

Configurazione di ODBC su ISE 2.3 con Oracle Database

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Configurazione](#)

[Passaggio 1. Configurazione di base di Oracle](#)

[Passaggio 2. Configurazione di base ISE](#)

[Passaggio 3. Configurazione dell'autenticazione utente](#)

[Passaggio 4. Configurare il recupero del gruppo](#)

[Passaggio 5. Configurazione del recupero degli attributi](#)

[Passaggio 6. Configurare i criteri di autenticazione/autorizzazione](#)

[Passaggio 7. Aggiunta di Oracle ODBC alle sequenze origine identità](#)

[Verifica](#)

[Registri attivi RADIUS](#)

[Report dettagliato](#)

[Risoluzione dei problemi](#)

[Credenziali utilizzate non corrette](#)

[Nome database errato \(nome servizio\)](#)

[Risoluzione dei problemi relativi alle autenticazioni degli utenti](#)

[Riferimenti](#)

Introduzione

In questo documento viene descritto come configurare Identity Services Engine (ISE) con Oracle Database per l'autenticazione ISE utilizzando Open Database Connectivity (ODBC).

L'autenticazione Open Database Connectivity (ODBC) richiede che ISE sia in grado di recuperare una password utente in formato testo normale. La password può essere crittografata nel database, ma deve essere decrittografata dalla stored procedure.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Cisco Identity Services Engine 2.3
- Concetti su database e ODBC
- Oracle

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Identity Services Engine 2.3.0.298
- Centos 7
- Oracle Database 12.2.0.1.0
- Oracle SQL Developer 4.1.5

Configurazione

Nota: Trattare le procedure SQL presentate in questo documento come esempi. Questo non è un metodo ufficiale e consigliato per la configurazione del database Oracle. Accertarsi di comprendere i risultati e l'impatto di ogni query SQL eseguita.

Passaggio 1. Configurazione di base di Oracle

In questo esempio Oracle è stato configurato con i seguenti parametri:

- Nome DB: **ORCL**
- Nome servizio: **orcl.vkumov.local**
- Port: **1521** (predefinito)
- Account creato per ISE con nome utente **ise**

Configurare il database Oracle prima di procedere.

Passaggio 2. Configurazione di base ISE

Creare un'origine identità ODBC in *Amministrazione > Origine identità esterna > ODBC* e verificare la connessione:

ODBC Identity Source

General **Connection** Stored Procedures Attributes Groups

ODBC DB connection details

* Hostname/IP[:port]

* Database name

Admin username ⓘ

Admin password

* Timeout

* Retries

* Database type

Test connection X

Connection succeeded

Stored Procedures

- Plain text password authentication - Not Configured
- Plain text password fetching - Not Configured
- Check username or machine exists - Not Configured
- Fetch groups - Not Configured
- Fetch attributes - Not Configured

Nota: ISE si connette a Oracle utilizzando il nome del servizio, quindi il campo [Nome database] deve essere compilato con il nome del servizio esistente in Oracle e non con il SID (o nome DB). A causa del bug [CSCvf06497](#), non è possibile usare i punti (.) nel campo [Nome database]. Questo bug è stato risolto in ISE 2.3.

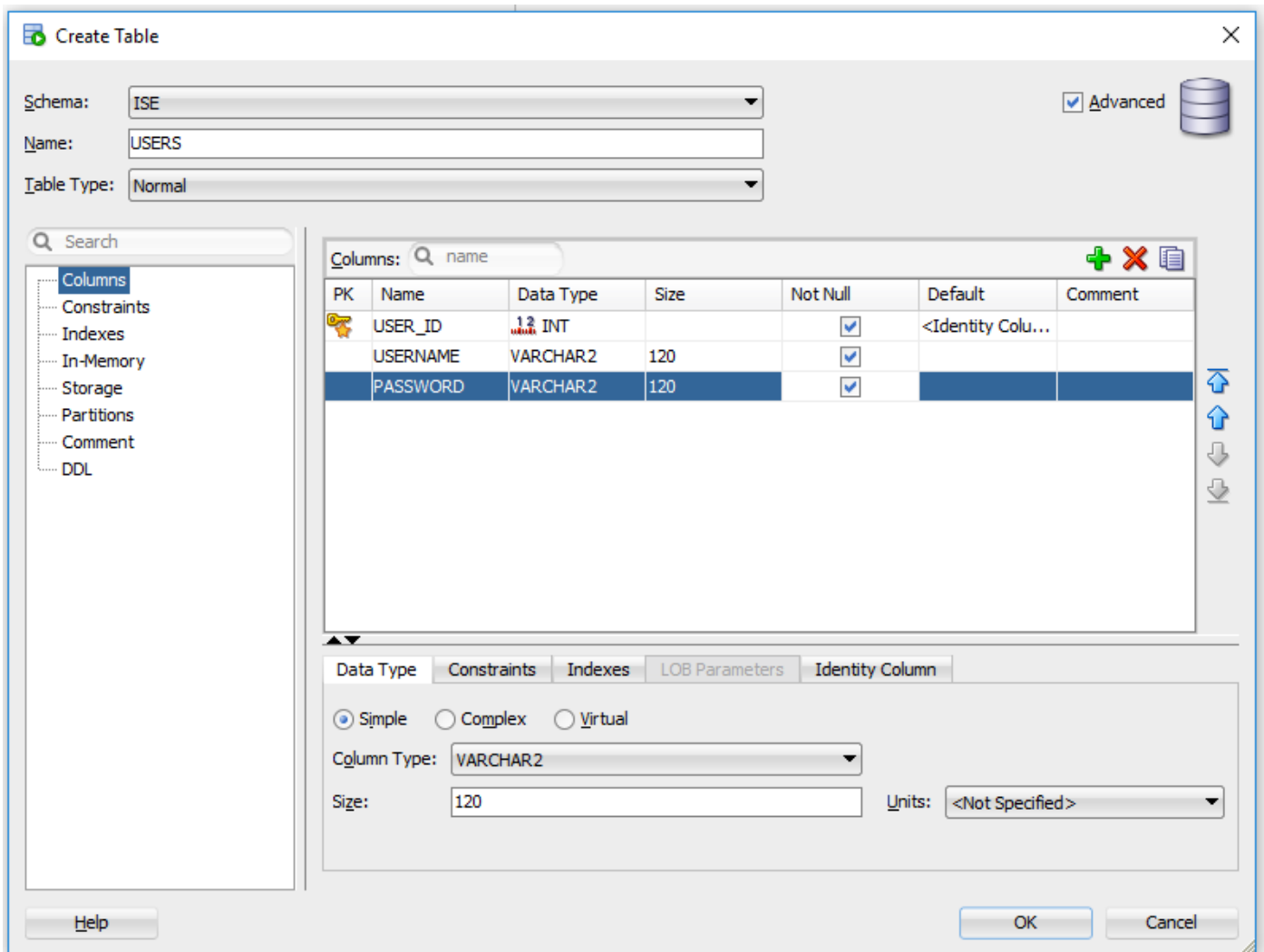
Passaggio 3. Configurazione dell'autenticazione utente

L'autenticazione ISE per ODBC utilizza stored procedure. È possibile selezionare il tipo di routine. In questo esempio vengono utilizzati i recordset come restituzione.

Per altre procedure, fare riferimento al [manuale Cisco Identity Services Engine Administrator Guide, release 2.3](#)

Suggerimento: È possibile restituire parametri denominati anziché resultSet. È un tipo di output diverso, la funzionalità è la stessa.

1. Creare la tabella con le credenziali degli utenti. Verificare di aver impostato le impostazioni di identità nella **chiave primaria**.



2. Aggiungere utenti

```
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. Creare una procedura di autenticazione con password in testo normale (utilizzata per PAP, metodo interno EAP-GTC, 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;
  END IF;
```

```

return resultSet;
end;
END ISEAUTH_R;

```

4. Creare una procedura per il recupero di password in testo normale (utilizzata per i metodi interni CHAP, MSCHAPv1/v2, EAP-MD5, LEAP, EAP-MSCHAPv2, 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. Creare una procedura per verificare l'esistenza del nome utente o del computer (utilizzata per MAB, riconnessione rapida di PEAP, EAP-FAST ed 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;

```

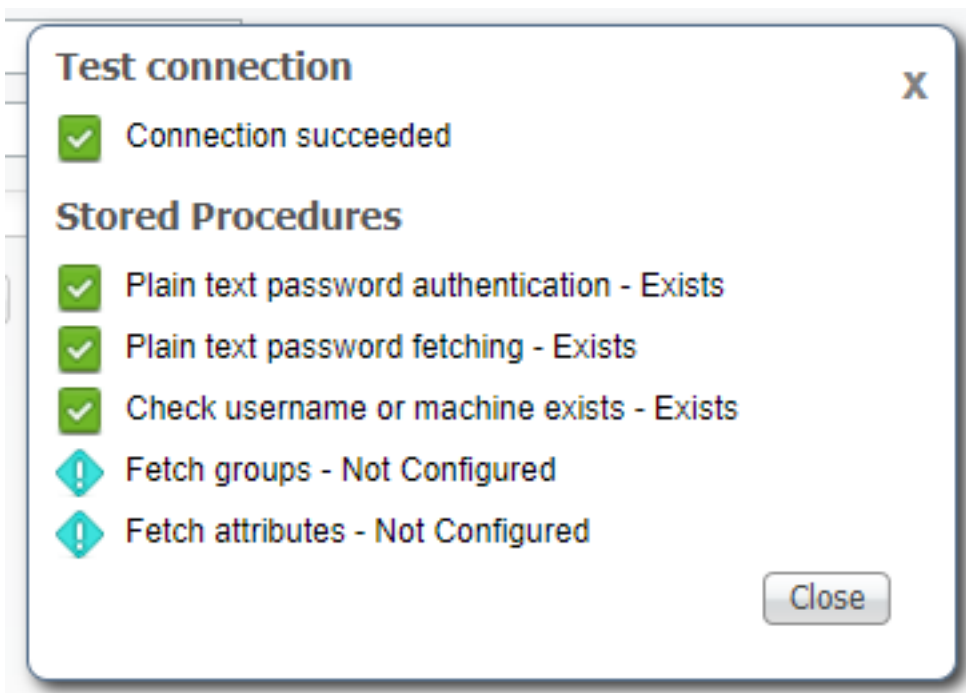
6. Configurare le procedure su ISE e salvare

ODBC Identity Source

General Connection **Stored Procedures** Attributes Groups

Stored procedure type	>Returns recordset		
Plain text password authentication	ISEAUTH_R	i	+
Plain text password fetching	ISEFETCH_R	i	+
Check username or machine exists	ISELOOKUP_R	i	+
Fetch groups		i	+
Fetch attributes		i	+
Search for MAC Address in format	XX-XX-XX-XX-XX-XX	i	

7. Tornare alla scheda Connessione e fare clic sul pulsante Test connessione



Passaggio 4. Configurare il recupero del gruppo

1. Creare tabelle contenenti gruppi di utenti e un'altra tabella utilizzata per il mapping multi-a-molti

```
-----  
-- DDL for Table GROUPS  
-----  
  
CREATE TABLE "ISE"."GROUPS"  
("GROUP_ID" NUMBER(*,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE  
9999999999999999999999999999999999999999999 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;
```

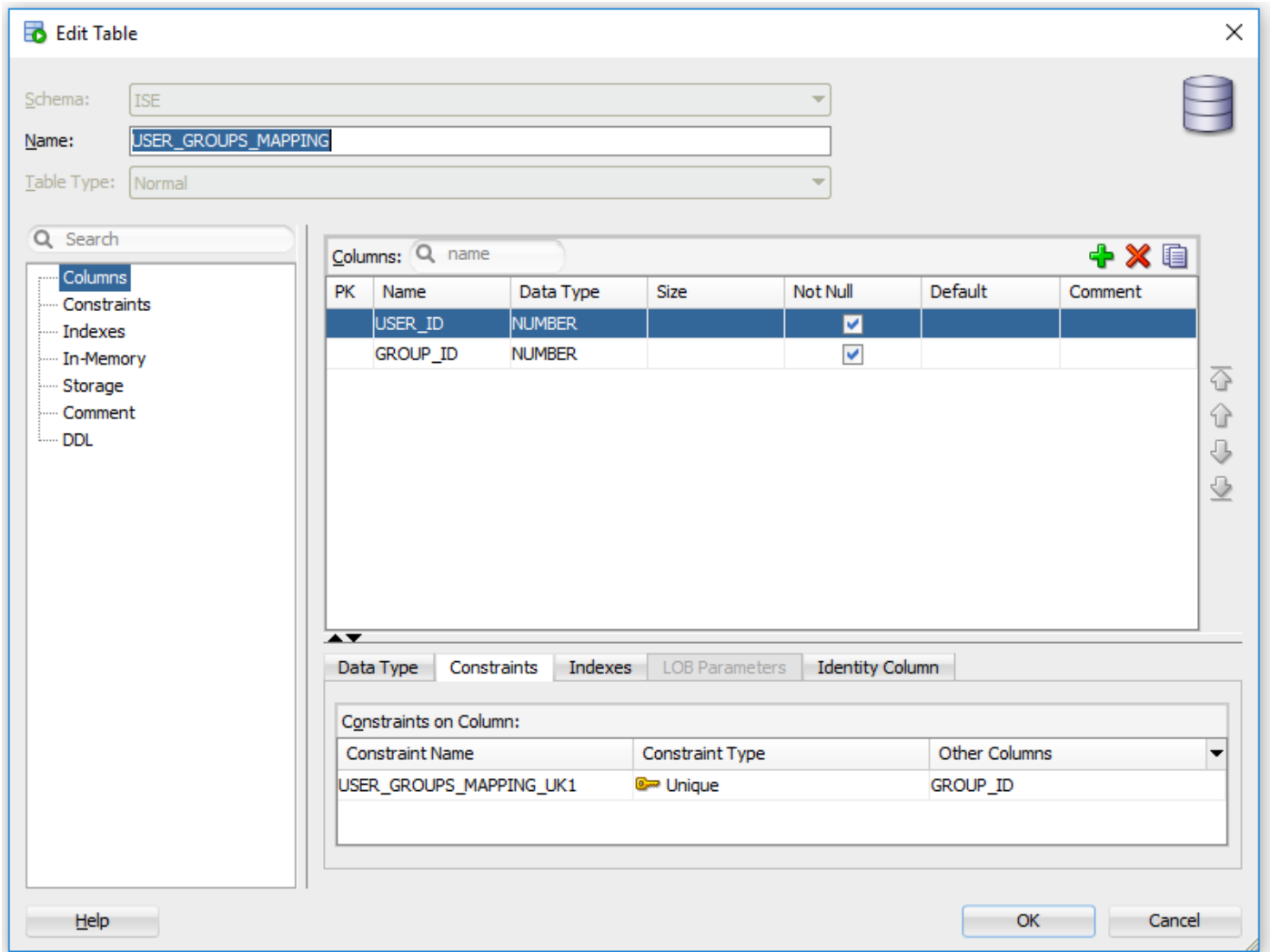
Dalla GUI:

The screenshot shows the 'Edit Table' window for the table 'GROUPS' in the 'ISE' schema. The table type is 'Normal'. The columns are:

PK	Name	Data Type	Size	Not Null	Default	Comment
<input checked="" type="checkbox"/>	GROUP_ID	NUMBER		<input checked="" type="checkbox"/>	<Identity Colu...	
<input type="checkbox"/>	GROUP_NAME	VARCHAR2	255	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	DESCRIPTION	CLOB		<input type="checkbox"/>		

The 'Constraints' tab is selected, showing a table of constraints on the column:

Constraint Name	Constraint Type	Other Columns
GROUPS_PK	Primary Key	



2. Aggiungere gruppi e mapping, in modo che **alice** e **bob** appartengano al gruppo **Users** e **admin** appartengano al gruppo **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. Creare una procedura di recupero di gruppo. Restituisce tutti i gruppi se username è ""

```
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 ;

```

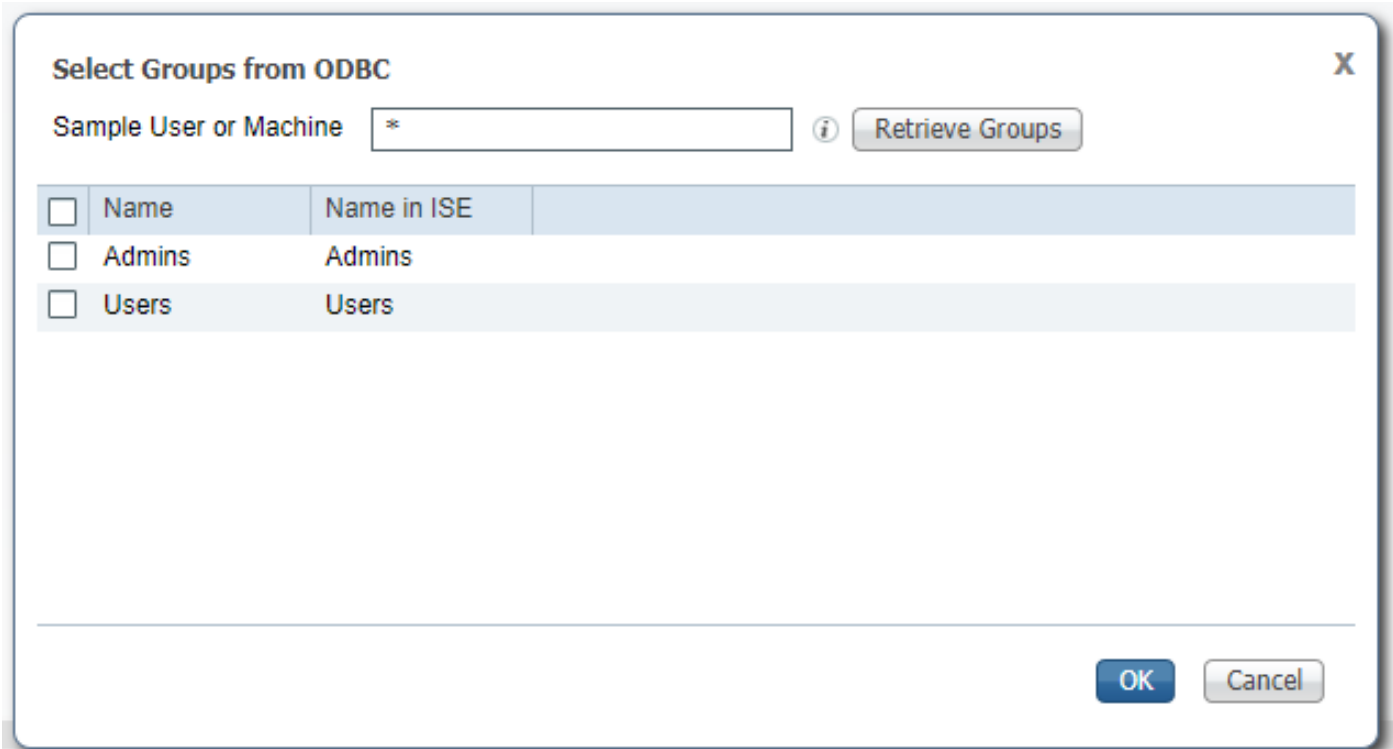
4. Mappare ai gruppi Fetch

[ODBC List > OracleDB](#)

ODBC Identity Source

General	Connection	Stored Procedures	Attributes	Groups
Stored procedure type		Returns recordset		
Plain text password authentication		ISEAUTH_R	i	+
Plain text password fetching		ISEFETCH_R	i	+
Check username or machine exists		ISELOOKUP_R	i	+
Fetch groups		ISEGROUPSH	i	+
Fetch attributes			i	+
Search for MAC Address in format		XX-XX-XX-XX-XX-XX	i	

5. Recuperare i gruppi e aggiungerli all'origine identità ODBC



Selezionare i gruppi necessari e fare clic su OK. Verranno visualizzati nella scheda **Gruppi**

[ODBC List](#) > **OracleDB**

ODBC Identity Source

General Connection Stored Procedures Attributes **Groups**

Edit + Add X Delete

<input type="checkbox"/>	Name	Name in ISE
<input type="checkbox"/>	Admins	Admins
<input type="checkbox"/>	Users	Users

Passaggio 5. Configurazione del recupero degli attributi

1. Per semplificare questo esempio, per gli attributi viene utilizzata una tabella semplice

```
-----
-- 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;
```

Dalla GUI:

Schema: ISE
Name: ATTRIBUTES
Table Type: Normal

Columns:

PK	Name	Data Type	Size	Not Null	Default	Comment
	USER_ID	NUMBER		<input checked="" type="checkbox"/>		
	ATTR_NAME	VARCHAR2	255	<input checked="" type="checkbox"/>		
	VALUE	VARCHAR2	255	<input type="checkbox"/>		

Constraints on Column:

Constraint Name	Constraint Type	Other Columns
ATTRIBUTES_FK1	Foreign Key	
ATTRIBUTES_PK	Primary Key	ATTR_NAME

2. Creare alcuni attributi per gli utenti

```
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. Creare una procedura. Come per il recupero dei gruppi, restituisce tutti gli attributi distinti se nomeutente è "*"

```
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 ;
```

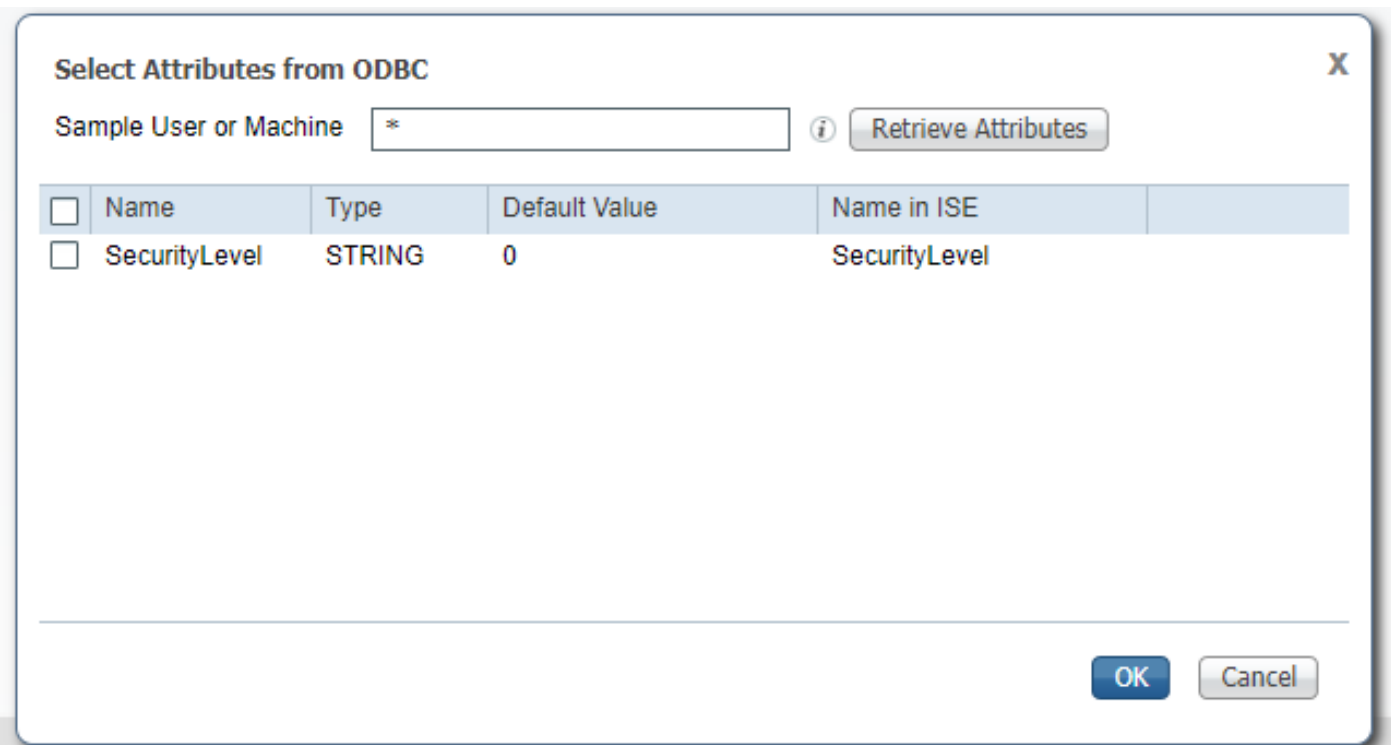
4. Mappare agli attributi Fetch

[ODBC List > OracleDB](#)

ODBC Identity Source

General	Connection	Stored Procedures	Attributes	Groups
Stored procedure type		Returns recordset		
Plain text password authentication		ISEAUTH_R		
Plain text password fetching		ISEFETCH_R		
Check username or machine exists		ISELOOKUP_R		
Fetch groups		ISEGROUPSH		
Fetch attributes		ISEATTRSH		
Search for MAC Address in format		XX-XX-XX-XX-XX-XX		

5. Recuperare gli attributi



Selezionare gli attributi e fare clic su OK.

Passaggio 6. Configurare i criteri di autenticazione/autorizzazione

In questo esempio sono stati configurati i seguenti criteri di autorizzazione semplici:

<input checked="" type="checkbox"/>	Allow admin network access	OracleDB ExternalGroups EQUALS Admins	PermitAccess	Select from list	1	
<input checked="" type="checkbox"/>	SecurityLevel too low	OracleDB SecurityLevel EQUALS 5	DenyAccess	Select from list	0	
<input checked="" type="checkbox"/>	Allow users network access	OracleDB ExternalGroups EQUALS Users	PermitAccess	Select from list	2	

Gli utenti con **SecurityLevel = 5** verranno rifiutati.

Passaggio 7. Aggiunta di Oracle ODBC alle sequenze origine identità

Passare a *Amministrazione > Gestione identità > Sequenze origine identità*, selezionare la sequenza e aggiungere ODBC alla sequenza:

Identity Source Sequence

▼ Identity Source Sequence

* Name

Description

▼ Certificate Based Authentication

Select Certificate Authentication Profile

▼ Authentication Search List

A set of identity sources that will be accessed in sequence until first authentication succeeds

Available

Selected

▼ Advanced Search List Settings

If a selected identity store cannot be accessed for authentication

- Do not access other stores in the sequence and set the "AuthenticationStatus" attribute to "ProcessError"
- Treat as if the user was not found and proceed to the next store in the sequence

Salvatela.

Verifica

A questo punto dovrebbe essere possibile autenticare gli utenti in base a ODBC e recuperare i relativi gruppi e attributi.

Registri attivi RADIUS

Eseguire alcune autenticazioni e passare a *Operazioni > RADIUS > Live Log*

Time	Status	Details	Repeat ...	Identity	Endpoint ID	Endpoint P...	Authenticat...	Authorizati...	Authorizati...	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 04:31:32.485 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

Come si può vedere, l'utente Alice ha **SecurityLevel = 5**, quindi l'accesso è stato rifiutato.

Report dettagliato

Fare clic su **Rapporto dettagliato** nella colonna **Dettagli** per la sessione interessante per controllare il flusso.

Rapporto dettagliato per l'utente Alice (rifiutato a causa del livello di protezione basso):

