

Questão 1

- O DETRAN deseja constituir um banco de dados para controlar as infrações ocorridas no estado.
- Os veículos são identificados pela placa e também descritos por chassi, cor predominante, modelo, categoria e ano de fabricação.
- Cada veículo possui um único proprietário, que é identificado por seu CPF. Deve-se saber o nome, endereço, bairro, cidade estado, telefone (vários), sexo, data de nascimento e idade.
- Todo veículo possui um único modelo; por exemplo, GOL MI, GOL 1.8, UNO CS, etc. Cada modelo é codificado por um número de 6 dígitos.
- Similarmente ao modelo, uma categoria deve ser atribuída a cada veículo; por exemplo, AUTOMÓVEL, MOTOCICLETA, CAMINHÃO, etc. Cada categoria é codificada por um número de 2 dígitos.
- Existem diversos tipos de infração, AVANÇO DE SINAL VERMELHO, PARADA SOBRE A FAIXA DE PEDESTRES, etc., identificada pelo código associado. A cada tipo de infração é associado um valor que deverá ser cobrado na ocorrência de infração.
- Uma infração é identificada pelo veículo infrator, data/hora e tipo de infração. Também é importante conhecer o local, velocidade aferida (se possível) e o agente de trânsito.
- Cada local é descrito pelo código, posição geográfica e velocidade permitida; um local é geralmente referenciado por seu código.
- Um agente de trânsito é conhecido através de sua matrícula, sendo também descrito pelo nome, data de contratação e tempo de serviço.

Questão 2

Requisitos para funcionamento como Hardware.

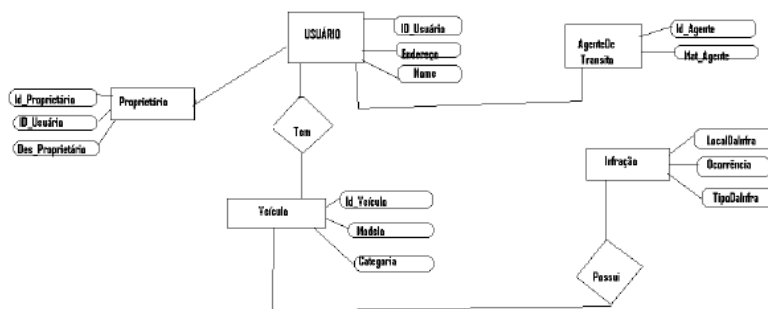
- Processador de 1 gigahertz (GHz) ou mais rápido, de 32 bits (x86) ou 64 bits (x64);
- 1 gigabyte (GB) de RAM (32 bits) ou 2 GB de RAM (64 bits);
- 5 GB de espaço disponível no disco rígido (32 bits) ou 20 GB (64 bits).

Questão 3.1. Definição do requisitos funcionais;

Requisitos :

1. Os veículos são identificados pela placa e também descritos por Chassi, Cor predominante, modelo, ano de fabricação.
2. Cada veículo possui um único proprietário, que é identificado por seu CPF e o Nome.
3. Todo veículo possui um único modelo; por exemplo, GOLF Flash.
4. Cada veículo tem uma categoria como: Carro, Moto ou Caminhão.
5. Existem diversos tipos de infração, AVANÇO DE SINAL VERMELHO, PARADA SOBRE A FAIXA DE PEDESTRES, etc.
6. Uma infração é identificada pelo veículo infrator.
7. Um infração de trânsito é identificada pelo veículo infrator.
8. Um agente tem o poder de multar um veículo, e multas geram pontos na carteira do motorista.

Questão 3.2. Modelagem conceitual;



Questão 3.3

O MySQL é um sistema que utiliza a linguagem [SQL](#). Atrvés de suas características.

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers [ODBC](#), [JDBC](#) e [.NET](#) e módulos de interface para diversas linguagens de programação, como [Delphi](#), [Java](#), [C/C++](#), [C#](#), [Visual Basic](#), [Python](#), [Perl](#), [PHP](#), [ASP](#) e [Ruby](#));
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de [hardware](#);
- Facilidade de uso;
- É um [Software Livre](#) com base na GPL (entretanto, se o programa que acessar o Mysql não for GPL, uma licença comercial deverá ser adquirida);
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid...
- Suporta controle transacional;

- Suporta Triggers;
- Suporta Cursors (Non-Scrollable e Non-Updateable);
- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;
- Interfaces gráficas ([MySQL Toolkit](#)) de fácil utilização cedidos pela MySQL Inc.

Agente						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idAgente	INTEGER	PK	NN	UNSIGNED		
Usuariu_idAgente	INTEGER	PK	NN	UNSIGNED		
Nm_Agent	NVARCHAR					
Dm_Agent	VARCHAR(4)					
Data_Criar	DATE					
Temp_Cas	DATE					
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idAgente		
Agente_PKindex		Index		Usuariu_idAgente		

AnoFabricacao						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idAnoFabricacao	INTEGER	PK	NN	UNSIGNED		
Ano	DATE					
Nm_Ano	VARCHAR(2)					
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idAnoFabricacao		

Categoria						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idCategoria	INTEGER	PK	NN	UNSIGNED		
Dmz_Cat	VARCHAR(4)					
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idCategoria		

Cidade						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idCidade	INTEGER	PK	NN	UNSIGNED		
UF_ufUF	INTEGER		NN	UNSIGNED		
Cidade	VARCHAR(2)					
Dmz_Cid	VARCHAR(4)					

DDD						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idDDD	INTEGER	PK	NN	UNSIGNED		
Dmz_DDD	VARCHAR(2)					
DDD	INTEGER			UNSIGNED		
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idDDD		

Endereco						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idEndereco	INTEGER	PK	NN	UNSIGNED		
Cidade_idCidade	INTEGER		NN	UNSIGNED		
Dmz_End	VARCHAR(4)					
CEP	NVARCHAR					
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idEndereco		
Endereco_PKindex		Index		Cidade_idCidade		

Infracao						
Columnname	Datatype	PrimaryKey	NotNull	Flags	Default Value	Comment
idInfracao	INTEGER	PK	NN	UNSIGNED		
Veiculo_idVeiculo	INTEGER		NN	UNSIGNED		
LocalInfracao_idLocalInfracao	INTEGER		NN	UNSIGNED		
OcorrenciaInf_idOcorrenciaInf	INTEGER		NN	UNSIGNED		
TiposInfracao_idTiposInfracao	INTEGER		NN	UNSIGNED		
Dmz_Infr	VARCHAR(3)					
Data_Infr	DATE					
Indexname		IndexType		Columns		
PRIMARY		PRIMARY		idInfracao		
Infracao_PKindex		Index		TiposInfracao_idTiposInfracao		
Infracao_PKindex		Index		OcorrenciaInf_idOcorrenciaInf		
Infracao_PKindex		Index		LocalInfracao_idLocalInfracao		
Infracao_PKindex		Index		Veiculo_idVeiculo		

LocalInfracao						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idLocalInfracao	INTEGER	PK	NN	UNSIGNED		
Desc_Local	VARCHAR(20)					
PontosGeograficos	VARCHAR(20)					
NumeroDePermite	FLCAT					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idLocalInfracao		

Modelo						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idModelo	INTEGER	PK	NN	UNSIGNED		
Det_Model	VARCHAR(20)					
Modelo	VARCHAR(20)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idModelo		

OcorrenciaInf						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idOcorrenciaInf	INTEGER	PK	NN	UNSIGNED		
ValorColado	FLCAT					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idOcorrenciaInf		

Proprietario						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idProprietario	INTEGER	PK	NN	UNSIGNED		
Usuario_idUsuario	INTEGER	PK	NN	UNSIGNED		
Del_Pro	VARCHAR(30)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idProprietario		
Proprietario_FKIndex1		Index		Usuario_idUsuario		

Sexo						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idSexo	INTEGER	PK	NN	UNSIGNED		
Del_Sexo	VARCHAR(20)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idSexo		

Telefone						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idTelefone	INTEGER	PK	NN	UNSIGNED		
Usuario_idUsuario	INTEGER	PK	NN	UNSIGNED		
DDD_idDDD	INTEGER	PK	NN	UNSIGNED		
TipTelefone_idTipTelefone	INTEGER	PK	NN	UNSIGNED		
Del_Tel	VARCHAR(30)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idTelefone		
Telefone_FKIndex1		Index		Usuario_idUsuario		
Telefone_FKIndex2		Index		TipTelefone_idTipTelefone		
Telefone_FKIndex3		Index		Usuario_idUsuario		

TipoInfracao						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idTipoInfracao	INTEGER	PK	NN	UNSIGNED		
Del_TipoInfracao	VARCHAR(30)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idTipoInfracao		

TipoTelefone						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idTipoTelefone	INTEGER	PK	NN	UNSIGNED		
Del_Tipo	VARCHAR(20)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idTipoTelefone		

UF						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idUF	INTEGER	PK	NN	UNSIGNED		
UF	VARCHAR(2)					
DelEstado	VARCHAR(40)					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idUF		

Usuario						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment
idUsuario	INTEGER	PK	NN	UNSIGNED		
Endereco_idEndereco	INTEGER	NN	UNSIGNED			
Sexo_idSexo	INTEGER	NN	UNSIGNED			
Nome	VARCHAR(60)					
DataNasc	DATE					
CPF	INT					
Idade	NUMERIC					
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		idUsuario		
Usuario_FKIndex1		Index		Sexo_idSexo		
Usuario_FKIndex2		Index		Endereco_idEndereco		


```
CREATE TABLE LocalInfracao (  
  idLocalInfracao INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Desc_Local VARCHAR(30) NULL,  
  PociacaoGeografica VARCHAR(30) NULL,  
  VelocidadePermitida FLOAT NULL,  
  PRIMARY KEY(idLocalInfracao)  
);
```

```
CREATE TABLE Sexo (  
  idSexo INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Des_sexo VARCHAR(20) NULL,  
  PRIMARY KEY(idSexo)  
);
```

```
CREATE TABLE UF (  
  idUF INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  UF VARCHAR(2) NULL,  
  Descricao VARCHAR(45) NULL,  
  PRIMARY KEY(idUF)  
);
```

```
CREATE TABLE TipoTelefone (  
  idTipoTelefone INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Des_Tipo VARCHAR(20) NULL,  
  PRIMARY KEY(idTipoTelefone)  
);
```

```
CREATE TABLE TipoInfracao (  
  idTipoInfracao INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Desc_Tp_Infracao VARCHAR(30) NULL,  
  PRIMARY KEY(idTipoInfracao)  
);
```

```
CREATE TABLE DDD (  
  idDDD INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Des_DDD VARCHAR(20) NULL,  
  DDD INTEGER UNSIGNED NULL,  
  PRIMARY KEY(idDDD)  
);
```

```
CREATE TABLE Categoria (  
  idCategoria INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Desc_Cat VARCHAR(45) NULL,  
  PRIMARY KEY(idCategoria)  
);
```

```
CREATE TABLE AnoFabricacao (  
  idAnoFabricacao INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  Ano DATE NULL,
```

```

Desc_Ano VARCHAR(30) NULL,
PRIMARY KEY(idAnoFabricacao)
);

```

```

CREATE TABLE Cidade (
  idCidade INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  UF_idUF INTEGER UNSIGNED NOT NULL,
  Cidade VARCHAR(20) NULL,
  Des_Cid VARCHAR(45) NULL,
  PRIMARY KEY(idCidade),
  FOREIGN KEY(UF_idUF)
    REFERENCES UF(idUF)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

CREATE TABLE Endereco (
  idEndereco INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Cidade_idCidade INTEGER UNSIGNED NOT NULL,
  Desc_End VARCHAR(45) NULL,
  CEP NUMERIC NULL,
  PRIMARY KEY(idEndereco),
  INDEX Endereco_FKIndex1(Cidade_idCidade),
  FOREIGN KEY(Cidade_idCidade)
    REFERENCES Cidade(idCidade)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

CREATE TABLE Usuario (
  idUsuario INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Endereco_idEndereco INTEGER UNSIGNED NOT NULL,
  Sexo_idSexo INTEGER UNSIGNED NOT NULL,
  Nome VARCHAR(50) NULL,
  DataNasc DATE NULL,
  CPF INT NULL,
  Idade NUMERIC NULL,
  PRIMARY KEY(idUsuario),
  INDEX Usuario_FKIndex1(Sexo_idSexo),
  INDEX Usuario_FKIndex2(Endereco_idEndereco),
  FOREIGN KEY(Sexo_idSexo)
    REFERENCES Sexo(idSexo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(Endereco_idEndereco)
    REFERENCES Endereco(idEndereco)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

CREATE TABLE Telefone (
  idTelefone INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Usuario_idUsuario INTEGER UNSIGNED NOT NULL,
  DDD_idDDD INTEGER UNSIGNED NOT NULL,
  TipoTelefone_idTipoTelefone INTEGER UNSIGNED NOT NULL,
  Desc_tel VARCHAR(20) NULL,
  PRIMARY KEY(idTelefone, Usuario_idUsuario, DDD_idDDD),
  INDEX Telefone_FKIndex1(TipoTelefone_idTipoTelefone),

  INDEX Telefone_FKIndex2(Usuario_idUsuario),
  FOREIGN KEY(TipoTelefone_idTipoTelefone)
    REFERENCES TipoTelefone(idTipoTelefone)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(Usuario_idUsuario)
    REFERENCES Usuario(idUsuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(DDD_idDDD)
    REFERENCES DDD(idDDD)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

CREATE TABLE Agente (
  idAgente INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Usuario_idUsuario INTEGER UNSIGNED NOT NULL,
  Mat_Agent NUMERIC NULL,
  Des_Agnt VARCHAR(45) NULL,
  Data_Conta DATE NULL,
  TempoCasa DATE NULL,
  PRIMARY KEY(idAgente, Usuario_idUsuario),
  INDEX Agente_FKIndex1(Usuario_idUsuario),
  FOREIGN KEY(Usuario_idUsuario)
    REFERENCES Usuario(idUsuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

CREATE TABLE Proprietario (
  idProprietario INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Usuario_idUsuario INTEGER UNSIGNED NOT NULL,
  Des_Pro VARCHAR(30) NULL,
  PRIMARY KEY(idProprietario, Usuario_idUsuario),
  INDEX Proprietario_FKIndex1(Usuario_idUsuario),
  FOREIGN KEY(Usuario_idUsuario)
    REFERENCES Usuario(idUsuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```


);

```
CREATE TABLE Veiculo (
  idVeiculo INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  AnoFabricacao_idAnoFabricacao INTEGER UNSIGNED NOT NULL,
  Proprietario_Usuario_idUsuario INTEGER UNSIGNED NOT NULL,
  Proprietario_idProprietario INTEGER UNSIGNED NOT NULL,
  Categoria_idCategoria INTEGER UNSIGNED NOT NULL,
  Modelo_idModelo INTEGER UNSIGNED NOT NULL,
  Desc_Veiculo VARCHAR(30) NULL,
  Chassi VARCHAR(30) NULL,
  PRIMARY KEY(idVeiculo),
  INDEX Veiculo_FKIndex1(Modelo_idModelo),
  INDEX Veiculo_FKIndex2(AnoFabricacao_idAnoFabricacao),
  INDEX Veiculo_FKIndex3(Categoria_idCategoria),
  INDEX Veiculo_FKIndex4(Proprietario_idProprietario,
  Proprietario_Usuario_idUsuario),
  FOREIGN KEY(Modelo_idModelo)
    REFERENCES Modelo(idModelo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(AnoFabricacao_idAnoFabricacao)
    REFERENCES AnoFabricacao(idAnoFabricacao)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(Categoria_idCategoria)
    REFERENCES Categoria(idCategoria)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(Proprietario_idProprietario, Proprietario_Usuario_idUsuario)
    REFERENCES Proprietario(idProprietario, Usuario_idUsuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

```
CREATE TABLE Infracao (
  idInfracao INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Veiculo_idVeiculo INTEGER UNSIGNED NOT NULL,
  LocalInfracao_idLocalInfracao INTEGER UNSIGNED NOT NULL,
  Ocorrencialnf_idOcorrencialnf INTEGER UNSIGNED NOT NULL,
  TipoInfracao_idTipoInfracao INTEGER UNSIGNED NOT NULL,
  Desc_Infra VARCHAR(30) NULL,
  Data_Infra DATE NULL,
  PRIMARY KEY(idInfracao),
  INDEX Infracao_FKIndex1(TipoInfracao_idTipoInfracao),
  INDEX Infracao_FKIndex2(Ocorrencialnf_idOcorrencialnf),
  INDEX Infracao_FKIndex3(LocalInfracao_idLocalInfracao),
  INDEX Infracao_FKIndex4(Veiculo_idVeiculo),
  FOREIGN KEY(TipoInfracao_idTipoInfracao)
```

```

REFERENCES TipoInfracao(idTipoInfracao)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(Ocorrencialnf_idOcorrencialnf)
REFERENCES Ocorrencialnf(idOcorrencialnf)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(LocalInfracao_idLocalInfracao)
REFERENCES LocalInfracao(idLocalInfracao)

ON UPDATE NO ACTION,
FOREIGN KEY(Veiculo_idVeiculo)
REFERENCES Veiculo(idVeiculo)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

```

Questao 7

Inserindo dados nas tabelas:

Tabela UF:

```

insert into uf (uf,descricao) values ('DF', 'Distrito Federal');
insert into uf (uf,descricao) values ('MG', 'Minas Gerais');
insert into uf (uf,descricao) values ('GO', 'Goias');

```

idUF	UF	Descricao
1	DF	Distrito Federal
2	MG	Minas Gerais
3	GO	Goias

tabela Cidade:

```

insert into cidade (uf_iduf,cidade) values (1, 'Brasilia');
insert into cidade (uf_iduf,cidade) values (1, 'Sobradinho');
insert into cidade (uf_iduf,cidade) values (2, 'Belo Horizonte');
insert into cidade (uf_iduf,cidade) values (3, 'Goiania');

```

idCidade	UF_idUF	Cidade	Des Cid
1	1	Brasilia	(NULL)
2	1	Sobradinho	(NULL)
3	2	Belo Horizonte	(NULL)
4	3	Goiania	(NULL)

tabela Endereco:

insert into endereco (cidade_idcidade, desc_end, cep) values (1, 'Asa Norte', 74000000);
insert into endereco (cidade_idcidade, desc_end, cep) values (2, 'Qd.08 Cj. 13', 76000000);
insert into endereco (cidade_idcidade, desc_end, cep) values (3, 'Bairro X rua Y casa 250', 38000000);
insert into endereco (cidade_idcidade, desc_end, cep) values (4, 'Bairro M rua N casa 12', 96000000);

idEndereco	Cidade_idCidade	Desc_End	CEP
1	1	Asa Norte	74000000
2	2	Qd.08 Cj. 13	76000000
3	3	Bairro X rua Y casa 250	38000000
4	4	Bairro M rua N casa 12	96000000

tabela sexo:

insert into sexo values (1, 'Masculino');
insert into sexo values (2, 'Feminino');

idSexo	Des_sexo
1	Masculino
2	Feminino

tabela usuario:

insert into usuario values (1, 1, 1, 'Joao Marcos de Souza', '1987-12-30', 11122233344, 23);
insert into usuario values (2, 2, 1, 'Paulo Moraes', '1960-12-30', 111111111111, 51);
insert into usuario values (3, 3, 2, 'Ana Maria', '1960-12-30', 222222222222, 51);
insert into usuario values (4, 4, 2, 'Maria Claudia', '1960-12-30', 333333333333, 51);

idUsuario	Endereco_idEndereco	Sexo_idSexo	Nome	DataNasc	CPF	Idade
1	1	1	Joao Marcos de Souza	1987-12-30	2147483647	23
2	2	1	Paulo Moraes	1960-12-30	2147483647	51
3	3	2	Ana Maria	1960-12-30	2147483647	51
4	4	2	Maria Claudia	1960-12-30	2147483647	51

tabela anoFabricacao

insert into anofabricacao values (1, '2009-01-01', 2009);
insert into anofabricacao values (2, '2010-01-01', 'ano 2010');
insert into anofabricacao values (3, '2011-01-01', 'ano 2011');

idAnoFabricacao	Ano	Desc_Ano
1	2009-01-01	2009
2	2010-01-01	ano 2010
3	2011-01-01	ano 2011

tabela proprietario:

insert into proprietario values (1, 1, 'Joao');
insert into proprietario values (2, 2, 'Paulo');

insert into proprietario values (3, 3, 'Ana');

idProprietario	Usuario_idUsuario	Des_Pro
1	1	Joao
2	2	Paulo
3	3	Ana
4	4	Maria

tabela modelo:

insert into modelo values (1,'corsa classic', 'Classic');
insert into modelo values (2,'Gol 1.6', 'Gol');
insert into modelo values (3,'Van', 'Van');
insert into modelo values (4,'Caminhao', 'Caminhao');

idModelo	Des_Mod	Modelo
1	corsa classic	Classic
2	Gol 1.6	Gol
3	Van	Van
4	Caminhao	Caminhao

tabela categoria:

insert into categoria values (1,'passeio');
insert into categoria values (2,'carga');
insert into categoria values (3,'transporte');

idCategoria	Desc_Cat
1	passeio
2	carga
3	transporte

tabela veiculo:

insert into veiculo values (1, 1, 1, 1, 1, 1, 'corsa', 'ABC1234', 'ABC123456789');
insert into veiculo values (2, 1, 2, 2, 1, 2, 'gol', 'ABC1111', 'ABC1111111111');
insert into veiculo values (3, 2, 3, 3, 2, 3, 'van', 'ABC2222', 'ABC2222222222');
insert into veiculo values (4, 3, 4, 4, 3, 4, 'caminhao', 'ABC3333', 'ABC333333333333');

idVeiculo	AnoFabricacao	idAnoFabricacao	Proprietario_Usuario_idUsuario	Proprietario_idProprietario	Categoria_idCategoria	Modelo_idModelo	Des_Vel	Placa	Chassi
1	2	1	1	1	1	1	1	1	1
2	2	1	2	2	2	2	2	2	2
3	2	2	3	3	3	3	3	3	3
4	2	3	4	4	4	4	4	4	4

tabela agente:

insert into agente values (1, 2, 123456, 'Agente/Fiscal', '1990-01-01', '21');

idAgente	Usuario_idUsuario	Mat_Agent	Des_Agnt	Data_Conta	TempoCasa
1	2	123456	Agente/Fiscal	1990-01-01	0000-00-00

tabela DDD:

insert into ddd values (1, 'DF', 61);

insert into ddd values (2, 'MG', 31);

insert into ddd values (3, 'GO', 62);

idDDD	Des_DDD	DDD
1	DF	61
2	MG	31
3	GO	62

tabela tipo de infracao:

insert into tipoinfracao values (1, 'Grave');

insert into tipoinfracao values (2, 'Gravissima');

idTipoInfracao	Desc_Tp_Infracao
1	Grave
2	Gravissima

tabela tipoTelefone:

insert into tipotelefone values (1, 'Residencial');

insert into tipotelefone values (2, 'Celular');

idTipoTelefone	Des_Tipo
1	Residencial
2	Celular

tabela telefone:

insert into telefone values (1,1,1,1, '33333333');

insert into telefone values (2,2,1,1, '33334444');

insert into telefone values (3,3,2,2, '99333333');

insert into telefone values (4,4,3,2, '99443333');

idTelefone	Usuario_idUsuario	DDD_idDDD	TipoTelefone_idTipoTelefone	Desc_tel
1	1	1	1	33333333
2	2	1	1	33334444
3	3	2	2	99333333
4	4	3	2	99443333

Questão 8. Prazo e estimativa de custo;

Eu sei que está relacionado a preço do sistema com um todo, como preço da licença do SGBD e o preço do banco! Qualquer coisa agente pergunta o professor!

Questão 9

O backup consistente do banco de dados é de extrema importância para que possamos manter a integridade dos dados caso haja uma falha do sistema, hardware ou até mesmo para corrigir eventuais falhas de usuários, como por exemplo, a remoção acidental de um banco de dados. Para isto, é importante a adoção de uma política consistente de backup (diariamente), bem como conhecer as possíveis técnicas para fazê-lo. No MySQL é possível fazermos backup binário do banco, isto é, será guardado uma cópia da estrutura de arquivos e diretórios que constituem os seus dos bancos de dados e tabelas.

Além disto, pode-se optar pelo backup dos dados, onde serão armazenados os dados em formato texto ou em forma de comandos SQL. Vamos descrever aqui como utilizar estas duas formas de backup para a execução de uma cópia consistente de dados.

Ao realizar o procedimento de backup cria-se uma imagem dos seus dados no momento da execução da rotina de backup. Quando houver problema s com o seu banco de dados que necessite do backup, você pode utilizar o seu último backup retornando só os dados para a situação em que o banco se encontrava no momento deste backup. O que acontece com os dados alterados ou inseridos entre o backup e a falha? No MySQL você pode habilitar um log binário de alterações (opção log-bin no arquivo de configuração), que armazenam todos os comandos que modificam a estrutura do banco de dados, sendo que estes podem ser utilizados para recuperar os dados não contidos no backup. Os logs são criados com a extensão que indica o número de sequência do log, que é incrementado sempre que um novo log é criado. Para "traduzir" o log binário em comandos SQL, utilize a ferramenta mysqlbinlog, sendo que a saída deste poderá ser utilizada diretamente como entrada para o MySQL.

Esse parte aqui não precisa!

Consultas

1) Quais os nomes dos usuários que tem carro?

```
select distinct nome from usuario, proprietario, veiculo where  
idusuario=usuario_idusuario and idproprietario=Proprietario_idProprietario;
```

"Joao Marcos de Souza"

"Paulo Morais"

"Maria Claudia"

2) Selecione todas os modelos de carro.

```
select modelo from modelo;
```

"Classic"

"Gol"

"Van"

"Caminhao"

3) Quais usuários tem mais de 30 anos?

```
select nome from usuario where idade > 30;
```

"Paulo Morais"

"Ana Maria"

"Maria Claudia"

4) Selecione todas as informações dos usuários cujo nome ou parte do nome possua "NA".

```
SELECT *
```

```
FROM usuario
```

```
WHERE nome LIKE '%NA%';
```

"3" "3" "2" "AnaMaria" "1960-12-30""2147483647" "51"

5) Selecione o nome de todos os usuários que são agentes.

```
SELECT nome from usuario, agente where Usuario_idUsuario=idusuario;
```

"Paulo Morais"

6) Selecione todos os nomes dos usuários que não possuem código 1 e 4.

```
SELECT nome
```

```
FROM usuario
```

```
WHERE idusuario NOT IN (1,4);
```

```
"nome"
```

```
"Paulo Morais"
```

```
"Ana Maria"
```

7) Ordene todos os usuários por código, em ordem decrescente.

```
SELECT idusuario, nome
```

```
FROM usuario
```

```
ORDER BY idusuario DESC;
```

```
idusuario  "nome"
4          "Maria Claudia"
3          "AnaMaria"
2          "Paulo Morais"
1          "Joao Marcos de Souza"
```

8) Selecione idade média dos usuários por sexo;

```
SELECT idsexo, avg(idade)
FROM usuario, sexo
where idsexo=Sexo_idSexo
GROUP BY idsexo;
```

```
idsexoavg(idade)
1          37
2          51
```

9) Selecione o número de usuários no sistema.

```
SELECT count(*)
```

```
FROM usuario;
```

```
count(*)
```

```
4
```

10)Selecione a menor e a maior idade dos usuários.

```
SELECT min(idade), max(idade)
```

```
FROM usuário;
```


min(idade)	max(idade)
23	51