

---

## C7. SQL Projection and Selection.

### Input SQL parameters from Web Page Items

As defined by relational algebra, the projection operator ( $\pi$ ) – deletes unwanted columns from a relation R and discards duplicates. The selection operator ( $\sigma$ ) – selects a subset of rows from R based on a selection condition. Both operators are implemented using SQL **SELECT** statement.

#### 1.1. Projecting columns in SQL

The general syntax of projection operation in SQL is:

```
SELECT [ALL | DISTINCT] prj_list  
FROM table;
```

The projection list (prj\_list) is a comma-separated list of fields from the table. As an alternative it can be replaced by the sign ‘\*’ to obtain 1:1 projection. In this case the name and order of the result fields will be the same as in original table. Fields can be renamed using the **AS** keyword.

The **DISTINCT** keyword force duplicates elimination. Default is **ALL**.



**ACTIVITY 1:** Using **SQL Workshop** -> **SQL Commands** run the following SQL commands:

```
SELECT id AS PersonID, name FROM Persons;  
SELECT DISTINCT name FROM Persons;  
SELECT * FROM Persons;
```

The general syntax of selection operation in SQL is:

```
SELECT prj_list
FROM table
WHERE select_condition;
```

```
SELECT * FROM Persons WHERE dep_id=1;
```

**WHERE** name **LIKE** 'J%' **AND** dep\_id=1;

---

## 1.3. Input SQL parameters from Web Page Items

APEX uses HTTP POST requests to send user data read from page items. It implements an SQL injection mechanism to pass corresponding POST variables to SQL interpreter. The syntax used in this case includes a colon sign ‘:’ as prefix of the POST variable. The variable name is the same as the APEX item name.



**ACTIVITY 3:** Create an interactive report on Persons table. The report will filter data using SQL select mechanism based on a dep\_id value read from the user. For reading the user input add an input field and a validation button.

- Step 1

From **Application Builder -> Create** add a new application named SQL\_DML.

- Step 2

Add a new page for the SQL\_DML application using the **Create Page** button. Choose the Interactive Report and follow the steps illustrated bellow:

Page and Region Attributes   Tab Options   SQL Query   Confirm

Cancel   Next >

Identify a page number and name. If the page number you specify does not exist, the wizard creates the page for you.

Application: 23580 - SQL\_DML

\* Page Number: 2

\* Page Name: SQL Selection

\* Region Template: Region without Buttons and Titles

\* Region Name: SQL Selection

Breadcrumb: do not use breadcrumbs on page

---

Page and Region Attributes
●
SQL Query
Confirm

← Cancel
Next >

Page: **2**

Tab Options:

- ☒ Do not use tabs
- ☐ Use an existing tab set and create a new tab within the existing tab set.
- ☐ Use an existing tab set and reuse an existing tab within that tab set.

> Tabs

---

Page and Region Attributes
Tab Options
●
SQL Query
Confirm

← Cancel
Next >

\* Enter a SQL SELECT statement

SELECT \* FROM Persons WHERE dep\_id=10\_ID

Query Builder

Link to Single Row View: No ▾

Uniquely Identify Rows by: ROWID ▾

Unique Column:

> SQL Query Example

- **Step 3**

Edit the report page. Add a new number input field from the **Create Page Item** context menu of the report item (named here 'SQL Selection'). Give the name D\_ID to this input, as presented below:

**Page Rendering**

- SQL Selection
  - Before Header
  - After Header
  - Before Regions
  - Regions
    - Body (3)
      - SQL Selection**

Edit  
Edit Report Attributes  
Edit Saved Reports  
Edit Print Attributes  
Create  
Create Page Item  
Create Page Item Button  
Create Region Button  
Create Dynamic Action  
Create Sub Region  
Copy  
Rename  
Expand All  
Collapse All
- After R
- Before
- After F
- Dynam

**Page Processing**

- After Submit
- Validating
- Processing
- After Processing
  - AJAX Callbacks

●
●
●


Item Type    Display Position and Name    Source


---


Cancel
Next >


Page: **2 - SQL Selection**


Select Item Type:


☐ Checkbox  



☐ Color Picker  



☐ Date Picker  



☐ Display Image  



☐ Display Only  



☐ File Browse...  



☐ Hidden  


☐ List Manager  


☒ Number Field  


☐ Password  


☐ Plug-ins  


☐ Popup LOV  


●
●
●
●
●

Item Type    **Display Position and Name**    Item Attributes    Settings    Source

---

< Cancel
Next >

Page: **2 - SQL Selection**

Display As: **Number Field**

\* Item Name:

\* Sequence:

\* Region:

> Items

●
●
●
●
●

Item Type    Display Position and Name    **Item Attributes**    Settings    Source

---

< Cancel
Next >

Page: **2 - SQL Selection**

Item Name: **D\_ID**

Display As: **Number Field**

Label:  [Clear]

Field Width:

Template:

> Existing Labels


Item Type   Display Position and Name   Item Attributes   **Settings**   Source


← Cancel Next >

Page: **2 - SQL Selection**

Item Name: **D\_ID**

Display As: **Number Field**

Value Required: **No** 

Format Mask: 999G999G999G999G999G999G990 

Minimum Value:

Maximum Value:

Number Alignment: **Left**

Search Dialog - Google Chrome

apex.oracle.com/pls/apex/www\_flow.show?p\_flow\_

Search Close

\$5,234.10  
5,234.10  
5,234.1000  
5,234  
5,234.10-  
-5,234.10  
<5,234.10>

Row(s) 1 - 7

javascript:passBack('999G999G999G999G999G999G990');

Item Type   Display Position and Name   Item Attributes   Settings   **Source**

← Cancel Create Item

Identify the source of the item. If the item source is null the default value will be used.

Page: **2 - SQL Selection**

Item Name: **D\_ID**

Display As: **Number Field**

Source Used: **Only when current value in session state is null**

\* Source Type: **Always Null**

Default:

Item Default Type: **Static Text with Session State Substitutions**

- **Step 4**

Edit the report page. Add a new Button from the **Create Page Item Button** context menu of the report item (named here 'SQL Selection'). Give the name P2\_SEARCH to this button, as presented below:

Button Region   Button Position   **Button Attributes**   Display Properties   Action When Clicked   Conditional Display

← Cancel Create Button   Next >

Page: **2 - SQL Selection**

Region: **SQL Selection**

\* Button Name:

\* Label:

Button Style: **HTML Button**

Button Attributes:

Button Region

Button Position

Button Attributes

Display Properties

**Action When Clicked**

Conditional Display

<

Cancel

Create Button

Next >

Page: 2 - SQL Selection

Region: SQL Selection

Button Name: P2\_SEARCH

Action: 

Submit Page

Execute Validations: 

Yes

Button Region

Button Position

Button Attributes

Display Properties

Action When Clicked

**Conditional Display**

<

Cancel

Create Button

Page: 2 - SQL Selection

Region: SQL Selection

Button Name: P2\_SEARCH

Condition Type: 

- Button NOT Conditional -

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

- Step 5
- Run the report page and test the user input.

SQL\_DML

Home

Dep\_Id

1

Search for Department Members

Q-

Go

Actions

Id	Name	Email	Bday	Sex	Dep Id
101	John Smith	j.smith@mycomp.com	07-FEB-87	M	1

1 - 1