

# Oracle APEX 5.1 Fundamentals II

APEX in de praktijk

Copyright Learco Brizzi  
Itium B.V.

## Inhoudsopgave

Inleiding.....	3
1. Workspace maken.....	4
2. Tools - Browser add-ons .....	6
3. Back Office basis .....	8
4. Back Office schermen.....	10
5. Universal Theme/Templates .....	23
6. Current record.....	29
7. Error handling .....	31
8. Presentatie en Logica.....	34
9. Modal dialogs.....	35
10. JavaScript en jQuery.....	40
11. Authentication, Front Office, Badges.....	42
12. Authorization .....	46
13. Session State Protection .....	48
14. Cross Site Scripting (XSS).....	49
15. Files in de database.....	50
16. Integratie.....	53
17. Plug-Ins.....	57
18. Dynamic Actions.....	58
19. Interactive Reports.....	61
20. Dictionary Views .....	62
Extra oefeningen.....	63
Bijlage: Model .....	64
Bijlage: generieke oplossing voor multiselect en shuttle .....	69

## Inleiding

In de komende oefeningen zullen we een simpel Document Management Systeem maken. Dit bestaat uit een front office applicatie, waarin eindgebruikers documenten kunnen zoeken en uploaden. In de back office applicatie kunnen o.a. gebruikers, pagina's, toegangsrechten en management rapportages worden gemaakt en bekeken.

De oefeningen in deze cursus zijn niet voor een beginnende APEX ontwikkelaar opgezet. Er wordt bij de diverse beschreven stappen niet altijd uitputtend beschreven op welke knoppen er precies gedrukt moet worden, alleen als dit noodzakelijk is om de stof beter te begrijpen. Als er bijvoorbeeld een pagina moet worden gemaakt wordt soms volstaan met 'maak een pagina met paginanummer 5' i.p.v. de gehele wizard stap voor stap te beschrijven.

Indien er in de oefeningen wordt verwezen naar scripts en documenten, zijn deze te vinden op de virtuele machine.

Toegang tot internet is mogelijk gedurende de cursus.

Er wordt gewerkt met de volgende software: 11gR2 XE, SQL Developer en Windows 7. Alle usernames en passwords voor Windows en Oracle zijn 'cursist'.

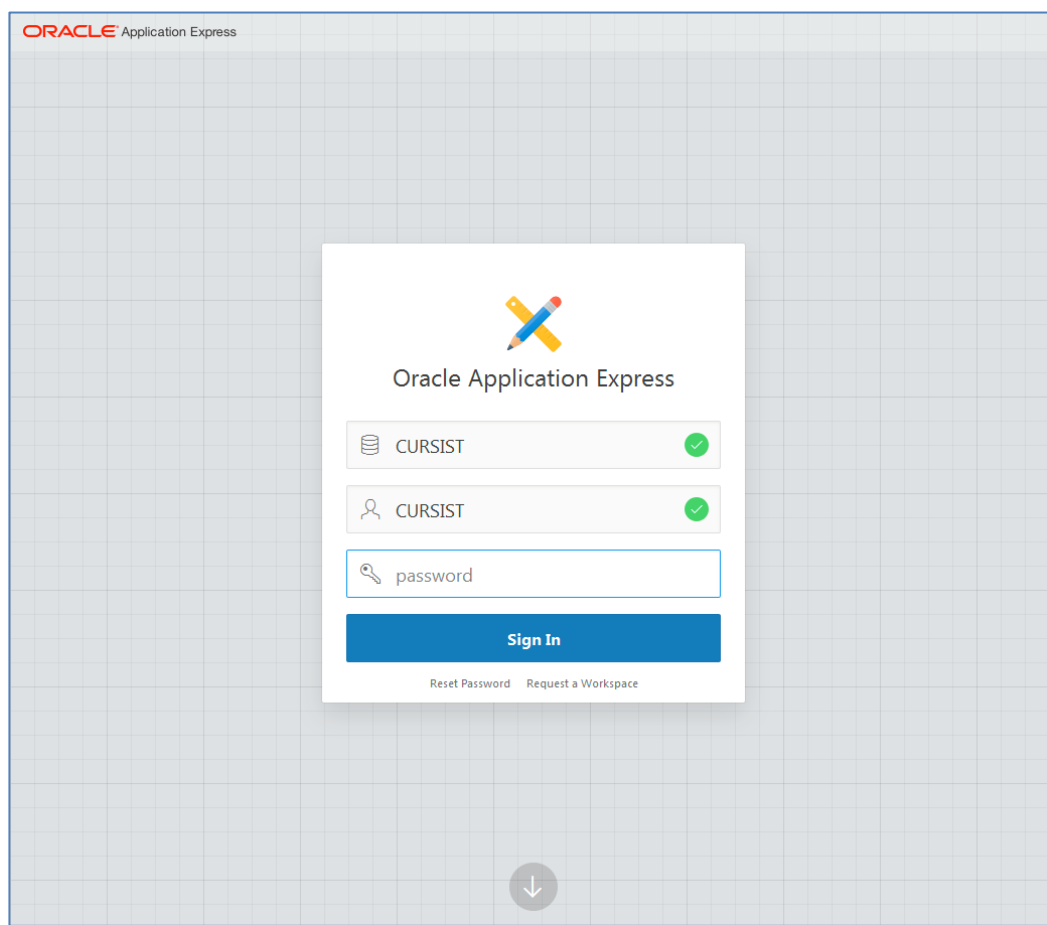
## 1. Workspace maken

Voordat we beginnen met de oefening moeten nog wat zaken worden uitgevoerd:

- Inloggen op het systeem met de gegevens in de map.
- Bij de persoonlijke gegevens moet eventueel de geboortedatum worden ingevuld. Dit komt later terug op het certificaat.
- De virtuele machine moet worden gestart (E:\vms\apex).
- In de virtuele machine moet eventueel de database worden gestart.

Om de oefening te kunnen maken moet er een Workspace worden gemaakt. Voer daartoe onderstaande stappen uit:

- Applicatie 4550 moet worden opgestart in de browser. Klik daartoe op de **Application Express Login** link op het bureaublad. De homepage van APEX wordt nu getoond.



- Log in met workspace 'internal', username 'admin' en als password 'cursist'.
- Het kan zijn dat er wordt gevraagd om het password te wijzigen. Doe dit dan en log in.
- Je krijgt nu een overzicht van de APEX instance, die hier kan worden beheerd.
- Klik op 'Manage Workspaces'.
- Op het scherm dat nu verschijnt kan een workspace worden gemaakt, door op 'Create Workspace' te klikken.
- In de wizard die nu verschijnt moeten de volgende gegevens worden ingevuld: Workspace Name: 'cursist', Re-use existing schema: 'No', Schema Name: 'cursist', Schema Password: 'cursist' en voor Space Quota voldoet 25 MB.
- Administrator Username: 'cursist', Administrator Password 'cursist' en een email adres.
- Na op Create Workspace geklikt te hebben, wordt de workspace nu aangemaakt. Dit kan even duren. Log uit.

- Log nu in met de zojuist gemaakte workspace en gebruiker. Indien er wordt gevraagd om het password te wijzigen, verander hem dan weer in 'cursist'.

In deze workspace kunnen nu applicaties worden gemaakt.

The screenshot shows the Oracle Application Express interface. At the top, there is a navigation bar with the Oracle logo and 'Application Express' text, followed by dropdown menus for 'App Builder', 'SQL Workshop', 'Team Development', and 'Packaged Apps'. To the right of these are search and help icons.

Below the navigation bar, there are four main application tiles: 'App Builder' (blue icon), 'SQL Workshop' (green icon), 'Team Development' (yellow icon), and 'Packaged Apps' (orange icon). Each tile has its name written below the icon.

The main content area is divided into three columns:
 

- Top Apps:** This column is currently empty.
- Top Users:** This column shows a bar chart for the user 'cursist' with a value of 1. The user's profile icon (a yellow circle with 'CU') is shown to the left of the bar.
- News and Messages:** This column contains a list of messages:
  - 'WS message' (Workspace Message)
  - 'Hallo: system message' (System Message)
  - '10 JUN Nieuws!' (News!) from 'cursist' dated '1.4 years ago'.

On the right side of the dashboard, there is a sidebar with the following sections:
 

- About:** A brief description of Application Express as a rapid Web application development tool.
- Dashboard:** A section with four circular gauges:
  - 'Applications' with the number 9.
  - 'Tables' with the number 63.
  - 'Features' with the number 0.
  - 'Packaged Apps' with the number 1.
- Available Updates:** A section indicating 'System is up-to-date'.
- Accessibility Mode:** A dropdown menu currently set to 'Standard'.

At the bottom of the page, there is a footer with the text 'Page events: 1 hours ( Edit Reporting Timeframe )', user information 'cursist', a globe icon, and the text 'Copyright © 1999, 2017, Oracle. All rights reserved.' and 'Application Express 5.1.3.00.05'.

## 2. Tools - Browser add-ons

Om het ontwikkelen wat makkelijker en overzichtelijker te maken in de Page Designer, zullen we een tool (browser add-on) installeren.



### Greasemonkey

De eerste add-on is Greasemonkey. Dit is al een wat oudere add-on, maar ook voor APEX kan hij z'n diensten bewijzen. Met Greasemonkey is het mogelijk om nadat een site geladen is, er specifieke JavaScript en dus eigen CSS op 'los te laten'. Hierdoor is het mogelijk om de interface een geheel eigen *look and feel* te geven. Zoek in Firefox naar de Greasemonkey add-on en installeer/activeer hem. Voeg vervolgens een nieuw **user script** toe. De volgende URL moet worden toegevoegd als 'include': **\*/f?p=4000:4500\*** De sterretjes horen er bij en fungeren als wildcard. Nu wordt elke site, waar deze URL in voorkomt door Greasemonkey nabewerkt. Het script dat moet worden gebruikt is meegeleverd onder de naam **greasemonkey.txt**.

Bekijk het script en probeer het enigszins te begrijpen. Als we nu naar de Page Designer gaan en eventueel een reload doen, zullen we zien dat de *look and feel* zijn aangepast. Uiteraard kan je dit naar eigen smaak aanpassen en uitbreiden.

Mochten de aanpassingen niet bevallen, dan kunnen ze makkelijk weer uit worden gezet.

### Optioneel:



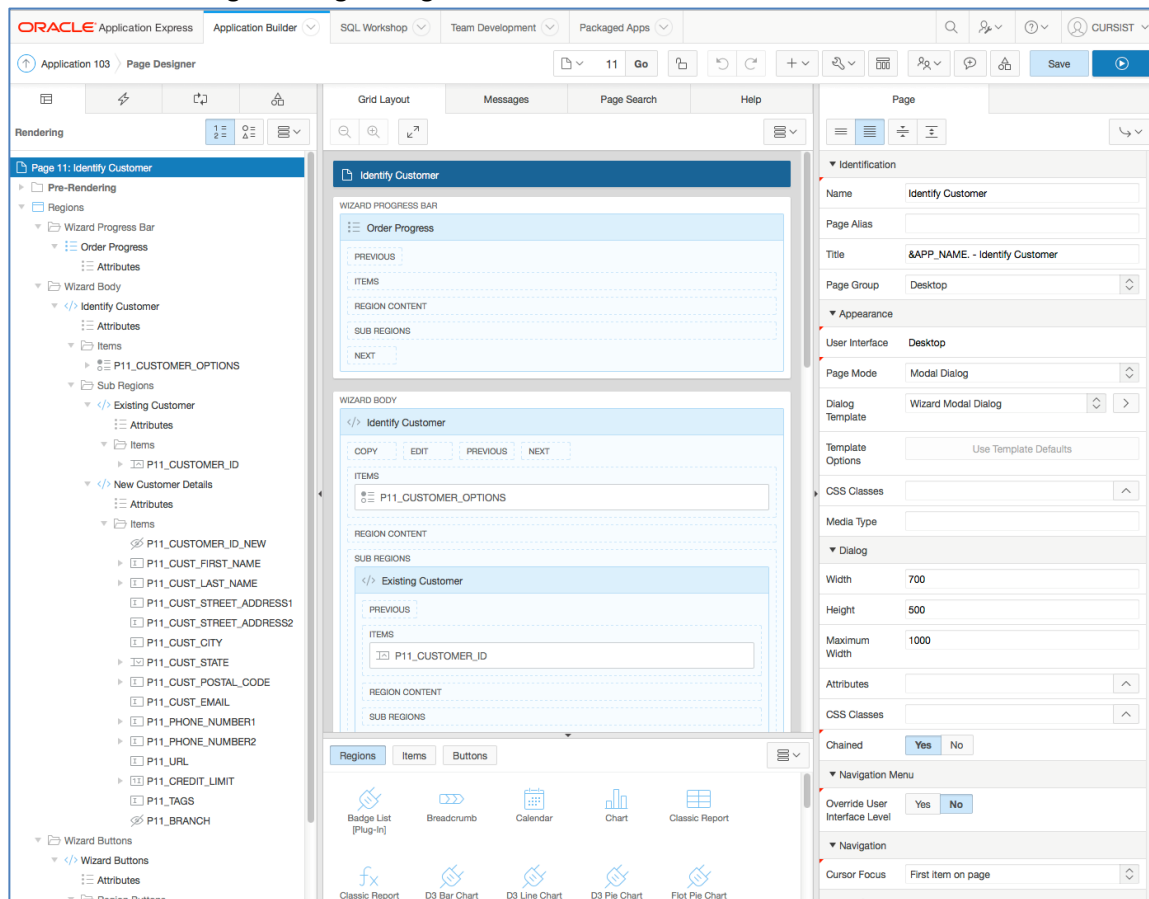
### XPlug

Mocht je werken met 5.0, dan is Xplug een goede add-on. Voor 5.1 is deze add-on eigenlijk minder interessant. XPlug is te vinden op: [http://blog.oratronik.org/?page\\_id=144](http://blog.oratronik.org/?page_id=144)

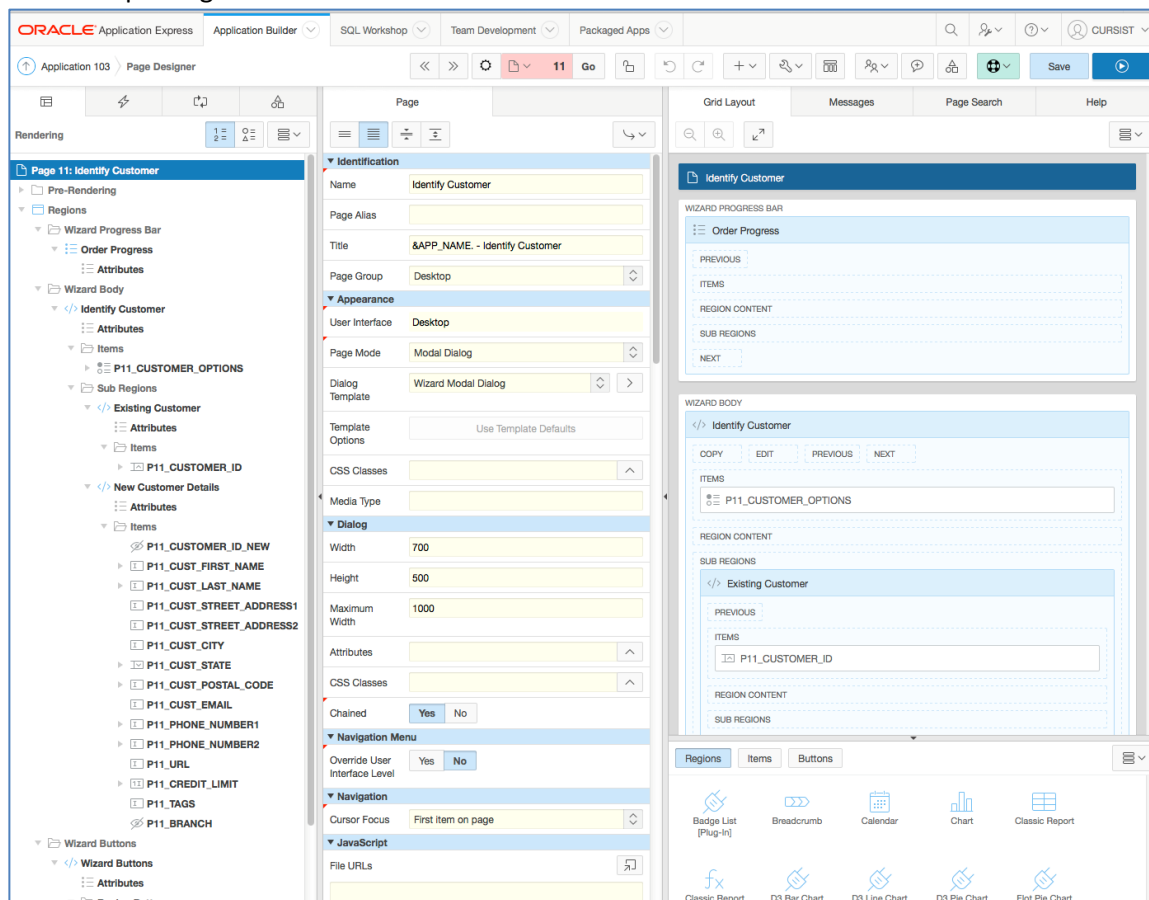
Met deze add-on kunnen we het grid in de Page Designer naar rechts verplaatsen, zodat we sneller met onze muis bij de properties zijn. Dat is in 5.1 goed geregeld, doordat je de tabs in de panes van de Page Designer zelf kan instellen. Er zijn ook wat kleuropties in te stellen en de (grote zwarte) tooltip is uit te zetten. Een andere handigheid zijn de twee buttons om heen en weer te navigeren tussen pagina's. Iets wat in de vroegere APEX versies zat, maar in 5.0 was verdwenen. In 5.1 is dit weer teruggekomen.

De instellingen kunnen worden gedaan met het menu dat onder de nieuwe groene knop te vinden is.

Hieronder is de originele Page Designer te zien:



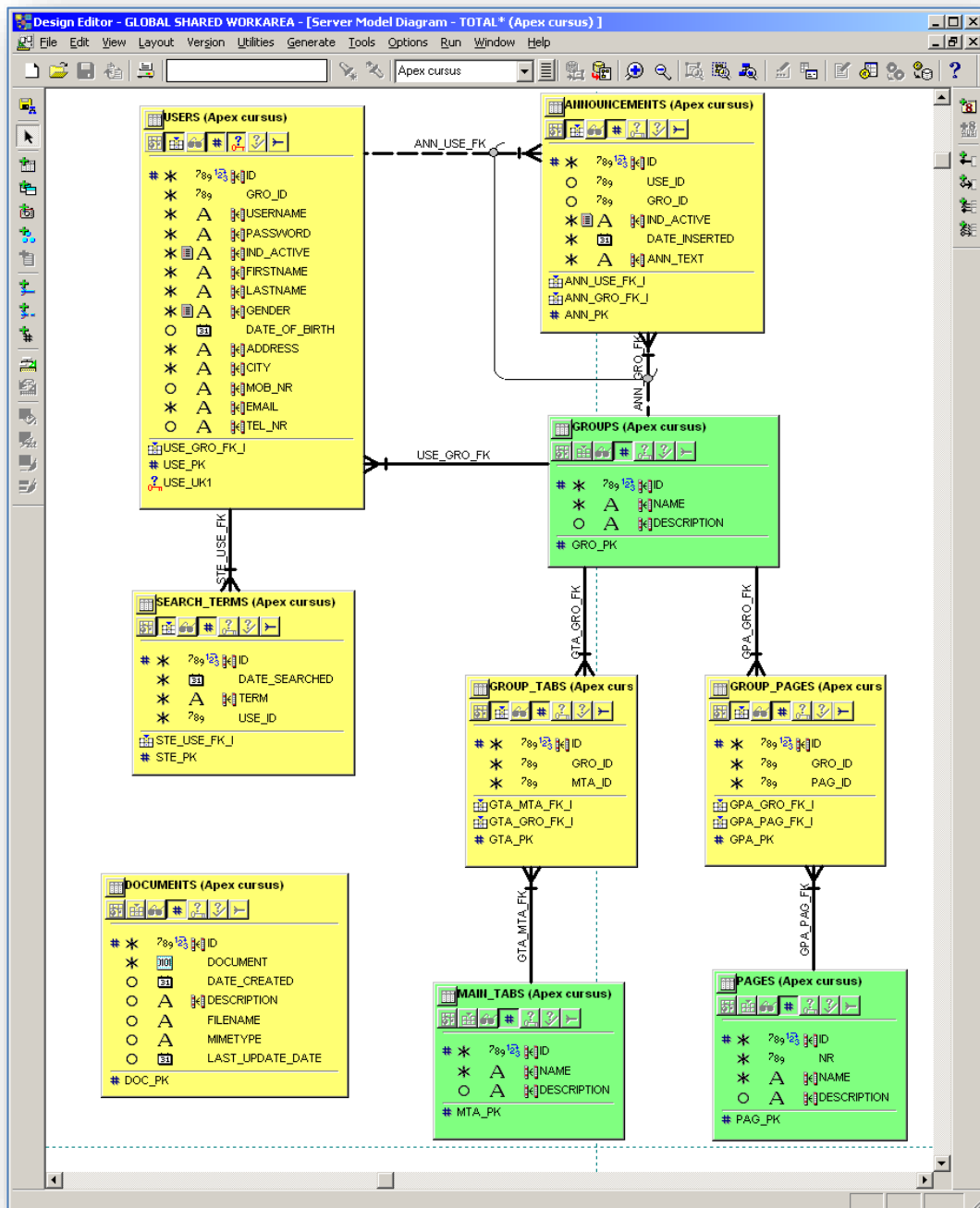
Na de aanpassingen:



### 3. Back Office basis

In deze oefening gaan we een begin maken met de back office applicatie. We maken eerst de tabellen en daarna een applicatie.

De tabelopzet voor de applicaties is eenvoudig gehouden. Hieronder is het datamodel weergegeven wat we zullen gebruiken gedurende de cursus. In werkelijke projecten zijn er vaak meer tabellen en attributen nodig. Zie bijlage Model voor de details.

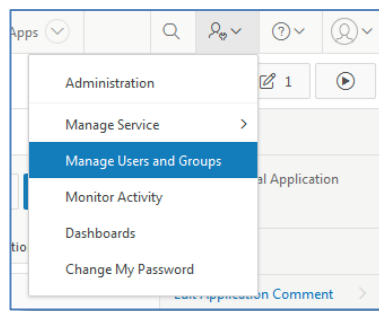


- Maak in de **SQL Workshop** een sequence. Noem deze **seq\_all**. Deze sequence zullen we de rest van de cursus gebruiken voor alle tabellen om de id's mee te vullen. Bij de wizards voor het maken van Forms wordt daarnaar gevraagd.

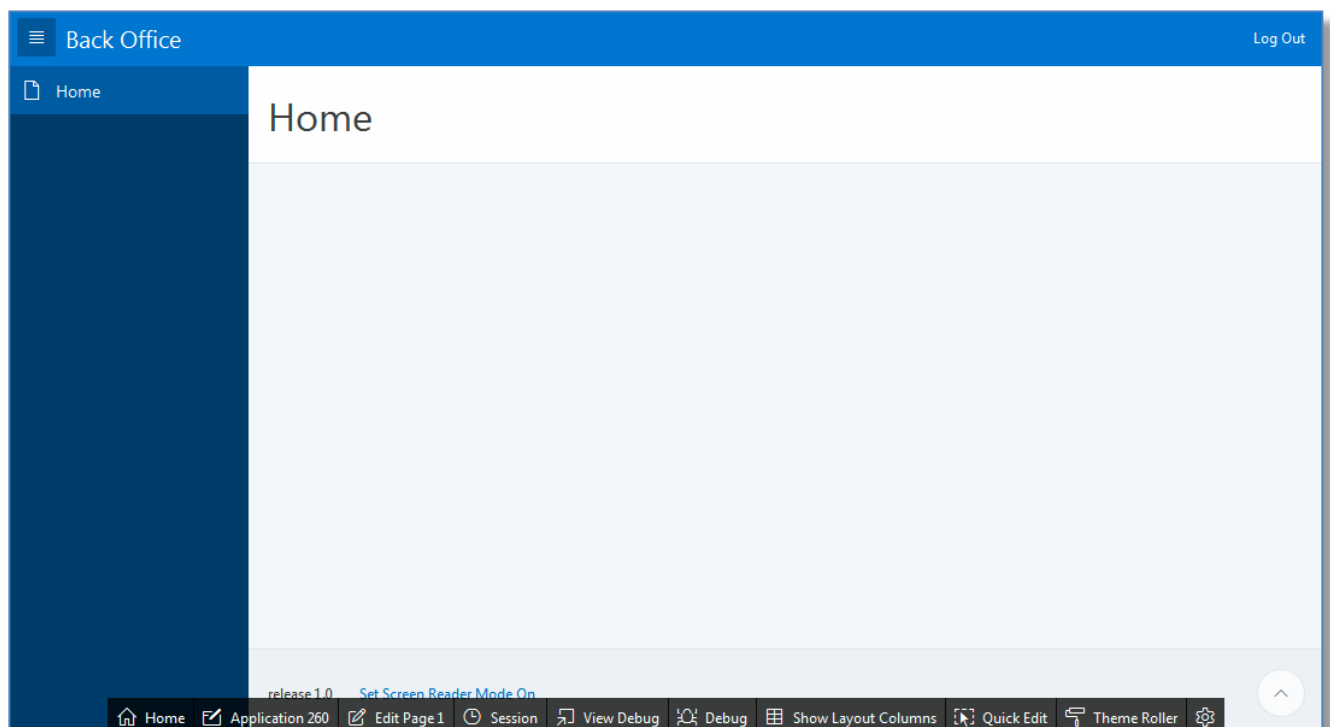
- Lees het script 'datamodel.sql' met behulp van de **SQL Scripts** functie in **SQL Workshop** in. Voer het script uit en controleer in de **Object Browser** van **SQL Workshop** of de objecten zijn aangemaakt.

We gaan nu een applicatie maken. Klik op de **Create** button in de **Application Builder** om de wizard te starten.

- Kies voor een **Desktop** applicatie.
- Noem de applicatie 'Back Office' en gebruik nummer **160**. Baseer de applicatie op het schema 'cursist'.
- Gebruik **Universal Theme** als thema.
- Gebruik als authenticatie schema het default **Application Express Accounts** schema.
- Datum formaat dd-mm-yyyy.
- Druk op **Create Application** aan het eind van de wizard.
- De applicatie is nu gemaakt en heeft een inlogpagina 101 en Home pagina 1.
- Klik op **Edit Application Properties**, onder de **tab User Interface** kan de titel worden aangepast of een logo worden toegekend aan de applicatie.
- Optioneel kunnen we een gebruiker aanmaken voor de Backoffice applicatie. Ga naar **Administration – Manage Users and Groups** en klik vervolgens op **Create User** om een eindgebruiker te maken:



- Start de applicatie en log in met deze user of 'cursist'. Het resultaat moet ongeveer als volgt uitzien:



## 4. Back Office schermen

### Inleiding

In deze oefening gaan we de eerste schermen maken voor de Back Office. Het doel van deze oefening is om diverse structuren de revue te laten passeren en te leren dat op zich elk scherm te maken is met (een combinatie van) **Reports, Forms, Interactive Reports** en **Interactive Grids**.

Het is een goede gewoonte om te beginnen met het maken van LOV's. Doorgaans komen de stamtabellen daar sowieso voor in aanmerking. Dit zullen dynamische LOV's worden. Daarnaast kunnen we voor kolommen gebaseerd op domeinen statische LOV's definiëren (of als Designer wordt gebruikt dynamische LOV's op de `cg$ref_codes` tabel). Ordenen op display value (ORDER BY 1) is ook een goede gewoonte. De users (uit de users tabel) zullen ook via een LOV worden gedisplaysed.

### Oefening

- Maak de volgende LOV's:

LOV Naam	Type	Tabel	Display Value	Return Value
LOV_PAG	Dynamisch	Pages	nr    '('    name    ')'	id
LOV_MTA	Dynamisch	Main_tabs	name	id
LOV_GRO	Dynamisch	Groups	name	id
LOV_USE	Dynamisch	Users	firstname    ' '    lastname	id
LOV_YESNO	Statisch	-	Yes, No	Y, N
LOV_GENDER	Statisch	-	Male, Female	M, F

We maken nu wat kleine aanpassingen op de eerste pagina van de applicatie:

- Ga naar de eerste pagina in de **Page Designer**.
- Aangezien we in deze cursus geen **breadcrumbs** zullen gebruiken, verwijderen we de **breadcrumb** region en de geassocieerde list entries.
- Voeg een **Static Content** region toe, zet er een welkomst tekst in en geef het een goede titel.

### Navigatie

Navigatie vindt plaats sinds versie 5.0 met het **Universal Theme** plaats door middel van **Lists** (voorheen gebeurde dat standaard nog met tabs). Voordelen t.o.v. tabs zijn de makkelijkere integratie met autorisatie en de mogelijkheid om een menu dynamisch te laten genereren d.m.v. een query.

Bij het **Universal Theme** kunnen twee **Lists** gebruikt worden voor de navigatie. Een **List** kan dienen als **Navigation Bar** (rechts boven) en een andere List kan dienen als **Navigation Menu**. Deze laatste kan verticaal (side) of horizontaal (top) georiënteerd zijn. De **Lists** zijn te definiëren bij de **Shared Components**.

- Klik op Edit **Application Properties** op applicatie niveau, onder de tab **User Interface** kan vervolgens gekozen worden voor de **Desktop** interface. Hier kunnen beide **Lists** worden gedefinieerd, alsmede de oriëntatie. Wijzig de **Position** om de oriëntatie aan te passen en kijk wat het effect is.

- Ga nu naar de Shared Components en klik op **Navigation Menu**.
- Open de **Desktop Navigation Menu** - List. Klik op de Home entry. Klik op **Image/Class** om de 'Font APEX' class 'fa-home' te selecteren en druk op Apply Changes. 'Font Awesome' is ook beschikbaar, indien wenselijk. Dit is in te stellen bij het thema en geldt dan voor de gehele applicatie.
- Maak een entry 'Basic Data'. Deze gaat dienen als hoofdmenu voor de resterende schermen in deze oefening. De pagina heeft **geen** target, dus erop klikken heeft geen navigatie effect.
- Open de **Desktop Navigation Bar List**. Voeg een entry vóór de Log Out link toe met Font APEX icoon fa-user, zodat is te zien wie er ingelogd is:



Het eerste scherm is een **Report with Form on Table** voor de pages tabel. Dit scherm is voor het onderhouden van pagina's in de Front Office applicatie. **Report with Form on Table** schermen lenen zich er vooral voor om simpele schermen te maken voor basistabellen met meerdere velden.

- Maak een nieuwe pagina 5. Kies voor page type **Form** en daarna voor **Report with Form on Table**.
- Kies als **Report Type** 'Classic'.
- We willen dat het form op dezelfde pagina verschijnt als het rapport, dus kies voor **Report Page Number** en **Form Page Number**: 5.
- Kies voor **Report Page Name** en voor **Form Page Name**: 'Pages'.
- Bij het **Navigation Menu** kiezen we voor een nieuwe entry genaamd 'Pages'. Deze komt onder het niveau van 'Basic Data', dit kan je aangeven bij **Parent Navigation Menu Entry**.
- Kies de pages tabel.
- Selecteer alle kolommen voor het rapport.
- Kies als **Primary Key Type** 'Select Primary Key Column(s)'. We hebben namelijk geen trigger code die de primary key vult bij inserts en willen onze sequence gebruiken. Indien er voor de ROWID-optie wordt gekozen, wordt deze optie niet weergegeven in de wizard. Voor alle nog te maken pages zullen we deze instelling gebruiken! De primary key voor alle tabellen is: **id**.
- We gaan seq\_all als sequence gebruiken. Kies bij **Sequence** 'Existing Sequence'. Dit geldt voor alle nog te maken schermen.
  - Opm. Als je Designer gebruikt met Table API's kan je **Existing Trigger** kiezen, of de ROWID-optie in de vorige stap.
- Selecteer alle kolommen voor het form.
- Beëindig de wizard door op **Create** te drukken.

- Er moeten nu nog wat zaken in de regions worden aangepast. Zet het **Sequence** nummer van het form, de **Static Content region** op 20. Beide regions hebben namelijk door de wizard **Sequence 10** gekregen. Omdat we de volgorde van de regions niet aan het toeval over willen laten, wijzigen we dit. Een alternatieve manier om dit te bewerkstelligen is **drag and drop**. Pak eenvoudigweg de region op en sleep hem naar de juiste plaats.
- Wijzig de **Region Title** van het report in 'pages' en de **Region Title** van het form in 'Page' (enkelvoud: je edit er maar 1 tegelijk).
- Run nu de page.
- Voeg de volgende data toe op deze pagina: 1 Welcome, 5 Documents en 10 Maintenance. Dit zijn de pages in onze (later te creëren) Front Office applicatie.
- Pas eventueel **Headings** en **Sorting** aan voor het rapport.

#### Let op

Waarschijnlijk zit er in de wizard een bug die tot gevolg heeft dat er **geen** menu entry in de **List** wordt gegenereerd, wanneer het form en report zich op dezelfde pagina bevinden.

- Voeg nu handmatig een entry toe in de Desktop Navigation Menu List voor pagina 5 en selecteer ook hiervoor een geschikte icoon. Zorg ervoor dat 'Basic Data' de **Parent List Entry** is.

#### Let op 2

Door een bugje wordt de het plaatje van Font Apex niet goed uitgelijnd voor item entries van het 2<sup>e</sup> niveau. We gaan daartoe een class toevoegen die op elke pagina wordt gebruikt.

- Maak een kopie van de **Page Template** 'Standard' en noem hem 'Standard Cursus'. Het origineel kan niet worden aangepast, omdat het **Theme** ge-subscribed is.
- Open deze kopie en navigeer naar de **Cascading Style Sheet - Inline**.
- Vul daar het volgende in:
 

```
.t-TreeNav .a-TreeView-node--leaf>.a-TreeView-content .fa {
    line-height: 32px;
}
```
- Navigeer naar **Shared Components** en klik op **Themes**. Selecteer het Universal Theme en navigeer naar **Component Defaults**. Stel voor **Page** het nieuwe **Template** 'Standard Cursus' in als de default.
- Nu zijn de plaatjes mooi uitgelijnd.

- Het resultaat ziet er ongeveer als volgt uit:

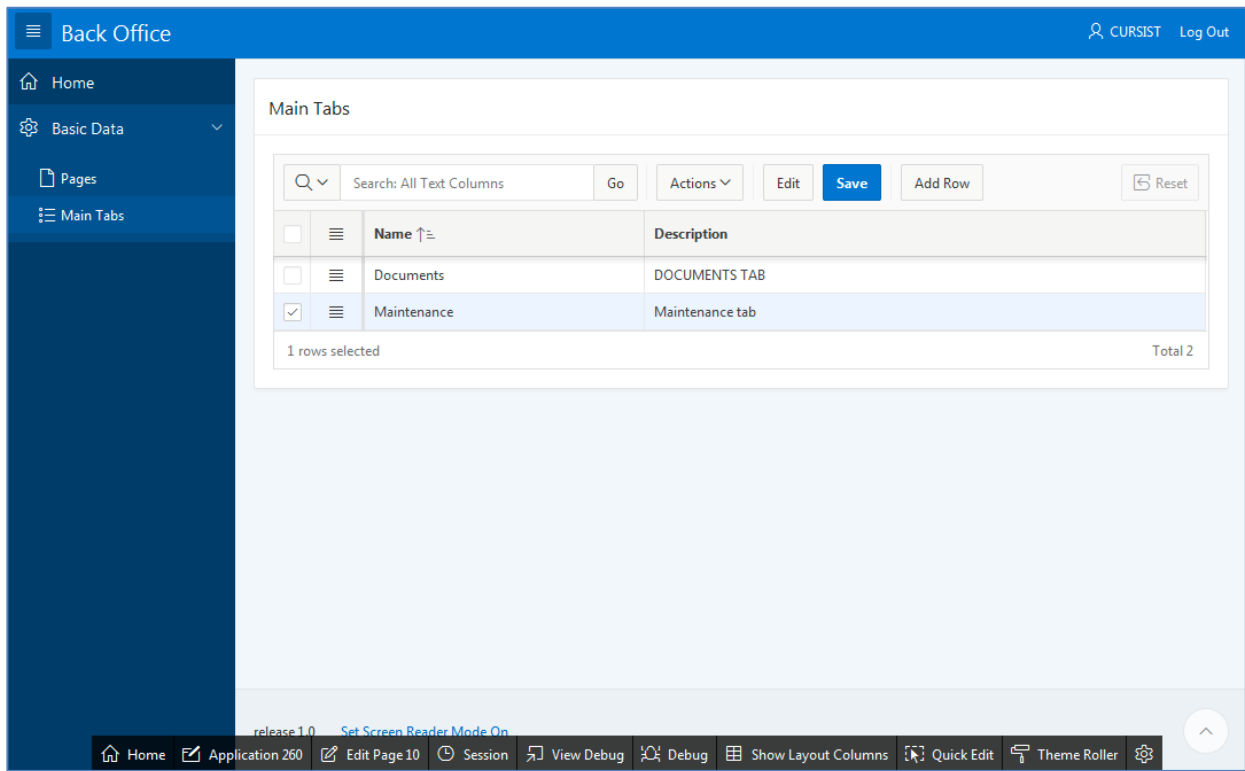
The screenshot shows the 'Back Office' interface. On the left is a navigation menu with 'Home', 'Basic Data', and 'Pages'. The main content area is divided into two sections. The top section, titled 'Pages', contains a table with the following data:

	Nr ↑	Name	Description
	1	Welcome	Start page
	5	Documents	Documents page
	10	Maintenance	Maintenance page
	40	test	test<script>alert('Hallo');</script>

Below the table is a '1 - 4' indicator and a 'Create' button. The bottom section, titled 'Page', is a form with fields for 'Nr', 'Name', and 'Description'. The 'Nr' and 'Name' fields have red asterisks indicating they are required. There are 'Cancel' and 'Create' buttons at the bottom of the form. At the very bottom of the screen, there is a status bar with various application controls like 'Home', 'Application 260', 'Edit Page 5', 'Session', 'View Debug', 'Debug', 'Show Layout Columns', 'Quick Edit', 'Theme Roller', and a scroll indicator.

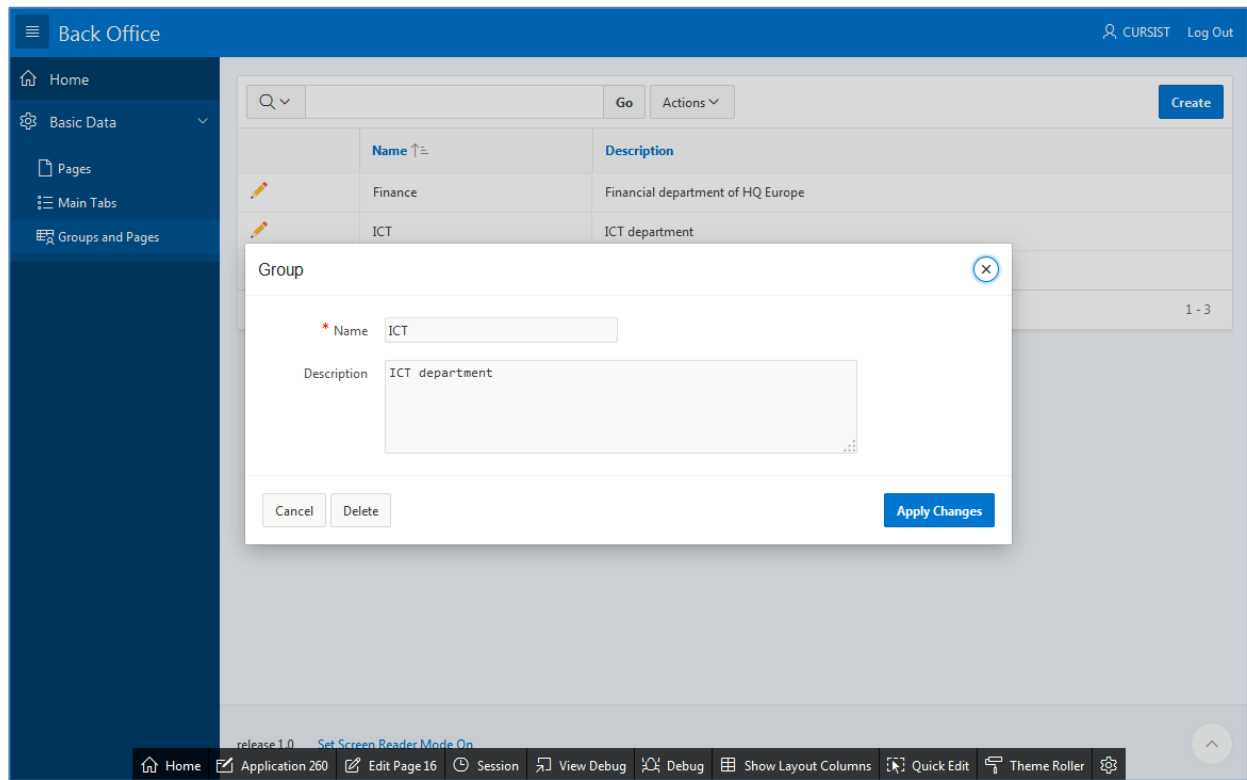
Het tweede scherm is een scherm om main\_tabs (1<sup>e</sup> niveau in het menu) te onderhouden van de Front Office applicatie. Deze dienen als autorisatie voor de navigatie.

- Maak een nieuwe page.
- Dit scherm maken we als een page met een **Interactive Grid** (de nieuwe uitgebreide versie voor een **Tabular Form / updatable report** ). Klik daartoe als **Page Type** Form en kies vervolgens op Editable Interactive Grid.
- Geef het scherm **Page Number** 10 en **Page Name** 'Main Tabs'.
- Dit scherm maken we onder hetzelfde hoofdmenu 'Basicdata' met als label 'Main Tabs'.
- Voeg een icoon van Font Apex toe aan het menu item.
- Om ervoor te zorgen dat het id de goede waarde krijgt bij een insert, moeten we de sequence seq\_all toevoegen aan de id kolom. Deze properties zijn onder het kopje **Default** te vinden van de id kolom.
- Run de page en experimenteer wat met de verschillende mogelijkheden van het Interactive Grid.
- Voeg de volgende data toe: 'Documents' en 'Maintenance'. Dit zijn de tabs voor in de Front Office applicatie.
- Bekijk het resultaat en pas eventueel nog de **Headings** en **Sorting** aan. Ook kan de breedte van de velden worden aangepast. Bijvoorbeeld het veld description mag iets breder zijn. Dit kan bij het **Actions** menu **Columns** worden aangepast. Na afloop van de wijzigingen moeten deze worden opgeslagen door in het **Actions** menu **Report – Save** te selecteren.
- Het resultaat zou er zo uit kunnen zien:



Nu gaan we pagina maken om de rechten van een groep te onderhouden voor wat betreft de toegankelijkheid van een groep tot bepaalde schermen en tabs (menu). Het onderhouden van de **groups** tabel doen we op hetzelfde scherm als **group\_pages** tabel. Het doel is een master detail scherm, dat we met de hand maken en verder uitbreiden.

- Als eerste stap maken we een nieuwe pagina voor de Groups tabel op basis van een **Report with Form on a Table**, zoals we eerder voor Pages hebben gemaakt. Dit is de Master. Voor het rapport kiezen we een **Interactive Report** op pagina 15 met als titel 'Groups and Pages'. Voor het form kiezen we een **Modal Dialog** op pagina 16 met als titel 'Group'. Pagina 15 moet onder het menu 'Basic Data' worden gekoppeld. Let op, gebruik alle velden in het rapport/form en als Primary key de id kolom, die wordt gevuld door sequence seq\_all.
- Na de wizard kunnen we e.e.a. een beetje aanpassen. Voeg een icoon toe voor het menu. Maak de id kolom van het rapport hidden. Voeg een standaard ordening toe aan de name kolom.
- Voeg een aantal groups toe als data, b.v.: 'ICT' en 'Finance'.
- Het resultaat ziet er ongeveer als volgt uit:



We gaan nu de pagina verder uitbreiden.

- Maak een hidden item P15\_ID.
- Voeg een link toe aan de kolom 'name' op het (master) rapport. Deze moet item P15\_ID vullen met het id, waarop geklikt wordt en op dezelfde pagina blijven. Wijzig daartoe de property **Type** van de kolom in 'Link' en vul de details in door op de **Target** button te klikken onder het property kopje **Link**. Als **Link Text** moet #NAME# gebruikt worden.
- Voeg een nieuwe region van het **Type** Interactive Grid als detail toe op basis van de Group Pages tabel. In de query moet een where clause worden gebruikt o.b.v. P15\_ID dat het master-id bevat.
- De region moet een **Server-side Condition** krijgen, zodat de region alleen getoond wordt als P15\_ID een waarde heeft.
- Maak de kolommen id en gro\_id hidden en koppel aan pag\_id de juiste LOV. Pas de **Heading** van pag\_id aan in 'Page'. Zet de **Column** en **Header Alignment** van pag\_id op **start**.
- Zorg ervoor dat bij het inserten van een record de sequence wordt gebruikt en dat als default voor de foreign key het master-id item wordt gebruikt.
- Na het deleten van een master record is het goed om de region voor de details te verbergen, anders kan er wellicht een detail toegevoegd worden onder een net verwijderde master. Dit is enigszins ingewikkeld, omdat het hier een Modal Dialog betreft. Later in de cursus gaan we er iets dieper op in. Ga naar pagina 16 en navigeer naar het **Close Dialog** process. Vul bij de property **Items to Return** P16\_ID in, zodat deze waarde doorgegeven wordt aan de aanroepende pagina. Op pagina 15 navigeren we naar de **Dynamic Actions** op de groups region (niet onder de **Create** button). We voegen aan de bestaande **Dynamic Action** een **True Action** toe: **Set Value** met als **Set Type: Dialog Return Item** met als **Return Item: P16\_ID**. Het **Affected Item** is P15\_ID. De waarde van P16\_id wordt dus bewaard in P15\_ID. Vervolgens voegen we een nieuwe **Dynamic Action** toe onder de vorige. Deze is conditioneel: Als het **Client-side Item** P15\_ID null is, dan voeren we als **True Action** het volgende uit: **hide** van de Pages region.
- Tenslotte passen we de layout nog wat aan. Zet de **Column Span** property op 7 voor de region. Kies voor **beide** regions het **Template: Standard** en bij de **Template Options** kiezen we voor de **Style: Remove Borders**. Over templates en layout volgt later meer.

Tot op dit moment is het Interactive Grid eigenlijk alleen maar voor het displayen van de details. Om de details updateable te maken passen we de een aantal zaken aan.

- Ga naar de id kolom. Zet de property **Primary Key** op **Yes**. Zet de **Default Type** voor deze kolom op **Sequence** met seq\_all als **Sequence**.
- Ga naar de attributes van het Interactive Grid en zet het **Edit - Enabled** op **Yes**. Nu wordt ook automatisch het DML process toegevoegd voor de DML afhandeling van het Interactive Grid.
- Pas de default voor de hidden foreign key kolom pag\_id aan naar P15\_ID.
- Run de page en koppel wat pages aan de verschillende groups. Dit dien later als autorisatie voor de Front Office applicatie.
- Het resultaat ziet er ongeveer als volgt uit:

The screenshot displays the APEX Back Office interface. The top navigation bar includes 'Back Office', 'CURSIST', and 'Log Out'. The left sidebar contains navigation options: 'Home', 'Basic Data', 'Pages', 'Main Tabs', and 'Groups and Pages'. The main content area is divided into two sections: 'Groups' and 'Pages'.

**Groups Section:** A table with columns 'Name' and 'Description'. It lists three groups: 'Finance' (Financial department of HQ Europe), 'ICT' (ICT department), and 'fddds' (fddd). A 'Create' button is visible in the top right.

Name	Description
Finance	Financial department of HQ Europe
ICT	ICT department
fddds	fddd

**Pages Section:** A table with a 'Page' column. It lists four pages: '1 (Welcome)', '10 (Maintenance)', '5 (Documents)', and '40 (test)'. The '1 (Welcome)' page is selected. A 'Save' button is visible in the top left of the table area.

Page
1 (Welcome)
10 (Maintenance)
5 (Documents)
40 (test)

The bottom status bar shows 'release 1.0', 'Set Screen Reader Mode On', and various utility icons like 'Home', 'Application 260', 'Edit Page 15', 'Session', 'View Debug', 'Debug', 'Show Layout Columns', 'Quick Edit', 'Theme Roller', and a refresh icon.

### Tip!

We hebben nu handmatig een Interactive Grid toegevoegd. We kunnen dit via een trucje met een wizard doen. Start de wizard op voor het maken van een nieuwe pagina en selecteer als doelpagina een bestaande pagina. Na afloop wordt het component toegevoegd aan de bestaande pagina. Uiteraard moet je daarna nog verdere details aanpassen.

### Opmerking

Op basis van oude wizard kunnen we een pagina als onderstaand plaatje maken. Wellicht dat deze in nieuwere versies van APEX wordt verwijderd. Bij **Create Page** kies je dan als **Page Type** voor **Legacy Page** en vervolgens op **Legacy Master Detail**.

Back Office
Log Out

- Home
- Basic Data
- Pages
- Main Tabs
- Groups and Pages

### Groups

Id	Name	Description
	ICT	ICT department
	Finance	Finance department

1 - 2

[Create](#)

### Group

Name \*

Description

[Cancel](#)
[Delete](#)
[Save](#)

### Groups Pages

	Page
<input type="checkbox"/>	1 (Welcome)
<input type="checkbox"/>	5 (Documents)

1 - 2

[Delete Checked](#)
[Add Row](#)

Home
 Application 160
 Edit Page 15
 Session
 View Debug
 Debug
 Show Grid
 Quick Edit
 Theme Roller

We gaan nu een shuttle gebruiken om de intersectie tabel **group\_tabs** te onderhouden.

- We willen een nieuwe region naast de onderste plaatsen met daarin de shuttle. Maak een nieuwe region met als titel 'Group Tabs'. Zorg ervoor dat voor deze region de **Start New Row** property op **No** staat.
- Deze region moet dezelfde **Conditional Display** properties hebben als de detail region: indien er geen master geselecteerd is mag de region niet zichtbaar zijn.
- Pas ook de Dynamic Action aan, zodat deze region ook wordt verborgen bij het deleten van een master (group) record.
- Plaats in de nieuwe region een **Shuttle Item** met als naam P15\_SHUTTLE. Gebruik de LOV\_MTA lov. Let op dat dit geen database item is.
- Nu moet er nog code toegevoegd worden om de shuttle beginwaarden te geven bij het selecteren van een nieuw master record. Ook voor het wegschrijven van de inhoud van de shuttle moet code worden geschreven. Een shuttle werkt net als een **Multiselect Item** met een door dubbele punten gescheiden lijstje met id's. Het is uiteraard ook mogelijk om hier een generieke oplossing voor te schrijven. Zie hiervoor de bijlage.

Om deze lijst te vullen gaan we wat code schrijven. Maak een **After Header - PL/SQL Process** dat na het fetchen van het id van het master record (het group id: p15\_id) de shuttle vult. Gebruik de volgende code:

```
DECLARE
-- get the selectlist value as a list e.g. 1:2:4
l_selected APEX_APPLICATION_GLOBAL.VC_ARR2;

BEGIN

IF :p15_id IS NOT NULL THEN

    SELECT mta_id
    BULK COLLECT INTO l_selected
    FROM group_tabs
    WHERE gro_id = :p15_id ;

    -- Assign the colon separated list to l_selected
    :p15_shuttle := APEX_UTIL.TABLE_TO_STRING(l_selected);

END IF;

END;
```

- Wegschrijven naar de database. Maak een **After Submit - PL/SQL Process** (After Submit) dat als laatste afgaat. Gebruik onderstaande code:

```
DECLARE

-- insert the selectlist value (as a list e.g. 1:2:4) into the intersection table
l_selected APEX_APPLICATION_GLOBAL.VC_ARR2;

BEGIN

IF :p15_id IS NOT NULL THEN

    -- convert the colon separated string of values into
    -- a PL/SQL array
    l_selected := APEX_UTIL.STRING_TO_TABLE(:p15_shuttle);

    -- clean up the intersection table first
    DELETE FROM group_tabs
    WHERE gro_id = :p15_id;

    -- Loop over array to insert lookup_ids and master_id
    -- in to intersection table
    FOR i IN 1..l_selected.count LOOP
        INSERT INTO group_tabs
        ( id
        , gro_id
        , mta_id)
        VALUES
        (seq_all.nextval
        , :p15_id
```

```

        , l_selected(i) );
    END LOOP;
END IF;
END;

```

- Voeg een **Button** toe die de pagina submit en dus de inhoud van de shuttle bewaard. Noem de button Save.
- Zet de **Appearance Hot** op **Yes** en zet de **Button Position** op 'Copy', dit is rechtsboven in de region template.
- Het resultaat zou er als volgt uit kunnen zien:

The screenshot shows the 'Back Office' interface with a sidebar on the left containing navigation options: Home, Basic Data, Pages, Main Tabs, and Groups and Pages. The main content area is divided into three sections:

- Groups:** A table with columns 'Name' and 'Description'. It contains two entries: 'Finance' (Financial department of HQ Europe) and 'ICT' (ICT department). A 'Create' button is visible in the top right.
- Pages:** A table with a 'Page' column. It lists four pages: '1 (Welcome)', '10 (Maintenance)', '5 (Documents)', and '40 (test)'. The first row is selected. A 'Save' button is visible in the top right.
- Group Tabs:** A section for managing tabs, showing a 'Documents Maintenance' tab. A 'Save' button is visible in the top right.

At the bottom of the interface, there is a status bar with various utility icons and text: 'release 1.0', 'Set Screen Reader Mode On', and a list of icons for Home, Application 260, Edit Page 15, Session, View Debug, Debug, Show Layout Columns, Quick Edit, Theme Roller, and a refresh icon.

- Nadat het scherm gemaakt is moet er nog data worden ingevuld voor de verschillende gemaakte groepen. Zorg ervoor dat elke groep pagina's en bijbehorende tabs/menu entries heeft. Bijvoorbeeld ICT mag bij het '(1)Welkomstscherm(1)' en '(5)Documenten' en de bijbehorende tab 'Documents', terwijl Finance overall bij mag.

Het scherm om de users tabel te onderhouden maken we ook als **Report with Form on Table**.

- Het report is een **Classic Report** en het form in de **Normal Page Mode**. Noem het scherm 'Users'. Zorg ervoor dat beide regions op dezelfde pagina terecht komen: 20.
- Na de wizard gaan we verder met de layout. Wijzig de volgorde van de regions en pas de titels aan.
- De volgende LOV's moeten gebruikt worden: LOV\_GRO, LOV\_GENDER en LOV\_YESNO voor de diverse Form items en Report columns. Voor Yes/No kolommen kan in een form het **Type Switch** gebruikt worden.
- Wijzig de region **Headers, Labels, Sorting** etc. Indien er veel items zijn in een report kunnen de niet belangrijke ook weggelaten worden. Ze komen uiteindelijk natuurlijk wel terug in het form.
- Wijzig de layout en breedte van de items op de **Static Content** forms region zodanig, dat de ruimte beter wordt benut. Experimenteer ook met het **Grid Properties**, door **Show Grid** aan te zetten. Het is nu handig om in de **Page Designer** de **Layout** tab actief te hebben om de diverse aanpassingen te zien.

Het grid bestaat uit 12 kolommen. Met **Start New Row** kan je aangeven of het item op een nieuwe rij moet beginnen. De waarde in **Column** geeft aan waar in het grid het item begint.

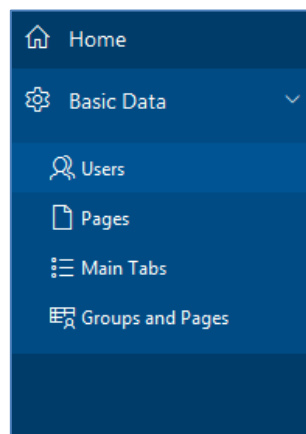
**Label Column Span** is default 2 in APEX 5.1 en geeft aan hoeveel ruimte het label inneemt. De waarde van de breedte van het gehele item is **Column Span**. Let op: deze is inclusief de waarde van **Label Column Span**! Het grid kan ook genest worden met **Subregions**. Om het grid live te tonen kan in de **Developer Toolbar** in de runtime omgeving de **Show Layout Columns** optie worden geselecteerd:

The screenshot shows a form titled 'User' with a grid layout. The grid has several columns of varying widths. The fields are arranged as follows:

- Active? (Yes/No buttons)
- Group (dropdown menu)
- Firstname (text input) and Lastname (text input)
- City (text input)
- Address (text input)
- Email (text input)
- Tel Nr (text input) and Mob Nr (text input)
- Gender (radio buttons for Male and Female)
- Username (text input) and Password (text input)
- Date Of Birth (calendar icon)

At the bottom, there are buttons for 'Cancel', 'Delete', and 'Apply Changes'.

- Het kan zijn dat er door de bug geen menu optie is aangemaakt. Voeg deze toe.
- Wijzig de Menu volgorde zodanig dat 'Users' het eerste scherm wordt. Ook als er op de hoofdmenu optie 'basicdata' wordt geklikt moet het Users scherm worden geopend én moet de Users entry ge-highlight (current) worden:



- Maak een aantal users aan met behulp van dit scherm.
- Het resultaat zou er als volgt uit kunnen zien:

Back Office CURSIST Log Out

Home  
Basic Data  
Users  
Pages  
Main Tabs  
Groups and Pages

### Users

Edit	Firstname ↑	Lastname	Group	City	Address	Username	Email	Active?
	Arie	Boeren	Finance	ALKMAAR	Koningsstraat 13	arie	arie@company.be	Yes
	Sjon	de boer	ICT	AMSTERDAM	Keizersgracht 43	sjon	sjon@werken.nl	Yes
	qw	qw	ICT	qw	qw	qw	qw	No

1 - 3

[Create](#)

---

### User

\* Active?  Yes  No

\* Group

\* Firstname  \* Lastname

\* City

\* Address

\* Email

Tel Nr  Mob Nr

\* Gender  Male  Female

\* Username  \* Password

Date Of Birth

[Cancel](#) [Delete](#) [Apply Changes](#)

Home Application 260 Edit Page 20 Session View Debug Debug Show Layout Columns Quick Edit Theme Roller

### Extra oefeningen

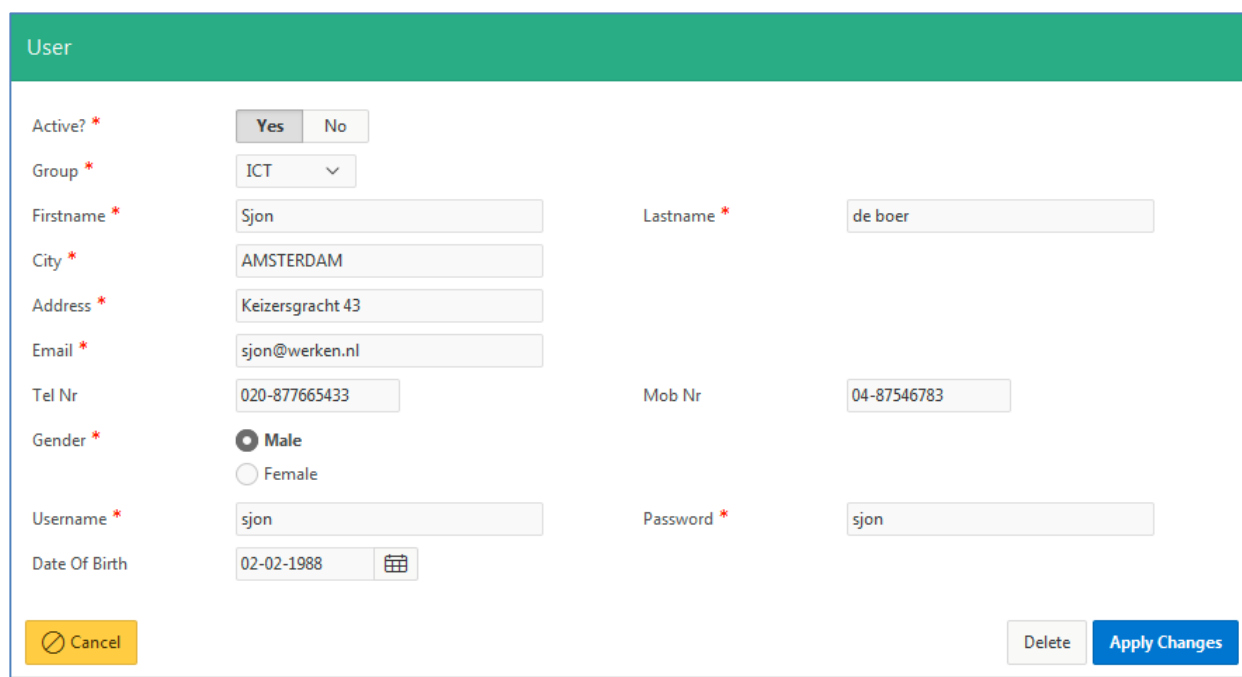
- Maak een 2<sup>e</sup> master detail scherm 17 met Groups en Group Pages op basis van twee **Interactive Grids**. Dit kan je doen d.m.v. de wizard voor **Page Type Form** en dan te kiezen voor **Single Page Master Detail**. Bekijk hoe de **Interactive Grids** aan elkaar gelinkt zijn.
- Een **Select List** met de property **Allow Multi Selection** op 'Yes', werkt op dezelfde wijze als een shuttle. Probeer dit uit.

## 5. Universal Theme/Templates

In deze oefening zullen we een aantal zaken bekijken met betrekking tot layout en de mogelijkheden die het Universal Theme biedt om daarin aanpassingen te doen.

### Template Options

- Navigeer naar Page 20 (users) in de **Page Designer**. Klik op de user **Region**, waarin het **Form** voor het invoeren en onderhouden van een user zich bevindt. In de Property Pane is onder het kopje **Appearance** de property **Template Options** te zien. Klik hier op de grijze grote button. Er verschijnt nu een modal window met de verschillende **Template Options** voor deze **Region**. Wijzig nu de volgende instellingen:
  - Item Padding -> Slim Padding
  - Accent -> Accent 1
  - Label Alignment -> Left
- Navigeer nu op Page 20 naar de **Cancel** button en wijzig onder het kopje **Appearance** de **Button Template** van **Text** naar **Text with icon**. Vervolgens wijzigen we bij de **Template Options** het **Type** in **Warning** en de **Icon Position** in **Left**. Tenslotte wijzigen we de property **Icon CSS Classes** in een geschikt **Font Awesome** plaatje, b.v. 'fa-undo-arrow' of 'fa-ban'. Druk op **Save** om alle wijzigingen op te slaan en bekijk het resultaat. Het zou er als volgt uit kunnen zien:



- Experimenteer ook nog wat met andere **Template Options**.

### Nieuwe Template Option Toevoegen

In de CSS files van APEX zitten veel meer classes dan dat er **Template Options** zijn. De volgende CSS class is bijvoorbeeld aanwezig:

```
.i-h720 > .t-Region-bodyWrap > .t-Region-body { height: 720px; }
```

We gaan deze class nu als **Template Option** toevoegen aan een de reeds bestaande **Template Option**: **Body Height**, waar nu 240px, 320px, 480px en 640px als keuze in aanwezig zijn.

- Navigeer naar de **Templates** in de **Shared Components**. Maak een kopie van de **Region Template** genaamd **Standard** en geef deze de naam 'Standard Cursus'. Het maken van de kopie is nodig, omdat het gehele **Theme ge-subscribed** is en de individuele **Templates** daarin niet aan te passen zijn. Open het (local) **Template** en bekijk de **Template Options**. Bij **Body Height** select list zijn de huidige opties te zien. Klik in de lijst daaronder op de '640px' **Template Option** en bekijk de

properties. Druk op **Cancel** klik nu op de **Add Template Option** button. Vul de volgende properties analoog aan de '640px' **Template Option** in:

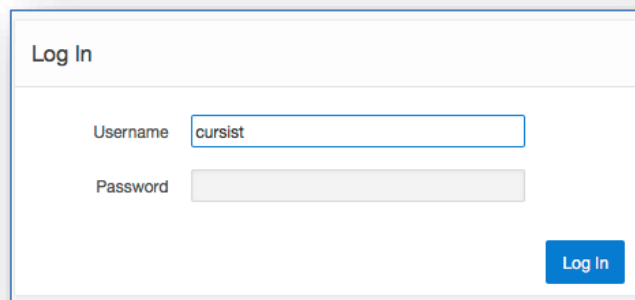
- Display Sequence -> 100
- Group -> Body Height
- Display Name -> 720px
- Option Identifier -> 720PX (default gevuld)
- CSS classes -> i-h720

Druk nu op **Create** en de **Template Option** is toegevoegd. Druk op **Apply Changes** voor het **Template**.

- Ga naar de **Page Designer**, kies een pagina en navigeer naar een **Region**. Pas de **Template** aan met het nieuwe **Template** 'Standard Cursus'. Als we nu naar de **Templates Option** kijken, zien we de nieuwe optie er voor **Body Height** tussen staan. Kies de nieuwe instelling (720px) en bekijk wat het resultaat is.
- We gaan er nu voor zorgen dat dit template het default template wordt voor alle nieuw te maken **Forms**. Navigeer naar **Shared Components** en klik op **Themes**. Selecteer het Universal Theme en navigeer naar **Component Defaults**. Stel voor **Region** het nieuwe **Template** 'Standard Cursus' in als de default.

### Voor versie 5.0: Inlog pagina aanpassen

De default inlogpagina gaan we iets aanpassen:



We gaan nu deze inlog pagina aanpassen, zodat ie wat meer lijkt op de mooiere inlogpagina van APEX zelf. (zie ook een blogpost van Dimitri Gielis)

- Navigeer naar de inlogpagina 101. De **Template** van de **Region** 'Log In' staat op 'Standard'. Wijzig deze in 'Login'.
- Vul bij **Icon CSS Classes** een Font Awesome class in om een plaatje toe te voegen. Eventueel kan er ook in de **Title** Property gebruik worden gemaakt van een <img> tag om een eigen plaatje toe te voegen. Een andere optie is een inline page css, bijvoorbeeld:

```
span.t-Login-logo {
  background-image: url(plaatje);
  background-size: cover;
  width: 296px;
  height: 96px;
}
```
- Zet met behulp van de **Template Options** de **Width** van de button op 'Stretch' en de **Size** op 'Large'.
- Ga nu naar de properties van P101\_USERNAME. Zet het **Template** op 'Hidden', hierdoor verdwijnt het label. Zet de **CSS Classes** op 'icon-login-username', hierdoor verschijnt er een icon in het item. En vul bij Value Placeholder 'username' in, dat verschijnt ook in het item en verdwijnt als je begint te typen.
- Iets soortgelijks doen we voor P101\_PASSWORD. Zet het **Template** op 'Hidden', de **CSS Classes** op 'icon-login-password' en als Value Placeholder 'password'.
- Tenslotte zetten we de **Value Required** property op 'Yes' van beide items en zetten we in de **Post-Text** property de waarde:

```
<span class="t-Login-iconValidation a-Icon icon-check"></span>
```

- Druk op **Save** om alle wijzigingen op te slaan en bekijk het resultaat. Het zou er als volgt uit kunnen zien:

#### Voor versie 5.1: Inlog pagina aanpassen

- Navigeer naar de inlogpagina 101.
- Zet de **Value Required** property op 'Yes' van de items P101\_USERNAME en P101\_PASSWORD en zet in de **Post-Text** property van beide items de waarde:  
`<span class="t-Login-iconValidation a-Icon icon-check"></span>`
- Navigeer naar de login **Button**. Zet bij **Behaviour** de **Execute Validations** property op **No**.
- Druk op **Save** om alle wijzigingen op te slaan en bekijk het resultaat.

#### Page Template aanpassen

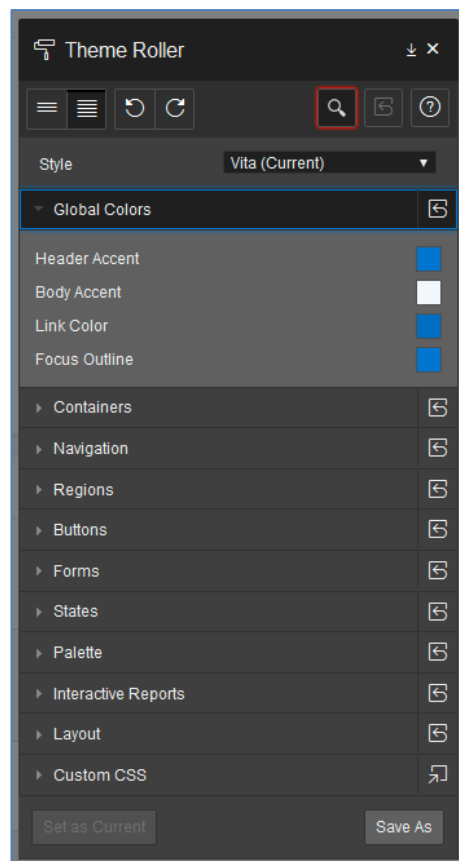
We gaan nu de tekst 'release 1.0 Set Screen Reader Mode On' aanpassen, welke onder elke pagina verschijnt. Let op: een aantal van onderstaande stappen is i.v.m. de bug in een vorige oefening wellicht al eerder uitgevoerd.

- Maak een kopie van de **Page Template** 'Standard' en noem hem 'Standard Cursus'. Het origineel kan immers niet worden aangepast, omdat het **Theme** ge-subscribed is.
- Open de kopie en navigeer naar de **Body**. Verwijder de tekst '#SCREEN\_READER\_TOGGLE#' en sla het **Template** op.
- We gaan er nu voor zorgen dat dit template het default template wordt voor alle nieuw te maken **Pages**. Navigeer naar **Shared Components** en klik op **Themes**. Selecteer het Universal Theme en navigeer naar **Component Defaults**. Stel voor **Page** het nieuwe **Template** 'Standard Cursus' in als de default.
- Navigeer nu naar **Application** niveau en klik op de **Edit Application Properties** button en wijzig de tekst van de Version property in b.v. 'Version 5.1'.
- Bekijk het resultaat op de applicatie pagina's.

## Theme Roller

Een handig hulpmiddel om de userinterface aan te passen is de **Theme Roller**. We gaan eerst kijken naar de verschillende **Theme Styles** die standaard aanwezig zijn bij een nieuwe **Universal Theme** applicatie voor de **Desktop User Interface**.

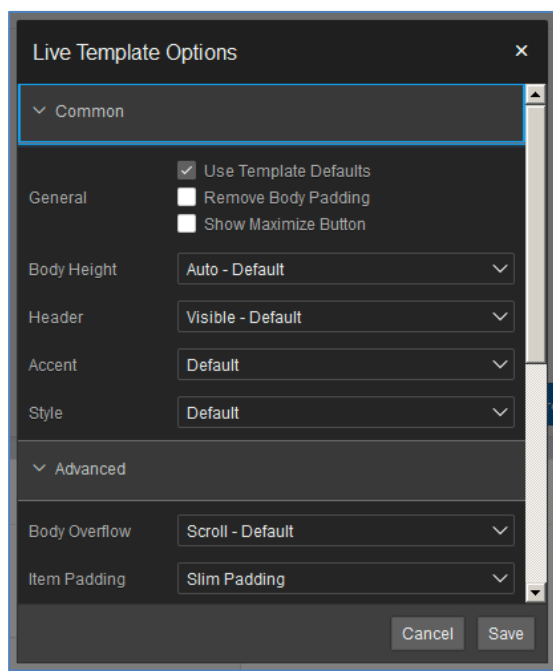
- Navigeer naar het **Universal Theme** en bekijk onderaan de pagina welke **Theme Styles** er zijn.
- Klik op **Vita-Slate** en wijzig de property **Is Current** in 'Yes', en bekijk het resultaat.
- Nu activeren we de **Theme Roller** door rechtsonder in de **Developer Toolbar** op **Theme Roller** te klikken. Nu verschijnt de **Theme Roller**. Bij **Style** kan voor het juiste **Theme Style** gekozen worden om aan te passen. Klik op de **Show All** button (2<sup>e</sup> van links), zodat alle properties te zien zijn. Als je individuele properties van Componenten wilt wijzigen, kan je deze aanpassen door op de naam te klikken van het Component Type.
- Klik op **Container** en wijzig de **Container Border Radius**. Hiermee kan je de Regions en Items een rondere hoek geven.
- Klik nu op Layout en wijzig de **Navigation Tree** breedte. Handig om te weten dat deze instelling hier 'verstopt' zit.
- Nadat het eruitziet, zoals je wenst, kunnen met Save As de instellingen worden opgeslagen onder een eigen naam als een JSON-string. In **Theme Styles** bij het **Theme** kan nu worden gewisseld van **Theme**.



## Live Template Options

Sinds versie 5.1 is het mogelijk om de Template Options live aan te passen in plaats van ze via de Page Designer steeds te moeten opslaan en in de runtime omgeving te moeten bekijken als een soort trial and error methode.

- Navigeer naar een scherm en klik op **Quick Edit** in de **Developer Toolbar**. Als je op een component of een region klikt navigeer je direct in de Page Designer naar dat component. Als je op het sleuteltje rechtboven klikt verschijnen de Live Template Options voor dat component. Je kan de **Template Options** nu direct aanpassen en meteen het effect zien. Experimenteer hiermee.



## Reports

De opbouw van de out of de box **Interactive Reports** en **Classic Reports** is nogal grof. We kunnen de hoogte van de records beïnvloeden met CSS. Daarnaast willen we dat de teksten in de cellen niet 'wrappen' naar de volgende regel. Dit geeft vaak een wat rommelige layout. Maak de volgende aanpassingen:

- Maak een kopie van het **Report Template** 'Standard' en noem deze 'Standard Cursus'. Open het nieuwe template. Navigeer naar **Column Template 1** en voeg de volgende inline style toe in de <td> tag:  
style="padding-top:4px !important; padding-bottom:4px !important; white-space:nowrap !important;"
- Bij een andere opgave gaan we wat verder in op dit template.
- Voorzie elke report region in de applicatie dat in aanmerking komt van dit template.
- Zorg er nu voor dat dit template het default template wordt voor alle nieuw te maken **Reports**. Dit is in te stellen bij **Shared Components - Themes**.
- Voor een **Interactive Report** moeten we dit bij de **Page Template** regelen, omdat er geen templates zijn voor **Interactive Reports**. Navigeer naar de door ons aangepaste **Page Template**: Standard Cursus.
- Navigeer naar de **Cascading Style Sheet – Inline** en voeg onderstaande CSS toe:

```
table.a-IRR-table td {
    padding-top:5px !important;
    padding-bottom:5px !important;
    white-space: nowrap !important;
}
```

## Forms

De opbouw van de standard region voor Forms is ook nogal 'ruimtelijk'. Dit kunnen we aanpassen.

- Navigeer naar het **Region Template** 'Standard Cursus' wat we reeds gemaakt hebben.
- Bij **Template Options – Preset Template Options** is te zien dat **Item Padding** op **Default** staat. Wijzig dit in **Slim Padding**. Nu worden alle Items iets dichter bij elkaar gezet.
- Pas het **Thema** aan om dit de default te laten zijn bij nieuwe forms. Dit is in te stellen bij **Region Defaults - Forms**.

## Extra Oefening

We gaan de verschillende mogelijkheden van **Font APEX** bekijken.

- Bekijk de demo in applicatie: <https://apex.oracle.com/pls/apex/f?p=42> (of installeer de **Universal Theme** packaged application in je **Workspace**). Navigeer naar icons.
- Klik op een icon en zie wat de mogelijkheden zijn. Bekijk ook de **Animations** en **Modifiers**.
- Als je een icoon hebt samengesteld, kan je de desbetreffende Icoon **CSS Class** kopiëren en bijvoorbeeld gebruiken in het menu of button. Ook kan je de **HTML Markup** gebruiken in een rapport bij bijvoorbeeld de **HTML Expression** property van een kolom.
- In de voorgaande versie werd **Font Awesome** gebruikt. Dit kan nog steeds worden gebruikt door dit op **Theme** niveau in te stellen voor de gehele applicatie. Bekijk waar deze property zit.

## 6. Current record

In deze oefening maken we een current record indicator door middel van het highlighten van het geselecteerde record. Indien een gebruiker ziet met welk record hij/zij bezig is, vermindert dat de kans op fouten tijdens het invoeren van gegevens.

Indien bijvoorbeeld op een **Report with Form on Table** scherm een record wordt geselecteerd om te bewerken, wordt er een hidden item met het id van dat record in de tabel gevuld, zodat het fetch proces dat tijdens het renderen uitgevoerd wordt, het desbetreffende record ophaalt uit de database.


### Classic Report

Bovenstaande gebruiken we om highlighten in een **Classic Report** te bewerkstelligen.

- Open het **Report Template** 'Standard Cursus' en kijk hoe het is opgebouwd.

Er zijn 4 **Column Templates**. Wanneer een rapport gerenderd wordt, worden de templates 1 voor 1 toegepast, afhankelijk van de **Column Template n Condition** en **Expression**. Indien de eerste conditie voldoet, wordt het eerste template toegepast. Indien deze niet voldoet, wordt naar de tweede conditie gekeken, etc.

- Bekijk de helptekst van de **Column Template n Condition** en **Column Template n Expression** probeer een conditie en template te formuleren, zodat het highlighten werkt.

Edit	Firstname	Lastname	Group	City	Address	Email	Active?
	Arie	Boeren	Finance	Haarlem	Karelsestraat 20	arie.boeren@apex.nl	Yes
	Sjon	Jansen	ICT	Oost Zaandam	Weggetje 88	sjon.jansen@apex.nl	Yes

Bekijk ook de property **Background color for current row** van **Row Highlighting** een stukje daaronder. Hiermee kan worden ingesteld welke kleur de rij krijgt als je je muis eroverheen beweegt. Dit is dus wat anders.

### Decode

Een alternatief is het gebruik van de **decode** functie in de query, maar dit is een minder mooie oplossing, zowel visueel en conceptueel (scheiden van logica en presentatie).

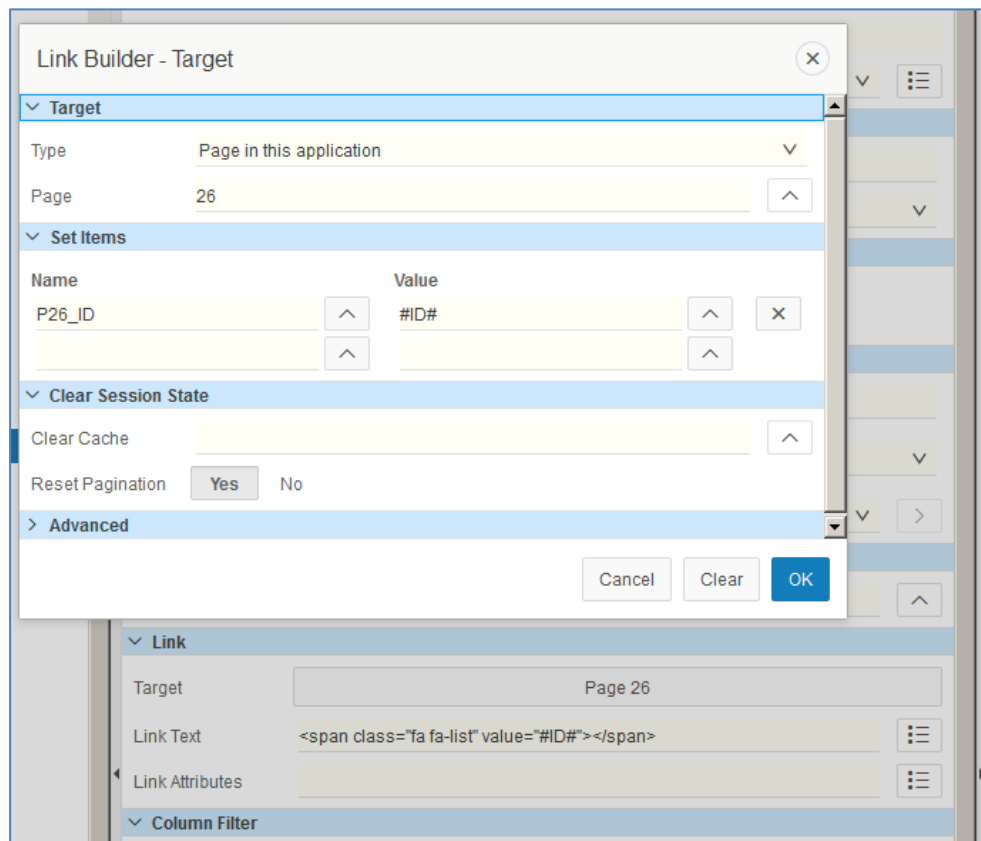
- Probeer dit door de 'pages' query op pagina 5 aan te passen.

## jQuery voor Classic en Interactive Report

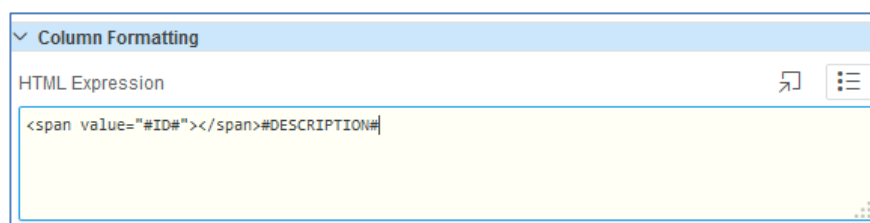
Er is ook een elegante oplossing met jQuery mogelijk, die voor beide typen rapporten werken.

- Ga naar de aangepaste **Page Template** 'Standard Cursus'. Navigeer naar **Execute when Page Loads**. Voeg daar onderstaande JavaScript toe :

```
var p_id = $v('P' + $v('pFlowStepId') + '_ID');
if (p_id) {
    $('span[value=' + p_id +']').parents('tr').children('td').css('background-color', '#FFFF99');
}
```
- Bij een report moet nu nog op kolom niveau het volgende worden opgenomen:  
<span class="fa fa-list" value="#ID#"></span>  
De class="fa fa-list" hoeft niet, maar hier kan meteen het icoontje worden opgenomen voor de link.  
Zie onderstaande voorbeeld voor de properties bij een Report **Link** kolom



Of bij **Column Formatting** voor een **Classic** of **Interactive Report**:



## 7. Error handling

In deze oefening gaan we bekijken hoe error handling geïmplementeerd kan worden.

Een **Error Handling Function** moet de volgende interface hebben:

```
function <name of function> (  
    p_error in apex_error.t_error )  
    return apex_error.t_error_result
```

De twee types staan gedefinieerd in de user guide: ([https://docs.oracle.com/database/apex-5.1/AEAPI/APEX\\_ERROR.htm#AEAPI2209](https://docs.oracle.com/database/apex-5.1/AEAPI/APEX_ERROR.htm#AEAPI2209)). Als je in de database kijkt bij de **apex\_error** package in het APEX\_050xxx schema, zie je een iets andere definitie.

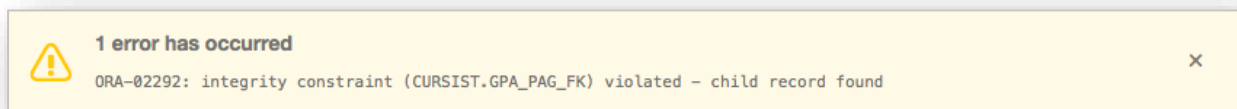
De #NOTIFICATION\_MESSAGE# is een substitution string in de **Page Template**. Dit is de plaats waar de error wordt getoond. Daarnaast kan de error 'Inline with Field' verschijnen op het scherm. Ook is binnen deze Template een **Error Page Template** te definiëren.

Op applicatie niveau is het mogelijk om aan te geven wat de default positie is van een eventuele melding: **Inline with Field** en/of in **Notification**. Op page niveau kan je een aanvullende tekst meegeven bij de **In-line Error Notification Text** property die eveneens wordt getoond op deze plaats.

We gaan nu wat experimenteren met de verschillende mogelijkheden van de door APEX geboden **Error Handling**.

- Installeer het script error\_handling.txt. Dit is de code die Oracle standaard als voorbeeld geeft. Het zal niet direct compileren, omdat er wordt verwezen naar een tabel, waarin foutmeldingen zijn opgeslagen. Het is uiteraard een goede gewoonte om het op deze manier te doen, maar in onze opgave gebruiken we dat niet en kan dat gedeelte worden uit gecommentarieerd.

De melding van Oracle die we gaan afvangen zie je hieronder. Run de applicatie en ga naar pagina 5, waar de pagina's worden onderhouden. Als je daar de Welcome-'pagina' probeert te deleten, krijg je een foreign key error:

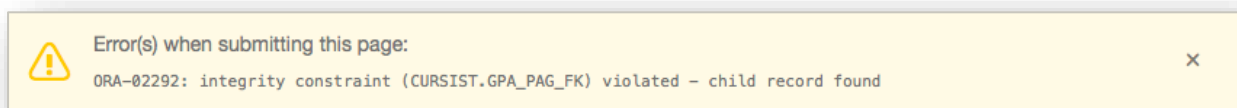


- Deze tekst is de 'unhandled error' tekst, die gedefinieerd kan worden bij het process dat deze error veroorzaakt. Experimenteer met de **Error** properties bij het **Process Row of PAGES** process (derde tab van het linker panel). Maak na afloop de property **Error Message** weer leeg.

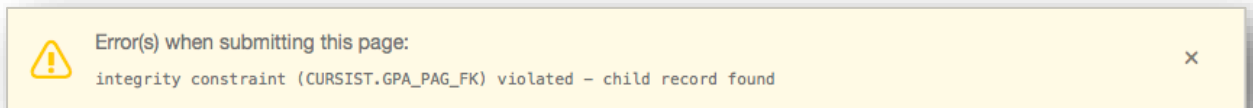
### Let op

Door een bug in APEX 5.1 werken sommige zaken alleen als op **Page** niveau bij **Advanced – Reload on Submit** de waarde **Always** is gekozen.

- Vul nu een tekst in op het **Rendering** onderdeel van de **page** bij de **In-line Error Notification Text** property, bijvoorbeeld: 'Error(s) when submitting this page:'. Deze melding wordt getoond als eerste regel van de melding, op de plaats van default text '1 error has occurred'. De melding ziet er nu als volgt uit:



- Nu voegen we de nieuwe functie toe bij de property **Error Handling Function**. De melding ziet er nu als volgt uit:



- Bekijk de documentatie van de API package APEX\_ERROR op internet.
- Pas de functie aan, zodat een nette melding wordt getoond bij deze foreign key error. Gebruik daarvoor een **if then end if** constructie op de plaats waar de aanroep naar de foutmeldingen tabel stond. Gebruik als **constraint\_name** 'GPA\_PAG\_FK'. Het resultaat zou er als volgt uit kunnen zien:



- Probeer de melding tevens **inline** bij het **Name** field te krijgen, zoals hieronder te zien is. Merk op dat er een '(Go to error)'-link achter de oorspronkelijke melding wordt toegevoegd.

Back Office CURSIST Log Out

Home Basic Data Users Pages Main Tabs Groups and Pages

Error(s) when submitting this page:  
This page is referenced and cannot be removed. (Go to error)

Pages

Edit	Nr ↑	Name	Description
	1	Welcome	Welcom page
	5	Documents	Documents page
	10	Maintenance	Maintenance page

1 - 3

Create

Page

Nr \*

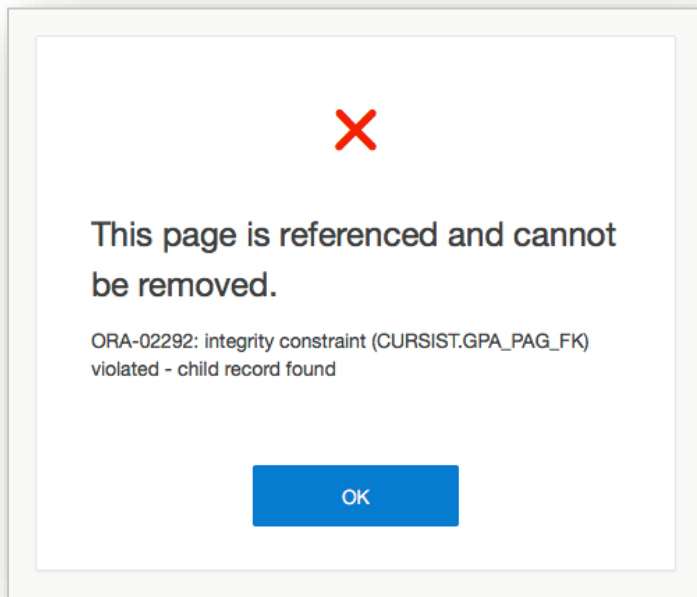
Name \*  ?  
This page is referenced and cannot be removed.

Description

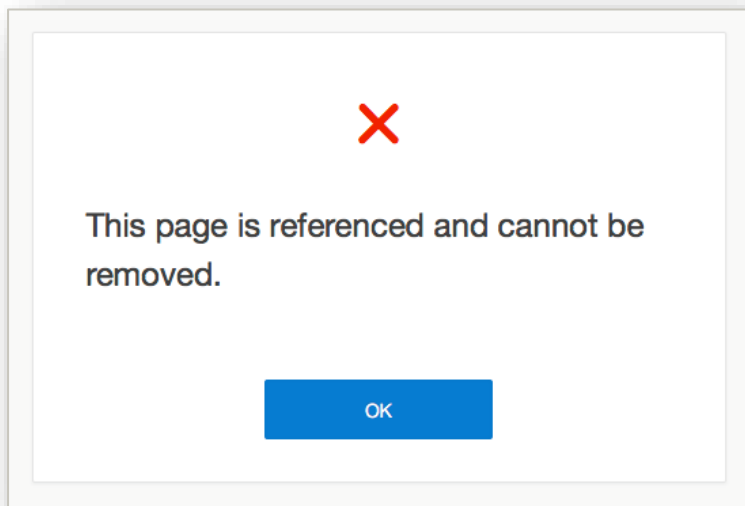
Cancel Delete Apply Changes

Home Application 160 Edit Page 5 Session View Debug Debug Show Grid Quick Edit Theme Roller

- Probeer nu als **display\_location** de package constante voor 'ON\_ERROR\_PAGE' te gebruiken. Wat opvalt is dat de tekst ingevuld bij de **In-line Error Notification Text** property niet meer wordt getoond, maar dat de oorspronkelijke fout is toegevoegd.



- Probeer de 'ORA-02292' hier weg te krijgen. Dit is op twee manieren te doen. Door de **Error Page Template** aan te passen of door de functie aan te passen. Doe het op de laatste manier, omdat voor andere fouten de extra informatie wellicht wel van belang kan zijn.





#### Tips!

- APEX kan soms zelf bepalen waar (dus bij welk item de inline melding moet staan) de error optreedt met de procedure **apex\_error.auto\_set\_associated\_item**. Zie ook de documentatie.
- Voor **processes** tijdens de **Processing** kunnen error meldingen weergegeven worden als er *iets* fout gaat, maar die worden alleen getoond als er geen error handling functie gedefinieerd is. Met behulp van de error handling functie kunnen de meldingen (en eventueel logging) preciezer worden bepaald en weergegeven.
- De melding voor een Validation worden via de **Error Message** property van de Validation afgehandeld.

## 8. Presentatie en Logica

Opmaak van de HTML kan in een rapport op kolom niveau geregeld worden met de **HTML Expression** property. Voordeel is het scheiden van logica en presentatie.

- Probeer onderstaande layout te bewerkstelligen op pagina 20 (users), zonder de query aan te passen. First Name en Last Name samenvoegen en vet afdrukken

Edit	Name ↑≡	Group	City	Address	Username	Email	Active?
	<b>Arie Boeren</b>	Finance	ALKMAAR	Koningsstraat 13	arie	arie@company.be	Yes
	<b>Sjon de boer</b>	ICT	AMSTERDAM	Keizersgracht 43	sjon	sjon@werken.nl	Yes

### Tip!

Natuurlijk kan er ook veel met CSS en templates bereikt worden.

## 9. Modal dialogs

We zullen in deze oefening meerdere soorten van modal dialogs gaan bekijken.

### Inline Dialog

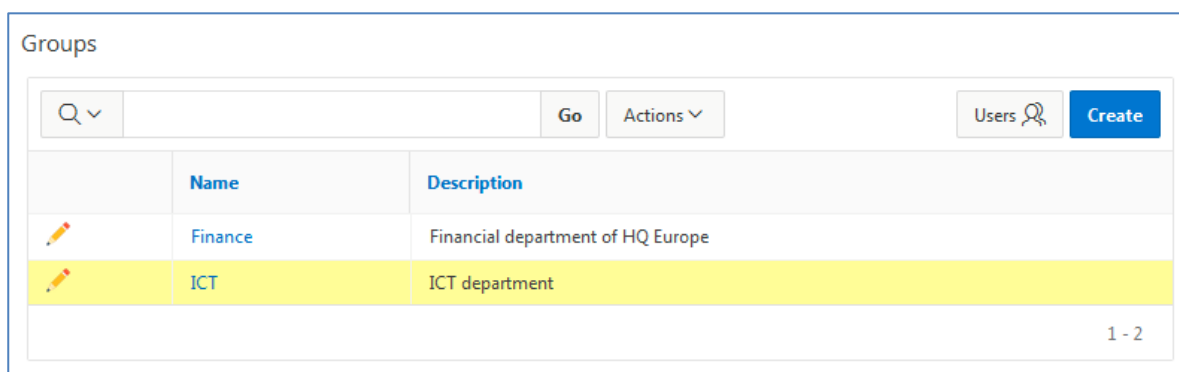
De eerste is de Region met het **Inline Dialog Template**. Dit is een **Region** op een pagina die in eerste instantie verborgen is en door het aanroepen met een JavaScript functie zichtbaar wordt. Omdat een **Inline Dialog** een **Region** op de **Page** zelf is, heeft een button op deze **Region** dus effect op de gehele pagina, net als elke andere Region, zoals bijvoorbeeld het submitten. Ook is er een Template aan gekoppeld met bijbehorende Template Options, zoals de grootte van de Dialog.



Voor eenvoudige zaken, zoals het displayen van informatie, navigatie, weinig validatie of page processing is een **inline Dialog** de meest effectieve keuze.

Voor een **Inline Dialog Region** geldt dat de **Static ID** property een waarde moet hebben om ermee te kunnen werken. De **Inline Modal Region** openen gaat met de volgende JavaScript functie: `openModal('static_id_name');` Voor `static_id_name` moet de **Static ID** van de **Region** ingevuld worden. Sluiten van de **Inline Modal Region** gaat met de JavaScript functie: `closeModal('static_id_name');` Ook kan de gebruiker met het kruisje rechtsboven op de Modal Region of met de **Escape** toets de **Modal Region** sluiten. Als er een andere actie gewenst is kan er op de normale manier een button worden toegevoegd aan de **Region**.

We gaan nu op pagina 15 een button toevoegen, waarmee we een Modal window willen openen dat de users van de gekozen group laat zien:

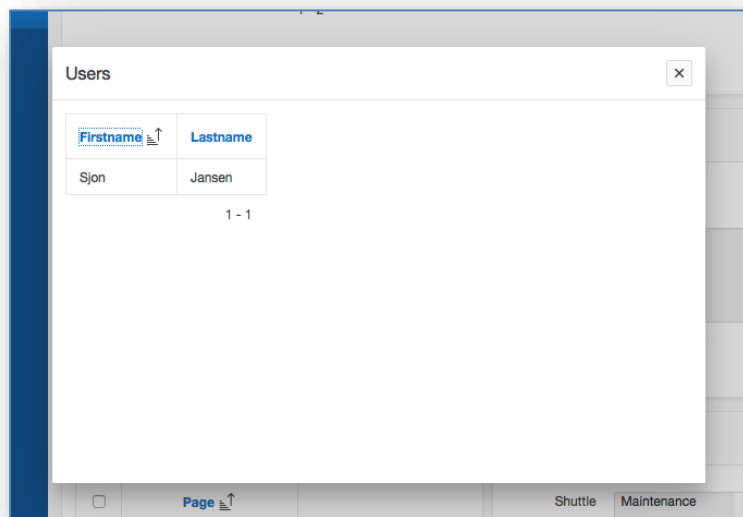
- Maak op pagina 15 bij de Groups **Region** (de 1<sup>e</sup> Region) een button met **Font APEX** plaatje fa-users.
- De button mag alleen te zien zijn als er een group is gekozen.



	Name	Description
	Finance	Financial department of HQ Europe
	ICT	ICT department

- Maak nu een **Region** aan met als **Template: Inline Dialog**. Op deze Region kan het rapportje worden gemaakt.
- Roep met behulp van een **Dynamic Action** op de button een Javascript functie aan, zoals hierboven beschreven is (Let erop dat **Fire On Page Load** property uit staat voor de JavaScript **Dynamic Action**).

Het resultaat ziet er ongeveer als volgt uit:



### Modal Dialog Page

Wanneer meer data interactie vereist is, is een eigen **Page** voor de **Dialog** (hierna **Modal Dialog Page** genoemd) het handigste, omdat daar ook alle **Page Processing** gebruikt kan worden. Er kan dus complexere interactie met de gebruiker plaatsvinden. Nadeel is dat het iets langzamer is, in vergelijking met een **Inline Dialog Region**.

De APEX dialogs zijn gebaseerd op de **jQuery Dialog Widget**. Op een **Inline Region Template** kunnen met behulp van Template Options diverse layout zaken ingesteld. Bij een Modal Page zijn er meerdere instellingen mogelijk met behulp van de **Page** properties. Zie de link <http://api.jqueryui.com/dialog/> en de voorbeelden op de **Help** pagina van de **Attributes** property voor een compleet overzicht van opties.

Om een **Modal Dialog Page** te maken moet de **Page Mode** property op **Modal Dialog** (of **Non-modal Dialog** voor een popup window) worden gezet. Dit kan tijdens de **Create Page Wizard** of achteraf. Als er naar zo'n pagina wordt g navigeerd, wordt deze automatisch **Modal geopend**.

Het *sluiten* kan door het maken van een **Close Dialog Page Process**. Je kan de page ook sluiten door een **Close Dialog Dynamic Action**. Beide manieren ondersteunen ook het teruggeven van **Page Items**, waar je vanaf de aanroepende **Page** naar kunt verwijzen. Gebruik een **Page Process**, wanneer de **Page** gesubmit dient te worden.

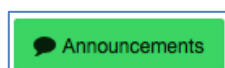
*Cancelen* kan door middel van een **Dynamic Action** gebaseerd op bijvoorbeeld een button, waarvan de **True Action** is gezet op **Cancel Dialog**.

Door middel van de **Dialog Closed Dynamic Action** op de aanroepende **Page** kan bijvoorbeeld een refresh worden uitgevoerd deze **Page**. Ook kunnen met de **True Actions** '**Set Value**', **Page Items worden teruggeven** die door de **Modal Dialog Page** zijn doorgegeven.

Zie voor meer informatie de **User's Guide**: <https://docs.oracle.com/database/apex-5.1/HTMDB/managing-pages-in-a-database-application.htm#GUID-BBFB0724-E33B-4846-AC94-A20E491B5D4E>

We gaan nu user announcements beheren met behulp van een **Modal Dialog Page**.

- Maak op pagina 20 bij de Users **Region** (de 1<sup>e</sup> Region) een button, **Type**: Succes, **Icon Position**: Left, **Icon CSS Class Font APEX**: fa-comment. De button mag uiteraard alleen te zien zijn als er een user is gekozen:



- De button moet navigeren naar pagina 25, die we hieronder gaan maken.
- Maak met de Wizard **Report with Form on Table**. Let daarbij op de volgende zaken:
- Baseer de pagina's op de announcements table.
- Het **Report** komt op pagina 25. Kies een voor een **Interactive Report**. Dit is een normale page.
- Het **Form** komt op pagina 26 en kies bij de Page Mode **Modal Dialog**.
- Maak geen menu entries aan voor deze pagina's. We navigeren immers m.b.v. de zojuist gecreëerde button naar deze pagina's, daarnaast is pagina 25 in de context van de gekozen user op pagina 20.
- Zorg ervoor dat wanneer pagina 25 en 26 in beeld zijn, de menu entry 'Users' actief is.
- De kolom gro\_id mag op beide schermen hidden worden gemaakt, die gebruiken we hier niet.
- Gebruik LOV's waar mogelijk.
- Om het **Interactive Report** wat mooier te maken kan voor een standard Template worden gekozen, waarbij voor **Style - Remove Borders** gekozen is. Pas de **Title** van de region ook aan.
- Let erop dat bij het **Report** alleen gegevens te zien zijn van de gekozen user (**where clause** in de query aanpassen m.b.t. use\_id).
- Zet voor de kolom ann\_text de **Security – Escape special characters** property op **No**. Op deze manier wordt de HTML geïnterpreteerd en ziet het er mooier uit. Dit is wel een potentieel security probleem.
- Let erop dat bij inserten van een nieuwe announcement, deze bij de gekozen user worden toegevoegd (**default** waarde van use\_id).
- Default waarde voor date\_inserted is sysdate. Zet Y als default waarde voor ind\_active.
- Maak op pagina 25 ook een button om terug te navigeren naar pagina 20.
- Voor de Text kolom in het **Form** gebruiken we een **Rich Text Editor**.
- Maak pagina 26 op, zodat het er ongeveer als volgt uitziet:

- Bekijk wat er gebeurt als de gebruiker na het aanpassen/toevoegen van een announcement weer terugkomt op pagina 25. Bekijk daarvoor de **Dynamic Actions** op de **Create** knop en de **Region** zelf. Dit zijn de aanroepende objecten.

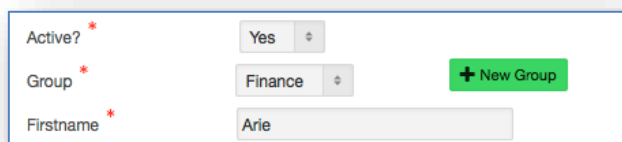
### Tips!

- Om te kijken of alles goed werkt is het handig om niet meteen de **Text Items** op **Hidden** te zetten. Als alles werkt en de id's worden goed gevuld kunnen de **Text Items** weer op **Hidden** gezet worden.
- Om een kolom header in een rapport te laten wrappen kan eenvoudig '<br/>' gebruikt worden.
- Dialogs kunnen ook genest worden: Zie ook de **Chained** property van een **Modal Dialog Page**.

## Modal Dialog Page – uitbreiden LOV

We gaan nu een **Modal Dialog Page** gebruiken om naast een LOV de gebruiker de mogelijkheid te geven om extra waarden in de tabel achter een **Select List** te inserten. We gaan dit doen op pagina 20 naast de Group LOV.

- Maak pagina 28 op basis van de **Form on a Table** wizard voor de groups table.
- Kies voor een Modal en maak geen menu entree aan voor deze pagina.
- Save en Delete buttons zijn niet nodig.
- Button op pagina 20 naar pagina 28, layout van de button: **Type:** Succes, **Size:** Small, **Icon Position:** Left, **Icon CSS Class Font APEX:** fa-plus:



- Kijk of de navigatie werkt.
- Er ontbreekt nu nog een belangrijk onderdeel. Namelijk dat de LOV van de **Select List** automatisch refresht na het toevoegen van een nieuwe group. Voeg aan het aanroepende **Dialog** object (de button) een **Dialog Closed Dynamic Action**, die dit bewerkstelligt.
- Tenslotte willen we graag dat de LOV automatisch de laatste waarde selecteert die is aangemaakt in de **Modal Dialog Page**. We willen de laatst gemaakte P28\_id gebruiken als waarde voor P20\_gro\_id. P28\_id is echter leeg, omdat er een **Reset Page** proces staat voordat het **Dialog** wordt afgesloten. We deleten daarom dit **Reset** proces. Om te voorkomen dat de oude waarden nog in de **Modal Dialog Page** staan bij het opnieuw openen, zetten we **Clear Cache** op 28 van de aanroepende button. Nu P28\_id de juiste waarde bevat, kunnen we een nieuwe **True Action** toevoegen, na de **Refresh Action** van de **Dynamic Action**, die de waarde van P20\_gro\_id zet m.b.v. **Set Value. Fire on Page Load** moet op No staan. Zie hiernaast.

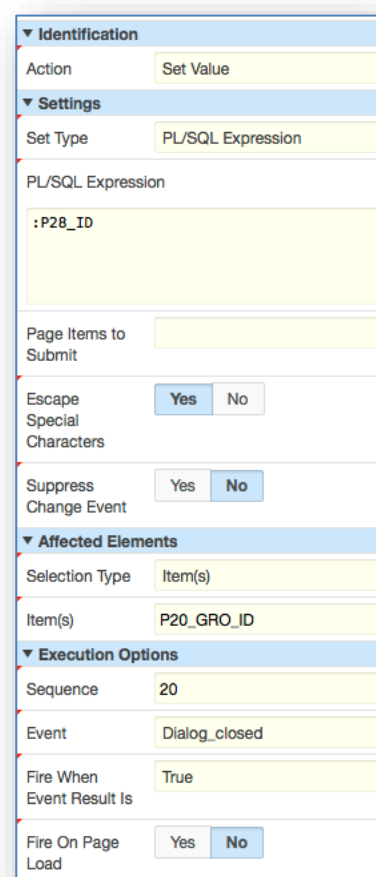
### Opmerking

Indien de LOV een popup was geweest, was deze oefening een stuk ingewikkelder geworden door de aard van een popup LOV in APEX. Daarnaast zouden er waarden moeten worden meegegeven door de Modal die daarna door de aanroepende pagina weer opgepakt moeten worden (zie bij Close Dialog Process Type de help tekst bij property Items to Return).

### Extra Oefening 1

Op deze manier kan er ook een algemene error en validatie handling worden opgezet die gebruik maakt van een **inline Dialog Region** i.p.v. de notificaties op het scherm. Dit kan ongeveer op de volgende wijze. Denk hier eens over na. Je hoeft het niet te maken, tenzij je genoeg tijd hebt.

- Maak een applicatie item G\_error.
- In de error handling functie moet de message worden opgeslagen met: `apex_util.set_session_state('G_ERROR',l_result.message);`
- Maak een Global Page 0, waarop een Inline Dialog Region wordt gemaakt met een hoog sequence nummer en met de tekst &G\_ERROR.
- Maak een Dynamic Action op Page 0, event Page Load, conditie G\_error IS NOT NULL. Eerste Action is het openen van de Modal, tweede Action is het leeg maken van G\_error.



- Eventueel #NOTIFICATION\_MESSAGE# verwijderen uit het Page Template.
- Opmaken van de Modal error Region.
- Uitbreiden voor Validaties, ook g\_error vullen.

Ook dit kan met een Modal Page, maar is iets complexer.

### **Extra Oefening 2**

Er zit in het datamodel ook de mogelijkheid om announcements te koppelen aan een Group. Maak daar een soortgelijke scherm voor op pagina 15 als de user-announcements. Je kan daar zelfs de schermen 25 en 26 voor aanpassen en uitbreiden, zodat ze 'herbruikbaar' worden.

## 10. JavaScript en jQuery

In deze oefening zullen we een aantal plaatsen zien waar we JavaScript kunnen toepassen. Daarnaast zullen we een toepassing zien van wat de jQuery library te bieden heeft.

### JavaScript op itemniveau

- Ga naar pagina 20 en bekijk de properties van het item 'city'. Navigeer naar de property **Advanced - Custom Attributes**. Lees ook de help tekst.
- Vul het volgende in: `onBlur="javascript:{this.value = this.value.toUpperCase(); }"`
- Bekijk het effect van deze toevoeging.
- APEX biedt zelf een beter alternatief. Zie bijvoorbeeld bij de **Server-side Condition** van het **Type Item IS NOT NULL**. Als je daar iets intypt is dit meteen in uppercase. Dit wordt door een CSS Class gedaan. Bekijk hoe dit werkt en gebruik dit i.p.v. de JavaScript functie.

### Context menu met jQuery

Een context menu is het menu dat verschijnt na het klikken op de rechtermuisknop. We gaan nu op het onderhoudsscherm voor **Main Tabs** een context menu toevoegen. Met dit menu willen we een record kunnen deleten en de description kunnen updaten naar uppercase.

Het menu maken we met een plugin van jQuery; een stukje JavaScript dat is ontwikkeld voor de jQuery library.

- Maak een simpel **Classic Report** op o.b.v. de main\_tabs tabel op pagina 11. Maak de 'id' kolom hidden.
- Als eerste voegen we een item toe dat de waarde vasthoudt voor het geselecteerde id, waarop we een actie willen uitvoeren. Maak daartoe een item genaamd P11\_ID. Dit mag een hidden item zijn, maar mag *niet Protected* zijn, omdat de waarde door het menu zal worden gewijzigd.
- Bij de JavaScript property op Page niveau **Function and Global Variable Declaration** moet de tekst van het script 'context\_menu.txt' worden gekopieerd. De gekopieerde code is de basis JavaScript functie voor het context menu.
- Vervolgens maken we de menu opties zelf. Dit doen we ook op page niveau, maar bij **Header text** het is immers gewoon HTML.

```
<div class="contextMenu" id="myMenu1">
  <ul>
    <li id="DEL"><span inline="backgroundcolor:red;" class="fa fa-trash-o"> Delete</span></li>
    <li id=""><hr/></li>
    <li id="UPPER"><span class="fa fa-change-case"> Upper</span></li>
  </ul>
</div>
```

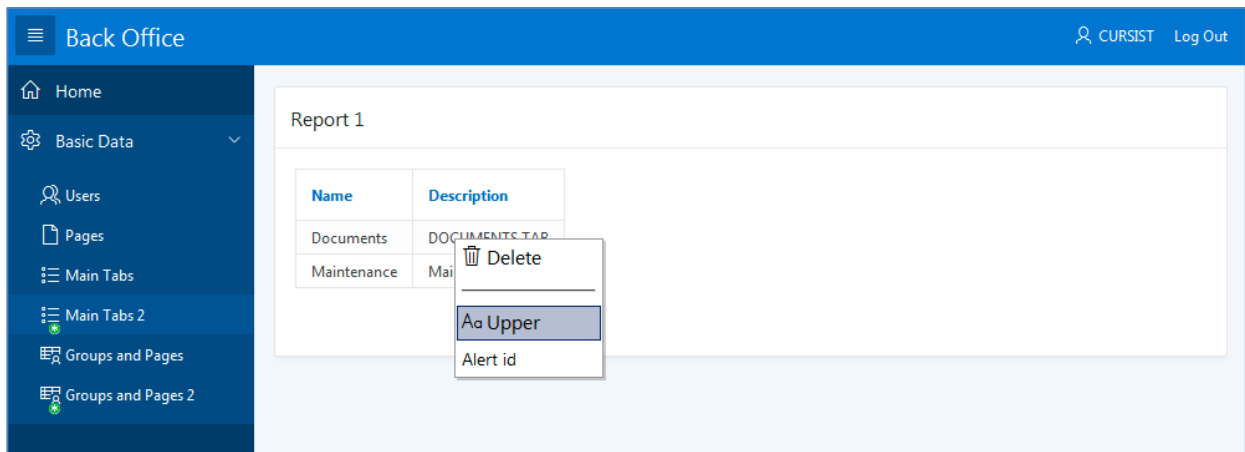
Duidelijk zijn de twee menuopties te zien en de scheidingslijn (hr) ertussen. Aan het eind van het vorige script zie je de code '\$(div.contextMenu).hide();' staan, wat er voor zorgt dat bovenstaande menuopties onzichtbaar worden gemaakt i.p.v. dat er een HTML list wordt weergegeven.

- Nu gaan we de code toevoegen welke de verschillende menu optie *acties* definiëren. Kopieer daartoe de tekst van het bestand context\_menu\_2.txt in de property **Execute when Page Loads**. Hieronder staat een fragment van de code. Hiermee wordt bewerkstelligd dat wanneer menu optie DEL wordt gekozen, het request de waarde DEL krijgt en dat P11\_ID als waarde het id van het huidige record krijgt.

```
'DEL': function(t) {
  apex.submit( {request:'DEL'
    ,set:{'P11_ID':
      $(t.closest('tr').find('span').attr('id') }
    }
  );
},
```

- Om ervoor te zorgen dat P11\_ID in bovenstaande code de waarde van het id krijgt door middel van de jQuery code '\$(t.closest('tr').find('span').attr('id'))', moet er in het rapport HTML worden toegevoegd aan een van de reeds bestaande kolommen. Voor bijvoorbeeld de 'name' column kan bij de de property **HTML Expression** de volgende waarde in worden gevuld:  
`#NAME# <span class="c_menu" id="#ID#"/>`

- Run het scherm en kijk of het context menu verschijnt na het klikken op de rechtermuisknop.
- Zorg ervoor dat beide menuopties daadwerkelijk gaan werken.
- Maak een extra menu optie wat het 'id' van het huidige record in een alert laat zien.



Bovenstaande is een mooi voorbeeld van het gebruik van jQuery en laat zien hoe je relatief eenvoudig een jQuery-plugin kan integreren in je pagina. Helemaal krachtig wordt het als bijvoorbeeld de menuopties en acties dynamisch met PL/SQL worden gegenereerd o.b.v. de inhoud van enkele tabellen en het 'huidige' paginanummer.

We hebben al eerder een jQuery oplossing gezien om records in **Classic** en **Interactive Reports** te highlighten.

## 11. Authentication, Front Office, Badges

In deze oefening gaan we de Front Office applicatie maken. Voor toegang tot deze applicatie maken we een authentication scheme dat een controle uitvoert tegen de user tabel door middel van een zelf gemaakte functie. Daarnaast maken we Badge functionaliteit in de Back Office.

### Front Office applicatie en eerste pagina

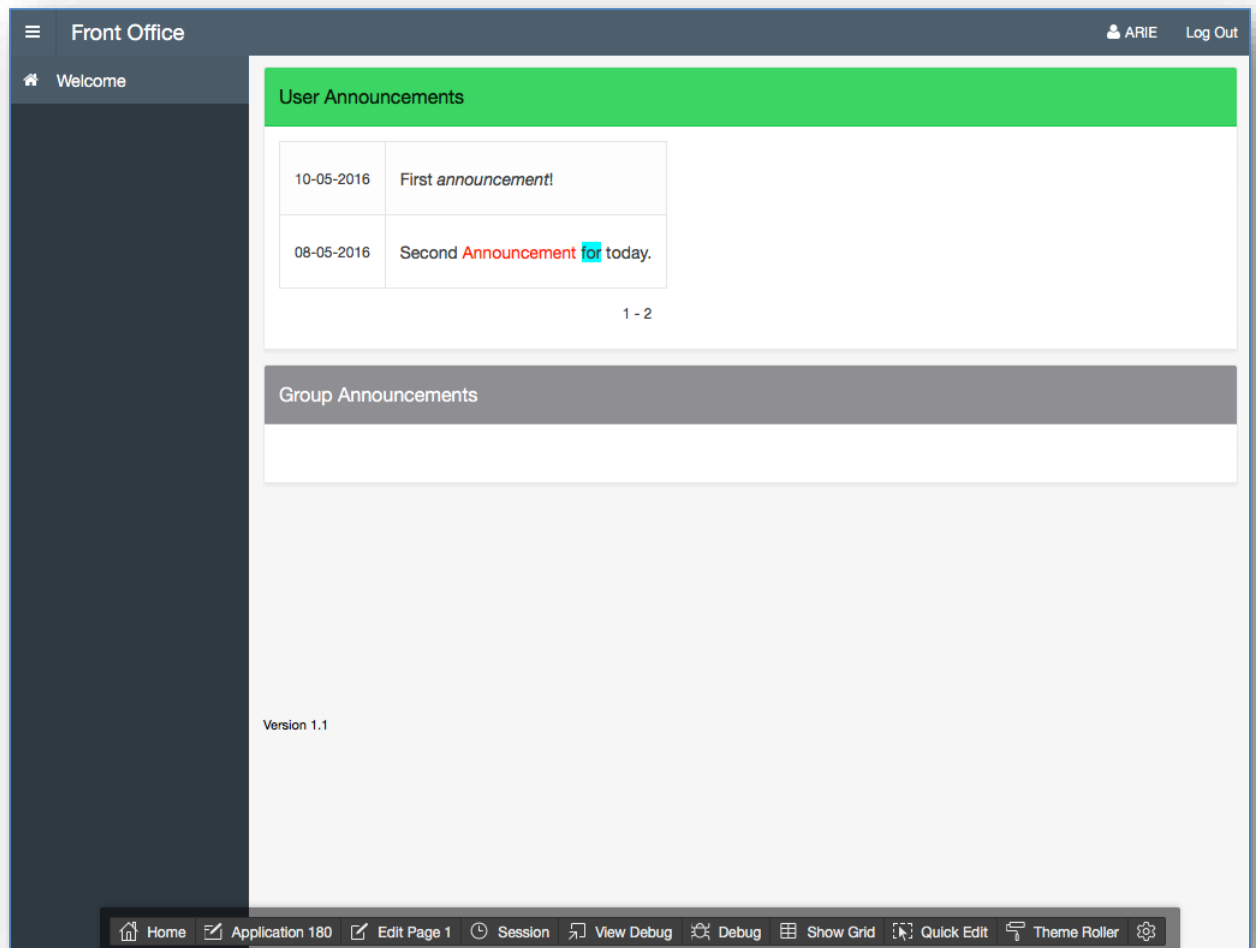
- We gaan nu de Front Office applicatie maken. Dit doen we grofweg hetzelfde als de Back Office applicatie. Start in de **Application Builder** de wizard om een applicatie te maken.
- Kies Desktop Application, noem de applicatie 'Front Office' en gebruik nummer **180**. Baseer de applicatie op het schema 'cursist'.
- Kies als Theme Style Vita-Slate (grijs tinten) om straks snel het onderscheid tussen beide applicaties te zien.
- Wijzig de naam van de **Blank Page** in 'Welcome'.
- Let op. Bij de vraag om shared components van een andere applicatie te kopiëren kiezen we nu **Yes**. We willen van applicatie 160 **User Interface Themes, List of Values en Navigation Bar Entries** overnemen.
- Bij de vraag **Subscribe Theme** kiezen we **No**. We willen eventueel zelf makkelijk aanpassingen kunnen doen.
- Gebruik als authenticatie schema het default **Application Express** schema.
- Datum formaat dd-mm-yyyy.
- Zet na afloop van de wizard een versie nummer.
- Wijzig het icoontje van de Welcome pagina in fa-home.

Bugje: niet alle Bar Entries worden gekopieerd en de Theme Style wordt niet goed gezet. Pas dit eventueel aan.

De applicatie is nu ook door dezelfde users te gebruiken die toegang hebben tot de Back Office applicatie. Deze users zijn voor de gehele **Workspace** geldig als er voor **Application Express schema Authentication** wordt gekozen. We willen de toegang echter beperken tot de users die we in de Back Office hebben aangemaakt (in de users tabel). Daartoe moeten we een functie maken die exact de volgende signature heeft: **(p\_username in varchar2, p\_password in varchar2) return Boolean**. Om deze functie correct te kunnen maken moeten we kijken wat er precies wordt doorgegeven aan deze functie door page 101 (de login page). Bekijk de parameters en de procedure (m.b.v. SQL Developer) die in het **Login** proces van pagina 101 wordt gebruikt.

- Maak deze functie met behulp van SQL\*Plus / SQL Developer of SQL Workshop. Laat de functie ook rekening houden met het feit dat de user wel actief moet zijn. In een productie omgeving is het raadzaam om deze functie in een package op te nemen.
- Ga nu naar de **Shared Components** en vervolgens naar de **Authentication Schemes**. Kijk welk **scheme current** is en selecteer deze. Wijzig het **Scheme Type** in **Custom**. Lees de helptekst van de property **Authentication Function Name** goed door en voer hier de naam in van de functie die we zojuist gemaakt hebben. Pas de naam van het scheme aan om aan te geven dat dit niet meer het origineel is.
- Maak een aantal **users** en **groups** aan in de de Back Office applicatie voor zover die er nog niet zijn en test de toegang tot de Front Office.
- Maak nog een functie die het id teruggeeft uit de **users** tabel van de ingelogde gebruiker. Dit id zullen we vaker nodig hebben in de Front Office applicatie en zullen dit dan ook bewaren in een **Application Item**. Deze items zijn te vinden bij de **Shared Components**. Maak deze aan met als naam **g\_use\_id**.
- Ga weer terug naar het aangepaste **Authentication Scheme**. Lees de helptekst van de property **Post-Authentication Procedure Name** en voeg de bijbehorende PL/SQL toe om **g\_use\_id** te vullen.
- Ga naar de eerste pagina.
- Verwijder de **breadcrumb** en de geassocieerde list entries.

- Voeg twee regions toe die de user announcements laten zien en de eventuele (indien de extra oefening is gemaakt) group announcements van de ingelogde user. Houd ook rekening met `ind_active` en ordenen op `date_inserted`.
- Zet **Heading - Type** property op None van de attributes van het rapport.
- Zet **Escape Special Characters** op No om de opmaak te behouden. Merk op dat dit een potentieel gevaar is (later).
- Het resultaat na inloggen kan er ongeveer zo uit zien:



### We gaan nu een self service pagina maken.

- Maak in de Front Office applicatie een pagina met een **Form** o.b.v. Users met nummer 20 dat via een button op de loginpagina te bereiken is met de tekst 'Register'. De buttons **Delete** en **Save** mogen verwijderd worden. Na Cancel of Submit moet weer naar pagina 101 genavigeerd worden. De pagina hoeft geen geassocieerde tab te hebben. Indien de (potentieële) gebruiker op de register **Button** klikt, komt hij op een pagina waar hij z'n gegevens en de group kan invullen.
- Per pagina is in APEX in te stellen of deze publiekelijk toegankelijk is of dat het authentication scheme toegepast moet worden, zie de **Page Attributes** onder het kopje **Security**. Deze pagina moet uiteraard publiekelijk toegankelijk zijn. Gebruik hetzelfde template voor de pagina als de login pagina.
- Default moeten de user gegevens met 'ind\_active' op 'N' worden gesubmit, zodat ie inactive is. Indien iemand nu op de Back Office 'active' op Yes zet kan de gebruiker inloggen in de Front Office applicatie.
- Controleer of het werkt.
- Het resultaat kan er ongeveer zo uit zien:

Front Office nobody Log Out

### Register New User

Group \*

Username \*

Password \*

Firstname \*

Lastname \*

Gender \*  Male  
 Female

Date Of Birth

Address \*

City \*

Mob Nr

Email \*

Tel Nr

---

Home Application 180 Edit Page 20 Session View Debug Debug Show Grid Quick Edit Theme Roller

## Badges

We gaan er nu voor zorgen dat een medewerker van de Back Office applicatie snel kan zien of er nieuwe mensen zijn die zich via de self service pagina aangemeld hebben. Dit gaan we met **Badges** doen.

- Ga naar applicatie 160.
- Maak een **Application Item**: G\_BADGE\_USERS.
- Maak een **Application Computation** voor dit Item, **Computation Point**: Before Header en als **Computation Type**: SQL Query (return single value):

```
SELECT DECODE( count(1)
              , 0, null
              , '['||count(1)||']')
FROM users
WHERE ind_active = 'N'
```

- Ga naar het **Navigation Menu** en wijzig het **List Entry Label** van Users in: 'Users &G\_BADGE\_USERS.'
- Doe *eventueel* iets soortgelijks voor het niveau erboven: Basic Data.
- Maak een kopie van **Side Navigation Menu- List Template** en
- Voeg onderstaande CSS toe aan de kopie van het **Template**, helemaal onderaan op het **inline** gedeelte:

```
.a-TreeView-badge {
  background-color:red !important;
  color: white;
  font-weight: bold;
}
```

- Bug: Door een bug moet dit op een andere plek gebeuren voor APEX 5.1, nl.: **List Template Before Rows. Daarnaast moet het CSS tussen <style>...</style> tags gezet worden.**
- Wijzig de **User Interface** voor de **Application**, zodat het gebruik maakt van deze nieuwe **Template** voor de **Navigation Menu - List**. Het resultaat ziet er als volgt uit:

Edit	Name ↑	Group	City	Address	Username	Email	Active?
	Arie Boeren	Finance	ALKMAAR	Koningsstraat 13	arie	arie@company.be	Yes
	Sjon de boer	ICT	AMSTERDAM	Keizersgracht 43	sjon	sjon@werken.nl	Yes
	qw qw	ICT	qw	qw	qw	qw	No

### Extra Oefening

Maak de inlogpagina ietsje mooier. Zie de oefening voor de Back Office.

## 12. Authorization

In deze oefening gaan we de **Authorization** die we in de Back Office hebben gedefinieerd toepassen op de users die inloggen op de Front Office.

- Maak in de Front Office applicatie alvast twee Blank Pages en maak er Menu Entries voor aan:
  - Page 5: Documents
  - Page 10: Maintenance
- Maak een functie die o.b.v. de ingelogde user en een paginanummer bepaalt of de user daar toegang tot heeft.
- We gaan nu een **Authorization Scheme** maken in de Front Office applicatie, gebaseerd op deze functie. Kies voor **PL/SQL Function Returning Boolean** en geef een error message en naam aan het scheme. Belangrijk is dat de functie op elke pagina weer opnieuw wordt aangeroepen (zie **Evaluation Point**). De functie kan er ongeveer zoals hieronder uitzien.

```
BEGIN
    RETURN pk_security.page_valid( p_use_id => :g_use_id
                                   , p_pag_nr => :APP_PAGE_ID);
END;
```

- Ga nu naar **Security Attributes** om het **Authorization Scheme** voor de gehele applicatie (elke pagina) toe te passen. Selecteer het nieuwe 'Scheme'. Om er voor te zorgen dat deze functie niet wordt aangeroepen indien de user niet bekend is (g\_use\_id is null), kiezen we voor **Run on Public Pages** de waarde 'No'. Leg de wijzigingen vast.
- Belangrijk is dat iedere groep toegang heeft tot pagina 1! Anders wordt de toegang alsnog verhinderd net na het inloggen. Bij de **User Interface** properties van de applicatie wordt namelijk vastgelegd naar welke pagina wordt genavigeerd na een succesvolle inlogactie. Bekijk deze instelling.
- Nu leggen we voor elke **Menu List Entries** een functie aanroep vast bij **Condition**. We kiezen weer voor **PL/SQL Function Returning Boolean**. De functie ziet er ongeveer als volgt uit:

```
BEGIN
    RETURN pk_security.page_valid( p_use_id => :g_use_id
                                   , p_pag_nr => 1);
END;
```

Dit werkt, maar is uiteraard erg omslachtig. We moeten voor elke **Menu List Entry** dus een andere aanroep doen. Merk op dat de page beveiliging het belangrijkste is. Je kan immers in de url direct een pagina oproepen. Het feit dat de **Menu List Entries** niet zichtbaar zijn maakt het wat netter.

Let op: Vanaf nu is het belangrijk om elke nieuwe toegevoegde pagina en **Menu Entry** (Main Tab) aan de Front Office applicatie te registreren in de Back Office en de verschillende users toegang tot deze pagina's en **Menu Entry** (Main Tab) te verlenen.

### Extra oefening:

- Maak een Dynamic List o.b.v. de pages tabel, die als basis dient voor de entries in het Navigation Menu. Koppel deze als **Navigation Menu** bij de **User Interface** properties. In de praktijk zal dit toch wat ingewikkelder worden door zaken, zoals:
  - Meerdere paginas per Menu Entry
  - Lager niveau toegang en hoger niveau niet (aan de tussenliggende niveaus kunnen ook pages gekoppeld zijn)

Een voorbeeld van de query zou er als volgt uit kunnen zien:

```
SELECT null
  , pag.name label
  , 'f?p=&APP_ID.:'| pag.nr ||':':APP_SESSION target
  , null is_current
  , 'fa-bars' image
FROM users use
  , group_pages gpa
  , pages pag
```

```
WHERE use.id = :g_use_id  
AND use.gro_id = gpa.gro_id  
AND gpa.pag_id = pag.id  
ORDER BY pag.nr
```

**Tip!**

De authorization instellingen op applicatie niveau overruled de instelling op pagina niveau, indien op applicatie niveau het **Authorization Scheme** tot **False** evalueert.

## 13. Session State Protection

We gaan in deze oefening de **Session State Protection** gebruiken.

- Voer voor beide applicaties het volgende uit. Ga naar Shared Components en klik op **Session State Protection**. Klik op de button **Set Protection** en zorg dat **Enable** staat aangevinkt. Klik op **Next** en op **Enable**.
- Klik opnieuw op **Session State Protection**. Klik op de button **Set Protection** Vink nu **Configure** aan en geef aan dat overal op session niveau checksums moeten worden gebruikt. **Page Access Protection** moet ook met checksums werken. Druk op **Next** en er is een overview te zien van de acties. Druk op **Finish** en bekijk de applicaties en URL's.
- Log uit van applicatie 160 en log weer in. Probeer vervolgens in deze applicatie de URL te 'tamperen' en kijk wat er gebeurt.

### Tip!

Vanaf versie 5.0 Staat Session state protection standaard aan.

## 14. Cross Site Scripting (XSS)

In deze oefening kijken we naar **Cross Site Scripting** en wat we daar aan kunnen doen.

- Ga naar pagina 5 van applicatie 160. Open de properties van de 'description' kolom in het rapport. Zet de **Escape Special Characters** property op **No**. Run de pagina. Voer als description van de page het volgende in: `test<script>alert('Hello world');</script>`. Submit de pagina en zie wat er gebeurt.
- Dit is op te lossen door het gebruik van `htf.escape_sc('string')` in de query of door **Escape Special Characters** property op **Yes** te zetten. Probeer dat in de query toe te voegen voor de description en zie wat er gebeurt. Tegenwoordig is de default het escapen van de inhoud. In een **Interactive Report** is er een kolom type genaamd **Remove HTML**.

Op het rapport is bij **Attributes** de property **Strip HTML** aan te zetten, maar dat werkt niet voor het displayen van waarden (zie de help).

Tenslotte is er een APEX package genaamd APEX\_ESCAPE met vele varianten van escape-en.

## 15. Files in de database

In deze oefening maken we een scherm om files een voor een te uploaden met de APEX functionaliteit **Declarative BLOB Support**. De tabel heeft daarvoor 4 velden nodig: de blob kolom voor het document en daarnaast drie technische velden die door APEX zelf worden onderhouden t.w. mimetype, filename en last\_update\_date.

- Open de nu nog lege pagina 10 'Maintenance' in de Front Office applicatie. Maak een **Report with Form on Table** scherm voor de documents tabel. **Form** en **Report** op pagina 10. Gebruik een 'classic report' in plaats van een 'interactive report' als **implementation** property.
- Het report moet de blob kolom bevatten. Het form hoeft de bovengenoemde drie technische kolommen niet te bevatten, deze worden later in de oefening automatisch gevuld.
- Let op dat het **Storage Type** van de documents kolom (het page item, niet de report column!) staat op 'BLOB column specified in Item Source attribute'. Indien dit op 'Table APEX\_APPLICATION\_TEMP\_FILES' (=WWV\_FLOW\_FILES) staat wordt de file in de algemene tabel gezet i.p.v. de BLOB kolom die we in onze eigen tabel hebben.
- Zet de date\_created kolom op sysdate als default. Het scherm ziet er ongeveer als volgt uit:

The screenshot shows the 'Front Office' application interface. On the left is a navigation menu with 'Welcome', 'Documents', and 'Maintenance'. The main area is titled 'Documents' and contains a table with the following data:

Edit	Document	Date Created	Description	Filename	Mimetype	Last Update Date
	<a href="#">Download</a>	08-05-2016	Begroting			
	<a href="#">Download</a>	08-05-2016	Beleid			
	<a href="#">Download</a>	08-05-2016	Contract de Ronde Venen			
	<a href="#">Download</a>	08-05-2016	Notepad	Notepad document.txt	text/plain	09-05-2016
	<a href="#">Download</a>	09-05-2016	Werkeloosheid	Werkeloosheid[1].ppt	application/powerpoint	09-05-2016

Below the table is a 'Document' form with the following fields:

- Document \*  No file selected. [Download](#)
- Date Created: 08-05-2016
- Description: Notepad

Buttons: Cancel, Delete, Apply Changes

- Let op: In oudere APEX versies (<= 4.0) kunnen problemen ontstaan met de validatie indien alles op één pagina staat. Zorg dan dat de gegenereerde validatie die de kolom p10\_documents controleert, niet afgaat bij het indrukken van de delete button. Dit is het gevolg van HTML, dat na submitten een **browse item** leeg maakt (evenals een **password item**).
- Zet het P10\_document **Required** property op **No**. i.v.m. het updaten van een record.
- Bugje in APEX 5.0 Zet Session State Protection uit voor P10\_id.

- Probeer een van de 5 files die op het filesystem staan te uploaden en kijk hoe dat gaat. Nadat er een file is geupload, verschijnt er een 'download' link in het rapport op de plaats van het document. Als je daarop klikt, kan de file weer worden gedownload en worden weggeschreven naar het filesystem.
- Wat opvalt, is dat de laatste 3 kolommen niet gevuld zijn. Daardoor weet APEX ook niet wat voor filetype het is en de naam is, als de file wordt gedownload om te openen of weg te schrijven. Om dit op te lossen moeten er nog wat zaken geconfigureerd worden. Navigeer naar de properties van p10\_document. Daar is het volgende te zien bij de property groep **Settings**:

Settings

Storage Type: BLOB column specified in Item Source attribute

MIME Type Column: MIMETYPE

Filename Column: FILENAME

Character Set Column:

BLOB Last Updated Column: LAST\_UPDATE\_DATE

Display Download Link: Yes No

Download Link Text:

Content Disposition: Attachment

File Types:

- Vul op de lege plaatsen de namen in van de database kolommen (MIMETYPE, FILENAME en LAST\_UPDATE\_DATE) en zet Content Disposition op 'attachment'. Let op de kolomnamen zijn case sensitive en moeten in hoofdletters worden ingevoerd. Vanaf nu worden de kolommen tijdens het uploaden automatisch gevuld.
- Verwijder het geuploade document en upload nu alle 5 documenten. Nu zijn de velden gevuld. Zie wat er gebeurt als je nu op de link in het form klikt. Dit is een stuk netter.

Documents

Edit	Document	Date Created	Description	Filename	Mimetype	Last Update Date
	<a href="#">Download</a>	08-05-2016	Begroting	Begroting_2008.xls	application/vnd.ms-excel	09-05-2016
	<a href="#">Download</a>	08-05-2016	Beleid	Beleid_op_straat.txt	text/plain	09-05-2016
	<a href="#">Download</a>	08-05-2016	Contract de Ronde Venen	Contact_De_Ronde_Venen.doc	application/msword	09-05-2016
	<a href="#">Download</a>	08-05-2016	Notepad	Notepad document.txt	text/plain	09-05-2016
	<a href="#">Download</a>	09-05-2016	Werkeloosheid	Werkeloosheid[1].ppt	application/powerpoint	09-05-2016

1 - 5

Create

- Nu moet iets soortgelijks worden gedaan voor de 'download-linkjes' in het rapport. Navigeer naar de rapportkolom Document en bekijk de property groep **Blob Attributes**. Pas de 3 kolommen aan. T.o.v. versie 4.2.1 zijn deze properties (weer) gewijzigd en werkt dit nu goed. Zie de volgende afbeelding:

▼ BLOB Attributes	
Table Owner	Parsing Schema
Table Name	DOCUMENTS
BLOB Column	DOCUMENT
Primary Key Column 1	ID
Primary Key Column 2	ID
Mime Type Column	MIMETYPE
Filename Column	FILENAME
Last Updated Column	LAST_UPDATE_DATE
Character Set Column	- Select -
▼ Appearance	
Download Text	
Content Disposition	Attachment

- Probeer bij **Content Disposition** zowel **Attachment** als **inline** uit en bekijk de verschillen. Indien we ook plaatjes hadden geupload konden die met **inline** meteen worden getoond i.c.m. het **Type 'Display Image'** i.p.v. **'Download BLOB'**. Nadeel van die methode is dat je geen controle hebt over b.v. de grootte van de plaatjes.

Door gebruik te maken van de functie **apex\_util.get\_blob\_file\_src**, heb je als ontwikkelaar meer controle over de manier waarop de files (in een rapport gedisplayed) kunnen worden.

#### Extra oefeningen.

- Probeer een plaatje te uploaden en te laten displayen in het rapport.
- Maak een extra kolom aan in het rapport, analoog zoals in de help tekst staat van de **(onhandige)** functionaliteit van **apex\_util.get\_blob\_file\_src**:

([http://download.oracle.com/docs/cd/E10513\\_01/doc/apirefs.310/e12855/apex\\_util.htm#insertedID28](http://download.oracle.com/docs/cd/E10513_01/doc/apirefs.310/e12855/apex_util.htm#insertedID28))

```
SELECT ID,
       NAME,
       CASE WHEN NVL(dbms_lob.getlength(document),0) = 0
            THEN NULL
            ELSE CASE WHEN attach_mimetype like 'image%'
                    THEN '<img
src=""||apex_util.get_blob_file_src('P4_DOCUMENT',id)|| "' />'
                    ELSE '<a
href=""||apex_util.get_blob_file_src('P4_DOCUMENT',id)|| "'>Download</a>'
                    end
       END new_img
FROM TEST_WITH_BLOB
```

Met deze functie kan je in de **<img>** tag height en width meegeven. Het makkelijkste is om zelf een procedure te maken.

## 16. Integratie

In deze oefening zullen naar integratie aspecten van APEX kijken. We gaan eerst het gebruik van **Oracle Text** aan onze applicatie toevoegen.

### Oracle Text

Oracle Text biedt de mogelijkheid om documenten te indexeren en te doorzoeken. Dit kan voor meer dan 400 typen (zoals doc, pdf, xls, txt, etc.). Daarnaast kunnen deze documenten worden omgezet naar HTML. De mogelijkheden van Oracle Text zijn zeer uitgebreid en we bekijken hier enkele aspecten.

- Maak een Text index op de documenten tabel m.b.v. **SQL Developer** of de **SQL Workshop**:  
CREATE INDEX doc\_index ON DOCUMENTS (document) INDEXTYPE IS ctxsys.CONTEXT  
PARAMETERS ('SYNC (ON COMMIT) TRANSACTIONAL')
- Indien dit niet lukt i.v.m. rechten kan eerst nog als sys het volgende worden uitgevoerd:  
GRANT EXECUTE ON SYS.DBMS\_RLS TO CURSIST;  
GRANT EXECUTE ON CTXSYS.CTX\_DDL TO CURSIST;  
GRANT EXECUTE ON CTXSYS.CTX\_DOC TO CURSIST;
- Maak een **Region** in scherm 5 'documenten' van de Front Office applicatie om de gebruikers te laten zoeken in de documenten tekst en de metadata. We beginnen met een **Report Region** en een **Classic Report**. We beginnen de query op dit moment eenvoudig:

```
SELECT doc.id          id
       , doc.description description
       , doc.last_update_date last_update_date
       , doc.filename   filename
       , doc.mimetype   mimetype
FROM documents doc
```

- Maak de layout van het rapport mooier door de **Headings, Sorting**, etc. aan te passen. Vul ook de velden van de document kolom, zoals op Page 10.
- We gaan nu zelf een zoekveld maken. Maak in deze region een gewoon **Page Item** van het type **Text Field**, genaamd P5\_search.
- Voeg naast het zoekveld een **Button** toe die na drukken de **Page** submit (en weer terugspringt naar pagina 5).
- Wijzig de query dusdanig dat deze de zoekresultaten beperkt tot records, waarbij het zoekveld voorkomt in de velden 'description' of 'filename'. Laat de zoekstring gehighlight zien in deze 2 velden in de resultaten.
- Tot nu toe zoekt de query alleen in de metadata (de velden 'description' en 'filename'). Nu willen we gebruik maken van de index Voeg aan de query de volgende conditie toe:

```
CONTAINS(doc.document, '%'||:P5_SEARCH||'%', 1) > 0
```

**Contains** is een Oracle Text operator en kan worden opgenomen in de query. Bekijk het resultaat en experimenteer met het zoeken. Indien nu een document gevonden wordt die slechts matcht op de inhoud en niet op de metadata kan je dit niet direct zien, er wordt niets gehighlight.

- Om voorgaande tekortkoming op te lossen gebruiken we de highlight functionaliteit van **Oracle Text**. Op het filesysteem staat een bestand ctx.sql. Gebruik deze code om de functie get\_highlight\_text te maken. Het voert te ver om de code hier uitgebreid te bespreken, maar voor nu is het belangrijk om te weten dat deze functie een text fragment teruggeeft met daarin gehighlight de gezochte text. Voeg onderstaande 'kolom' toe en bekijk het resultaat.

```
, get_highlight_text('doc_index',doc.id,'%||:P5_SEARCH||%',40) text
```

Omdat deze kolom gebruik maakt van HTML en we dit ook willen zien, moet **Escape Special Characters** op **No** staan.

- Zie hieronder een voorbeeld van het resultaat:



```

var options = {title:'How Much Pizza I Ate Last Night',
              'width':400,
              'height':300};

var chart = new google.visualization.PieChart(document.getElementById('chart_div'));

chart.draw(data, options);
}
</script>
</head>

<body>
<div id="chart_div"></div>
</body>
</html>

```

- We willen dit met een **PL/SQL Dynamic Content** region maken. We moeten dus een stukje PL/SQL schrijven dat bovenstaande HTML genereert. Vervolgens kunnen we met de htp.p procedure deze HTML in een region krijgen. We beginnen met het maken van een **PL/SQL Dynamic Content** region onder de twee reeds gemaakte regions. Onderstaande code genereert de benodigde HTML voor de region:

```

DECLARE

    l_term_string VARCHAR2(1000);

CURSOR c_ste IS
    SELECT '[' || term || ',' || COUNT(1) || ']' term
    FROM search_terms
    GROUP BY term
    ORDER BY 1;

BEGIN

    FOR r_ste IN c_ste LOOP
        l_term_string := l_term_string || ',' || r_ste.term;
    END LOOP;
    l_term_string := substr(l_term_string,2);
    htp.p('
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
google.charts.load("current", {"packages":["corechart"]});
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
    var data = new google.visualization.DataTable();

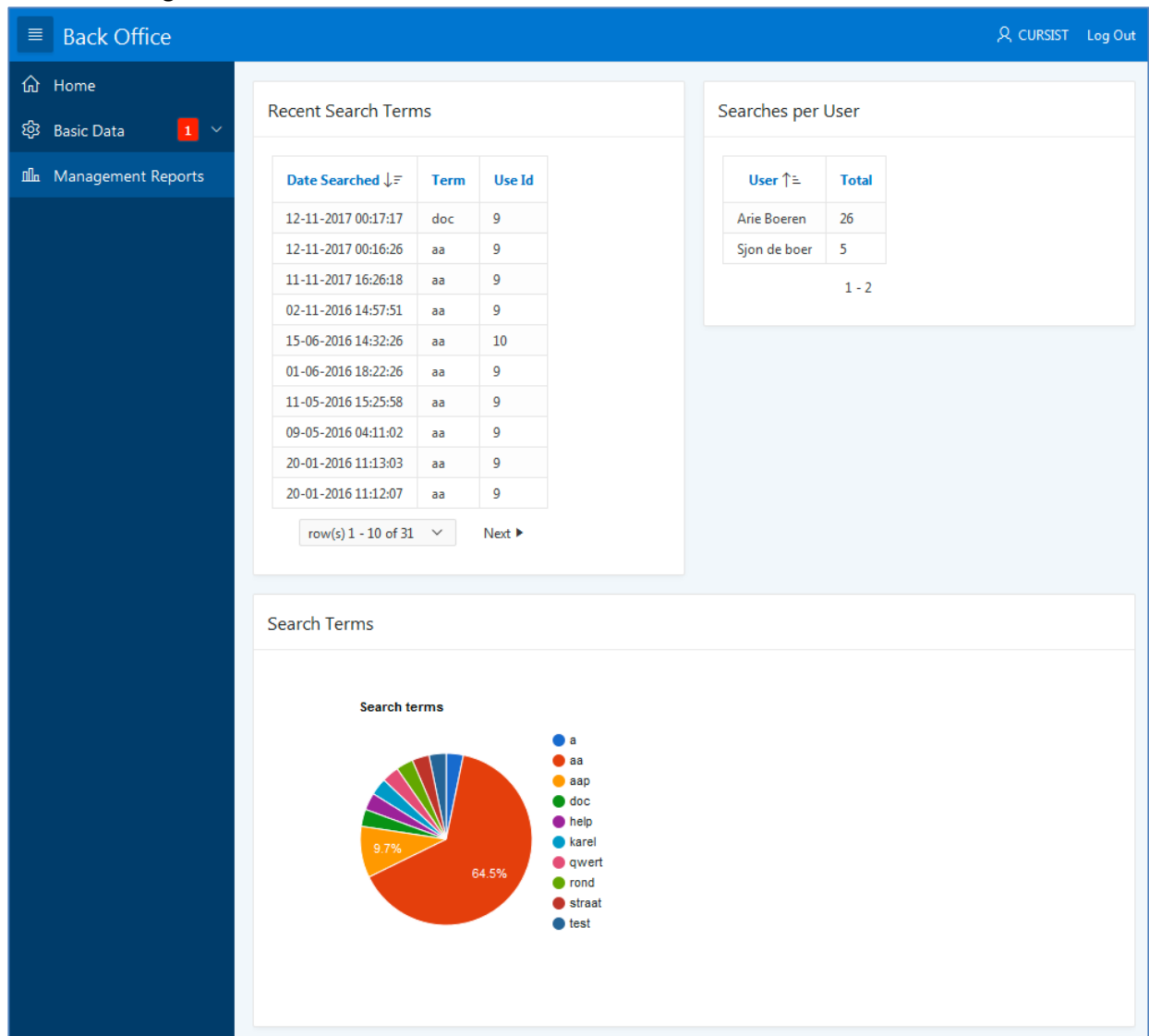
    data.addColumn("string", "Term");
    data.addColumn("number", "nr");
    data.addRows([
]);
    htp.p(l_term_string);
    htp.p(' ');
    var options = {"title":"Search terms",
                  "width":400,
                  "height":300};
    var chart = new google.visualization.PieChart(document.getElementById("chart_div"));
    chart.draw(data, options);
}
</script>
<div id="chart_div"></div>
');

END;

```

Dit script staat in het bestand google.txt. Probeer te begrijpen hoe dit werkt. Kijk op de site van Google om wat te experimenteren met kleuren, charttypes, etc.

- Geef nu wat zoekopdrachten om de search\_terms tabel te vullen. Het resultaat kan er ongeveer als volgt uit zien:



### Tips!

- Er zijn m.b.t. het JavaScript ook andere manieren om dit in de pagina te integreren. Een deel van de JavaScript kan bijvoorbeeld ook op page niveau gedefinieerd worden.
- Uiteraard kunnen hier ook de **charts** (inclusief automatische refresh) van APEX gebruikt worden, maar het gaat hierbij om een demonstratie van integratie. Daarnaast kunnen bijvoorbeeld ook analytische functies worden toegepast in de rapporten.
- Op de website van Google staat ook nog een oude versie die *echt* een plaatje teruggeeft.

Zie: [https://developers.google.com/chart/image/docs/gallery/pie\\_charts](https://developers.google.com/chart/image/docs/gallery/pie_charts)

Met een url als:

<https://chart.googleapis.com/chart?cht=p3&chd=t:60,40&chs=250x100&chl=Hello|World> wordt een pie chart teruggegeven. Integratie met APEX kan door dit simpel in een <img> gebruiken, zoals het voorbeeld hieronder:

```

```

Deze api is weliswaar 'deprecated', maar blijft voorlopig ook bestaan.

## 17. Plug-Ins

In deze oefening gaan we een zogenaamde 'star rating' item type plug-in gebruiken. We gebruiken deze om een document een rating te kunnen geven.

- Om de rating op te kunnen slaan moeten we de tabel documenten aanpassen. Voeg een kolom 'rating' toe met als datatype Number.
- Voeg de plug-in toe als shared component aan applicatie 180. Gebruik het script **Plugin\_star\_rating.sql**. Dit is de plugin die ook van de Oracle site te downloaden is. Bekijk de properties en druk op ok.
- Ga naar pagina 10. Voeg aan het rapport de kolom 'rating' toe.
- Voeg in het Form een item toe, van het type Plug-In, kies daarna voor het juist toegevoegde 'Star Rating'. Let erop dat het item gekoppeld wordt aan de nieuwe database kolom 'rating' en dat een waarde niet verplicht is. Zet het aantal 'stars' op 10.
- Het resultaat zal er ongeveer als volgt uitzien:

The screenshot shows the 'Front Office' application interface. On the left is a navigation menu with 'Welcome', 'Documents', and 'Maintenance'. The main area is titled 'Documents' and contains a table with the following data:

Edit	Document	Date Created	Description	Filename	Mimetype	Last Update Date	Rating
	<a href="#">Download</a>	08-05-2016	Begroting	Begroting_2008.xls	application/vnd.ms-excel	09-05-2016	
	<a href="#">Download</a>	08-05-2016	Beleid	Beleid_op_straat.txt	text/plain	09-05-2016	5
	<a href="#">Download</a>	08-05-2016	Contract de Ronde Venen	Contact_De_Ronde_Venen.doc	application/msword	09-05-2016	
	<a href="#">Download</a>	08-05-2016	Notepad	Notepad document.txt	text/plain	09-05-2016	
	<a href="#">Download</a>	09-05-2016	Werkeloosheid	Werkeloosheid[1].ppt	application/powerpoint	09-05-2016	

Below the table is a 'Document' form with the following fields:

- Document:  No file selected. [Download](#)
- Date Created:
- Description:
- Rating:

Buttons:

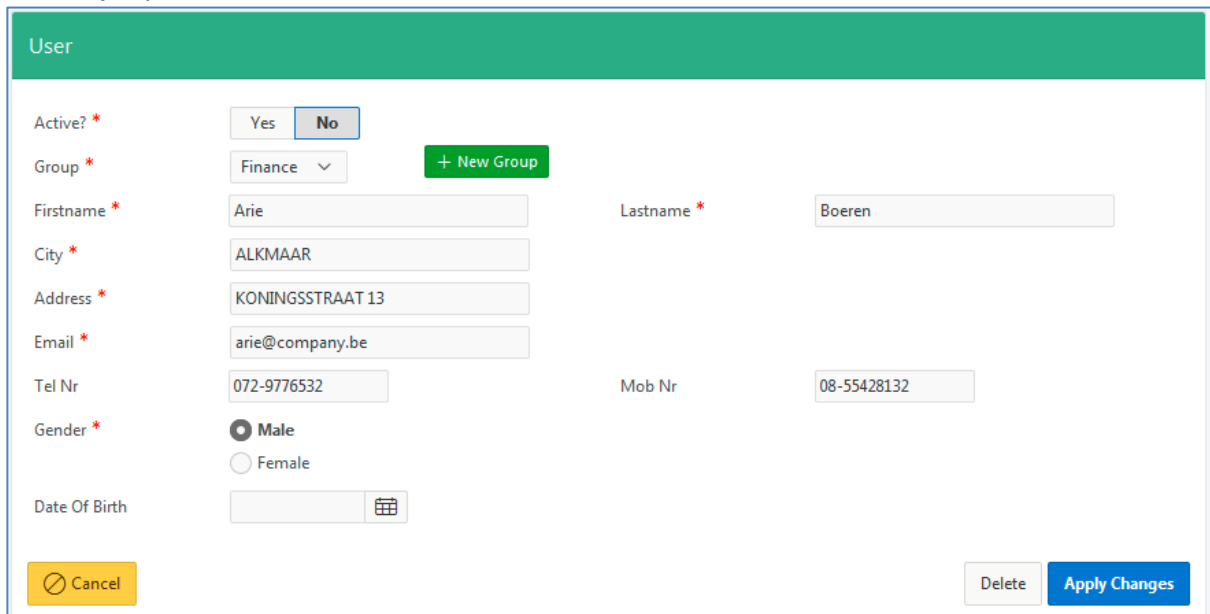
At the bottom of the application, there is a status bar with icons for Home, Application 180, Edit Page 10, Session, View Debug, Debug, Show Grid, Quick Edit, and Theme Roller.

## 18. Dynamic Actions

We gaan in deze oefening een aantal dynamic actions maken. Een eenvoudig hide/show oefening en een wat complexer refresh voorbeeld.

In de eerste oefening maken we een eenvoudige dynamic action op het onderhoudsscherm voor de users, pagina 20 in applicatie 160. We willen dat de velden username en password alleen zichtbaar zijn als een user actief is.

- Navigeer naar het item 'p20\_ind\_active'. Druk op de rechtermuisknop en selecteer **Create Dynamic Action. Event: Change**.
- Als **Client-side Condition**: P20\_IND\_ACTIVE 'equal' to Y.
- De **True Action** wordt **Show** en we maken ook een **Opposite False Action**, als ind\_active op N wordt gezet moeten de velden namelijk niet meer worden getoond.
- Vervolgens selecteren we de items die we willen verbergen.
- Bekijk op het scherm het resultaat.



Nu gaan we een iets ingewikkeldere **Dynamic Action** maken, die een report laat refreshen o.b.v. een selectlist. Op zich is dit niet nieuw, maar een Dynamic Action kan ervoor zorgen dat we dit kunnen maken zonder dat de pagina gesubmit wordt. Dit gaan we doen met de tabel 'announcements' en een selectlist o.b.v. de 'users'. Voor deze oefening is het belangrijk dat er wel wat announcements zijn gemaakt voor de users, om een en ander goed te kunnen testen.

- Maak een nieuwe pagina in Back Office applicatie. Kies daarvoor nummer 35. Hierop moet een report o.b.v. van de tabel 'announcements' worden gemaakt. Voeg een **Font APEX** plaatje b.v. fa-comment toe bij het **Menu**.
- Bekijk het rapport en maak wat aanpassingen om hem af te maken (ordering, headings, lov's i.p.v. ind\_active, verbergen van id en use\_id, gro\_id, etc.)
- Maak een nieuwe **Static Content** region dat vóór de reportregion gedisplays wordt.
- Maak op deze region een item P35\_use\_id. Het **displaytype** moet **Select List** zijn o.b.v. de user lov. Set de property **Warn on Unsaved Changes** uit.
- Pas de query van het rapport aan, zodat deze alleen de records laat zien van de geselecteerde user.
- Bekijk de pagina. Wat opvalt is dat een wijziging van de select list geen wijziging van het rapport als gevolg heeft (blijft leeg).
- Zorg er nu voor dat de pagina gesubmit wordt, wanneer de selectlist wijzigt. Bekijk de pagina opnieuw. Op zich doet ie nu wat de bedoeling is, ware het niet dat er steeds een page refresh plaatsvindt en dat de pagina helemaal opnieuw moet worden opgebouwd. Haal de **submit page** action er weer vanaf.
- Voeg een Dynamic Action toe aan de select list. Geef er een naam aan en zorg dat deze afgaat op het **Change** event, er is geen conditie nodig.

- Als action kiezen we **Component – Refresh**. Merk ook op dat er hier gekozen kan worden voor **Submit Page**. We kiezen vervolgens dat we de region ‘Announcements’ willen refreshen.
- Bekijk het resultaat. Wat opvalt is dat het rapport niet wijzigt. Dit is het gevolg van het feit dat de waarde van P35\_use\_id nog niet in session state is ‘gezet’ door de **Dynamic Action**.
- Om dit te kunnen doen kunnen we een extra **action** toevoegen, die vóór het refreshen moet plaatsvinden. Voeg een actie **Execute PL/SQL code** toe met als code: null; Dit lijkt vrij nutteloos, maar bij deze **action** is het mogelijk om items in de session state te zetten bij property **Page Items to Submit**. Bekijk de help van deze property en vul daarna P35\_USE\_ID in als item.
- Bij een classic report moet ook **Partial Page Refresh** property op **Yes** staan.
- Run nu de pagina en controleer of alles werkt.

### Tips!

- Vanaf versie 4.1 wordt de mogelijkheid om items te zetten makkelijker:  
*The new attribute "Page Items to Submit" has been added to regions of type IR, Classic Report, Tabular Form and plug-ins having that feature enabled. The values of the specified page items will be submitted with a "Partial Page Refresh" request to set them in session state, so that they can be used in the WHERE clause of the regions SQL statement. A "Partial Page Refresh" can be triggered with the "Refresh" dynamic action, by changing the sorting or by paginating.*
- Ook is het mogelijk om onderstaande JavaScript te gebruiken om een item in session state te zetten:
 

```

      /* set item value in session state */
      function setSessionValue(pitem, pValue) {
        var get = new htmldb_Get(null,$v('pFlowid'),null,0);
        get.add(pitem, pValue);
        get.get();
        get = null;
      }
      
```
- Tenslotte is het mogelijk om PL/SQL te gebruiken om een item in session state te zetten:
 

```

      APEX_UTIL.set_session_state(
        p_name => 'PX_MY_ITEM'
        , p_value => 'wibble');
      
```

### Resultaat:

The screenshot shows the Oracle APEX Back Office interface. The top navigation bar includes 'Back Office', 'CURSIST', and 'Log Out'. The left sidebar has 'Home', 'Basic Data' (with a red notification icon), 'Management Reports', and 'Announcements'. The main content area shows a 'User' dropdown set to 'Arie Boeren'. Below this is the 'Announcements' section, which contains a table with the following data:

User	Group	Active?	Date inserted	Ann text
Arie Boeren		Yes	16-02-2012	This an nncement.

At the bottom right of the table, it says '1 - 1'.

## Extra Oefening.

In deze oefening gaan we gebruik maken van een plug-in, die ook een aantal eigen **Dynamic Actions** heeft; een slider. Met deze plug-in gaan we de leeftijd van een 'user' manipuleren.

- Breidt de tabel 'users' uit met een extra kolom 'age' van het type NUMBER.
- Voeg in de Front Office applicatie de kolom toe op de self service pagina.
- Doe hetzelfde op het user scherm in de Back Office applicatie.
- Voeg de slider plug-in toe met het script **Plugin\_slider\_with\_dynamic\_action.sql**.
- Wijzig het **display type** van het item in de Back Office applicatie in 'Slider'.
- Zet de **Display orientation** op **Vertical**. (i.v.m. bug? Waarschijnlijk wordt de plugin niet meer ondersteund voor versie 5.1.)
- Zet het bereik op 0-100. Kijk naar het resultaat. Wat opvalt is dat er een dollar (\$) teken voor staat. Voor een leeftijd is dat natuurlijk niet goed. Probeer de PL/SQL/JavaScript code van de plug-in aan te passen, zodat dit niet meer gebeurt (2 plaatsen). Mooier was het natuurlijk geweest om dit m.b.v. een parameter te kunnen sturen.
- Voeg nu een **Dynamic Action** toe aan de slider. In de events staan nu enkele events van de slider plug-in. We maken gebruik van het **Slide [Slider]** event. De conditie is **greater than 50**.
- We willen de **Style** veranderen indien de leeftijd groter dan 50 is. Daartoe gebruiken we als **Action: Set Style**. Als **Style Name** gebruiken we **color** en als **Value** gebruiken we **red**. Als **False Action** doen we hetzelfde, alleen wordt de kleur **black**.
- We passen de **Dynamic Action** toe op **jQuery Selector** '#P20\_AGE\_display'. Deze waarde is case sensitive.
- Run de pagina en kijk of het werkt.
- Als je met Firebug kijkt naar de slider kan je object met id (#) P20\_AGE\_display ook zien en kijken wat er met **color** gebeurt indien je deze wijzigt.

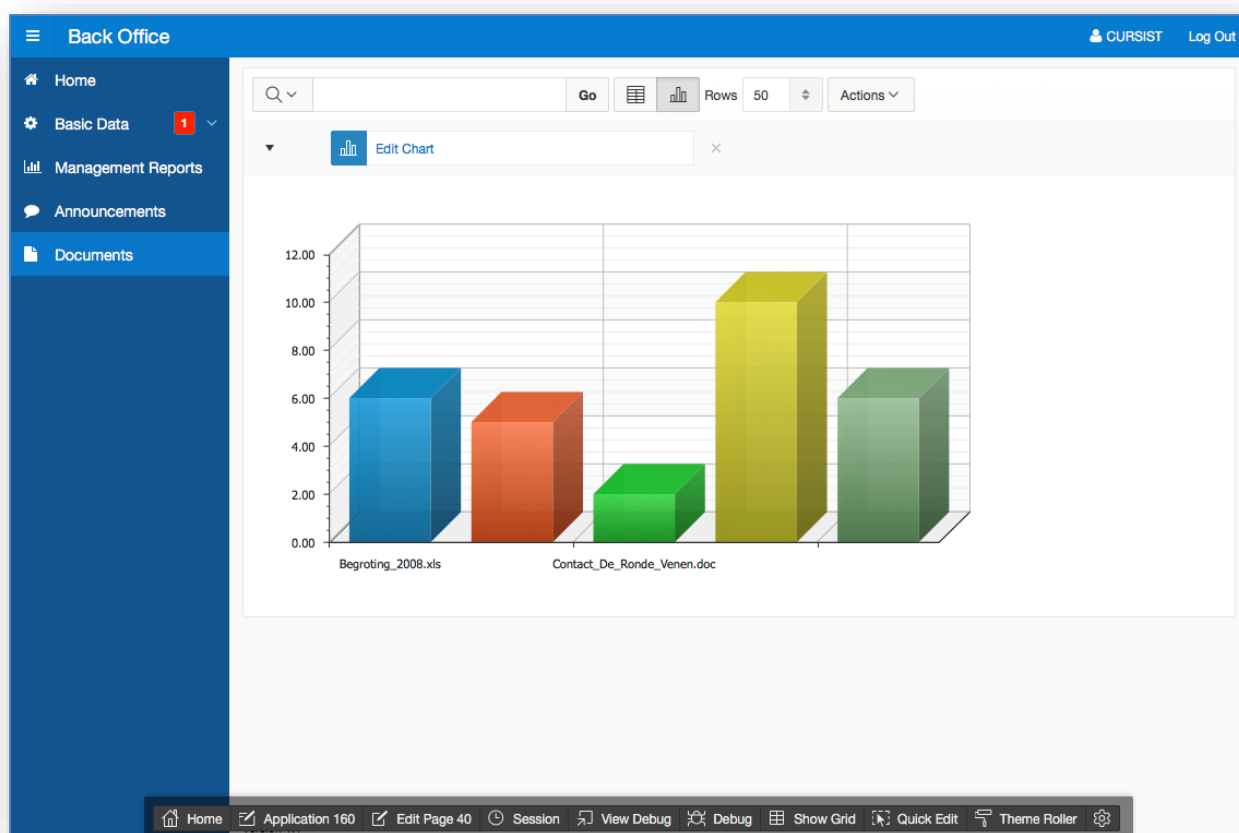
The screenshot shows the 'Back Office' application interface. On the left is a navigation menu with options like Home, Basic Data, Users, Pages, Main Tabs, Main Tabs 2, Groups and Pages, Management Reports, and Announcements. The main content area is divided into two sections. The top section, titled 'Users', displays a table with columns: Edit, Name, Group, City, Address, Username, Email, Active?, and Age. The table contains three rows of user data. The bottom section, titled 'User', shows the edit form for a user named 'Arie Boeren'. The form includes fields for Active?, Group, Firstname, City, Address, Email, Tel Nr, Gender, Username, Date Of Birth, Age, Lastname, and Password. The Age field is highlighted in red and has a slider control below it. The Age value is currently 52. At the bottom of the page, there is a status bar with various icons and text like 'Home', 'Application 260', 'Edit Page 20', 'Session', 'View Debug', 'Debug', 'Show Layout Columns', 'Quick Edit', 'Theme Roller', and a version number '5.1'.

Edit	Name	Group	City	Address	Username	Email	Active?	Age
	Arie Boeren	Finance	ALKMAAR	Koningsstraat 13	arie	arie@company.be	Yes	52
	Sjon de boer	ICT	AMSTERDAM	Keizersgracht 43	sjon	sjon@werken.nl	Yes	45
	qw qw	ICT	qw	qw	qw	qw	No	

## 19. Interactive Reports

Met **Interactive Reports** kan de gebruiker zelf de presentatie en andere zaken bepalen van de gegevens. De programmeur bepaalt de basisset (query) en de mogelijkheden die aan de gebruiker worden gegeven.

- Maak een **Interactive Report** op basis van de 'documents' tabel
- Bekijk de help van het **Interactive Report** en experimenteer met de **Interactive Report** mogelijkheden.
- Maak een chart.
- Bestudeer ook de subscription mogelijkheden.
- Log in als gebruiker. Sla meerdere instellingen op onder een bepaalde naam en bekijk hoe dit werkt, na uitloggen en weer inloggen moeten de bewaarde instellingen bewaard zijn. Deze kunnen (vanaf versie 4.0) ook gedeeld worden met andere gebruikers.
- Als de ontwikkelaar het scherm vanuit APEX opstart kunnen de instellingen als 'default' worden opgeslagen. Eigenlijk vindt er dus een deel van het ontwikkelwerk plaats in de runtime omgeving.



Het ligt in de lijn der verwachtingen dat **Interactive Grids** soortgelijke mogelijkheden gaan krijgen.

## 20. Dictionary Views

Met dictionary views kan je onder andere Quality checks doen.

- Bekijk de output van de volgende queries:

Controleer de alignment van items:

```
SELECT
    application_name,
    page_id,
    item_name,
    label_alignment as align,
    item_element_width as width,
    item_element_max_length as max
FROM
    apex_application_page_items
WHERE
    display_as = 'Text Field'
ORDER BY
    application_name,
    page_id,
    Item_name
```

Welke items hebben geen help:

```
SELECT
    application_id,
    page_id,
    item_name,
    display_as
FROM
    apex_application_page_items
WHERE
    display_as <> 'Hidden'
AND
    item_help_text is null
ORDER BY
    application_name,
    page_id,
    item_name
```

Ook is het mogelijk om de Views te gebruiken in je eigen programma's.

## Extra oefeningen

### Monitoring

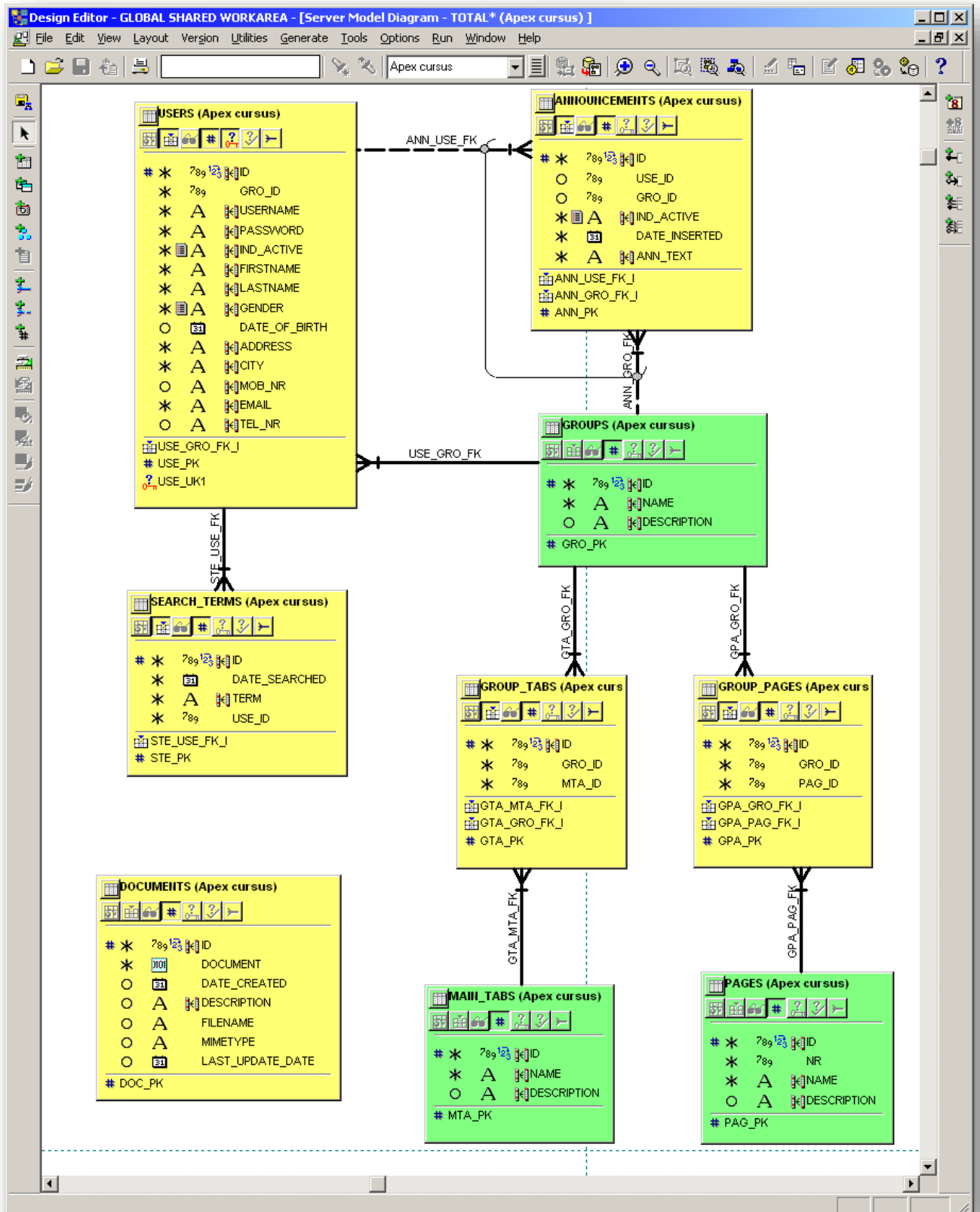
Maak nu in de Back Office een nieuwe hoofdtab 'System' met daaronder een subtab en scherm genaamd 'Response Times', waarmee de prestatie van de applicatie kan worden gemonitord. Maak op deze pagina een rapport en gebruik daarvoor de volgende query:

```
SELECT page_id
,      min(elapsed_time) as min_time
,      avg(elapsed_time) as average_time
,      max(elapsed_time) as max_time
,      count(elapsed_time) as hits
FROM apex_workspace_activity_log
WHERE application_id = 180 /* Front office */
GROUP BY page_id
```

### Document authorization

Breid de database uit met een intersectietabel tussen documenten en groups, zodat daar kan worden opgeslagen welke groep welke documenten kan bekijken. Maak hiervoor nieuwe onderhoudsschermen. Pas de zoek query zodanig aan, dat deze alleen documenten toont, waartoe de gebruikers (groep) toegang heeft.

## Bijlage: Model



```
-- Creating Table 'CG_REF_CODES'
CREATE TABLE CG_REF_CODES
(RV_DOMAIN VARCHAR2(100) NOT NULL
, RV_LOW_VALUE VARCHAR2(240) NOT NULL
, RV_HIGH_VALUE VARCHAR2(240)
, RV_ABBREVIATION VARCHAR2(240)
, RV_MEANING VARCHAR2(240))
```

```

)
/

-- Creating Table 'DOCUMENTS'
CREATE TABLE DOCUMENTS
  (ID NUMBER(10) NOT NULL
  ,DOCUMENT BLOB NOT NULL
  ,DATE_CREATED DATE DEFAULT sysdate
  ,DESCRIPTION VARCHAR2(2000)
  ,FILENAME VARCHAR2(255)
  ,MIMETYPE VARCHAR2(255)
  ,LAST_UPDATE_DATE DATE
  )
/

-- Creating Table 'GROUPS'
CREATE TABLE GROUPS
  (ID NUMBER(10) NOT NULL
  ,NAME VARCHAR2(50) NOT NULL
  ,DESCRIPTION VARCHAR2(2000)
  )
/

-- Creating Table 'USERS'
CREATE TABLE USERS
  (ID NUMBER(10) NOT NULL
  ,GRO_ID NUMBER(10) NOT NULL
  ,USERNAME VARCHAR2(50) NOT NULL
  ,PASSWORD VARCHAR2(50) NOT NULL
  ,IND_ACTIVE VARCHAR2(1) DEFAULT 'Y' NOT NULL
  ,FIRSTNAME VARCHAR2(50) NOT NULL
  ,LASTNAME VARCHAR2(50) NOT NULL
  ,GENDER VARCHAR2(1) NOT NULL
  ,DATE_OF_BIRTH DATE
  ,ADDRESS VARCHAR2(50) NOT NULL
  ,CITY VARCHAR2(50) NOT NULL
  ,MOB_NR VARCHAR2(25)
  ,EMAIL VARCHAR2(50) NOT NULL
  ,TEL_NR VARCHAR2(25)
  )
/

-- Creating Table 'MAIN_TABS'
CREATE TABLE MAIN_TABS
  (ID NUMBER(10) NOT NULL
  ,NAME VARCHAR2(50) NOT NULL
  ,DESCRIPTION VARCHAR2(2000)
  )
/

-- Creating Table 'GROUP_TABS'
CREATE TABLE GROUP_TABS
  (ID NUMBER(10) NOT NULL
  ,GRO_ID NUMBER(10) NOT NULL
  ,MTA_ID NUMBER(10) NOT NULL
  )
/

-- Creating Table 'SEARCH_TERMS'
CREATE TABLE SEARCH_TERMS
  (ID NUMBER(10) NOT NULL
  ,DATE_SEARCHED DATE DEFAULT sysdate NOT NULL
  ,TERM VARCHAR2(100) NOT NULL
  ,USE_ID NUMBER(10) NOT NULL
  )
/

-- Creating Table 'GROUP_PAGES'
CREATE TABLE GROUP_PAGES
  (ID NUMBER(10) NOT NULL
  ,GRO_ID NUMBER(10) NOT NULL
  ,PAG_ID NUMBER(10) NOT NULL
  )
/

-- Creating Table 'PAGES'
CREATE TABLE PAGES
  (ID NUMBER(10) NOT NULL
  ,NR NUMBER NOT NULL
  ,NAME VARCHAR2(50) NOT NULL
  ,DESCRIPTION VARCHAR2(2000)
  )
/

```

```

-- Creating Table 'ANNOUNCEMENTS'
CREATE TABLE ANNOUNCEMENTS
  (ID NUMBER(10) NOT NULL
  ,USE_ID NUMBER(10)
  ,GRO_ID NUMBER(10)
  ,IND_ACTIVE VARCHAR2(1) NOT NULL
  ,DATE_INSERTED DATE DEFAULT sysdate NOT NULL
  ,ANN_TEXT VARCHAR2(2000) NOT NULL
  )
/

-- Creating Index 'X_CG_REF_CODES_1'
CREATE INDEX X_CG_REF_CODES_1 ON CG_REF_CODES
  (RV_DOMAIN
  ,RV_LOW_VALUE)
/

-- Creating Index 'USE_GRO_FK_I'
CREATE INDEX USE_GRO_FK_I ON USERS
  (GRO_ID)
/

-- Creating Index 'GTA_MTA_FK_I'
CREATE INDEX GTA_MTA_FK_I ON GROUP_TABS
  (MTA_ID)
/

-- Creating Index 'GTA_GRO_FK_I'
CREATE INDEX GTA_GRO_FK_I ON GROUP_TABS
  (GRO_ID)
/

-- Creating Index 'STE_USE_FK_I'
CREATE INDEX STE_USE_FK_I ON SEARCH_TERMS
  (USE_ID)
/

-- Creating Index 'GPA_GRO_FK_I'
CREATE INDEX GPA_GRO_FK_I ON GROUP_PAGES
  (GRO_ID)
/

-- Creating Index 'GPA_PAG_FK_I'
CREATE INDEX GPA_PAG_FK_I ON GROUP_PAGES
  (PAG_ID)
/

-- Creating Index 'ANN_USE_FK_I'
CREATE INDEX ANN_USE_FK_I ON ANNOUNCEMENTS
  (USE_ID)
/

-- Creating Index 'ANN_GRO_FK_I'
CREATE INDEX ANN_GRO_FK_I ON ANNOUNCEMENTS
  (GRO_ID)
/

-- Creating Primary Key on 'DOCUMENTS'
ALTER TABLE DOCUMENTS
  ADD (CONSTRAINT DOC_PK PRIMARY KEY
  (ID))
/

-- Creating Primary Key on 'GROUPS'
ALTER TABLE GROUPS
  ADD (CONSTRAINT GRO_PK PRIMARY KEY
  (ID))
/

-- Creating Primary Key on 'USERS'
ALTER TABLE USERS
  ADD (CONSTRAINT USE_PK PRIMARY KEY
  (ID))
/

-- Creating Primary Key on 'MAIN_TABS'
ALTER TABLE MAIN_TABS
  ADD (CONSTRAINT MTA_PK PRIMARY KEY
  (ID))
/

-- Creating Primary Key on 'GROUP_TABS'
ALTER TABLE GROUP_TABS

```

```

ADD (CONSTRAINT GTA_PK PRIMARY KEY
      (ID))
/

-- Creating Primary Key on 'SEARCH_TERMS'
ALTER TABLE SEARCH_TERMS
ADD (CONSTRAINT STE_PK PRIMARY KEY
      (ID))
/

-- Creating Primary Key on 'GROUP_PAGES'
ALTER TABLE GROUP_PAGES
ADD (CONSTRAINT GPA_PK PRIMARY KEY
      (ID))
/

-- Creating Primary Key on 'PAGES'
ALTER TABLE PAGES
ADD (CONSTRAINT PAG_PK PRIMARY KEY
      (ID))
/

-- Creating Primary Key on 'ANNOUNCEMENTS'
ALTER TABLE ANNOUNCEMENTS
ADD (CONSTRAINT ANN_PK PRIMARY KEY
      (ID))
/

-- Creating Unique Key on 'USERS'
ALTER TABLE USERS
ADD (CONSTRAINT USE_UK1 UNIQUE
      (USERNAME))
/

-- Creating Foreign Key on 'USERS'
ALTER TABLE USERS ADD (CONSTRAINT
      USE_GRO_FK FOREIGN KEY
      (GRO_ID) REFERENCES GROUPS
      (ID))
/

-- Creating Foreign Key on 'GROUP_TABS'
ALTER TABLE GROUP_TABS ADD (CONSTRAINT
      GTA_GRO_FK FOREIGN KEY
      (GRO_ID) REFERENCES GROUPS
      (ID))
/

-- Creating Foreign Key on 'GROUP_TABS'
ALTER TABLE GROUP_TABS ADD (CONSTRAINT
      GTA_MTA_FK FOREIGN KEY
      (MTA_ID) REFERENCES MAIN_TABS
      (ID))
/

-- Creating Foreign Key on 'SEARCH_TERMS'
ALTER TABLE SEARCH_TERMS ADD (CONSTRAINT
      STE_USE_FK FOREIGN KEY
      (USE_ID) REFERENCES USERS
      (ID))
/

-- Creating Foreign Key on 'GROUP_PAGES'
ALTER TABLE GROUP_PAGES ADD (CONSTRAINT
      GPA_PAG_FK FOREIGN KEY
      (PAG_ID) REFERENCES PAGES
      (ID))
/

-- Creating Foreign Key on 'GROUP_PAGES'
ALTER TABLE GROUP_PAGES ADD (CONSTRAINT
      GPA_GRO_FK FOREIGN KEY
      (GRO_ID) REFERENCES GROUPS
      (ID))
/

-- Creating Foreign Key on 'ANNOUNCEMENTS'
ALTER TABLE ANNOUNCEMENTS ADD (CONSTRAINT
      ANN_USE_FK FOREIGN KEY
      (USE_ID) REFERENCES USERS
      (ID))
/

```

```
-- Creating Foreign Key on 'ANNOUNCEMENTS'  
ALTER TABLE ANNOUNCEMENTS ADD (CONSTRAINT  
  ANN_GRO_FK FOREIGN KEY  
  (GRO_ID) REFERENCES GROUPS  
  (ID))  
/
```

## Bijlage: generieke oplossing voor multiselect en shuttle

```
create or replace PACKAGE BODY PK_APEX_UTIL IS

-- Cursor Bodies
-- Sub-Program Units
/* set_selectlist */
PROCEDURE SET_SELECTLIST
(P_LIST IN VARCHAR2
,P_INTERSECTION_TABLE IN VARCHAR2
,P_LOOKUP_FK_NAME IN VARCHAR2
,P_MASTER_FK_NAME IN VARCHAR2
,P_MASTER_FK_VALUE IN VARCHAR2
)
IS
-- insert the selectlist value (as a listm e.g. 1:2:4) into the intersection table
-- This process fires After Submit and after the wizard generated process that handles inserts, updates and deletes
-- on the master table.

l_selected HTMLDB_APPLICATION_GLOBAL.VC_ARR2;
l_sql_statement VARCHAR2(1000);

BEGIN

IF p_lookup_fk_name IS NOT NULL AND
p_intersection_table IS NOT NULL AND
p_master_fk_name IS NOT NULL AND
p_master_fk_value IS NOT NULL THEN

-- convert the colon separated string of values into
-- a PL/SQL array
--
l_selected := HTMLDB_UTIL.STRING_TO_TABLE(p_list);
-- clean up the intersection table first
l_sql_statement := 'DELETE FROM ' || p_intersection_table || ' ' ||
'WHERE ' || p_master_fk_name || '=' || p_master_fk_value;

EXECUTE IMMEDIATE l_sql_statement;
--
-- Loop over array to insert lookup_ids and master_id
-- in to intersection table
--
FOR i IN 1..l_selected.count LOOP
l_sql_statement := 'INSERT INTO ' || p_intersection_table || ' ' ||
'(' || p_master_fk_name || ',' || p_lookup_fk_name || ') ' ||
'VALUES (' || p_master_fk_value || ',' || l_selected(i) || ')';
-- no parent key exception
BEGIN
EXECUTE IMMEDIATE l_sql_statement;
EXCEPTION WHEN OTHERS THEN
NULL;
END;

END LOOP;

END IF;

END;
```

```

FUNCTION GET_SELECTLIST
(P_INTERSECTION_TABLE IN VARCHAR2
,P_LOOKUP_FK_NAME IN VARCHAR2
,P_MASTER_FK_NAME IN VARCHAR2
,P_MASTER_FK_VALUE IN VARCHAR2
)
RETURN VARCHAR2
IS
-- get the selectlist value as a list e.g. 1:2:4
-- create the process to fire After Header and after the wizard generated Fetch Row process.

l_selected HTMLDB_APPLICATION_GLOBAL.VC_ARR2;
l_sql_statement VARCHAR2(1000);

BEGIN

IF p_lookup_fk_name IS NOT NULL AND
p_intersection_table IS NOT NULL AND
p_master_fk_name IS NOT NULL AND
p_master_fk_value IS NOT NULL THEN

l_sql_statement := 'SELECT ' || p_lookup_fk_name || ' ' ||
'FROM ' || p_intersection_table || ' ' ||
'WHERE ' || p_master_fk_name || '=' || p_master_fk_value ;

EXECUTE IMMEDIATE l_sql_statement BULK COLLECT INTO l_selected;

END IF;

-- Assign the colon separated list to l_selected
RETURN HTMLDB_UTIL.TABLE_TO_STRING(l_selected);

END;
END PK_APEX_UTIL;

```