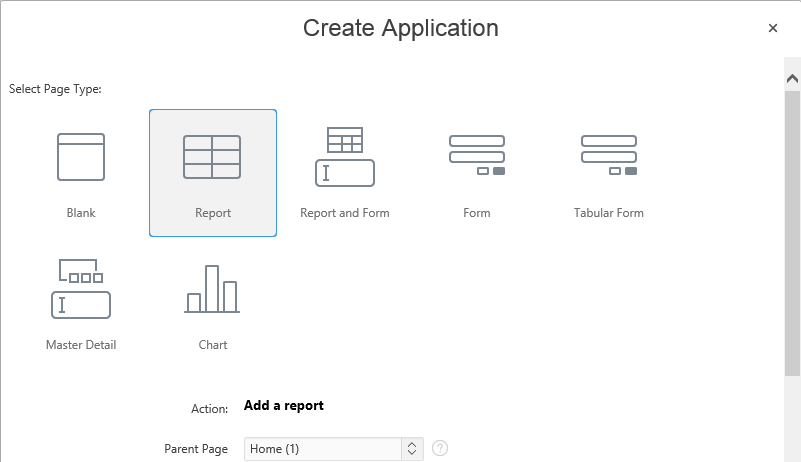
FUNC\_COPY\_AVAILABLE\_STR(copy\_id) "Available"

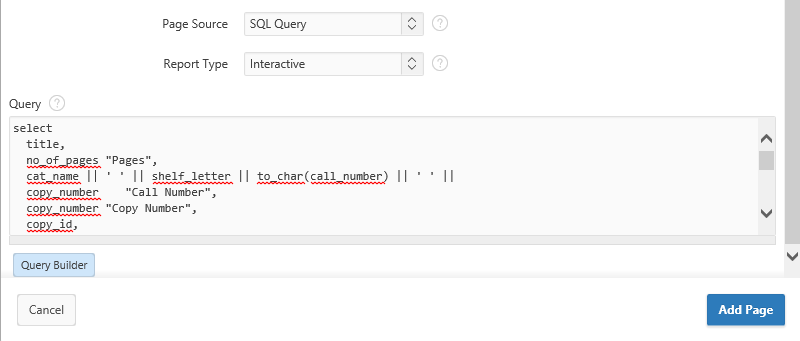
from

copies c inner join book b on c.book\_id = b.book\_id

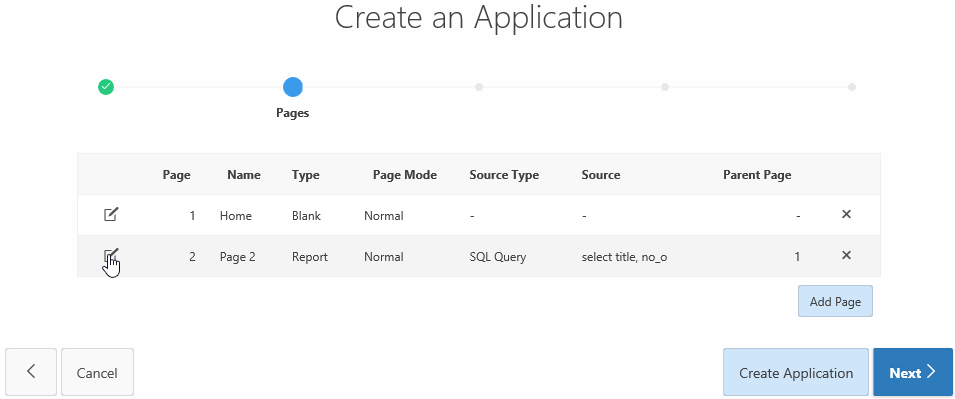
inner join cat on cat.cat\_id = b.cat\_id

order by title, call\_number





Click **Add Page**, APEX will add this page into your application.



You now click the **Edit** button to give a name for the new created page:

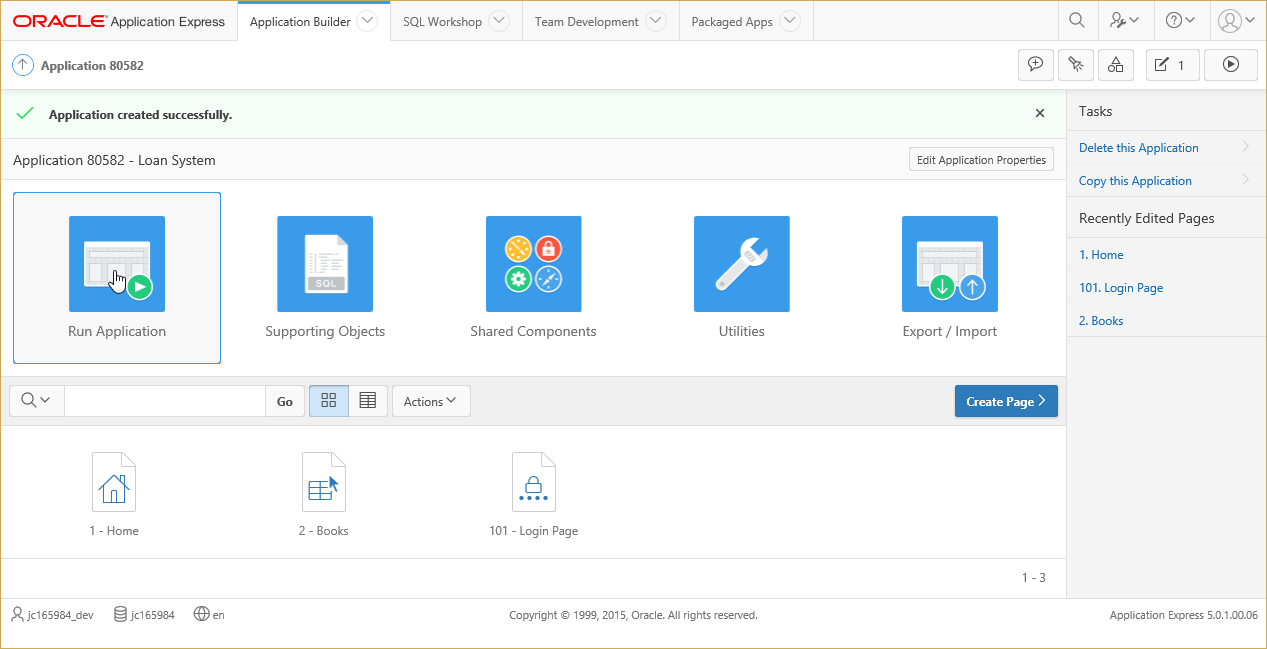
**Page Name**: Books

**Sequence**: 10

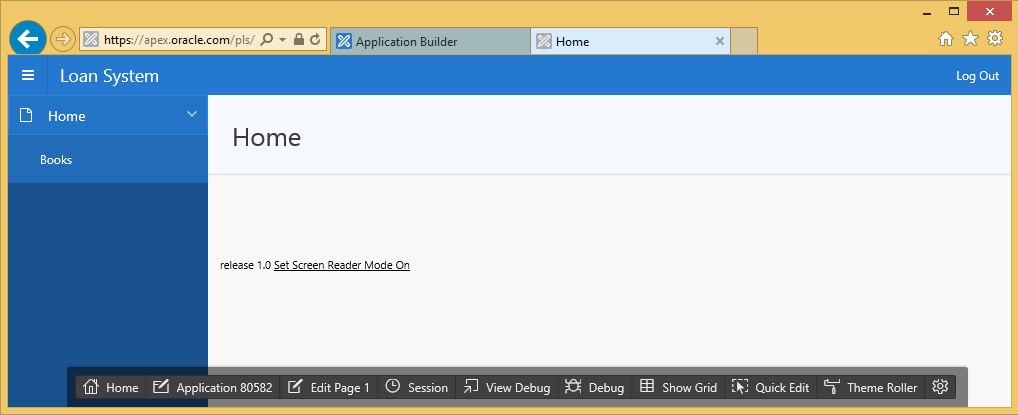
Click **Apply Changes**.

Click **Create Application** and click **Create Application** one more time for confirmation.

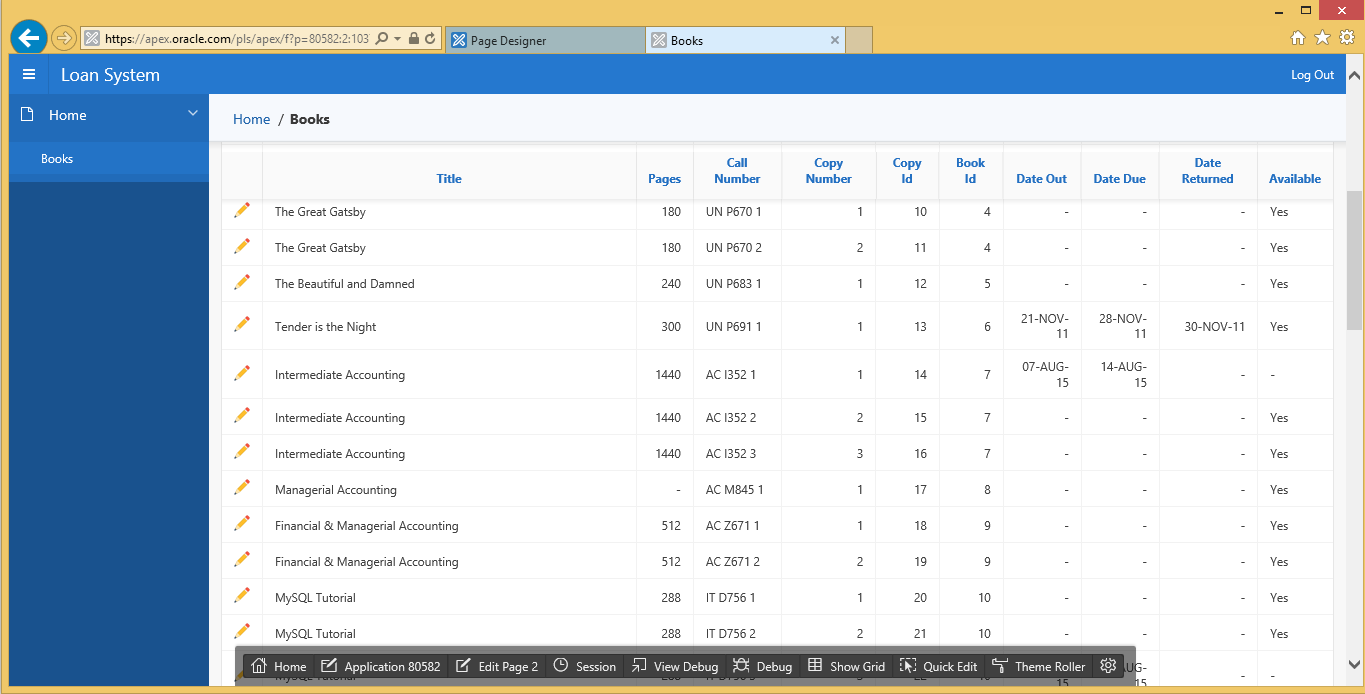
APEX will create your “Loan System” application:



Click **Run Application** to test, your web application will be opened on a new window as follows:

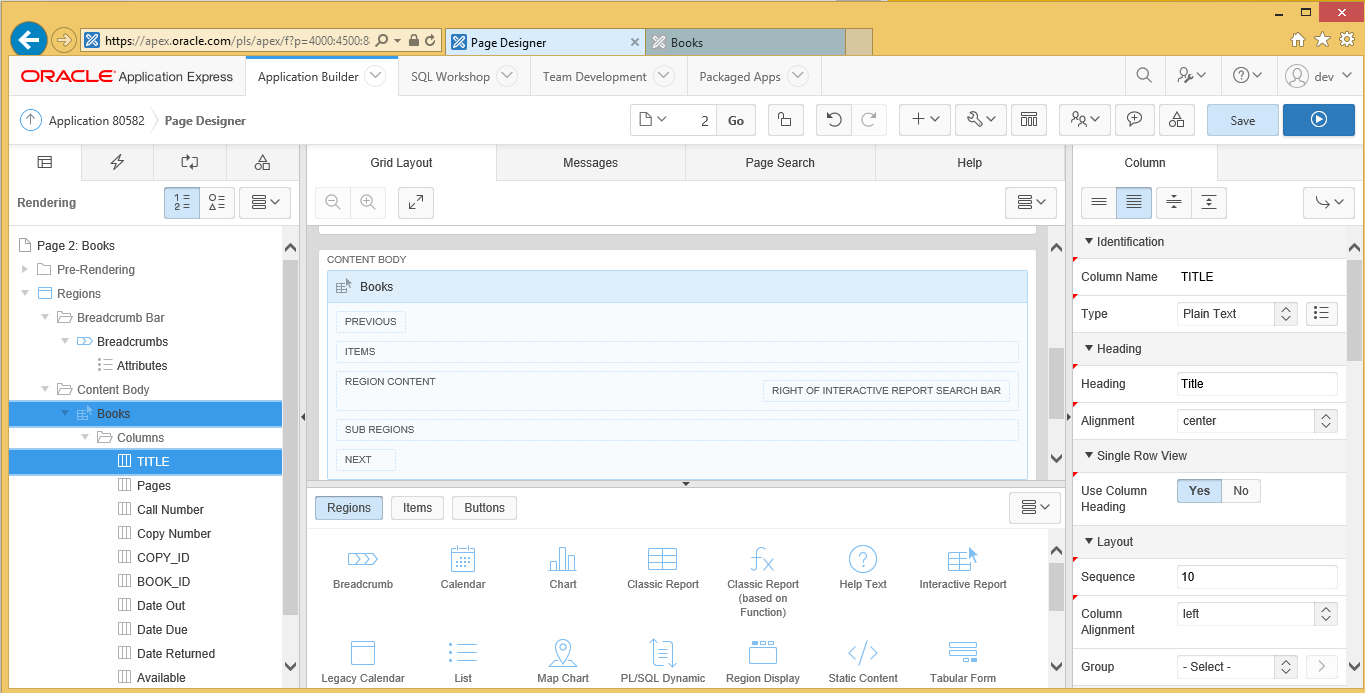


Click Books to see the content of the Books page:



Note: APEX supplies a handy bar at the bottom of the page. It contains useful links to edit your application, for example, to edit pages. Click the link **Edit Page 2**, APEX will display the page structure on the **Page Designer** and allow you to edit it.

On the left of the **Page Designer**, you can see the **Columns**; if you click on one column, for example, **TITLE**; APEX will display the properties of the column **TITLE** on the right.

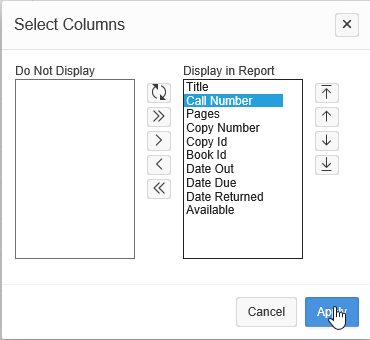


Go back to the **Books** web page. You can see the orders of columns are **Title**, **Pages**, **Call Number,** ... ,**Available.** These orders are actually followed the orders of selected columns in the query.

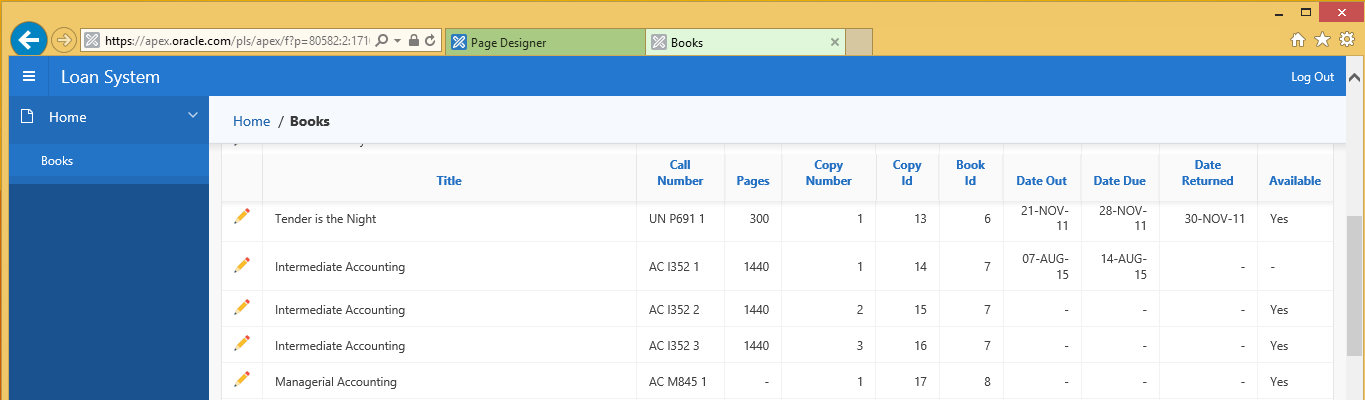
You can change the orders by selecting **Actions** →**Select Columns**



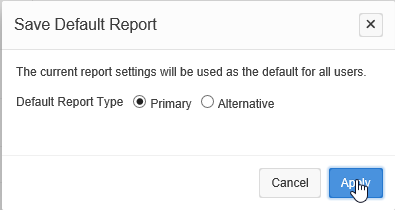
You can then move the columns around, for example, move **Call Number** up to the second:



Click **Apply**. You can see the appearance of columns is updated.



If you want to keep this appearance permanently, you then need to select **Save Report** from the **Actions** drop-down list and save **As Default Report Settings → Apply**.



The application should have the following 6 pages:

|  |  |
| --- | --- |
|  | This page is the first page. It will be displayed when the application is run. This page contains links to the book page, the loan page and the overdue report page. |

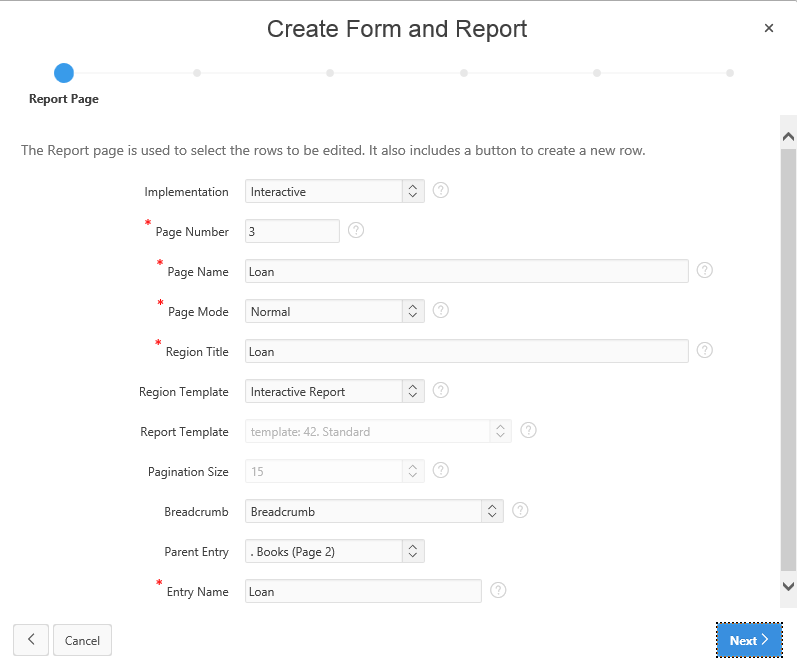
|  |  |
| --- | --- |
|  | This page displays the following info:  - Copy Id  - Book Id  - Book Title  - Call Number: The combination of cat\_name, shelf\_letter, call\_number and copy\_number  - Date Out (a new column added by calling a function to return the date\_out)  - Date Due (a new column added by calling a function to return the date\_due)  - Date Returned (a new column added by calling a function to return the date\_returned)  - Number of Pages  - Available: It will tell if a copy is available for loan or not. It should display “Yes” if available otherwise null. The column value “Yes” has a link to the “Create/Edit Loan” page – page 4. For example, if user clicks on “Yes” on the record having copy\_id of 47, it will then send the copy\_id to Page 4 for creating a new loan record. |

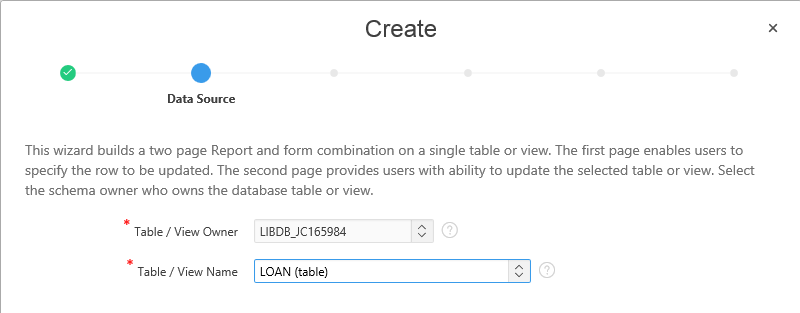
|  |  |
| --- | --- |
|  | This page displays the following info:  - Loan Id  - Borrower Id  - Copy Id  - Date Out  - Date Due  - Date Returned  - Click on  to send the loan record to the “Create/Edit Loan” page for editing. |
|  | This page displays the following info:  - Loan Id: from loan\_seq if create new loan record  - Borrower Id: can be selected from the list of borrower IDs  - Copy Id: can be selected from the list of copy IDs  - Date Out: default value is current system date if creating a new loan record  - Date Due: default value is current system date + 7 if creating a new loan record  - Date Returned: NULL is allowed |
|  | This page displays the following info:  - Borrower First Name  - Borrower Last Name  - Borrower email  - Title  - Call Number: The combination of cat\_name, shelf\_letter, call\_number and copy\_number  - Overdue: A number of overdue days |
|  | This page is automatically added by Oracle Application Express. |

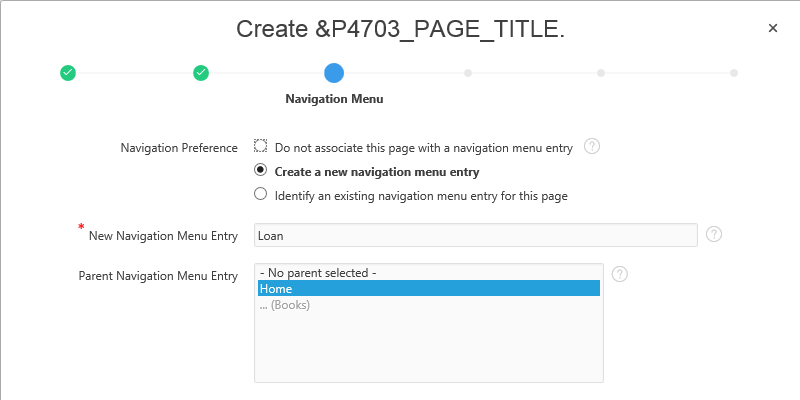
We already created **Home** and **Books** pages. However, we haven’t created a link on the column **Available** yet because we haven’t created the page “Create/Edit Loan” – page 4. We now continue creating the next 2 pages: Page 3 and Page 4.

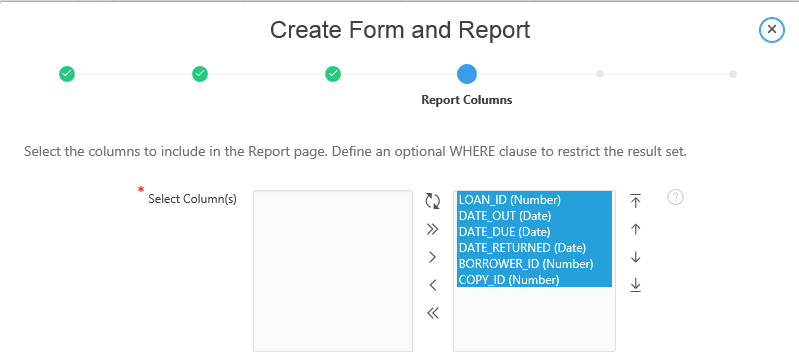
**Create Page 3: Loan & Page 4: Create/Edit Loan**

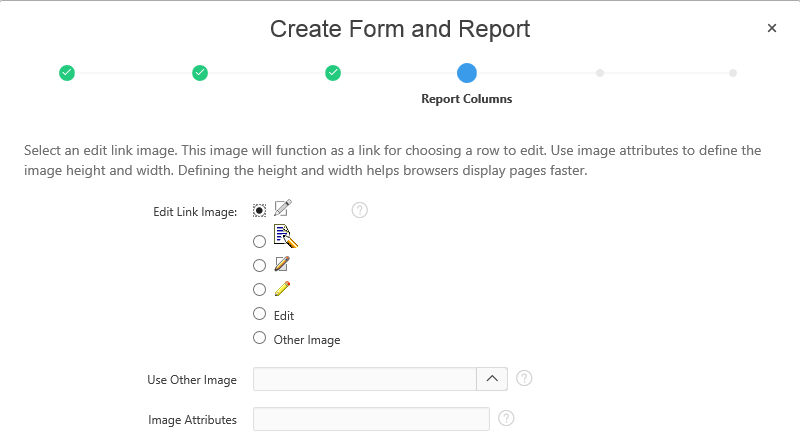
Back to the top level of the “Loan System” application, click **Create Page** → **Form** → **Form on a Table with Report**. Follow the below images to enter or select information.

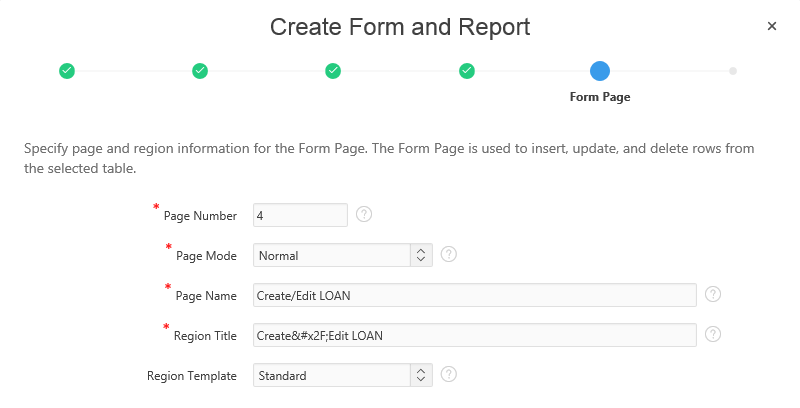




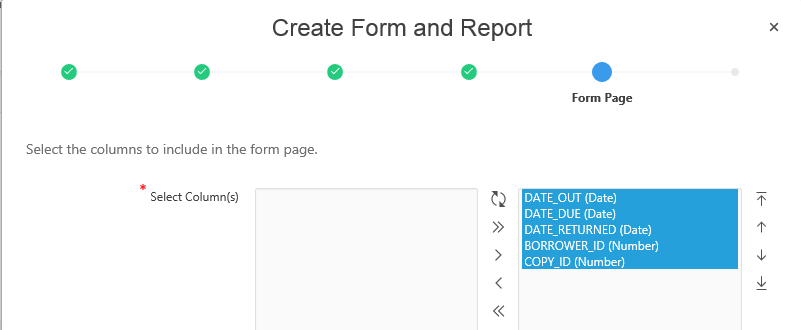


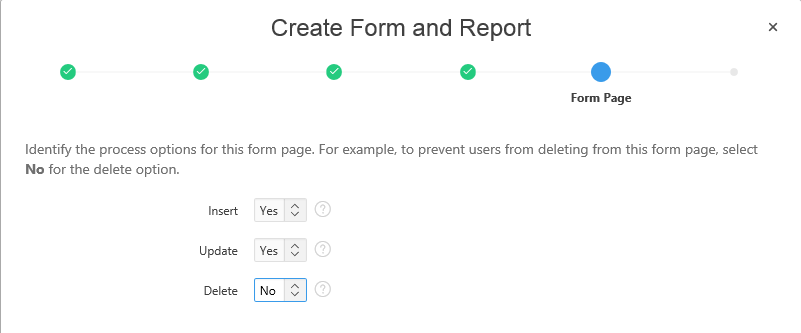






Click **Next** → keep **Primary Key Type**: Managed by Database (ROWID) → **Next**.

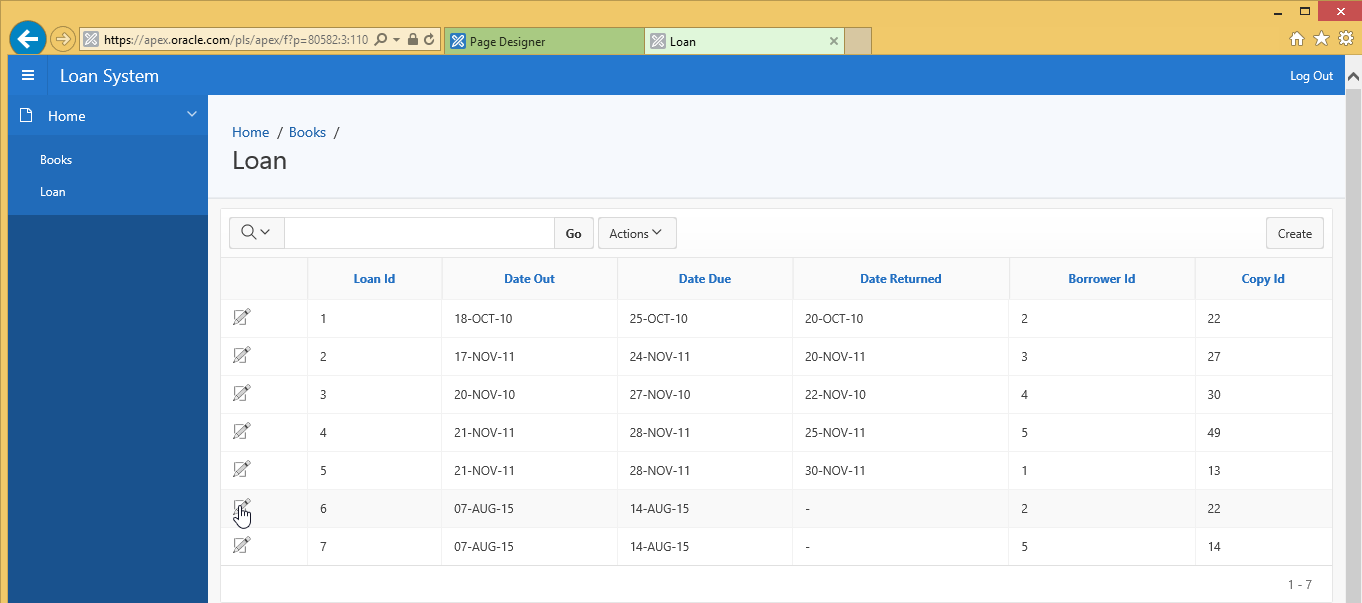




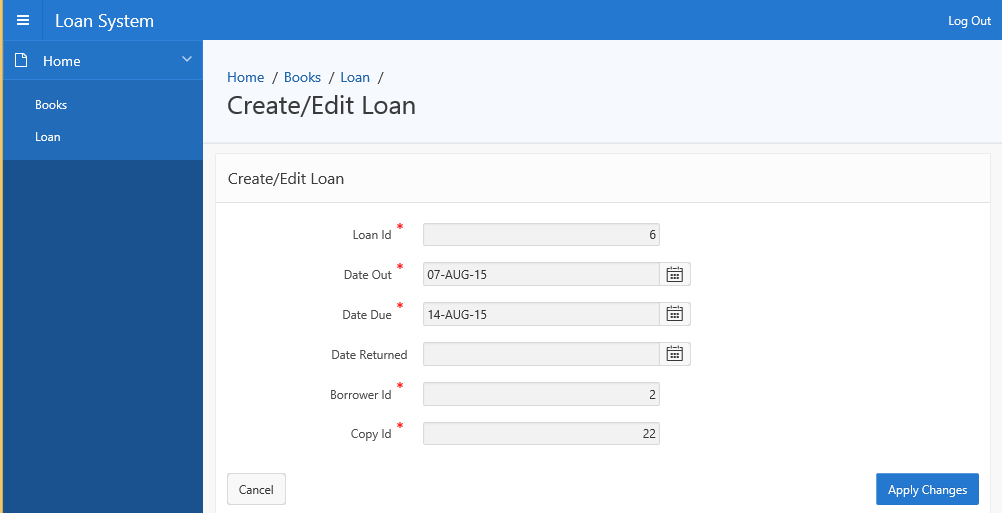
Click **Create**. APEX will create Page 3 and 4 for you and re-direct you to the **Page Designer**.

Re-run your “Loan System” application to test Page 3 and 4.

On the **Loan** page, click the **Edit** button on the 6th record (loan\_id = 6):



The 6th record will be sent to the **Create/Edit** **Loan** page:

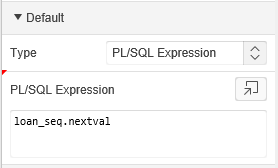
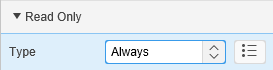


Just hit **Apply Changes** without any modification, you can see the system takes you back to the **Loan** page.

Back to the editing of the **Loan** page on the **Page Designer**, on the left, locate and select the button **CREATE**, you will see its properties on the right. From the behaviour **Action** and **Target**, you can see that if you click on the button CREATE, your application will redirect you to Page 4 (**Create/Edit Loan** page).

You now open the **Create/Edit Loan** page on the **Page Designer**. We will modify a number of properties of a number of columns.

Select the column **P4\_Loan\_ID** and set the default value as **loan\_seq.nextval** and set **Read Only** as **Always**.

Save changes and re-run the application to test.

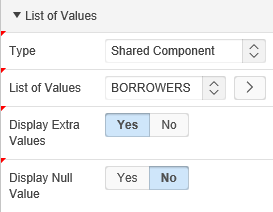
Back to the **Create/Edit Loan** page on the **Page Designer**, select the column **P4\_BORROWER\_ID**, you can see on the right its type is **Number Field**. We want to be able to pick a borrower\_id from a list of borrower\_ids.

To do so, firstly, we have to create a LOV (List of Values). Click the **Shared Components**  on the top right corner → **Lists of Values** → **Create** → **Create List of Values:** From Scratch → **Name**: Enter **borrowers** for **Name**→ **Type**: Dynamic → Enter this query onto the Query editor:

select borrower\_fname || ' ' || borrower\_lname name, borrower\_id from borrower

Click **Create List of Values**.

Back to the **Create/Edit Loan** page on the **Page Designer**, select the column **P4\_BORROWER\_ID**, on **Type** → select **Popup LOV** and you have to scroll down to **List of Values** to set:



Do the same for P4\_COPY\_ID:

LOV name: **copies**

LOV query:

select 'copy id: ' || to\_char(copies.copy\_id) || ' - '

|| title copy\_title, copies.copy\_id

from copies, book

where copies.book\_id = book.book\_id

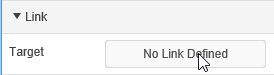
Select P4\_DATE\_OUT and set the default value to SYSDATE.

Select P4\_DATE\_DUE and set the default value to SYSDATE + 7.

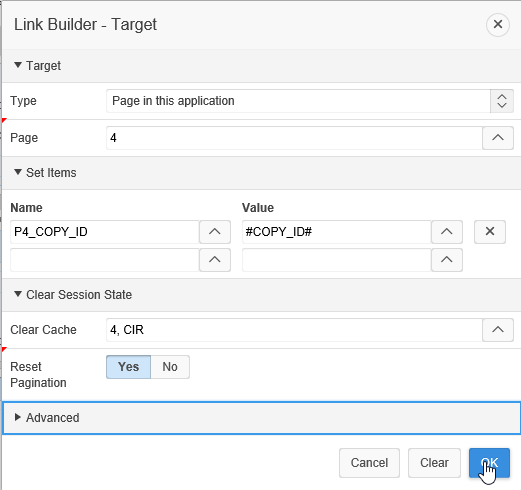
Save changes and re-run the application to test.

Finally, open the **Books** page on the **Page Designer**, we want to add a link on the column **Available** to the “**Create/Edit Loan**” page – page 4. For example, if user clicks on “Yes” on the record having copy\_id of 47, it will then send the copy\_id to Page 4 for creating a new loan record.

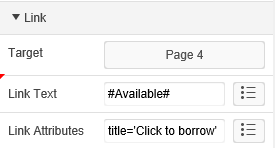
To make that link, select the column **Available**, select **Type** as **Link** and then scroll down to click on “No Link Defined”:



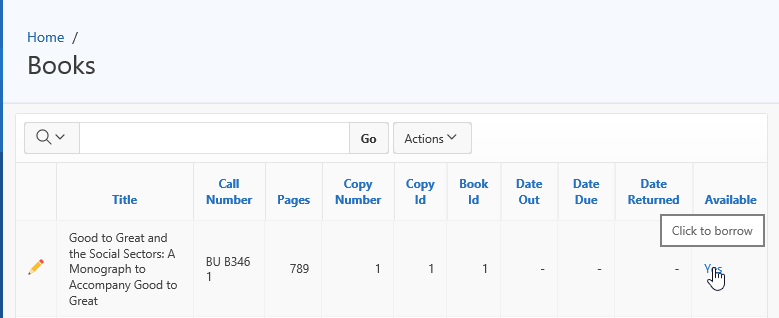
The **Link Buider** window will popup. Do the following settings:

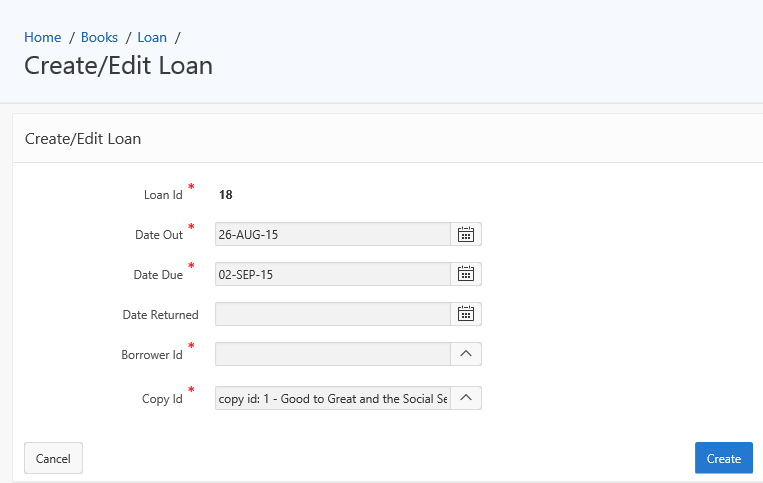


Click OK and then you also have to give a **Link Text** and **Link Attributes**.



Save changes and re-run the application to test.





Done! We have created the first 4 pages. The task “Loan a book” is basically done!

**Task 4 Improvement (20 marks)**

Is anything that you can make the task better?

For example,

* It can be to improve the display of information
* To improve data integrity by constraints
* To improve data integrity by triggers or
* To create a reasonable content for the homepage

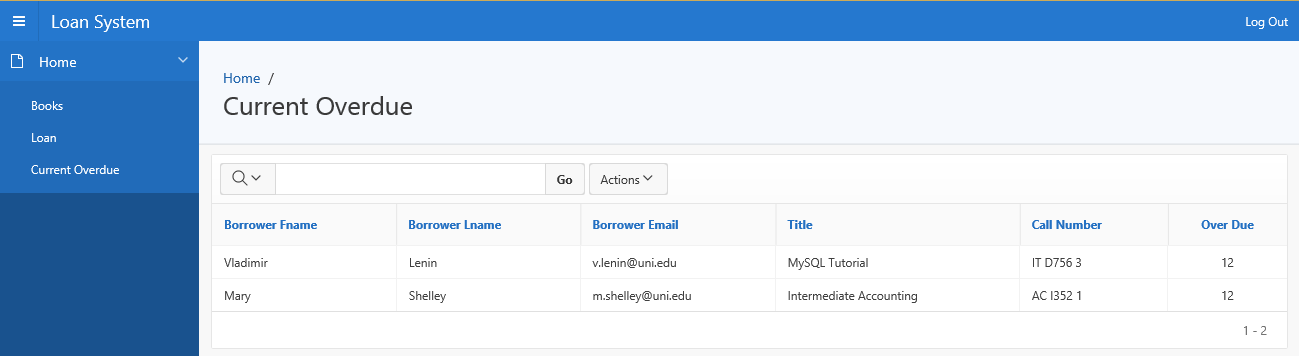
Let me know if you can make some improvement.

Continue to next page...

**Task 5 Overdue report (20 marks)**

The end user (librarian staff) is able to view all current overdue books. **Do it yourself**.

Note: A copy is overdue if **date\_returned** is **null** and the current date is at least more than 1 day from **date\_due**.



**SUBMISSION:**

Submission date: **By 5pm Friday Week10**.

Submit a doc file as jcnumber.doc (e.g. jc123456.doc) on LearnJCU containing:

- T**he** **details of the administrator, the developer and the end user accounts.**

- What you have achieved in “Loan a book”

- What you have achieved in “Improvement”

- What you have achieved in “Overdue Report”

- Anything else you would like to share with your lecturer