

# Ficha 7 – Introdução ao Oracle<sup>TM</sup> Application Express

Bases de Dados, FCT-NOVA

Ano letivo 2017/18

**Grupo 1.** Depois de fazer login no APEX, no canto superior direito pode encontrar uma ligação para a documentação (“Help”). No página que abre, há um caminho (*breadcrumbs*) de Home -> Quick Start -> About the Workspace Home Page. Siga a ligação para “Quick Start” para uma visão geral do APEX. Leia brevemente os vários tópicos usando a ligação “Next”.

Quando chegar à página “Application Builder Concepts”, leia também os dois primeiros tópicos: “Understanding the Difference Between a Worksheet and a Database Application” e “What Is a Page?”

**Grupo 2.** Neste exercício iremos criar uma aplicação em APEX. Uma aplicação em APEX é um conjunto de páginas interligadas que permitem interagir com uma base de dados. Para isso será preciso ter criado a base de dados das matrículas, usada na Ficha 6, e inserido os dados. Se não o fez ainda ou para garantir que esteja tudo completo, execute primeiro o script aula6.sql que acompanha esta ficha (usando primeiro “Upload” em SQL Workshop -> SQL Scripts -> Upload (Script) depois de ter escolhido este ficheiro com “Browse...”, e a seguir clicar no símbolo do semaforo abaixo de “Run” e depois “Run Now”). Pode verificar o resultado de correr o script clicando no símbolo das paginas abaixo de “View Results”.

1. Da página de entrada do APEX, entre na página “Application Builder”. Se seguir a ligação “Help” desta página verá uma descrição sucinta das funcionalidades na página de entrada do Application Builder. A documentação online do APEX é contextual, pelo que pode sempre recorrer ao “Help” em caso de necessidade. Escolha “Create” para criar uma nova aplicação seguindo os passos abaixo e premindo “Next” após cada selecção:

- (a) Application Type: Database
- (b) From Scratch
- (c) Name: Matriculas; Schema: seleccionar o seu schema, se necessário (BDXXXXXX); o resto deixar como está
- (d) Em Pages, adicionar uma página do tipo “Blank” com o nome “Início”. Essa será a página de entrada da aplicação.
- (e) Adicionar uma página do tipo Report, subordinada à página “Início”, que tenha como fonte de dados a tabela CURSOS.
- (f) Após adicionar a página “Cursos”, repare que no cimo da página de adição de páginas tem uma listagem das páginas da sua aplicação. Siga o link para a página “Cursos” para ter acesso às suas propriedades, que pode alterar se quiser.
- (g) Termine a definição da sua aplicação, seguindo “Next” e escolhendo “No tabs” na página das tabs, “No shared components”, autenticação pelo “Application express” e, no “User Interface”, escolha o tema 2. Finalmente confirme a criação da aplicação.
- (h) Teste a sua aplicação em “Run Application”. Note que terá de repetir o login exactamente como teria se fosse um utilizador desta aplicação.

**Grupo 3.** Vamos agora acrescentar à nossa aplicação uma página com a listagem dos alunos matriculados, nomes e respectivos cursos. Para isso será necessário criar uma nova página do

tipo “Report”, “Interactive Report”, no nome e região da página preencher “Matriculas”, activar os *breadcrumbs* (o caminho de páginas seguidas que aparece na parte superior de cada página) e, como página progenitora, escolher a página “Início”. Depois de especificar que não quer tabs e inserir esta query SQL como fonte para os dados desta página:

```
select MATRICULAS.NUMERO as "Número",
       COLOCADOS.NOME as "Nome",
       MATRICULAS.DATAMATR as "Data",
       CURSOS.NOMECURSO as "Curso"
from MATRICULAS inner join COLOCADOS using (IDCANDIDATO, CURSO)
       inner join CURSOS using (CURSO)
```

Passe o “Link to Single Row View” para “No” e, em seguida, “Finish”. Depois de criada a página pode executá-la em “Run Page”.

**Grupo 4.** Finalmente, vamos criar uma ligação da página inicial para este relatório. Na página do “Application Builder”, siga a ligação para a página “Início” e mude a vista para “Component View”. Cada página contém vários elementos dispostos em várias regiões. Na secção “Regions” de “Page Rendering”, siga o link “List” da entrada “Navigation”. Este leva-o para a página que lista as ligações que partem da página inicial. Neste momento, a lista só contém a página dos Cursos. Com o botão “Create List Entry” aceda à página para criar uma entrada nesta lista. Especifique a posição 30, a etiqueta de “Matriculas” e, como alvo, a página de Matriculas que criou no exercício anterior. Depois de criar a entrada na lista das ligações (“Create”) execute a página de início com o botão “Run”.

**Grupo 5.** Vamos agora criar duas páginas, um “form” e um “report”, para aceder à tabela Matriculas. Antes disso, precisamos de criar um trigger que nos ajude a inserir novos alunos obtendo um identificador da sequência correcta (que já fez na Ficha 6). Este é o trigger que devem implementar (em SQL Workshop -> SQL Commands) para que se possa inserir um novo aluno sem ter de especificar um identificador único:

```
create or replace TRIGGER COD_ALUNO
BEFORE INSERT ON matriculas
FOR EACH ROW
DECLARE
    num_aluno number;
BEGIN
    SELECT seq_num_aluno.nextval
    INTO num_aluno
    FROM dual;
    :new.NUMERO := num_aluno;
END;
/
```

- No Application Builder, seleccione a aplicação das matrículas.
- Prima “Create Page”, seleccione “Form”, “Next” e “Form on a table with report”, para criar ambas as páginas.
- Deixe o proprietário default (BDxxxx) e, “Next” e, em “Table / View Name”, escolha a tabela Matriculas.
- Preencha o nome da página e o título da região (e.g. Relatório de Matrículas e Matrículas), “Breadcrumb” e a página inicial como ancestral.

- Depois do “Next”, deixe o valor por omissão (sem tabs), “Next” e, na página do *report*, selecione todas as colunas (com o Ctrl) e, no passo seguinte, aceite a imagem proposta por omissão.
- Para definir a página do *form*, dê de nome, título da região e nome da entrada algo sugestivo como “Criar ou editar matrículas”.
- Escolha o campo NUMERO como chave primária e “Existing Trigger” em “Define the source for the primary key columns”. Selecione todas as colunas disponíveis para mostrar no *form* e os restantes valores como fornecidos por omissão. No último painel basta fazer “Finish” e depois testar a página (tenta, por exemplo, inscrever o candidato 123928 no curso 9367 e o candidato 140097 no curso 9367 no ano 2016 numa data à escolha).

**Grupo 6.** Altere a página do relatório das matrículas para mostrar o nome do aluno e do curso em vez dos identificadores. Na página do relatório que criou no exercício anterior, escolha a região das Matrículas e substitua a “Region Source” por esta query:

```
select m.NUMERO "NUMERO",
c.NOME "Nome",
r.NOMECURSO "Curso",
m.DATAMATR "Data de Matrícula",
m.ANO "Ano"
from "#OWNER#". "MATRICULAS" m, "#OWNER#". "COLOCADOS" c, "#OWNER#". "CURSOS" r
WHERE m.IDCANDIDATO = c.IDCANDIDATO and c.CURSO = r.CURSO and m.ANO = c.ANO
```

Prima “Apply Changes” e corra a página. Note que pode ter de escolher as colunas a mostrar para ver as alterações à página, quando esta estiver em execução. Os relatórios do APEX possibilitam um elevado nível de configuração aos utilizadores (independente do programador). Além de escolher as colunas a apresentar pode aplicar filtros, realçar colunas, ordenar e até realizar alguns cálculos e obter gráficos (para mais informações consulte a ajuda no botão “Action”). No fim o utilizador pode guardar a vista que obtém. Igualmente, enquanto testa a sua aplicação o programador deve confirmar e guardar a vista inicial que considera mais adequada.

**Grupo 7.** Para mostrar, em cada momento, a quantos créditos ECTS é que o aluno já se inscreveu, vamos criar a seguinte função (no “SQL Commands”):

```
create or replace FUNCTION calc_ects(aluno NUMBER) RETURN NUMBER
IS
    totaLECTS NUMBER;
BEGIN
    SELECT sum(ECTS)
    INTO totaLECTS
    FROM inscricoes i, cadeiras c
    WHERE aluno=i.numero AND i.cadeira = c.cadeira;
    RETURN totaLECTS;
END calc_ects;
```

Depois, na definição da página do relatório de matrículas, vamos seleccionar novamente a região Matrículas e alterar novamente a fonte dessa tabela:

```
select m.NUMERO "NUMERO",
c.NOME "Nome",
r.NOMECURSO "Curso",
m.DATAMATR "Data de Matrícula",
m.ANO "Ano",
calc_ects(m.NUMERO) "ECTSs"
from "#OWNER#". "MATRICULAS" m, "#OWNER#". "COLOCADOS" c, "#OWNER#". "CURSOS" r
WHERE m.IDCANDIDATO = c.IDCANDIDATO and c.CURSO = r.CURSO and m.ANO = c.ANO
```

Quando executar a página, selecione a nova coluna dos ECTSs para que seja visível.

**Grupo 8.** Para concluir, vamos criar uma lista de valores (*List of Values, LOV*) com os cursos em substituição do código de curso. Na página de definição do form de editar e criar matriculas, prima o ícone “Create” na área “List of Values” de “Shared Components”. Escolha “From Scratch”, nome “Cursos”, tipo “Dynamic” e na query preencha:

```
select nomecurso, curso
from cursos
order by nomecurso
```

Note que a query de uma LoV deve seleccionar sempre primeiro os valores a mostrar e, em segundo lugar, os valores a devolver. Neste caso, queremos mostrar os cursos e devolver os códigos correspondentes.

Depois de criar a LoV, na área de “Page Rendering”, em “Items”, siga o link para o código de curso e, em “Display As” escolher “Select List”. Depois, abaixo, em “List of Values”, seleccionar a LoV criada (“Cursos”). Clique “Apply Changes” e corra a página para testar.