

PROJETO DE BASES DE DADOS

PARTE 2

Grupo 24

Turno BD22517957L10

INSTITUTO SUPERIOR TÉCNICO LEIC-A



Professor: Taras Lykhenko

Diogo Pereira 86409 - 4 horas - 33%

João Silva 86445 - 4 horas - 33%

Tiago Mesquita 86520 - 4 horas - 33%

schema.sql

```
create table camara (
    numCamara numeric(16) not null unique,
    constraint pk_camara primary
key(numCamara)
);

create table video (
    dataHoraInicio timestamp not null,
    dataHoraFim timestamp not null,
    numCamara numeric(16) not null,
    constraint pk_video primary
key(dataHoraInicio, numCamara),
    constraint fk_camara foreign key(numCamara)
references camara(numCamara) ON DELETE
CASCADE ON UPDATE CASCADE
);

create table segmentoVideo (
    numSegmento numeric(16) not null,
    duracao time not null,
    numCamara numeric(16) not null ,
    dataHoraInicio timestamp not null,
    constraint pk_segmentoVideo primary
key(numSegmento, dataHoraInicio,
numCamara),
    constraint fk_video foreign
key(numCamara,dataHoraInicio) references
video(numCamara,dataHoraInicio) ON DELETE
CASCADE ON UPDATE CASCADE
);
--not in natural habitat
create table local (
    moradaLocal varchar(80) not null,
    constraint pk_local primary key(moradaLocal)
);

create table processoSocorro (
```

```
    numProcessoSocorro numeric(16) not null
unique,
    constraint pk_processoSocorro primary
key(numProcessoSocorro)
    --RI: todo o processo de socorro está
associado a um ou mais EventoEmergencia
);
```

```
create table eventoEmergencia (
    numTelefone numeric(16) not null,
    instanteChamada timestamp not null,
    nomePessoa varchar(80) not null,
    moradaLocal varchar(80) not null,
    numProcessoSocorro numeric(16),
    unique (numTelefone, nomePessoa),
    constraint pk_eventoEmergencia primary
key(numTelefone, instanteChamada),
    constraint fk_local foreign key(moradaLocal)
references local(moradaLocal) ON DELETE
CASCADE ON UPDATE CASCADE,
    constraint fk_processoSocorro foreign
key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

```
create table entidadeMeio (
    nomeEntidade varchar(80) not null unique,
    constraint pk_entidadeMeio primary
key(nomeEntidade)
);
```

```
create table meio (
    numMeio numeric(16) not null,
    nomeMeio varchar(80) not null,
    nomeEntidade varchar(80) not null,
    constraint pk_meio primary key(numMeio,
nomeEntidade),
    constraint fk_entidadeMeio foreign
key(nomeEntidade) references
```

```
entidadeMeio(nomeEntidade) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

```
create table meioCombate (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    constraint pk_meio_c primary key(numMeio,
nomeEntidade),
    constraint fk_meio foreign
key(numMeio,nomeEntidade) references
meio(numMeio,nomeEntidade) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

```
create table meioApoio (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    constraint pk_meio_a primary key(numMeio,
nomeEntidade),
    constraint fk_meio foreign
key(numMeio,nomeEntidade) references
meio(numMeio,nomeEntidade) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

```
create table meioSocorro (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    constraint pk_meio_s primary key(numMeio,
nomeEntidade),
    constraint fk_meio foreign
key(numMeio,nomeEntidade) references
meio(numMeio,nomeEntidade) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

```
create table transporta (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    numVitimas SMALLINT not null,
```

```

    numProcessoSocorro numeric(16) not null,
    constraint pk_transporta primary
key(numMeio, nomeEntidade,
numProcessoSocorro),
    constraint fk_meioSocorro foreign
key(numMeio,nomeEntidade) references
meioSocorro(numMeio,nomeEntidade) ON
DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_processoSocorro foreign
key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

```

create table alocado (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    numHoras SMALLINT not null,
    numProcessoSocorro numeric(16) not null,
    constraint pk_alocado primary key(numMeio,
nomeEntidade, numProcessoSocorro),
    constraint fk_meioApoio foreign
key(numMeio,nomeEntidade) references
meioApoio(numMeio,nomeEntidade) ON
DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_processoSocorro foreign
key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

```

create table acciona (
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    numProcessoSocorro numeric(16) not null,
    constraint pk_acciona primary key(numMeio,
nomeEntidade, numProcessoSocorro),
    constraint fk_meio foreign
key(numMeio,nomeEntidade) references
meio(numMeio,nomeEntidade) ON DELETE
CASCADE ON UPDATE CASCADE,

```

```

    constraint fk_processoSocorro foreign
key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

```

create table coordenador (
    idCoordenador numeric(16) not null unique,
    constraint pk_coordenador primary
key(idCoordenador)
);

```

```

create table vigia (
    moradaLocal varchar(80) not null,
    numCamara numeric(16) not null,
    constraint pk_vigia primary key(moradaLocal,
numCamara),
    constraint fk_local foreign key(moradaLocal)
references local(moradaLocal) ON DELETE
CASCADE ON UPDATE CASCADE,
    constraint fk_camara foreign key(numCamara)
references camara(numCamara) ON DELETE
CASCADE ON UPDATE CASCADE
);

```

```

create table audita (
    idCoordenador numeric(16) not null,
    numMeio numeric(16) not null,
    nomeEntidade varchar(80) not null,
    numProcessoSocorro numeric(16) not null,
    dataHoraInicio time not null CHECK
(dataHoraInicio < dataHoraFim),
    dataHoraFim time not null CHECK
(dataHoraInicio < dataHoraFim),
    dataAuditoria date not null CHECK
(dataAuditoria <= now()),
    texto varchar(8000) not null,
    constraint pk_audita primary
key(idCoordenador, numMeio, nomeEntidade,
numProcessoSocorro),

```

```

    constraint fk_coordenador foreign
key(idCoordenador) references
coordenador(idCoordenador) ON DELETE
CASCADE ON UPDATE CASCADE,
    constraint fk_acciona foreign key(numMeio,
nomeEntidade, numProcessoSocorro)
references acciona(numMeio, nomeEntidade,
numProcessoSocorro) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

create table solicita (
    idCoordenador numeric(16) not null,
    numCamara numeric(16) not null,
    dataHoraInicioVideo timestamp not null,
    dataHoraInicio timestamp not null,
    dataHoraFim timestamp not null CHECK
(dataHoraInicio < dataHoraFim),
    constraint pk_solicita primary
key(idCoordenador, dataHoraInicioVideo,
numCamara),
    constraint fk_coordenador foreign
key(idCoordenador) references
coordenador(idCoordenador) ON DELETE
CASCADE ON UPDATE CASCADE,
    constraint fk_video foreign
key(numCamara,dataHoraInicioVideo)
references video(numCamara,dataHoraInicio)
ON DELETE CASCADE ON UPDATE
CASCADE
);

```

queries.sql

```

--1.
select numProcessoSocorro from (
select numProcessoSocorro,
count((numMeio,nomeEntidade))
from acciona

```

```

group by numProcessoSocorro
) as a
inner join (
select max(count) from (
select numProcessoSocorro,
count((numMeio,nomeEntidade))
from acciona
group by numProcessoSocorro
) as aux
) as b
ON b.max = a.count;
--2.
select nomeEntidade from (
select nomeEntidade,
count(nomeEntidade)
from acciona natural join (
select * from eventoEmergencia
where instanteChamada >= '2018.06.01
00:00:00'
and instanteChamada < '2018.09.01
00:00:00' ) as t
group by nomeEntidade) as a
inner join (
select max(count) from (
select nomeEntidade,
count(nomeEntidade)
from acciona natural join (
select * from eventoEmergencia
where instanteChamada >= '2018.06.01
00:00:00'
and instanteChamada < '2018.09.01
00:00:00' ) as b
group by nomeEntidade
) as c

```

```

) as d
ON d.max = a.count ;
--3.
select distinct numProcessoSocorro from
(
select
numProcessoSocorro,numMeio,nomeEn
tidade from
acciona natural join eventoEmergencia
where instanteChamada >= '2018.01.01
00:00:00'
and instanteChamada < '2019.01.01
00:00:00'
and moradaLocal = 'Oliveira do Hospital'
except
select
numProcessoSocorro,numMeio,nomeEn
tidade from
audita natural join eventoEmergencia
where instanteChamada >= '2018.01.01
00:00:00'
and instanteChamada < '2019.01.01
00:00:00'
and moradaLocal = 'Oliveira do Hospital'
)as w;
--4.
select
count((numSegmento,numCamara))
from vigia natural join video natural join
segmentoVideo
where duracao > '00:01:00'
and dataHoraInicio >= '2018.08.01
00:00:00'

```

```

and dataHoraFim < '2018.09.01
00:00:00'
and moradaLocal = 'Monchique';
--5.
select numMeio, nomeEntidade from
meioCombate natural join acciona
except
select numMeio,nomeEntidade from
meioApoio natural join acciona;
--6.
select distinct nomeEntidade
from acciona d
where not exists (
select numProcessoSocorro
from acciona
except
select numProcessoSocorro
from (acciona natural join meioCombate)
as b
where b.nomeEntidade =
d.nomeEntidade
);

```

add.php

```

if ($type == "local" or $type ==
"entidadeMeio" or $type ==
"processoSocorro") {
$result = $db->prepare("INSERT
INTO $type ($attr1) VALUES (:val);")+
$result->bindParam(':val',
$db_id1);
$result->execute();

```

```

        echo("<p>{$attr1} {$db_id1}
adicionado(a) com sucesso a
{$type}</p>");

        if ($type == "processoSocorro") {

            $result = $db-
>prepare("INSERT INTO
eventoEmergencia VALUES
(:numTel,:instChamada,:nomePessoa,:m
oradaLocal,:numProcessoSocorro);");
            $result->bindParam(':numTel',
$db_id2);
            $result-
>bindParam(':instChamada', $db_id4);
            $result-
>bindParam(':nomePessoa', $db_id3);
            $result-
>bindParam(':moradaLocal', $db_id5);
            $result-
>bindParam(':numProcessoSocorro',
$db_id1);
            $result->execute();
        }
    }
    else if ($type == "meio") {

        $result = $db->prepare("INSERT
INTO $type ($attr1,$attr2,$attr3)
VALUES
(:numMeio,:nomeMeio,:nomeEntidade);"
);

```

```

        $result->bindParam(':numMeio',
$db_id1);
        $result->bindParam(':nomeMeio',
$db_id2);
        $result-
>bindParam(':nomeEntidade', $db_id3);
        $result->execute();

        $combate =
isset($_REQUEST['comb']) ?
$_REQUEST['comb'] : "";
        $socorro =
isset($_REQUEST['sos']) ?
$_REQUEST['sos'] : "";
        $apoio =
isset($_REQUEST['apoio']) ?
$_REQUEST['apoio'] : "";
        echo("<p>{$db_id3} - {$db_id1}
adicionado(a) com sucesso a
$type</p>");

        if ($combate=="yes") {
            $result = $db-
>prepare("INSERT INTO meioCombate
($attr1,$attr3) VALUES
(:numMeio,:nomeEntidade);");
            $result-
>bindParam(':numMeio', $db_id1);
            $result-
>bindParam(':nomeEntidade', $db_id3);
            $result->execute();
            echo("<p>{$db_id3} -
{$db_id1} adicionado(a) com sucesso a
meioCombate</p>");

```

```

        }
        if ($socorro=="yes") {
            $result = $db-
>prepare("INSERT INTO meioSocorro
($attr1,$attr3) VALUES
(:numMeio,:nomeEntidade);");
            $result-
>bindParam(':numMeio', $db_id1);
            $result-
>bindParam(':nomeEntidade', $db_id3);
            $result->execute();
            echo("<p>{$db_id3} -
{$db_id1} adicionado(a) com sucesso a
meioSocorro</p>");
        }
        if ($apoio=="yes") {
            $result = $db-
>prepare("INSERT INTO
meioApoio($attr1,$attr3) VALUES
(:numMeio,:nomeEntidade);");
            $result-
>bindParam(':numMeio', $db_id1);
            $result-
>bindParam(':nomeEntidade', $db_id3);
            $result->execute();
            echo("<p>{$db_id3} -
{$db_id1} adicionado(a) com sucesso a
meioApoio</p>");
        }
    }

    else if ($type ==
"eventoemergencia") {

```

```

        $result = $db->prepare("INSERT
        INTO $type VALUES
        (:val1,:val2,:val3,:val4,:val5);");
        $result->bindParam(':val1',
        $db_id1);
        $result->bindParam(':val2',
        $db_id3);
        $result->bindParam(':val3',
        $db_id2);
        $result->bindParam(':val4',
        $db_id4);
        $result->bindParam(':val5',
        $db_id5);

        $result->execute();
        echo("<p>{$db_id4} reportado
        por {$db_id3} registrado(a) com sucesso
        para o processo {$db_id3}</p>");
    }

    else if ($type == "acciona") {
        $result = $db->prepare("INSERT
        INTO acciona
        (numMeio,nomeEntidade,numProcessoS
        ocorro) VALUES (:val1,:val2,:val3);");

        $result->bindParam(':val1',
        $db_id1);
        $result->bindParam(':val2',
        $db_id2);
        $result->bindParam(':val3',
        $db_id3);
        $result->execute();
    }

```

```

        echo("<p>{$db_id2} - {$db_id1}
        accionado(a) com sucesso para o
        processo {$db_id3}</p>");
    }

```

remove.php

```

        if ($type == local or $type ==
        entidadeMeio or $type ==
        processoSocorro) {

            $result = $db->prepare("DELETE
            FROM $type WHERE $attr1 = :val;");
            $result->bindParam(':val',
            $db_id1);
            $result->execute();
            echo("<p>{$attr1} {$db_id1}
            removido(a) com sucesso a
            {$type}</p>");
            //}
        }

        else if ($type == "meio" or $type ==
        "eventoEmergencia" ) {
            $result = $db-
            >prepare("DELETE FROM $type
            WHERE $attr1 = :val1 and $attr3
            = :val2;");
            $result->bindParam(':val1',
            $db_id1);
            $result->bindParam(':val2',
            $db_id3);
            $result->execute();
            echo("<p>{$db_id3} -
            {$db_id1} removido(a) com sucesso a
            $type</p>");
        }
    }

```

```

        //}
    }
    echo("<a href={$back} ><input
    type='submit' value='Retornar'></a>");
}

```

update.php

```

$result = $db->prepare("UPDATE $type
SET $attr1 = :numMeio, $attr2
= :nomeMeio , $attr3 = :nomeEntidade
WHERE $attr1 = :oldNum and
$attr3=:oldEnt ;");
$result->bindParam(':oldNum',
$db_id1);
$result->bindParam(':oldEnt',
$db_id3);
$result->bindParam(':numMeio',
$db_id4);
$result->bindParam(':nomeMeio',
$db_id5);
$result-
>bindParam(':nomeEntidade', $db_id6);
$result->execute();
$combate =
isset($_REQUEST['comb']) ?
$_REQUEST['comb'] : "";
$socorro =
isset($_REQUEST['sos']) ?
$_REQUEST['sos'] : "";

```

```

$apoio =
isset($_REQUEST['apoio']) ?
$_REQUEST['apoio'] : "";
    echo("<p>{$db_id6} - {$db_id4}
adicionado(a) com sucesso a
$type</p>");

    if ($combate=="yes") {
        $result = $db->prepare("INSERT
        INTO meioCombate ($attr1,$attr3)
        VALUES (:numMeio,:nomeEntidade);");
        $result->bindParam(':numMeio',
        $db_id4);
        $result-
        >bindParam(':nomeEntidade', $db_id6);
        $result->execute();
        echo("<p>{$db_id6} - {$db_id4}
adicionado(a) com sucesso a
meioCombate</p>");
    }
    if ($socorro=="yes") {
        $result = $db->prepare("INSERT
        INTO meioSocorro ($attr1,$attr3)
        VALUES (:numMeio,:nomeEntidade);");
        $result->bindParam(':numMeio',
        $db_id4);
        $result-
        >bindParam(':nomeEntidade', $db_id6);
        $result->execute();
        echo("<p>{$db_id6} - {$db_id4}
adicionado(a) com sucesso a
meioSocorro</p>");
    }

```

```

    }
    if ($apoio=="yes") {
        $result = $db->prepare("INSERT
        INTO meioApoio($attr1,$attr3) VALUES
        (:numMeio,:nomeEntidade);");
        $result->bindParam(':numMeio',
        $db_id4);
        $result-
        >bindParam(':nomeEntidade', $db_id6);
        $result->execute();
        echo("<p>{$db_id6} - {$db_id4}
adicionado(a) com sucesso a
meioApoio</p>");
    }

```

list.php

```

if ($type == processoSocorro) {
    $result = $db->prepare("SELECT
    * FROM acciona WHERE
    $attr1=:val ;");
    $result->bindParam(':val',
    $db_id1);
    $result->execute();
    echo("<h3>Lista de
    meios</h3><table border='1'\>\n");
    echo("<tr><td>Numero</td><td>Entidad
    e</td><td>Remover</td></tr>\n");
    foreach($result as $row)
    {
        echo("<tr><td>");
    }
}

```

```

        echo($row['nummeio']);
        echo("</td><td>");
    }
    echo($row['nomeentidade']);
    echo("</td><td><a
    href='\"remove.php?back=meio.php&type
    =meio&attr1=numMeio&attr3=nomeEnti
    dade&db_id1={$row['nummeio']}&db_i
    d3={$row['nomeentidade']}\>delete</a>
    </td></tr>\n");
    }
    echo("</table>\n<p></p>") }
    else if ($type == "local") {
        $result = $db->prepare("SELECT
        nummeio,nomeentidade FROM acciona natural
        join eventoemergencia natural join meiosocorro
        WHERE $attr1=:val ;");
        $result->bindParam(':val', $db_id1);
        $result->execute();
        echo("<h3>Lista de meios</h3><table
        border='1'\>\n");
    }
    echo("<tr><td>Numero</td><td>Entidade</td><td>
    Remover</td></tr>\n");
    foreach($result as $row){
        echo("<tr><td>");
        echo($row['nummeio']);
        echo("</td><td>");
        echo($row['nomeentidade']);
        echo("</td><td><a
        href='\"remove.php?back=locais.php&type=meio&att
        r1=numMeio&attr3=nomeEntidade&db_id1={$r
        ow['nummeio']}&db_id3={$row['nomeentidade']
        }\>delete</a></td></tr>\n");
    }
    echo("</table>\n<p></p>"); }
}

```