



# **Faculdade de Ciências e Tecnologias Universidade Nova de Lisboa**

## **Base de Dados**

### **Autores:**

Henrique Garcês nº 34436

Ricardo Pinto nº 37716

Rui Queirós nº 34468

**Docente:** Profº Joaquim Silva

**Turno:** P6

# Índice

Introdução .....	2
Diagrama de Entidades-Relações .....	3
Capítulo 1 - Comentários às entidades e relações .....	4
Capítulo 2 – Opções de <i>design</i> .....	7
2.1 Decisões tomadas .....	7
2.2 Alterações à versão preliminar .....	7
Capítulo 3 - Limitações .....	8
Capítulo 4 – Aplicação .....	9
4.1 Interface .....	9
4.2 Página Inicial .....	9
4.3 Listagem de dados .....	9
4.4 Valores Derivados .....	9
4.5 Inserção, remoção e atualização de tuplos .....	9
4.6 Possibilidade de preencher valores de atributos correspondentes a relações sem se ter conhecimento de códigos e/ou restringindo o domínio dos valores seleccionados .....	10
4.7 Links de Navegação ( <i>breadcrumbs</i> ) .....	10
4.8 <i>Reports</i> Interligados .....	10
4.9 Detalhes Condicionais .....	10
4.10 <i>Master-Detail Form</i> .....	10
Capítulo 5 – Manual do Utilizador .....	11
5.1 Início .....	11
5.2 Eventos/Master Detail Form .....	11
5.3 Criar, editar e eliminar entradas .....	13
5.4 Obter mais informações sobre uma entrada .....	16
5.5 Restrições .....	17
Capítulo 6 - Código SQL .....	18
6.1 Criação das tabelas .....	18
6.2 Vistas .....	21
6.3 <i>Triggers</i> .....	22
6.4 Funções .....	28

# Introdução

O nosso trabalho tem como objetivo a gestão de concertos e festivais de uma promotora de música.

Nesta base de dados a promotora pretende guardar os vários eventos que vai promover e ainda os gastos que irá ter, bem como as receitas resultantes de publicidades, apoios e vendas de bilhetes feitas por outras empresas.

Cada concerto vai ser realizado por uma ou mais bandas ou artistas, numa dada hora, num dado local e com os preços variante. Não é possível existir mais do que um concerto da mesma banda no mesmo dia, nem mais do que um concerto no mesmo local no mesmo dia. Os preços dos bilhetes variam de concerto para concerto.

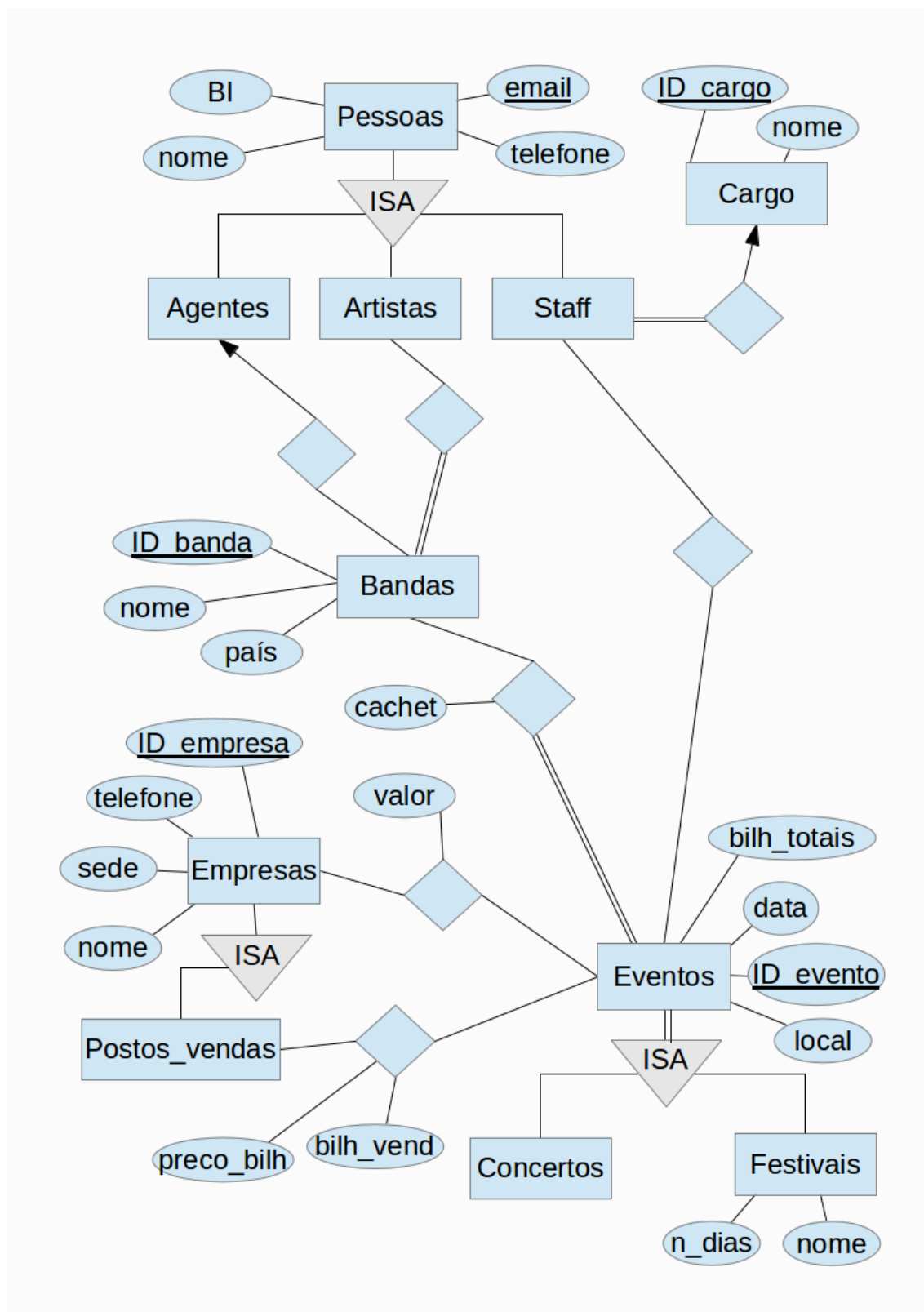
A promotora também promove festivais. Cada festival decorre num dado lugar com um dado número de dias a uma dada hora e pode ter várias bandas no cartaz a atuar. Assume-se que o festival tem apenas um palco e portanto não pode haver bandas a atuar à mesma hora. No caso dos festivais, os bilhetes têm preço unitário, mas podem ser comprados para um dia só, ou um passe para os dias todos.

Em ambos os casos apenas fica registado o número de bilhetes vendidos e o valor acumulado, não se podendo vender mais bilhetes do que a lotação máxima.

As bandas estão identificadas pelo nome, pelo país de origem, pelo nome do agente, pelo contacto telefónico do agente e e-mail do agente.

Na nossa base de dados existem ainda pessoas que assumem funções na promotora, outras que constituem as bandas e ainda outras que representam os agentes das bandas.

## Diagrama de Entidades-Relações



# Capítulo 1 - Comentários às entidades e relações

## ***1.1 Entidade pessoa***

A entidade *pessoa* representa qualquer tipo de pessoa cujos dados deverão ser registados na base de dados. Esta entidade possui como atributos o número do BI, o e-mail, que é a sua chave primária, o nome e telefone.

## ***1.2 Entidade artistas***

A entidade *artistas* é uma especialização da entidade *pessoa*, e representa os artistas que formam uma banda. A sua chave primária é o e-mail.

## ***1.3 Entidade staff***

A entidade *staff* é uma especialização da entidade *pessoa*, e representa alguém que trabalha na promotora. A sua chave primária é o e-mail. Cada empregado possui um cargo.

## ***1.4 Entidade agentes***

A entidade agente é uma especialização da entidade *pessoa*. A sua chave primária é o e-mail. Um agente trata e gere os compromissos de uma banda. No entanto, algumas bandas podem não ter um agente.

## ***1.5 Entidade cargo***

A entidade *cargo* representa os cargos do staff da gestora de concertos cujos dados deverão ser registados na base de dados. Esta entidade possui como atributos um identificador, que é a sua chave primária, e o nome do cargo.

## ***1.6 Entidade bandas***

A entidade *bandas* é uma entidade que representa um artista ou um conjunto de artistas. Esta entidade possui os atributos ID único que serve de chave primária para identificar a banda, país de origem, nome da banda.

## ***1.7 Entidade eventos***

A entidade *eventos* é uma entidade que representa os vários eventos organizados pela promotora e que serão registados na base de dados. Esta entidade possui os atributos ID, que serve de chave primária para identificar o evento, data e local.

### ***1.8 Entidade concertos***

A entidade *concertos* é uma especialização da entidade *eventos*. Esta entidade é identificada pelo ID único que serve de chave primária, referenciando o atributo ID dos concertos.

### ***1.9 Entidade festivais***

A entidade *festivais* é uma especialização da entidade *eventos*. Esta entidade é identificada pelo ID, que serve de chave primária, contém ainda um atributo nome, e número de dias que indica em quantos dias pode um festival ocorrer.

### ***1.10 Entidade empresas***

A entidade *empresas* é uma entidade que representa as empresas que fornecem qualquer apoio a um evento. Esta entidade é identificada pelo ID único que serve de chave primária e contém ainda os atributos nome da empresa, sede e telefone.

### ***1.11 Entidade Postos\_vendas***

A entidade *Postos\_vendas* é uma especialização da entidade *empresas* e representa uma empresa que trata da venda de bilhetes de um evento. Esta entidade é identificada pelo ID que serve de chave primária, referenciando o atributo ID da entidade *empresas*.

### ***1.12 Relação Evento\_PostoDeVenda***

A relação entre os eventos e os postos de venda contam com vários atributos nomeadamente o atributo para o número de bilhetes vendidos, preço dos bilhetes e número total de bilhetes disponíveis.

### ***1.13 Relação Empresa\_Evento***

As empresas podem fornecer um valor monetário a troco de publicidade nos eventos, portanto esta relação tem um atributo valor que representa aquilo que a empresa paga ao evento.

### ***1.14 Relação Banda\_Evento***

A relação *Banda\_Evento* conta também com um atributo, o cachet, que é o valor que uma banda cobra para atuar num evento.

### ***1.15 Relação Evento\_Staff***

Cada evento tem sempre um staff associado.

### ***1.16 Relação Bandas\_Artistas***

Todas as bandas têm de ter obrigatoriamente artistas, daí a relação ser total do lado das bandas. Um artista pode fazer parte de mais que uma banda (ou fazer parte de uma e ter também uma carreira a solo).

7

### ***1.17 Relação Bandas\_Agentes***

Cada banda pode ter no máximo um agente, no entanto nem todas as bandas optam por contratar um. Os agentes podem gerir mais que uma banda.

### ***1.18 Relação Staff\_Cargo***

Cada elemento do staff tem um único cargo dentro da promotora.

## Capítulo 2 – Opções de *design*

### 2.1 Decisões tomadas

Durante a concepção do diagrama ER, tivemos de decidir entre várias opções de *design*. Em primeiro lugar, considerámos ter uma agregação entre bandas e agentes para estarem relacionadas com um evento. No entanto, isso não faria sentido, uma vez que certas bandas podem não ter agente e o que interessa estar relacionado com o evento é a própria banda.

Outra dúvida que surgiu foi a opção de ter o cargo de um elemento do staff representado como atributo dessa entidade ou como uma entidade que estaria relacionada com o staff. Escolhemos a segunda, porque impede que elementos com o mesmo cargo tivessem nomes de cargo diferentes ao serem inseridos na base de dados, o que lhe retiraria alguma consistência. Ter o cargo como entidade dá logo uma ideia exacta de quais os cargos que pode haver na promotora.

A certa altura, achámos que certas entidades que o diagrama continha pareciam demasiado forçadas ou não faziam muito sentido. Foi o caso das receitas de um evento. Em vez de criar uma entidade para as representar, decidimos atribuir um valor à relação entre empresas e eventos (publicidade e apoios) e outros atributos na relação entre eventos e postos de venda que representam os bilhetes vendidos, eliminando logo a ideia de ter uma entidade *bilhetes*. Para obter a receita de um evento bastará calcular a soma dos apoios e publicidades com os bilhetes vendidos em função do preço destes. Contudo há que subtrair a essa quantia o cachet a pagar às bandas que atuam.

### 2.2 Alterações à versão preliminar

Após a entrega de versão preliminar do projeto, decidimos fazer algumas alterações no *design* da base de dados que nos pareceram necessárias à medida que implementávamos a aplicação.

- Eventos passou a ser uma generalização total e disjunta, uma vez que um evento forçosamente ou é um concerto ou é um festival, não podendo ser ambos ou nenhum.
- Bilhetes totais passou a ser um atributo dos eventos.



## Capítulo 3 - Limitações

As funções para derivação de valores não estão 100% funcionais. Por exemplo, para o caso em que no cálculo do resultado haja algum valor nulo, o resultado final será nulo o que não deveria acontecer. Infelizmente, dado que estas funções foram os últimos aspectos que implementámos, não tivemos tempo para pensar em restrições de integridade que evitassem estes casos, sem comprometer outra qualquer característica da base de dados.

Também no caso de existirem postos de venda diferentes a vender bilhetes para o evento e dá um erro por retornar mais linhas do que aquelas que aquelas que eram esperadas.

Por último, quando a base de dados está vazia, falta um botão nos eventos para inserir eventos. Tem que ser inserida manualmente.

# Capítulo 4 – Aplicação

## 4.1 Interface

A interface desenvolvida foi construída no sistema APEX. Aqui desenvolvemos várias páginas que permitem consultar todos os elementos da base de dados como informações sobre os eventos, bandas e pessoas que fazem parte das bandas, que trabalham nos eventos ou agentes das bandas.

É ainda possível comprar bilhetes nos postos de venda que são representados por empresas, sendo também possível consultar todas as empresas criadas.

Para além de todas estas consultas também é possível criar novos tuplos. Apesar de alguns erros, todos os requisitos foram implementados. As restantes secções deste capítulo são dedicadas a descrever essas funcionalidades pedidas.

## 4.2 Página Inicial

Esta página é automaticamente apresentada quando se inicia a aplicação, logo após se efectuar o *login*. A partir daqui são disponibilizados links de acesso a todas as tabelas da base de dados.

## 4.3 Listagem de dados

Pode-se aceder às várias listagens de dados a partir da página inicial da aplicação, uma vez que todas as páginas que lá aparecem contêm uma listagem de dados. Nessas listagens são omitidos os atributos que dificultariam a compreensão do utilizador, mas que interessam para o estado da base de dados (por exemplo, as ID das entidades).

## 4.4 Valores Derivados

Os valores derivados podem ser encontrados em três locais distintos. Dentro dos festivais podemos encontrar um valor derivado que é a data de fim do festival calculado através do número de dias do festival e da data do seu primeiro dia. Ou então podemos encontrar dentro dos eventos os valores de receitas calculado através da venda de bilhetes, do *cachet* que as bandas pedem para um dado evento e valores pagos por empresas e ainda bilhetes disponíveis que é a diferença entre os bilhetes totais de um evento e os bilhetes vendidos.

## 4.5 Inserção, remoção e atualização de tuplos

É possível inserir vários tuplos, nomeadamente novas pessoas que poderão ser agentes, staff da promotora ou artistas. É possível adicionar novas bandas, novos eventos (festivais ou concertos) e novas empresas.

Por exemplo dentro de staff é possível criar uma nova pessoa para o staff através do botão "Create" ou então através do botão "ADD" associar uma dessas pessoas criadas a um evento. Para apagar/editar entramos no ícone que representa o lápis.

## **4.6 Possibilidade de preencher valores de atributos correspondentes a relações sem se ter conhecimento de códigos e/ou restringindo o domínio dos valores seleccionados**

Na criação de um artista há um pop-LOV que nos permite seleccionar uma banda a que o artista pertence ficando logo associado à mesma. Do mesmo modo se pode associar uma banda a um agente que figure na LOV.

## **4.7 Links de Navegação (*breadcrumbs*)**

Para além da página inicial, podemos encontrar exemplos desta funcionalidade nas listagens das bandas, dos postos de venda e das empresas. Para todas estas listagens, há *breadcrumbs* nos nomes das entidades que redireccionam o utilizador para uma página contendo a informação sobre uma banda, posto de venda ou empresa, respectivamente.

## **4.8 Reports Interligados**

Analogamente aos links de navegação e associada a estes, esta funcionalidade também está presente nas listagens de bandas, postos de venda e empresas. Para o primeiro caso, o acesso ao nome de uma banda, mostra um *report* dos elementos dessa banda. No caso seguinte, um clique no nome de um posto exposto na tabela apresenta ao utilizador um *report* dos eventos para os quais o posto de venda disponibiliza bilhetes. De forma semelhante, aceder à informação de uma empresa via o nome na respectiva tabela, mostra todos os eventos para os quais essa empresa oferece apoio monetário.

## **4.9 Detalhes Condicionais**

Dentro da página dos artistas onde são listados todos os artistas que existem na base de dados, podemos listá-los pela banda a que pertencem. Outra amostra desta funcionalidade encontra-se na página do staff, na qual podemos listar os empregados consoante o cargo que ocupam na empresa.

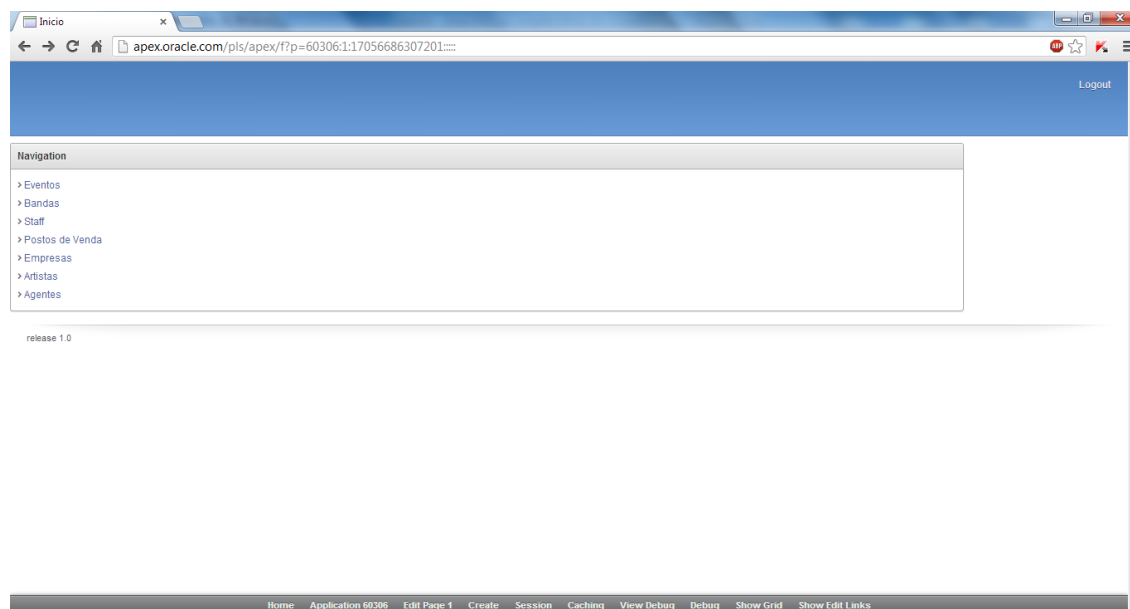
## **4.10 Master-Detail Form**

Na página dos eventos também podemos aceder ao *Master Detail* que nos permite adicionar bandas a um evento ou atualizar informações sobre o mesmo. Para isso carrega-se no ícone do lápis com uma folha antes do nome do evento para acedermos ao seu formulário.

# Capítulo 5 – Manual do Utilizador

## 5.1 Início

Ao executar a aplicação, o utilizador deve iniciar sessão com as suas credenciais (nome de utilizador e *password*). Após estas terem sido confirmadas, o utilizador é redireccionado para a página inicial da aplicação, que contém links de navegação para os diversos relatórios da base de dados.



Página inicial.

## 5.2 Eventos/Master Detail Form

Na página de eventos, temos duas tabelas, uma para festivais e outra para concertos. São dadas ao utilizador as opções de ordenar a listagem e editar ou remover um evento. Para ordenar consoante um atributo, basta clicar no nome desse conjunto de atributos. Por defeito, a ordenação é ascendente, mas um novo clique torna-a descendente.

Evento

Inicio > Eventos

Concertos

ID_EVENTO	NOMEFEST	LOCAL	DATA	NUMBILHETES	Bilhetes Disponíveis	Receitas
	Coldplay Pavilhão Atlântico	Pavilhão Atlântico	12-MAY-13	2000	1950	3750
	y no Coliseu dos Recreios	Coliseu dos Recreios	01-APR-13	2000	-	-
	z no Coliseu do Porto	Coliseu do Porto	03-FEB-13	2000	-	-
	k no Campo Pequeno	Campo Pequeno	06-JAN-13	2000	-	-

Festivais

NOMEFEST	LOCAL	DATA	NUMDIAS	NUMBILHETES	Bilhetesdisponiveis(Id Evento)
	Super Bock Super Rock	Praia do Meo	10-JUL-13	3	2000
	Sudoeste TMN	Herderdade da Pomba	10-JUL-13	5	2000
	Rock in Rio	Parque da Bela Vista	30-MAY-12	5	2000
	Portugal Ao Vivo	Palácio de Cristal	02-MAR-13	2	2000
	Pimbamisto	Quinta Rende o Peixe	14-FEB-13	1	2000
	Optimus Alive	Passeio Marítimo de Algés	07-MAR-13	3	2000

release 1.0 Home Application 60306 Edit Page 34 Create Session Caching View Debug Debug Show Grid Show Edit Links

Página de eventos, com concertos ordenados por ordem alfabética descendente do local e festivais ordenados por ordem alfabética ascendente do nome.

Vejamos agora como editar ou remover um evento.

Tanto para festivais como para concertos, se clicarmos no ícone cinzento com um lápis disposto à esquerda de cada entrada, somos redireccionados para um formulário *Master Detail*.

Neste formulário, podemos alterar alguns campos do evento, sendo que para que estas alterações surtam efeito é preciso clicar em “Apply Changes”, no topo direito do formulário. Junto a esta opção, estão as de “Delete”, que elimina esta entrada e outras duas em forma de setas (“<”, “>”) que nos redireccionam para a entrada anterior ou posterior à actual, respectivamente.

Evento

Inicio > Eventos > Master Details Festivais

Edit\_V\_FESTIVAIS

Nome do Festival \* Sudoeste TMN

Local \* Herdade da Pomba

Data \* 10-JUL-13

Número de Dias \* 5

Último dia do Festival 14-JUL-13

Número de Bilhetes \* 2000

Cancel Delete Apply Changes < >

V\_BANDAS\_EVENTOS Detail

Delete Checked Add Row

Id Banda	Cachet
<input type="checkbox"/> MUSE	1200
<input checked="" type="checkbox"/> Metallica	1000

1 - 2

release 1.0 Home Application 60306 Edit Page 49 Create Session Caching View Debug Debug Show Grid Show Edit Links

Master Detail de um festival. Na parte inferior, temos a entrada da banda “Metallica” seleccionada para ser eliminada.

Por baixo do formulário para editar, temos outro para adicionar bandas a um evento. O formulário apresenta todas as bandas que irão tocar nesse evento. Para adicionar uma

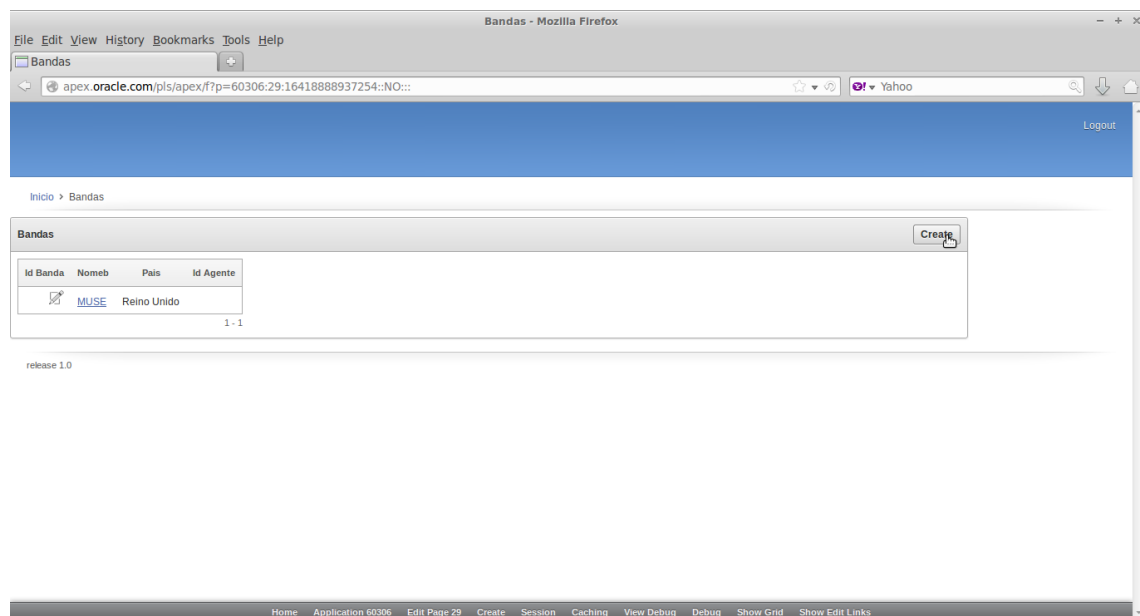
banda clica-se em “Add Row” no canto superior direito, o que faz aparecer uma nova entrada onde se pode seleccionar a banda a partir de uma lista de valores e atribuir-lhe um *cachet*. Ao lado de cada entrada também podemos observar um pequeno quadrado. Esta característica serve para seleccionar as bandas que queiramos eliminar do evento. Após estas terem sido seleccionadas, é preciso clicar em “Delete checked” para que a alteração se realize. Se seleccionarmos o primeiro quadrado (ao lado de “Id Banda”), todas as bandas são automaticamente seleccionadas para eliminação.

## 5.3 Criar, editar e eliminar entradas

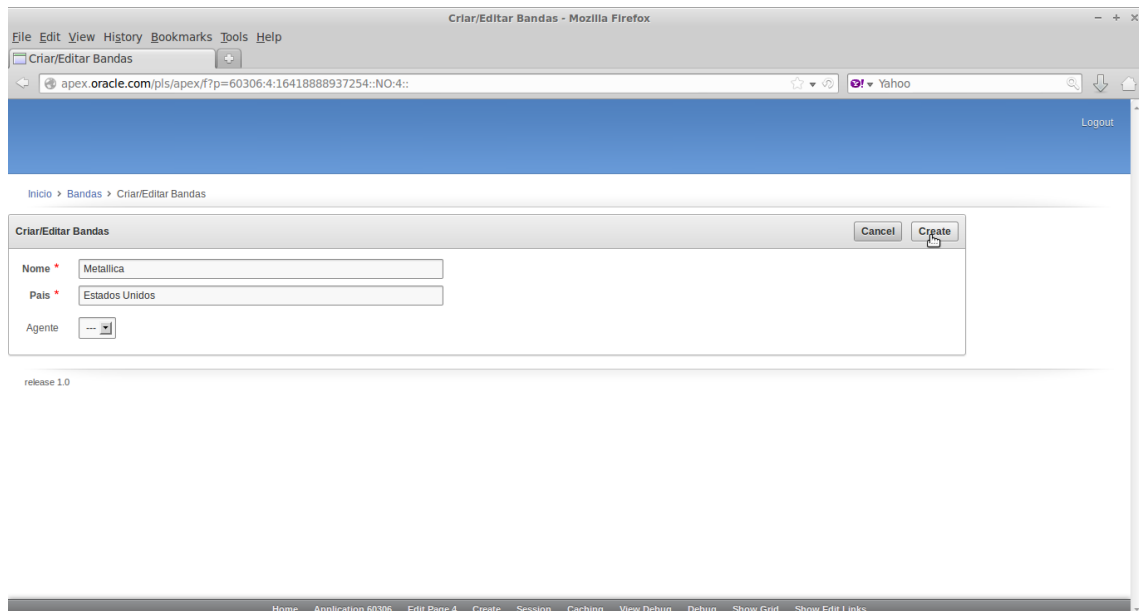
Para descrever esta secção do manual, vamos recorrer à página das bandas como exemplo, sendo que para staff, postos de venda, empresas, artistas e agentes, as funcionalidades aqui descritas funcionam de forma análoga.

Na página das bandas, temos as opções de criar e editar uma banda, recorrendo ao link “Create” ou ao ícone cinzento à esquerda de cada entrada. Para ambas as opções podemos atribuir um nome e um país bem como um agente, a partir de uma lista de valores, contendo todos os agentes registados na base de dados. Para aplicar estas modificações, é preciso clicar em “Create”, se estivermos a criar uma banda, ou “Apply Changes”, se a estivermos a editar.

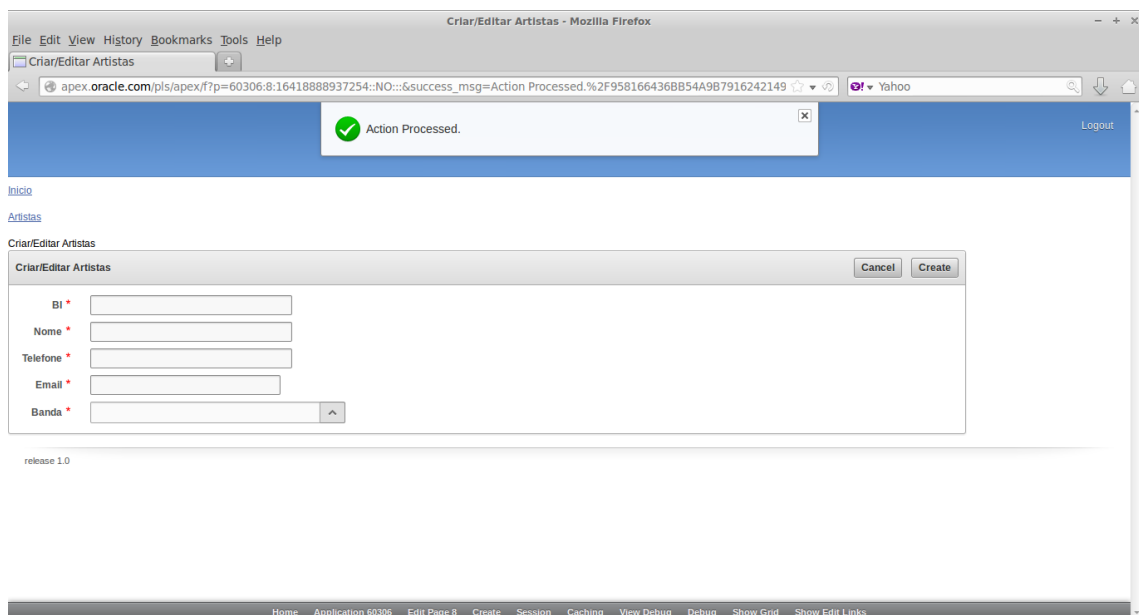
Se quisermos eliminar uma banda, seleccionamos o ícone de edição na entrada dessa banda e clicamos em “Delete”.



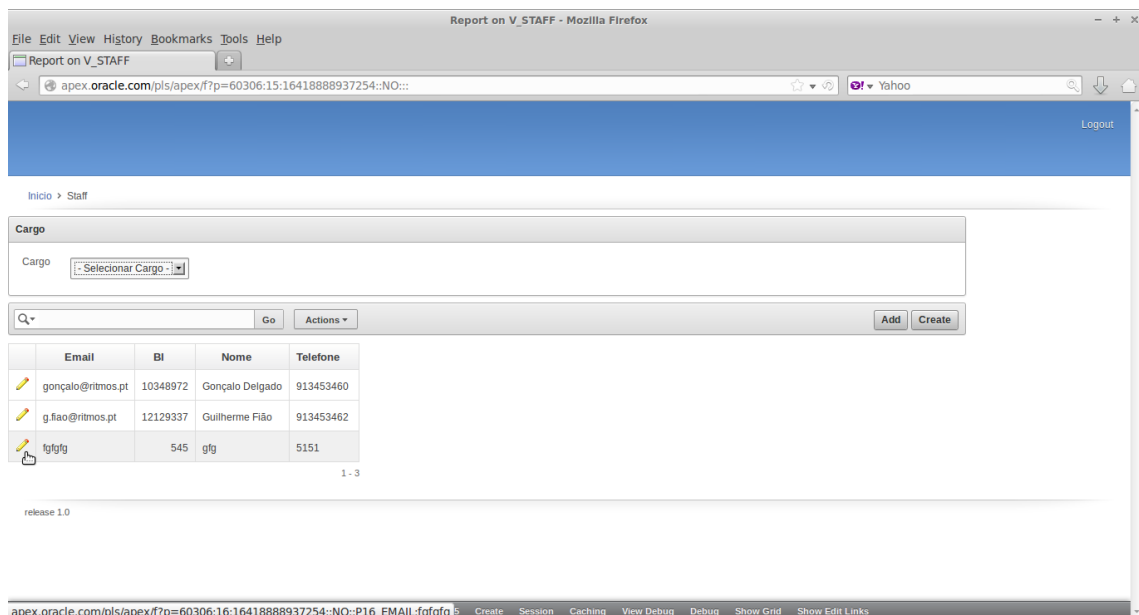
Criar Banda (1): Clicamos em “Create”, na página de bandas.



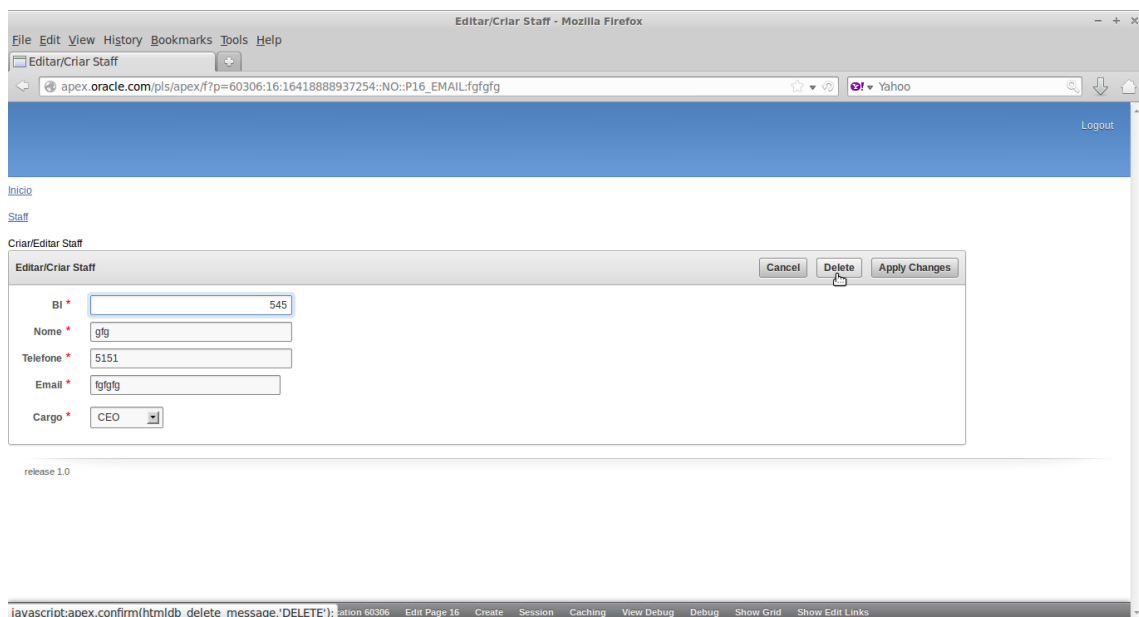
Criar Banda (2): Preenchemos os campos e clicamos novamente em “Create”. Se quisermos cancelar esta ação e voltar à página das bandas, clicamos então em “Cancel”.



Criar Banda (3): Se a ação for executada com sucesso, aparecerá a mensagem “Action processed” no topo da página.

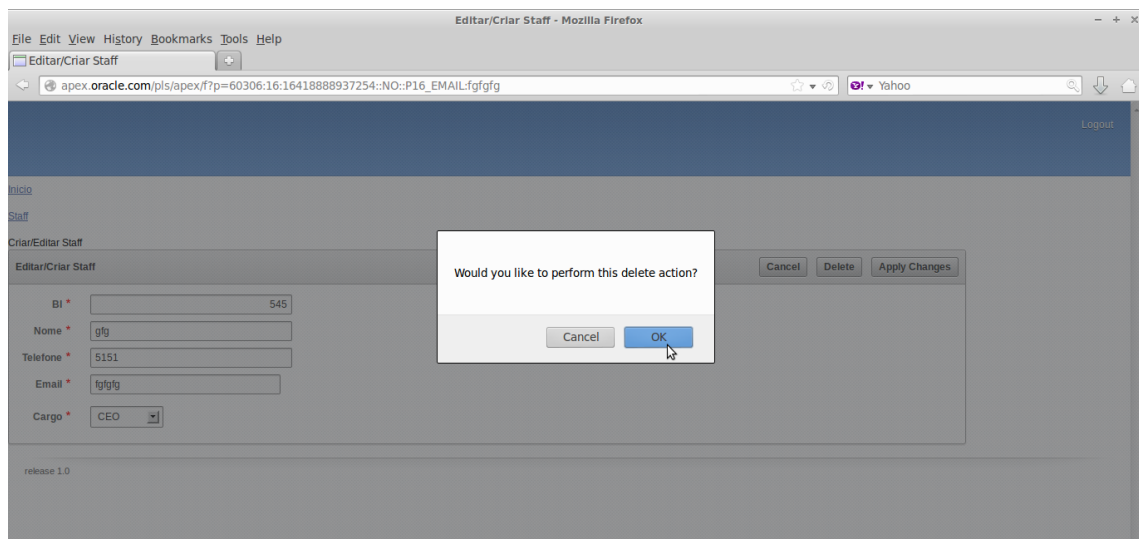


Eliminar Staff (1): Seleccionamos o ícone em forma de lápis à esquerda da entrada.

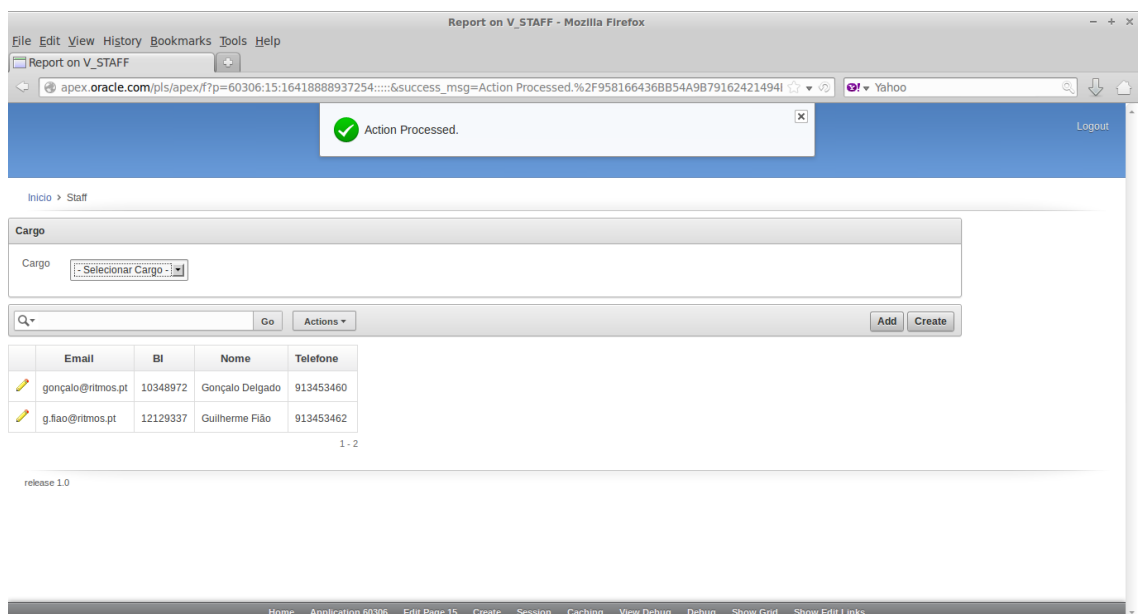


Eliminar Staff (2): Seleccionamos "Delete".





Eliminar Staff (3): Clicamos em OK para confirmar a opção.



Eliminar Staff (4): A mensagem “Action processed” aparecerá no topo da página.

## 5.4 Obter mais informações sobre uma entrada

Para obter mais informações sobre uma banda, basta clicar no seu nome e seremos reencaminhados para a página de artistas, sendo que todos os artistas listados fazem parte da banda seleccionada.

As páginas de postos de venda e empresas oferecem a mesma funcionalidade. Clicar no nome de um posto de venda, redirecciona o utilizador para uma página que contém uma listagem dos eventos para os quais esse posto vende bilhetes. No caso das empresas, o utilizador é redireccionado para uma página com a listagem de eventos para os quais, a empresa oferece apoio monetário.

## 5.5 Restrições

O utilizador também tem que ter em atenção que nem sempre as acções de alteração ou criação de tuplos são realizadas com sucesso, caso estas violem restrições de integridade da base de dados. Por exemplo, se não introduzirmos nenhum carácter no nome de um evento, banda ou artista, a acção não será executada e o sistema informará o utilizador do erro ocorrido, que neste caso é o facto de o nome não poder ser obrigatoriamente vazio.

Na edição e criação de entradas, os campos sensíveis são destacados com um asterisco vermelho.

Master Detail

apex.oracle.com/pls/apex/www\_flow.accept

1 error has occurred

- Nome must have some value. (Go to error)

Logout

Início > Eventos > Master Detail Concertos

Edit V\_CONCERTOS

Cancel Delete Apply Changes >

Nome \*  
Nome must have some value.

Local \* Pavilhão Atlântico

Data \* 12-MAY-13

Número de bilhetes \* 2000

1 of 4

V\_BANDAS\_EVENTOS Detail

Delete Checked Add Row

Id Banda	Cachet
Coldplay	3000

release 1.0

Home Application 60306 Edit Page 44 Create Session Caching View Debug Debug Show Grid Show Edit Links

Erro (1): A introdução de um concerto com nome vazio não é aceite.

Master Detail

apex.oracle.com/pls/apex/www\_flow.accept

1 error has occurred

- Número de bilhetes must be Numeric. (Go to error)

Logout

Início > Eventos > Master Detail Concertos

Edit V\_CONCERTOS

Cancel Delete Apply Changes >

Nome \* Coldplay Pavilhão Atlântico

Local \* Pavilhão Atlântico

Data \* 12-MAY-13

Número de bilhetes \* asfdgasg  
Número de bilhetes must be Numeric.

1 of 4

V\_BANDAS\_EVENTOS Detail

Delete Checked Add Row

Id Banda	Cachet
Coldplay	3000

release 1.0

Home Application 60306 Edit Page 44 Create Session Caching View Debug Debug Show Grid Show Edit Links

Erro (2): O preenchimento do número de bilhetes do evento com um valor que não seja numérico não é aceite.

# Capítulo 6 - Código SQL

## 6.1 Criação das tabelas

```
DROP TABLE eventos CASCADE CONSTRAINTS;
```

```
CREATE TABLE eventos (  
    ID_evento number (8) NOT NULL,  
    nomeFest varchar2 (30) NOT NULL,  
    local varchar2 (40) NOT NULL,  
    data date NOT NULL,  
    numBilhetes number (5) NOT NULL,  
    PRIMARY KEY (ID_evento)
```

```
);
```

```
DROP TABLE concertos CASCADE CONSTRAINTS;
```

```
CREATE TABLE concertos (  
    ID_evento number (8) NOT NULL,  
    PRIMARY KEY (ID_evento),  
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
```

```
CASCADE
```

```
);
```

```
DROP TABLE festivais CASCADE CONSTRAINTS;
```

```
CREATE TABLE festivais (  
    ID_evento number (8) NOT NULL,  
    numDias number (3) NOT NULL,  
    PRIMARY KEY (ID_evento),  
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
```

```
CASCADE
```

```
);
```

```
DROP TABLE empresas CASCADE CONSTRAINTS;
```

```
CREATE TABLE empresas (  
    ID_empresa number (8) NOT NULL,  
    nomeEmp varchar2 (40),  
    sede varchar2 (100),  
    telefone varchar2 (12),  
    PRIMARY KEY (ID_empresa)
```

```
);
```

```
DROP TABLE postos_venda CASCADE CONSTRAINTS;
```

```
CREATE TABLE postos_venda (  
    ID_empresa number (8) NOT NULL,  
    PRIMARY KEY (ID_empresa),  
    FOREIGN KEY (ID_empresa) REFERENCES empresas ON DELETE
```

```
CASCADE
```

```
);
```

```
DROP TABLE cargo CASCADE CONSTRAINTS;
```

```
CREATE TABLE cargo (  
    ID_cargo number (8) NOT NULL,  
    descriçao varchar2 (25) NOT NULL,  
    PRIMARY KEY (ID_cargo)
```

```
);
```

```

DROP TABLE pessoas CASCADE CONSTRAINTS;
CREATE TABLE pessoas (
    nrBI number (10) NOT NULL,
    nomeP varchar2 (40) NOT NULL,
    telefone varchar2 (12) NOT NULL,
    email varchar2 (35) NOT NULL,
    PRIMARY KEY (email)
);

DROP TABLE staff CASCADE CONSTRAINTS;
CREATE TABLE staff (
    email varchar2 (30) NOT NULL,
    ID_cargo number (8) NOT NULL,
    PRIMARY KEY (email),
    FOREIGN KEY (email) REFERENCES pessoas ON DELETE CASCADE,
    FOREIGN KEY (ID_cargo) REFERENCES cargo ON DELETE CASCADE
);

DROP TABLE artistas CASCADE CONSTRAINTS;
CREATE TABLE artistas (
    email varchar2 (30) NOT NULL,
    PRIMARY KEY (email),
    FOREIGN KEY (email) REFERENCES pessoas ON DELETE CASCADE
);

DROP TABLE agentes CASCADE CONSTRAINTS;
CREATE TABLE agentes (
    email varchar2 (30) NOT NULL,
    PRIMARY KEY (email),
    FOREIGN KEY (email) REFERENCES pessoas ON DELETE CASCADE
);

DROP TABLE bandas CASCADE CONSTRAINTS;
CREATE TABLE bandas (
    ID_banda number (8) NOT NULL,
    nomeB varchar2 (40) NOT NULL,
    pais varchar2 (30) NOT NULL,
    ID_agente varchar2 (30),
    PRIMARY KEY (ID_banda),
    FOREIGN KEY (ID_agente) REFERENCES agentes ON DELETE
CASCADE
);

DROP TABLE eventos_empresas CASCADE CONSTRAINTS;
CREATE TABLE eventos_empresas (
    valor number (10) NOT NULL,
    ID_evento number (8) NOT NULL,
    ID_empresa number (8) NOT NULL,
    PRIMARY KEY (ID_evento, ID_empresa),
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
CASCADE,
    FOREIGN KEY (ID_empresa) REFERENCES empresas ON DELETE
CASCADE
);

DROP TABLE bandas_eventos CASCADE CONSTRAINTS;

```

```

CREATE TABLE bandas_eventos (
    cachet number (10) NOT NULL,
    ID_evento number(8) NOT NULL,
    ID_banda number (8) NOT NULL,
    PRIMARY KEY (ID_banda, ID_evento),
    FOREIGN KEY (ID_banda) REFERENCES bandas ON DELETE CASCADE,
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
CASCADE
);

DROP TABLE venda_bilhetes CASCADE CONSTRAINTS;
CREATE TABLE venda_bilhetes (
    precoBilh number (8) NOT NULL,
    bVendidos number (8) NOT NULL,
    ID_evento number (8) NOT NULL,
    ID_empresa number (8) NOT NULL,
    PRIMARY KEY (ID_empresa, ID_evento),
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
CASCADE,
    FOREIGN KEY (ID_empresa) REFERENCES postos_venda ON DELETE
CASCADE
);

DROP TABLE bandas_artistas CASCADE CONSTRAINTS;
CREATE TABLE bandas_artistas (
    ID_banda number (8) NOT NULL,
    email varchar (30) NOT NULL,
    PRIMARY KEY (ID_banda, email),
    FOREIGN KEY (ID_banda) REFERENCES bandas ON DELETE CASCADE,
    FOREIGN KEY (email) REFERENCES artistas ON DELETE CASCADE
);

DROP TABLE staff_eventos CASCADE CONSTRAINTS;
CREATE TABLE staff_eventos (
    ID_evento number(8) NOT NULL,
    ID_staff varchar2 (30) NOT NULL,
    PRIMARY KEY (ID_staff, ID_evento),
    FOREIGN KEY (ID_staff) REFERENCES staff ON DELETE CASCADE,
    FOREIGN KEY (ID_evento) REFERENCES eventos ON DELETE
CASCADE
);

DROP SEQUENCE seq_evento;
CREATE SEQUENCE seq_evento INCREMENT BY 1 START WITH 1 ;
DROP SEQUENCE seq_empresa;
CREATE SEQUENCE seq_empresa INCREMENT BY 1 START WITH 1 ;
DROP SEQUENCE seq_banda;
CREATE SEQUENCE seq_banda INCREMENT BY 1 START WITH 1 ;
DROP SEQUENCE seq_cargo;
CREATE SEQUENCE seq_cargo INCREMENT BY 1 START WITH 1 ;

```

## 6.2 Vistas

```
CREATE OR REPLACE FORCE VIEW "V_AGENTES" ("NRBI", "NOME",  
"TELEFONE", "EMAIL") AS  
    select nrBI, nomeP, telefone, email  
from agentes natural inner join pessoas  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_ARTISTAS" ("NRBI", "NOME",  
"TELEFONE", "EMAIL") AS  
    select nrBI,  
        nomeP,  
        telefone,  
        email  
from artistas natural inner join pessoas  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_ARTISTAS_BANDA" ("NRBI",  
"NOME", "TELEFONE", "EMAIL", "ID_BANDA") AS  
    select p.nrbi, p.nomep, p.telefone, p.email, ba.id_banda  
from artistas a, pessoas p, bandas_artistas ba  
where a.email = p.email and p.email = ba.email  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_BANDAS_EVENTOS" ("PK",  
"ID_BANDA", "CACHET", "ID_EVENTO") AS  
    select ROWID pk, id_banda, cachet, id_evento  
from bandas_eventos  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_CONCERTOS" ("ID_EVENTO",  
"NOMEFEST", "LOCAL", "DATA", "NUMBILHETES") AS  
    select ID_evento,  
        nomefest,  
        local,  
        data,  
        numBilhetes from eventos e  
where e.ID_evento not in (select ID_evento from v_festivais)  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_EMPRESAS" ("ID_EMPRESA",  
"NOMEEMP", "SEDE", "TELEFONE") AS  
    select e.id_empresa, e.nomeemp, e.sede, e.telefone  
from empresas e  
where e.id_empresa not in (select id_empresa from  
v_postos_venda)  
/
```

```
CREATE OR REPLACE FORCE VIEW "V_EVENTOS_EMPRESAS" ("PK",  
"ID_EVENTO", "ID_EMPRESA", "VALOR") AS  
    select ROWID pk, id_evento, id_empresa, valor  
from eventos_empresas  
/
```

```

CREATE OR REPLACE FORCE VIEW "V_FESTIVAIS" ("ID_EVENTO",
"NOFEST", "LOCAL", "DATA", "NUMDIAS", "NUMBILHETES") AS
    select ID_evento,
           nomeFest,
           local,
           data,
           numDias,
           numBilhetes
from eventos natural inner join festivais
/

CREATE OR REPLACE FORCE VIEW "V_VENDA_BILHETES" ("PK",
"ID_EVENTO", "ID_EMPRESA", "PRECIBILH", "BVENDIDOS") AS
    select ROWID pk, id_evento, id_empresa, precobilh, bvendidos
from venda_bilhetes
/

CREATE OR REPLACE FORCE VIEW "V_POSTOS_VENDA" ("ID_EMPRESA",
"NOFEMP", "SEDE", "TELEFONE") AS
    select id_empresa, nomeemp, sede, telefone
from postos_venda natural inner join empresas
/

CREATE OR REPLACE FORCE VIEW "V_STAFF" ("NRBI", "NOME",
"TELEFONE", "EMAIL", "ID_CARGO") AS
    select nrBI, nomeP, telefone, email, id_cargo
from pessoas natural inner join staff
/

CREATE OR REPLACE FORCE VIEW "V_STAFF_EVENTOS" ("PK",
"ID_STAFF", "ID_EVENTO") AS
    select ROWID pk, id_staff, id_evento
from staff_eventos
/

```

## 6.3 Triggers

```

create or replace TRIGGER banda_pk
    BEFORE INSERT ON bandas
    FOR EACH ROW
BEGIN
    SELECT seq_banda.nextval
    INTO :new.ID_banda
    FROM dual;
END;

create or replace TRIGGER cargo_pk
    BEFORE INSERT ON cargo
    FOR EACH ROW
BEGIN
    SELECT seq_cargo.nextval
    INTO :new.ID_cargo
    FROM dual;
END;

create or replace TRIGGER empresa_pk
    BEFORE INSERT ON empresas

```

```

    FOR EACH ROW
BEGIN
    SELECT seq_empresa.nextval
        INTO :new.ID_empresa
        FROM dual;
END;

create or replace TRIGGER eventos_pk
    BEFORE INSERT ON eventos
    FOR EACH ROW
BEGIN
    SELECT seq_evento.nextval
        INTO :new.ID_evento
        FROM dual;
END;

create or replace trigger trg_insert_agentes
instead of insert on v_agentes
for each row
begin
insert into pessoas(nrBI, nomeP, telefone, email)
values(:NEW.nrBI, :NEW.nomeP, :NEW.telefone, :NEW.email);
insert into agentes(email)
values(:NEW.email);
end;

create or replace trigger trg_insert_artistas
instead of insert on v_artistas_banda
for each row
begin
insert into pessoas(nrBI, nomeP, telefone, email)
values(:NEW.nrBI, :NEW.nomeP, :NEW.telefone, :NEW.email);
insert into artistas(email)
values(:NEW.email);
insert into bandas_artistas(ID_banda, email)
    values(:NEW.ID_banda, :NEW.email);
end;

create or replace trigger trg_insert_bandas_eventos
instead of insert on v_bandas_eventos
for each row
begin
insert into bandas_eventos(cachet, id_evento, id_banda)
values(:NEW.cachet, :NEW.id_evento, :NEW.id_banda);
end;

create or replace trigger trg_insert_concerto
instead of insert on v_concertos
for each row
begin
insert into eventos(id_evento, nomefest, local, data,
numbilhetes)
values(:NEW.id_evento, :NEW.nomefest, :NEW.local, :NEW.data,
:NEW.numbilhetes);
insert into concertos(id_evento)
values(:NEW.id_evento);

```



```

end;

create or replace trigger trg_insert_eventos_empresas
instead of insert on v_eventos_empresas
for each row
begin
insert into eventos_empresas(id_evento, id_empresa, valor)
values(:NEW.id_evento, :NEW.id_empresa, :NEW.valor);
end;

create or replace trigger trg_insert_festival
instead of insert on v_festivals
for each row
begin
insert into eventos(id_evento, nomefest, local, data,
numbilhetes)
values(:NEW.id_evento, :NEW.nomefest, :NEW.local, :NEW.data,
:NEW.numbilhetes);
insert into festivais(id_evento, numdias)
values(:NEW.id_evento, :NEW.numdias);
end;

create or replace trigger trg_insert_staff
instead of insert on v_staff
for each row
begin
insert into pessoas(nrBI, nomeP, telefone, email)
values(:NEW.nrBI, :NEW.nomeP, :NEW.telefone, :NEW.email);
insert into staff(email, ID_cargo)
values(:NEW.email, :NEW.ID_cargo);
end;

create or replace trigger trg_insert_staff_eventos
instead of insert on v_staff_eventos
for each row
begin
insert into staff_eventos(id_evento, id_staff)
values(:NEW.id_evento, :NEW.id_staff);
end;

create or replace trigger trg_insert_venda_bilhetes
instead of insert on v_venda_bilhetes
for each row
begin
insert into venda_bilhetes(id_evento, id_empresa, precobilh,
bvendidos)
values(:NEW.id_evento, :NEW.id_empresa, :NEW.precobilh,
:NEW.bvendidos);
end;

create or replace trigger trg_del_agentes
instead of delete on v_agentes
referencing new as new old as old
begin
delete from pessoas
where email=:old.email;

```

```

        delete from agentes
        where email=:old.email;
    end;

create or replace trigger trg_del_artistas
instead of delete on v_artistas_banda
referencing new as new old as old
begin
    delete from pessoas
    where email=:old.email;
    delete from agentes
    where email=:old.email;
end;

create or replace TRIGGER TRG_Del_bandas_eventos
INSTEAD OF DELETE ON v_bandas_eventos
FOR EACH ROW
BEGIN
    DELETE FROM bandas_eventos
    WHERE ROWID = :OLD.pk;
END;

create or replace TRIGGER "TRG_DEL_CONCERTO"
instead of delete on v_concertos
referencing new as new old as old
begin
    delete from eventos
    where id_evento=:old.id_evento;
    delete from concertos
    where id_evento=:old.id_evento;
end;

create or replace TRIGGER TRG_del_eventos_empresas
INSTEAD OF DELETE ON v_eventos_empresas
FOR EACH ROW
BEGIN
    DELETE FROM eventos_empresas
    WHERE ROWID = :OLD.pk;
END;

create or replace TRIGGER "TRG_DEL_FESTIVAL"
instead of delete on v_festivals
referencing new as new old as old
begin
    delete from eventos
    where id_evento=:old.id_evento;
    delete from festivals
    where id_evento=:old.id_evento;
end;

create or replace TRIGGER "TRG_DEL_STAFF"
instead of delete on v_staff
referencing new as new old as old
begin
    delete from pessoas
    where email=:old.email;

```

```

        delete from staff
        where email=:old.email;
    end;

create or replace TRIGGER TRG_del_staff_eventos
INSTEAD OF DELETE ON v_staff_eventos
FOR EACH ROW
BEGIN
DELETE FROM staff_eventos
WHERE ROWID = :OLD.pk;
END;

create or replace TRIGGER TRG_del_venda_bilhetes
INSTEAD OF DELETE ON v_venda_bilhetes
FOR EACH ROW
BEGIN
DELETE FROM venda_bilhetes
WHERE ROWID = :OLD.pk;
END;

create or replace trigger trg_up_agentes
instead of update on v_agentes
referencing new as new old as old
begin
    update pessoas
    set nrBI = :new.nrBI,
    nomeP = :new.nomeP,
    telefone = :new.telefone
    where email = :old.email;
end;

create or replace trigger trg_up_artistas
instead of update on v_artistas_banda
referencing new as new old as old
begin
    update pessoas
    set nrBI = :new.nrBI,
    nomeP = :new.nomeP,
    telefone = :new.telefone,
    email = :new.email
    where email = :old.email;
end;

create or replace trigger trg_up_bandas_eventos
instead of update on v_bandas_eventos
referencing new as new old as old
begin
    update bandas_eventos
    set id_evento = :new.id_evento,
    id_banda = :new.id_banda,
    cachet = :new.cachet
where ROWID = :NEW.pk;
end;

create or replace trigger trg_up_concertos
instead of update on v_concertos

```

```

referencing new as new old as old
begin
    update eventos
    set nomefest = :new.nomefest,
    local = :new.local,
    data = :new.data,
    numbilhetes = :new.numbilhetes
    where id_evento = :old.id_evento;
end;

create or replace trigger trg_up_eventos_empresas
instead of update on v_eventos_empresas
referencing new as new old as old
begin
    update eventos_empresas
    set id_evento = :new.id_evento,
    id_empresa = :new.id_empresa,
    valor = :new.valor
where ROWID = :NEW.pk;
end;

create or replace trigger trg_up_festivais
instead of update on v_festivais
referencing new as new old as old
begin
    update eventos
    set nomefest = :new.nomefest,
    local = :new.local,
    data = :new.data,
    numbilhetes = :new.numbilhetes
    where id_evento = :old.id_evento;
    update festivais
    set numdias = :new.numdias
    where id_evento=:old.id_evento;
end;

create or replace trigger trg_up_staff
instead of update on v_staff
referencing new as new old as old
begin
    update pessoas
    set nrBI = :new.nrBI,
    nomeP = :new.nomeP,
    telefone = :new.telefone
    where email = :old.email;
end;

create or replace trigger trg_up_staff_eventos
instead of update on v_staff_eventos
referencing new as new old as old
begin
    update staff_eventos
    set id_evento = :new.id_evento,
    id_staff = :new.id_staff
where ROWID = :NEW.pk;
end;

```

```

create or replace trigger trg_up_staff_eventos
  instead of update on v_staff_eventos
  referencing new as new old as old
  begin
    update staff_eventos
      set id_evento = :new.id_evento,
          id_staff = :new.id_staff
where ROWID = :NEW.pk;
  end;

```

## 6.4 Funções

```

create or replace FUNCTION bilhetesdisponiveis(ID number) return
number
IS

```

```

var number;
bilhetes number;
disponiveis number;

```

```

BEGIN
  select SUM(numbilhetes - bvendidos) into var from eventos e,
venda_bilhetes v where e.ID_EVENTO = ID AND v.ID_EVENTO = ID;
  select numbilhetes into bilhetes from eventos e where
e.ID_EVENTO = ID;

```

```

if var > bilhetes
  then disponiveis := var - bilhetes;
else
  disponiveis := var;
END IF;

```

```

return disponiveis;
END bilhetesdisponiveis;
/

```

```

create or replace FUNCTION receita(ID number) return number
IS
valor number;
BEGIN
  select NVL(TO_NUMBER(bvendidos),
0)*NVL(TO_NUMBER(precobilh), 0) + NVL(TO_NUMBER(valor), 0) -
NVL(TO_NUMBER(cachet), 0) into valor
from eventos e, venda_bilhetes v, eventos_empresas f,
bandas_eventos b where e.ID_EVENTO = ID
AND v.ID_EVENTO = ID AND f.ID_Evento = ID AND b.ID_Evento =
ID;
  return valor;
END receita;
/

```