

# **RAS before RAG:** **Real Application Security** **Fundamentals** **for Gen AI Apps**



**December 3, 2025**

**Karen Cannell**  
CTHO  
TH Technology

**Jim Czuprynski**  
Chief StoryTeller  
Zero Defect Computing, Inc.

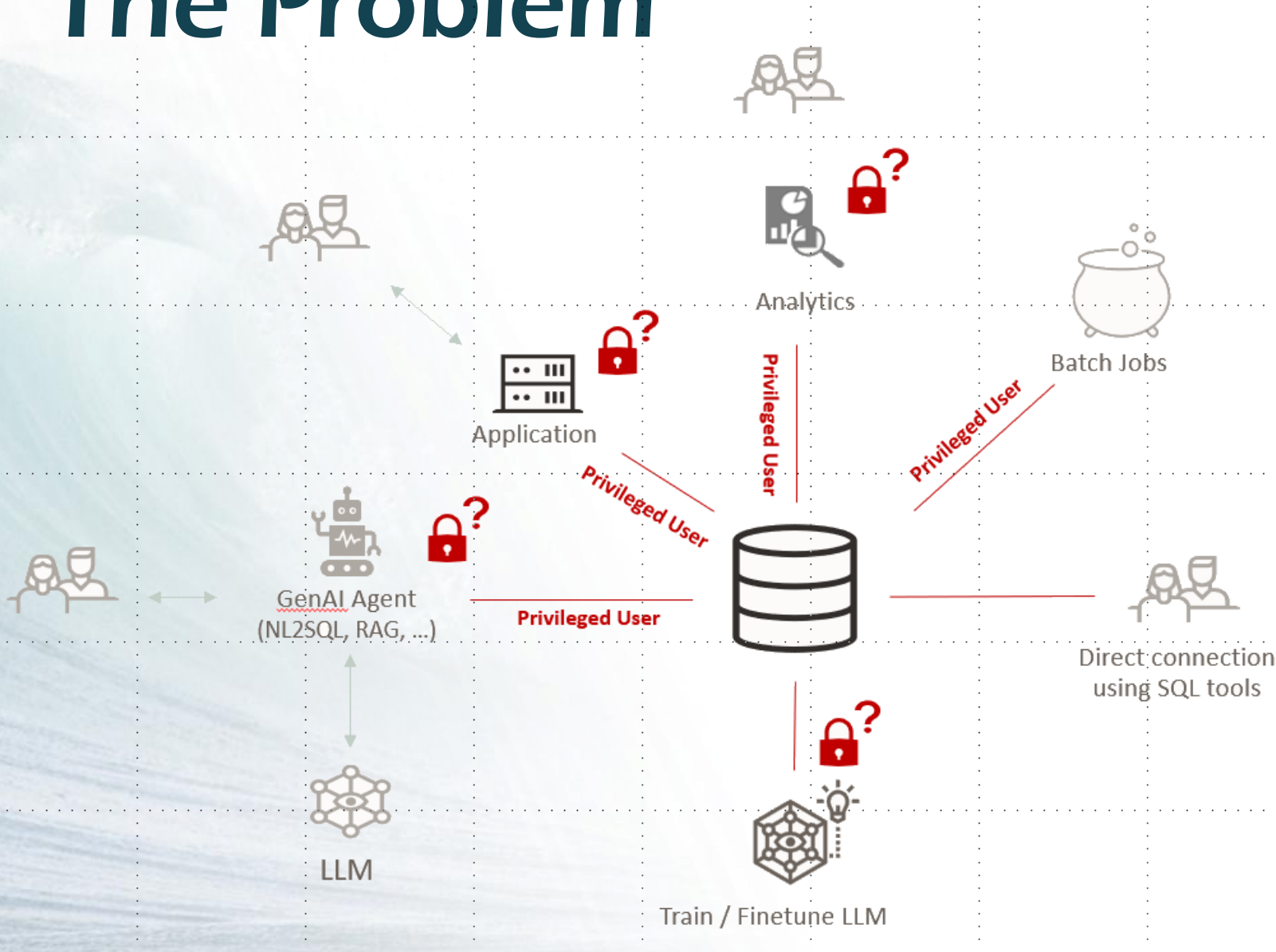
# Data Security is not ...

```
SQL>GRANT SELECT ON HR.EMPLOYEES TO BOB
```

## **Fine-grained Data Security is more than ...**

- Creating different roles in the Database
- Creating different technical users in the Database
- Splitting table between different schemas

# The Problem

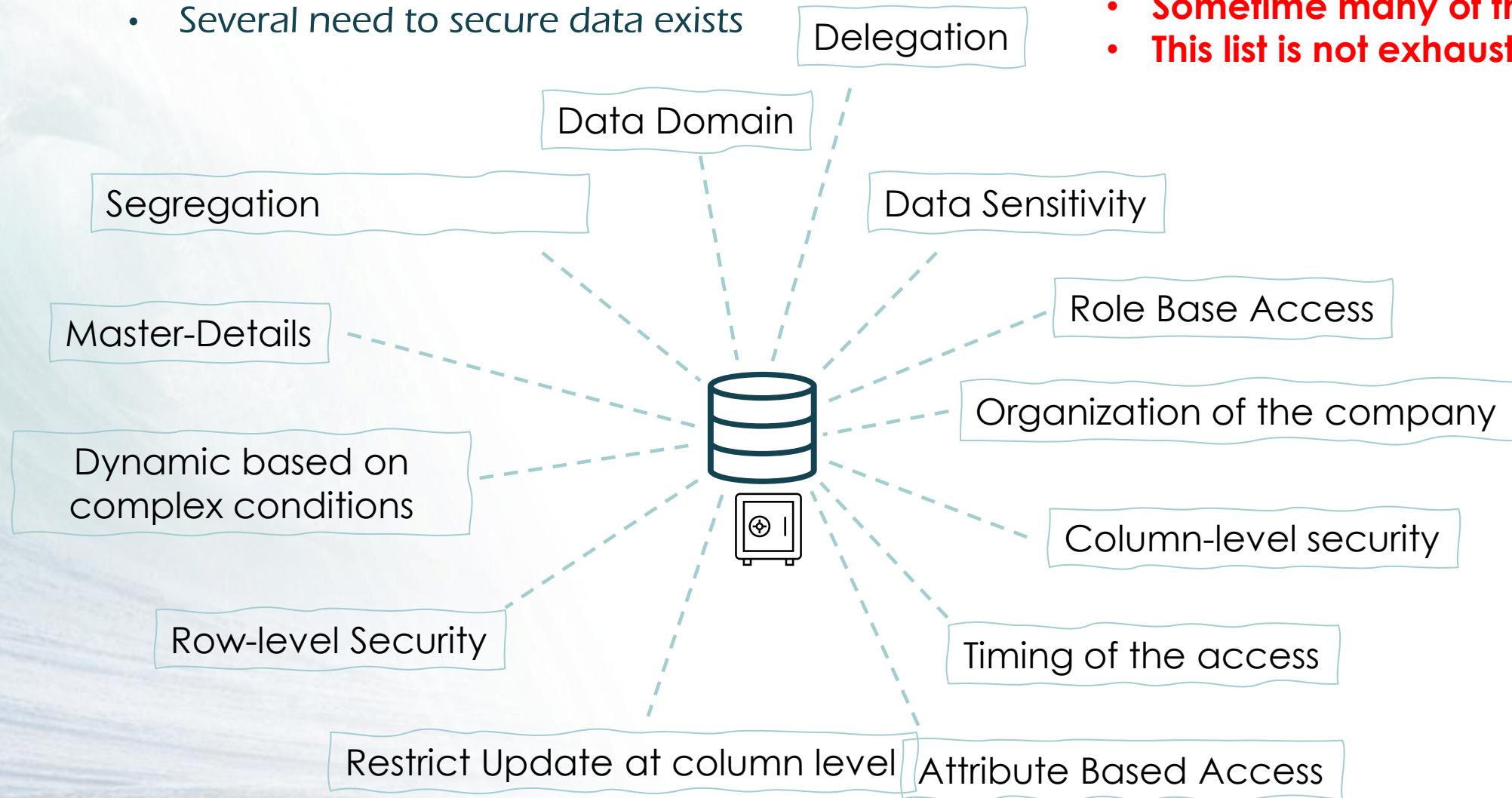


- Security checks embedded in application logic
- Identity of end user not propagated to the Database
- High risks with big-user connection
- Fragmented security
- Data not protected from direct connection
- No application user audit
- Complex development and maintenance

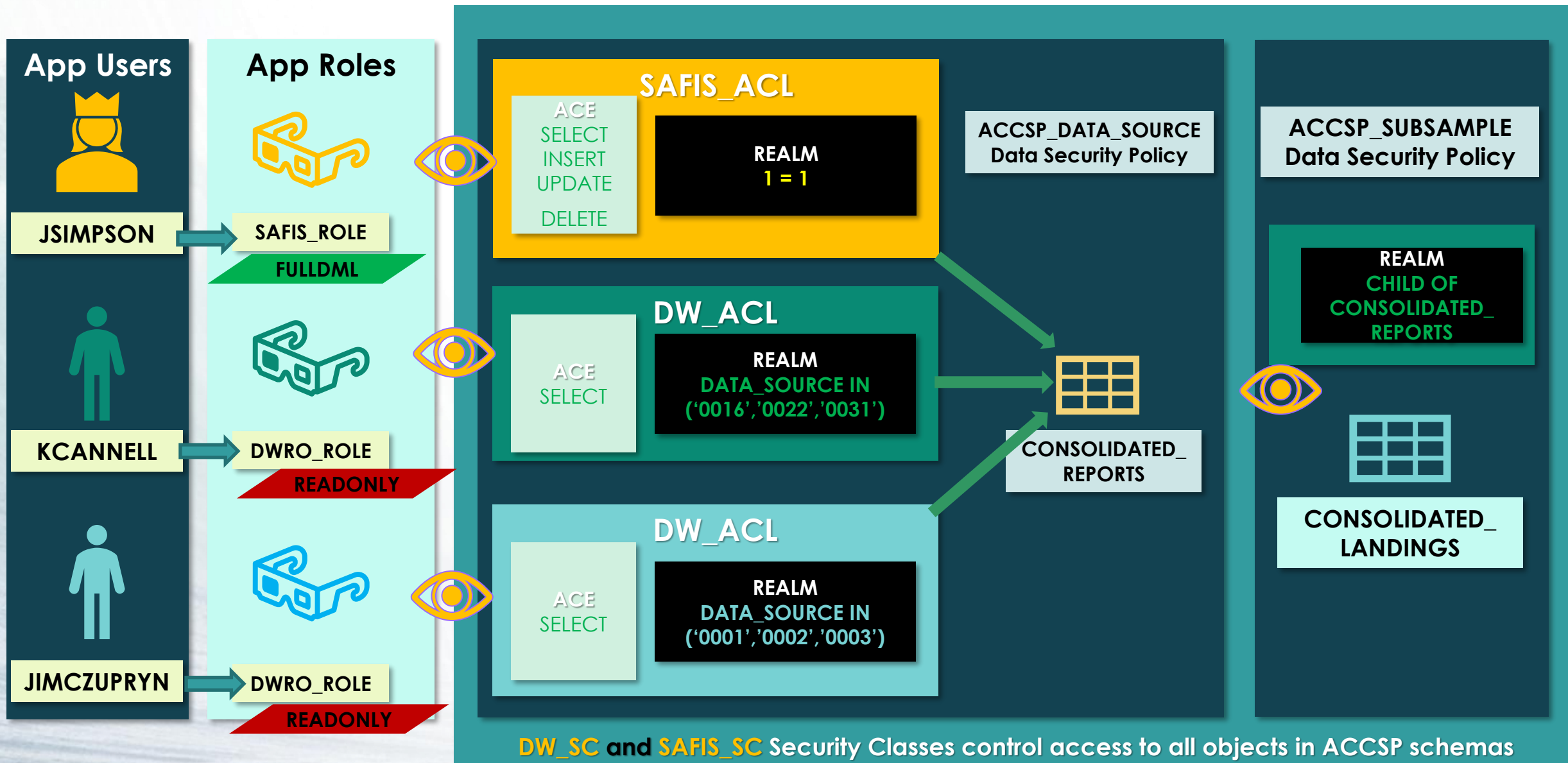
# Data Security Patterns

- Several need to secure data exists

- Sometime many of those overlap
- This list is not exhaustive

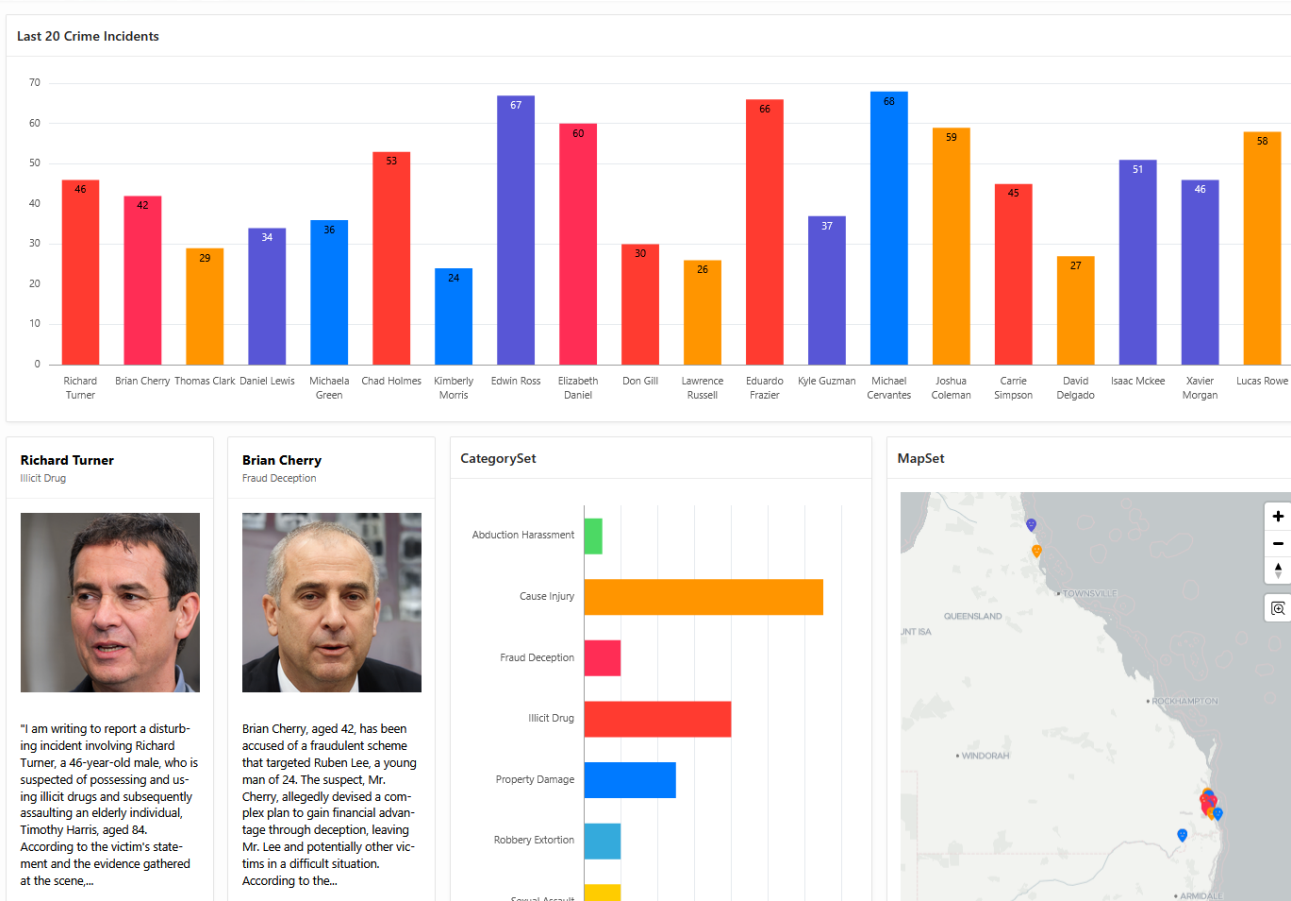


# RAS Components: ACLs, ACEs, Realms, Policies, & Classes





# Crime Aggregate Dashboard



## Coordinated Crime Response & Intelligence Operations

During a surge in coordinated criminal activity across urban and regional jurisdictions, a unified crime intelligence platform proved critical in **supporting law enforcement and public safety operations**.

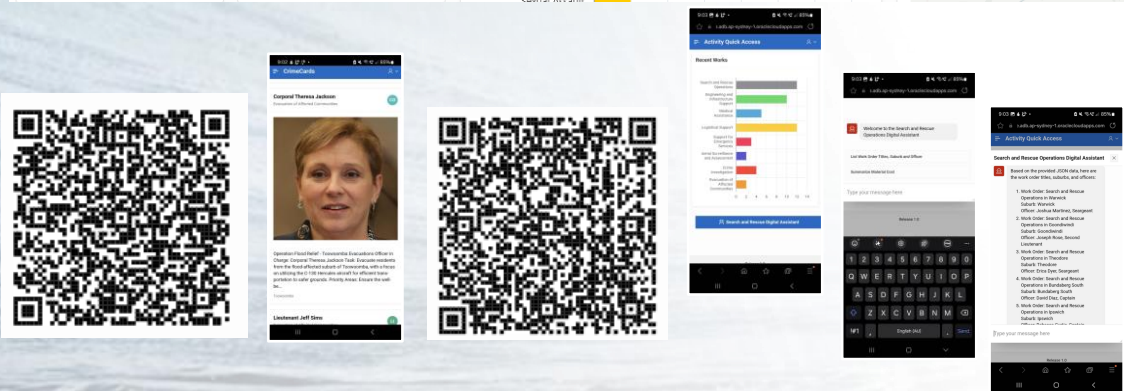
Tracking incidents across categories, given they maybe connected

- Abduction Harrassment
- Cause Injury
- Fraud Deception
- Illicit Drug
- Property Damage
- Robbery Extortion
- Sexual Assault
- Theft
- Weapons Explosives

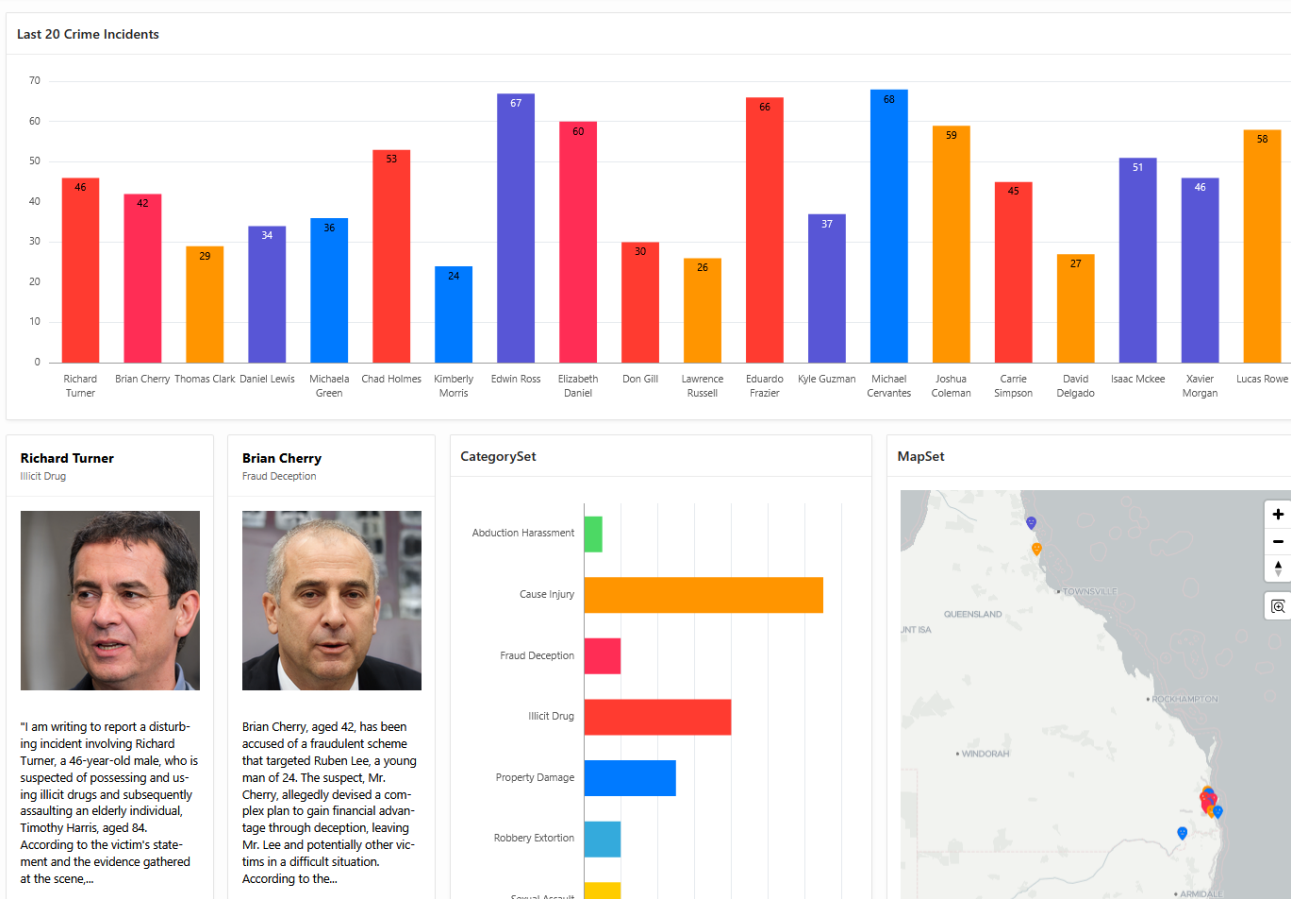
The centralized **Crime Aggregate Dashboard** played a vital role in providing real-time intelligence, improving incident visibility, and enabling agencies to respond swiftly and collaboratively.

## Outcome

The Crime Aggregate Dashboard proved indispensable in unifying data, coordinating real-time actions, and enhancing public trust through transparency and responsiveness. It served not only as a tactical tool for frontline responders but also as a strategic platform for long-term crime prevention and policy-making.



# Crime Aggregate Dashboard



## Coordinated Crime

During a surge in crime across regional jurisdictions, this tool was critical in supporting

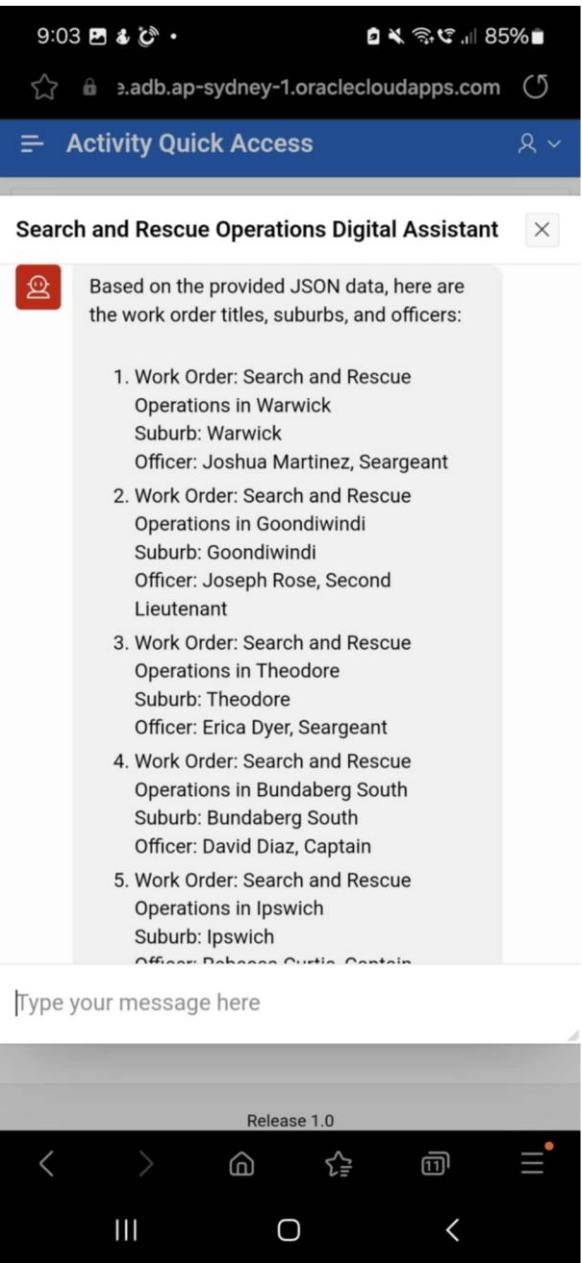
## Tracking incidents

- Abduction Harassment
- Cause Injury
- Fraud Deception
- Illicit Drug
- Property Damage
- Robbery Extortion
- Sexual Assault
- Theft
- Weapons Explosives

The centralized Crime Aggregate Dashboard is providing real-time data and insights, enabling agencies to

## Outcome

The Crime Aggregate Dashboard is coordinating real-time data, providing transparency and visibility for frontline response and crime prevention activities.



an and  
n proved  
operations.

nected

il role in  
lity, and

unifying data,  
t through  
tactical tool  
or long-term

# Access Demo Journey

## CrimesV40 – Access by State and Crime Type

### Part 1A

#### Regular Police

- RyanEdwards

- JakeSimmons
- MarkDonovan
- NathanCooper
- OliviaFraser
- RachelSutton
- ZoeMitchell
- EmilyDawson
- JessicaKlose

Aross All

Queensland

South Australia

Tasmania

Western Australia

New South Wales

Victoria

ACT

Northern Territory

### Part 1B

#### Organised crime org\_fraud\_drugs

- EthanHall

Fraud Deception

Illicit Drug

#### CauseInjury

- DavidLee

Weapons Explosives

## Show Add and Drop Access

### Part 2

#### Show Add & Drop

#### Break\_Ins

Property\_damage & Theft

Property damage

- SarahKim

Theft

#### CauseInjury

- KevinWhite

Cause Injury

#### Victim\_support

(incident, Victim details Only)

- EmilyChen

Victim\_support

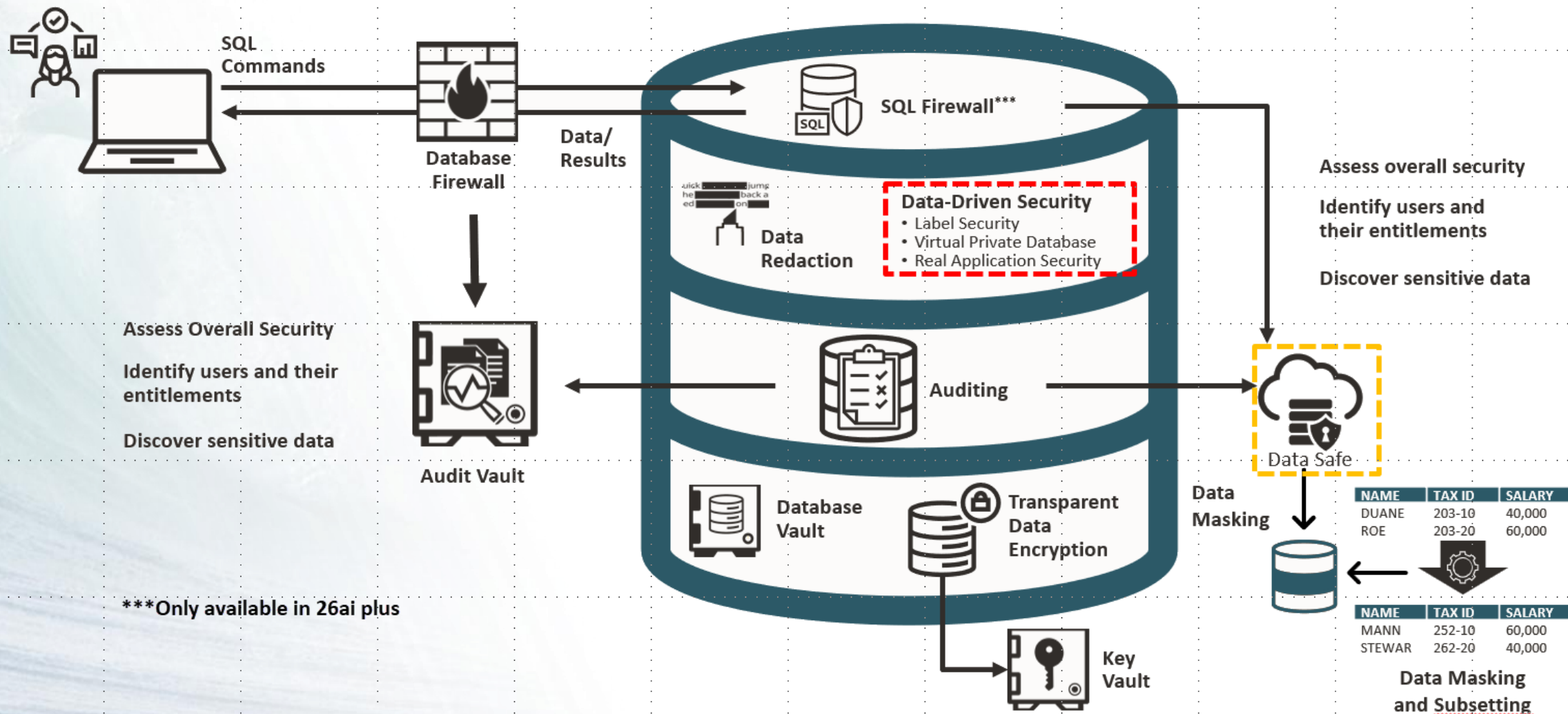




# **RAS**

# ***Capabilities***

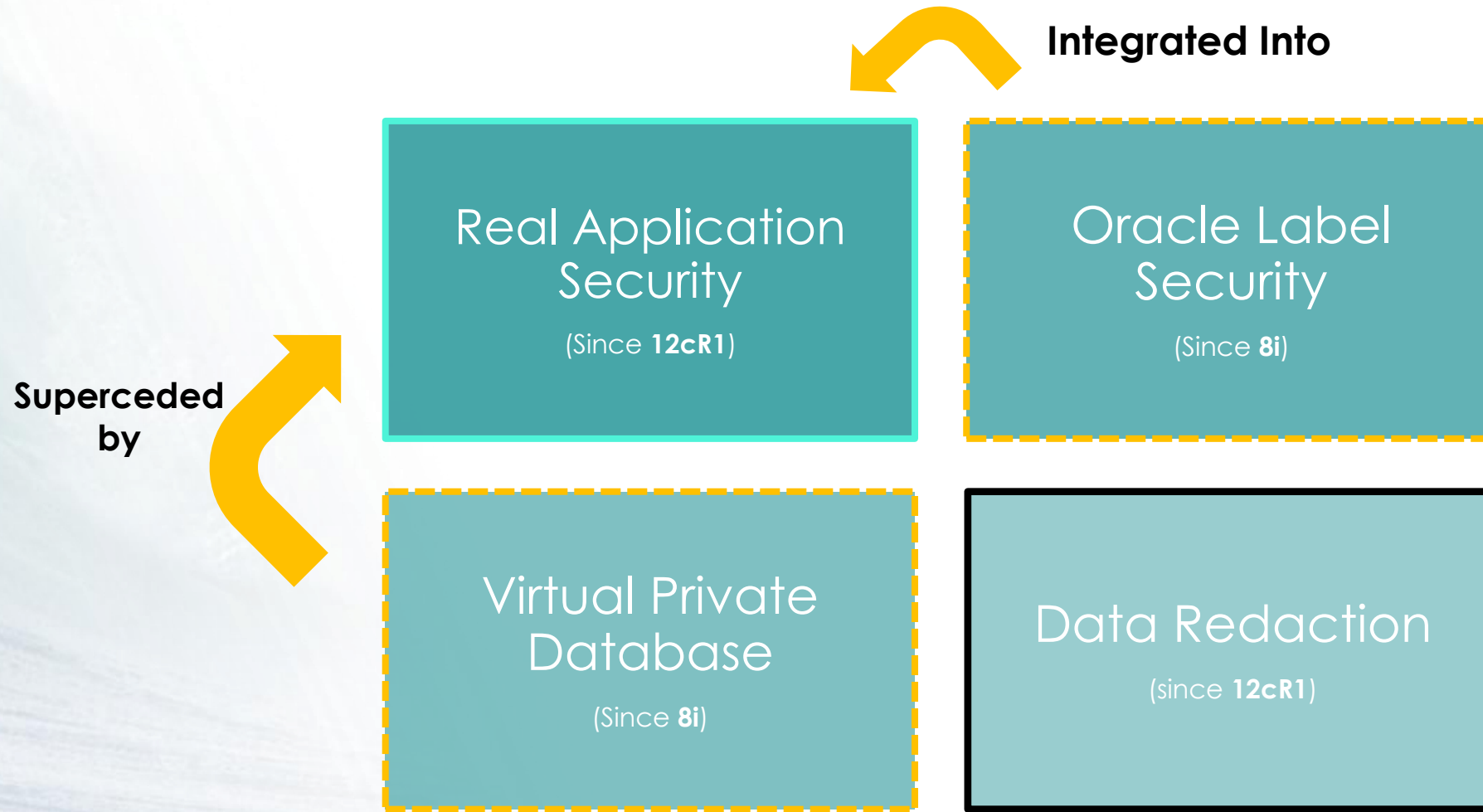
# Maximum Security Architecture



# Key elements before considering **Fine-Grained** Data Security

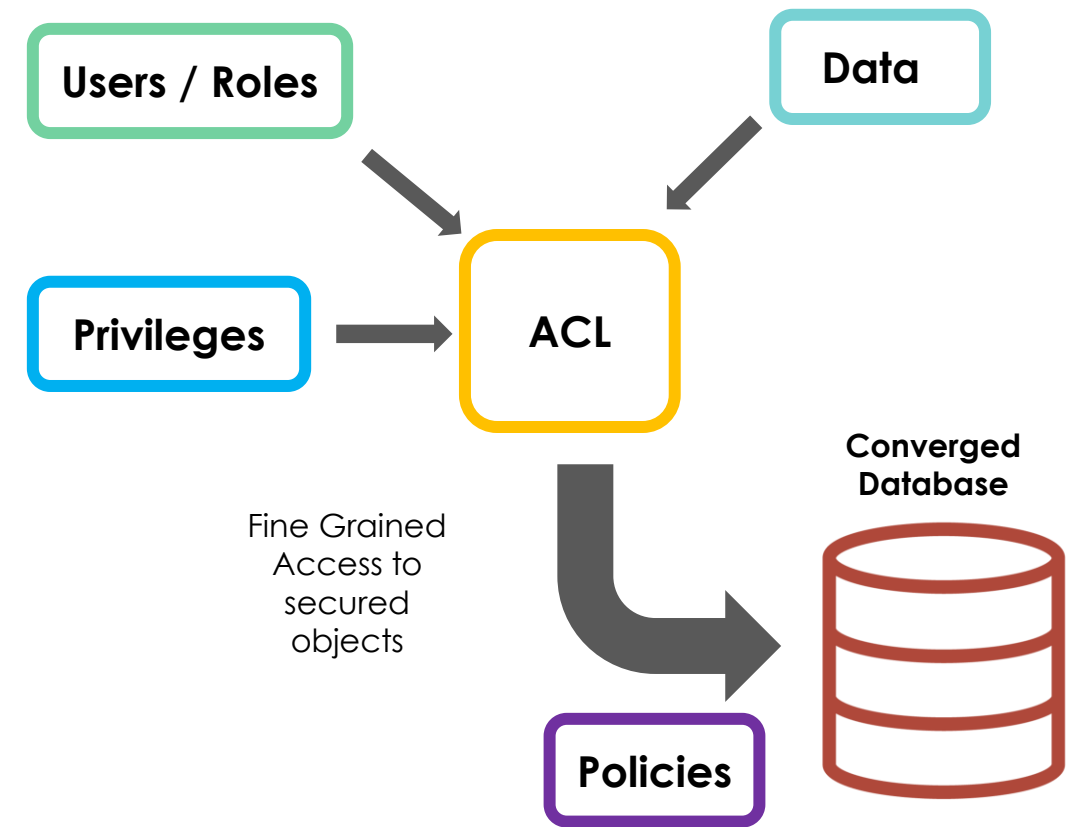
- How well do you know your data model?
- Have you identified all **different consumers** of your application's data?
- Do you know **how** you want to secure the data?
  - Do you need to limit access to particular **rows**?
  - Or do you need to also! hide data in specific **columns**?
- Fine-grained data security is almost always more than an IT activity!
  - It involves **Data Governance / Compliance / Development** teams

# So ... What Are My Options?



# Focus on Real Application Security

- Application / Business - Role based Security
- Support Declarative security Policies
- Enables end-to-end security for applications
- Row-level security
- Column-level security
- Can support complex security requirements
- True concept of application session <> database session
- Native integration with APEX
- Supports Master-Details security policies
- Access rules can be context-aware





# Practical Examples for RAS Usage

(A non-exhaustive list!)

- Securing **Analytics & BI**
- **Restricting access** to downstream applications
- Providing **enhanced security** for business end users who need **direct SQL access** to the database
- Allowing access to production for a **less-privileged** user
  - No need to create an **anonymized** copy or PDB clone!
- Securing **new kinds of workloads** in the converged database
  - **Vector RAG**
  - **JSON Duality**
  - **Select AI**

# Real Application Security (RAS)

How does it work concretely ?

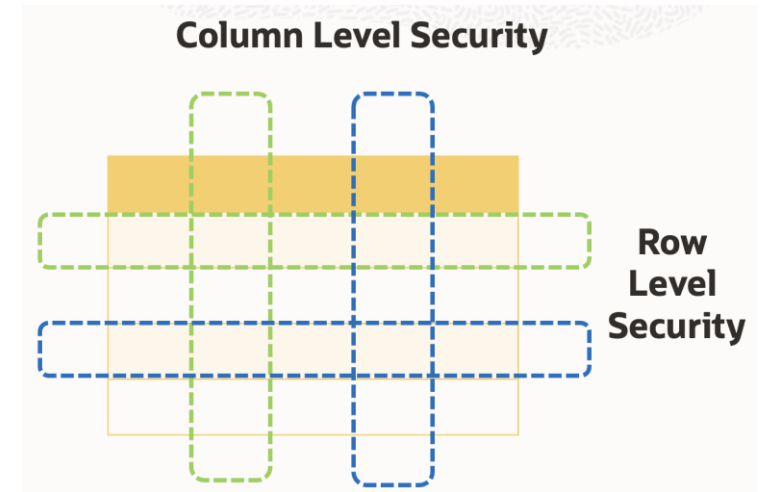
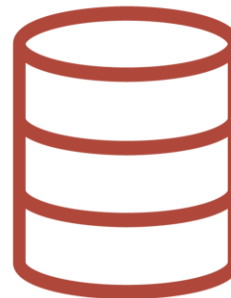
Two ways to be in the context of a RAS session :

- Connect as a DB User, then switch to a RAS session
- Direct Login Application Users (e.g. APEX)



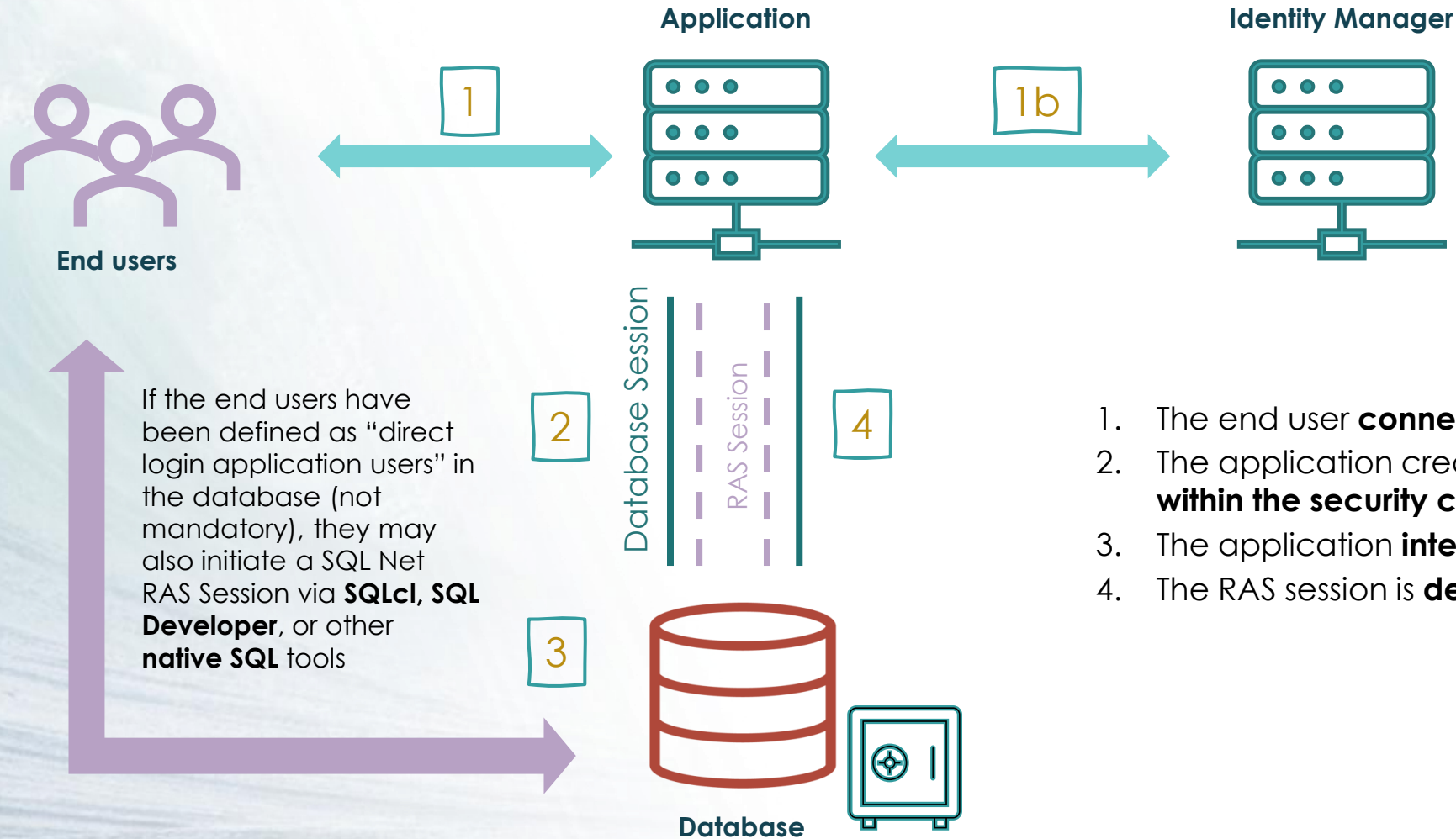
Database Session (v\$session)

RAS Session (DBA\_XS\_SESSIONS)



# Real Application Security

## Architecture of solution



If the end users have been defined as "direct login application users" in the database (not mandatory), they may also initiate a SQL Net RAS Session via **SQLcl**, **SQL Developer**, or other **native SQL** tools

1. The end user **connects** to the application
2. The application creates and attaches a RAS session **within the security context** of the end user
3. The application **interacts** with the database
4. The RAS session is **detached** from the database session

# Real Application Security

What is required by the application

- The application must be able to handle a RAS session
  - If not, you can still use a **direct-login** application user
- The APIs can be controlled via either PL/SQL APIs or Java APIs
- APEX by Example is integrated with RAS and can handle attaching and detaching a RAS session

# Real Application Security

- Create and Switch to a RAS session from a Database User - Internal

- The database user must have the **ADMINISTER\_SESSION** RAS privilege
- The RAS application user “**JIM**” is defined locally and his roles are also defined locally

```
SQL> CONNECT TECHNICAL_USER/<a_strong_pwd>@MY_23_AI_DB

DECLARE
    sessionid RAW(16)

BEGIN
    SYS.DBMS_XS_SESSIONS.CREATE_SESSION('JIM',sessionid);
    SYS.DBMS_XS_SESSIONS.ATTACH_SESSION(sessionid);
    . . .
    -- RAS security is now applied, so perform all
    -- required tasks within the established RAS session
    . . .
    SYS.DBMS_XS_SESSIONS.DETACH_SESSION;
    SYS.DBMS_XS_SESSIONS.DESTROY_SESSION(sessionid);
END;
/
```



# Real Application Security

## Create and Switch to a RAS session from a Database User - External

- The Database user must have the **ADMINISTER\_SESSION** RAS privilege
- The RAS application user “**BOB**” is **not** defined in the database
- The roles within the session are **dynamically passed** when the session is attached
- This allows the application to activate the **dynamic** roles conditionally
  - For example, based on the user's group membership in **Active Directory**

```
SQL> CONNECT TECHNICAL_USER/<a_strong_pwd>@MY_23_AI_DB

DECLARE
  sessionid RAW(16);
BEGIN
  SYS.DBMS_XS_SESSIONS.CREATE_SESSION('BOB',sessionid, true);
  SYS.DBMS_XS_SESSIONS.ATTACH_SESSION(
    sessionid,
    XS$NAME_LIST('HR_STAFF'));
  SYS.DBMS_XS_SESSIONS.DETACH_SESSION;
  SYS.DBMS_XS_SESSIONS.DESTROY_SESSION(sessionid);
END;
/
```

Is\_External

Dynamic Role  
List

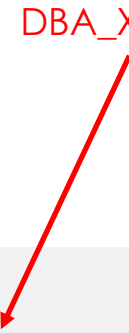
# Real Application Security

## Direct Login Application Users

- Similar from an end user's point of view
- Can be used as **end users**
- Can be used for **applications**
  - Technical users that require a strict security context
  - **Aimed at applications that aren't able to handle RAS session management operations (create / attach / detach / destroy session)**

```
BEGIN
  SYS.XS_PRINCIPAL.CREATE_USER('BOB');
  SYS.XS_PRINCIPAL.SET_PASSWORD('BOB','2Hrd2Guess');
  SYS.XS_PRINCIPAL.GRANT_ROLES('BOB','XSCONNECT');
  SYS.XS_PRINCIPAL.GRANT_ROLES('BOB','HR_MANAGER');
END;
/

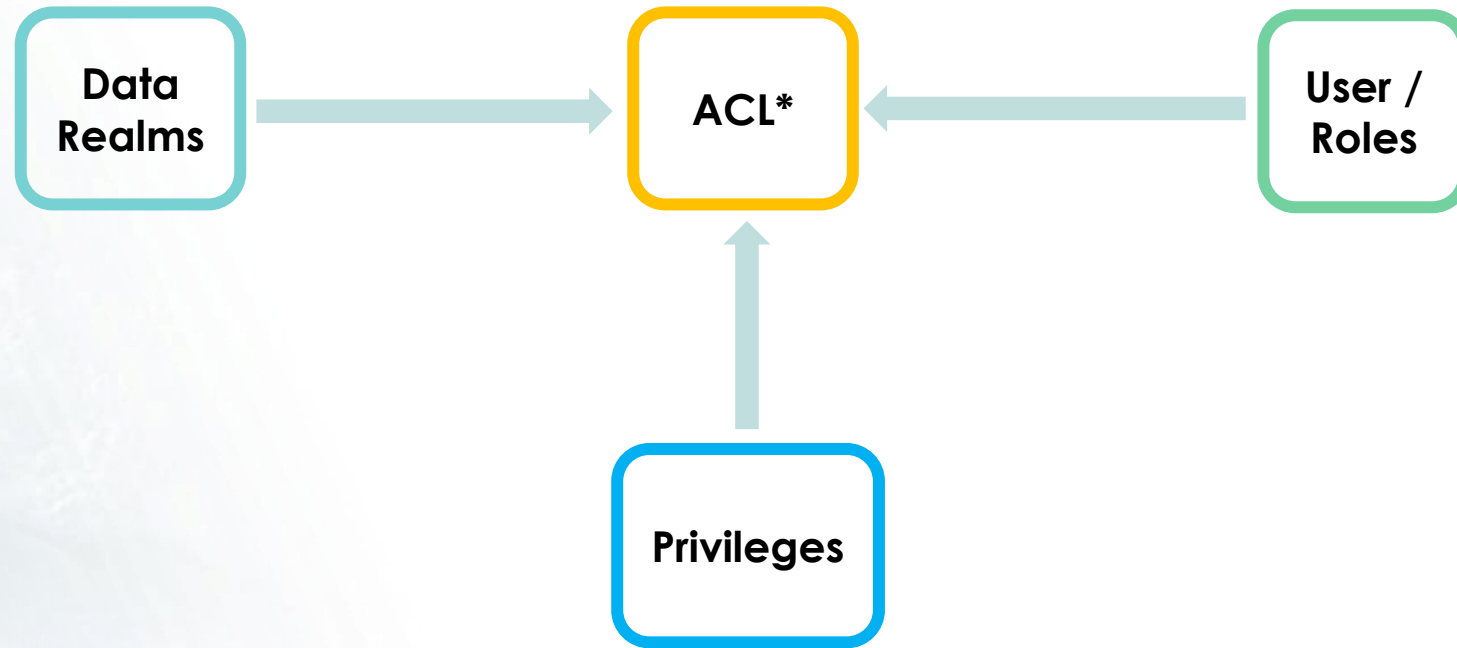
SQL> CONNECT BOB/2Hrd2Guess@MY_23_AI_DB
```



DBA\_XS\_USERS

# RAS Components

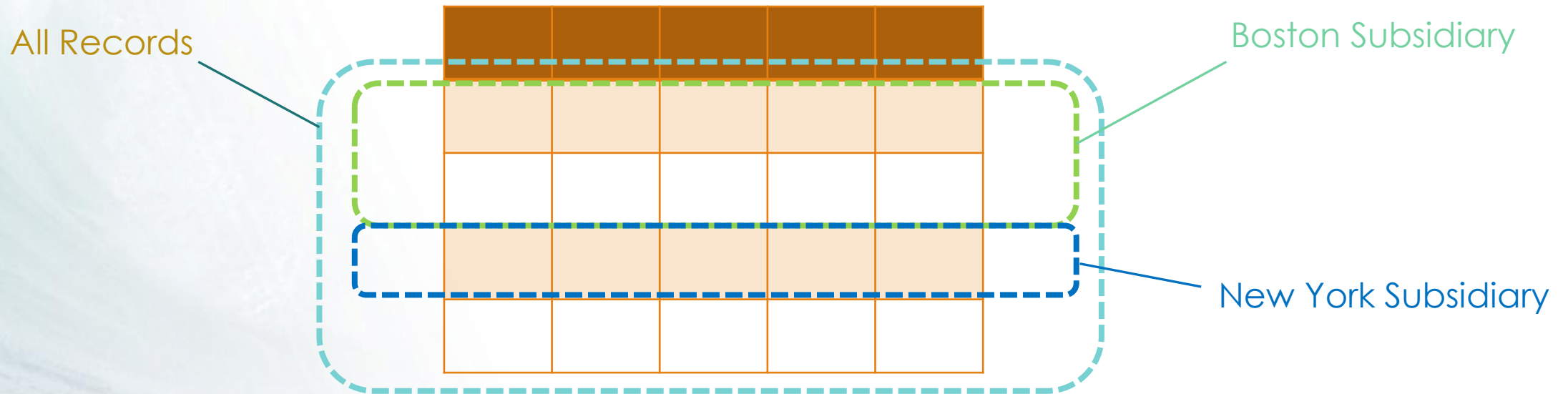
The 3 dimensions of Data Security



\* Access Control List

# RAS Components

A **realm** is nothing more than a **group of rows** representing a specific set of **business objects**



# RAS Components

Realm + privileges – Visual example

Name	Manager	National ID	Salary	Mobile
Adam Fripp	Steve Stiles			650-123-3234
Neena Kochar	Steve Stiles			650-124-8234
Nancy Greenberg	Neena Kochar	000-51-4569	120300	515-123-4567
Luis Popp	Nancy Greenberg		69000	515-123-4234
John Chen	Nancy Greenberg		82000	515-123-8181
Daniel Faviat	Nancy Greenberg		9000	515-123-7777

Self Data Realm → Nancy Greenberg

Manager Data Realm [ Luis Popp, John Chen, Daniel Faviat ]

Public Data Realm [ Entire Table ]

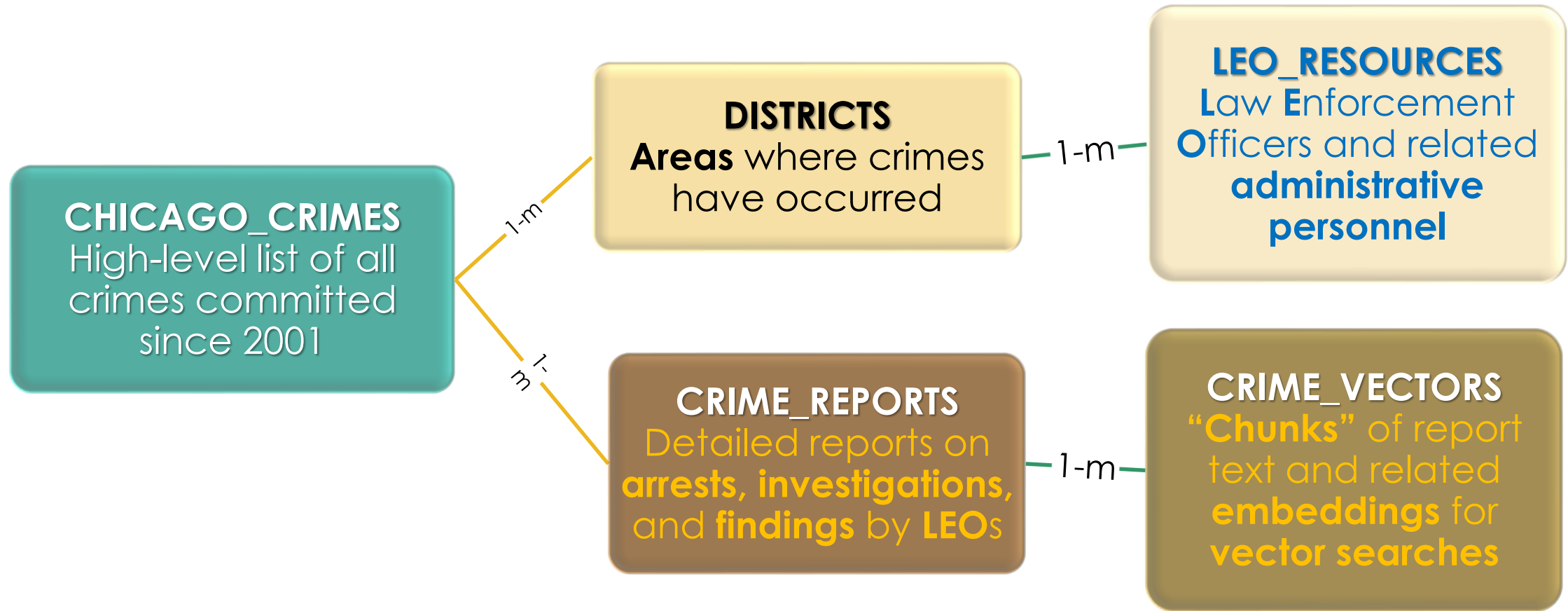
Select ID Privilege ↑ National ID

Select Salary Privilege ↑ Salary



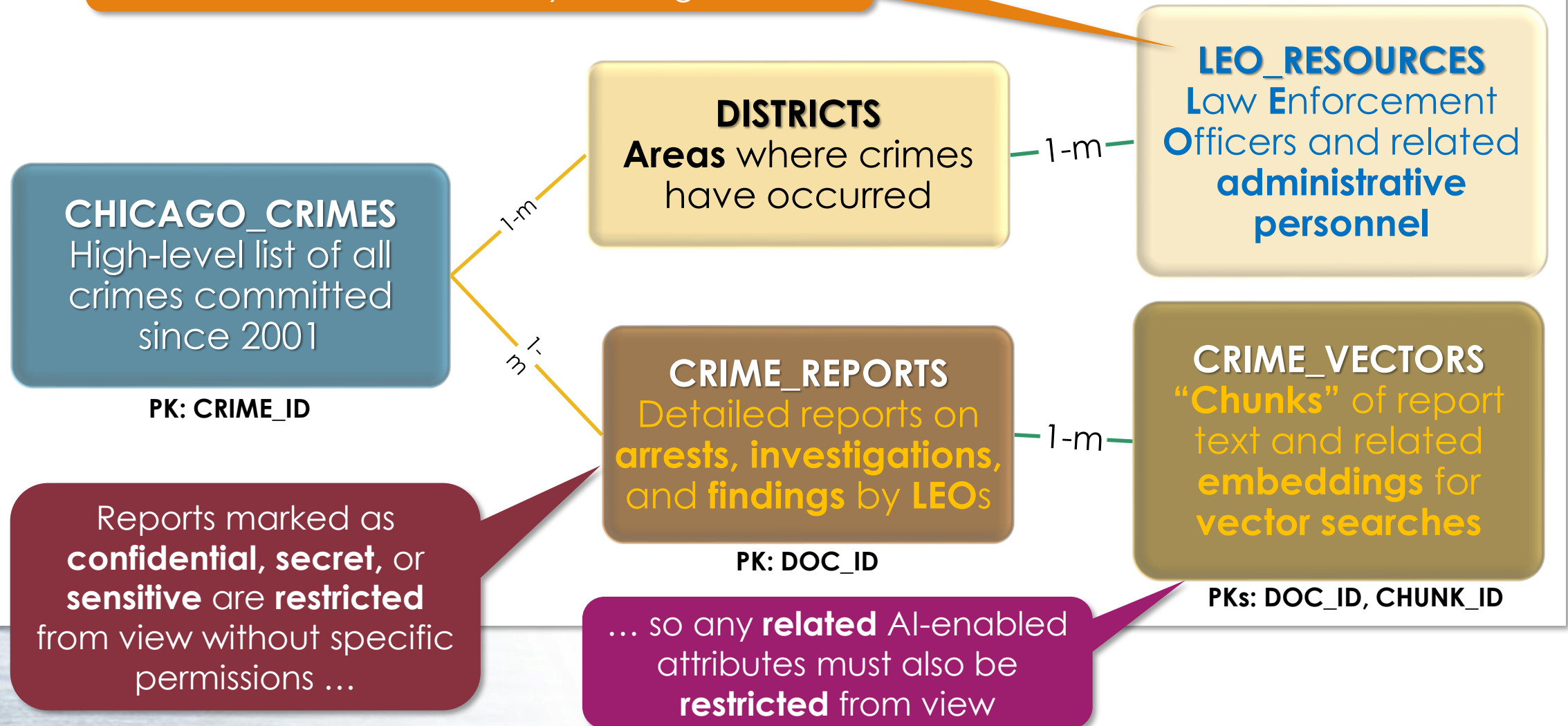
# **A Practical Demo: Probing Chicago Police Department (CPD) Crime Data With RAG In APEX**

# CPD: Schema

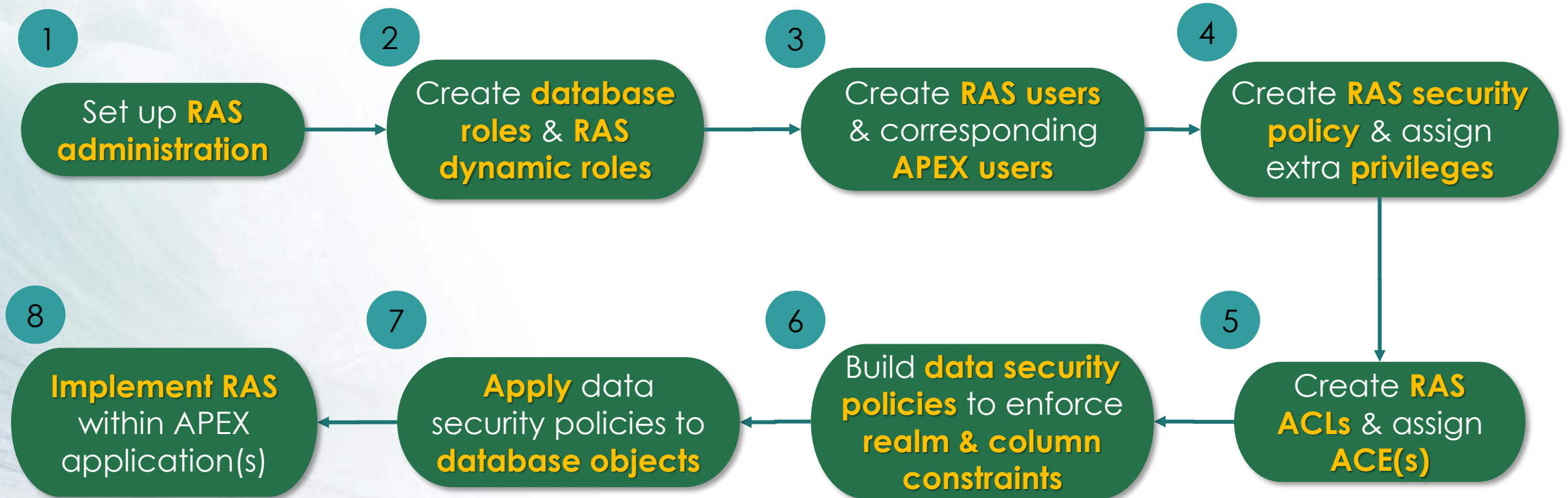


# CPD: Security Requirements

LEOs are generally limited from viewing crimes **outside** of the district they're assigned to



# RAS In APEX: Installation Checklist



# Setting up RAS Administration

1

```
BEGIN
  XS_ADMIN_CLOUD_UTIL.GRANT_SYSTEM_PRIVILEGE(
    'PROVISION', 'ADMIN');
  XS_ADMIN_CLOUD_UTIL.GRANT_SYSTEM_PRIVILEGE(
    'ADMIN_ANY_SEC_POLICY', 'ADMIN');
END;
```

This gives the **ADMIN** user sufficient permissions to manage all **RAS tasks**

Create table to control **security access limits** for **RAS User accounts**

These entries control access to data within specific **police districts** & whether **any** sensitive data can be viewed at specific **levels** of sensitivity

```
DROP TABLE IF EXISTS cpd.ras_secured_users PURGE;
CREATE TABLE IF NOT EXISTS cpd.ras_secured_users (
  rsu_username VARCHAR2(25) NOT NULL
, rsu_keytype VARCHAR2(12) NOT NULL
, rsu_keyvalue VARCHAR2(12) NOT NULL
, rsu_comment VARCHAR2(128));

ALTER TABLE cpd.ras_secured_users
  ADD CONSTRAINT ras_secured_users_pk
  PRIMARY KEY (
    rsu_username
  , rsu_keytype
  , rsu_keyvalue)
  USING INDEX (
    CREATE UNIQUE INDEX cpd.ras_secured_users_pk_idx
    ON cpd.ras_secured_users (
      rsu_username
    , rsu_keytype
    , rsu_keyvalue));
```



# RAS Users & Security Permissions

1

RSU_USERNAME	RSU_KEYTYPE	RSU_KEYVALUE	RSU_COMMENT
LSNELLING	DISTRICT	1	Chief sees everything
LSNELLING	DISTRICT	2	Chief sees everything
LSNELLING	DISTRICT	3	Chief sees everything
LSNELLING	DISTRICT	. . .	Chief se
LSNELLING	DISTRICT	22	Chief se
LSNELLING	DISTRICT	24	Chief se
LSNELLING	DISTRICT	25	Chief se
LSNELLING	VISIBILITY	PII	Full P
LSNELLING	SENSITIVITY	CONFIDENTIAL	Chief se
LSNELLING	SENSITIVITY	NORMAL	Chief se
LSNELLING	SENSITIVITY	SECRET	Chief se
LSNELLING	SENSITIVITY	SENSITIVE	Chief se

**LSNELLING** can see everything, everywhere

RSU_USERNAME	RSU_KEYTYPE	RSU_KEYVALUE	RSU_COMMENT
SNGUYEN	DISTRICT	1	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	12	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	18	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	19	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	20	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	24	PR Assistant - Area 3 only
SNGUYEN	VISIBILITY	PII	Full PII allowed
SNGUYEN	SENSITIVITY	NORMAL	Full PII allowed
SNGUYEN	SENSITIVITY	SENSITIVE	Full PII allowed
SNGUYEN	SENSITIVITY	CONFIDENTIAL	Full PII allowed
SNGUYEN	SENSITIVITY	SECRET	Full PII allowed

Meanwhile, **SNGUYEN** can see documents at all sensitivities, but **only in 6 districts**

# RAS Users & Security Permissions

1

RSU_USERNAME	RSU_KEYTYPE	RSU_KEYVALUE	RSU_COMMENT
LSNELLING	DISTRICT	1	Chief sees everything
LSNELLING	DISTRICT	2	Chief sees everything
LSNELLING	DISTRICT	3	Chief sees everything
LSNELLING	DISTRICT	. . .	Chief se
LSNELLING	DISTRICT	22	Chief se
LSNELLING	DISTRICT	24	Chief se
LSNELLING	DISTRICT	25	Chief se
LSNELLING	VISIBILITY	PII	Full P
LSNELLING	SENSITIVITY	CONFIDENTIAL	Chief se
LSNELLING	SENSITIVITY	NORMAL	Chief se
LSNELLING	SENSITIVITY	SECRET	Chief se
LSNELLING	SENSITIVITY	SENSITIVE	Chief se

RSU_USERNAME	RSU_KEYTYPE	RSU_KEYVALUE	RSU_COMMENT
SNGUYEN	DISTRICT	1	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	12	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	18	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	19	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	20	PR Assistant - Area 3 only
SNGUYEN	DISTRICT	24	PR Assistant - Area 3 only
SNGUYEN	VISIBILITY	PII	Full PII allowed
SNGUYEN	SENSITIVITY	NORMAL	Full PII allowed
SNGUYEN	SENSITIVITY	SENSITIVE	Full PII allowed
SNGUYEN	SENSITIVITY	CONFIDENTIAL	Full PII allowed
SNGUYEN	SENSITIVITY	SECRET	Full PII allowed

RSU_USERNAME	RSU_KEYTYPE	RSU_KEYVALUE	RSU_COMMENT
NKURCHAWSKA	DISTRICT	2	Student - Area 1 only
NKURCHAWSKA	DISTRICT	3	Student - Area 1 only
NKURCHAWSKA	DISTRICT	7	Student - Area 1 only
NKURCHAWSKA	DISTRICT	8	Student - Area 1 only
NKURCHAWSKA	DISTRICT	9	Student - Area 1 only
NKURCHAWSKA	VISIBILITY	NONE	Student - No PII allowed
WJACKSON	DISTRICT	8	New district, 11-20-25
WJACKSON	DISTRICT	10	old district
WJACKSON	VISIBILITY	PII	Full PII allowed
WJACKSON	SENSITIVITY	NORMAL	Limited per internal affairs
WJACKSON	SENSITIVITY	SENSITIVE	Limited per internal affairs

**NKURCHAWSKA** has extremely limited viewing privileges ...

... and **WJACKSON** has been assigned some interestingly tight privileges

# Create Database Roles & RAS Dynamic Roles

2

```
CREATE ROLE cpd_readonly;
GRANT SELECT ON cpd.chicago_crimes TO cpd_readonly;
GRANT SELECT ON cpd.cpd_districts TO cpd_readonly;
GRANT SELECT ON cpd.iucr_codes TO cpd_readonly;
GRANT SELECT ON cpd.leo_resources TO cpd_readonly;
GRANT SELECT ON cpd.crime_reports TO cpd_readonly;
GRANT SELECT ON cpd.crime_vectors TO cpd_readonly;
GRANT SELECT ON cpd.ras_secured_users TO cpd_readonly;

CREATE ROLE cpd_fullddl;
GRANT cpd_readonly TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.chicago_crimes TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.cpd_districts TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.iucr_codes TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.leo_resources TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.crime_reports TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.crime_vectors TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.ras_secured_users TO cpd_fullddl;

CREATE ROLE cpd_pii_limited;
GRANT SELECT ON cpd.chicago_crimes TO cpd_pii_limited;
GRANT SELECT ON cpd.cpd_districts TO cpd_pii_limited;
GRANT SELECT ON cpd.iucr_codes TO cpd_pii_limited;
GRANT SELECT ON cpd.leo_resources TO cpd_pii_limited;
GRANT SELECT ON cpd.crime_reports TO cpd_pii_limited;
GRANT SELECT ON cpd.crime_vectors TO cpd_pii_limited;
GRANT SELECT ON cpd.ras_secured_users TO cpd_pii_limited;
```

# Create Database Roles & RAS Dynamic Roles

2

```
CREATE ROLE cpd_readonly;
GRANT SELECT ON cpd.chicago_crimes TO cpd_readonly;
GRANT SELECT ON cpd.cpd_districts TO cpd_readonly;
GRANT SELECT ON cpd.iucr_codes TO cpd_readonly;
GRANT SELECT ON cpd.leo_resources TO cpd_readonly;
GRANT SELECT ON cpd.crime_reports TO cpd_readonly;
GRANT SELECT ON cpd.crime_vectors TO cpd_readonly;
GRANT SELECT ON cpd.ras_secured_users TO cpd_readonly;

CREATE ROLE cpd_fullddl;
GRANT cpd_readonly TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.chicago_crimes TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.cpd_districts TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.iucr_codes TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.leo_resources TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.crime_reports TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.crime_vectors TO cpd_fullddl;
GRANT INSERT, UPDATE, DELETE ON cpd.ras_secured_users TO cpd_fullddl;

CREATE ROLE cpd_pii_limited;
GRANT SELECT ON cpd.chicago_crimes TO cpd_pii_limited;
GRANT SELECT ON cpd.cpd_districts TO cpd_pii_limited;
GRANT SELECT ON cpd.iucr_codes TO cpd_pii_limited;
GRANT SELECT ON cpd.leo_resources TO cpd_pii_limited;
GRANT SELECT ON cpd.crime_reports TO cpd_pii_limited;
GRANT SELECT ON cpd.crime_vectors TO cpd_pii_limited;
GRANT SELECT ON cpd.ras_secured_users TO cpd_pii_limited;
```

Database roles enable **SELECT** and **DML** access to **database objects** ...



# Create Database Roles & RAS Dynamic Roles

2

```
CREATE ROLE cpd_readonly;
GRANT SELECT ON cpd.chi TO cpd_readonly;
GRANT SELECT ON cpd.cpd TO cpd_readonly;
GRANT SELECT ON cpd.iuc TO cpd_readonly;
GRANT SELECT ON cpd.leo TO cpd_readonly;
GRANT SELECT ON cpd.cri TO cpd_readonly;
GRANT SELECT ON cpd.cri TO cpd_readonly;
GRANT SELECT ON cpd.ras TO cpd_readonly;

CREATE ROLE cpd_fulldm1;
GRANT cpd_readonly TO cpd_fulldm1;

CREATE ROLE cpd_pii_limited;
GRANT SELECT ON cpd.chi TO cpd_pii_limited;
GRANT SELECT ON cpd.cpd TO cpd_pii_limited;
GRANT SELECT ON cpd.iuc TO cpd_pii_limited;
GRANT SELECT ON cpd.leo TO cpd_pii_limited;
GRANT SELECT ON cpd.cri TO cpd_pii_limited;
GRANT SELECT ON cpd.cri TO cpd_pii_limited;
GRANT SELECT ON cpd.ras TO cpd_pii_limited;

CREATE ROLE cpd_dwro;
GRANT SELECT ON cpd.chi TO cpd_dwro;
GRANT SELECT ON cpd.cpd TO cpd_dwro;
GRANT SELECT ON cpd.iuc TO cpd_dwro;
GRANT SELECT ON cpd.leo TO cpd_dwro;
GRANT SELECT ON cpd.cri TO cpd_dwro;
GRANT SELECT ON cpd.cri TO cpd_dwro;
GRANT SELECT ON cpd.ras TO cpd_dwro;

BEGIN
  SYS.XS_PRINCIPAL.CREATE_DYNAMIC_ROLE(
    name      => 'CPD_DYN_FULL_ROLE'
    ,duration  => NULL, scope => XS_PRINCIPAL.SESSION_SCOPE
    ,description => 'Dynamic Role - permits full SELECT access and DML access to all data,
    including viewing sensitive data in crime reports and vectorized embeddings'
    ,acl       => NULL
  );
  SYS.XS_PRINCIPAL.CREATE_DYNAMIC_ROLE(
    name      => 'CPD_DYN_PII_ROLE'
    ,duration  => NULL, scope => XS_PRINCIPAL.SESSION_SCOPE
    ,description => 'Dynamic Role - permits full SELECT access to all data, +including+ viewing
    sensitive crime reports and related and any related VECTOR embeddings based
    on permitted sensitivity level(s)'
    ,acl       => NULL
  );
  SYS.XS_PRINCIPAL.CREATE_DYNAMIC_ROLE(
    name      => 'CPD_DYN_DWRO_ROLE'
    ,duration  => NULL, scope => XS_PRINCIPAL.SESSION_SCOPE
    ,description => 'Dynamic Role - permits full SELECT access to all data,
    but +restricts+ viewing sensitive data in crime reports and
    vectorized embeddings'
    ,acl       => NULL
  );
END;

GRANT cpd_fulldm1 TO cpd_dyn_full_role;
GRANT cpd_pii_limited TO cpd_dyn_pii_role;
GRANT cpd_readonly TO cpd_dyn_dwro_role;
```

... while **RAS dynamic roles** control access to **database roles**



# Create RAS & APEX Users

3

```
DECLARE
  CURSOR curUserIDs IS
    SELECT userid, leo_lname, leo_fname FROM cpd.leo_resources;
BEGIN
  FOR u IN curUserIDs
    LOOP
      BEGIN
        SYS.XS_PRINCIPAL.CREATE_USER (
          name          => u.userid
        ,schema         => 'CPD'
        ,description    => 'CPD Internal User: ' || u.leo_lname || ',' || u.leo_fname);
        SYS.XS_PRINCIPAL.GRANT_ROLES (grantee => u.userid, role => 'XSCONNECT');
        SYS.DBMS_XS_PRINCIPALS.SET_PASSWORD(
          ,username => u.userid
          ,password => 'S3c-re_P5w0rd'
          ,type => XS_PRINCIPAL.XS_SHA512);
      END;
    END LOOP;
  END;
```

This creates **RAS user accounts** within the database for all identified **LEOs** ...

# Create RAS & APEX Users

3

```
DECLARE
CURSOR curUserIDs IS
  SELECT userid, leo_lname, leo_fname FROM cpd.leo_resources;
BEGIN
  FOR u IN curUserIDs
  LOOP
    BEGIN
      SYS.XS_PRINCIPAL.
        name      =
        ,schema    =
        ,description =
      SYS.XS_PRINCIPAL.
      SYS.DBMS_XS_PRINC
        ,username => u.
        ,password => 'S
        ,type => XS_PRI
    END;
  END LOOP;
END;
```

... and this creates **RAS APEX user accounts** for **LEOs** as well

```
DECLARE
  l_wid APEX_WORKSPACES.WORKSPACE_ID%TYPE;
CURSOR curUserIDs IS
  SELECT userid, leo_lname, leo_fname FROM cpd.leo_resources;
BEGIN
  SELECT workspace_id INTO l_wid FROM apex_workspaces WHERE workspace = 'CPD';
  APEX_UTIL.SET_SECURITY_GROUP_ID(l_wid);
  FOR u IN curUserIDs
  LOOP
    APEX_UTIL.CREATE_USER(
      p_user_name      => u.userid
      ,p_email_address => 'jim.czuprynski@gmail.com'
      ,p_web_password  => 'S3c-re_P5w0rd'
      ,p_change_password_on_first_use => 'N'
      ,p_description   => 'CPD Internal User: ' || u.leo_lname || ',' || u.leo_fname
      ,p_account_locked => 'N'
      ,p_developer_privs => NULL
    );
  END LOOP;
  COMMIT;
END;
```

This creates **RAS user accounts** within the database for all identified **LEOs** ...

# Create RAS Security Policy

4

```
BEGIN
  SYS.XS_SECURITY_CLASS.CREATE_SECURITY_CLASS(
    name          => 'CPD.SECCLS'
    ,parent_list => xs$name_list('SYS.DML')
    ,priv_list    => NULL
    ,description => 'CPD Security Class');
```

This creates a **special additional privilege** we can assign to specific ACLs in the next step

```
  SYS.XS_SECURITY_CLASS.ADD_PRIVILEGES(
    sec_class => 'CPD.SECCLS'
    ,priv => 'VIEW_RESTRICTED_PII'
    ,implied_priv_list => XS$NAME_LIST('"SELECT"')
    ,description => 'Permits viewing of restricted crime data within selected tables');
END;
```

# Create RAS ACLs & Assign ACEs

5

```
DECLARE
  aces    XS$ACE_LIST := XS$ACE_LIST();
BEGIN
```

```
  aces.EXTEND(1);
  aces(1) :=
    XS$ACE_TYPE(
      privilege_list => XS$NAME_LIST('SELECT', 'INSERT', 'UPDATE', 'DELETE', 'VIEW_RESTRICTED_PII')
      ,principal_name => 'CPD_DYN_FULL_ROLE');
  SYS.XS_ACL.CREATE_ACL(
    name          => 'CPD.SUPR_ACL'
    ,ace_list     => aces
    ,sec_class    => 'CPD.SECCLS'
    ,description  => 'CPD Supervisor-Level ACL. Full DML privileges are permitted plus all
                     PII and RAG viewing is permitted');
  . . .
```

ACL **SUPR\_ACL** offers **full DML, SELECT**, and **PII viewing privileges**

# Create RAS ACLs & Assign ACEs

5

```
DECLARE
  aces
BEGIN
  aces.EX
  aces(1)
    XS$AC
    pri
    ,pri
  SYS.XS_
    name
    ,ace_l
    ,sec_c
    ,descr
  . . .
```

```
. . .
aces(1) :=
  XS$ACE_TYPE(
    privilege_list => XS$NAME_LIST('SELECT','VIEW_RESTRICTED_PII')
    ,principal_name => 'CPD_DYN_PII_ROLE');
SYS.XS_ACL.CREATE_ACL(
  name          => 'CPD.PII_ACL'
  ,ace_list     => aces
  ,sec_class    => 'CPD.SECCLS'
  ,description  => 'CPD PII-Limited ACL. All PII viewing is permitted');

aces(1) :=
  XS$ACE_TYPE(
    privilege_list => XS$NAME_LIST('SELECT')
    ,principal_name => 'CPD_DYN_DWRO_ROLE');
SYS.XS_ACL.CREATE_ACL(
  name          => 'CPD.DWRO_ACL'
  ,ace_list     => aces
  ,sec_class    => 'CPD.SECCLS'
  ,description  => 'CPD Read-Only ACL. No PII or RAG viewing is permitted');

END;
```

rs full  
viewing

ACL **PII\_ACL** allows  
SELECT privileges  
with PII access

ACL **DWRO\_ACL** allows  
limited SELECT privileges but  
denies PII access



# Build DSP For CHICAGO\_CRIMES

6

```
DECLARE
  realms XS$REALM_CONSTRAINT_LIST := XS$REALM_CONSTRAINT_LIST();
BEGIN
  realms.EXTEND(3);

  -- SUPR_ACL: Full access to all table contents (no restrictions based on data values)
  realms(1) :=
    XS$REALM_CONSTRAINT_TYPE(
      realm => '1=1'
      ,acl_list => XS$NAME_LIST('CPD.SUPR_ACL'));

  -- PII_ACL: Permits access to rows only if CPD.CHICAGO_CRIMES.DISTRICT is found
  -- within a specified subset of values found in CPD.RAS_SECURED_USERS for the
  -- pertinent user account
  realms(2) :=
    XS$REALM_CONSTRAINT_TYPE(
      realm =>
        'DISTRICT IN (
          SELECT RSU_KEYVALUE
          FROM CPD.RAS_SECURED_USERS
          WHERE RSU_KEYTYPE = ''DISTRICT''
            AND RSU_USERNAME = XS_SYS_CONTEXT(''XS$SESSION'', ''SESSION_XS_USER''))'
      ,acl_list => XS$NAME_LIST('CPD.PII_ACL'));
  . . .
```

For **SUPR\_ACL**, setting its realm to a Boolean TRUE condition essentially **allows all rows** to be accessed

For **PII\_ACL**, its realm applies this selection criteria **to any query**, thus limiting access to **only those rows a user is allowed to see**

# Build DSP For CHICAGO\_CRIMES

```

DECLARE
  realms XS$REALM_CONSTRAINT_LIST := XS$REALM_CONSTRAINT_LIST();
BEGIN
  realms.EXTEND(3);

  -- DWRO_ACL: Permits access to rows only if if CPD.CHICAGO_CRIMES.DISTRICT is found
  -- within a specified subset of values found in CPD.RAS_SECURED_USERS for the
  -- pertinent user account, and also +RESTRICTS+ viewing of selected PII columns
  realms(3) :=
    XS$REALM_CONSTRAINT_TYPE(
      realm =>
        'DISTRICT IN (
          SELECT RSU_KEYVALUE
          FROM CPD.RAS_SECURED_USERS
          WHERE RSU_KEYTYPE = 'DISTRICT'
          AND RSU_USERNAME = XS_SYS_CONTEXT('XS$SESSION', 'SESSION_XS_USER'))'
      ,acl_list => XS$NAME_LIST('CPD.DWRO_ACL'));

  SYS.XS_DATA_SECURITY.CREATE_POLICY(
    name                => 'CPD.CHICAGO_CRIMES_DS'
    ,realm_constraint_list => realms
    ,column_constraint_list => NULL
    ,description         => 'Security policy limiting access to CPD.CHICAGO_CRIMES');

END;

```

The realm for **DWRO\_ACL** is essentially identical to the one for **PII\_ACL**

This creates the Data Security Policy for table **CHICAGO\_CRIMES**

# Build DSP For CRIME\_REPORTS

6

```
DECLARE
  realms  XS$REALM_CONSTRAINT_LIST  := XS$REALM_CONSTRAINT_LIST();
  cols    XS$COLUMN_CONSTRAINT_LIST := XS$COLUMN_CONSTRAINT_LIST();
BEGIN
  realms.EXTEND(1);
  cols.EXTEND(2);
  realms(1) :=
    XS$REALM_CONSTRAINT_TYPE(
      realm => 'SENSITIVITY IN (
        SELECT RSU_KEYVALUE
        FROM CPD.RAS_SECURED_USERS
        WHERE RSU_KEYTYPE = 'SENSITIVITY'
        AND RSU_USERNAME = XS_SYS_CONTEXT('XS$SESSION', 'SESSION_XS_USER'))'
      ,acl_list => XS$NAME_LIST('CPD.PII_ACL'));

  cols(1) :=
    XS$COLUMN_CONSTRAINT_TYPE(
      column_list => XS$LIST('SENSITIVITY')
      ,privilege => 'VIEW_RESTRICTED_PII');
  cols(2) :=
    XS$COLUMN_CONSTRAINT_TYPE(
      column_list => XS$LIST('CRIME_DOCUMENT')
      ,privilege => 'VIEW_RESTRICTED_PII');
  SYS.XS_DATA_SECURITY.CREATE_POLICY(
    name => 'CPD.CRIME_REPORTS_DS'
    ,realm_constraint_list => realms
    ,column_constraint_list => cols
    ,description => 'Controls access to CPD.CRIME_REPORTS'
  );
END;
```

This **realm constraint** looks at the RAS control table for which privileges (e.g. **SECRET, NORMAL**) to enforce for rows a RAS user may view

These **column constraints** determine which columns are restricted should the RAS user not have been granted the **VIEW\_RESTRICTED\_PII** privilege

# Build DSP For CRIME\_VECTORS

6

```
DECLARE
  realms  XS$REALM_CONSTRAINT_LIST := XS$REALM_CONSTRAINT_LIST();
  cols    XS$COLUMN_CONSTRAINT_LIST := XS$COLUMN_CONSTRAINT_LIST();
BEGIN
  cols.EXTEND(3);

  realm := XS$REALM_CONSTRAINT_LIST(
    XS$REALM_CONSTRAINT_TYPE(
      parent_schema => 'CPD', parent_object => 'CRIME_REPORTS'
      ,key_list =>
        XS$KEY_LIST(
          XS$KEY_TYPE(primary_key => 'DOC_ID'
            ,foreign_key => 'DOC_ID', foreign_key_type => 1))));

  cols(1) := XS$COLUMN_CONSTRAINT_TYPE(
    column_list => XS$LIST('CHUNK_ID'), privilege => 'VIEW_RESTRICTED_PII');
  cols(2) := XS$COLUMN_CONSTRAINT_TYPE(
    column_list => XS$LIST('DOC_CHUNK'), privilege => 'VIEW_RESTRICTED_PII');
  cols(3) := XS$COLUMN_CONSTRAINT_TYPE(
    column_list => XS$LIST('EMBEDDINGS'), privilege => 'VIEW_RESTRICTED_PII');

  SYS.XS_DATA_SECURITY.CREATE_POLICY(
    name => 'CPD.CRIME_VECTORS_DS'
    ,realm_constraint_list => realm
    ,column_constraint_list => cols
    ,description =>
      'Controls access to CPD.CRIME_VECTORS based on
      parent-child relationship TO CPD.CRIME_REPORTS');
END;
```

This **realm constraint** uses the foreign key relationship between rows in the **CRIME\_REPORTS** table that are **parents** of rows in the **CRIME\_VECTORS** table to enforce viewing restrictions

These **column constraints** block the use of document chunks and their related embeddings **during generative AI or RAG operations**

# Apply Security Policies to Objects

7

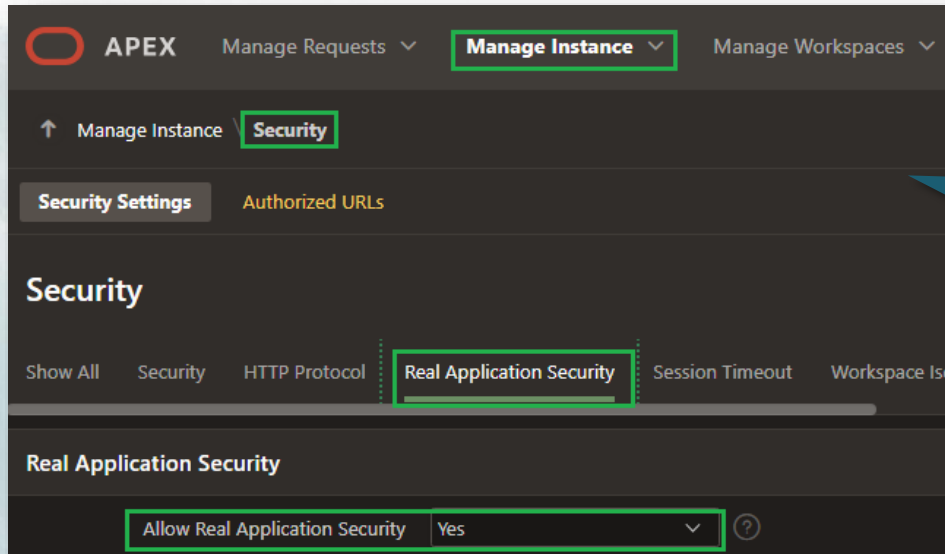
```
BEGIN
  SYS.XS_DATA_SECURITY.APPLY_OBJECT_POLICY(
    policy => 'CPD.CHICAGO_CRIMES_DS'
    , schema => 'CPD'
    , object => 'CHICAGO_CRIMES');
  SYS.XS_DATA_SECURITY.APPLY_OBJECT_POLICY(
    policy => 'CPD.CRIME_REPORTS_DS'
    , schema => 'CPD'
    , object => 'CRIME_REPORTS');
  SYS.XS_DATA_SECURITY.APPLY_OBJECT_POLICY(
    policy => 'CPD.CRIME_VECTORS_DS'
    , schema => 'CPD'
    , object => 'CRIME_VECTORS');
END;
```

Finally, this step applies the **data security policies** to their appropriate **database objects**



# Enabling RAS For APEX Applications

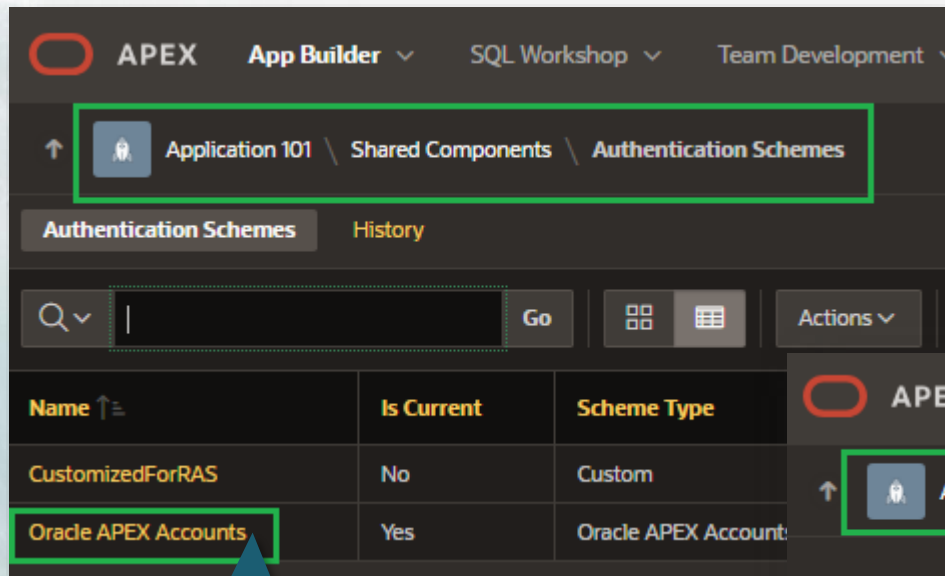
8



You may need to have your APEX administrator **enable your APEX instance for RAS**

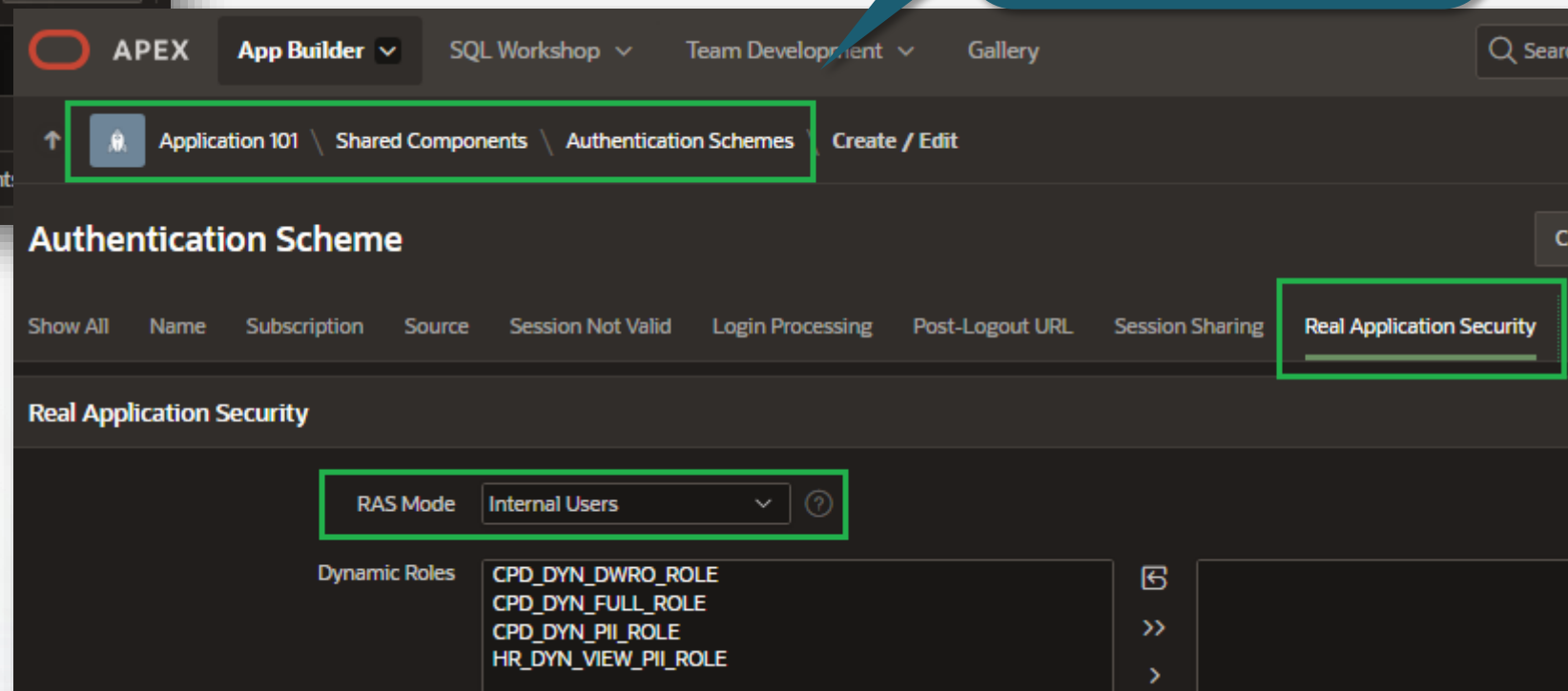
# Enabling RAS For APEX Applications

8



Open the **Authentication Schemes** page for **APEX Accounts** ...

... navigate to the **Real Application Security** option, and then choose **Internal Users** to activate RAS



# Enabling RAS For APEX Applications

8

Authentication Scheme

Show All Name Subscription Source Session Not Valid Login Processing Post-Logout URL Session Sharing Real Application Security

There are no subscribers to this authentication scheme.

Source

PL/SQL Code ?

```
1  PROCEDURE post_auth IS
2    visibility_level VARCHAR2(12);
3  BEGIN
4
5    visibility_level := 'FULL';
6
7    SELECT rsu_keyvalue
8    INTO visibility_level
9    FROM cpd.ras_secured_users
10   WHERE rsu_username = :APP_USER
11         AND rsu_keytype = 'VISIBILITY';
12
13   IF visibility_level = 'PII' THEN
14     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_PII_ROLE'));
15   ELSIF visibility_level = 'FULL' THEN
16     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_FULL_ROLE'));
17   ELSE
18     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_DWRO_ROLE'));
19   END IF;
20
21 EXCEPTION
22   WHEN NO_DATA_FOUND THEN
23     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_DWRO_ROLE'));
24 END;
```

Add a post-authentication PL/SQL procedure that assigns an appropriate **RAS Dynamic Role** based on the setting in **RAS\_SECURED\_USERS** for PII view privileges ...

# Enabling RAS For APEX Applications

8

The image shows a screenshot of the Oracle APEX interface. On the left, the 'PL/SQL Code' editor displays a procedure named 'post\_auth' with the following code:

```
1 PROCEDURE post_auth IS
2   visibility_level VARCHAR2(12);
3 BEGIN
4
5   visibility_level := 'FULL';
6
7   SELECT rsu_keyvalue
8     INTO visibility_level
9     FROM cpd.ras_secured_users
10    WHERE rsu_username = :APP_USER
11          AND rsu_keytype = 'VISIBILITY';
12
13   IF visibility_level = 'PII' THEN
14     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_PII_ROLE'));
15   ELSIF visibility_level = 'FULL' THEN
16     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_FULL_ROLE'));
17   ELSE
18     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_DWRO_ROLE'));
19   END IF;
20
21 EXCEPTION
22   WHEN NO_DATA_FOUND THEN
23     APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS (p_group_names => APEX_T_VARCHAR2('CPD_DYN_DWRO_ROLE'));
24 END;
```

On the right, the 'Authentication Scheme' configuration page is shown. The 'Post-Authentication Procedure Name' field is highlighted with a green box and contains the value 'POST\_AUTH'. A green arrow points to this field. A blue callout bubble points to the 'Post-Authentication Procedure Name' field with the text: "... and then assign the procedure as the **Post-Authentication** procedure for the authentication scheme". Another blue callout bubble points to the 'POST\_AUTH' value with the text: "Add a post-authentication PL/SQL procedure that assigns an appropriate **RAS Dynamic Role** based on the setting in **RAS\_SECURED\_USERS** for PII view privileges ...".



# **Live Demo: RAS Before You RAG**



# Real Application Security Advantages

- **Strong security**
  - Central enforcement – no need to reinvent the security wheel for each application
  - Audit end-user activity
- **Simpler, faster development**
  - Declarative policy, no coding ( unlike VPD!)
  - Easier Maintenance
  - Application Patterns
- **High-Performance Access Control**
  - Optimized within core database

# Useful Technical Resources

- [Real Application Security Java API Reference](#)
- [Real Application Security Session Service Java API Reference](#)
- [Real Application Security Administrator's and Developer's Guide](#)

# In-Depth RAS Coding Examples

- [Protecting Vectorised Data in RAG Applications with Oracle 23ai Real Application Security](#)
- [A Session Within a Session: Demystifying Real Application Security Contexts](#)
- [ACCSP Featured in Oracle's Business Innovations with Oracle APEX Series](#)
- Franco Ucci's AskTom session on RAS (video coming soon!)

# Meet the Real RAS Experts!



**Franco Ucci**

**Senior Director, Cloud Architects (Australia)**

[franco.ucci@oracle.com](mailto:franco.ucci@oracle.com)

Franco has held various positions in Oracle Corporation from being a Technical Consultant through to his current position, of Senior Director, where he helps drive Adoption of Digital Models utilizing Data, APIs, AI and Cloud Based Technologies.



**Thomas Minne**

**Data Security Black Belt**

[thomas.m.minne@oracle.com](mailto:thomas.m.minne@oracle.com)

Member of the EMEA Office of the CTO  
Helping customers around EMEA on Data Security Project, in a nutshell : all products & features around the Maximum Security Architecture of the Oracle Database. Leading the Real Application Security topic in my team, in EMEA and in the **universe**.

**Franco and Thomas are two of the best experts of practical RAS implementations at Oracle – don't hesitate to reach out to them directly if you have any other questions about RAS!**

# **RAS before RAG:** **Real Application Security** **Fundamentals** **for Gen AI Apps**



**December 3, 2025**

**Karen Cannell**  
CTHO  
TH Technology

[kcannell@thtechnology.com](mailto:kcannell@thtechnology.com)

**Jim Czuprynski**  
Chief StoryTeller  
Zero Defect Computing, Inc.

[jczuprynski@zerodefectcomputing.com](mailto:jczuprynski@zerodefectcomputing.com)