

Modelagem do banco de dados e da aplicação

Site: [TCU Moodle Site](#)

Curso: Curso Básico de Apex

Livro: Modelagem do banco de dados e da aplicação

Impresso por: RUAN HELENO CORREA DA SILVA

Data: quarta, 30 Mar 2022, 02:32

Índice

1. SQL Workshop

2. Introdução ao banco de dados Oracle

2.1. Principais tipos de dados

2.2. Principais objetos

3. Identificação do problema

3.1. Padrão de nomenclatura de objetos de banco

3.2. Normalização

3.3. Modelo de dados

4. Criando as tabelas do sistema

4.1. Criando a tabela Aluno

4.2. Criando as demais tabelas

5. Verificando os objetos criados

6. Fazendo a carga inicial de dados

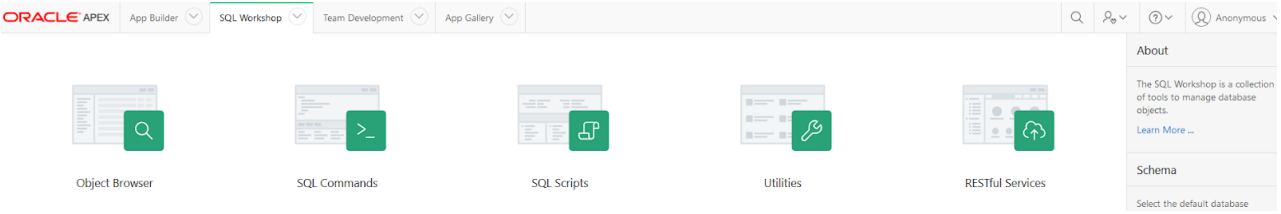
6.1. Utilizando os Scripts SQL

6.2. Carregando dados pelo Data Workshop

7. Resumo

1. SQL Workshop

O SQL Workshop é um componente do APEX que permite a visualização e o gerenciamento de objetos de banco de dados, tais como tabelas, views, funções, procedures, triggers, packages, etc.



No SQL Workshop é possível acessar:

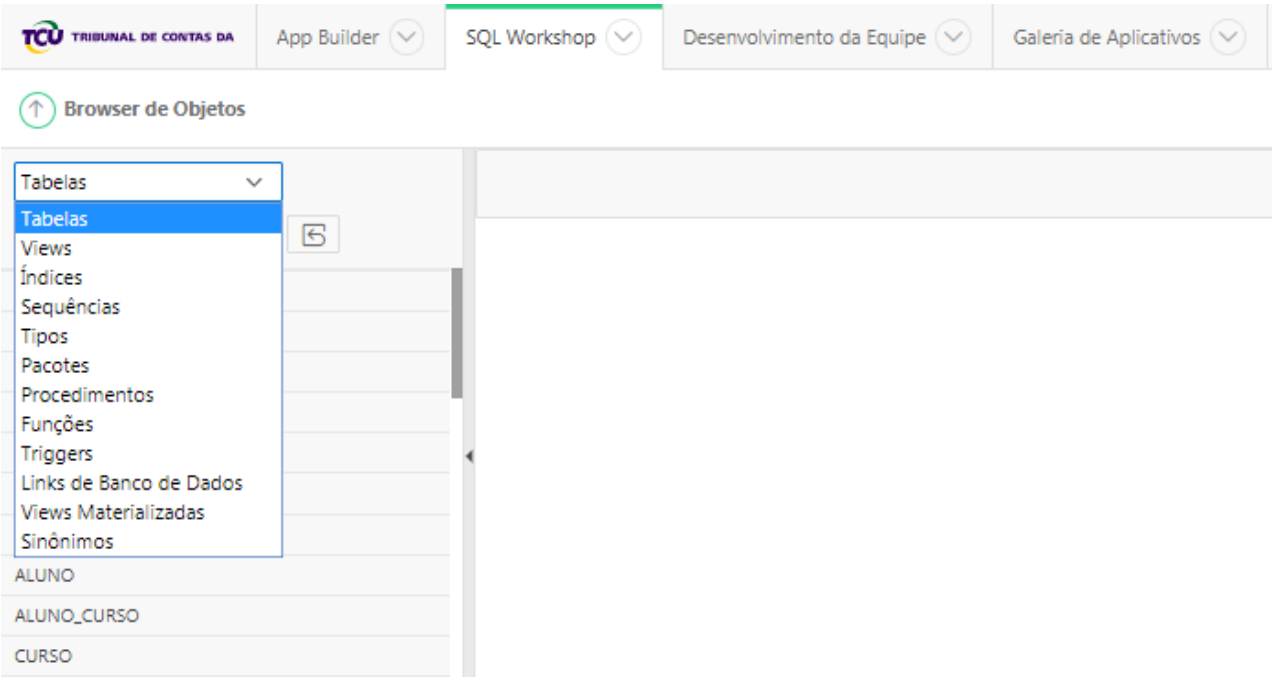
- Browser de Objetos;
- Comandos SQL;
- Scripts SQL;
- Utilitários; e,
- Serviços RESTful.

a. Browser de Objetos

Esta parte do SQL Workshop é responsável pela criação, alteração, visualização e exclusão de objetos no seu esquema de banco de dados.

Os objetos disponíveis são: Tabelas, Views, Índices, Sequências, Tipos, Pacotes, Procedimentos, Funções, Triggers, Links de Banco de Dados, Views Materializadas e Sinônimos.

Veja a árvore de seleção de objetos. Ao se selecionar uma opção na árvore, são listados os objetos daquele tipo.



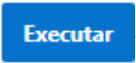
No SQL Workshop não se manipula o aplicativo, mas apenas os objetos de banco de dados.

Por meio do botão que fica no canto superior direito , você terá acesso ao assistente de criação de todos esses objetos de banco.

b. Comandos SQL

Você já estudou SQL (*Structured Query Language*)? Já ouviu falar em PL/SQL (*Procedural Language/Structured Query Language*)? Então, agora chegou a hora de conhecermos um pouco sobre essas tecnologias!

A área "Comandos SQL" é o melhor lugar dentro do APEX para testar, validar e otimizar códigos SQL e PL/SQL (linguagem de programação do banco Oracle). Você poderá implementar comandos para manipular os dados do banco.

Nesta parte da ferramenta você poderá escrever um código SQL ou PL/SQL e executá-lo diretamente sobre o banco. Para isso, basta digitar o código e clicar no botão Executar .

DICA: Outra forma de executar é selecionar o trecho de código desejado e pressionar simultaneamente as teclas Ctrl e Enter: <Ctrl> + <Enter>.

O resultado da execução aparecerá na aba da parte inferior da janela. Além do resultado, é gerado o plano de execução (caminho percorrido pelo Oracle para acessar os dados) que pode ser acessado pela aba Explicação e automaticamente o comando fica registrado na aba Histórico.

Existe a possibilidade de você querer executar um ou mais comandos e salvá-los, para utilização posterior. O botão “**Salvar**” permite isso e a recuperação destes comandos é feita através da aba “**Instrução SQL Salva**”. Salve com nomes significativos para você lembrar depois.

Resultados

Explicação

Descrever

Instrução SQL Salva

Histórico

Proprietário ?

Localizar ?

Linhas

- Todos os Usuários -

10

Ir

Excluir Marcado(s)

<input type="checkbox"/>	Proprietário	Nome	Descrição	SQL	Atualizado por	Atualizado
<input type="checkbox"/>	ALUNO	Consulta Aluno	-	SELECT * FROM aluno;	ALUNO	Agora
<input type="checkbox"/>	ALUNO	Consulta Professor	-	SELECT * FROM professor;	ALUNO	9 segundos atrás
<input type="checkbox"/>	ALUNO	Consulta Turma	-	SELECT * FROM turma;	ALUNO	32 segundos atrás

Para ver o tipo de objeto ou a descrição de uma tabela utilize o comando DESC ou DESCRIBE <nome_do_objeto>.

Agora é com você!

- Vá ao Browser de Objetos e verifique se existe uma tabela com o nome *emp*. Caso não exista, acesse “SQL Workshop” > “Utilitários” > “Conjunto de Dados de Amostra” e instale os objetos “EMP / DEPT” em inglês. Verifique novamente se a tabela existe no "Browser de Objetos".
- Abra o “Comandos SQL” e digite: SELECT * FROM emp;
- Selecione a instrução digitada no passo anterior e pressione <crtl> + <enter> ou pressione o botão “**Executar**”. O resultado deve ser semelhante ao da imagem abaixo.

Comandos SQL

EsquemaAPEX_ANDERSON

☒ Commit Automático

Linhas10

Limpar Comando

Localizar Tabelas

Salvar

Executar

SELECT * FROM emp;

Resultados

Explicação

Descrever

Instrução SQL Salva

Histórico

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17/11/1981	5000	-	10
7698	BLAKE	MANAGER	7839	01/05/1981	2850	-	30
7782	CLARK	MANAGER	7839	09/06/1981	2450	-	10
7566	JONES	MANAGER	7839	02/04/1981	2975	-	20
7788	SCOTT	ANALYST	7566	09/12/1982	3000	-	20
7902	FORD	ANALYST	7566	03/12/1981	3000	-	20
7369	SMITH	CLERK	7902	17/12/1980	800	-	20
7499	ALLEN	SALESMAN	7698	20/02/1981	1600	300	30
7521	WARD	SALESMAN	7698	22/02/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	28/09/1981	1250	1400	30

Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.

10 linhas retornadas em 0,00 segundos

Fazer Download

andersonf apex_anderson pt-br

Copyright © 1999, 2018, Oracle. Todos os direitos reservados.

Application Express 18.2.0.00.12

- Vá na aba “Explicação” e veja o plano de execução desse comando SQL.
- Vá na aba “Histórico” e veja os comandos executados.
- Salve esse SQL e o recupere na aba “Instrução SQL Salva”. Depois apague o comando salvo.

7. Agora digite o código abaixo para saber mais sobre a tabela EMP: DESCRIBE emp;

8. Digite e execute outro SQL: SELECT * FROM emp WHERE deptno = :COD_DEPARTAMENTO;

Teste o SQL com os valores 10, 20 e 30 para o parâmetro :COD_DEPARTAMENTO.

c. Scripts SQL

Esta parte do APEX permite que você crie (botão **“Criar>”**), apague, importe (botão **“Fazer Upload>”**), visualize e exporte scripts SQL (conjunto de comandos SQL agrupados).

Na prática, essa área é muito útil quando é necessário executar uma sequência de comandos SQL de uma só vez, normalmente incluindo, alterando e/ou excluindo estruturas de tabelas.

Scripts SQL

Q

Ir

Ações

Excluir Marcado(s)

Fazer Upload

SQL Rápida

Criar

	Editar	Proprietário	Nome	Criado	Atualizado por	Atualizado	Bytes	Resultados	Executar
		ANDERSONRF	chamada.sql	2 semanas atrás	ANDERSONRF	2 semanas atrás	810	1	

1 - 1

Tarefas

Gerenciar Resultados

Mostrar Cotas

Exportar

Importar

andersonrf

apex_anderson

pt-br

Copyright © 1999, 2018, Oracle. Todos os direitos reservados.

Application Express 18.2.0.00.12

d. Utilitários

Nesta parte do APEX podemos acessar alguns Utilitários que permitem executar operações relacionadas ao banco de dados.

Os itens de menu mais usados são Carga/Descarga de Dados (dentro de Data Workshop), Gerar DDL e Relatórios de Objetos.

TRIBUNAL DE CONTAS DA

App Builder

SQL Workshop

Desenvolvimento da Equipe

Galeria de Aplicativos

Q

Utilitários

Utilitários

Data Workshop

Carregar e descarregar dados usando arquivos de texto, XML ou uma planilha.

SQL Rápida

Generate SQL using shorthand syntax

Gerar DDL

Gerar scripts para todos os objetos do banco de dados ou para os selecionados em um esquema.

Comparação de Esquemas

Mostrar diferenças entre os objetos do banco de dados em dois esquemas diferentes.

Lixeira

Restaurar objetos eliminados do banco de dados.

Sobre o BD

Revisar detalhes do banco de dados. Observação: uma conta do banco de dados que tem a atribuição de DBA é obrigatória.

Construtor de Consultas

Criar consultas graficamente, adicionando tabelas em um painel e selecionando colunas para retorno.

Conjuntos de Dados de Amostra

Install, refresh, or remove sample datasets.

Padrões da Interface do Usuário

Especificar propriedades de layout para gerar itens/colunas consistentemente em páginas e aplicativos.

Métodos em Tabelas

Gerar scripts de API para operações DML nas tabelas especificadas.

Relatórios de Objetos

Acessar vários relatórios em tabelas, exceções, segurança, objetos e código PL/SQL.

Monitor de Banco de Dados

Executar relatórios de atividade do banco de dados. Observação: uma conta do banco de dados que tem a atribuição de DBA é obrigatória.

Criar Objeto

Tabela

Exibir

Índice

Sequência

Tipo

Pacote

Procedimento

Função

Trigger

Link de Banco de Dados

View Materializada

Sinônimo

andersonrf

apex_anderson

pt-br

Copyright © 1999, 2018, Oracle. Todos os direitos reservados.

Application Express 18.2.0.00.12

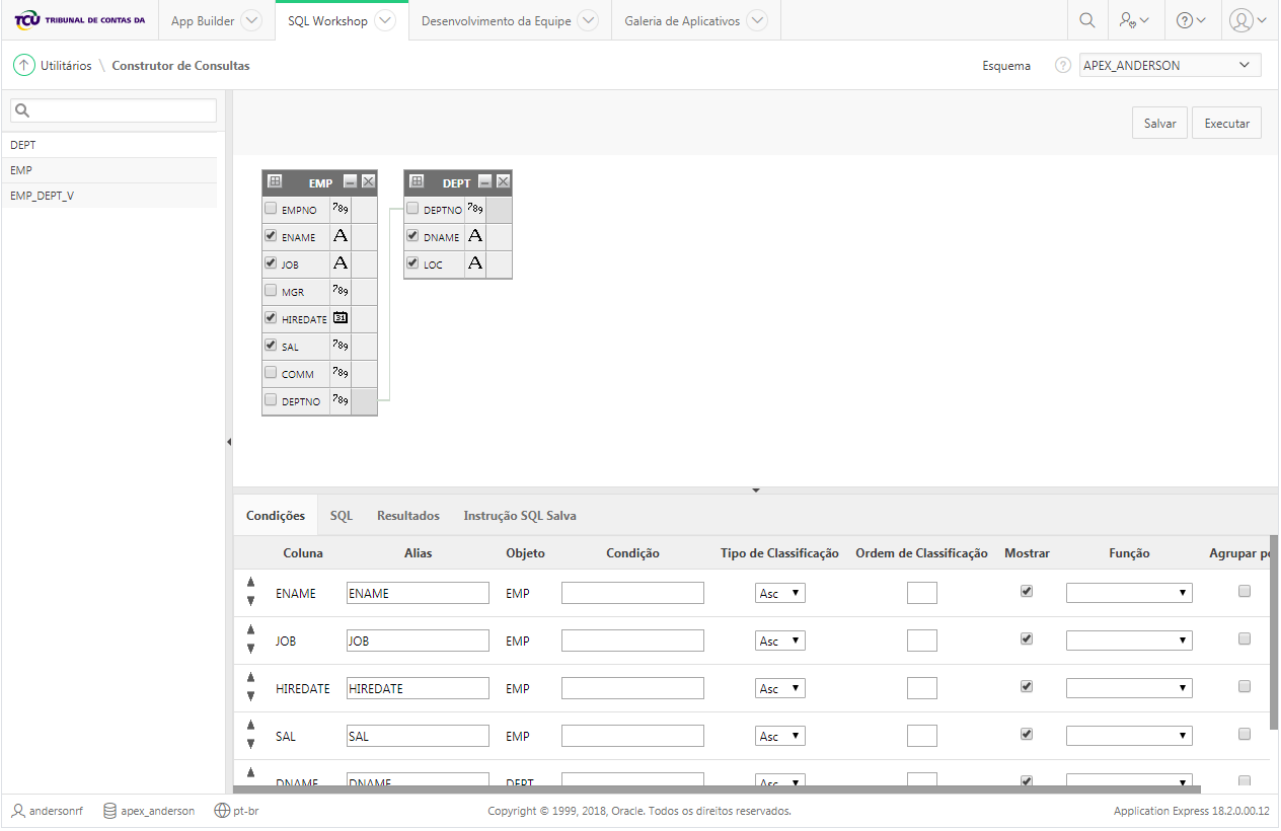
Um utilitário que só é útil para quem não domina SQL é o **“Construtor de Consultas”** (ou “Query Builder”). Ele permite que você monte consultas SQL visualmente. Para isso, dê um clique sobre quaisquer das tabelas (painel à esquerda). Neste momento a tabela será exibida na parte superior da tela. Caso você precise de mais de uma tabela e entre elas tenha que fazer uma junção (join), clique nas colunas, das 2 (duas) tabelas a serem ligadas. Para escolher os campos que aparecerão na

consulta. Marque as caixas de seleção (checkboxes) ao lado do campo desejado.

Veja abaixo um exemplo de uso do "**Construtor de Consultas**" em que foram utilizadas as tabelas EMP e DEPT que foram ligadas pela coluna DEPTNO e serão exibidas as colunas marcadas (com check) ENAME, JOB, HIREDATE, SAL, DNAME e LOC.

Para verificar se os resultados estão corretos, clique no botão "**Executar**" para rodar a consulta SQL e visualizar os resultados.

Caso queira salvar o comando construído, clique no botão "**Salvar**". Ao fazer isso o comando também ficará disponível nas "Instruções SQL Salvas" do "**Comandos SQL**".



Agora é com você!

- No "Construtor de Consultas", crie visualmente uma consulta SQL que retorna o nome do empregado, bem como o nome do departamento e do chefe do empregado. Para isso utilize as tabelas EMP e DEPT.
1. Acesse a tela do Construtor de Consultas (Query Builder).
 2. Encontre e clique na tabela DEPT na lista a esquerda.
 3. Clique na tabela EMP.
 4. Clique novamente na tabela EMP na lista. (Uma tabela EMP será para o empregado e outra para o chefe, que também é um empregado!)
 4. Clique no quadrado após 789 ao final do campo MGR da tabela EMP e faça o join clicando no quadrado após 789 ao final do campo EMPNO da tabela EMP_1.
 5. Na tabela EMP, clique no quadrado ao final do campo DEPTNO e clique no quadrado ao final do campo DEPTNO na tabela DEPT para fazer o relacionamento entre essas colunas.
 6. Marque os campos DNAME na tabela DEPT, ENAME na tabela EMP e ENAME na tabela EMP_1.
 7. Na aba "Condições" na parte de baixo, altere o campo Alias para "Departamento", "Empregado" e "Chefe", conforme mostrado abaixo.

Utilitários \ Construtor de Consultas

Esquema ⓘ APEX_ANDERSON ▾

SalvarExecutar

DEPT

EMP

EMP_DEPT_V

DEPT

EMP

EMP_1

Condições

SQL

Resultados

Instrução SQL Salva

Coluna	Alias	Objeto	Condição	Tipo de Classificação	Ordem de Classificação	Mostrar	Função	Agrupar por
▲ ▼ DNAME	Departamento	DEPT		Asc ▼		<input checked="" type="checkbox"/>	▼	<input type="checkbox"/>
▲ ▼ ENAME	Empregado	EMP		Asc ▼		<input checked="" type="checkbox"/>	▼	<input type="checkbox"/>
▲ ▼ ENAME	Chefe	EMP_1		Asc ▼		<input checked="" type="checkbox"/>	▼	<input type="checkbox"/>

andersonrf

apex_anderson

pt-br

Copyright © 1999, 2018, Oracle. Todos os direitos reservados.

Application Express 18.2.0.00.12

8. Clique na aba “SQL” e veja a consulta que foi gerada.
9. Clique no botão “**Executar**”
10. Veja abaixo na aba “Resultados” o que foi gerado pela consulta.
- Pronto! Você pode ainda clicar no botão “**Salvar**”, se desejar gravar o SQL para uso posterior.

2. Introdução ao banco de dados Oracle

Este módulo do sistema dedica-se a ambientar as pessoas que não estão familiarizadas com o banco de dados Oracle.

Para se desenvolver no APEX é importante conhecer bem o Oracle.

Veremos os principais tipos de dados e objetos de banco que são mais utilizados no APEX.

2.1. Principais tipos de dados

É importante que você leia as seções abaixo e conheça os principais tipos de dados no Oracle.

1) CHAR (*tam*)

Tipo alfanumérico que recebe até 2000 caracteres, onde *tam* representa o número de caracteres. Caso o valor informado para o campo seja menor do que o tamanho máximo do mesmo, o Oracle irá preencher os espaços restantes com espaços em branco. Indicado para quando temos uma quantidade de caracteres fixa e de preenchimento obrigatório.

Exemplo de uso: SIGLA_UF CHAR(2);

O exemplo indica que o nome do campo é SIGLA_UF e o tipo é CHAR com 2 caracteres. Mesmo que a gente atribua apenas um caractere ao campo, o espaço ocupado para armazenamento será o de 2 caracteres no banco de dados.

Campos que normalmente usam esse tipo de dados: UF (DF, SP, RJ, GO etc), Sexo (M/F), flag de Sim ou Não (S/N) tipo de endereço (R/C) (residencial / comercial)

2) VARCHAR2(*tam*)

Tipo alfanumérico que recebe até 4000* caracteres, onde *tam* representa o número de caracteres. Caso o valor informado para o campo seja menor do que o tamanho máximo do mesmo, o Oracle ocupará apenas o espaço necessário para o armazenamento da string, economizando assim o espaço não utilizado. Indicado para campos alfanuméricos em geral.

Exemplo de uso: EMAIL VARCHAR2(200);

O exemplo indica que o nome do campo é EMAIL e o tipo é VARCHAR2 com o máximo de até 200 caracteres. O banco de dados armazenará apenas a quantidade de caracteres atribuída.

Campos que normalmente usam esse tipo de dados: Nome, email, login etc.

** Este limite máximo pode ser estendido pelo DBA para até 32k caracteres a partir da versão 12c do banco de dados Oracle*

Obs: Utilize sempre que necessário o tipo VARCHAR2! NUNCA utilize o tipo VARCHAR, pois está em desuso.

3) NUMBER (*i,d*)

Tipo numérico que recebe de 1×10^{-130} a $9.99...9 \times 10^{125}$ com até 38 dígitos significativos. Ao se passar os parâmetros *i* (quantidade de números armazenados) e *d* (quantidade de casas decimais).

Ex: Caso não seja passado valores para *d*, só serão armazenados números inteiros e se nem *i* nem *d* receberem valores o Oracle aloca o tamanho máximo para o campo. Indicado para dados numéricos em geral.

Exemplos de uso:

- SALARIO NUMBER(10,2) – Significa que dos 10 números alocados, dois irão representar as casas decimais, possibilitando armazenar 8 números inteiros e dois decimais.
- QUANTIDADE NUMBER(10) – Significa que serão alocados dez números inteiros.
- COD NUMBER – Significa que será alocada a capacidade máxima de um tipo numérico.

Campos que normalmente usam esse tipo de dados: Quantidade, COD (chave primária numérica) etc.

Obs: Procure utilizar o tipo NUMBER sem definir a precisão para não limitar a informação que o campo pode receber.

4) DATE

Armazena uma data contendo ano, mês, dia, hora, minuto, segundo.

Muito cuidado com quando fizer comparações de igual com campos do tipo DATE, pois o Oracle SEMPRE armazena também hora, minuto e segundo nestes campos.

Exemplo de uso: DATA_NASCIMENTO DATE;

Campos que normalmente usam esse tipo de dados: Data de nascimento, data e hora de alteração de um dado em tabela de auditoria etc.

Para visualizar a hora, minuto e segundo, deve-se utilizar a função TO_CHAR.

Para "zerar" a hora, minuto e segundo, utilize a função TRUNC.

Para armazenar datas, procure utilizar o tipo DATE!

5) CLOB

Armazena um texto (conjunto de caracteres) de até (4 GB - 1) * DB_BLOCK_SIZE.

Para obter o tamanho do DB_BLOCK_SIZE, faça a consulta: SELECT tablespace_name, block_size FROM user_tablespaces;

Utilize este campo com prudência uma vez que o tamanho do dado lá inserido pode ser muito grande causando um desperdício de espaço no banco de dados.

Além disso, o uso deste tipo de dados é mais complexo necessitando de packages especializadas.

Exemplo de uso: CURRICULO CLOB;

Campos que normalmente usam esse tipo de dados: conteúdo de atas ou documentos presentes em um formulário. Outputs que geram um HTML.

6) BLOB

Armazena um binário de até (4 GB - 1) * DB_BLOCK_SIZE.

Para obter o tamanho do DB_BLOCK_SIZE, faça a consulta: SELECT tablespace_name, block_size FROM user_tablespaces;

Evite utilizar este campo uma vez que o tamanho do dado lá inserido pode ser muito grande causando um desperdício de espaço no banco de dados.

Existem outras formas bem mais baratas de lidar com arquivos desse tipo, como links para eles no Portal, no servidor de arquivos, no GED, etc.

Além disso, o uso deste tipo de dados é mais complexo necessitando de packages especializadas.

Exemplo de uso: FOTO BLOB;

Campos que normalmente usam esse tipo de dados: fotos, vídeos, arquivos pdf doc xls etc.

Para mais informações sobre os tipos de dados, acesse a [documentação da Oracle](#).

2.2. Principais objetos

No Oracle, ao criarmos um usuário no banco (com permissão para criar objetos), automaticamente é criado um esquema (schema).

Este esquema é uma estrutura logicamente construída para organizar todos os objetos criados no banco de dados pelo usuário.

A seguir veremos os principais tipos de objetos.

Para mais informações sobre esquemas acesse o [link](#).

a. Tabelas

Tabela é a estrutura básica para armazenamento de dados dentro do banco de dados Oracle. Elas são formadas por linhas e colunas, e para criá-las, você deve ter o perfil de desenvolvedor no espaço de trabalho do APEX.

A criação das tabelas e outros objetos do banco de dados, no espaço de trabalho APEX, é feito em: SQL Workshop > Browser de Objetos > clicando em + ▾, no canto superior direito da tela.

Quando da criação da tabela, o usuário deverá informar: nome da tabela, nome das colunas, tipo de dados, se o campo é de preenchimento obrigatório ou não.

Importante: O nome da tabela deve ter até 30 caracteres, ser único no esquema, não conter espaços em branco ou caracteres especiais, com exceção do underscore "_".

Veja na imagem abaixo, um exemplo da tabela Aluno.

ALUNO

+

▼

Tabela

Dados

Índices

Modelo

Constraints

Concessões

Estatísticas

Padrões da UI

Triggers

Dependências

SQL

Adicionar Coluna

Modificar Coluna

Renomear Coluna

Eliminar Coluna

Renomear

Copiar

Eliminar

Truncar

Criar Tabela de Consulta

Nome da Coluna	Tipo de Dados	Anulável	Padrão	Chave Primária
COD	NUMBER	Não	-	1
NOME	VARCHAR2(50)	Não	-	-
DATA_NASCIMENTO	DATE	Sim	-	-
IND_SEXO	CHAR(1)	Não	-	-
ENDERECO_COMPLETO	VARCHAR2(200)	Sim	-	-
TELEFONE	VARCHAR2(20)	Sim	-	-
EMAIL	VARCHAR2(20)	Sim	-	-

Fazer Download

|

Imprimir

Para mais informações sobre tabelas acesse o [link](#).

b. Regras de validação (Constraint)

As *constraints* são regras de validações criadas dentro do banco de dados para garantir a integridade de dados, independentemente da aplicação.

Podem ser criadas juntamente com a tabela ou adicionadas posteriormente.

Elas se dividem em quatro tipos:

- **Primary Key (chave primária)** – Quando o campo recebe essa constraint, fica garantido que esse campo possuirá valores únicos e não nulos.
- Não confunda o nome da constraint com o nome da chave primária;
-
- **Foreign Key (chave estrangeira)** – Quando o campo recebe essa constraint, fica garantido que se o campo for preenchido com algum valor, o mesmo deverá existir na tabela mestre (Master). Quando você cria no APEX uma constraint de FK, o padrão de nomenclatura é <<nome da tabela >> + _FK.
- Quando você cria várias Foreign Keys na mesma tabela, renomeie o constraint, pois não é permitido nomes repetidos.
-
- **Check** – Essa constraint é uma das mais versáteis, pois pode trabalhar com validações simples, como "campo não pode ser nulo", até restrições mais complexas, como "o valor do campo deve estar dentro de um intervalo determinado", entre outras.
-

- **Unique** – Quando um campo recebe essa constraint, fica garantido que o valor dado para o campo é único e não se repetirá. Por exemplo, um campo de CPF em um cadastro de pessoas.

Observação: O nome atribuído à constraint serve apenas como um identificador. Não gaste muito tempo definindo este nome, procure seguir o padrão sugerido pelo APEX sempre que possível.

Veja na tela abaixo, as regras de validação (constraints) referentes à tabela Professor da aplicação que será implementada durante o curso.

Browser de Objetos

Tabelas

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

Esquema

APEX_CURSO

PROFESSOR

Tabela

Dados

Índices

Modelo

Constraints

Concessões

Estatísticas

Padrões da UI

Triggers

Dependências

SQL

Criar

Eliminar

Ativar

Desativar

Constraint	Tipo	Condição de Pesquisa	Constraint Relacionada	Colunas	Regra de Exclusão	Status	Última Alteração	Índice	Inválido
PROFESSOR_PK	Primária	-	-	COD	-	ENABLED	07/08/20 19:02:58	PROFESSOR_PK	-
SYS_C001876177	Verificação	"COD" IS NOT NULL	-	-	-	ENABLED	07/08/20 19:02:58	-	-
SYS_C001876178	Verificação	"NOME" IS NOT NULL	-	-	-	ENABLED	07/08/20 19:02:58	-	-
SYS_C001876179	Verificação	"IND_NIVEL" IS NOT NULL	-	-	-	ENABLED	07/08/20 19:02:58	-	-
SYS_C001876180	Verificação	"EMAIL" IS NOT NULL	-	-	-	ENABLED	07/08/20 19:02:58	-	-
PROFESSOR_CK_IND_NIVEL	Verificação	IND_NIVEL in ('F','M','S','P')	-	-	-	ENABLED	07/08/20 19:03:39	-	-

A constraint PROFESSOR_PK define o campo COD como chave primária.

As constraints iniciadas com SYS_ são as que foram geradas automaticamente pelo Oracle quando marcamos a opção Não Nulo (no APEX) que verificam se o conteúdo do campo ao qual elas se referem está preenchido.

A constraint PROFESSOR_CK_NIVEL verifica se o campo NIVEL contém apenas algum dos seguintes valores (10 - graduado, 20 - especialista, 30 - certificado, 40 - mestre, 50 - doutor).

Quando formos criar no APEX a estrutura do banco, você poderá colocar em prática todos os conceitos desse tópico, inclusive os de constraints.

Para mais informações sobre regras de validação acesse o [link](#).

c. Uso de sequência (sequence) e triggers

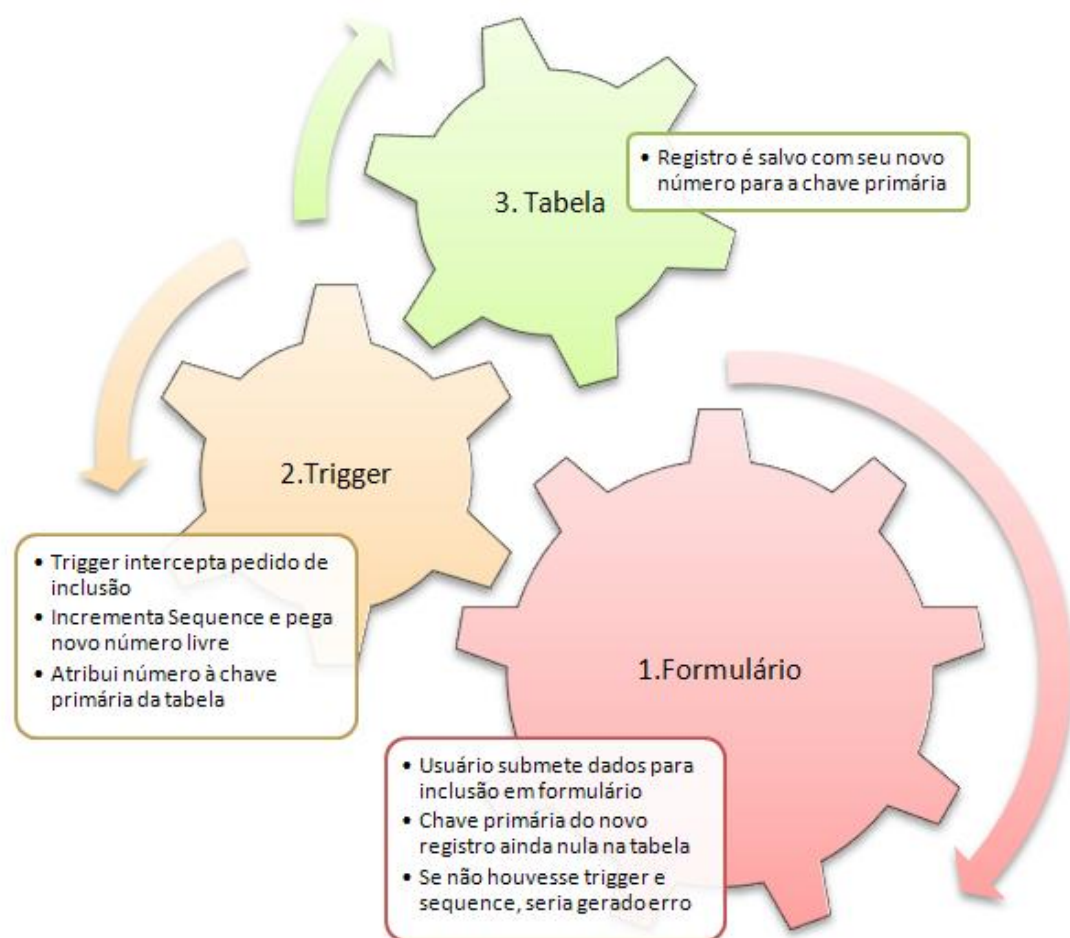
Sequências são objetos do banco de dados Oracle, especializados em armazenar e manter uma sequência de números obedecendo a uma regra previamente definida.

Na prática, uma sequência é muito utilizada para que o banco de dados saiba qual o próximo número de chave primária que será atribuído quando da criação de novos registros nas tabelas.

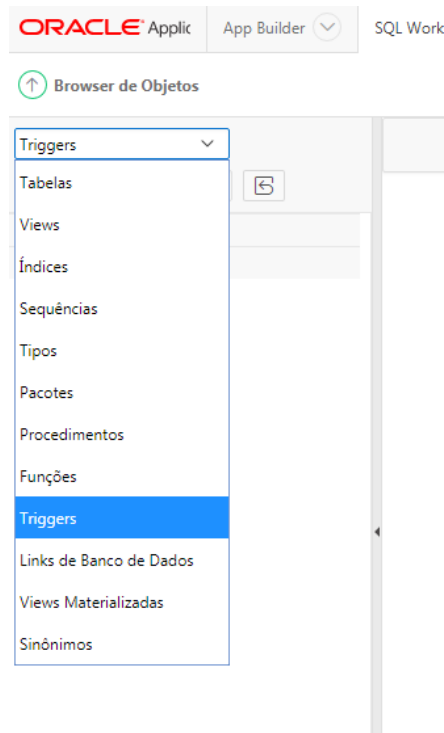
Enquanto isso, uma trigger é um objeto de banco cuja função é disparar alguma ação quando ocorre uma inclusão, uma remoção ou uma atualização de registro.

Uma trigger pode atingir um ou mais registros dessa tabela ou outras.

O diagrama abaixo, dá uma ideia do mecanismo transacional que pode ser criado por meio de sequências e triggers, quando um usuário submete dados para serem incluídos em um formulário.



No APEX, você pode verificar os objetos de sequências e triggers no banco de dados, simplesmente, consultando no "Browser de Objetos", selecionando um desses objetos na caixa de seleção à esquerda da tela.



No APEX, ao clicarmos na opção para criação de uma tabela, seguiremos uma série de passos, dentre os quais encontraremos a opção para dizermos se queremos ou não que seja criada uma sequência e/ou uma trigger.

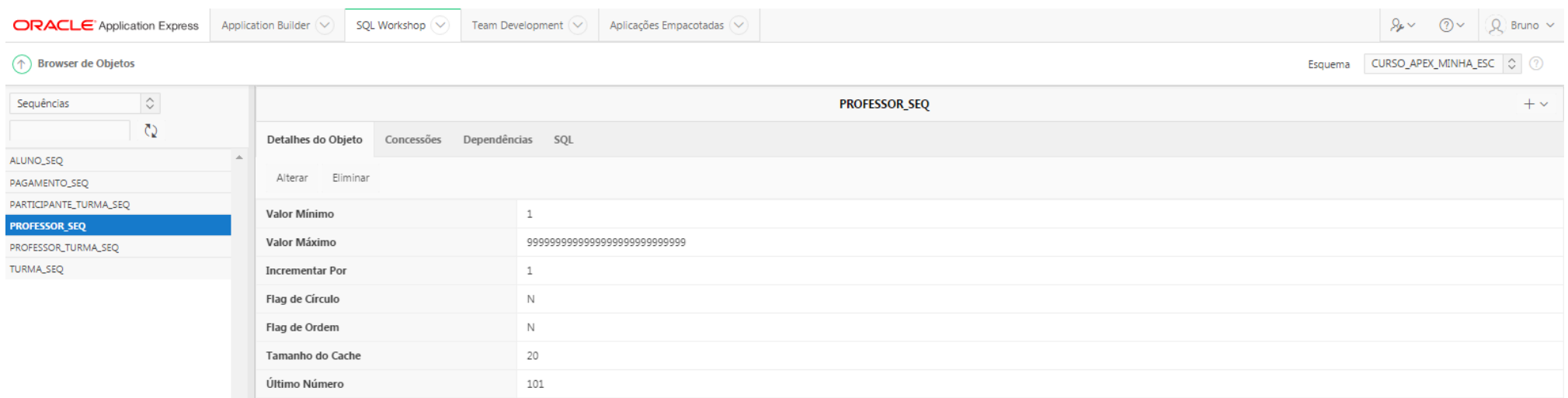
c.1. Sequência (Sequence)

É um objeto do banco de dados Oracle, especializado em armazenar e manter uma sequência de números obedecendo a regra definida na criação do objeto.

No APEX, ao criarmos uma tabela, temos a opção de criar uma sequência, que por padrão é criada como NOME_DA_TABELA_SEQ, e uma Trigger (objeto que veremos com mais detalhes a seguir) que cuidarão da população do campo que tem a chave primária.

Importante: Apesar do nome do objeto ser "Sequência", no Oracle acontecem alguns "saltos" na numeração gerada pela sequence. Caso este não seja um requisito necessário em seu sistema, não se preocupe com essa característica do Oracle!

Abaixo veremos a estrutura da sequence responsável por manter a chave primária da tabela PROFESSOR.



O principal aqui são os campos Valor Mínimo, que por padrão é 1 e o Último Número, que no exemplo está em 101. Isso significa que, nesse exemplo, os números 1 a 101 já foram usados, alterados e/ou deletados, e que o próximo número livre será o 102 (para uma chave primária, por exemplo).

Para mais informações sobre sequências acesse o link http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_6015.htm#sthref7018

c.2. Trigger

É um gatilho no banco de dados que dispara quando alguma alteração de dados na tabela acontece.

Ele pode ser disparado antes ou depois de um insert, delete ou update atingindo uma linha por vez ou um o conjunto de linhas.

Como exemplo de utilidade das triggers, podemos citar a que o APEX cria para implementar a chave primária com numeração automática.

Exemplo de código gerado pelo APEX para popular automaticamente a chave primária numérica COD da tabela PROFESSOR_TURMA:

```
CREATE OR REPLACE TRIGGER bi_professor_turma
BEFORE INSERT
ON professor_turma
FOR EACH ROW
BEGIN
    IF :NEW."COD" IS NULL
    THEN
        SELECT "PROFESSOR_TURMA_SEQ".NEXTVAL
        INTO :NEW."COD"
        FROM DUAL;
    END IF;
END;
```

Comentários sobre o código acima:

CREATE OR REPLACE – Essa parte do comando cria o objeto ou substitui o objeto existente pelo conteúdo do novo código.

TRIGGER – tipo do objeto a ser criado.

bi_professor_turma – nome do objeto a ser criado. Este é o padrão do APEX, o bi significa que essa trigger é executada antes da inserção (Before Insert) e o resto é o nome da tabela.

BEFORE INSERT ON professor_turma – essa parte indica quando a trigger vai ser disparada e qual o evento que irá dispará-la. Neste caso, quando o usuário for executar um insert no banco e confirmar a transação (commit), automaticamente será executado o código presente no corpo da trigger.

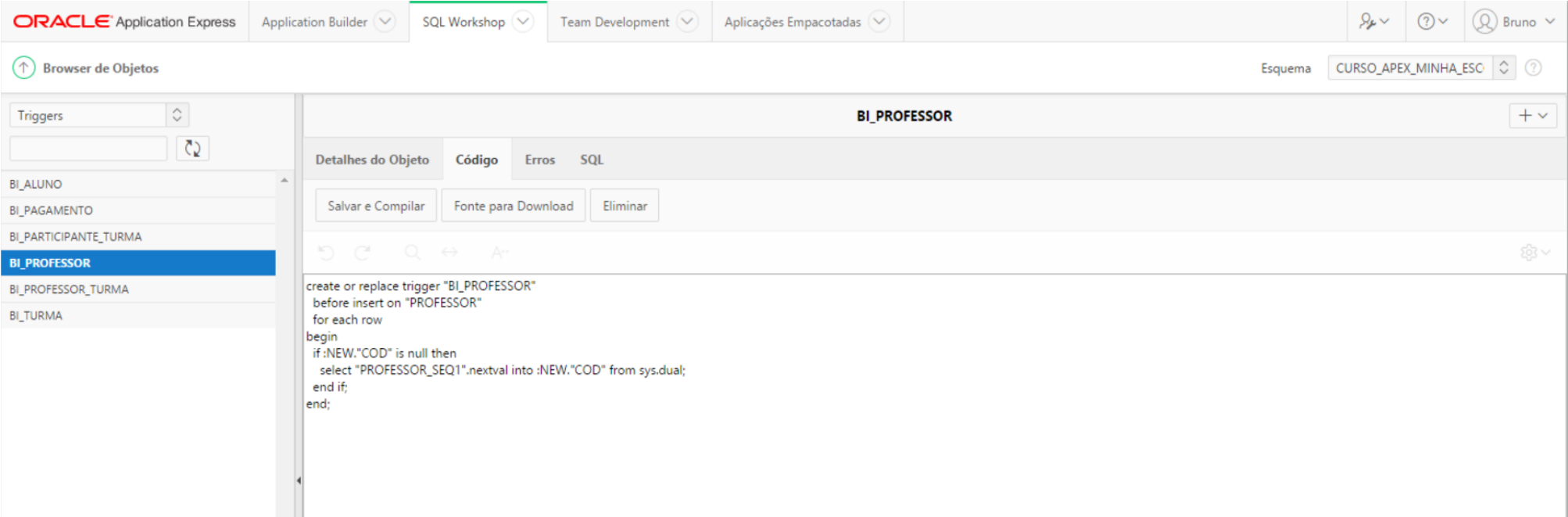
FOR EACH ROW – essa parte indica que, para cada linha inserida na tabela, o banco irá executar uma vez o código presente no corpo da trigger.

`BEGIN IF :NEW."COD" IS NULL THEN SELECT "PROFESSOR_TURMA_SEQ".NEXTVAL INTO :NEW."COD" FROM DUAL; END IF; END;` – esse é o corpo da trigger, o conjunto de comandos que serão processados quando ela for disparada. Basicamente, ao se inserir um novo registro, a trigger pega o próximo valor da sequence PROFESSOR_TURMA_SEQ com o método (comando) NEXTVAL e o atribui à chave primária COD.

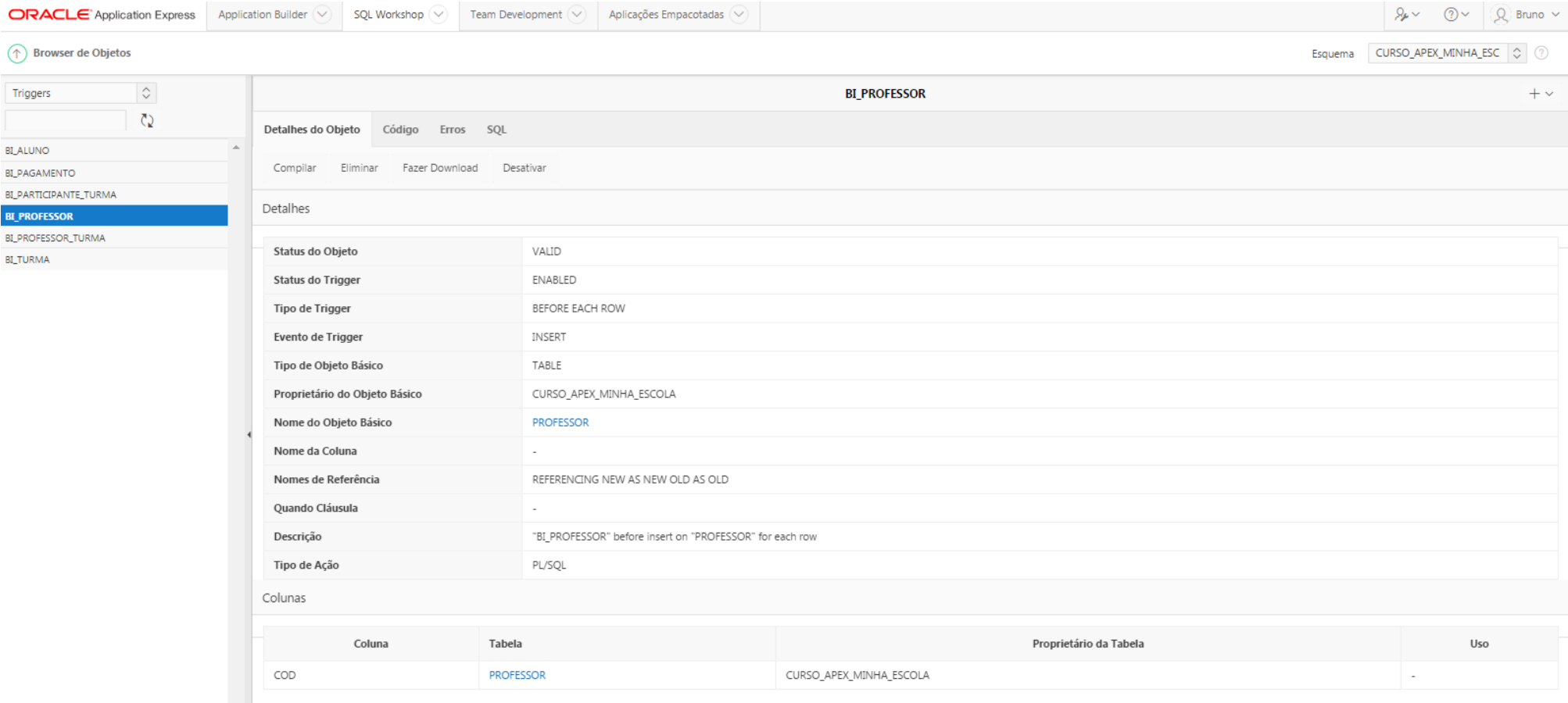
Obs: o :NEW existente no corpo da trigger, é uma referência ao valor passado pelo usuário durante o insert para o campo precedido do :NEW.

No APEX temos como as principais formas de visualização de uma trigger as telas **Detalhes do Objeto** e **Código**.

Código



Detalhes do Objeto



Para mais informações sobre triggers acesse o link http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_7004.htm#sthref7670

d. Índice (Index)

O próprio nome já nos remete para uma analogia verdadeira, os índices dos livros.

A função do índice no banco é exatamente a mesma, acelerar a busca da informação quando procuramos por um assunto específico.

Em caso de tabelas grandes (mais de um milhão de registros) é recomendável a criação de índices para que o tempo de resposta de uma consulta diminua drasticamente.

Dica: Se você faz referência a um campo da tabela, numa cláusula WHERE de uma instrução SQL, então analise se seria ou não o caso de criar um índice para esse campo.

Podemos ver na imagem abaixo um exemplo de um índice visto dentro do APEX.

ORACLEApplication Express

Application BuilderSQL WorkshopTeam DevelopmentAplicações Empacotadas

Bruno

↑

Browser de Objetos

EsquemaCURSO_APEX_MINHA_ESC

Índices

ALUNO_PK

ESCOLARIDADE_PK

PAGAMENTO_PK

PARTICIPANTE_TURMA_CON

PARTICIPANTE_TURMA_PK

PROFESSOR_PK

PROFESSOR_TURMA_CON

PROFESSOR_TURMA_PK

TURMA_PK

PROFESSOR_PK

+ ▾

Detalhes do ObjetoEstatísticasSQL

Desativar

Eliminar

Reconstruir

Nome do Índice

PROFESSOR_PK

Tipo de Índice

NORMAL

Proprietário da Tabela

CURSO_APEX_MINHA_ESCOLA

Nome da Tabela

PROFESSOR

Tipo de Tabela

TABLE

Unicidade

UNIQUE

Compactação

DISABLED

Tamanho do Prefixo

-

Nome do Tablespace

APEX_14587665143903906829

Status

VALID

Última Análise

07/11/15 00:14:43

Colunas de Índice

COLUMN_NAME	COLUMN_EXPRESSION	COLUMN_POSITION
COD	-	1

Para mais informações sobre índices acesse o [link](#).

Agora é com você!

Entre no seu ambiente “APEX -> SQL Workshop -> Browser de objetos” e explore os objetos existentes por lá.

3. Identificação do problema

Neste curso, faremos uma aplicação chamada ***Minha Escola*** a fim de exemplificar os recursos e técnicas básicas disponibilizadas pela ferramenta APEX.

Ela automatizará o funcionamento de uma organização educacional.

Teremos o controle de:

- disciplinas ministradas;
- alunos matriculados;
- pagamentos realizados;
- turmas ministradas por cada professor; e, por fim,
- temos que ter o controle das notas dos alunos.

Foram colocadas algumas regras de negócio (RN):

RN 01: Toda turma tem um ou mais professores, uma sala de aula, e vários alunos.

RN 02: As notas são classificadas de 0 (sem rendimento) a 10 (excelente rendimento).

RN 03: O aluno só pode entrar em uma turma que possua vagas.

RN 04: Todos os alunos têm nota em cada uma das turmas que participa.

RN 05: Todos os valores monetários são em reais (R\$), podendo eventualmente ser gratuitos.

Agora é com você!

1º passo - Para facilitar o seu entendimento do curso, coloque nos seus Favoritos o link https://contas.tcu.gov.br/ords/f?p=curso_apex.

(Para logar, utilize o usuário: **CONVIDADO** e a senha: **cursoapex**).

Lá você terá uma primeira ideia de como ficará o sistema Minha Escola em seu modo mais básico: sem o padrão visual e sem integração com o Login Integrado do TCU, sem maiores tratamentos de erro, etc.

2º passo - Navegue pela aplicação Minha Escola, clique nos links e botões disponíveis, insira, altere e exclua dados de teste.

3.1. Padrão de nomenclatura de objetos de banco

Para iniciar a modelagem dos dados, é muito importante utilizarmos uma **nomenclatura padrão**, pois agiliza o uso, facilita a leitura, facilita o entendimento do conteúdo (semântica) e facilita a pesquisa dessa instrução, tanto na sua aplicação como no banco de dados.

Informações simples e padronizadas fazem uma grande diferença para quem precisa analisar uma documentação ou precisa interpretar algum código.

É muito importante utilizarmos nomenclatura padrão para designação dos objetos do banco de dados, pois facilita a leitura e o entendimento do conteúdo, quer seja armazenado, quer seja pesquisado.

Caso você seja do TCU, veja o padrão de nomenclatura dos objetos no link da Comunidade APEX:

[https://acesso1.tcu.gov.br/comunidade/apex/Wiki/Página Inicial/Padrão de Nomenclatura para Objetos de Banco de Dados de Aplicações APEX.aspx](https://acesso1.tcu.gov.br/comunidade/apex/Wiki/Página%20Inicial/Padrão%20de%20Nomenclatura%20para%20Objetos%20de%20Banco%20de%20Dados%20de%20Aplicações%20APEX.aspx)

Caso não seja do TCU, busque o padrão de nomenclatura do seu local de trabalho.

A utilização de CONSTRAINTs (NOT NULL, FK, PK, Unique, Check) evita o erro causado pelo esquecimento do programador. Os dados tornam-se mais robustos. Uma tabela que possui muitos atributos Nullable, corre enorme risco de, em pouco tempo, não ter informação consistente em seus registros. Por outro lado, se tudo for desnecessariamente obrigatório, rapidamente veremos usuários finais digitando os famigerados 9999999999 nos campos só para o sistema gravar o registro.

A implementação das chaves estrangeiras (FK) garante que não será salvo qualquer dado nesta tabela se sua relação com a outra tabela não estiver satisfeita (FK sem conteúdo ou inválido). Para não ter outro valor diferente de "M" ou "F" no atributo IND_SEXO, alimentamos essa crítica. Podemos criar uma CONSTRAINT com INDEX associado para implementar chaves fortes, como Matrícula de Funcionário - que não deve ser a chave primária!

Podemos ainda utilizar outro recurso do APEX denominado Defaults de UI para designar como queremos a aparência dos conjuntos de dados. Mais adiante entraremos em detalhes sobre os Defaults de UI.

Agora é com você!

Analise o modelo de dados abaixo, que será utilizado pelo MinhaEscola.

Procure por erros de:

- 1) Tipo de dados incorreto
- 2) Nome de campo com erro de digitação
- 3) Normalização do modelo
- 4) Atendimento aos requisitos
- 5) Padronização de nomes de campo no singular

Este modelo atenderia aos requisitos de negócio da MinhaEscola?

Aluno		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
nome	Varchar2(50)	Não
data_nascimento	Date	
ind_sexo	Char(1)	Não
endereco_compelto	Varchar2(200)	
telefone	Varchar2(50)	
email	Varchar2(50)	Não

Participante_Turma		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
cod_aluno (FK)	Number	Não
cod_turma (FK)	Number	Não
nota	Date	

Pagamento		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
cod_participante_turma (FK)	Number	Não
data	Date	
valor	Number (10,2)	

Turma		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
descricao	Varchar2(50)	Não
periodo	Varchar2(50)	Não
horario	Varchar2(50)	Não
sala	Varchar2(20)	
qtd_vagas	Number	Não
valor_curso	Number	Não
ind_estado	Number	Não

Professor_Turma		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
cod_professor (FK)	Number	Não
cod_turma (FK)	Number	Não

Professor		
Campo	Tipo de dados	Opcional
cod (PK)	Number	Não
nome	Varchar2(50)	Não
nivel	Number	Não
curriculo	Varchar2(4000)	
telefones	Varchar2(50)	
email	Varchar2(50)	Não

3.2. Normalização

A normalização dos dados para bancos de dados relacionais (como o Oracle) é essencial para a qualidade do sistema desde sua concepção.

Uma definição de estrutura de dados pouco normalizada conduz a problemas posteriores. No curto prazo, problemas na confecção rápida de formulários e relatórios. No médio prazo, baixa robustez para incorporar novas funcionalidades. E no longo prazo, altíssimo custo de manutenção, levando normalmente ao abandono do sistema e seu completo redesenvolvimento.

Mesmo sistemas que trazem uma grande simplicidade em sua estrutura de dados, em geral, com duas ou três tabelas, sem classificações ou relações, necessitam da normalização como pré-requisito. Sugerimos que você persiga a Terceira Forma Normal (3FN), pois tem a melhor relação custo-benefício.

Se você não estiver familiarizado com a construção de um modelo de dados normalizado na 3FN, sugerimos que você busque aprender um pouco mais. Será essencial para o seu dia a dia como desenvolvedor APEX.

Agora é com você!

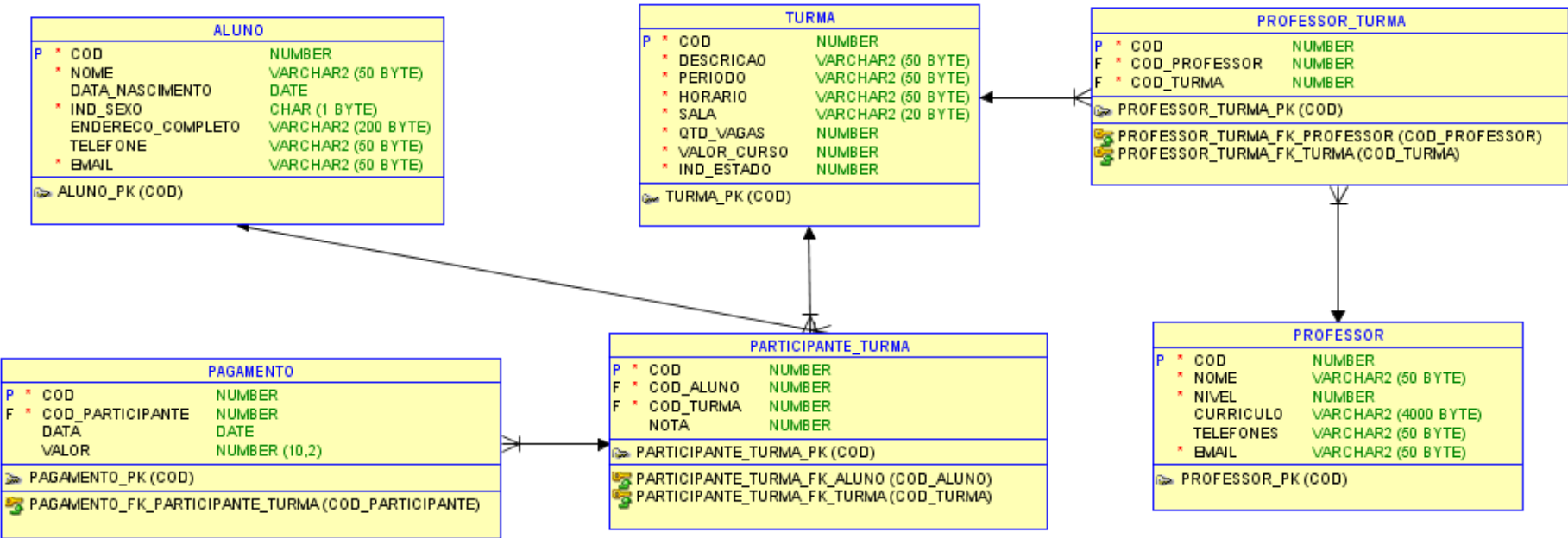
Caso você tenha dificuldade em normalização de tabelas, busque na internet algum material sobre o assunto.

No Google encontramos diversos materiais: <https://www.google.com/search?&q=normaliza%C3%A7%C3%A3o+de+banco+de+dados>

(Tempo estimado: 10 minutos)

3.3. Modelo de dados

Veja abaixo o modelo de dados que atende aos requisitos do sistema Minha Escola.



Já temos um modelo com condições de construirmos nossas tabelas no espaço de trabalho do APEX.

Agora é com você

Analise o modelo de dados acima com atenção para entender como funcionará o sistema (**3 minutos**)

4. Criando as tabelas do sistema

Agora que já conhecemos um pouco da nomenclatura do Oracle, chegou a hora de implementarmos nosso modelo completo no banco de dados utilizando o componente "**SQL Workshop**" do APEX.

Como vimos, o APEX é uma ferramenta de desenvolvimento web, residente dentro do banco de dados.

Quando você se conecta ao espaço de trabalho do APEX, você também está abrindo uma conexão com um esquema (owner) do banco de dados.

No APEX, para a construção dos objetos de banco, podemos utilizar o **SQL Workshop**. Veja alguma das vantagens e desvantagens de se utilizar esta abordagem:

Vantagens

- 1) Permite a criação de tabelas utilizando assistentes (wizards);
- 2) Está acessível utilizando apenas um browser, não necessitando da instalação de outra ferramenta;
- 3) Ao se criar uma tabela, é possível criar automaticamente a sua seqüência (sequence) e a sua trigger (gatilho).

Desvantagens

- 1) Não possui uma interface gráfica para criação de relacionamentos entre tabelas;
- 2) Pelo fato de ser web, necessita de conexão com a internet;
- 3) Não permite diferenciar que desenvolvedor APEX criou a tabela.

Observação: Não será objeto de estudo deste curso, mas é possível fazer a manipulação de objetos de banco de dados utilizando outras ferramentas de mercado, tais como SQL Developer, Toad for Oracle e PL/SQL Developer.

Na próxima seção, vamos iniciar a construção da tabela ALUNO do projeto Minha Escola!

4.1. Criando a tabela Aluno


Faremos neste curso a criação dos objetos de banco utilizando o **Browser de Objetos** do menu **SQL Workshop**.

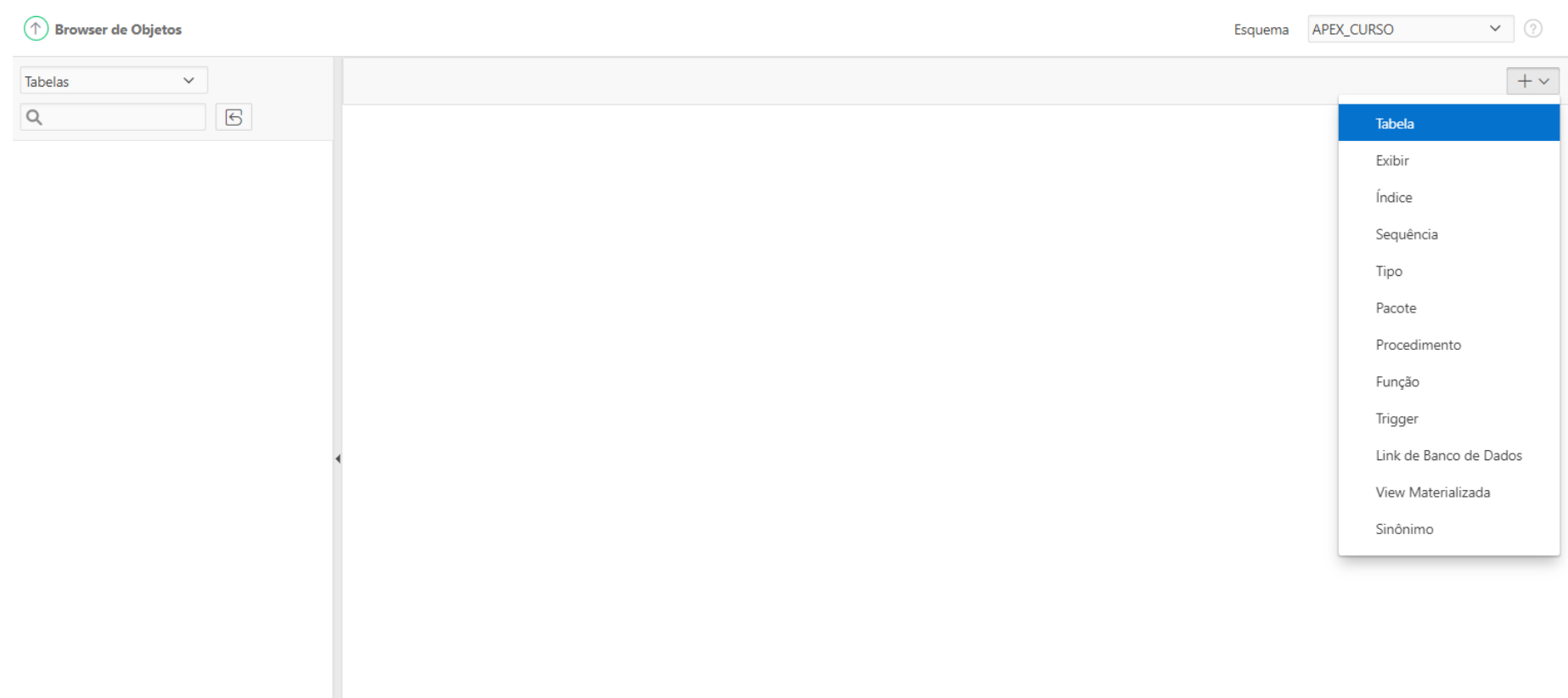
No canto superior direito da interface, há um botão que permite a criação de quaisquer objetos de banco: tabelas, views, índices, sequências, tipos, pacotes, procedimentos, funções, triggers, links de banco de dados, views materializadas e sinônimos.

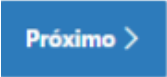
Agora é com você!

Siga os passos abaixo para criar a tabela ALUNO e os objetos de banco auxiliares.

Não se esqueça de ler as explicações adicionais!

1º Passo - Clique em **SQL Workshop** e, em seguida, no ícone **Browser de Objetos**. Clique no ícone de criação  que está localizado no canto superior direito da interface e selecione a opção "**Tabela**" conforme imagem abaixo.



2º Passo - Preencha as colunas como demonstrado abaixo. Ao terminar, clique no botão "**Próximo >**" . Vale a pena ressaltar que ao marcar o checkbox **Não Nulo**, os campos correspondentes passarão a ser obrigatórios. Quando você for criar formulários com o APEX, surgirão validações que impedirão a inserção de valores nulos para estes campos.

Criar Tabela



Colunas

* Nome da Tabela ?

☐ Preservar Maiúsculas/Minúsculas

Nome da Coluna	Tipo	Precisão	Escala	Não Nulo	Mover
<input type="text" value="COD"/>	NUMBER ▾	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	^ ▾
<input type="text" value="NOME"/>	VARCHAR2 ▾		<input type="text" value="50"/>	<input checked="" type="checkbox"/>	^ ▾
<input type="text" value="DATA_NASCIMENTO"/>	DATE ▾			<input type="checkbox"/>	^ ▾
<input type="text" value="IND_SEXO"/>	CHAR ▾		<input type="text" value="1"/>	<input checked="" type="checkbox"/>	^ ▾
<input type="text" value="ENDERECO_COMPLETO"/>	VARCHAR2 ▾		<input type="text" value="200"/>	<input type="checkbox"/>	^ ▾
<input type="text" value="TELEFONE"/>	VARCHAR2 ▾		<input type="text" value="20"/>	<input type="checkbox"/>	^ ▾
<input type="text" value="EMAIL"/>	VARCHAR2 ▾		<input type="text" value="20"/>	<input checked="" type="checkbox"/>	^ ▾
<input type="text"/>	- Selecionar Tipo de Dados - ▾				^ ▾

Adicionar Coluna

Cancelar

Próximo >

Dicas:
Utilize maiúsculas para nomes de tabelas e de campos
NÃO marque a caixa " Preservar Maiúsculas/Minúsculas "
(Isso permitirá digitar em seus SELECTs nomes de tabelas tanto em maiúsculas quanto em minúsculas)
Procure colocar os nomes no singular
Utilize _ (underline) como separador. Exemplo: DATA_NASCIMENTO, VALOR_CURSO, IND_ESTADO
Utilize como chave estrangeira COD_<nome da tabela referenciada>. Exemplo: COD_TURMA, COD_PROFESSOR

Já o campo Mover altera a posição da coluna, podendo subi-la ou descê-la em referência aos campos anteriores ou posteriores, dependendo da seta que você escolher. Tente organizar os campos da forma mais lógica possível do ponto de vista de formulários para o usuário final. Essa ordem será aproveitada pelos assistentes do APEX.

3º Passo - Nesta tela, escolheremos a Chave Primária: Preenchido por uma nova sequência. Com isso, o APEX criará automaticamente o objeto Sequência (Sequence).

Criar Tabela

Chave Primária

Nome da tabela:

ALUNO

Chave Primária:

Não Há Chave Primária

Preenchido por uma nova sequência

Preenchido por uma sequência existente

Não preenchido

Preenchido pela coluna Identidade

* Nome de Constraint da Chave Primária

ALUNO_PK

* Chave Primária

COD(NUMBER)

* Nome da Sequência

ALUNO_SEQ

Chave Primária

Uma chave primária permite que cada linha de uma tabela seja identificada de maneira única.

Se optar por preencher a sua chave primária com base em uma nova sequência, você deverá informar o nome da nova sequência. Se optar por preencher a sua chave primária com base em uma sequência existente, você deverá selecionar a sequência. Ambos os métodos resultam na geração de um trigger para a sua tabela. Você também pode optar por não gerar uma chave primária. Este é o único método que permite definir uma chave primária composta que é uma chave primária formada por mais de duas colunas.

<

Cancelar

Próximo >

Os atributos **Nome de Constraint da Chave Primária** e **Nome da Sequência** não precisam ser alterados, pois já estão no padrão do TCU.

Na **Chave Primária** selecione o campo COD(NUMBER). Em seguida clique "**Próximo >**".

4º passo - Esta tela é utilizada quando você possui uma chave estrangeira entrando na tabela. Como não há chave estrangeira entrando na tabela ALUNO, clique em "**Próximo >**"

ATENÇÃO: EVITE utilizar chave estrangeira com a opção "**Exclusão em cascata**"!!!
Prefira a opção "**Não Permitir Exclusão**" e, caso necessário, faça a exclusão dos registros nas tabelas filhas pela aplicação!

Criar Tabela

Chave Estrangeira

Chaves Estrangeiras

Chave Estrangeira

Colunas

Tabela Referenciada

Colunas Referenciadas

Ação

Adicionar Chave Estrangeira

Nome

ALUNO_FK

Não Permitir Exclusão

Exclusão em Cascata

Definir Nulo ao Excluir

Selecionar Coluna(s)-Chave

COD

NOME

DATA_NASCIMENTO

IND_SEXO

ENDERECO_COMPLETO

TELEFONES

EMAILS

>>

>

<

<<

Faz Referência à Tabela

Coluna(s) Referenciada(s)

>>

>

<

<<

Adicionar

Caso houvesse chave estrangeira, o procedimento seria:

- escolher o nome da chave estrangeira. O padrão do TCU determina que o nome deve ser: <<NOME_TABELA>>_FK_<NOME_TABELA_PAIS> (Fique atento ao limite de 30 caracteres);
- clique no campo que é a chave estrangeira na tabela e em seguida no símbolo ">" ;

- 3. digitar o nome da tabela pai (ou mestre) no campo "Faz Referência à Tabela" ou clicar no símbolo para ver uma lista de todas as tabelas até então criadas.
Clique sobre a tabela desejada;
- 4. escolha a coluna da lista **Coluna(s) Referenciadas** e em seguida no símbolo ">";
- 5. para adicionar a chave estrangeira, clique em "**Adicionar**"; (*Não esqueça deste passo!*)
- 6. após incluir todas as chaves estrangeiras, clique em "Próximo >"

Caso você se esqueça de criar as Chaves Estrangeiras neste passo, lembre-se: é possível criá-las depois.

5º Passo - Criando as regras de restrição (constraint) de dados no banco. Essas constraints são criadas para garantir que independentemente da entrada de dados, o banco só aceitará valores que passarem pelo critério aqui expresso. O exemplo abaixo mostra a regra que restringe os valores do campo ind_sexo a M ou F. Caso o valor seja diferente desses, irá acontecer um erro disparado pelo banco de dados.

Veremos agora como criar a regra de restrição abaixo mostrada:

- 1. clique na opção Verificar e em seguida na caixa de texto;
- 2. escreva na caixa de texto a regra que fica assim: `ind_sexo in ('M','F');`
- 3. no campo Nome coloque o nome da constraint no padrão TCU que seria ALUNO_CK_IND_SEXO, onde:
 - o ALUNO - é o nome da tabela em que a constraint vai agir;
 - o CK - indica que esta constraint é de verificação (check);
 - o IND_SEXO - é o nome da coluna em que a constraint vai agir;
- 4. para finalizar a criação desta restrição, clique em "**Adicionar**";
- 5. caso precise remover uma regra basta clicar no X após o campo Coluna(s)/Verificação;
- 6. se todas as regras de restrições já tiverem sido implementadas, clique em "**Próximo>**".

Criar Tabela

Constraints

Nome da Constraint	Tipo	Coluna(s)/Verificação
	<div>Tipo de Constraint<div><div><div></div></div>Verificar<div><div></div></div>Única</div></div>	<div>Condição de Verificação<div>IND_SEXO IN ('M','F')</div></div>

Coluna(s)-Chave

COD

NOME

DATA_NASCIMENTO

ENDERECO_COMPLETO

TELEFONE

EMAIL

>>

>

<

<<

IND_SEXO

* Nome

ALUNO_CK_IND_SEXO

6º Passo - Finalizando a criação da tabela. Este é o último passo antes da tabela ser criada no banco de dados. Seja curioso e clique em SQL e você verá todo o comando SQL que será executado para criar sua tabela, o índice de chave primária, a sequence e a trigger. Clique no botão "**Criar Tabela**" para que a tabela e todos os objetos auxiliares sejam efetivamente criados no banco de dados.

Criar Tabela

Confirmar

Confirme a solicitação.

Esquema:

CURSO_APEX_MINHA_ESCOLA

Nome da tabela:

ALUNO

> SQL

<

Cancelar

Criar Tabela

Após a geração da tabela, o APEX irá voltar para a primeira tela do Browser de Objetos, desta vez, mostrando a estrutura da tabela criada como vemos na imagem abaixo.

PRODUCAO

App Builder

SQL Workshop

Desenvolvimento da Equipe

Galeria de Aplicativos

Browser de Objetos

Esquema

APEX_CURSO

Tabelas

ALUNO

ALUNO

Tabela

Dados

Índices

Modelo

Constraints

Concessões

Estatísticas

Padrões da UI

Triggers

Dependências

SQL

Adicionar Coluna

Modificar Coluna

Renomear Coluna

Eliminar Coluna

Renomear

Copiar

Eliminar

Truncar

Criar Tabela de Consulta

Nome da Coluna	Tipo de Dados	Anulável	Padrão	Chave Primária
COD	NUMBER	Não	-	1
NOME	VARCHAR2(50)	Não	-	-
DATA_NASCIMENTO	DATE	Sim	-	-
IND_SEXO	CHAR(1)	Não	-	-
ENDEREÇO_COMPLETO	VARCHAR2(200)	Sim	-	-
TELEFONE	VARCHAR2(20)	Sim	-	-
EMAIL	VARCHAR2(20)	Sim	-	-

Fazer Download | Imprimir

Na próxima seção iremos criar as outras tabelas do projeto Minha Escola. Até lá!

4.2. Criando as demais tabelas

Vamos agora criar todas as demais tabelas e seus objetos de banco associados para o projeto "Minha Escola".

Em cada seção abaixo, crie a tabela correspondente e os objetos de banco auxiliares, inclusive as chaves estrangeiras. Não se esqueça de ler as explicações adicionais.

Obs.: Em todas as tabelas considere como a chave primária a coluna COD

a. Tabela TURMA

Nome da Coluna	Tipo	Escala	Não Nulo
COD	NUMBER		√
DESCRICAO	VARCHAR2	50	√
PERIODO	VARCHAR2	50	√
HORARIO	VARCHAR2	50	√
SALA	VARCHAR2	20	
QTD_VAGAS	NUMBER		√
VALOR_CURSO	NUMBER		√
IND_SITUACAO	CHAR	1	√

Obs. 1: o período e o horário são do tipo VARCHAR2 para comportar descrição livre. Se fosse requisito do sistema ordenar e organizar turmas por horário, por exemplo, mais tabelas derivadas e uma normalização mais completa seria necessária.

Obs. 2: veja que a escala do campo SALA só tem 20 caracteres. Evite colocar valores maiores que o necessário (otimiza o banco), mas lembre-se de deixar alguma margem para o futuro.

Constraint de chave primária: TURMA_PK (usar nome padrão)

Nome da sequência: TURMA_SEQ (usar nome padrão)

Nome da constraint	Tipo	Restrição	Justificativa
TURMA_CK_QTD_VAGAS	Verificação (check)	qtd_vagas >= 0	Aumento da qualidade da informação, evitando quantidade negativa de vagas.
TURMA_CK_VALOR_CURSO	Verificação (check)	valor_curso >= 0	Permite apenas cursos pagos, ou seja, com valor positivo.
TURMA_CK_IND_SITUACAO	Verificação (check)	IND_SITUACAO in ('M', 'A', 'E')	Restringe as situações possíveis de uma turma: M - Matrículas abertas A - Em andamento E - Curso encerrado Obs: Para campos indicativos , utilize códigos explicativos como os usados acima. Evite usar códigos numéricos como 1, 2, 3, pois dificulta o entendimento. Só use códigos numéricos, quando for chave estrangeira.

b. Tabela PROFESSOR

Nome da Coluna	Tipo	Escala	Não Nulo
COD	NUMBER		X
NOME	VARCHAR2	50	X
IND_NIVEL	CHAR	1	X
CURRICULO	VARCHAR2	4000	
TELEFONE	VARCHAR2	50	
EMAIL	VARCHAR2	50	X

Obs. 1: os campos TELEFONE e EMAIL são do tipo VARCHAR2 e com espaço suficiente para comportar descrição livre. Caso seja requisito armazenar mais de um telefone ou mais de um e-mail, avalie a possibilidade de criar uma tabela para armazenar os contatos.

Obs. 2: veja que os campos IND_NIVEL e CURRICULO estão sem acento. Evite utilizar acento em nomes de objetos e campos no Oracle!

Constraint de chave primária: PROFESSOR_PK (usar nome padrão)

Nome da sequência: PROFESSOR_SEQ (usar nome padrão)

Nome da constraint	Tipo	Restrição	Justificativa

PROFESSOR_CK_IND_NIVEL	Verificação (check)	IND_NIVEL in ('F','M','S','P')	Indica o nível de escolaridade do professor. Foi utilizado um código indicativo em vez de uma tabela, pois é esperado que os níveis de escolaridade não sofram alteração. F - Fundamental M - Médio S - Superior P - Pós-graduação
------------------------	---------------------	--------------------------------	---

c. Tabela ALUNO_TURMA

Nome da Coluna	Tipo	Não Nulo
COD	NUMBER	X
COD_ALUNO	NUMBER	X
COD_TURMA	NUMBER	X
NOTA	NUMBER	

Constraint de chave primária: ALUNO_TURMA_PK (usar nome padrão)

Nome da sequência: ALUNO_TURMA_SEQ (usar nome padrão)

Chave estrangeira	Colunas	Tabela referenciada	Colunas referenciadas
ALUNO_TURMA_FK_ALUNO	COD_ALUNO	ALUNO	COD
ALUNO_TURMA_FK_TURMA	COD_TURMA	TURMA	COD

Nome da constraint	Tipo	Restrição	Justificativa
ALUNO_TURMA_CK_NOTA	Verificação (check)	nota >= 0	Restringe as notas para valores não negativos.

d. Tabela PROFESSOR_TURMA

Nome da Coluna	Tipo	Não Nulo
COD	NUMBER	X
COD_PROFESSOR	NUMBER	X
COD_TURMA	NUMBER	X

Constraint de chave primária: PROFESSOR_TURMA_PK (usar nome padrão)

Nome da sequência: PROFESSOR_TURMA_SEQ (usar nome padrão)

Chave estrangeira	Colunas	Tabela referenciada	Colunas referenciadas
PROFESSOR_TURMA_FK_PROFESSOR	COD_PROFESSOR	PROFESSOR	COD
PROFESSOR_TURMA_FK_TURMA	COD_TURMA	TURMA	COD

e. Tabela PAGAMENTO

Nome da Coluna	Tipo	Precisão	Escala	Não Nulo
COD	NUMBER			X
COD_ALUNO_TURMA	NUMBER			X
DATA	DATE			

VALOR	NUMBER			
-------	--------	--	--	--

Constraint de chave primária: PAGAMENTO_PK (usar nome padrão)

Nome da sequência: PAGAMENTO_SEQ (usar nome padrão)

Chave estrangeira	Colunas	Tabela referenciada	Colunas referenciadas
PAGAMENTO_FK_ALUNO_TURMA	COD_ALUNO_TURMA	ALUNO_TURMA	COD

Obs: Procure utilizar os campos numéricos sem precisão e escala para evitar problemas futuros (principalmente no PLSQL).

Parabéns!!! Você finalizou uma etapa importante do sistema!

EsquemaAPEX_CURSO?			Chaves Estrangeiras Não Indexadas
Q Ir Ações Redefinir			Este relatório exibe tabelas que não têm chaves estrangeiras indexadas. A falha ao indexar chaves estrangeiras pode resultar em desempenho de join deficiente e bloquear a escalação. Normalmente, é uma boa prática de programação indexar as chaves estrangeiras.
Nome da Tabela	Nome da Constraint	Coluna de Chave Estrangeira 1	
ALUNO_TURMA	ALUNO_TURMA_FK_ALUNO	COD_ALUNO	
ALUNO_TURMA	ALUNO_TURMA_FK_TURMA	COD_TURMA	
PAGAMENTO	PAGAMENTO_FK_ALUNO_TURMA	COD_ALUNO_TURMA	
PROFESSOR_TURMA	PROFESSOR_TURMA_FK_TURMA	COD_TURMA	
PROFESSOR_TURMA	PROFESSOR_TURMA_FK_PROFESSOR	COD_PROFESSOR	
1 - 5			

3 - Para resolver, a situação do campo COD_ALUNO_TURMA na tabela PAGAMENTO, devemos clicar na aba **SQL Workshop** e em seguida no **Browser de Objetos** e selecionar na tabela PAGAMENTO a aba **Índices** e clicar no botão "**Criar**".

Browser de Objetos

EsquemaAPEX_CURSO?

Tabelas

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

PAGAMENTO

TabelaDadosÍndicesModeloConstraintsConcessõesEstatísticasPadrões da UITriggersDependênciasSQL

CriarEliminar

Nome do Índice	Unicidade	Colunas	Status	Tipo de Índice	Temporário	Particionado	Status da Função	Índice de Join
PAGAMENTO_PK	UNIQUE	COD	VALID	NORMAL	N	NO	-	NO

4 - Selecione o tipo de índice "Normal" e clique em "**Próximo >**"

Browser de Objetos

EsquemaAPEX_CURSO?

Tabelas

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

PAGAMENTO

Criar Índice

Esquema: APEX_CURSO?

Tabela: PAGAMENTO?

Tipo de Índice: ☒ Normal ☐ Texto?

CancelarPróximo >

5 - Informe o nome do índice: PAGAMENTO_IDX_COD_ALUNO_TURMA, selecione a opção "**Não Único**" em Unicidade e escolha como "Coluna 1 do Índice" a opção "**COD_ALUNO_TURMA (NUMBER)**". Após isso, clique em "**Próximo >**".

↑
Browser de Objetos

EsquemaAPEX_CURSO

Tabelas

ALUNO
ALUNO_TURMA
PAGAMENTO
PROFESSOR
PROFESSOR_TURMA
TURMA

PAGAMENTO

+

▼

Criar Índice

Esquema: APEX_CURSO

Tabela: PAGAMENTO

* Nome do ÍndicePAGAMENTO_IDX_COD_ALUNO_TURMA

?

☐ Preservar Maiúsculas/Minúsculas

UnicidadeNão Único

?

* Coluna 1 do ÍndiceCOD_ALUNO_TURMA (NUMBER)

?

Coluna 2 do Índice- Seleccionar Coluna -

?

Coluna 3 do Índice- Seleccionar Coluna -

?

Coluna 4 do Índice- Seleccionar Coluna -

?

<Cancelar

Próximo >

6 - Clique no botão "**Criar Índice**".

↑
Browser de Objetos

EsquemaAPEX_CURSO

Tabelas

ALUNO
ALUNO_TURMA
PAGAMENTO
PROFESSOR
PROFESSOR_TURMA
TURMA

PAGAMENTO

+

▼

Confirmar sua solicitação

Objeto: PAGAMENTO_IDX_COD_ALUNO_TURMA

Ação: CREATE_INDEX

<Cancelar

Criar Índice

▶

SQL

7 - Com o índice criado, veremos a estrutura do mesmo.

↑

Browser de Objetos

Esquema

APEX_CURSO

?

Índices

Q

ALUNO_PK

ALUNO_TURMA_PK

PAGAMENTO_IDX_COD_ALUNO_TURMA

PAGAMENTO_PK

PROFESSOR_PK

PROFESSOR_TURMA_PK

TURMA_PK

PAGAMENTO_IDX_COD_ALUNO_TURMA

+ v

Detalhes do Objeto

Estatísticas

SQL

Desativar

Eliminar

Reconstruir

Nome do Índice

PAGAMENTO_IDX_COD_ALUNO_TURMA

Tipo de Índice

NORMAL

Proprietário da Tabela

APEX_CURSO

Nome da Tabela

PAGAMENTO

Tipo de Tabela

TABLE

Unicidade

NONUNIQUE

Compactação

DISABLED

Tamanho do Prefixo

-

Nome do Tablespace

APEX_1366930248539506764

Status

VALID

Última Análise

20/08/20 10:41:56

8 - Por fim verifique novamente que o relatório de "Chaves Estrangeiras Não Indexadas" **NÃO** apresenta mais a coluna COD_ALUNO_TURMA da tabela PAGAMENTO .

↑

Utilitários \ Relatórios de Objetos \ Chaves Estrangeiras Não Indexadas

Esquema

APEX_CURSO

?

Q v

Ir

Ações v

Redefinir

Nome da Tabela

Nome da Constraint

Coluna de Chave Estrangeira 1

ALUNO_TURMA

ALUNO_TURMA_FK_ALUNO

COD_ALUNO

ALUNO_TURMA

ALUNO_TURMA_FK_TURMA

COD_TURMA

PROFESSOR_TURMA

PROFESSOR_TURMA_FK_TURMA

COD_TURMA

PROFESSOR_TURMA

PROFESSOR_TURMA_FK_PROFESSOR

COD_PROFESSOR

1 - 4

Chaves Estrangeiras Não Indexadas

Este relatório exibe tabelas que não têm chaves estrangeiras indexadas. A falha ao indexar chaves estrangeiras pode resultar em desempenho de join deficiente e bloquear a escalação. Normalmente, é uma boa prática de programação indexar as chaves estrangeiras.

Agora é com você!

- 1 - Crie os índices das Chaves Estrangeiras Não Indexadas da tabela ALUNO_TURMA (ALUNO_TURMA_FK_TURMA e ALUNO_TURMA_FK_ALUNO).
- 2 - Faça o mesmo para a tabela PROFESSOR_TURMA (PROFESSOR_TURMA_FK_TURMA e PROFESSOR_TURMA_FK_PROFESSOR).

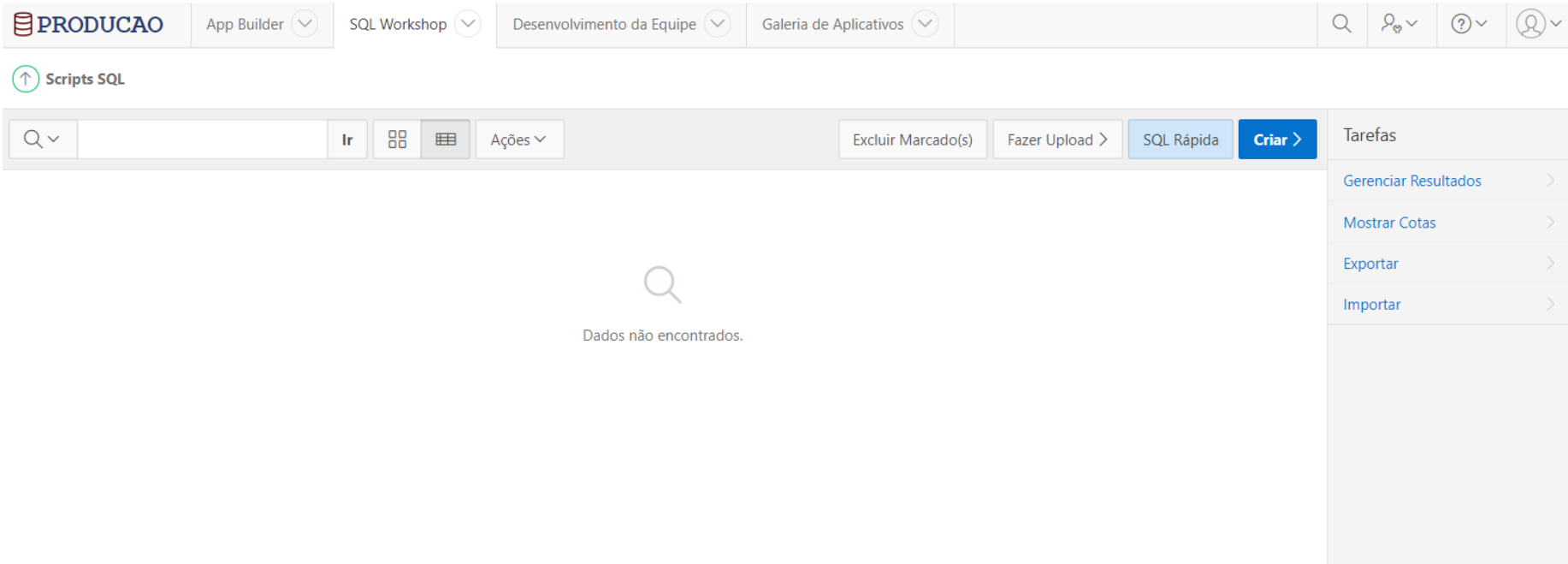
6. Fazendo a carga inicial de dados

Nesta seção iremos fazer uma carga inicial de dados em algumas das tabelas criadas utilizando Scripts SQL, Comandos SQL e Data Workshop.

6.1. Utilizando os Scripts SQL

Veremos a seguir o passo a passo para a execução de script SQL para carga de dados de exemplo.

Passo 1 – Dentro do APEX, clique em **SQL Workshop**, dentro dele clique em **Scripts SQL** e clique no botão "**Criar**>".



Passo 2 - Digite um nome para o script, copie e cole o código abaixo e clique em "**Executar**".

```
--INÍCIO DO CÓDIGO A COPIAR--

insert into aluno values(aluno_seq.nextval,'Alberto Domingues Esteves',to_date('07/11/1974','dd/mm/yyyy'),'M','QRSW-05 Bl.B-3 Apt .206 - Sudoeste',null,'alberto@apex.com');

insert into aluno values(aluno_seq.nextval,'Ana Maria Alves',to_date('30/12/1981','dd/mm/yyyy'),'M','SQN 406 bloco D apart 107',null,'ana@apex.com');

insert into aluno values(aluno_seq.nextval,'Carlos Alberto Alves',to_date('27/03/1980','dd/mm/yyyy'),'M','QNL 30 TAGUATINGA NORTE DF',null,'carlos@apex.com');

insert into aluno values(aluno_seq.nextval,'Cristina Pires Domingues',to_date('18/11/1979','dd/mm/yyyy'),'M','QNA 36 CASA 08',null,'cristina@apex.com');

insert into aluno values(aluno_seq.nextval,'David Cunha Goncalves',to_date('03/04/1984','dd/mm/yyyy'),'M','SQS 116 BLOCO E APT 401',null,'david@apex.com');

insert into aluno values(aluno_seq.nextval,'Eusebio dos Santos Aguiar',to_date('15/05/1985','dd/mm/yyyy'),'M','SHCES QUADRA 1405 BLOCO H APARTAMENTO 304',null,'eusebio@apex.com');

insert into aluno values(aluno_seq.nextval,'Fabrice Stephane Alves Pereira',to_date('31/10/1988','dd/mm/yyyy'),'M','QI 16 BLOCO O APT. 301',null,'fabrice@apex.com');

insert into aluno values(aluno_seq.nextval,'Fernanda Maria Rodrigues',to_date('24/06/1975','dd/mm/yyyy'),'M','SQSW 504 BLOCO H APARTAMENTO 209',null,'fernanda@apex.com');

insert into aluno values(aluno_seq.nextval,'Joao Paulo Barros da Silva',to_date('26/12/1978','dd/mm/yyyy'),'M','QD.104 LT.08 BL.A APTO 301',null,'joao@apex.com');

insert into aluno values(aluno_seq.nextval,'Tania Maria Pereira',to_date('11/08/1984','dd/mm/yyyy'),'M','HIGS 709 BLOCO C CASA 21 ASA SUL',null,'tania@apex.com');

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Alexandre Domingues Campos','P','Pós graduado em Engenharia de Computação','61 99999-9999','alexandre@apex.com');

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Fernando Edmundo Chermont Vidal','M','Experiência em programação competitiva. Possui diversas certificações profissionais','61 98888-8888','fernando@apex.com');

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Jose Augusto Abreu Sa Fortes','S','Bacharel em Ciência da Computação','61 97777-7777','jose@apex.com');

--FIM DO CÓDIGO A COPIAR--
```

Scripts SQL \ Editor de Script

Nome do Script

Insert curso apex

?

Cancelar

Fazer Download

Excluir

Salvar

Executar

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

insert into aluno values(aluno_seq.nextval,'Alberto Domingues Esteves',to_date('07/11/1974','dd/mm/yyyy'),'M','QRSW-05 B1.B-3 Apt .206 - Sudoeste',null,'alberto@apex.com');

insert into aluno values(aluno_seq.nextval,'Ana Maria Alves',to_date('30/12/1981','dd/mm/yyyy'),'M','SQN 406 bloco D apart 107',null,'ana@apex.com');

insert into aluno values(aluno_seq.nextval,'Carlos Alberto Alves',to_date('27/03/1980','dd/mm/yyyy'),'M','QNL 30 TAGUATINGA NORTE DF',null,'carlos@apex.com');

insert into aluno values(aluno_seq.nextval,'Cristina Pires Domingues',to_date('18/11/1979','dd/mm/yyyy'),'M','QNA 36 CASA 08',null,'cristina@apex.com');

insert into aluno values(aluno_seq.nextval,'David Cunha Goncalves',to_date('03/04/1984','dd/mm/yyyy'),'M','SQS 116 BLOCO E APT 401',null,'david@apex.com');

insert into aluno values(aluno_seq.nextval,'Eusebio dos Santos Aguiar',to_date('15/05/1985','dd/mm/yyyy'),'M','SHCES QUADRA 1405 BLOCO H APARTAMENTO 304',null,'eusebio@apex.com');

insert into aluno values(aluno_seq.nextval,'Fabrice Stephane Alves Pereira',to_date('31/10/1988','dd/mm/yyyy'),'M','QI 16 BLOCO O APT. 301',null,'fabrice@apex.com');

insert into aluno values(aluno_seq.nextval,'Fernanda Maria Rodrigues',to_date('24/06/1975','dd/mm/yyyy'),'M','SQSW 504 BLOCO H APARTAMENTO 209',null,'fernanda@apex.com');

insert into aluno values(aluno_seq.nextval,'Joao Paulo Barros da Silva',to_date('26/12/1978','dd/mm/yyyy'),'M','QD.104 LT.08 BL.A APTO 301',null,'joao@apex.com');

insert into aluno values(aluno_seq.nextval,'Tania Maria Pereira',to_date('11/08/1984','dd/mm/yyyy'),'M','HIGS 709 BLOCO C CASA 21 ASA SUL',null,'tania@apex.com');

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Alexandre Domingues Campos','P','Pós graduado em Engenharia de Computação','61 99999-9999','alexandre@apex.co

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Fernando Edmundo Chermont Vidal','M','Experiência em programação competitiva. Possui diversas certificações p

insert into professor (nome, ind_nivel, curriculo, telefone, email) values ('Jose Augusto Abreu Sa Fortes','S','Bacharel em Ciência da Computação','61 97777-7777','jose@apex.com');

Passo 3 – Veremos então a tela de confirmação. Basta então clicarmos no botão **Executar Agora** para confirmamos a execução do script. Após esta confirmação, o script será executado.

Executar Script

Você solicitou a execução do script a seguir. Confirme sua seleção.

Nome do Script	insert curso apex
Criado	em 25/10/20 18:56:38 por ANDERSONRF
Atualizado	-
Número de Instruções	13
Tamanho do Script em Bytes	2.296

Cancelar

Executar Agora

Passo 4 - Nesta tela, veremos que o script foi executado corretamente, com o status "Concluído".

Scripts SQL \ Resultados

Script: insert curso apex

Status: Concluído

Exibir:

Detalhe

Resumo

Linhas: 15

Ir

Criar Aplicativo Usando Script

Editar Script

Número	Decorrido	Instrução	Feedback	Linhas
1	0,01	insert into aluno values(aluno_seq.nextval,'Alberto Domingue	1 linha(s) inserida(s).	1
2	0,01	insert into aluno values(aluno_seq.nextval,'Ana Maria Alves	1 linha(s) inserida(s).	1
3	0,00	insert into aluno values(aluno_seq.nextval,'Carlos Alberto	1 linha(s) inserida(s).	1
4	0,00	insert into aluno values(aluno_seq.nextval,'Cristina Pires	1 linha(s) inserida(s).	1
5	0,00	insert into aluno values(aluno_seq.nextval,'David Cunha Gon	1 linha(s) inserida(s).	1
6	0,00	insert into aluno values(aluno_seq.nextval,'Eusebio dos San	1 linha(s) inserida(s).	1
7	0,00	insert into aluno values(aluno_seq.nextval,'Fabrice Stephan	1 linha(s) inserida(s).	1
8	0,00	insert into aluno values(aluno_seq.nextval,'Fernanda Maria	1 linha(s) inserida(s).	1
9	0,00	insert into aluno values(aluno_seq.nextval,'Joao Paulo Barr	1 linha(s) inserida(s).	1
10	0,00	insert into aluno values(aluno_seq.nextval,'Tania Maria Per	1 linha(s) inserida(s).	1
11	0,00	insert into professor (nome, ind_nivel, curriculo, telefone	1 linha(s) inserida(s).	1
12	0,00	insert into professor (nome, ind_nivel, curriculo, telefone	1 linha(s) inserida(s).	1
13	0,00	insert into professor (nome, ind_nivel, curriculo, telefone	1 linha(s) inserida(s).	1

Fazer Download

linha(s) 1 - 13 de 13

Para vermos o que foi inserido, podemos clicar na aba **SQL Workshop**, em seguida, no ícone **Browser de Objetos**, clicar na tabela **ALUNO** e em seguida na aba **Dados**.

Browser de Objetos

EsquemaAPEX_CURSO

Tabelas

Q

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

ALUNO

+

Tabela

Dados

Índices

Modelo

Constraints

Concessões

Estatísticas

Padrões da UI

Triggers

Dependências

SQL

Consulta

Contar Linhas

Inserir Linha

Dados

EDITAR	COD	NOME	DATA_NASCIMENTO	IND_SEXO	ENDereco_COMPLETO	TELEFONE	EMAIL
	1	Alberto Domingues Esteves	07/11/74	M	QRSW-05 Bl.B-3 Apt .206 - Sudoeste	-	alberto@apex.com
	2	Ana Maria Alves	30/12/81	M	SQN 406 bloco D apart 107	-	ana@apex.com
	3	Carlos Alberto Alves	27/03/80	M	QNL 30 TAGUATINGA NORTE DF	-	carlos@apex.com
	4	Cristina Pires Domingues	18/11/79	M	QNA 36 CASA 08	-	cristina@apex.com
	5	David Cunha Goncalves	03/04/84	M	SQS 116 BLOCO E APT 401	-	david@apex.com
	6	Eusebio dos Santos Aguiar	15/05/85	M	SHCES QUADRA 1405 BLOCO H APARTAMENTO 304	-	eusebio@apex.com
	7	Fabrice Stephane Alves Pereira	31/10/88	M	QI 16 BLOCO O APT. 301	-	fabrice@apex.com
	8	Fernanda Maria Rodrigues	24/06/75	M	SQSW 504 BLOCO H APARTAMENTO 209	-	fernanda@apex.com
	9	Joao Paulo Barros da Silva	26/12/78	M	QD.104 LT.08 BLA APT0 301	-	joao@apex.com
	10	Tania Maria Pereira	11/09/84	M	UHS 700 BLOCO C CASA 31 ASA SU	-	tania@apex.com

Agora verificaremos os dados criados na tabela **PROFESSOR**.

Browser de Objetos

EsquemaAPEX_CURSO

Tabelas

Q

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

PROFESSOR

+

Tabela

Dados

Índices

Modelo

Constraints

Concessões

Estatísticas

Padrões da UI

Triggers

Dependências

SQL

Consulta

Contar Linhas

Inserir Linha

Dados

EDITAR	COD	NOME	IND_NIVEL	CURRICULO	TELEFONE	EMAIL
	1	Alexandre Domingues Campos	P	Pós graduado em Engenharia de Computação	61 99999-9999	alexandre@apex.com
	2	Fernando Edmundo Chermont Vidal	M	Experiência em programação competitiva. Possui diversas certificações profissionais	61 98888-8888	fernando@apex.com
	3	Jose Augusto Abreu Sa Fortes	S	Bacharel em Ciência da Computação	61 97777-7777	jose@apex.com

Fazer Download

linha(s) 1 - 3 de 3

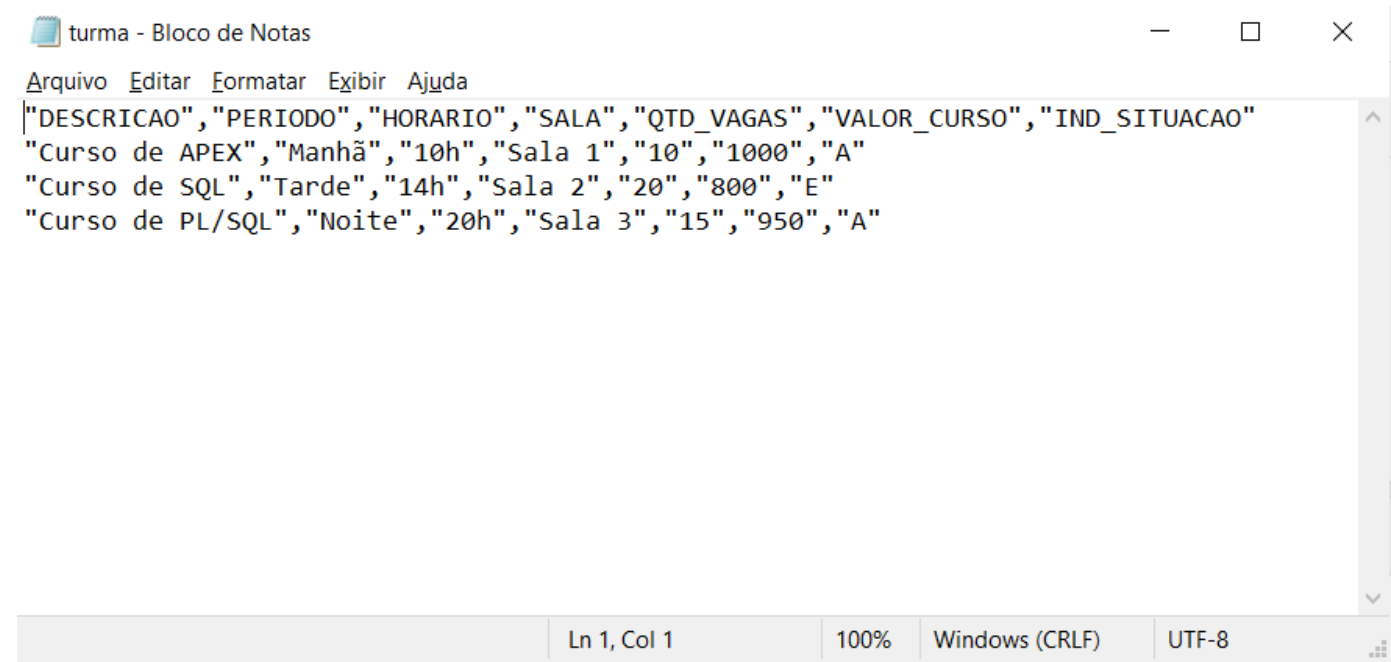
6.2. Carregando dados pelo Data Workshop

Nesta seção veremos um exemplo de como é possível inserir dados em uma tabela por meio do Data Workshop.

Agora é com você!













Passo 1 - Baixe para seu computador o arquivo contendo dados da tabela TURMA [clikando aqui](#).

Abra o arquivo e verifique se foi baixado corretamente.



Passo 2 - Acesse **SQL Workshop**, depois em **Utilitários**, escolha **Data Workshop**.

Utilitários

Utilitários		Criar Objeto
 Data Workshop Carregar e descarregar dados usando arquivos de texto, XML ou uma planilha.	 Construtor de Consultas Criar consultas graficamente, adicionando tabelas em um painel e selecionando colunas para retorno.	Tabela >
 SQL Rápida Generate SQL using shorthand syntax	 Conjuntos de Dados de Amostra Install, refresh, or remove sample datasets.	Exibir >
 Gerar DDL Gerar scripts para todos os objetos do banco de dados ou para os selecionados em um esquema.	 Padrões da Interface do Usuário Especificar propriedades de layout para gerar itens/colunas consistentemente em páginas e aplicativos.	Índice >
 Comparação de Esquemas Mostrar diferenças entre os objetos do banco de dados em dois esquemas diferentes.	 Métodos em Tabelas Gerar scripts de API para operações DML nas tabelas especificadas.	Sequência >
 Lixeira Restaurar objetos eliminados do banco de dados.	 Relatórios de Objetos Acessar vários relatórios em tabelas, exceções, segurança, objetos e código PL/SQL.	Tipo >
 Sobre o BD Revisar detalhes do banco de dados. Observação: uma conta do banco de dados que tem a atribuição de DBA é obrigatória.	 Monitor de Banco de Dados Executar relatórios de atividade do banco de dados. Observação: uma conta do banco de dados que tem a atribuição de DBA é obrigatória.	Pacote >
		Procedimento >
		Função >
		Trigger >
		Link de Banco de Dados >
		View Materializada >
		Sinônimo >

Passo 3 - Como o arquivo com os dados é um arquivo de texto, selecione **Dados de Texto**.

Utilitários \ Data Workshop

Carga de Dados	Descarga de Dados	Repositório
 Dados de Texto	 para Texto	 Repositório de Importações
Dados XML	para XML	Importações de Planilhas
Dados de Planilha		

Passo 4 - Selecione Carregar Para: "**Tabela existente**" e Carregar de: "**Arquivo obtido por upload (separados por vírgula ou delimitado por tabulação)**" e pressione o botão "**Próximo>**".

Carregar Dados - Destino e Método

Destino e Método

Carregar Para:

Tabela existente

?

Nova tabela

Carregar de:

Arquivo obtido por upload (separado por vírgula ou delimitado por tabulação)

?

Copiar e colar

Cancelar

Próximo >

Passo 5 - Em Nome da Tabela selecione a tabela **TURMA** e clique em "**Próximo>**".

Carregar Dados - Proprietário e Nome da Tabela

Proprietário e Nome da Tabela

Selecione o esquema de banco de dados e o nome da tabela em que você gostaria de carregar dados.

*

Proprietário da Tabela

APEX_CURSO

▼

?

*

Nome da Tabela

TURMA

▼

?

<

Cancelar

Próximo >

Passo 6 - Em Arquivo, selecione o arquivo contendo os dados da turma baixados no Passo 1, em **Separador**, mantenha digitado a vírgula "," e em **Opcionalmente Incluído Por**, digite aspas duplas, pois os dados no arquivo estão encapsulados por aspas duplas, certifique-se de que está marcada a opção: "**A primeira linha contém nomes de colunas**" e clique em "**Próximo>**"

Carregar Dados - Detalhes dos Arquivos

Detalhes do Arquivo

Use essa página para localizar o arquivo que será submetido a upload. Se a primeira linha contiver nomes de colunas, selecione **A primeira linha contém nomes de coluna.**

* Arquivo

Escolher arquivo

turma.txt

Visualizar

DESCRICAO	PERIODO	HORARIO	SALA	QTD_VAGAS	VALOR_CURSO	IND_SITUACAO
Curso de APEX	Manhã	10h	Sala 1	10	1000	A
Curso de SQL	Tarde	14h	Sala 2	20	800	E
Curso de PL/SQL	Noite	20h	Sala 3	15	950	A

* Separador

,

Opcionalmente Incluído Por

"

☒ A primeira linha contém nomes de colunas.

Conjunto de Caracteres do Arquivo

Unicode UTF-8

> Globalização

<

Cancelar

Próximo >

Passo 7 - Confira se o mapeamento de colunas foi realizado corretamente na tabela TURMA e clique em "Carregar Dados".

Carregar Dados - Mapeamento de Coluna

Mapeamento de Coluna

Esta página mostra uma prévia de como seus dados serão carregados. Os nomes de colunas do banco de dados devem corresponder às colunas nos dados. Para fazer upload de colunas, selecione **Sim** ou **Não**. Um asterísco (*) indica uma coluna obrigatória. Para fazer upload de dados para a tabela selecionada, clique em **Carregar Dados**.

Esquema: APEX_CURSO

Nome da Tabela: TURMA

Nomes de Colunas	DESCRICAO - varchar2(50)	PERIODO - varchar2(50) *	HORARIO - varchar2(50) *	SALA - varchar2(20)	QTD_VAGAS - number *	VALOR_CURSO - number *	IND_SITUACAO - char *
Formato							
Fazer Upload	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Linha 1	Curso de APEX	Manhã	10h	Sala 1	10	1000	A
Linha 2	Curso de SQL	Tarde	14h	Sala 2	20	800	E
Linha 3	Curso de PL/SQL	Noite	20h	Sala 3	15	950	A

<

Cancelar

Carregar Dados

Passo 8 - Pronto! Você receberá uma mensagem de sucesso.

Utilitários \ Data Workshop \ Repositório de Cargas de Dados de Texto

Dados Carregados.

Mostrar

Meus Arquivos de Importação

Definir

Excluir Marcado(s)

	Detalhes	Arquivo	Importado Por	Importado Em	Tipo	Esquema	Tabela	Bytes	Bem-sucedido	Falhou	Para ser excluído
<input type="checkbox"/>		turma.txt	ANDERSONRF	1 segundos atrás	Importação de Texto	APEX_CURSO	TURMA	249	3	0	30/01/21

1-1

Repositório

Esta página exibe o status dos dados de Texto carregados.

Para acessar o arquivo carregado, selecione seu nome.

Para remover uma entrada de status selecione-a e clique em **Excluir Marcado(s)**.

Tarefas

Repositório de Importações de Planilhas

É importante conferir se os dados foram realmente carregados na tabela TURMA. Para isso acesse no Browser de Objetos na aba Dados e veja as informações que constam na tabela TURMA.

PRODUCAO

App Builder

SQL Workshop

Desenvolvimento da Equipe

Galeria de Aplicativos

Browser de Objetos

EsquemaAPEX_CURSO

Tabelas

ALUNO

ALUNO_TURMA

PAGAMENTO

PROFESSOR

PROFESSOR_TURMA

TURMA

TURMA

TabelaDadosÍndicesModeloConstraintsConcessõesEstatísticasPadrões da UITriggersDependênciasSQL

ConsultaContar LinhasInserir Linha

Dados

EDITAR	COD	DESCRICAO	PERIODO	HORARIO	SALA	QTD_VAGAS	VALOR_CURSO	IND_SITUACAO
	1	Curso de APEX	Manhã	10h	Sala 1	10	1000	A
	2	Curso de SQL	Tarde	14h	Sala 2	20	800	E
	3	Curso de PL/SQL	Noite	20h	Sala 3	15	950	A

Fazer Download

linha(s) 1 - 3 de 3

7. Resumo

Vimos neste módulo:

- Sobre o que se trata o sistema Minha Escola;
- Como se faz um Modelo Entidade Relacionamento (MER);
- Os principais tipos de dados e objetos do banco de dados Oracle;
- Como criar tabelas, relacionamentos, sequências, regras de validação de banco no **SQL Workshop**;
- Como checar se o nosso modelo está com alguma falha por meio dos **Relatórios de Exceções**;
- Como carregar dados utilizando **Scripts SQL** e **Data Workshop**.