Autocarros Portugal

Bases de
Dados P2G2
Alexandre Martins 103552
Giovanni Santos 115637
31 de junho de 2024

Introdução

Baseia-se numa pequena aplicação para gerir uma companhia de autocarros.

Gere os dados dos autocarros, viagens e seus condutores

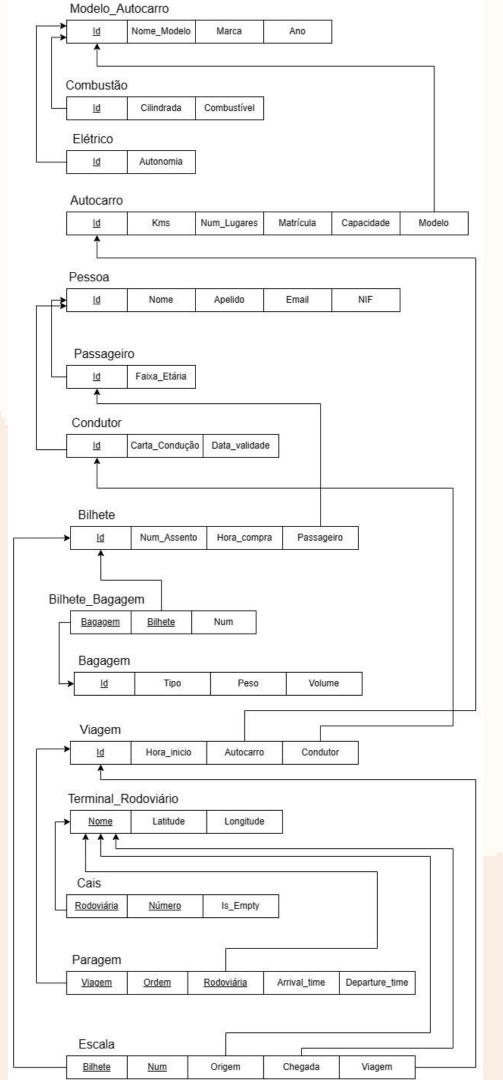
Mantém um registo de todos os passageiros

.

Mantém um registo de todos os bilhetes

<u>ld</u> peso Is_Empty Número volume Cais Bagagem Primeiro Apelido Nome Num_assento Nome Email Possui Completo Define Hora_compra NIF <u>ld</u> Nome Origem <u>ld</u> Pessoa Escala Bilhete Faixa etária Possui Terminal_Rodoviário Latitude Destino Longitute is-a Passageiro Reserva Localizada Origem Faz parte Hora_inicio Departure_time Data_validade <u>ld</u> Paragem Ordem Viagem Condutor Conduz Efetua (Carta_Condução) Arrival_time Destino Nome_modelo Marca Realiza <u>Id</u> Modelo_Autocarro Possui Autocarro (matrícula) Kms <u>ld</u> disjoint is-a Capacidade Num_lugares Combustão Elétrico Combustível Autonomia

ER



Boyce-Codd Normal Form (BCNF)

UDFs

•Obter todas as viagens dado uma certa origem e destino.

```
create function EAP.Viagens_por_Origem_Destino (@origem varchar(15), @destino varchar(15)) returns table
as
    return (
    select *
    from EAP.Viagem
    where @origem in (select rodoviaria from EAP.Viagem_paragens(id))
        and @destino in (select rodoviaria from EAP.Viagem_paragens(id))
        and EAP.ordem_paragem(id, @origem) < EAP.ordem_paragem(id, @destino)
    )
go</pre>
```

	id	hora_inicio	autocarro_id	condutor_id
1		07:00:00.0000000	1	1

Stored Procedures

. Adiciona uma Pessoa e de seguida um Condutor.

```
create procedure EAP.addCondutor (@nome varchar(20), @apelido varchar(20), @email varchar(20) = null, @nif char(9) = null, @carta_conducao char(12))
as
begin
set nocount on;

begin transaction;
    disable trigger EAP.Total_Pessoa on EAP.Pessoa
    exec EAP.addPessoa @nome, @apelido, @email, @nif
    alter table EAP.Pessoa enable trigger Total_Pessoa

declare @pessoa_id int
    select @pessoa_id = max(id) from EAP.Pessoa
    insert into EAP.Condutor(pessoa_id, carta_conducao) values (@pessoa_id, @carta_conducao)
commit transaction
end
go
```

Triggers

• Quando um funcionário é despedido, os seus tuplos relevantes são eliminados em todas as tabelas necessárias.

```
create trigger EAP.Total_Pessoa on EAP.Pessoa
instead of insert
as

if not exists(select * from inserted where id in (select pessoa_id from EAP.Passageiro) or id in (select pessoa_id from EAP.Condutor))
begin

raiserror('Erro! Nao e possivel adicionar Pessoa sem especializacao',16,1)
end
else
insert into EAP.Pessoa select nome,apelido,email,nif from inserted

go
```

Triggers

• Quando um funcionário é despedido, os seus tuplos relevantes são eliminados em todas as tabelas necessárias.

```
create trigger EAP.Total_Pessoa on EAP.Pessoa
instead of insert
as
    if not exists(select * from inserted where id in (select pessoa_id from EAP.Passageiro) or id in (select pessoa_id from EAP.Condutor))
    begin
        raiserror('Erro! Nao e possivel adicionar Pessoa sem especializacao',16,1)
    end
    else
        insert into EAP.Pessoa select nome,apelido,email,nif from inserted
go
```

Indexes

create index idxParagemRodoviaria on EAP.Paragem (rodoviaria)

É muito comum fazermos pesquisa por rodoviaria ao invés de chave na tabela Paragem

DEMO

