Configurar o ODBC em ISE 2.3 com base de dados Oracle

Índice

Introdução

Este documento descreve como configurar o Identity Services Engine (ISE) com base de dados Oracle para a autenticação ISE usando a conectividade de bases de dados aberto (ODBC).

A autenticação da conectividade de bases de dados aberto (ODBC) exige o ISE poder buscar uma senha do usuário do texto simples. A senha pode ser cifrada no base de dados, mas tem que ser decifrada pelo procedimento armazenado.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Identity Services Engine 2.3
- Base de dados e conceitos ODBC
- Oracle

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Identity Services Engine 2.3.0.298
- Centos 7
- Base de dados Oracle 12.2.0.1.0
- Colaborador 4.1.5 do Oracle SQL

Configurar

> **Nota:** Trate os procedimentos SQL apresentados neste documento como exemplos. Esta não é um oficial e uma maneira recomendada de configuração do Oracle DB. Assegure-se de que você compreenda o resultado e o impacto de cada pergunta que SQL você compromete.

Etapa 1. Configuração básica do Oracle

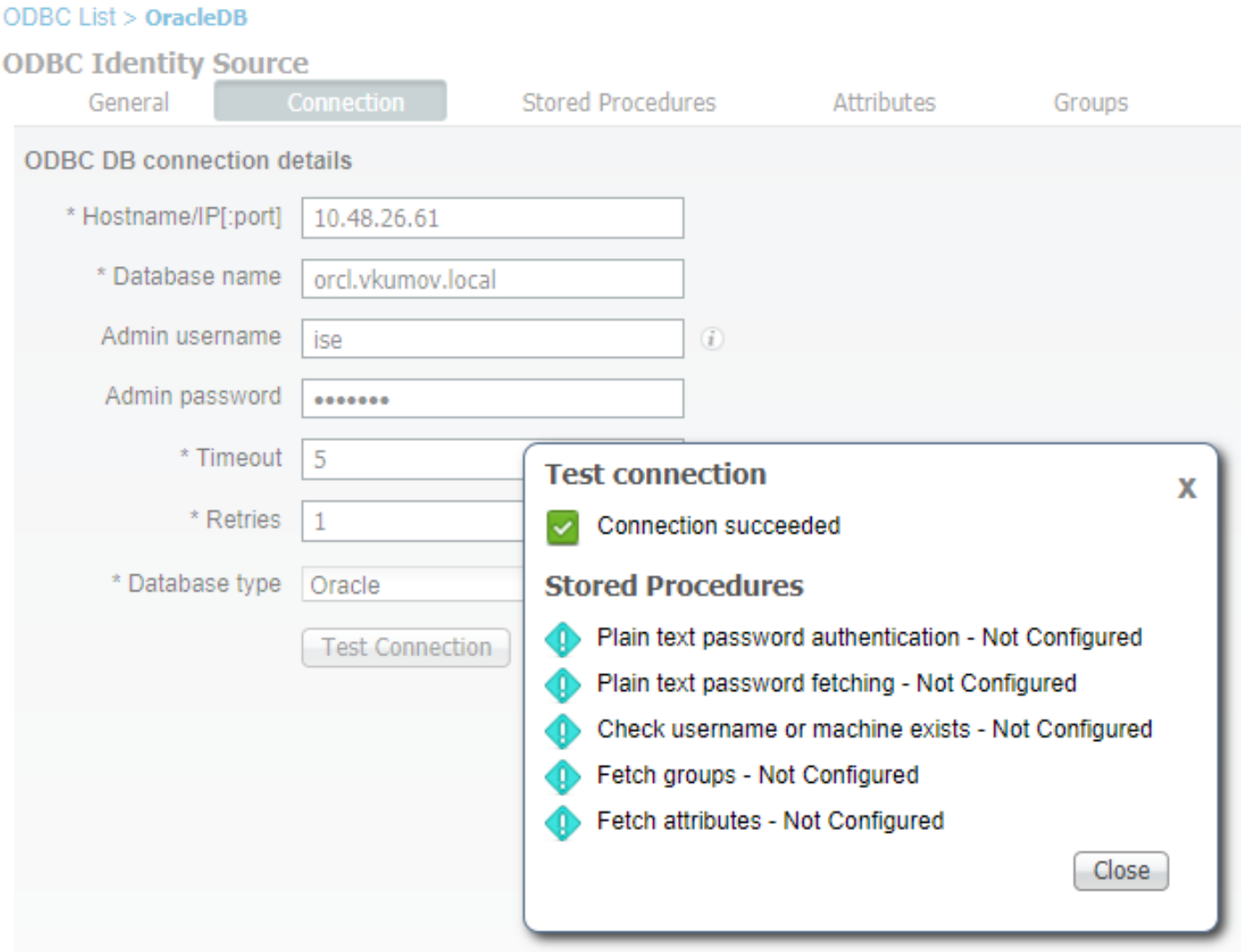
Neste exemplo o Oracle foi configurado com seguintes parâmetros:

- Nome DB: **ORCL**
- Nome do serviço: **orcl.vkumov.local**
- Porta: **1521** (padrão)
- Criado esclareça o ISE com **ise** username

Você deve configurar seu Oracle antes de continuar mais.

Etapa 2. Configuração básica ISE

Crie uma fonte da identidade ODBC na *administração > fonte externo da identidade > ODBC* e conexão de teste:



Nota: O ISE conecta ao Oracle usando o nome do serviço, daqui o campo do [Database name] deve ser enchido com o nome do serviço que existe no Oracle, nome não de SID (ou DB). Devido aos pontos do erro CSCvf06497 (.) não pode ser usado no campo do [Database name]. Este erro é fixado em ISE 2.3.

Etapa 3. Configurar a autenticação de usuário

A autenticação ISE ao ODBC usa procedimentos armazenados. É possível selecionar o tipo de procedimentos. Neste exemplo nós usamos recordsets como o retorno.

Para outros procedimentos, refira o guia do administrador do Cisco Identity Services Engine, a liberação 2.3

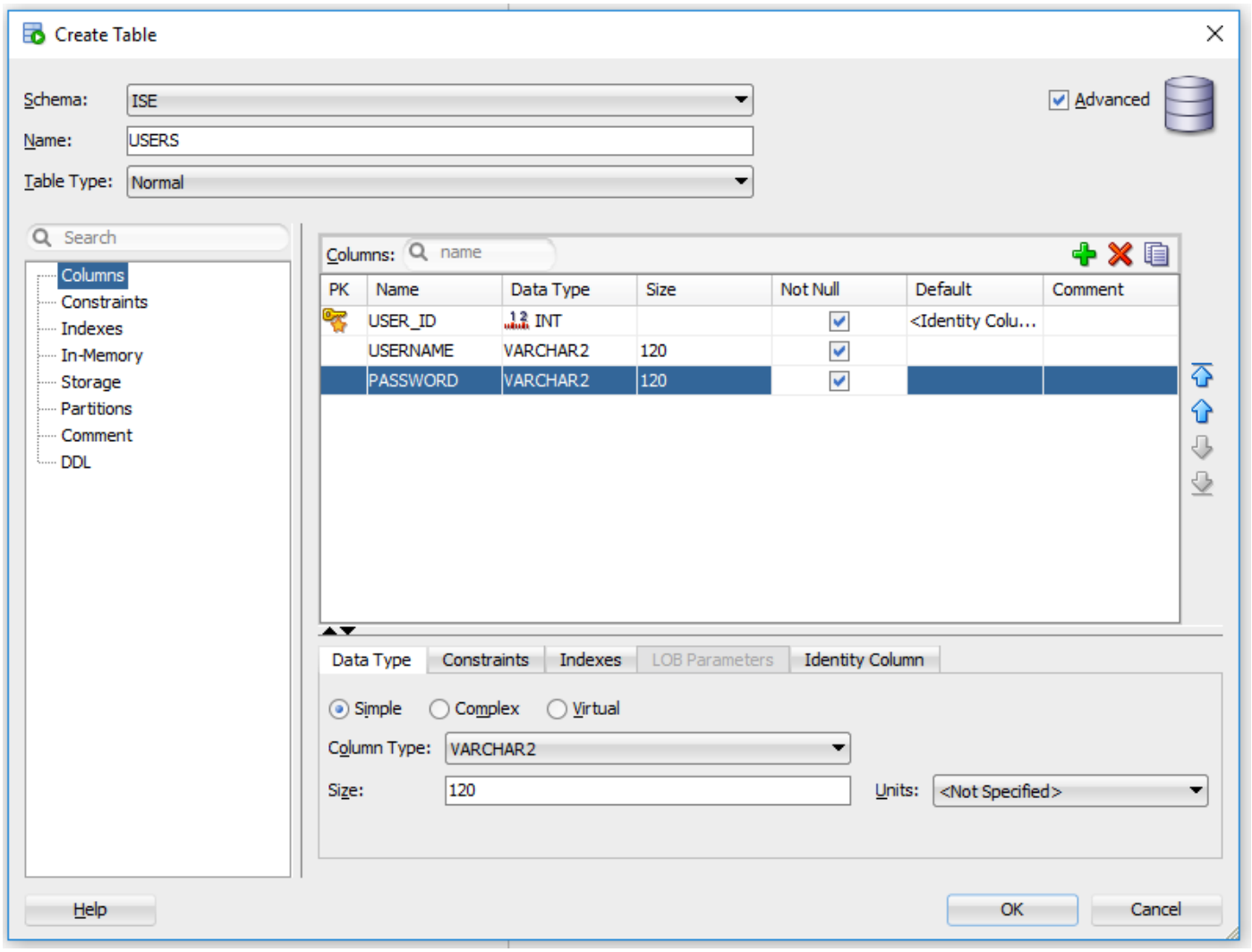
Dica: É possível retornar parâmetros Nomeados em vez do resultSet. É apenas um tipo diferente de saída, funcionalidade é o mesmo.

1. Crie a tabela com as credenciais dos usuários. Certifique-se que você ajustou os ajustes da identidade no **chave principal.**

```
-------------------------------------------------------
--  DDL for Table USERS
```

```
--------------------------------------------------------

  CREATE TABLE "ISE"."USERS"
   ("USER_ID" NUMBER(*,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER  NOCYCLE  NOKEEP
NOSCALE ,
"USERNAME" VARCHAR2(120 BYTE),
"PASSWORD" VARCHAR2(120 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index USERS_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "ISE"."USERS_PK" ON "ISE"."USERS" ("USER_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  Constraints for Table USERS
--------------------------------------------------------

  ALTER TABLE "ISE"."USERS" MODIFY ("USER_ID" NOT NULL ENABLE);
  ALTER TABLE "ISE"."USERS" MODIFY ("USERNAME" NOT NULL ENABLE);
  ALTER TABLE "ISE"."USERS" MODIFY ("PASSWORD" NOT NULL ENABLE);
  ALTER TABLE "ISE"."USERS" ADD CONSTRAINT "USERS_PK" PRIMARY KEY ("USER_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
```

Ou do colaborador GUI SQL:



2. Adicionar usuários

```
INSERT INTO "ISE"."USERS" (USERNAME, PASSWORD) VALUES ('alice', 'password1')
INSERT INTO "ISE"."USERS" (USERNAME, PASSWORD) VALUES ('bob', 'password1')
INSERT INTO "ISE"."USERS" (USERNAME, PASSWORD) VALUES ('admin', 'password1')
```

3. Crie um procedimento para a autenticação de senha do texto simples (usada para o método interno PAP, EAP-GTC, o TACACS)

```
create or replace function ISEAUTH_R
(
  ise_username IN VARCHAR2,
  ise_userpassword IN VARCHAR2
) return sys_refcursor AS
BEGIN
  declare
    c integer;
    resultSet SYS_REFCURSOR;
  begin
    select count(*) into c from USERS where USERS.USERNAME = ise_username and USERS.PASSWORD =
ise_userpassword;
    if c > 0 then
      open resultSet for select 0 as code, 11, 'good user', 'no error' from dual;
    ELSE
      open resultSet for select 3, 0, 'odbc','ODBC Authen Error' from dual;
    END IF;
    return resultSet;
  end;
END ISEAUTH_R;
```

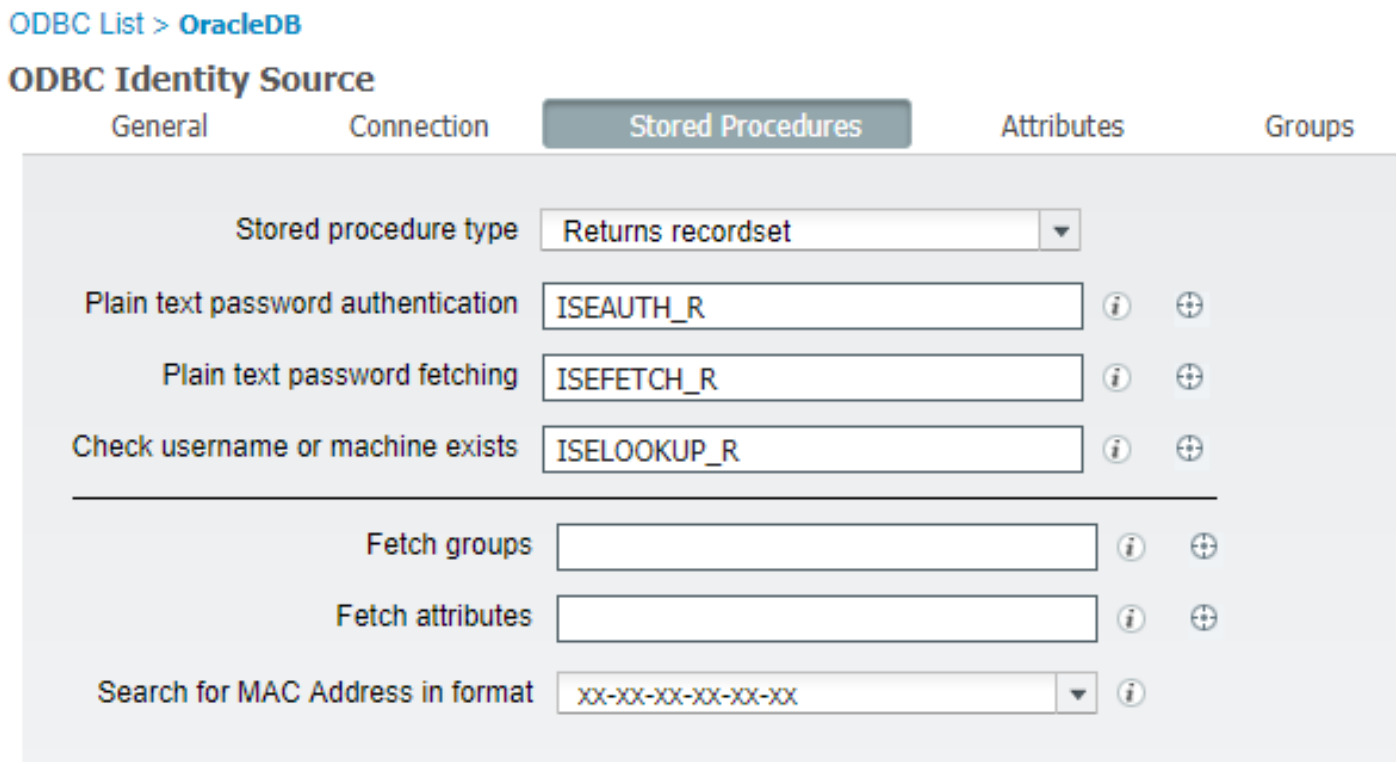
4. Crie um procedimento para a busca da senha do texto simples (usada para a RACHADURA, MSCHAPv1/v2, EAP-MD5, PULO, método interno do EAP-MSCHAPv2, o TACACS)

```
create or replace function ISEFETCH_R
(
  ise_username IN VARCHAR2
) return sys_refcursor AS
BEGIN
  declare
    c integer;
    resultSet SYS_REFCURSOR;
  begin
    select count(*) into c from USERS where USERS.USERNAME = ise_username;
    if c > 0 then
      open resultSet for select 0, 11, 'good user', 'no error', password from USERS where
USERS.USERNAME = ise_username;
      DBMS_OUTPUT.PUT_LINE('found');
    ELSE
      open resultSet for select 3, 0, 'odbc','ODBC Authen Error' from dual;
      DBMS_OUTPUT.PUT_LINE('not found');
    END IF;
    return resultSet;
  end;
END;
```

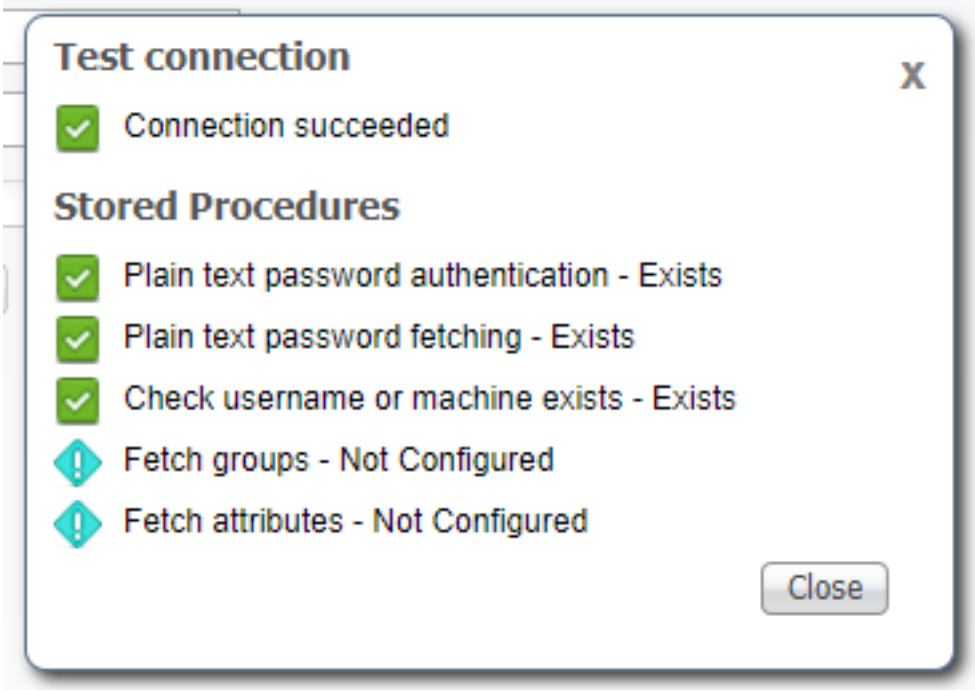
5. Crie um procedimento para o username da verificação ou a máquina existe (usado para o MAB, rápido reconecte do PEAP, EAP-FAST e do EAP-TTLS)

```
create or replace function ISELOOKUP_R
(
  ise_username IN VARCHAR2
) return sys_refcursor AS
BEGIN
  declare
    c integer;
    resultSet SYS_REFCURSOR;
  begin
    select count(*) into c from USERS where USERS.USERNAME = ise_username;
    if c > 0 then
      open resultSet for select 0, 11, 'good user', 'no error' from USERS where USERS.USERNAME =
ise_username;
    ELSE
      open resultSet for select 3, 0, 'odbc','ODBC Authen Error' from dual;
    END IF;
    return resultSet;
  end;
END;
```

6. Configurar procedimentos no ISE e salvar



7. Vá para trás à aba da conexão e clique o botão do [Test Connection]



Etapa 4. Configurar a recuperação do grupo

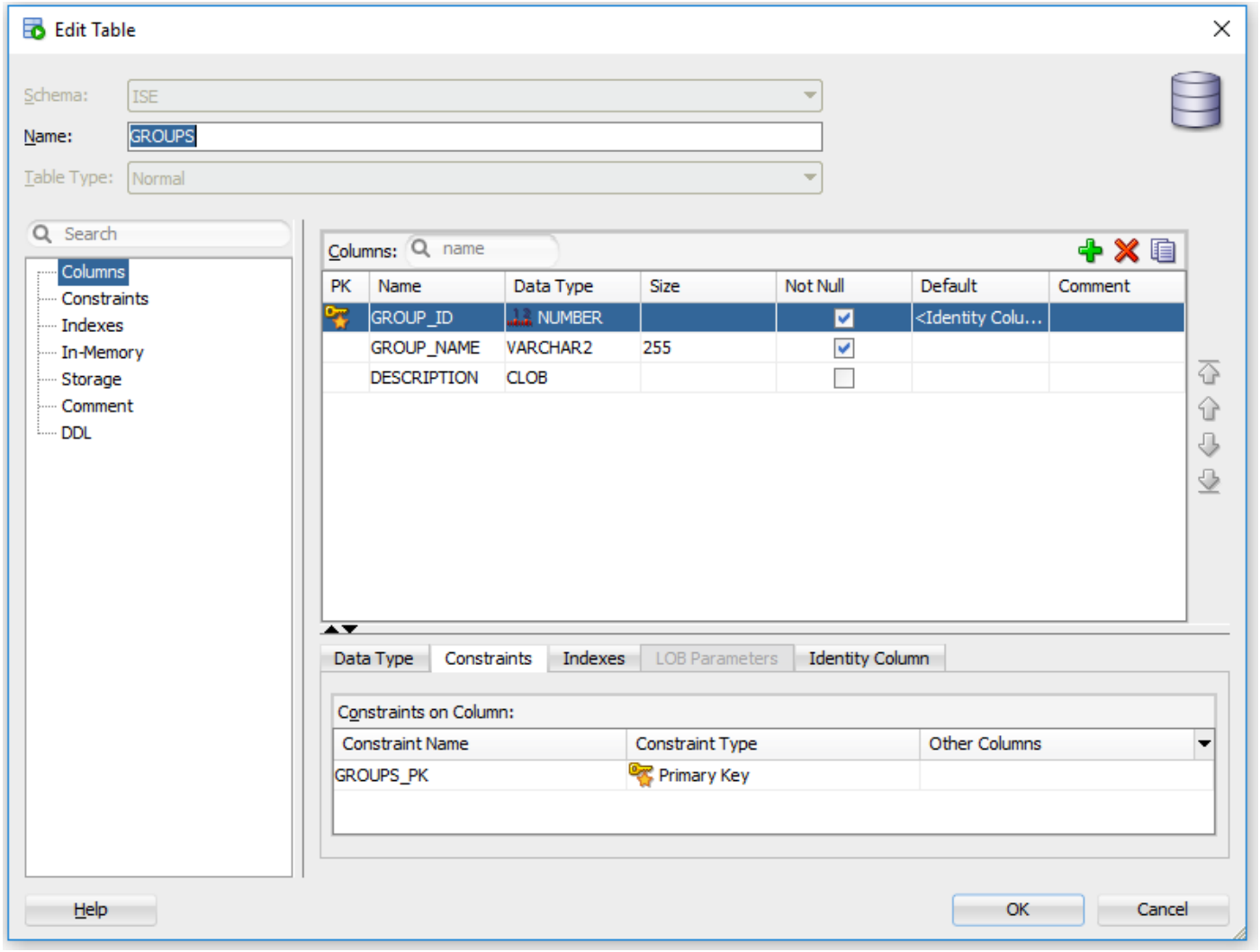
1. Crie as tabelas que contêm grupos de usuário e as outras usadas para muito-à-muitos o mapeamento

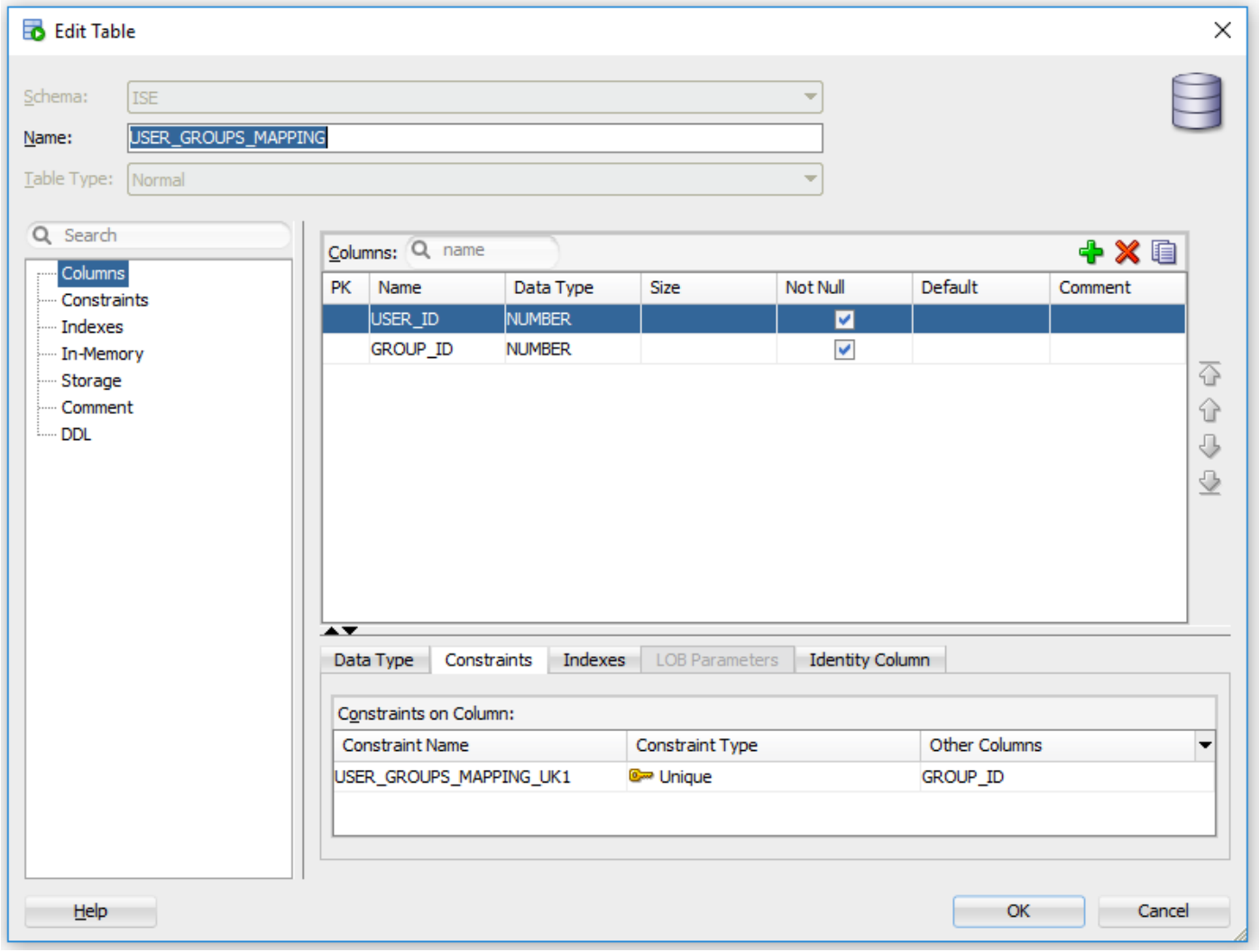
```
--------------------------------------------------------
--  DDL for Table GROUPS
--------------------------------------------------------

  CREATE TABLE "ISE"."GROUPS"
   ("GROUP_ID" NUMBER(*,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE
99999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER  NOCYCLE  NOKEEP
```

```
NOSCALE ,
"GROUP_NAME" VARCHAR2(255 BYTE),
"DESCRIPTION" CLOB
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"
 LOB ("DESCRIPTION") STORE AS SECUREFILE (
  TABLESPACE "USERS" ENABLE STORAGE IN ROW CHUNK 8192
  NOCACHE LOGGING  NOCOMPRESS  KEEP_DUPLICATES
  STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)) ;
--------------------------------------------------------
--  DDL for Table USER_GROUPS_MAPPING
--------------------------------------------------------

  CREATE TABLE "ISE"."USER_GROUPS_MAPPING"
   ("USER_ID" NUMBER(*,0),
"GROUP_ID" NUMBER(*,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index GROUPS_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "ISE"."GROUPS_PK" ON "ISE"."GROUPS" ("GROUP_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index USER_GROUPS_MAPPING_UK1
--------------------------------------------------------

  CREATE UNIQUE INDEX "ISE"."USER_GROUPS_MAPPING_UK1" ON "ISE"."USER_GROUPS_MAPPING" ("USER_ID",
"GROUP_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  Constraints for Table GROUPS
--------------------------------------------------------

  ALTER TABLE "ISE"."GROUPS" MODIFY ("GROUP_ID" NOT NULL ENABLE);
  ALTER TABLE "ISE"."GROUPS" MODIFY ("GROUP_NAME" NOT NULL ENABLE);
  ALTER TABLE "ISE"."GROUPS" ADD CONSTRAINT "GROUPS_PK" PRIMARY KEY ("GROUP_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
```

```
--  Constraints for Table USER_GROUPS_MAPPING
------------------------------------------------------

  ALTER TABLE "ISE"."USER_GROUPS_MAPPING" MODIFY ("USER_ID" NOT NULL ENABLE);
  ALTER TABLE "ISE"."USER_GROUPS_MAPPING" MODIFY ("GROUP_ID" NOT NULL ENABLE);
  ALTER TABLE "ISE"."USER_GROUPS_MAPPING" ADD CONSTRAINT "USER_GROUPS_MAPPING_UK1" UNIQUE
("USER_ID", "GROUP_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
```

Do GUI:





2. Adicionar grupos e mapeamentos, de modo que **Alice** e o **prumo** pertençam para agrupar **usuários** e o admin pertença para agrupar **Admins**

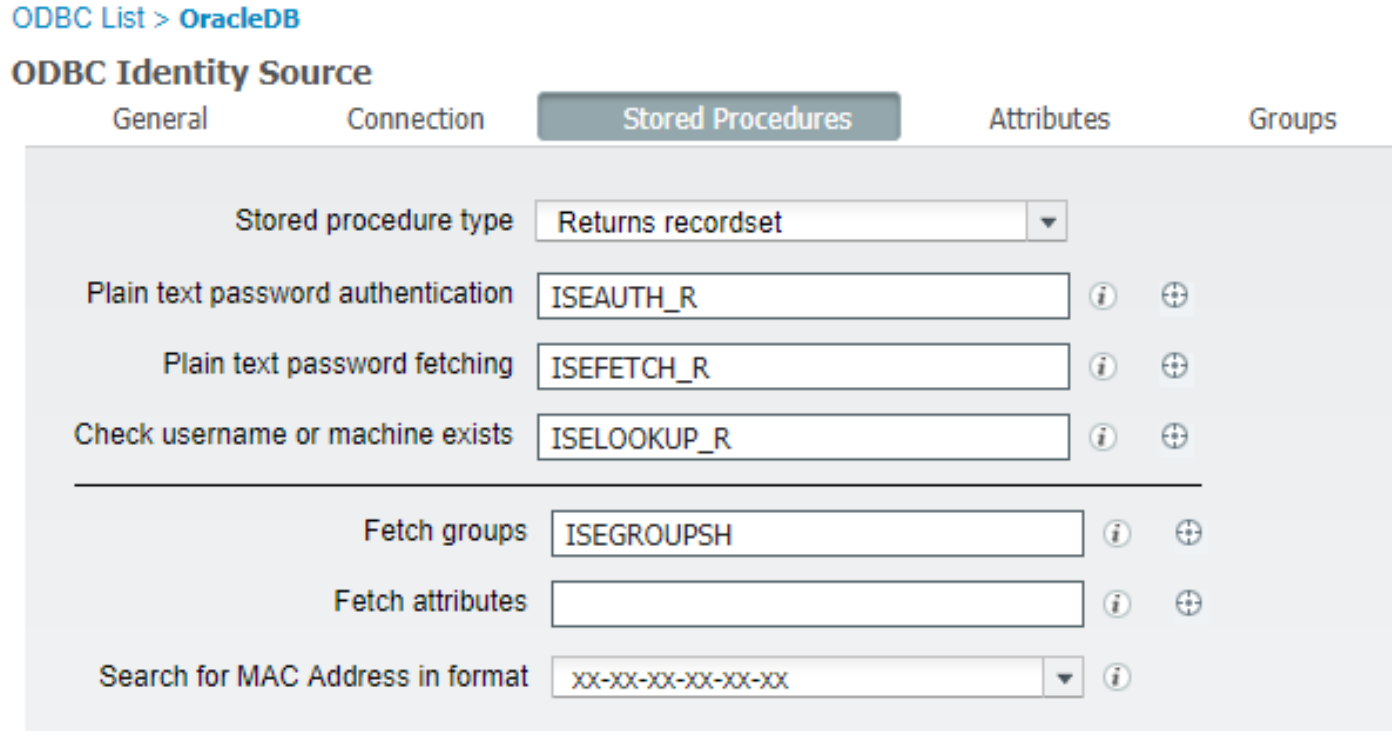
```
-- Adding groups
INSERT INTO "ISE"."GROUPS" (GROUP_NAME, DESCRIPTION) VALUES ('Admins', 'Group for
administrators')
INSERT INTO "ISE"."GROUPS" (GROUP_NAME, DESCRIPTION) VALUES ('Users', 'Corporate users')

-- Alice and Bob are users
INSERT INTO "ISE"."USER_GROUPS_MAPPING" (USER_ID, GROUP_ID) VALUES ('1', '2')
INSERT INTO "ISE"."USER_GROUPS_MAPPING" (USER_ID, GROUP_ID) VALUES ('2', '2')

-- Admin is in Admins group
INSERT INTO "ISE"."USER_GROUPS_MAPPING" (USER_ID, GROUP_ID) VALUES ('3', '1')
```

3. Crie um procedimento da recuperação do grupo. Retorna todos os grupos se o username é “*”

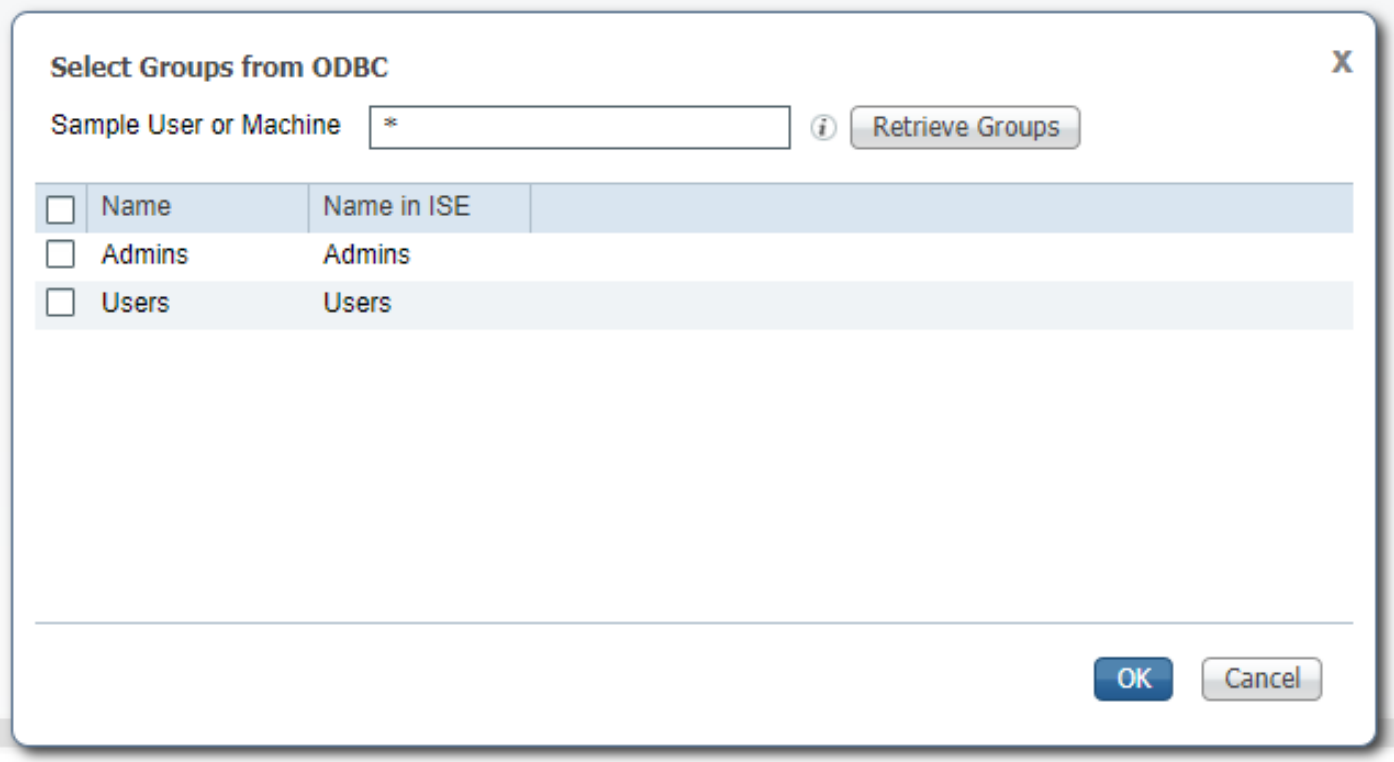
```
create or replace function ISEGROUPSH
(
  ise_username IN VARCHAR2,
  ise_result OUT int
) return sys_refcursor as
BEGIN
  declare
    c integer;
    userid integer;
    resultSet SYS_REFCURSOR;
  begin
    IF ise_username = '*' then
      ise_result := 0;
```

```
      open resultSet for select GROUP_NAME from GROUPS;
    ELSE
      select count(*) into c from USERS where USERS.USERNAME = ise_username;
      select USER_ID into userid from USERS where USERS.USERNAME = ise_username;
      IF c > 0 then
          ise_result := 0;
          open resultSet for select GROUP_NAME from GROUPS where GROUP_ID IN ( SELECT m.GROUP_ID
from USER_GROUPS_MAPPING m where m.USER_ID = userid );
      ELSE
          ise_result := 3;
          open resultSet for select 0 from dual where 1=2;
      END IF;
    END IF;
    return resultSet;
  end;
END ;
```

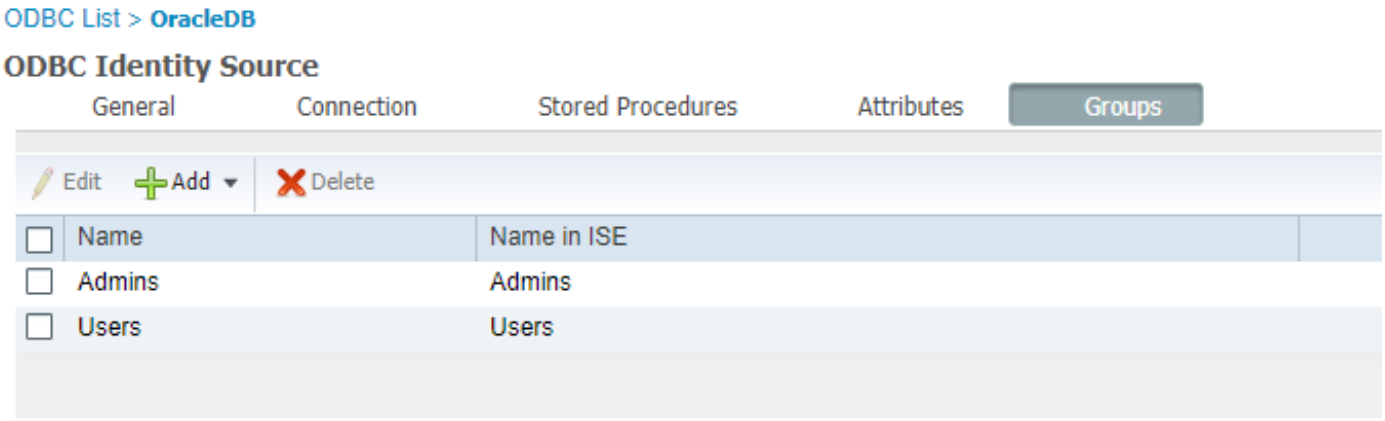
4. Trace-o **para buscar grupos**



5. Busque os grupos e adicionar-los na **fonte da identidade ODBC**



Select precisou grupos e [OK] do clique, aparecerão na aba dos **"grupos"**



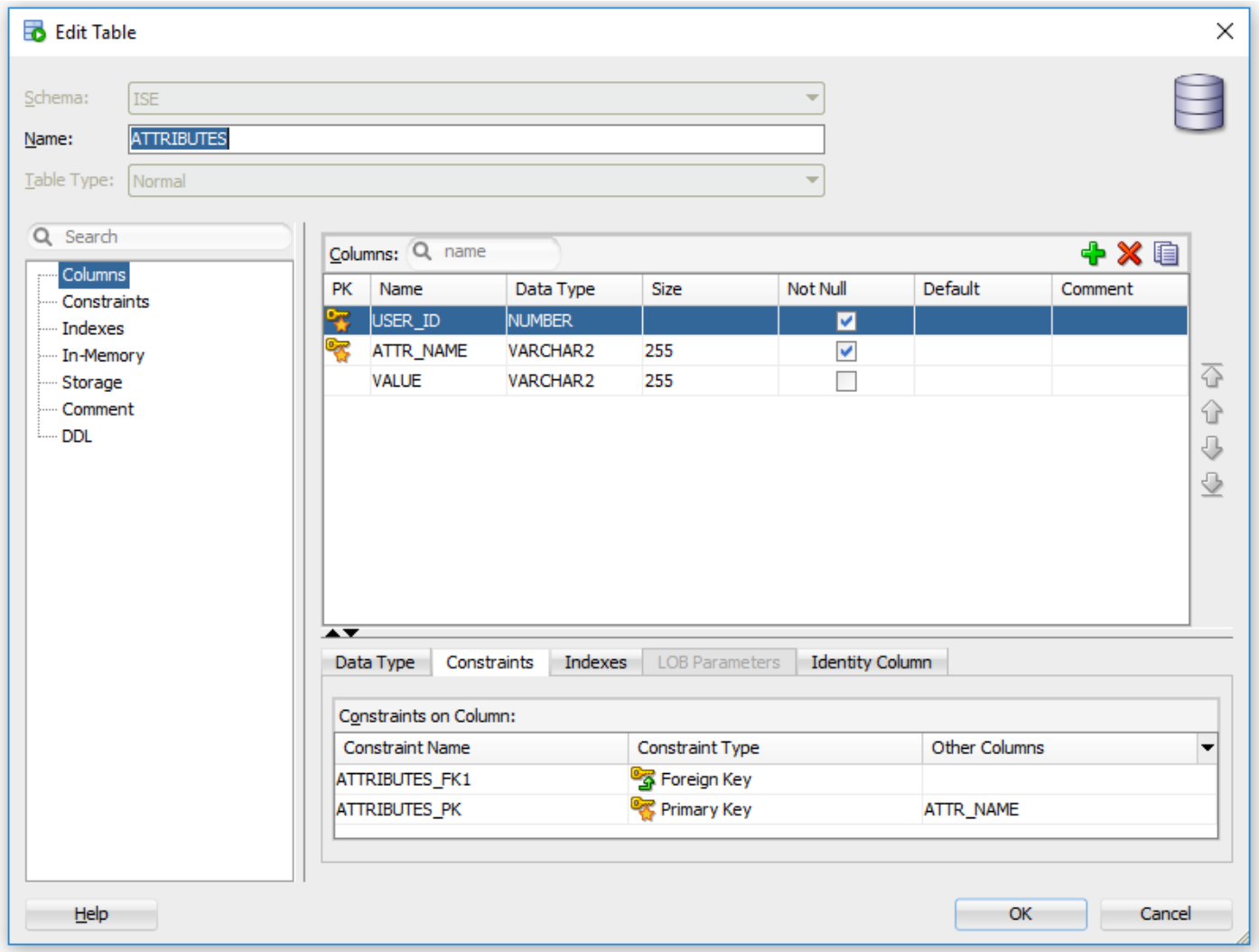
Etapa 5. Configurar a recuperação dos atributos

1. A fim simplificar este exemplo, uma tabela lisa é usada para atributos

```
--------------------------------------------------------
--  DDL for Table ATTRIBUTES
--------------------------------------------------------

  CREATE TABLE "ISE"."ATTRIBUTES"
   ("USER_ID" NUMBER(*,0),
"ATTR_NAME" VARCHAR2(255 BYTE),
"VALUE" VARCHAR2(255 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index ATTRIBUTES_PK
```

```
--------------------------------------------------------

  CREATE UNIQUE INDEX "ISE"."ATTRIBUTES_PK" ON "ISE"."ATTRIBUTES" ("ATTR_NAME", "USER_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  Constraints for Table ATTRIBUTES
--------------------------------------------------------

  ALTER TABLE "ISE"."ATTRIBUTES" MODIFY ("USER_ID" NOT NULL ENABLE);
  ALTER TABLE "ISE"."ATTRIBUTES" MODIFY ("ATTR_NAME" NOT NULL ENABLE);
  ALTER TABLE "ISE"."ATTRIBUTES" ADD CONSTRAINT "ATTRIBUTES_PK" PRIMARY KEY ("ATTR_NAME",
"USER_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
```

Do GUI:



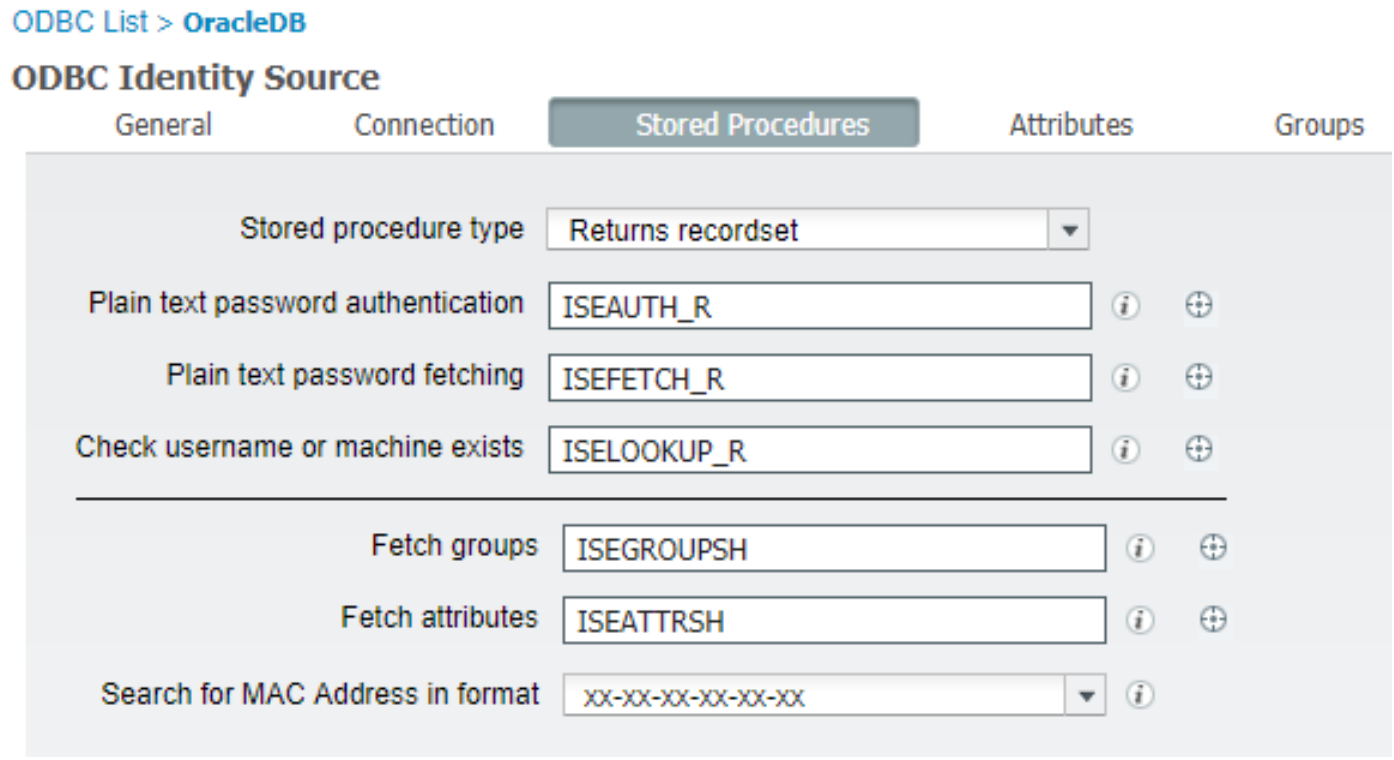
2. Crie alguns atributos para usuários

```
INSERT INTO "ISE"."ATTRIBUTES" (USER_ID, ATTR_NAME, VALUE) VALUES ('3', 'SecurityLevel', '15')
INSERT INTO "ISE"."ATTRIBUTES" (USER_ID, ATTR_NAME, VALUE) VALUES ('1', 'SecurityLevel', '5')
INSERT INTO "ISE"."ATTRIBUTES" (USER_ID, ATTR_NAME, VALUE) VALUES ('2', 'SecurityLevel', '10')
```

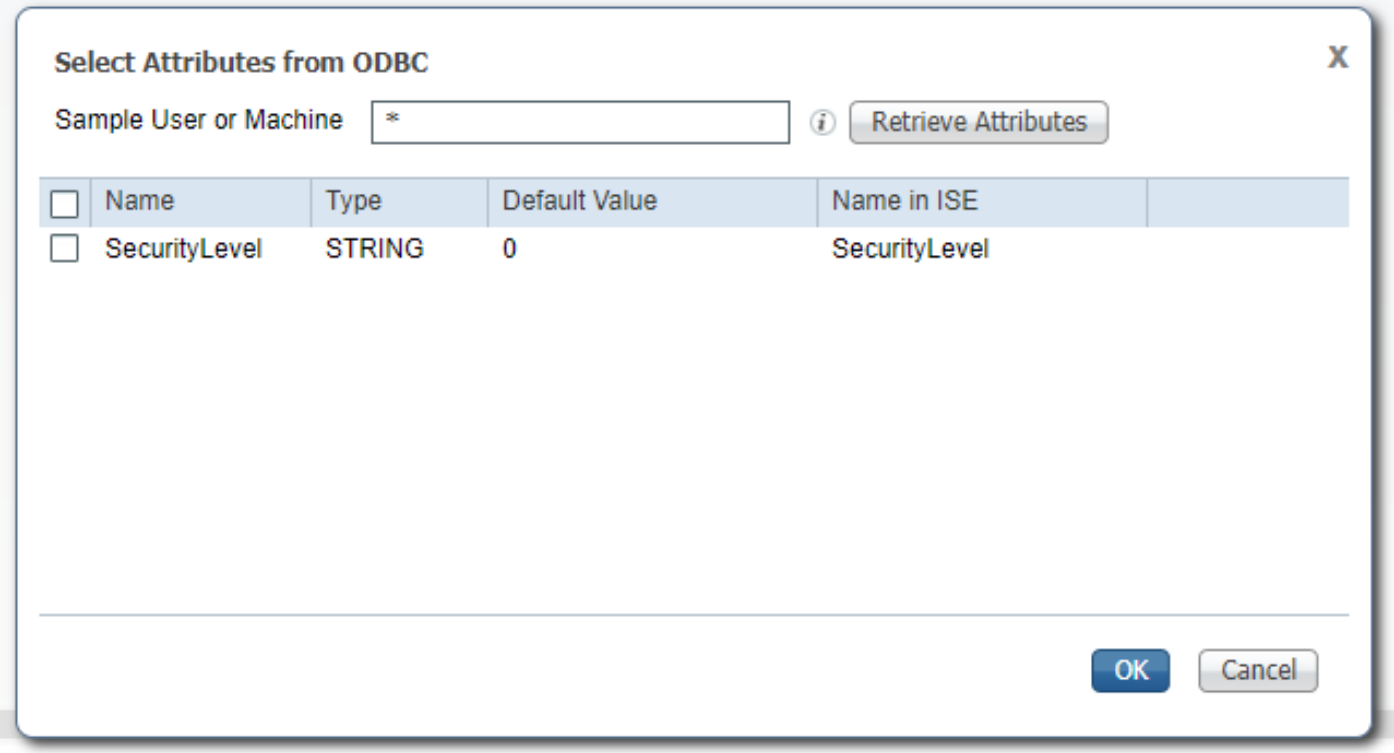
3. Crie um procedimento. Mesmos que com recuperação dos grupos, retornará todos os atributos distintos se o username é “*”

```
create or replace function ISEATTRSH
(
  ise_username IN VARCHAR2,
  ise_result OUT int
) return sys_refcursor as
BEGIN
  declare
    c integer;
    userid integer;
    resultSet SYS_REFCURSOR;
  begin
    IF ise_username = '*' then
      ise_result := 0;
      open resultSet for select DISTINCT ATTR_NAME, '0' as "VAL" from ATTRIBUTES;
    ELSE
      select count(*) into c from USERS where USERS.USERNAME = ise_username;
      select USER_ID into userid from USERS where USERS.USERNAME = ise_username;
      if c > 0 then
          ise_result := 0;
          open resultSet for select ATTR_NAME, VALUE from ATTRIBUTES where USER_ID = userid;
      ELSE
          ise_result := 3;
          open resultSet for select 0 from dual where 1=2;
      END IF;
    END IF;
    return resultSet;
  end;
END ;
```

4. Trace-o para buscar atributos



5. Busque os atributos



Selecione atributos e clique o [OK].

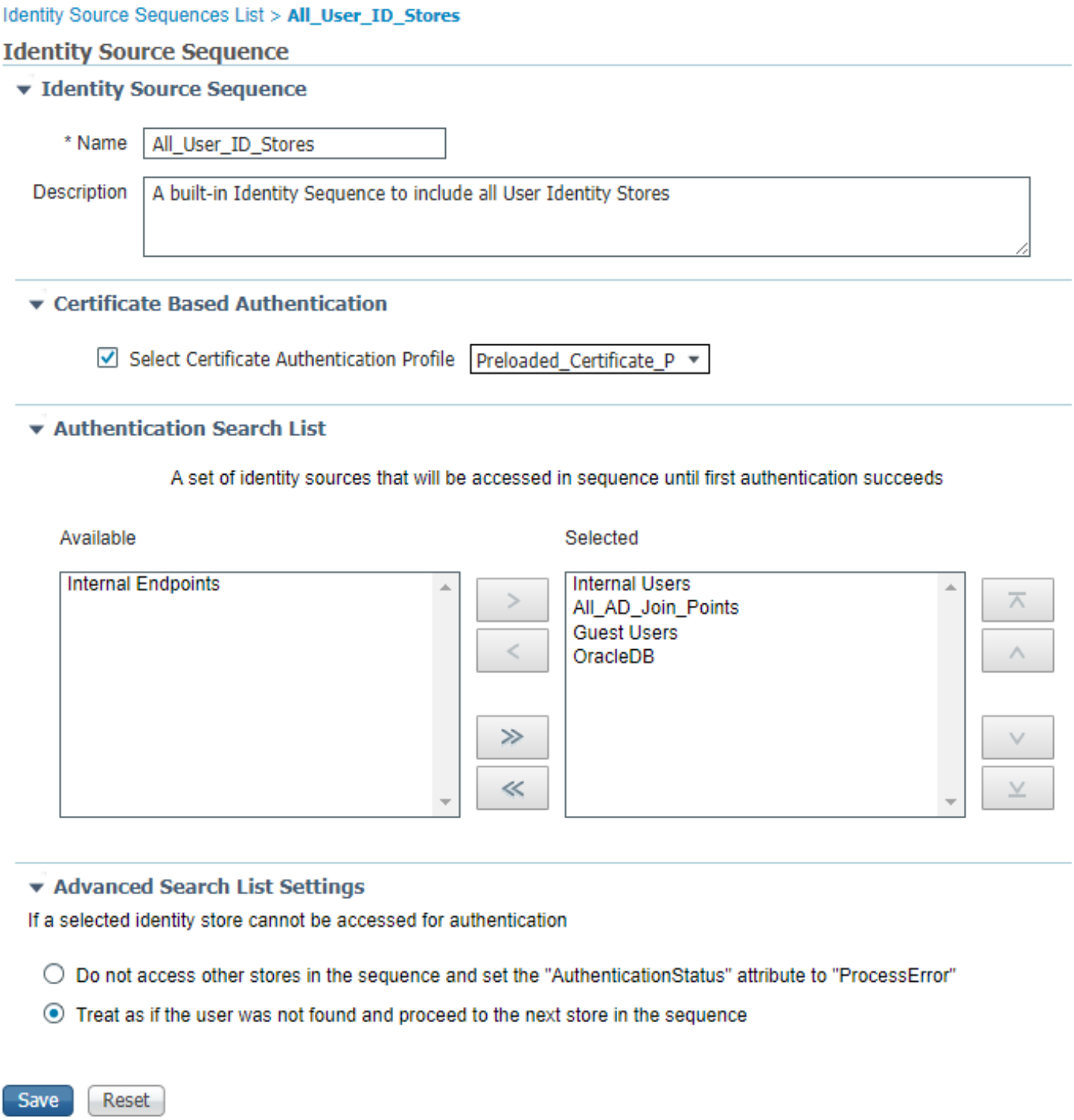
Etapa 6. Configurar políticas da autenticação/autorização

Neste exemplo as seguintes políticas simples da autorização foram configuradas:

Os usuários com **SecurityLevel = 5** serão negados.

Etapa 7. Adicionar o Oracle ODBC às sequências da fonte da identidade

Navegue às *sequências da fonte da administração > do Gerenciamento de identidades > da identidade,* selecione sua sequência e adicionar o ODBC à sequência:



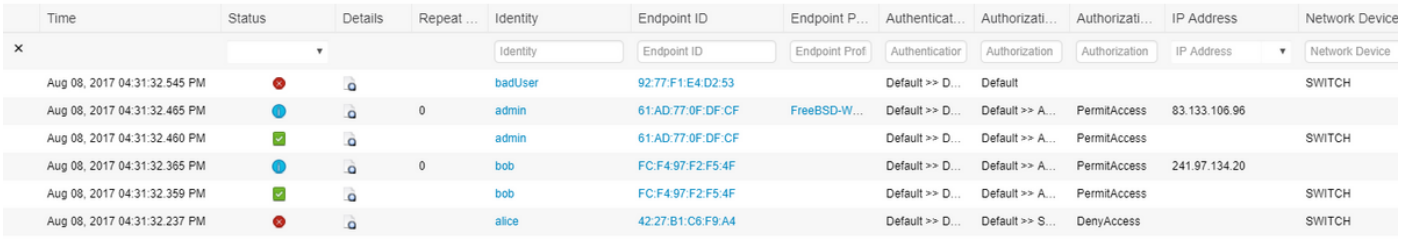
Salvar a.

Verificar

Você deve agora poder autenticar até agora usuários contra o ODBC e recuperar seus grupos e atributos.

Logs vivos do RAIO

Execute algumas autenticações e navegue às *operações > ao RAIO > logs vivos*



Time	Status	Details	Repeat ...	Identity	Endpoint ID	Endpoint P...	Authenticat...	Authorizati...	Authorizati...	IP Address	Network Device
				Identity	Endpoint ID	Endpoint Prof	Authenticatior	Authorization	Authorization	IP Address	Network Device
Aug 08, 2017 04:31:32.545 PM				badUser	92:77:F1:E4:D2:53		Default >> D...	Default			SWITCH
Aug 08, 2017 04:31:32.465 PM			0	admin	61:AD:77:0F:DF:CF	FreeBSD-W...	Default >> D...	Default >> A...	PermitAccess	83.133.106.96	
Aug 08, 2017 04:31:32.460 PM				admin	61:AD:77:0F:DF:CF		Default >> D...	Default >> A...	PermitAccess		SWITCH
Aug 08, 2017 04:31:32.365 PM			0	bob	FC:F4:97:F2:F5:4F		Default >> D...	Default >> A...	PermitAccess	241.97.134.20	
Aug 08, 2017 04:31:32.359 PM				bob	FC:F4:97:F2:F5:4F		Default >> D...	Default >> A...	PermitAccess		SWITCH
Aug 08, 2017 04:31:32.237 PM				alice	42:27:B1:C6:F9:A4		Default >> D...	Default >> S...	DenyAccess		SWITCH

Como você pode ver, o usuário Alice tem **SecurityLevel = 5,** daqui o acesso foi rejeitado.

Relatórios de detalhes

Clique sobre **relatórios de detalhes na** coluna dos **detalhes** para a sessão interessante para verificar o fluxo.

Relatório detalhado para o usuário Alice (rejeitado devido a baixo SecurityLevel):