Esse documento tem como base a explicação do MySQL que eu criei-com base no solicitado no desafio final-

* Por mais que não tenha sido uma etapa obrigatória eu achei necessário indicar o porque das coisas desse documento, tendo por objetivo uma maior facilidade da análise dos desafios(realizada pelo competente encarregado);
* Mais uma vez, esse documento pretende ser bastante objetivo. Uma vez que, a correção será feita por um competente detentor dos conhecimentos necessários para a análise ( logo, jargões e comentários mais cirúrgicos serão implementados com uma maior constância);

VAMOS OCMEÇAR POR UMA VISÃO MAIS GERAL E ALGUMAS TOMADAS DE DECISÕES QUE FORAM MAIS GLOBAIS NESSE MEU DOCUMENTO

-engine=innoDB: ele foi usado para garantir uma maior integridade na minha referencia ;

-Charset ou colação: serviu “apenas” par armazenar acentos de forma correta

(usei o utf8mb4) e como o sistema não é dito “grande” eu já juntei com o uso do “INT\_AUTO\_INCREMENT”;

-como a ideia é ser sempre eficiente e não gastar dados de uma maneira desnecessária, optei em armazenar datas em “DATE”, pois eu não queria que as horas fossem contabilizadas ( achei desnecessário o uso das horas para saber o quando foi alugado, quando na verdade eu contabilizar os dias torna mais eficiente-pensei em um modelo de equação do 1 grau, em que a minha variável x=entenda “x” como a quantidade de dias, deixa menos abstrato as representações-);

-Sobre os valores decimais que eu utilizei (10,2): pensei em utilizar esse recurso para evitar o tal do erro flutuante (abarcando um valor suficiente-para o sistema não travar- suficiente para a relação diárias/valores);;

---pegando esse laço o uso do “ENUM” serviu para manter esses valores restritos e consistentes (facilitando a leitura e validação dos dados/código utilizado) ;

--por que eu utilizei o “UNIQUE” no CPF/PLACA

Evita uma duplicidade no recurso (não posso ter mais de uma uma) ;

--eu pensei em utilizar esse recurso pois a parte do CPF e da Placa não é nada mais do que uma “senha”, mas como assim uma senha?

--Simples, por ser uma relção da análise combinatória (mais especificadamente um arranjo) não pode, pela relação existir mais de uma relação de mesmo valor e de valores zerados na sua totalidade permutativa(sendo necessário retirar uma análise de todas as possibilidades). Logo, eu pensei nesse recurso para deixar a sua particularidade numérica;

AGORA, JÁ VISTO A MINHA IDEIA MAIS “GLOBAL” POR ASSIM DIZER TORNA-SE MAIS INTERESSANTE UMA ANÁLISE MAIS INDIVIDUALIZADA DOS BLOCOS

* Irei seguir as mesmas ideias nesses blocos até então estabelecidas anteriormente ;

SOBRE A TABELA “CLIENTES”

* Create table cliente(

id\_cliente int primary key

auto\_increment,

nome varchar(100) not null,

cpf char(11) not null unique,

endereco varchar(150),

telefone varchar(100),

data\_cadastro date not null

* id\_cliente int primary key

auto\_increment,

-ele identifica unicamente cada cliente( mesma coisa do RGM de alguma faculdade que você pensar...cada registro é único de cada inscrito);;

--o resultado dele é que cada cliente vai poder ter uma individualidade cadastral para eventuais análises de recursos e de “auxílios” futuros;

* nome VARCHAR(100) not null

-Aqui o nome torna-se obrigatório

-a utilização do 100 foi para abarcar todos os nomes possíveis;

--o resultado disso é que os clientes possuem obrigatoriedade de terem um nome cadastrado;

* cpf CHAR(11) not null unique

-aqui foi a mesma ideia que eu comentei na parte da “visão mais global”, não explicarei novamente para evitar a redundância(já que não é apenas o meu trabalho que o CO Wallace irá analisar);

* endereço VARCHAR(150)
* telefone VARCHAR(15)
* email VARCHAR

-Aqui os campos eu deixei opcional(ou parcialmente opcionais) e os tamanhos escolhidos foi mais uma convenção;

-aqui o cliente tem a opção de colocar esses dados cadastrais ( cabendo ao atendente, por exemplo, mostrar as vantagens do sistema obter essas informações, como a ideia de descontos futuros e de um maior canal comunicativo ( empresa-cliente) ) ;

* data\_cadastro DATE NOT NULL

-utilizei para realizar a data de cadastro do cliente( e como mencionei na parte de uma “visão mais global” achei desnecessário a utilização da hora exata do cadastro);

--resultando em uma gama de histórico de quando os clientes adentraram no sistema ;

SOBRE A TABELA “FUNCIONÁRIOS”

* Para evitar, mais uma vez, redundância de termos já falados/discutidos/utilizados irei pular algumas funções (não por um futuro esquecimento e sim para garantir a linearidade dos pensamentos e da leitura do competente ) ;
* create table funcionario(

id\_funcionario int primary key

auto\_increment,

nome varchar(100) not null,

cpf char(11) not null unique,

cargo varchar(50) not null,

telefone varchar(15),

email varchar(100),

data\_admissao date not null

)engine=InnoDB;

* cargo VARCHAR(50) not null:

-sendo obrigatório para saber( ter um controle) de quem fez determinado atendimento ou determinado registro;

* data\_admissao DATE NOT NULL

-útil para o controle de funcionários;

--o resultado seria que os funcionários existam como referência em contrato dos alugueis ;

--utilizando o FK em aluguel;

SOBRE A TABELA “VEÍCULOS”

create table veiculo(

id\_veiculo int primary key

auto\_increment,

placa varchar(10) not null unique,

modelo varchar(50) not null,

cor varchar(30),

ano\_fabricacao int,

valor\_diaria decimal(10,2) not null,

status

enum('disponível','alugado','manutencao') default 'disponível'

)engine=InnoDB;

* Placa VARCHAR(10) NOT NULL UNIQUE

-Já fiz essa análise, juntamente com o CPF na área “Visão Global”(Irei apenas resgatar algumas ideias) ;

--a placa é o identificador real do veículo (por isso eu utilizei a função “UNIQUE”, pois ela retira uma ideia de duplicidade, que no mundo real não pode- de maneira alguma- acontecer);

--O resultado disso tudo é que ao colocar, em um novo cadastro feito por outro usuário, um veículo com a placa de outro usuário o cadastro não prossegue( pois já existe essa placa, a do exemplo, no inventário) ;

* Modelo,cor,ano\_fabricacao

-são apenas os dados descritivos do veículo;

* Valor\_diaria DECIMAL(10,2) NOT NULL

--utilizei o valor em formato decimal poque ele me fornece uma maior precisão dos dados numéricos;

--resultando, para o meu sistema, calcular o valor\_total do meu aluguel de uma maneira mais confiável( e quando eu digo mais confiável eu quero dizer sem arredondamentos mais grosseiros ( tanto para mais-prejudicando o cliente-

Tanto para menos-prejudicando a empresa-);

* STATUS ENUM(...) DEFAULT

‘disponivel’

--ele serve justamente para controlar a eventual disponibilidade do carro( nesse nosso exemplo) solicitado por um eventual cliente à loja;

--o resultado disso é um melhor controle da entrada( novo veículo disponível) e da saída dos veículos...facilitando na hora da entrega e da demanda por veículos específicos, por exemplo;

SOBRE A TABELA “ALUGUEL”

* create table aluguel(

id\_aluguel int auto\_increment

primary key,

id\_cliente int not null,

id\_funcionario int not null,

id\_veiculo int not null,

data\_inicio date not null,

data\_fim date not null,

valor\_total decimal(10,2) not null,

status

enum('ativo','finalizado','atrasado')

default 'ativo',

constraint fk\_aluguel\_cliente

foreign key (id\_cliente)

references cliente(id\_cliente)

on update cascade

on delete restrict,

constraint fk\_aluguel\_funcionario

foreign key(id\_funcionario)

references veiculo(id\_veiculo)

on update cascade

on delete restrict

)engine=innodb;

* ID\_ALUGUEL PK:

-identifica cada contrato de aluguel( assim como no exemplo do CPF/PLACA essa vai ser, da mesma maneira uma chave primária)

* ID\_CLIENTE, ID\_FUNCIONARIO, ID\_VEICULO SÃO FKs

-ou seja, elas mantem uma integridade referencial;

--logo, você não pode criar um aluguel sem um cliente, cadastrado unicamente, sem um funcionário, cadastrado unicamente, sem um veículo que não estar disponível, ou seja inexistente ;

* ON DELETE RESTRICT

-pq eu utilizei ele?

--a ideia era ,simplesmente, proteger o histórico( pensei assim, caso nessa minha Análise Combinatória-sim, tentei usar matemática em tudo- existisse um cliente, funcionário, veículo relacionado poderia haver uma exclusão, e isso atrapalharia meus dados até então coletados) ;

* ON UPDATE CASCADE  
  --utilizei caso o PK fosse alterado( aqui eu tentei pensar muito mais muito além), queria realmente que ele (PKs) tivesse uma espécie de “guardião” para garantir a sua integridade;
* Data\_inicio
* Data\_fim
* Valor\_total

-eles, todos os mencionados, controlam datas e os valores dos eventuais contratos

* STATUS ENUM(..) DEFAULT ‘ATIVO’

-Controla os ciclos de contrato;

(pensei da seguinte maneira)

--Não é possível inserir um aluguel se o id\_veiculo não existir(fazendo total sentido isso);

--assim como também não é possível apagar um cliente ou veículo se ele tiver em condição de aluguel(claro, sem antes remover/encerrar os alugueis anteriormente....seria o meu ‘RESTRICT’);

SOBRE A TABELA “PAGAMENTO”

create table pagamento(

id\_pagamento int auto\_increment

primary key,

id\_aluguel int not null unique,

valor decimal(10,2) not null,

date\_pagamento date not null,

metodo

enum('cartao','pix','boleto') not null,

status

enum('pendente','concluido','cancelado')default 'pendente',

constraint fk\_pagamento\_aluguel

foreign key(id\_aluguel)

references aluguel(id\_aluguel)

on update cascade

on delete cascade

) engine=InnoDB;

* Foi, na minha opinião, a parte um pouco mais complexa de relacionar por conta dos atributos anteriores que precisava de uma relação extremamente pessoal com essa tabela
* ID\_ALUGUEL INT NOT NULL

-gostaria de explicar o porque eu não utilizei o UNIQUE como em alguns exemplos anteriores;

---Pensei em um viés um pouco mais “social”(como assim?)

---Se eu colocasse o “UNIQUE’ eu estaria dizendo/garantindo que um aluguel tenha no máximo apenas uma forma de pagamento( boleto, cartão, pix, dentre outros)

---ao não utilizar o “UNIQUE” eu quis dizer, ao meu cliente que existem várias opções dele garantir esse aluguel, tendo a ideia de ajudar a concorrência que em sua modelagem utilizou esse “UNIQUE”

--ou seja, aqui é permitido parcelamento/múltiplos pagamentos por aluguel(dando a opção de pagar por apenas uma modalidade ou pela junção dessas variadas modalidades)

* ON DELETE CASCADE no FK

-Pelo simples motivo que se um aluguel for removido( por qualquer que seja o motivo) o seu pagamento associado também será(por que realmente é necessário) ser removido- e de forma automática- evitando registros “sem completude cadastral”;

--o resultado disso tudo fica que ao deletar, por exemplo, “ALUGUEL”, o pagamento correspondente some de maneira automática, graças a essa espécies de laço criada;

* METODO ENUM(...) e status

-Mais uma vez, eles controlam a forma de pagamento e o estado( a exemplo dos status pendentes e concluído);

SOBRE A TABELA “MANUTENÇÃO”

create table manutencao (

id\_manutencao int auto\_increment

primary key,

id\_veiculo int not null,

descricao varchar(260) not null,

custo decimal(10,2),

data\_manutencao date not null,

responsavel varchar(100),

constraint fk\_manutencao\_veiculo

foreign key(id\_veiculo)

references veiculo(id\_veiculo)

on update cascade

on delete cascade

)engine=InnoDB;

* porque eu utilizei/criei essa tabela

--Ela serve para o requisito de registrar uma manutenção dos veículos em algum futuro;

---Pensei que, caso não existisse essa informação, algum veículo-eventualmente poderia apresentar algum defeito e ficar em “off” por um tempo( e tudo bem isso acontecer, mas a empresa precisa ter essa relação em mente, para saber o que fazer e tirar, por exemplo, esse veículo do status de “disponível” até o eventual conserto);

* ON DELETE CASCADE

Pq eu pensei nisso?

-se remover o veículo eu preciso apagar as futuras manutenções associadas ( seria um gasto desnecessário para a empresa eu não ter mais esse veículo em circulação e realizar uma futura manutenção nele);

* Descrição
* Custo
* Data\_manutencao

--a ideia de tudo isso é cobrar o mínimo exigido e ter controle sobre esses processos;

ANÁLISE FINALIZADA

* Pensei em deixar mais completo o trabalho, por mais que essa etapa não tenha sido exigida;
* Esse documento precisa passar a ideia de uma espécie de conversa com o competente que estar lendo, já que não tenho a oportunidade de explicar pessoalmente ao Instrutor ou ao CO Wallace das minhas ideais/pensamentos;
* Esse documento vai passar a ideia de ser algo pessoal( para que o competente possa entender as minhas ideias), mas com um cunho profissional ( abarcando tanto os entendimentos dos instrutores, como de alguma empresa que solicitou esse recurso) ;