

Storing and Accessing Reports in the Database

Purpose

This tutorial shows you how to store and access a report in the database.

Time to Complete

Approximately 15 minutes

Topics

This tutorial covers the following topics:

- ☐ [Overview](#)
- ☐ [Prerequisites](#)
- ☐ [Creating a Table to Store Your Reports](#)
- ☐ [Generating and Storing a Report in the Database](#)
- ☐ [Accessing the Stored Report in the Database](#)
- ☐ [Summary](#)

Viewing Screenshots

 **Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)**

Note: Alternatively, you can place the cursor over an individual icon in the following steps to load and view only the screenshot associated with that step. You can hide an individual screenshot by clicking it.

Overview

Beginning with Oracle Application Express 3.1, you can store a report in a BLOB column in your database. In this tutorial, you will create a new table to store the reports that are generated and then create a report of the contents of the table.

[Back to Topic List](#)

Prerequisites

Before you perform this tutorial, you should:

1. Perform the [Creating a PDF Report with Multiple Queries](#) tutorial.
2. Perform the [Including Dynamic Images in Your Report](#) tutorial.

[Back to Topic List](#)

Creating a Table to Store Your Reports

In this section, you upload and run a script that creates a table to store the reports that you generate. Perform the following steps:

1. Open your browser and enter the following URL to log in to Oracle Application Express (change the <hostname> to localhost, your specific hostname or apex.oracle.com).

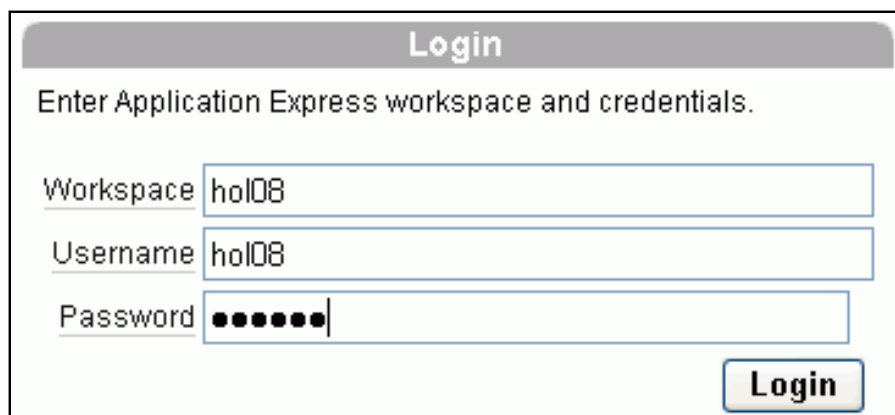
http://<hostname>:8080/apex

2. To log in to Oracle Application Express, enter the following details, and click **Login**.

Workspace: **<your workspace name>**

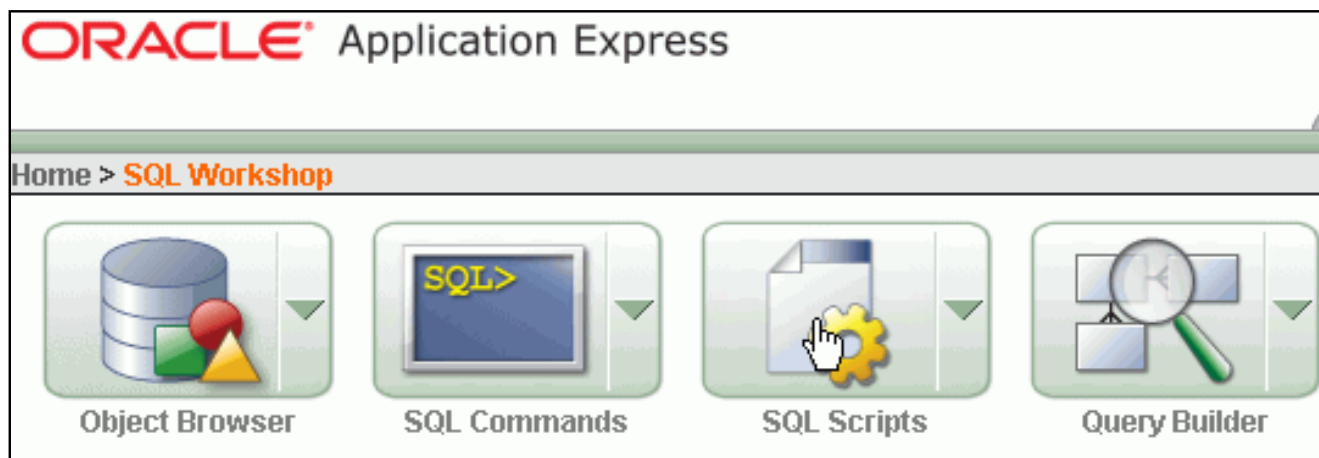
Username: **<your username>**

Password: **<your password>**

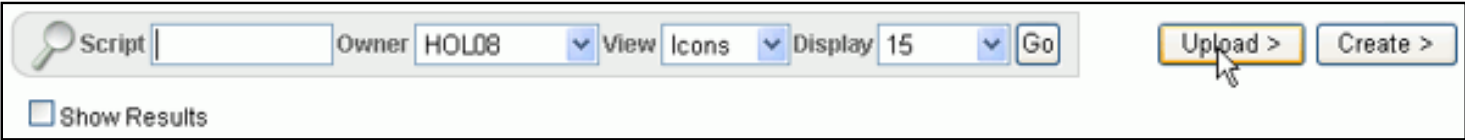


The screenshot shows the 'Login' page of Oracle Application Express. It has a title bar 'Login' and a subtitle 'Enter Application Express workspace and credentials.' Below this are three input fields: 'Workspace' with the value 'ho108', 'Username' with the value 'ho108', and 'Password' with masked characters '••••••'. A 'Login' button is located at the bottom right of the form.

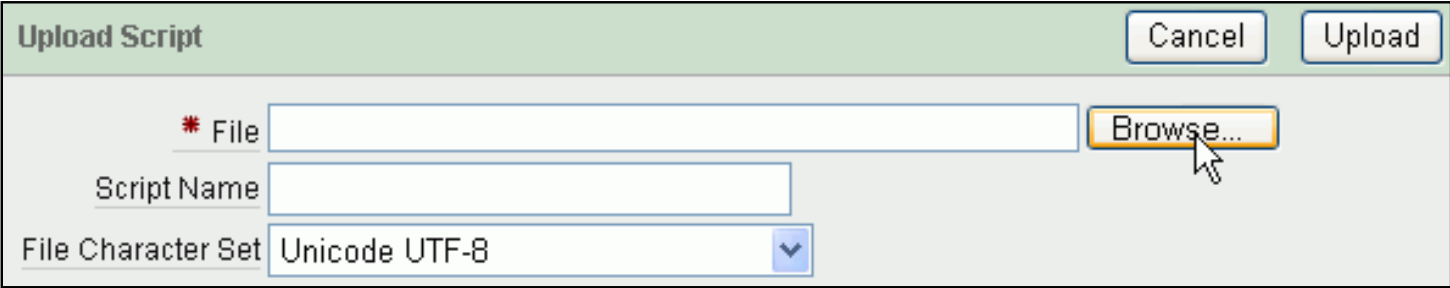
3. Click the **SQL Workshop** tab. Then click **SQL Scripts**.



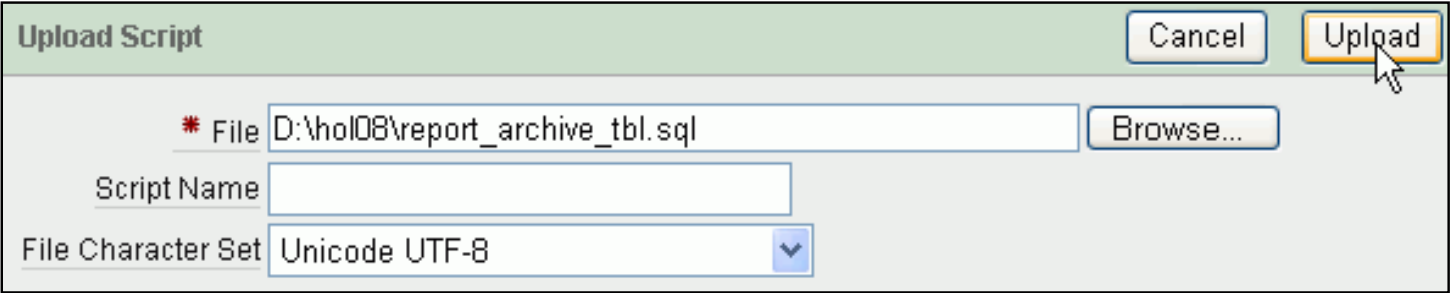
4. Click **Upload**.



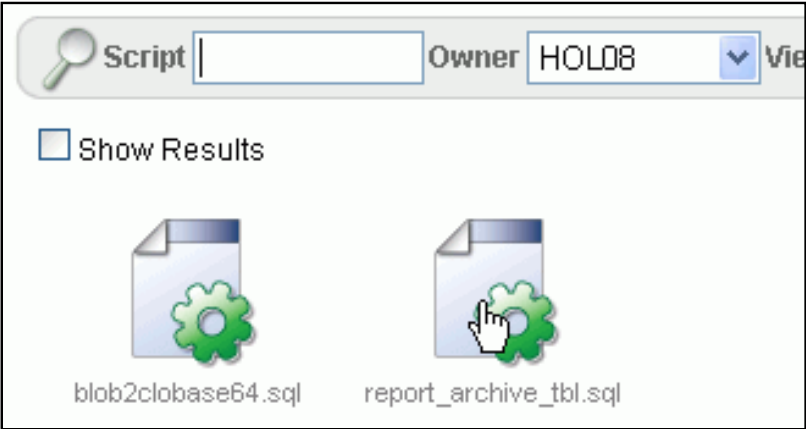
5. Click **Browse...** and select the [report_archive_tbl.sql](#) file (located where you downloaded and unzipped the prerequisite files) and click **Open**.



6. Click **Upload**.



7. Click the **report_archive_tbl.sql** script icon.



8. Click **Run**.

Script Name

report_archive_tbl.sql

Cancel

Download

Delete

Save

Run

Undo

Redo

Find

1

CREATE TABLE "REPORT_ARCHIVE"

2

{

3

"ID" NUMBER,

4

"FILENAME" VARCHAR2(4000),

5

"REPORT" BLOB,

6

"MIME_TYPE" VARCHAR2(255),

7

"CREATED_DATE" DATE,

8

"CREATED_BY" VARCHAR2(4000),

9

CONSTRAINT "REPORT_ARCHIVE_PK" PRIMARY KEY ("ID") ENABLE

10

);

11

12

create sequence "REPORT_ARCHIVE_SEQ"

13

start with 1

14

increment by 1

15

nocache

16

nocycle

17

noorder;

18

19

20

CREATE OR REPLACE TRIGGER "BI_REPORT_ARCHIVE"

21

before insert on report_archive

22

for each row

23

.

9. Click **Run** to confirm.

Run Script

Cancel

Run

You have requested to run the following script. Please confirm your request.

Script Name	report_archive_tbl.sql
Created	on 07/16/2008 08:22:09 AM by HOL08
Updated	on 07/16/2008 08:23:47 AM by HOL08
Number of Statements	3
Script Size in Bytes	531

10. Click the **View Results** icon.

☐	Script	Run By	Started ▼	Elapsed	Status	Statements	Bytes	View Results
	report_archive.tbl.sql	HOL08	1 seconds ago		Submitted			
row(s) 1 - 1 of 1								

11. The report_archive table was created successfully.

Number ▲	Elapsed	Statement	Feedback	Rows
1	1.86	CREATE TABLE "REPORT_ARCHIVE" ("ID" NUMBER, "FILENAM	Table created.	0
2	0.03	create sequence "REPORT_ARCHIVE_SEQ" start with 1 increment	Sequence created.	0
3	0.06	CREATE OR REPLACE TRIGGER "BI_REPORT_ARCHIVE" before i	Trigger created.	0
row(s) 1 - 3 of 3				

[Back to Topic List](#)

Generating and Storing a Report in the Database

In this section, you create a process that stores the report in the database. Perform the following steps:

1. Navigate to the Application Builder. Select the **Printing Application**.

 Application

View

Icons ▼

 Display

15 ▼

Go



OEHR Sample
Objects for OU -
109



Printing Application
- 108



Sample Application
- 107

2. Click the **Home** page.

Application: 108 - Printing Application



Run Application



Supporting Objects



Shared Components



Export / Import

Page

View

Icons

Display

15

Go

Create Page >



1 - Home



101 - Login

3. You need to create a process that will store the report in the database. Under Processes, click the Create icon.

Page

<

>

View

Definition

Go

RunCopyDeleteCreate >

Page Rendering



Page Processing



Page

Page Name: Home

Template: Application default

Title: Home

Header Text:

HTML Header:

Footer Text:

HTML Body:

Build Option:

Help Text: No help is available for this

Authorization: No

Page Group:

Cached: No

Regions



Computations



Validations



Processes



Branches



After Processing

10 Go To Page 0 Conditional

10 Go To Page 0 Conditional

4. Select **PL/SQL** and click **Next**.

Create Page Process Cancel Next >

Page: **1 - Home**

Select the category of the process you wish to create:

☒ PL/SQL ☐ Reset Pagination ☐ On Demand

☐ Session State ☐ Data Manipulation ☐ Web Services

☐ Form Pagination ☐ Close popup window

5. Enter **store_blobquery_report** for the Name and click **Next**.

Create Page Process Cancel < Previous Next >

Page: **1 - Home**

* Name

* Sequence

* Point ▼

* Type **PL/SQL anonymous block**

6. Enter the following code in the PL/SQL Page Process area and click **Next**.

```
declare
  l_report blob;
begin
  l_report := apex_util.get_print_document (
    p_application_id      => :APP_ID,
    p_report_query_name   => 'blobquery',
    p_report_layout_name  => 'blobquery',
    p_report_layout_type  => 'rtf',
    p_document_format     => 'pdf'
  );
  insert into report_archive (
    filename,
    mimetype,
    report,
    created_date,
    created_by
  ) values (
    'BLOB Query Search Results ('||to_char(sysdate, 'DDMonYYYY')||')',
    'application/pdf',
    l_report,
    sysdate,
    :USER
  );
end;
```

Create Page Process Cancel < Previous **Next >** Create Process

Page: **1 - Home**
Point: **On Submit - After Computations and Validations**

*** Enter PL/SQL Page Process**

```
declare
  l_report blob;
begin
  l_report := apex_util.get_print_document (
    p_application_id      => :APP_ID,
    p_report_query_name   => 'blobquery',
    p_report_layout_name  => 'blobquery',
    p_report_layout_type  => 'rtf',
    p_document_format     => 'pdf'
  );
end;
```

☐ Do not validate PL/SQL code (parse PL/SQL code at runtime only).

7. Click **Next**.

Create Page Process Cancel < Previous **Next >** Create Process

Page: **1 - Home**
Point: **On Submit - After Computations and Validations**

Success Message

Failure Message

8. Select **PRINT_REPORT_WITH_IMAGES** for When Button Pressed and click **Create Process**.

Create Page Process Cancel < Previous **Create Process**

Page: **1 - Home**
Point: **AFTER_SUBMIT**

When Button Pressed **PRINT_REPORT_WITH_IMAGES (Print Report With Images)**

Condition Type
- Process Not Conditional -

[PL/SQL] [item=value] [item not null] [request=e1] [page in] [page not in] [exists] [none] [never]

Expression 1

9. You also want to create a process to store the Multi Query report. Under Processes, click the Create icon.

The screenshot shows the Oracle APEX Page Editor interface. At the top, there is a search bar with 'Page 1' and a 'View' dropdown set to 'Definition'. To the right are buttons for 'Run', 'Copy', 'Delete', and 'Create >'. The main area is divided into two columns. The left column has a 'Page Rendering' tab with icons for a folder, a document, a table, a list, a grid, and a gear. Below this is the 'Page' section with fields for 'Page Name: Home', 'Template: Application default', 'Title: Home', 'Header Text:', 'Footer Text:', 'HTML Header:', 'HTML Body:', 'Help Text: No help is available for this', 'Authorization: No', 'Page Group:', and 'Cached: No'. At the bottom of the left column is the 'Regions' section with icons for a table, a document, and a plus sign. The right column has a 'Page Processing' tab with icons for a folder, a document, a table, a list, a grid, a gear, and an arrow. Below this are sections for 'Computations', 'Validations', 'Processes', and 'Branches'. The 'Processes' section is currently selected and shows a table with the following data:

Process Name	Process Type	Process Action
10 store_blobquery_report	PL/SQL anonymous block	Conditional

10. Select **PL/SQL** and click **Next**.

The screenshot shows the 'Create Page Process' dialog box. At the top, there is a title bar with 'Create Page Process' and buttons for 'Cancel' and 'Next >'. Below the title bar, it says 'Page: 1 - Home'. The main area is titled 'Select the category of the process you wish to create:'. There are seven radio buttons with corresponding icons: 'PL/SQL' (gear icon), 'Reset Pagination' (table with arrows), 'On Demand' (gear icon), 'Session State' (circular arrows), 'Data Manipulation' (table with arrows), 'Web Services' (globe), 'Form Pagination' (table with arrows), and 'Close popup window' (document with arrow). The 'PL/SQL' radio button is selected, and a mouse cursor is pointing at the 'Next >' button.

11. Enter **store_multiquery_report** for the Name and click **Next**.

Create Page Process

Page: 1 - Home

* Name

* Sequence

* Point

* Type **PL/SQL anonymous block**

12. Enter the following code in the PL/SQL Page Process area and click **Next**.

```
declare
  l_report blob;
begin
  l_report := apex_util.get_print_document (
    p_application_id      => :APP_ID,
    p_report_query_name   => 'multiquery',
    p_report_layout_name  => 'multiquery',
    p_report_layout_type  => 'rtf',
    p_document_format     => 'pdf'
  );
  insert into report_archive (
    filename,
    mimetype,
    report,
    created_date,
    created_by
  ) values (
    'Multi Query Search Results ('||to_char(sysdate,'DDMonYYYY')||')',
    'application/pdf',
    l_report,
    sysdate,
    :USER
  );
end;
```

Create Page Process

Cancel

< Previous

Next >

Create Process

Page: **1 - Home**
Point: **On Submit - After Computations and Validations**

* Enter PL/SQL Page Process

```
declare
  l_report blob;

begin

  l_report := apex_util.get_print_document (
    p_application_id      => :APP_ID,
    p_report_query_name   => 'multiquery',
    p_report_layout_name  => 'multiquery',
    p_report_layout_type  => 'rtf',
    p_document_format     => 'pdf'
  );
```

☐ Do not validate PL/SQL code (parse PL/SQL code at runtime only).

13. Click **Next**.

Create Page Process

Cancel

< Previous

Next >

Create Process

Page: **1 - Home**
Point: **On Submit - After Computations and Validations**

Success Message

Failure Message

14. Select **PRINT_REPORT** for When Button Pressed and click **Create Process**.

Create Page Process

Cancel

< Previous

Create Process

Page: 1 - Home

Point: AFTER_SUBMIT

When Button Pressed

PRINT_REPORT (Print Report)

Condition Type

- Process Not Conditional -

[PL/SQL] [item=value] [item not null] [request=e1] [page in] [page not in] [exists] [none] [never]

Expression 1

15. Click **Run**.

Page process store_multiquery_report created on page 1.

Page 1

< > View

Definition

Go

Run

Copy

Delete

Create >

Page Rendering

Page Processing

Page

Page Name: Home

Template: Application default

Title: Home

Header Text:

Footer Text:

HTML Header:

HTML Body:

Build Option:

Help Text: No help is available for this

Authorization: No

Page Group:

Cached: No

Regions

Computations

Validations

Processes

After Submit

10 store_blobquery_report PL/SQL Conditional anonymous block

20 store_multiquery_report PL/SQL Conditional anonymous block

16. Click **Print Report With Images**. When the dialogue to open or save the report appears, you can click **Cancel**.



17. Click **Print Report**. When the dialogue to open or save the report appears, you can click **Cancel**.



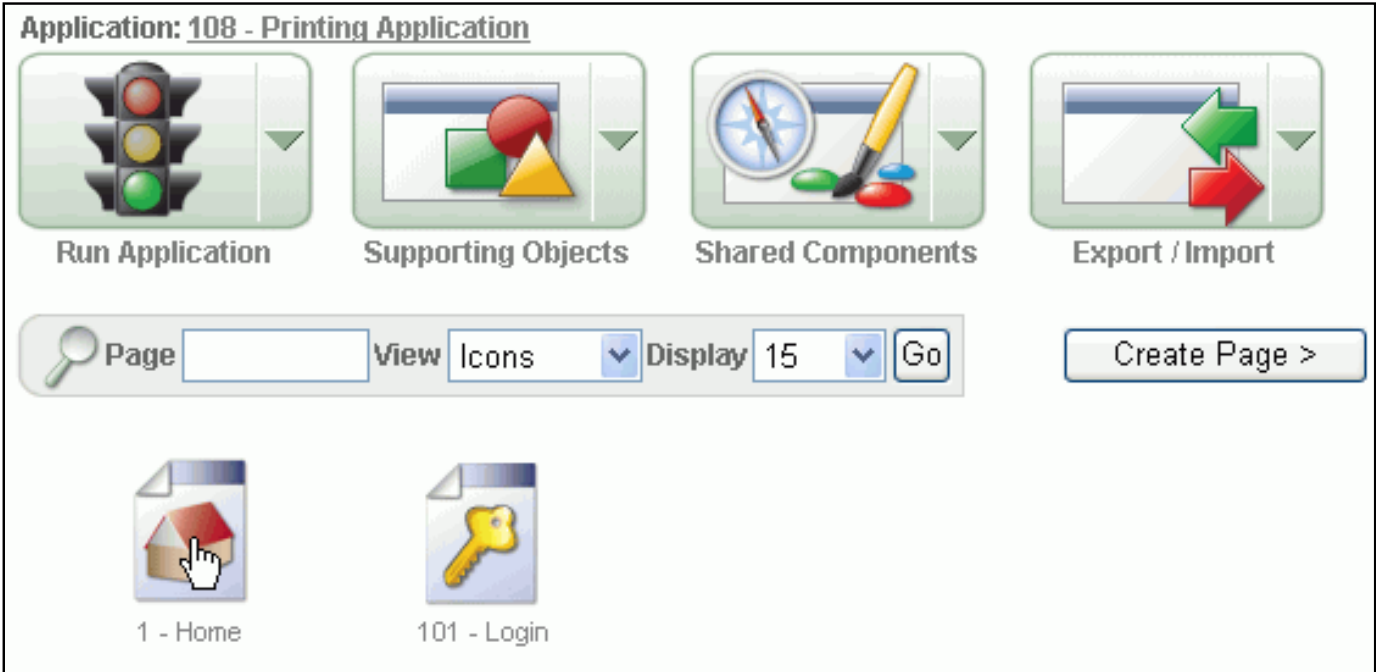
Note: Two rows were inserted into the database for each report you ran.

[Back to Topic List](#)

Accessing the Stored Report in the Database

In this section, you create a report of the data in the report_archive table in the database. Perform the following steps:

1. Navigate to your Printing Application page. Click the **Home** page.



- 3.** Select **Report** and click **Next**.

Create Region

Cancel

< Previous

Next >

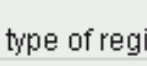
Page: 1 - Home

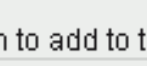
Identify the type of region to add to this page:

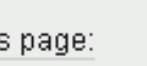
☐ HTML

☐ Multiple HTML

☒ Report



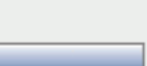


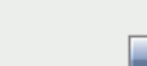


☐ Form

☐ Chart

☐ Breadcrumb







4. Make sure **SQL Report** is selected and click **Next**.

Create Region

Cancel< PreviousNext >

Page: 1 - Home

Report Implementation:

☒ SQL Report

☐ Interactive Report

☐ Report on collection containing Web service result

☐ Wizard Report

5. Enter **List of Stored Reports** for Title and click **Next**.

Create Region

Cancel< PreviousNext >

Page: 1 - Home

Region Source Type: SQL Query

* Title

List of Stored Reports

Region Template

Reports Region

Display Point

Page Template Body (3. items above region content)

[Body] [Pos.1] [Pos.2] [Pos.3] [Pos.4]

* Sequence

20

Column

1

6. Enter the following SQL in the SQL Query area and click **Create Region**.

```
SELECT id,  
       filename,  
       mimetype,  
       created_date,  
       created_by,  
       dbms_lob.getlength(report) report_length  
FROM   report_archive
```

Create Region

Cancel < Previous Next > Create Region

Page: **1 - Home**
Region Title: **List of Stored Reports**

Enter SQL Query or PL/SQL function returning a SQL Query:

```
SELECT id,  
       filename,  
       mimetype,  
       created_date,  
       created_by,  
       dbms_lob.getlength(report) report_length  
FROM   report_archive
```

Query Builder

Columns Headings: ☒ Derived from query columns ☐ Generic columns

Max.Columns

7. The report was created. You need to make a few changes to the Report Attributes. Under Regions, select the **Report** link.

Page Rendering

Page

Page Name: Home

Template: Application default

Title: Home

Header Text:

HTML Header:

Footer Text:

HTML Body:

Build Option:

Help Text: No help is available for this

Authorization: No

Page Group:

Cached: No

Regions

Display Point: Page Template Body (3) ▾ ▴

10 ☐ Home

HTML

20 ☒ List of Stored Reports

Report

Display Point: Region Position 01

1 >>> Breadcrumbs

Breadcrumb Entry

Buttons

Region: Home ▾ ▴

10 Print Report

Submit as "PRINT_REPORT"

8. Under Headings Type, click **Custom**. Change the name of the Report Length header to **Report**. Deselect the Show check box for **MIMETYPE**. You also want to create a link to download the report in the Report Length column. Select the edit icon for Report Length.

Region Definition
Report Attributes
Print Attributes

Region Name: List of Stored Reports

Show All
Column Attributes
Layout and Pagination
Sorting
Messages
Report Export
Break Formatting

Column Attributes

Headings Type:
☐ Column Names
☐ Column Names (InitCap)
☒ Custom
☐ PL/SQL
☐ None

Alias	Link	Edit	Heading	Column Alignment	Heading Alignment	Show
ID			Id	left		☒
FILENAME			Filename	left		☒
MIMETYPE			Mimetype	left		☐
CREATED_DATE			Created Date	left		☒
CREATED_BY			Created By	left		☒
REPORT_LENGTH			Report	left		☒

When moving the last column further down, it will show up as the first column of your report.
When moving the first column up, it will be moved to the end of your report.

9. Under Column Formatting, enter the following string in the Number / Date Format field and click **Apply Changes**.

DOWNLOAD:REPORT_ARCHIVE:REPORT:ID::MIMETYPE:FILENAME:::inline:[Download]

Note this string contains the following parameters:

<Format Mask>:<BLOB Table>:<BLOB Column>:<Primary Key Column 1>::<MIME Type Column>:
<Filename Column>:::<Content Disposition>:<Download Link Text>

Column Attributes: REPORT_LENGTH Cancel Apply Changes <

Show All Definition Formatting Tabular Form Element Lists of Values Link Authorization Conditions Comments

Column Definition

Column Name **REPORT_LENGTH**

Column Heading

Show Column Heading Alignment

Compute Sum Column Alignment

Sortable Column

Include In Export

Column Formatting

Number / Date Format

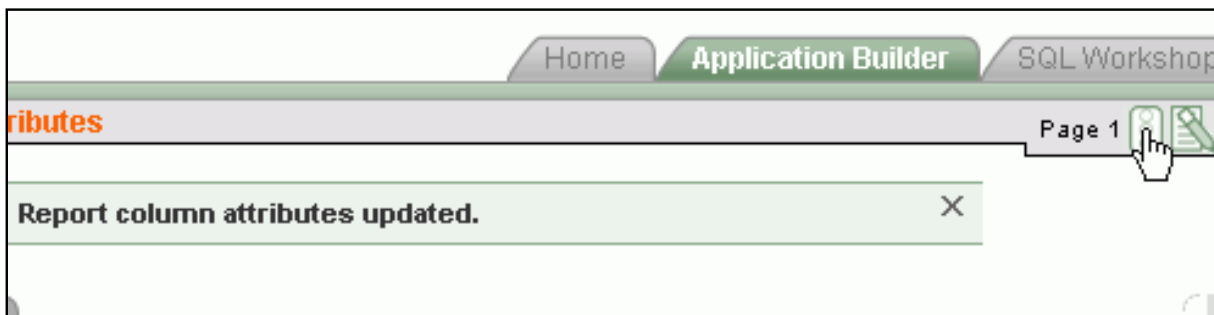
CSS Class

CSS Style

Highlight Words

HTML Expression

- Click **Run Page**.



- The list of reports that you ran in the previous section of this tutorial are in the list. To view a report, click the **[Download]** link.

Home

Print Report

Print Report With Images

List of Stored Reports

Id	Filename	Created Date	Created By	Report
5	Multi Query Search Results (16Jul2008)	16-JUL-2008	HOL08	[Download]
4	BLOB Query Search Results (16Jul2008)	16-JUL-2008	HOL08	[Download]

1 - 2

12. The report is displayed successfully.

List of Employees					
First Name	Last Name	Email	Hire Date	Job	Salary
Steven	King	SKING	17-JUN-1987	AD_PRES	24000
Neena	Kochhar	NKOCHHAR	21-SEP-1989	AD_VP	17000
Lex	De Haan	LDEHAAN	13-JAN-1993	AD_VP	17000
Alexander	Hunold	AHUNOLD	03-JAN-1990	IT_PROG	9000
Bruce	Ernst	BERNST	21-MAY-1991	IT_PROG	6000
Diana	Lorentz	DLORENTZ	07-FEB-1999	IT_PROG	4200
Kevin	Mourgos	KMOURGOS	16-NOV-1999	ST_MAN	5800
Trenna	Rajs	TRAJS	17-OCT-1995	ST_CLERK	3500

[Back to Topic List](#)

In this tutorial, you learned how to:

- ☒ Create a table to store your reports
- ☒ Generate and store a report in the database
- ☒ Access the stored report in the database

[Back to Topic List](#)

 **Place the cursor over this icon to hide all screenshots.**