Oracle Mandatory Access Control for Property & Casualty Insurance - Auto Claims

Ajay Batra A0298397B
Elamparithi Kannan A0298319M
Indrayani Wanjari A0298405W
Pavin Rex Baskar A0298567A
Rishikesh Kadiyala A0298806L

***Abstract*—This report details the design and implementation of Mandatory Access Control (MAC) with the help of Oracle Label Security (OLS) for our auto claims application in the domain of Property & Casualty Insurance. The MAC and OLS ensure that sensitive data access to related tables is restricted based on user and data labels along with compartments and groups through different business process steps to enhance database security.**

***Index Terms*—Mandatory Access Control (MAC), Oracle Label Security (OLS), Database Security, Property & Casualty Insurance - Auto Claim, Labels, Compartments, Groups, Security Policies, ER Model.**

I Introduction

Mandatory Access Control (MAC) is a security model that enables system-wide policy that restricts access to resources based on user clearance levels and object sensitivity labels. Oracle Label Security (OLS) is an Oracle-based database-level implementation for MAC that is used to provide granular control over data access within Oracle databases. OLS complements MAC by enforcing data security policies at the database level, ensuring that only authorized users can access and manipulate sensitive information, thereby enhancing overall data protection and compliance.

II Application Overview

P&C Insurance domain is an insurance domain that protects against potential risks associated with ownership and liability.

- It encompasses a broad range of insurance products that cover both damage to property and legal responsibility for injuries or damage caused to others.
- Unlike life or health insurance, P&C insurance focuses on safeguarding assets and mitigating liability risks, both for individuals and businesses.

II-A Types of P&C Insurance

- Homeowner's insurance
- Auto insurance
- Workers Compensation
- Marine insurance

For our project purpose, we will be considering an auto insurance (vehicle insurance) application implementation. Regarding auto insurance, (or any other P&C insurance types), there are 2 main processes involved:

1. Policy creation and maintenance:

- A user/client takes his/her auto insurance with the company and the company creates an auto insurance policy for the client, based on various factors such as premium plan chosen, coverage and sub-coverage added for the plan, validity, etc.,
- The insurance company is responsible for notifying the policyholder on renewals and claim payments.

2. Claims:

- Process of filing and settling claims when policyholders experience losses covered under their insurance policies.
- Claims are typically filed after events such as accidents, property damage, or liability-related incidents. (We here focus on only accidents, vehicle theft, vandalism, and auto insurance liability (where the policyholder is at fault))
- Efficient claims handling is crucial for both insurers and policyholders, as it determines the financial compensation policyholders receive and plays a critical role in customer satisfaction and retention.
- The process of handling P&C insurance claims can be complex, involving claim initiation, investigation, assessment of damage or liability, settlement negotiation, and payment.

II-B Coverages available for auto policy

1. Collision coverage:

- Pays for repairing or replacing the insured's vehicle after an accident involving another car or a stationary object (like a tree or guardrail), regardless of who is at fault.
- In short words, covers expenses after colliding with any car/object.

2. Comprehensive coverage:

- Theft
- Vandalism
- Natural disasters (floods, hurricanes, earthquakes)
- Fire
- Falling objects (e.g., tree branches)
- Animal collisions (e.g., hitting a deer)
- Glass damage (e.g., broken windshields)

3. Uninsured Motorist coverage: Protects the policyholder if they are hit by a driver who does not have insurance. It typically covers medical expenses, lost wages, and other injury-related costs.

4. Medical Payments coverage (MedPay): For the health costs of the client and the passengers of the client's vehicle, injured in an accident, regardless of whose fault. This coverage is considered separate for some claim types, and is included with collision/comprehensive coverage for the purpose of simplicity. Our application takes the latter inclusion into consideration.

5. Liability coverage:

- Covers medical expenses, lost wages, pain and suffering, and legal fees when the insured driver causes injury or death to others in an accident.
- Covers the cost of repairing or replacing another person's property (typically their vehicle) that is damaged in an accident caused by the insured driver.

II-C Key steps involved in auto claim process

1. Claim filing:

1.a. Initial Notification: The policyholder contacts the insurer to report the loss or incident. This can be done through various channels, including phone, online portals, or mobile apps. The policyholder provides essential details, including the nature of the loss, the date, location, and any other relevant information.

1.b. Documentation: The policyholder is usually required to provide documentation to support the claim. This might include photos of the damage, repair estimates, police reports (for theft or accidents), and medical records (for injury claims).

2. Claim acknowledgment: The insurer/insurance company acknowledges receipt of the claim and assigns a claim adjuster to investigate. A claim number is typically generated, and the policyholder is informed about the next steps in the process.

3. Claim investigation:

3.a. Assessing the Damage and Validating the Claim: The adjuster will check the authenticity of the claim. The adjuster, as a representative of our company may interview the parties involved in an accident. He/she will investigate to determine who is legally responsible for the damage or injury. This may involve reviewing police reports, witness statements, and other evidence.

3.b. Checking Policy Coverage: Once the primary validation is done, the adjuster reviews the insurance policy and to verify whether the policy is in place or expired.

3.c. Estimating Costs: The adjuster estimates the repair or replacement costs for vehicle damage or calculates medical and legal expenses for liability claims and verifies it with the deductible limits available for the claimant/policyholder's auto policy. The adjuster also notifies the manager to assign vendors for any additional services required:

- Towing and labor (Roadside assistance),
- Salvage vendors (Scrap dealers),
  - When the repair cost exceeds the current value of the car, the insurance company declares it as a total loss and compensates the current value of the car to the client. The damaged car is given out to salvage vendors, from whom the insurance company will earn by selling the damaged vehicle.
- Glass Repair vendors.

4. Claim settlement and payment:

- The adjuster sends the claim validation and investigation results to the manager. He/she is responsible for processing full payments (if the limits are within range) or partial payments. The manager on behalf of our company issues a payment for the agreed-upon amount. Payment may be made to the policyholder directly or to the representative of the policyholder (on policy policyholder's death during the accident).
- After the payment, the claim is closed, and the record is maintained by our insurance company.

Figure 1 presents an overview of the property and casualty insurance - auto claim process, with the detailed steps varying according to different roles, and security clearance levels/labels as described in the following sections.

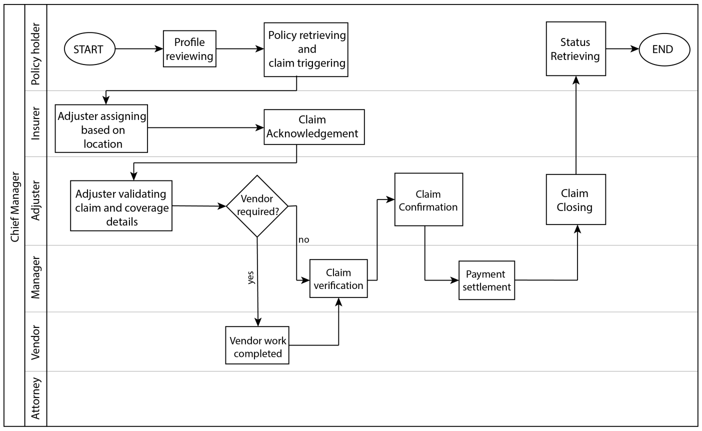


Fig. 1: Application flow overview

II-D Roles involved in auto claim process

1) **Policy holder**
   a) The person who holds an auto insurance policy in our company and who is capable of raising claims when a loss incident occurs.
2) **Adjuster**
   a) Representative of our company assigned to a particular claim to manage and process the full flow.
   b) Is responsible for the ownership of the assigned claim requirements.
   c) Works under a manager.
   d) Differs for country to country(one adjuster per country for our application use-case)
3) **Manager**
   a) Heads several adjusters.
   b) Only has access to a single continent's claim data which included all countries in which the company operates in that continent.
   c) Overseas the claim process and issues payments if required.
4) **Chief Manager**
   a) Acts as the Head of all regional managers and has full ownership of all the claim data across continents, registered with the company.
   b) Can oversee the End-to-end claim process flow.
5) **Vendor**
   a) Assigned by the manager.
   b) Is responsible for carrying out third-party services.

c) Each Country has a large-scale vendor for each of three major requirements: Towing& Labor, Salvage and Glass Repair.

6) **Attorney**

a) Is Continent-specific.

b) If a claimant wants to appeal further legally for a closed/rejected claim, he/she can do so; Our company's attorney's will represent the case as our representative.

7) **Auditor**

a) Responsible for the company's annual audit process

b) Mainly for performance maintenance purposes.

III ER Model and Tables

Our auto insurance claim application uses multiple tables to store different types of data. In this section, we provide an overview of the key dynamic and static tables used in the process (see Table I for a summary of the main tables). Figure 2 presents the ER diagram for the database structure.

III-A Static Tables

- **coverage_cd**: Lists coverage and sub-coverage types for auto policies.
- **payment_type_cd**: Specifies the type of payments that can be made for auto claims.
- **injury_cd**: Lists possible injuries related auto claims.
- **vendor_cd**: Stores information on vendors providing services like towing/labor, salvage, and glass repair.
- **manager_cd:** List our company's managers' details.
- **adjuster_cd:** List our company's adjusters' details.

III-B Dynamic Tables

- **policy_holder**: Holds policyholder information including personal details.
- **auto_policy**: Stores data about the auto insurance policies owned by policyholders, including coverage, premium, and status.
- **claim**: Manages information on filed claims such as type, amount, and status.
- **payment**: Contains details of payments made for settled/ongoing claims.
- **injury**: Logs injuries related to specific claims, with injury codes, severity, and medical expenses.

III-C Views

- **adjuster_claim_vw**: View implemented especially for adjusters containing all relevant columns regarding the claims.
- **attorney_claim_vw**: View containing all the required data if his/her assistance is required for a particular legal case.
- **manager_claim_vw**: View for the manager containing the claim, policy and payment details to process and oversee them
- **vendor_claim_vw**: View created for vendors containing claim incident details which is available for viewing; entails with an additional **trigger** to restrict updation of claim details using this view.

TABLE I: Available tables in our application - Overview

Table Name	**Description**	**Primary Keys**
policy_holder	Stores customer information	policy_holder_id
auto_policy	Stores auto insurance policy details	policy_id
claim	Stores claims filed by policyholders	claim_id
payment	Contains payment details for claims	payment_id, claim_id
injury	Logs injuries related to specific claims	claim_id, injury_code
coverage_cd	Defines coverage types for policies	coverage_code, sub_coverage_code
payment_type_cd	Specifies types of payments	payment_type_code
injury_cd	Lists possible injuries related to claims	injury_code
vendor_cd	Stores information on vendors providing services	vendor_id
adjuster_cd	List our company's adjusters details	adjuster_id
manager_cd	List our company's managers details	manager_id

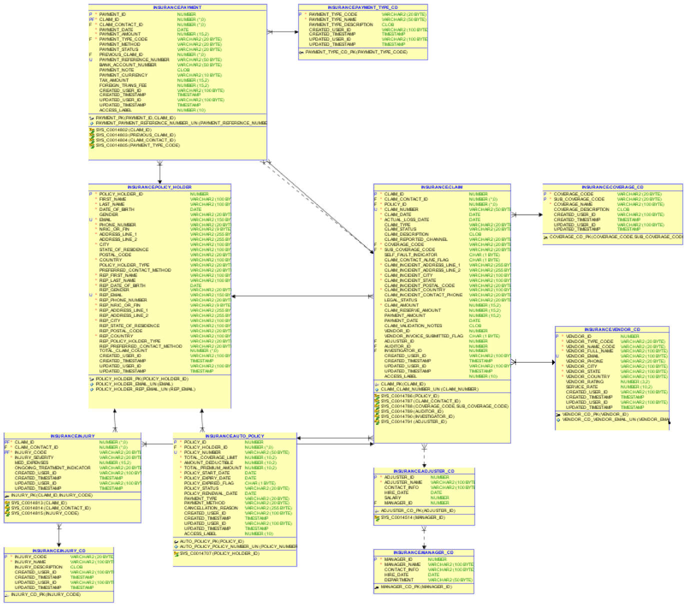


Fig. 2: ER Diagram for the Insurance Application

IV Oracle Label Security Overview

Oracle Label Security (OLS) is an advanced security tool in Oracle Database that enables highly detailed control over data access at the row level, utilizing security labels. OLS allows organizations to define and enforce data classification policies that label data by sensitivity, compartment, and group. This system helps organizations safeguard confidential or sensitive information and manage access by data classification, ensuring that users only see or modify data they're authorized to access.

Advantages of Oracle Label Security:

• **Improved Data Protection**: With row-level security, OLS safeguards sensitive data, preventing unauthorized access and supporting regulatory compliance, such as GDPR and HIPAA.

• **Detailed Access Control**: OLS offers precise user access management. Users can view or change only the rows that align with their clearance, giving organizations the ability to manage access down to individual data rows.

• **Support for Role-Based Access**: OLS facilitates role-based access control, allowing some users to have read-only permissions while others may have full access, helping enforce separation of duties.

• **Seamless Integration**: OLS integrates smoothly with Oracle Database applications, making it easy for administrators to configure data labels and access controls within existing database structures.

Funtioning of Oracle Label Security:

1. **Label Assignment**:

- Data labels are assigned to rows based on their sensitivity. Administrators define these labels within a policy and apply them to rows, indicating the security classification of each piece of data.
- For instance, customer records in an insurance database might be labeled Confidential to restrict access to only authorized staff.

2. **Access Control and Label Comparison**:

- When a user tries to access labeled data, OLS checks the user's clearance against the row's label. If the user's clearance level, group, and compartment match or exceed the label, access is granted; otherwise, it's denied.
- This mechanism enables organizations to enforce strict access rules, reducing the risk of unauthorized data exposure.

3. **Policy Implementation**:

- OLS policies are applied at the table level. Once a policy is in place, each row in the table is secured according to its label, and Oracle enforces access controls based on the policy and user privileges.
- Policies can be modified as security requirements evolve, making OLS adaptable to various data security needs.

V Application design (adhering to OLS)

Our insurance application demanded OLS for the purpose of implementing MAC to control the data access according to users and labels. With OLS policy being applied to the schema's table(s), it proved effortless to add more user accesses, on top of Fine-Grained Access control implementation with Oracle VPD, if requirements arise. Though collective implementation of Oracle VPD and OLS is possible, we have designed our application to showcase the implementation of OLS alone. We will look at the levels, compartments and groups defined for our application and proceed to look at the data and user mapping for the corresponding labels

V-A OLS Levels

As OLS levels are considered data sensitivity hierarchial, we have designed our application with three levels, which apply for both user and table data labels.

- **L1 (Level 1):** Level 1 indicates the least sensitive data level.
- **L2 (Level 2):** Level 2 indicates an intermediate level of security for our application data.
- **L3 (Level 3):** Level 3 indicates the highest security level in our application's OLS configuration.

V-B OLS Compartments

We have designed 2 compartments for our application which are based on the focus of operations during the entire claim process.

- **OT (Operational Team)**: This compartment comprises of all the internal and external operational team of our claim application, such as vendors etc.,
- **LG (Legal Team)**: This compartment consists of attorneys who represent the company legally.

V-C OLS Groups

OLS Groups for our application are based on claim locations globally in the demographic regions, in which we serve clients. Moreover, the groups are defined hierarchically in our OLS policy configuration.

Super parent Group:

- **Global(GL)**

Parent Groups:

- **AS (Asia):** Contains claims where the loss incident country is India and/or Singapore.
- **EU (Europe):** Contains claims where the loss incident country is UK and/or France.
- **NA (North America):** Contains claims where the loss incident country is US and/or Canada.

Child Groups:

- **IN:** Contains claims where the loss incident country is India.
- **SG:** Contains claims where the loss incident country is Singapore.
- **UK:** Contains claims where the loss incident country is United Kingdom.
- **FR:** Contains claims where the loss incident country is France.
- **US:** Contains claims where the loss incident country is United States.
- **CN:** Contains claims where the loss incident country is Canada.

V-D Label mapping for data and users:

The users are labeled based on their roles. Currently for our application, we consider the following roles:

- ADJUSTER
- MANAGER
- VENDOR
- ATTORNEY

We will see in details on the user labels to be mapped for each role:

V-D1 VENDOR:

- **Level: L1, Compartment: OT.**
- Vendor has been designed to have access to less sensitive data across claim table, and hence level assigned for the user label is **L1**, for which the claim status is **’Pending’.**
- Compartment is **OT**, as vendors belong to the operational segment of claim process.
- Groups are assigned based on the vendor’s operating continent(one vendor company for each continent for each service).
- Since our application has 3 vendor services, and in total 3 continents in total, total of 9 vendors are under our application management and are assigned the VENDOR role.

V-D2 ATTORNEY:

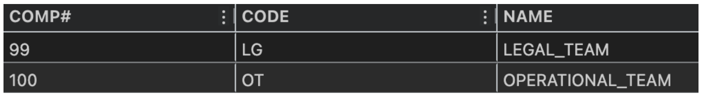
- **Level: L1, Compartment: LG.**
- Attorneys also are designed to access less sensitive data across 2 tables(claim and auto_policy), and hence the level assigned is **L1**, for which the claim status is **’InReview’**.
- Compartment is **LG**, as attorneys belong to the legal side of the claim process.
- Groups are assigned based on the attorney’s positioned continent(one attorney for each continent).
- Total of 3 attorneys serve our company’s claims department.

V-D3 ADJUSTER:

- **Level: L2, Compartments: OT, LG.**
- Adjusters, as they are responsible for the assigned claim process, they possess access to the claim data with intermediate level of security.
- Level is given as L2 and compartment as OT, LG for adjusters.
- Level is given on the basis of the claim status to be **’Open’ or ’Validating’**.
- Compartment is assigned both OT and LG because the adjuster has to see the claim which is ’**Pending**’ with the vendor and **’InReview’** with the attorney.
- Groups are assigned based on the adjuster’s operating country given that one adjuster takes care of the whole country’s claim data.
- Total of 6 attorneys are employed in our country.

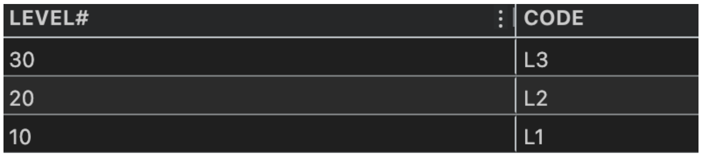
V-D4 MANAGER:

- **Level: L3, Compartments: OT, LG.**
- Managers get the access to claim, policy and payment data, which are of highest security level, as they oversee the entire claim process.
- Level is given as L3 and compartment as OT, LG.
- Level is given on the basis of the claim status to be **’Closed’ or ’Rejected’**. Adjusters cant see the closed/rejected claims as they focus on current operations only.
- Compartment is assigned both OT and LG because the manager has to see the claim which is ’**Pending**’ with the vendor and **’InReview’** with the attorney, **’Open’** or **’Validating’** with the adjuster.



COMP#	CODE	NAME
99	LG	LEGAL_TEAM
100	OT	OPERATIONAL_TEAM

Fig. 3: Levels available for our OLS implementation



LEVEL#	CODE
30	L3
20	L2
10	L1

Fig. 4: Compartments available for our OLS implementation

VI Implementation and verification results

VI-A Policy implementation:

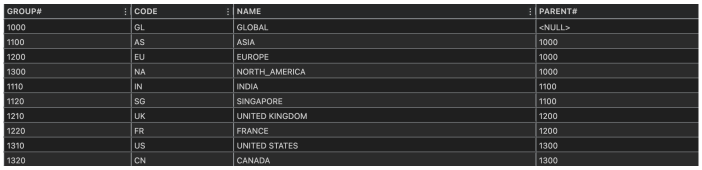
We have implemented the below OLS policy for the tables **claim, auto_policy, payment** in the schema **INSURANCE.**

```
BEGIN
  SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    policy_name =>
                ’INSURANCE_CLAIM_OLS_POL’,
    schema_name => ’INSURANCE’,
    table_name  => ’CLAIM’,
    table_options => ’READ_CONTROL,
            WRITE_CONTROL, CHECK_CONTROL’);
END;
/

BEGIN
  SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    policy_name =>
                ’INSURANCE_CLAIM_OLS_POL’,
    schema_name => ’INSURANCE’,
    table_name  => ’AUTO_POLICY’,
    table_options => ’READ_CONTROL,
            WRITE_CONTROL, CHECK_CONTROL’);
END;
/

BEGIN
  SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    policy_name =>
                ’INSURANCE_CLAIM_OLS_POL’,
    schema_name => ’INSURANCE’,
    table_name  => ’PAYMENT’,
    table_options => ’READ_CONTROL,
            WRITE_CONTROL, CHECK_CONTROL’);
END;
/
```

We are defining the same policies for all the tables as our implementation of label access is the same with more depth ensuring cross table confidentiality and synchronization.

VI-B Available labels and grant permissions:

Below are the available levels, compartments and groups defined for the OLS policy. **Fig. 3, 4 and 5 show the available levels, compartments and groups for implementation**



GROUP#	CODE	NAME	PARENT#
1000	GL	GLOBAL	<NULL>
1100	AS	ASIA	1000
1200	EU	EUROPE	1000
1300	NA	NORTH_AMERICA	1000
1110	IN	INDIA	1100
1120	SG	SINGAPORE	1100
1210	UK	UNITED KINGDOM	1200
1220	FR	FRANCE	1200
1310	US	UNITED STATES	1300
1320	CN	CANADA	1300

Fig. 5: Groups available for our OLS implementation

The users are created for the roles ADJUSTER, MANAGER, VENDOR and ATTORNEY and the roles are given required grants on the tables.

1) **ADJUSTER :** SELECT, UPDATE access for claim, auto_policy tables
2) **MANAGER :** SELECT, UPDATE access for claim, auto_policy, payment tables
3) **VENDOR :** SELECT, UPDATE access for claim table, with a trigger for restricted column update.
4) **ATTORNEY :** SELECT access for claim, auto_policy tables.

VI-C Views implementation:

As we inferred in the previous subsections, all 4 roles have select access on claim table, combined with requirement specific tables. The select at the role user's end is simplified by the use of views as each select has some restrictions and validations to be done:

adjuster_claim_vw: For adjusters, has validation for valid adjuster check against the static adjuster_cd table.

attorney_claim_vw: For attorneys, joining only the necessary columns of claim and auto_policy table for legal purpose.

manager_claim_vw: For managers, has validation for valid manager check against the static manager_cd table and displaying only the necessary columns of the joined claim, auto_policy and payment tables.

vendor_claim_vw: For vendors, has validation for valid vendor check against the static vendor_cd table.

VI-D Trigger implementation:

Two types of triggers are implemented for this OLS policy contained INSURANCE schema.

1. claim_trigger : it is a table trigger, used to update data label of each row, during any modification query on that particular row. The data label set can be performed using label function also, but the reason for choosing trigger is its rapid responsiveness and ensured correctness. (Refer IX-A APPENDIX for code).

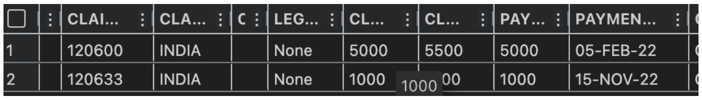
2. vendor_update_restriction_trigger: According to the business flow of vendor, he/she should be able to update only the documents upload flag(vendor_invoice_submitted_flag), but can see necessary rows. To facilitate such restriction, INSTEAD OF functionality of trigger is implemented.

VI-E Testing Scenarios:

VI-E1 Selection scenarios:

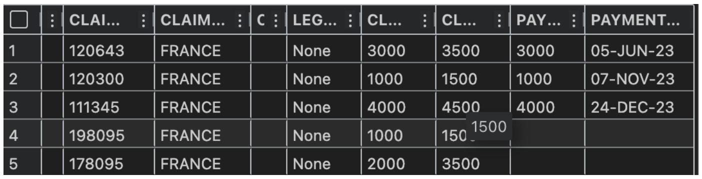
We will look at test select scenarios which prove the implementation of OLS policy and labels.

Adjuster:



□	⋮	CLAI...	CLA...	C	LEG...	CL...	CL...	PAY...	PAYMEN...
1		120600	INDIA		None	5000	5500	5000	05-FEB-22
2		120633	INDIA		None	1000	1000	1000	15-NOV-22

Fig. 6: ADJUSTER_IN select result



□	⋮	CLAI...	CLAIM...	C	LEG...	CL...	CL...	PAY...	PAYMENT...
1		120643	FRANCE		None	3000	3500	3000	05-JUN-23
2		120300	FRANCE		None	1000	1500	1000	07-NOV-23
3		111345	FRANCE		None	4000	4500	4000	24-DEC-23
4		198095	FRANCE		None	1000	1500		
5		178095	FRANCE		None	2000	3500		

Fig. 7: ADJUSTER_FR select result

- Lets take an adjuster ADJUSTER_IN, who deals with all the claims in the country - India, with the claim status of 'Open', 'Pending', 'Validating' and 'InReview'(as per our business rules mentioned before).
- When we query the adjuster view, which is common for all the adjusters under the role, we get only the claims from India and with the mentioned statuses.
- For the sake of visibility, we have modified the view to show only the required columns for testing.

```
SELECT * FROM INSURANCE.ADJUSTER_CLAIM_VW;
```

Fig. 6 gives the query result showing only the INDIA claims with the needed statuses.

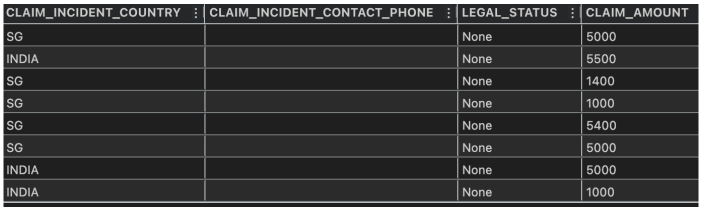
Lets consider ADJUSTER_FR, Fig. 7 shows the result intended validating the OLS policy implementation.

Manager:

- Lets take a Manager MANAGER_ASIA, who deals with all the claims in the countries - India and Singapore, with all claim statuses(as per our business rules mentioned before).
- When we query the manager view, which is common for all the managers under the role, we get only the claims from India, SG as they are under continent group Asia and with the mentioned statuses.
- For the sake of visibility, we have modified the view to show only the required columns for testing.
- The additional takeaway is we have used join to display the auto_policy and payment table data(which are needed) as they are also enforced with the same OLS policy.

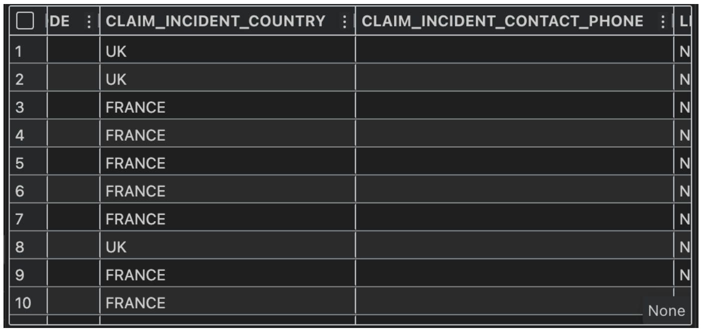
```
SELECT * FROM INSURANCE.MANAGER_CLAIM_VW;
```

Fig. 8 gives the query result showing only the ASIA claims with the needed statuses.



CLAIM_INCIDENT_COUNTRY	CLAIM_INCIDENT_CONTACT_PHONE	LEGAL_STATUS	CLAIM_AMOUNT
SG		None	5000
INDIA		None	5500
SG		None	1400
SG		None	1000
SG		None	5400
SG		None	5000
INDIA		None	5000
INDIA		None	1000

Fig. 8: MANAGER_ASIA select result



□	DE	CLAIM_INCIDENT_COUNTRY	CLAIM_INCIDENT_CONTACT_PHONE	LI
1		UK		N
2		UK		N
3		FRANCE		N
4		FRANCE		N
5		FRANCE		N
6		FRANCE		N
7		FRANCE		N
8		UK		N
9		FRANCE		N
10		FRANCE		None

Fig. 9: MANAGER_EUROPE select result

Lets consider MANAGER_EUROPE, Fig. 9 shows the result intended validating the OLS policy implementation.

Vendor:

- Lets take an vendor V_AS_TL, who deals with all the vendor services concerning Towing and labor in the countries - India and Singapore(collectively Asia), with the claim status of 'Pending'(as per our business rules mentioned before).
- When we query the vendor view, which is common for all the vendors under the role, we get only the claims from India, SG as they are under continent group Asia and with the mentioned statuses.
- For the sake of visibility, we have modified the view to show only the required columns for testing.

```
SELECT * FROM INSURANCE.VENDOR_CLAIM_VW;
```

Attorney:

- Lets take an attorney ATTORNEY_ASIA, who deals with all the legal issues in the countries - India and Singapore, with all claim status of 'InReview'(as per our business rules mentioned before).
- When we query the attorney view, which is common for all the attorneys under the role, we get only the claims from India, SG as they are under continent group Asia and with the mentioned statuses.
- For the sake of visibility, we have modified the view to show only the required columns for testing.
- The additional takeaway is we have used join to display the auto_policy table data(which are needed) as they are also enforced with the same OLS policy.

```
SELECT * FROM INSURANCE.ATTORNEY_CLAIM_VW;
```

VI-E2 Modification scenarios

We will look at test select scenarios which prove the implementation of OLS policy and labels.

Adjuster and Manager: For both adjusters and managers, the update flow on OLS implemented claim table, contains 3 scenarios:

1) Can update country of the view permitted rows, and once updated, the access_label changes according to our claim_trigger. If still the access label comes under the view/update of that particular adjuster/manager, he/she can perform operations on it. Otherwise it is restricted.
2) Can update claim_status of the view permitted rows, and once updated, the access_label changes according to our claim_trigger. If still the access label comes under the view/update of that particular adjuster/manager, he/she can perform operations on it. Otherwise it is restricted.
3) Can try to update rows outside the view/ label restriction; but the end result being the user not able to update such rows as they are out of the user's label inclusion.

Attorney: For attorneys, as there is no privilege to update the rows, they cant do so, even if the row label that they intend to fall under the correct label privileges(they can only view it).

Vendor: According to our business rule, any vendor is supposed to only update the vendor_invoice_submitted_flag column for the label restricted rows and this is implemented using the trigger: *vendor_update_restriction_trigger.*

```
CREATE OR REPLACE TRIGGER
VENDOR_UPDATE_RESTRICTION_TRIGGER
INSTEAD OF UPDATE ON
            insurance.VENDOR_CLAIM_VW
FOR EACH ROW
BEGIN
  IF UPDATING
        ('VENDOR_INVOICE_SUBMITTED_FLAG')
    THEN UPDATE insurance.claim
    SET vendor_invoice_submitted_flag =
        :NEW.vendor_invoice_submitted_flag
    WHERE claim_id = :OLD.claim_id;
  ELSE
    RAISE_APPLICATION_ERROR
        (-20001,
        'Only VENDOR_INVOICE_SUBMITTED_FLAG
        can be updated:
        RESTRICTION FOR VENDORS');
  END IF;
END;
```

VII Special privileges in OLS - Implementation

Users with specialized privileges for OLS policy can perform CRUD actions on the tables, without having the user label. They can also intend to have user labels if we want to implement further restrictions for business needs.

VII-A Implemented special privileges:

The privileges available are:

- **READ:** grants the user full read access to the tables under the policy, and is restricted to another operations on those tables.
- **FULL:** grants the user full operational access to all the tables under the policy.

For our application implementation, we have two such users who require the above privileges:

1. CHIEF_MANAGER: Chief manager is responsible for the global overseeing of the claims, and any operations pertaining them. We need not give him/her a user label, but can apply FULL privileges to the chief manager, at a policy level.

```
    BEGIN
  SA_USER_ADMIN.SET_USER_PRIVS(
    user_name     => ’CHIEF_MANAGER’,
    policy_name   =>
            ’INSURANCE_CLAIM_OLS_POL’,
    privileges     => ’FULL’
  );
END;
/
```

2. CHIEF_AUDITOR: Chief auditor represents a team of our company’s auditors who will perform audit on all the claim data available on a quarterly and annual basis. For this purpose, we can apply READ privilege on the chief auditor user, as it is not recommended to assign him/her an user role, just for a periodic operation.

```
    BEGIN
  SA_USER_ADMIN.SET_USER_PRIVS(
    user_name     => ’CHIEF_AUDITOR’,
    policy_name   =>
            ’INSURANCE_CLAIM_OLS_POL’,
    privileges     => ’READ’
  );
END;
/
```

VII-B Other special privileges:

There are WRITE_UP, WRITE_DOWN and WRITE_ACROSS privileges which provide the users with the ability to modify access levels, compartments and groups of the policy enforced data. We have neglected these privileges in our application because we have dynamic label setting via *claim_trigger*. Nevertheless, these privileges can be implemented considering the business requirements of future enhancements.

VIII Conclusion

Oracle Label Security (OLS) and Mandatory Access Control (MAC) provide a powerful and comprehensive approach to securing sensitive data within Oracle databases. By leveraging views to implement access control policies, we have created a flexible and efficient solution that ensures only authorized users can access and manipulate sensitive information. This approach enhances data security, protects against unauthorized access, and maintains compliance with industry regulations. With OLS and MAC, we can confidently safeguard our organization’s critical data assets.

IX APPENDIX

IX-A claim_trigger reference code:

```
CREATE OR REPLACE TRIGGER
claim_trigger BEFORE
    UPDATE OR INSERT ON claim
        FOR EACH ROW
DECLARE
    t_level VARCHAR2(10);
    t_comp  VARCHAR2(10);
    t_group VARCHAR2(10);
BEGIN
    IF :new.claim_status = ’Pending’
    THEN
        t_level := ’L1’;
        t_comp := ’OT’;
    ELSIF :new.claim_status = ’InReview’
    THEN
        t_level := ’L1’;
        t_comp := ’LG’;
    ELSIF :new.claim_status IN (’Open’,
    ’Validating’) THEN
        t_level := ’L2’;
        t_comp := ’OT’;
    ELSE
        t_level := ’L3’;
        t_comp := ’OT’;
    END IF;

    IF :new.claim_incident_country
    IN (’INDIA’, ’IN’) THEN
        t_group := ’IN’;
    ELSIF :new.claim_incident_country
    IN (’FRANCE’, ’FR’) THEN
        t_group := ’FR’;
    ELSIF :new.claim_incident_country
    IN (’CANADA’, ’CN’) THEN
        t_group := ’CN’;
    ELSIF :new.claim_incident_country
    IN (’US’) THEN
        t_group := ’US’;
    ELSIF :new.claim_incident_country
    IN (’UK’) THEN
        t_group := ’UK’;
    ELSIF :new.claim_incident_country
    IN (’SG’) THEN
        t_group := ’SG’;
    ELSE
        t_group := ’GL’;
    END IF;

    :new.access_label := char_to_label
    (’INSURANCE_CLAIM_OLS_POL’, t_level || ’:’
        || t_comp|| ’:’|| t_group);

    UPDATE INSURANCE.AUTO_POLICY
    SET ACCESS_LABEL =
    char_to_label(’INSURANCE_CLAIM_OLS_POL’,
    t_level|| ’:’|| t_comp|| ’:’|| t_group)
    where POLICY_ID = :new.POLICY_ID;

    UPDATE INSURANCE.PAYMENT SET ACCESS_LABEL
    = char_to_label(’INSURANCE_CLAIM_OLS_POL’,
    t_level|| ’:’|| t_comp|| ’:’|| t_group)
    where CLAIM_ID = :new.CLAIM_ID;
END;
```