

S112 – Banco de Dados

Professor MSc. Eng. Eng. Márcio José de Lemos

E-mail: marcio.lemos@senairs.org.br

<http://lattes.cnpq.br/4769158065464009>

Projeto de Banco de Dados



SQL

A linguagem **SQL** é basicamente dividida em três tipos de comandos:

SQL = DDL + DML + DCL

SQL = DDL + DML + DCL

DDL (definição de dados)

- Comandos: CREATE, DROP, ALTER

DML (manipulação de dados)

- Comandos: SELECT, INSERT, UPDATE e DELETE

DCL(controle de dados)

- Comandos: GRANT e REVOKE

DCL

DCL (*Data Control Language*) Linguagem de Controle de Dados ---> categoria não adotada oficialmente.

DCL

DCL: conjunto de comandos usado em sistemas multiusuário para **definir os privilégios** de acesso aos dados a cada usuário.

DCL

Os comandos de controle de acesso aos dados são usados para **implementar segurança** e **privacidade** em bancos de dados.

Comandos SQL -DCL

Para definir as permissões a um usuário utiliza-se o comando **GRANT**.

Sintaxe:

```
GRANT    <permissao>    ON    <base>.<tabela>    TO  
'<user>'@'localhost' IDENTIFIED BY '<senha>' WITH  
GRANT OPTION;
```

Comandos SQL -DCL

Para adicionar permissão de:

leitura,

inserção e

modificação.

```
GRANT select, insert, update ON controle.* TO  
'MarcioLemos'@'%' WITH GRANT OPTION;
```

Comandos SQL -DCL

E para visualizar as permissões que foram aplicadas:

SHOW GRANTS FOR 'MarcioLemos'@'%';

Comandos SQL -DCL

Exemplo com todos os privilégios:

```
GRANT ALL PRIVILEGES ON controle.* TO  
'MarcioLemos'@'%' WITH GRANT OPTION;
```

Privilégios e controle de acessos (comandos **GRANT** e **REVOKE**)

DCL

Os comandos **GRANT** e **REVOKE** permitem aos administradores do sistema criar utilizadores e conceder e revogar direitos aos utilizadores do MySQL em **quatro (4) níveis de privilégios**.

Criar um utilizado

```
CREATE USER 'Dia'@'localhost'IDENTIFIED BY 'obscure';
```

```
CREATE USER 'Lua'@'localhost'IDENTIFIED BY 'obscure';
```

```
CREATE USER 'Mar'@'localhost'IDENTIFIED BY 'obscure';
```

SINTAXE

GRANT priv_type [(column_list)] [, tipo_priv [(column_list)] ...]

ON {tbl_name | * | *.* | db_name.*}

TO user_name [IDENTIFIED BY [PASSWORD] 'password'] [, user_name [IDENTIFIED BY [PASSWORD] 'password'] ...]

[REQUIRE NONE | [{SSL|X509}]

[CIPHER cipher [AND]]

[ISSUER issuer [AND]]

[SUBJECT subject]]

[WITH [GRANT OPTION | MAX_QUERIES_PER_HOUR # | MAX_UPDATES_PER_HOUR # | MAX_CONNECTIONS_PER_HOUR #]]



SINTAXE

REVOKE priv_type

[(column_list)] [, priv_type [(column_list)] ...]

ON {tbl_name | * | *.* | db_name.*}

FROM user_name [, user_name ...]

SINTAXE

Exemplo 1 comando:

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP  
ON bankaccount.*  
TO Pedro@localhost  
IDENTIFIED BY 'obscure';
```

Nível Global

Privilégios globais aplicam para todos os bancos de dados em um determinado servidor.

Nível Global

Estes privilégios são armazenados na tabela **mysql.user**.

GRANT ALL ON *.* e **REVOKE ALL ON *.***
concederão e revogarão apenas privilégios globais.



Nível da Base de Dados

Privilégios de bancos de dados aplicam-se a todas as tabelas em um determinado banco de dados.

Estes privilégios são armazenados nas **tabelas** **mysql.db** e **mysql.host**.

Nível da Base de Dados

GRANT ALL ON db.* e **REVOKE ALL ON db.***
concederão e **revogarão** apenas privilégios de
banco de dados.

Nível das tabelas

Privilégios de tabelas aplicam-se a todas as colunas em uma determinada tabela.

Estes privilégios são armazenados na tabela **mysql.tables_priv**.

Nível das tabelas

GRANT ALL ON db.table e REVOKE ALL ON db.table concederão e revogarão apenas privilégios de tabelas.

Nível das colunas

Privilégios de colunas aplicam-se a uma única coluna em uma determinada tabela.

Estes privilégios são armazenados na tabela **mysql.columns_priv**

Para as instruções **GRANT** e **REVOKE**,
tipo_priv pode ser especificado como
um dos seguintes:

ALL [PRIVILEGES]	Configura todos os privilégios simples exceto WITH GRANT OPTION
ALTER	Permite o uso de ALTER TABLE
CREATE	Permite o uso de CREATE TABLE
CREATE TEMPORARY TABLES	Permite o uso de CREATE TEMPORARY TABLE
DELETE	Permite o uso de DELETE
DROP	Permite o uso de DROP TABLE.
EXECUTE	Permite que o utilizador execute stored procedures (MySQL 5.0)
FILE	Permite o uso de SELECT ... INTO OUTFILE e LOAD DATA INFILE.
INDEX	Permite o uso de CREATE INDEX e DROP INDEX
INSERT	Permite o uso de INSERT

INSERT

Permite o uso de INSERT

INDEX

Permite o uso de CREATE INDEX e DROP INDEX

LOCK TABLES	Permite o uso de LOCK TABLES em tabelas nas quais se tem o privilégio SELECT.
PROCESS	Permite o uso de SHOW FULL PROCESSLIST
REFERENCES	Para o futuro
RELOAD	Permite o uso de FLUSH
REPLICATION CLIENT	Da o direto ao utilizador de perguntar onde o slave/master está.
REPLICATION SLAVE	Necessário para a replicação dos slaves (para ler logs binário do master).
SELECT	Permite o uso de SELECT
SHOW DATABASES	SHOW DATABASES mostra todos os banco de dados.
SHUTDOWN	Permite o uso de mysqladmin shutdown

SHUTDOWN

Permite o uso de mysqladmin shutdown

SHOW DATABASES

SHOW DATABASES mostra todos os banco de dados.

SUPER

Permite a conexão (uma vez) mesmo se max_connections tiverem sido alcançados e executa o comando CHANGE MASTER, KILL thread, mysqladmin debug, PURGE MASTER LOGS e SET GLOBAL

UPDATE

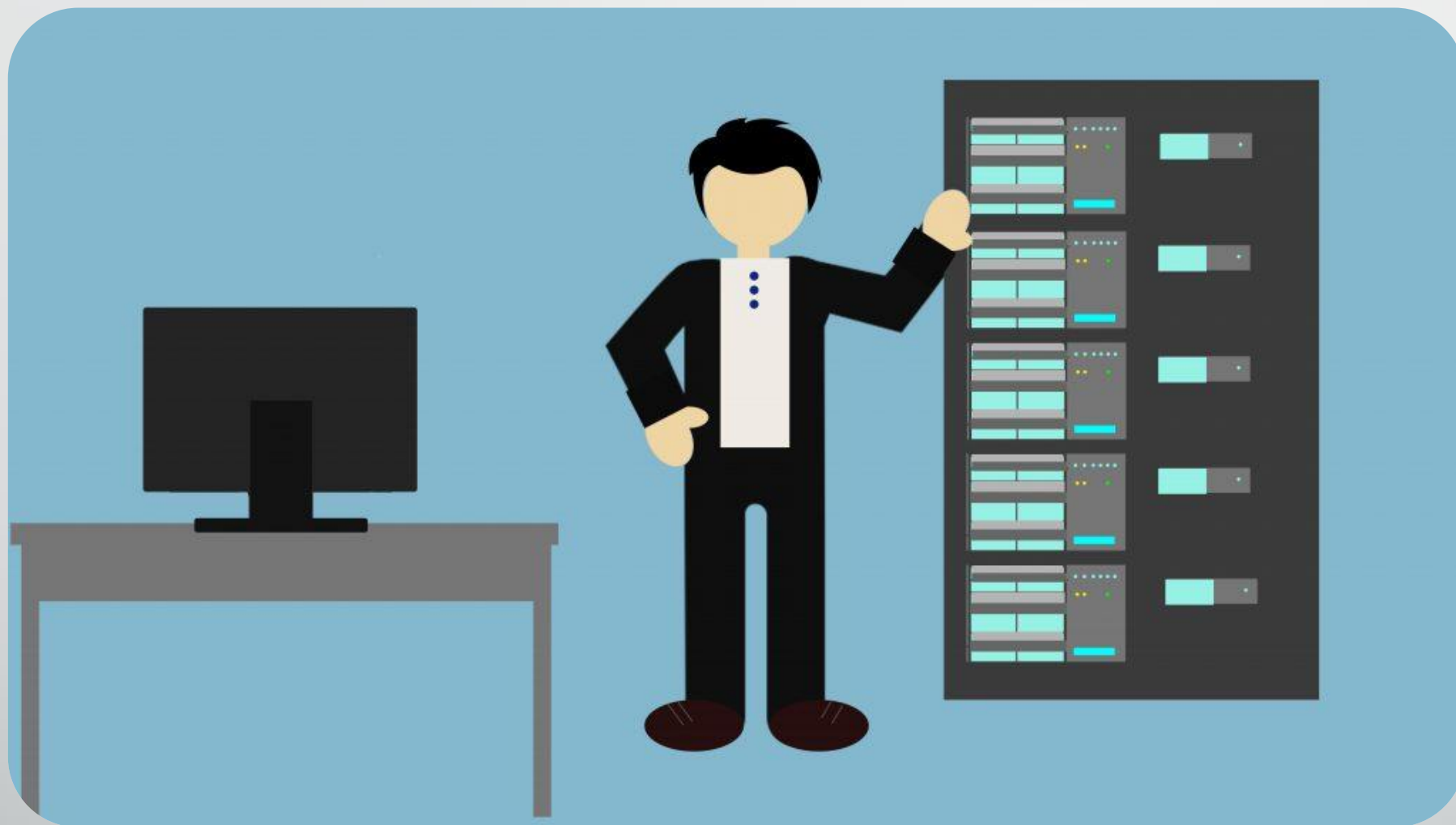
Permite o uso de UPDATE

USAGE

Sinônimo para ``sem privilégios."`

GRANT OPTION

Sinônimo para WITH GRANT OPTION



Exemplo 2 de comando:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO  
Carlos@localhost IDENTIFIED BY 'alguma_senha'  
WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO  
Carlos@localhost IDENTIFIED BY 'alguma_senha'  
WITH GRANT OPTION;
```

```
mysql> SHOW GRANTS FOR 'Carlos'@localhost;
```

Exemplo 3 de comando:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO  
monty@'%' IDENTIFIED BY 'alguma_senha' WITH  
GRANT OPTION;
```

Um super utilizador completo que pode conectar ao servidor de qualquer lugar, mas deve utilizar uma senha '**alguma_senha**' para fazer isto.

Perceba que devemos utilizar instruções **GRANT** para **monty@localhost** e **monty@"%"**.

Retirar privilégios

- REVOKE ALL PRIVILEGES, GRANT OPTION FROM user [, user] ...

```
mysql>REVOKE INSERT ON *.* FROM 'teste'@'localhost';
```

Mostrar os privilégios

`mysql>SHOW GRANTS FOR 'root'@'localhost';`



Apagar um Sql "User"

```
mysql>DROP USER 'carlos'@'localhost';
```

```
mysql>SHOW GRANTS FOR 'Deise'@localhost;
```

```
mysql>DROP USER 'Deise'@'localhost';
```

Comandos SQL - DCL

Para definir as permissões a um usuário utiliza-se o comando GRANT.

Sintaxe:

```
GRANT <permissao> ON <base>.<tabela> TO  
'<user>'@'localhost' IDENTIFIED BY '<senha>'  
WITH GRANT OPTION;
```

Para adicionar permissão de leitura, inserção e modificação:

```
GRANT select, insert, update ON controle.* TO  
'emerson'@'%' WITH GRANT OPTION;
```

Comandos SQL - DCL

E para visualizar as permissões que foram aplicadas:

```
SHOW GRANTS FOR 'emerson'@'%';
```

Exemplo com todos os privilégios:

```
GRANT ALL PRIVILEGES ON controle.* TO  
'emerson'@'%' WITH GRANT OPTION;
```

Comandos SQL - DCL

Para remover uma permissão é utilizado o comando REVOKE.

```
REVOKE DROP, DELETE ON controle.* FROM  
'emerson'@'%';
```

No exemplo acima esta revogando/removendo as seguintes permissões: **DROP e DELETE** do usuário **emerson** sobre a base de dados **controle**.

```
REVOKE ALL PRIVILEGES ON controle.*  
FROM 'emerson'@'%';
```

Exemplo 1 CREATE USER SQL:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'localhost'  
WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'MarcioLemos'@'%'  
WITH GRANT OPTION;
```

```
mysql> CREATE USER 'admin'@'localhost' IDENTIFIED BY  
'admin_pass';
```

```
mysql> CREATE USER 'Cao'@'localhost';
```

```
mysql> GRANT RELOAD,PROCESS ON *.* TO 'Cao'@'localhost';
```


Exemplo 2 CREATE USER SQL:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO custom@localhost  
IDENTIFIED BY 'alguma_senha' WITH GRANT OPTION;
```

```
mysql> SHOW GRANTS FOR custom@localhost;
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP  
ON db_alunos.* TO 'custom'@'localhost';
```

```
mysql> DROP USER 'custom'@'localhost';
```


Exercícios SQL

Tabela de CDs			
Campo	Tipo	Tamanho	Chave
Código	+		*
Nome	A	50	
DataCompra	D		
ValorPago	\$		
LocalCompra	A		
Album	L		

Tabela de Musicas			
Campo	Tipo	Tamanho	Chave
CodigCD	1		*
Numero	1		*
Nome	A	50	
Artista	A	50	
Tempo	T		

- 1) Criar o Banco de dados (Discoteca) e posteriormente as tabelas (tab-cd e tab-musica);
- 2) Definir indices para as tabelas;
- 3) Inserir alguns dados;
- 4) Mostrar todos os cds;
- 5) Mostrar os campos nome e data da compra dos cds ordenados por nome;
- 6) Mostrar o total gasto com a compra dos Cds;
- 7) Mostrar todas as músicas (todos os campos) do cds código 1;
- 8) Mostre a quantidade de músicas cadastradas;
- 9) Mostre a média de duração das músicas cadastradas;
- 10) Mostre uma listagens de músicas em ordem alfabética.

Criar o Banco de dados (DBDiscoteca) :

```
CREATE DATABASE DBdiscoteca;
```

Criar as tabelas (tab_cd e tab_musica):

```
CREATE TABLE tab_cd (codigo int(5), nome char(30), datacompra date, valorpago float, localcompra char(20), album boolean, PRIMARY KEY (codigo) );
```

```
CREATE TABLE tab_musica (codigocd int(5), numero int(5), nome char(50), artista char(50), tempo time, PRIMARY KEY (codigocd, numero) );
```

Exercícios SQL

Inserir alguns dados:

```
INSERT INTO tab_cd VALUES (1,"Balada 2015", 2020/05/01, 18.50, "submarino", True);  
INSERT INTO tab_cd VALUES (2,"Swing Total", 2020/05/02, 20.00, "submarino", TRUE);  
INSERT INTO tab_cd VALUES (3,"Amigos para Sempre", 2020/05/03, 40.00,  
"Americanas", false);
```

```
INSERT INTO tab_musica VALUES (1,1, "Estou De Olho", "Guto Boy", 2.55);  
INSERT INTO tab_musica VALUES (1,2, "Virando o Copo", "Guto Boy", 5);  
INSERT INTO tab_musica VALUES (1,2, "Vivendo Cada Dia", "Guto Boy", 8);
```

Exercícios SQL

Mostrar todos os cds;

```
Select * from tab_cd;
```

**Mostrar os campos nome e data da compra dos cds
ordenados por nome;**

```
Select nome, datacompra From tab_cd Order By nome;
```

Mostrar o total gasto com a compra dos Cds;

Select Sum(valorpago) as Total From tab_cd;

Mostrar todas as músicas (todos os campos) do cdscódigo 1;

Select * From tab_musica Where codigocd=1;

Mostre a quantidade de músicas cadastradas;

Select Count(*) as Qtde From tab_musica;

Mostre a média de duração das músicas cadastradas;

Select AVG(tempo) as Media From tab_musica;

Select AVG(tempo) as Media From tab_musica;

Mostre uma listagens de músicas em ordem alfabética;

Select * From tab_musica Order By nome;

Mostrar todos os cdsque são albuns;

Select * From tab_cd Where album=True;

Mostre o cdque custou mais caro.

Select Max(valorpago) From tab_cd;

Exercício prático Aula 1

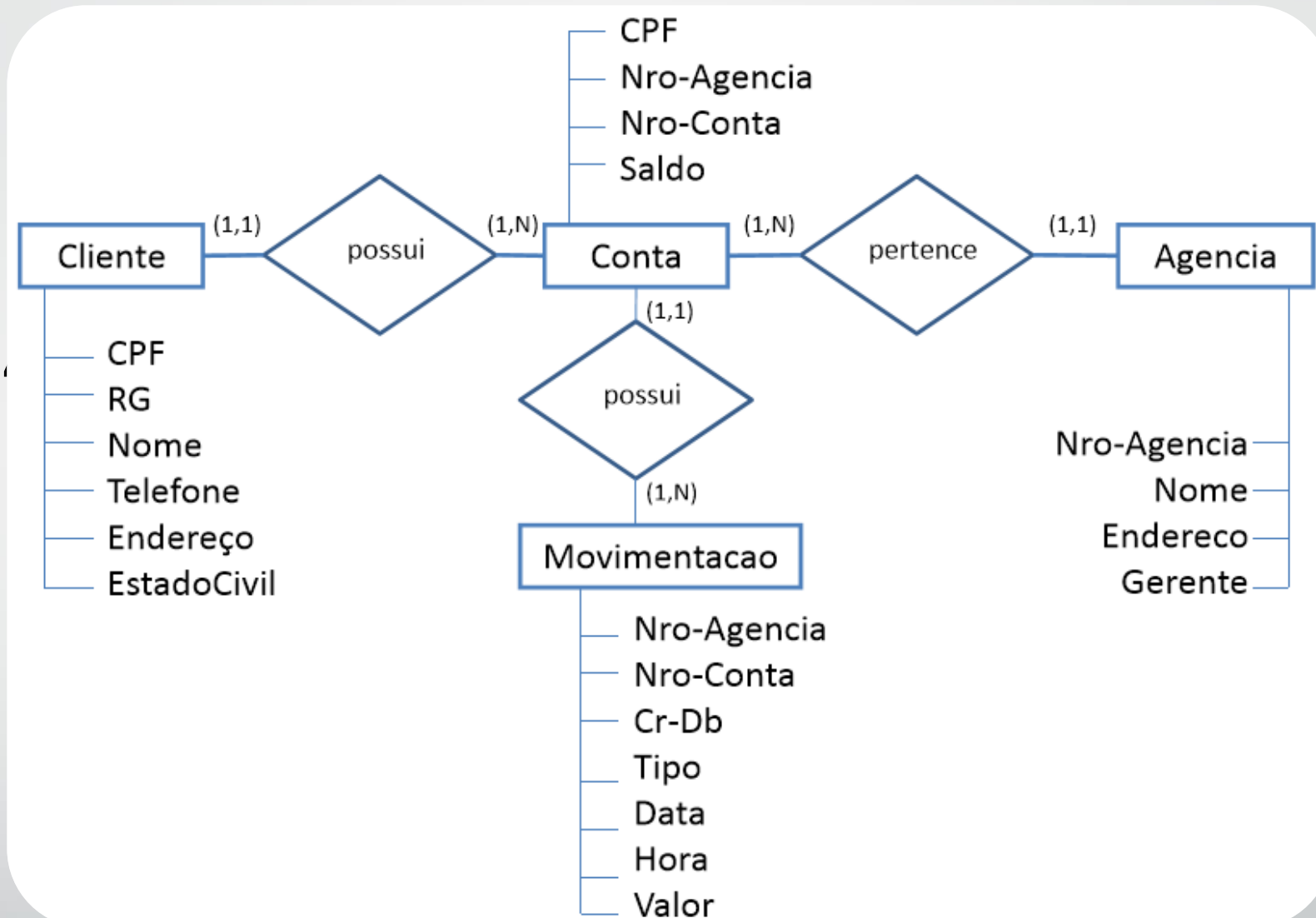
Considerando o
Modelo Conceitual

Ao lado (Crie o BD)!!

Atividade deve ser
entregue

via T2Ke realizada

Individualmente.







Obrigado pela atenção

Sigo à disposição pelo e-mail:

marcio.lemos@senairs.org.br