

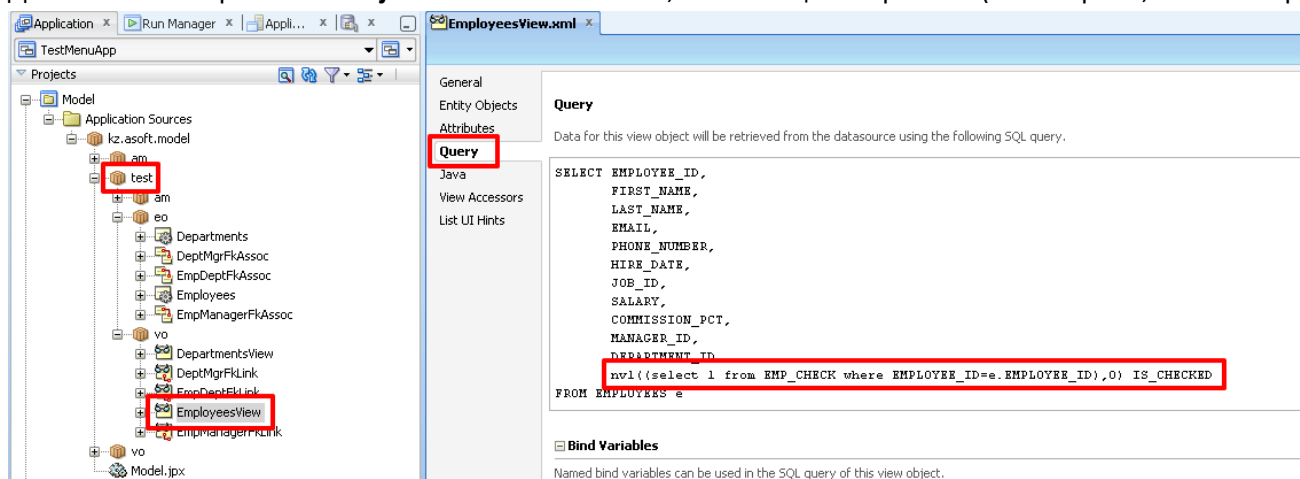
Про полезный метод doDML в EntityObject

Реализуем мульти чекбоксы (заменяем UPDATE на INSERT или DELETE в другой таблице)

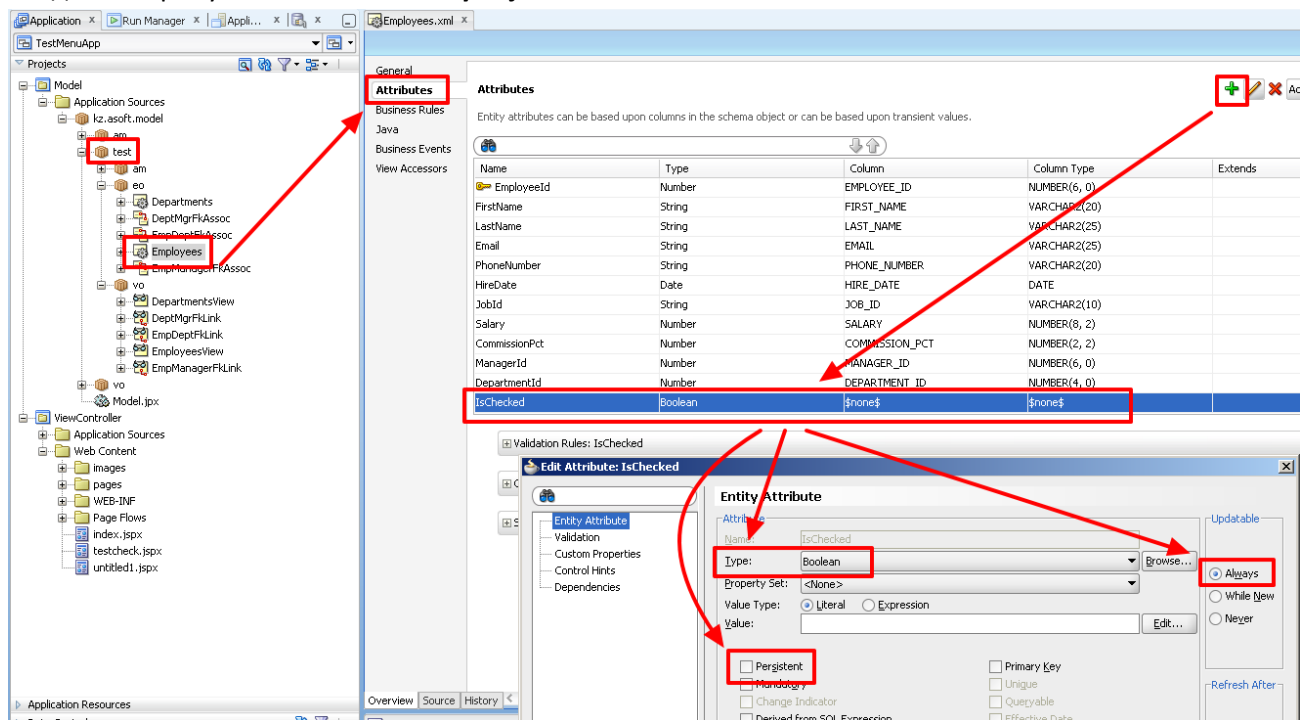
Для примера создана таблица EMP_CHECK, содержащая сведения об отмеченных сотрудниках:

```
CREATE TABLE HR.EMP_CHECK (  
    EMPLOYEE_ID NUMBER(6,0) NOT NULL PRIMARY KEY  
);
```

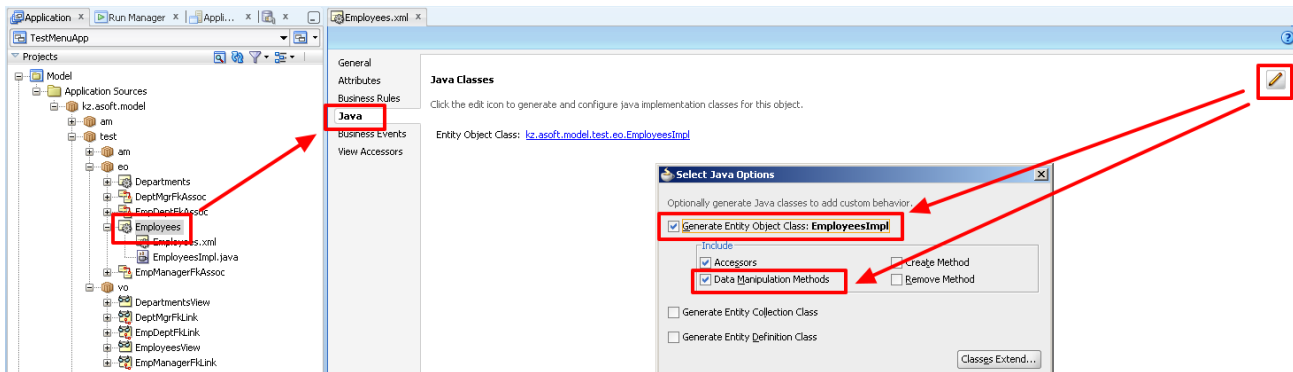
Добавляем в запросе **ViewObject** вычисляемое поле, отвечающее за флажок (1 – выбрано, 0 – не выбрано):



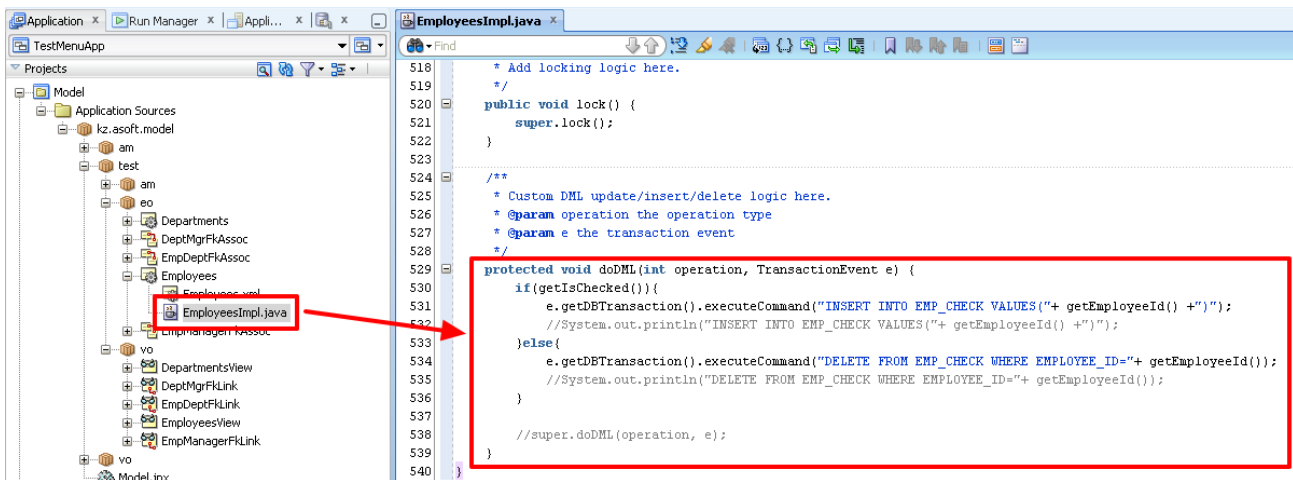
Создание атрибута IsChecked в EntityObject



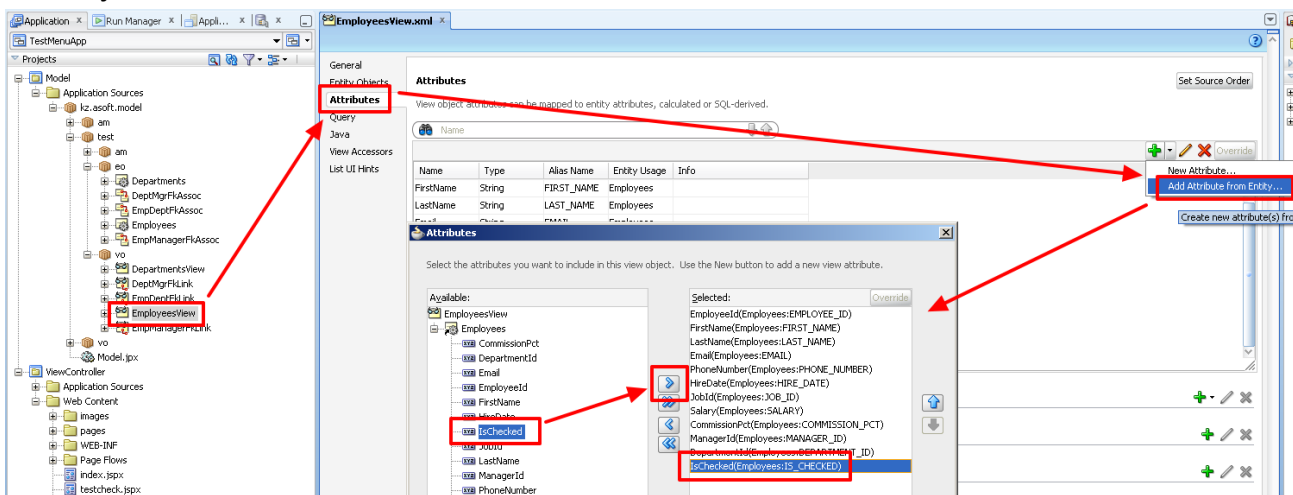
Создаем Java-класс для EntityObject



Переписываем метод doDML (подсовываем свои DML команды)



Удаляем старый атрибут IsChecked из ViewObject и после переносим атрибут IsChecked из EntityObject во ViewObject



Привязываем атрибут IsChecked к столбцу IS_CHECKED

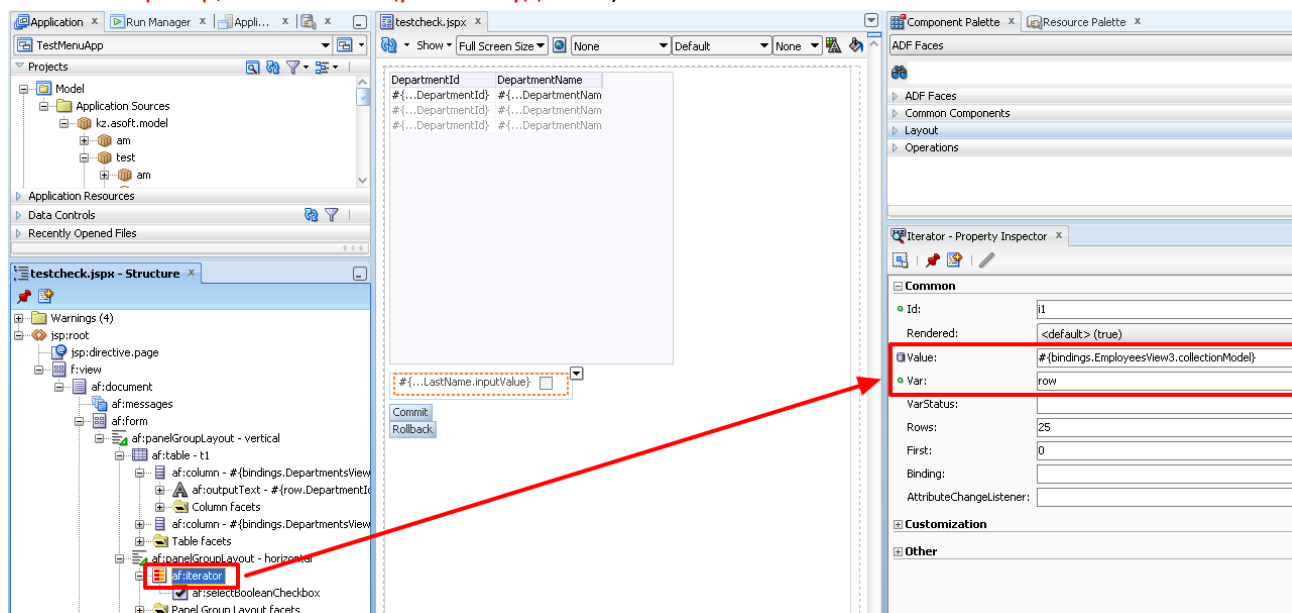
The screenshot shows the Oracle ADF IDE interface. On the left, the 'Projects' pane shows the 'TestMenuApp' project with a tree view of application sources. The 'EmployeesView' is highlighted. In the center, the 'Attributes' pane displays a table of attributes for the 'Employees' entity. The 'IsChecked' attribute is highlighted in red. On the right, the 'Edit Attribute: IsChecked' dialog is open, showing the 'View Attribute' tab. The 'Mapped to Column or SQL' checkbox is checked, and the 'Alias' is set to 'IS_CHECKED'.

Name	Type	Alias Name	Entity Usage	Info
FirstName	String	FIRST_NAME	Employees	
LastName	String	LAST_NAME	Employees	
Email	String	EMAIL	Employees	
PhoneNumber	String	PHONE_NUM...	Employees	
HireDate	Date	HIRE_DATE	Employees	
JobId	String	JOB_ID	Employees	
Salary	Number	SALARY	Employees	
CommissionPct	Number	COMMISSIO...	Employees	
ManagerId	Number	MANAGER_ID	Employees	
DepartmentId	Number	DEPARTMEN...	Employees	
IsChecked	Boolean	IS_CHECKED	Employees	

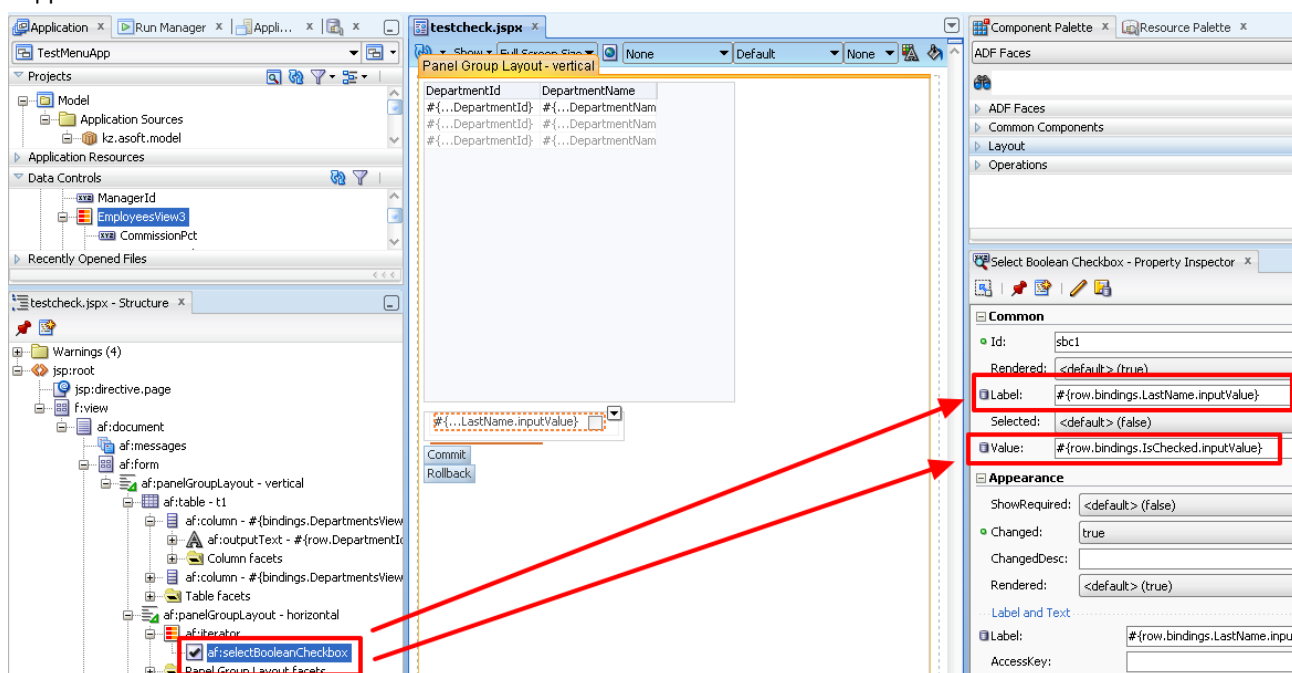
Создание страницы

The screenshot shows the Oracle ADF IDE interface. On the left, the 'Projects' pane shows the 'TestMenuApp' project with a tree view of application sources. The 'testcheck.jspx' file is highlighted. In the center, the 'testcheck.jspx' page is shown in design view. The page contains a data control named 'EmployeesView3' with a table of columns. The 'DepartmentId' and 'DepartmentName' columns are highlighted in red. The 'testcheck.jspx' file is also highlighted in the 'Projects' pane.

Настройка итератора – задаем свойства Value и Var (чтобы не создавать объекты данных вручную я сначала скидываю временную таблицу которая строится по **EmployeeView3**, потом копирую значение из ее свойства Value в итератор, после таблицу можно удалить)



Задаем свойства чекбокса:

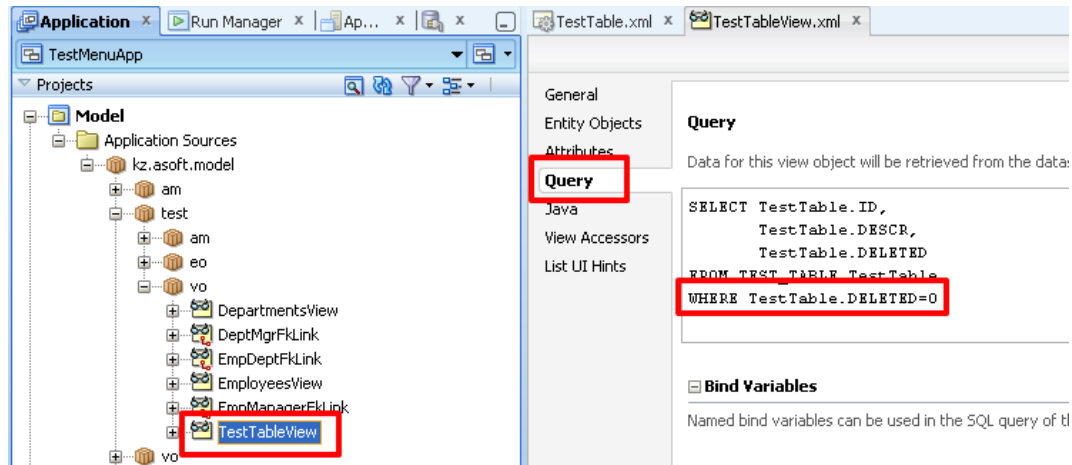


Реализация логического удаления при помощи метода doDML

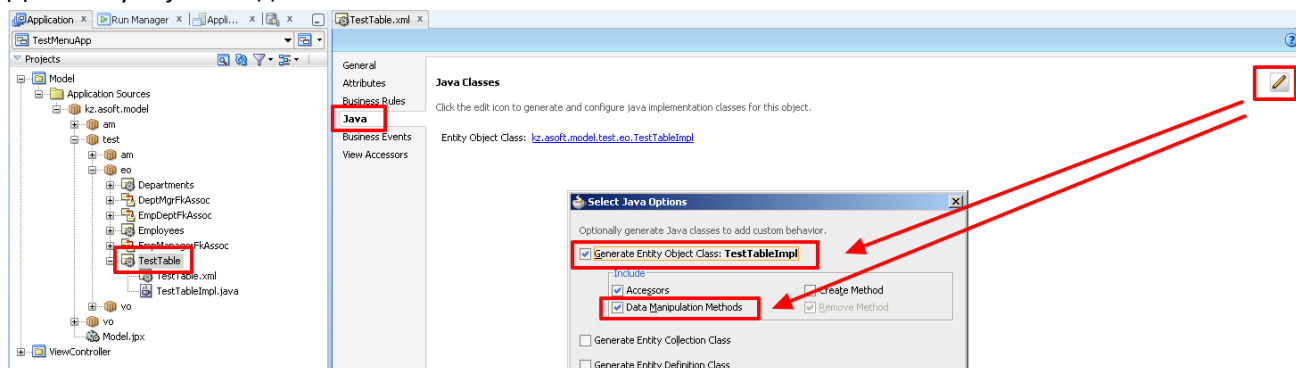
Для примера реализуем таблицу с полем **DELETED**, в котором будет отметка о логическом удалении строки:

```
CREATE TABLE HR.TEST_TABLE (  
    ID          NUMBER NOT NULL PRIMARY KEY,  
    DESCR       VARCHAR2(20 CHAR) NOT NULL,  
    DELETED     NUMBER(1,0) DEFAULT 0 NOT NULL  
);
```

Во ViewObject будет отражать только записи у которых DELETED=0:



Для EntityObject создаем Java-класс:



Переопределяем метод **remove** и меняем код метода **doDML**:

```
/**  
 * переопределяем метод remove() и флажки задаем в нем  
 */  
@Override  
public void remove() {  
    setDeleted(new Number(1)); // помечаем поле - отметка об логическом удалении  
  
    super.remove();  
}  
  
/**  
 * переписываем doDML, теперь вместо DELETE делается UPDATE  
 */  
protected void doDML(int operation, TransactionEvent e) {  
    if (operation == DML_DELETE) {  
        operation = DML_UPDATE;  
    }  
  
    super.doDML(operation, e);  
}
```