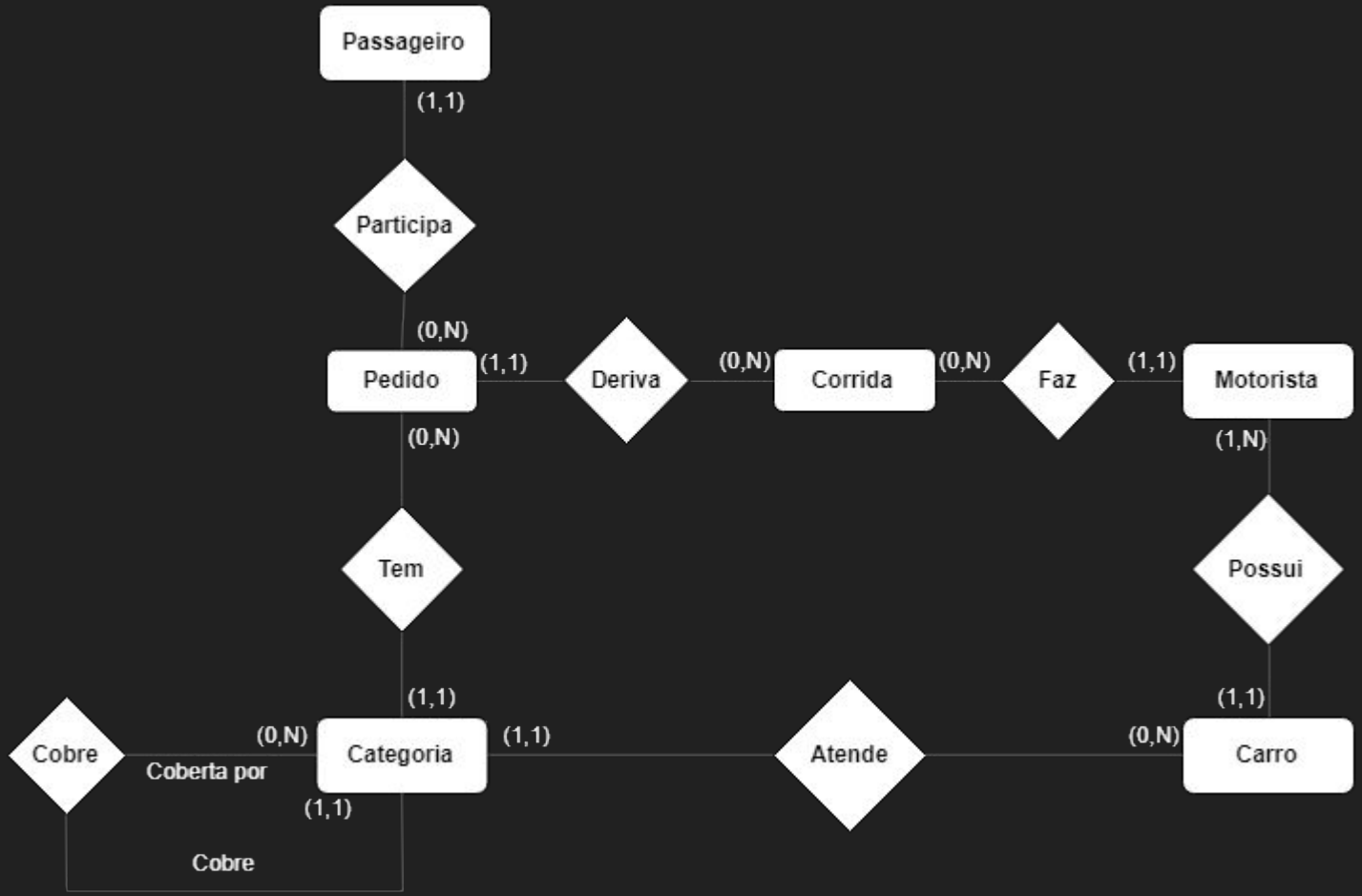
UBER

Modelo Entidade-Relacionamento



Passageiro

- **CPF** (*chave primária*)
- Nome
- E-mail
- Telefone
- Total de corridas
- Avaliação
  - média calculada das notas recebidas em corridas realizadas
  - valor decimal entre 0 e 5
- Endereço de casa
- Endereço do trabalho

Passageiro

VALIDAÇÃO

`verif_passageiro:` trigger antes de inserção na tabela

- Verifica formato do CPF
- Verifica formato do e-mail

Motorista

- **CPF** (*chave primária*)
- Nome
- E-mail
- Telefone
- Total de corridas
- Avaliação
  - média calculada das notas recebidas em corridas realizadas
  - valor decimal entre 0 e 5
- Carro *(chave estrangeira*)
  - número do RENAVAM do carro com o qual o motorista trabalha

Motorista

VALIDAÇÃO

`verif_motorista`: trigger antes de inserção na tabela

- Verifica formato do CPF
- Verifica formato do e-mail

Categoria

- ID (*chave primária*)
- Título (“UberX”, “UberSelect”, “UberBlack”)
- Cobre (*chave estrangeira*)
  - Outra categoria “coberta” pela categoria desse ID
  - Auto-relacionamento que define a hierarquia das categorias de serviço

Carro

- RENAVAM (*chave primária*)
- Placa
- Marca
- Modelo
- Ano
- Categoria (*chave estrangeira*)
  - Define o relacionamento entre um carro e a categoria que ele atende

Categoria

HIERARQUIA
Algumas categorias, como UberBlack, podem atender a pedidos de categorias inferiores como UberX. Isso é definido por um auto-relacionamento entre uma categoria superior e uma inferior.

`check_categoria`: trigger que, numa inserção ou atualização na tabela Corrida, verifica se a categoria é a desejada pelo passageiro ou então uma categoria superior que também *cobre* a categoria desejada.
Exemplo: se o pedido foi feito para uma corrida “UberBlack”, uma tentativa de criar uma corrida para esse pedido com “UberX” levantaria uma exceção, pois “UberX” não cobre “UberBlack”.

Pedido

- ID (*chave primária*)
- Passageiro (*chave estrangeira*)
- Categoria (*chave estrangeira*)
- Endereço de origem
- Endereço de destino
- *Timestamp*: abertura do pedido
- *Timestamp*: seleção de motorista para atender ao pedido
- *Timestamp*: fechamento do pedido (chegada do motorista ou cancelamento)
- *Status* → “aberto”, “esperando motorista”, “atendido”, “cancelado pelo motorista”, “cancelado pelo passageiro”
- Custo (preço a ser pago pelo passageiro, geralmente em caso de taxa de cancelamento)

Pedido

VALIDAÇÃO

`atualizar_pedido`: trigger que atualiza as informações do pedido conforme alterações de status acontecem.

- Guarda *timestamp* quando um motorista aceita atender o pedido.
- Guarda *timestamp* quando o passageiro cancela um pedido.
  - Inclui multa caso isso seja feito 5 minutos após um motorista ser selecionado.
  - Guarda *timestamp* e retorna status para “aberto” (buscando motorista) caso o motorista cancele a corrida.
  - Guarda *timestamp* quando o pedido passa para “atendido”, ou seja, o motorista chegou e começou a corrida.

Pedido

VALIDAÇÃO

`pedidos_sobrepostos`: trigger que impede a inserção de pedidos feitos pelo mesmo passageiro em períodos de tempo sobrepostos.
Um passageiro só pode fazer outro pedido uma vez que o seu último pedido aberto tenha sido fechado (atendido ou cancelado).

Corrida

- ID da corrida (*chave primária*)
- ID do pedido que gerou a corrida (*chave estrangeira*)
- Motorista (*chave estrangeira*)
- *Timestamp* de início
- *Timestamp* de fim
- Endereço inicial
- Endereço final
- Avaliação do motorista (feita pelo passageiro)
- Avaliação do passageiro (feita pelo motorista)
- Custo (preço total da corrida, a ser pago pelo passageiro)

Corrida é uma relação entre um pedido e o motorista que o atendeu

Corrida

VALIDAÇÃO

`corridas_sobrepostas`: trigger que verifica que uma nova corrida não está sendo criada no mesmo período de tempo que uma corrida já existente com o mesmo motorista ou o mesmo passageiro

`uber_select`: trigger que verifica a condição para um motorista fazer uma corrida da categoria “UberSelect”, ou seja, que ele tenha avaliação média maior que 4.5

Corrida

ATUALIZAÇÃO

`atualizar_medias`: trigger que, uma vez que uma corrida é inserida com as avaliações dadas para o motorista e para o passageiro, atualiza as respectivas avaliações médias deles nas tabelas de Motorista e Passageiro.

Procedures

AREAS_PROBLEMATICAS()

Procedure que gera um *ranking* (uma tabela, em ordem) de áreas problemáticas, onde há os maiores índices de cancelamento de corridas.

ESTATISTICAS()

Procedure que calcula e imprime algumas estatísticas a respeito das corridas, como por exemplo, quantidade de corridas feitas em certos intervalos de tempo e média de avaliação dos motoristas por categoria.