DATABASE SYSTEM

COMPONENT 2 (70%)

Title: Cars4U vehicle hire company (SECURITY)

**Student Name:** Aastha Niraula

**Student id:** c7227225

**CONTENTS**

Table of Contents

[INTRODUCTION TO THE REPORT 3](#_Toc61351504)

[Task 1: Implementation and Testing 4](#_Toc61351505)

[1.1 Implementation of trigger 1 4](#_Toc61351506)

[Trigger for Customer Auditing (Data INSERT, UPDATE, DELETE) 4](#_Toc61351507)

[1.2. Testing of Trigger 1 6](#_Toc61351508)

[1.3 Additional Trigger for Customer and Insurance table 8](#_Toc61351509)

[Maintaining Data Integrity of Constraints 8](#_Toc61351510)

[Testing of Additional Trigger 9](#_Toc61351511)

[1.4 Implementation of trigger 2 11](#_Toc61351512)

[Trigger for signing contract on Company Holidays 11](#_Toc61351513)

[1.5 Testing of trigger 2 12](#_Toc61351514)

[Task 2. Application Builder Report 13](#_Toc61351515)

[1. SQL Apex Report Number: 1 13](#_Toc61351516)

[2. SQL Apex Report Number: 2 14](#_Toc61351517)

[3. SQL Apex Report Number: 3 15](#_Toc61351518)

[4. SQL Apex Report Number: 4 16](#_Toc61351519)

[Task 3. Application Builder Forms 17](#_Toc61351520)

[Home Page 17](#_Toc61351521)

[1. Master Detail for Customer, Hire and Contract 18](#_Toc61351522)

[Creating Sequence and Trigger 19](#_Toc61351523)

[Sequence and trigger for Customer ID 19](#_Toc61351524)

[Sequence and trigger for Hire Number 20](#_Toc61351525)

[CRUD Functionality 21](#_Toc61351526)

[2. Master Detail for Vehicle, Hire and Invoice 24](#_Toc61351527)

[3. Dashboard for Vehicle, Hire and Invoice 25](#_Toc61351528)

[4. Dashboard for Customer based on Gender 27](#_Toc61351529)

[5. Using APEX Functionality 28](#_Toc61351530)

[a. Using Radio Group 28](#_Toc61351531)

[b. Using Popup LOV and adding SQL command 29](#_Toc61351532)

[6. Required Fields 32](#_Toc61351533)

[7. User login 33](#_Toc61351534)

[Creating Users 33](#_Toc61351535)

[Editing the Actions 34](#_Toc61351536)

[Bibliography 36](#_Toc61351537)

[Appendixes 37](#_Toc61351538)

[TASK1: 37](#_Toc61351539)

[Trigger 1 37](#_Toc61351540)

[Testing 38](#_Toc61351541)

[Additional Trigger 38](#_Toc61351542)

[Testing 38](#_Toc61351543)

[Trigger 2 39](#_Toc61351544)

[Testing 39](#_Toc61351545)

[TASK 2: 39](#_Toc61351546)

[Report1 39](#_Toc61351547)

[Report2 39](#_Toc61351548)

[Report3 40](#_Toc61351549)

[Report4 40](#_Toc61351550)

[SEQUENCE AND TRIGGERS 40](#_Toc61351551)

# INTRODUCTION TO THE REPORT

The report is divided into three tasks, each consists of its own functionality and has its own unique requirements. For this assessment, I chose Option 1 which is “The security of your data by implementing server and client-side solutions”. As per the requirements triggers, reports, master details, dashboards have been implemented.

The first task contains two triggers along with an additional trigger to maintain data integrity in the table after applying the first trigger. The triggers are used in auditing Customer table, maintaining data integrity of constraints and raising an application error in case of Contract table where the condition is not fulfilled for Holidays. The testing codes for each trigger with the evidences is attached along with the trigger implementation.

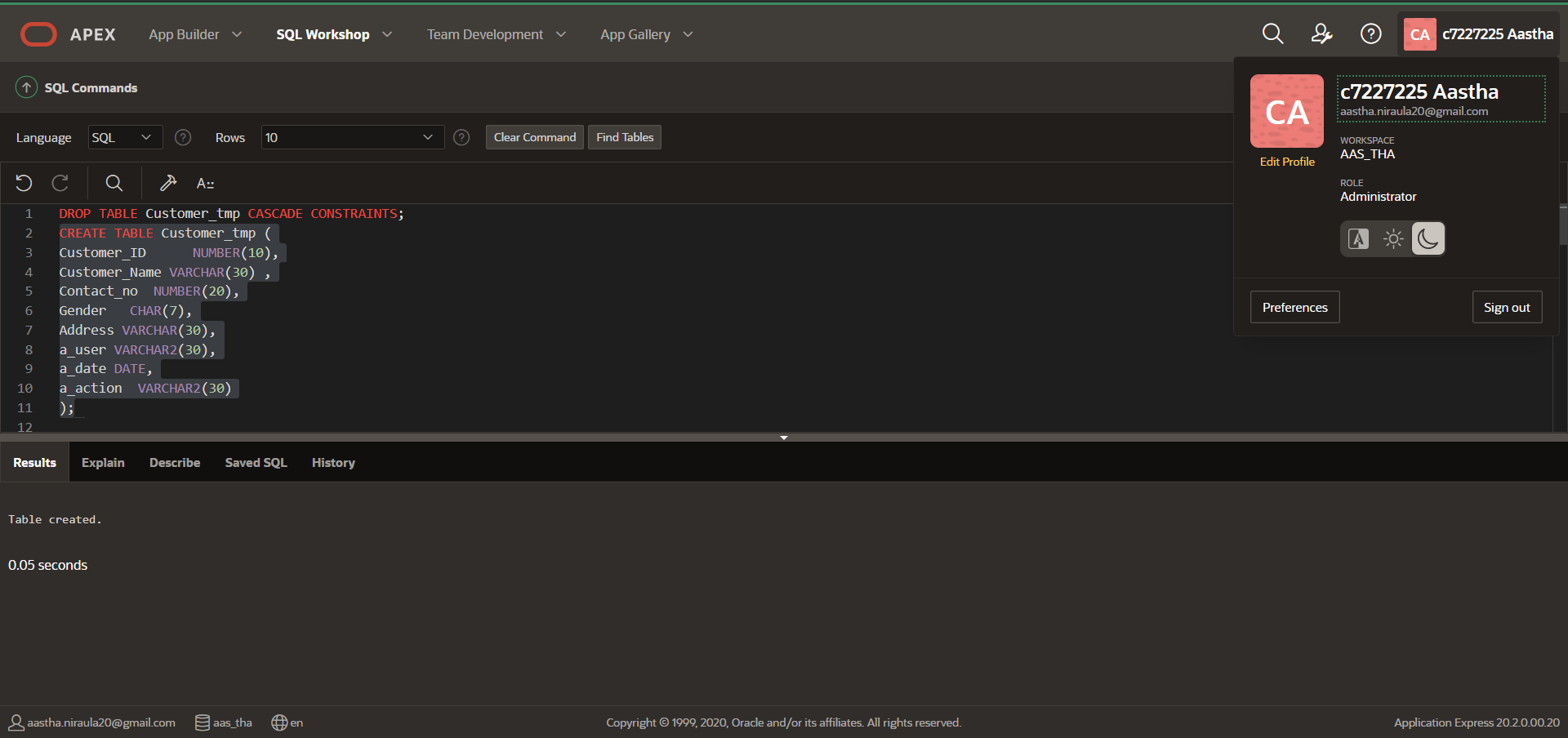
Similarly, task two includes production of an application with the help of Application Builder for producing four SQL reports of advance and intermediate level. The detail function of each report is mentioned along with the SQL codes. Further validation of the work is done by including screenshots of output for each report.

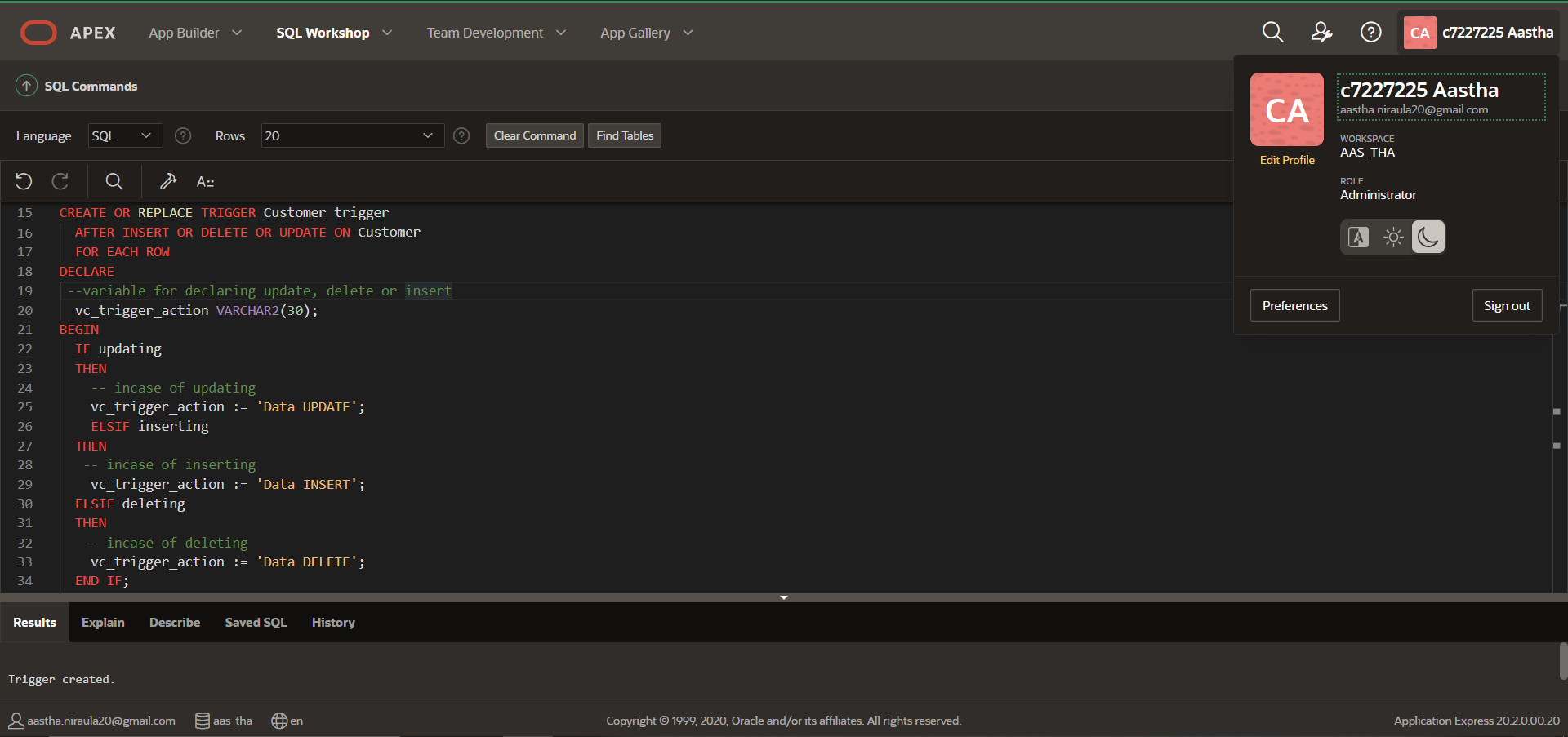
Task three includes a home page for the application, two master details for different tables and CRUD functionality has been implemented. The tables used for first Master Detail are: Customer, Hire, Contract whereas for second: Vehicle, Hire and Invoice are used. Along with these, dashboards have also been created that includes bar, chart, pie chart, etc. In order to automatize the PKs two sequences has also been created along with the triggers. Variations is seen in terms of using APEX Functionality: LOV, Radio Buttons, etc. have been used. The application also includes reports and for maintaining security two users have been created, administrator and contributor where each user has different authentication rule.

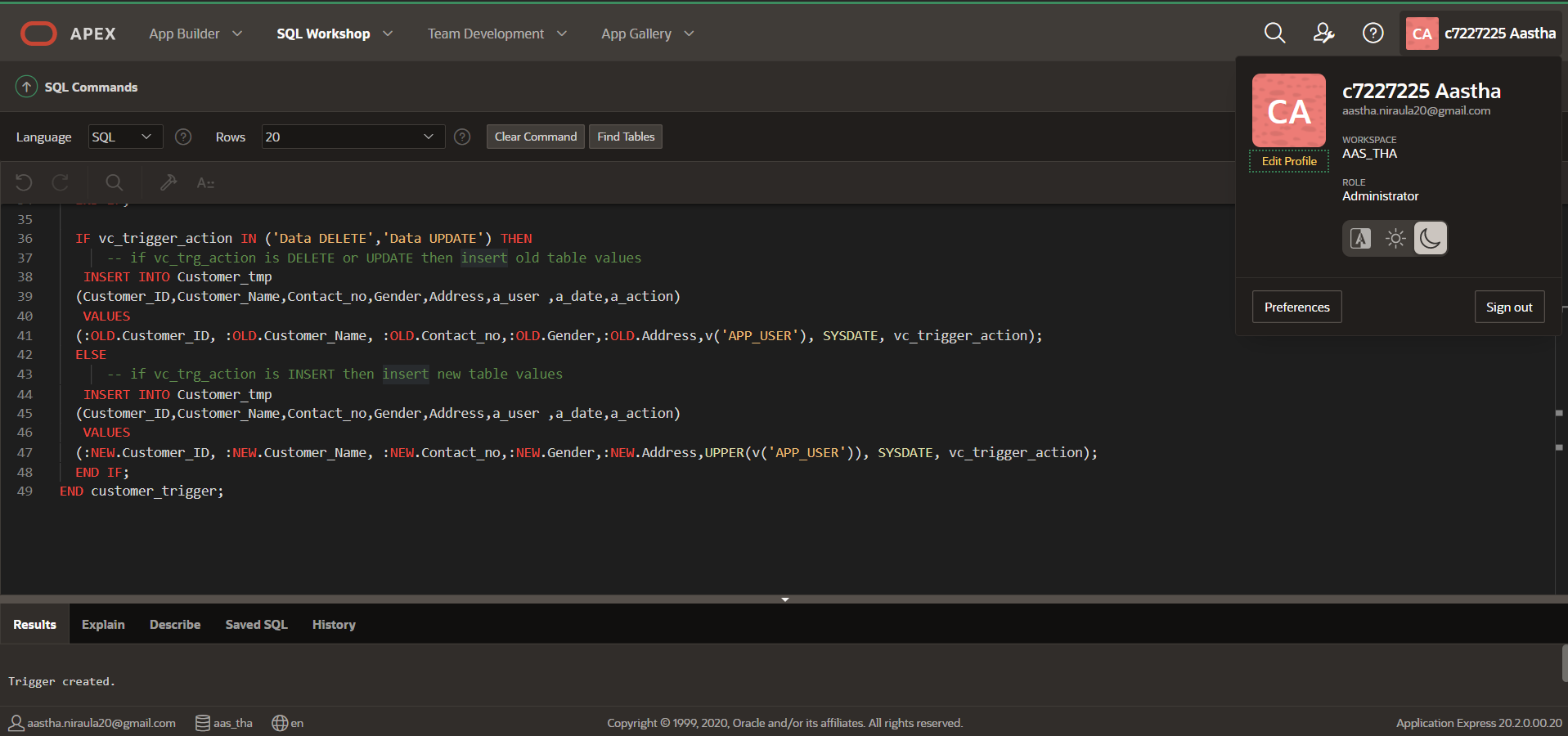
# Task 1: Implementation and Testing

## 1.1 Implementation of trigger 1

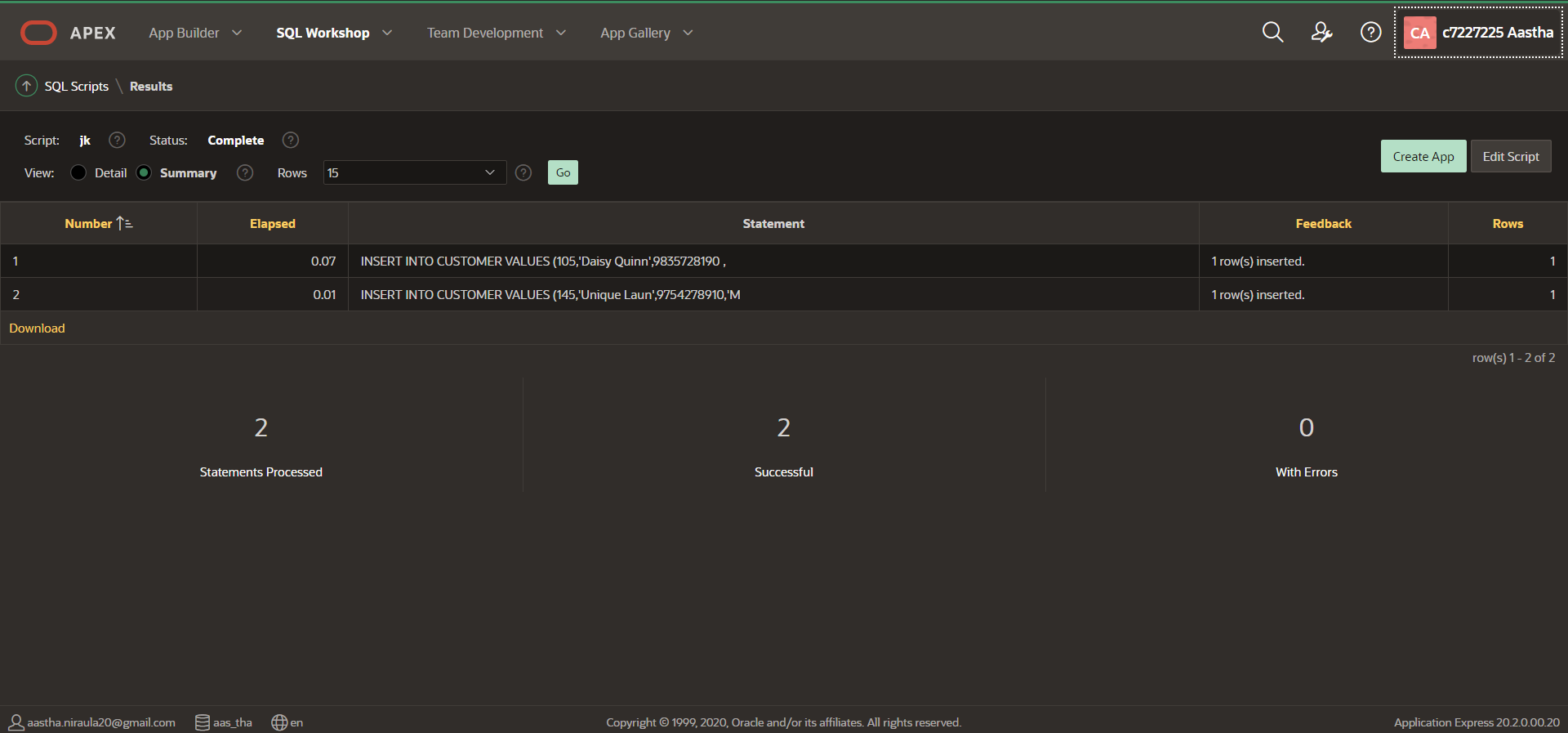
### Trigger for Customer Auditing (Data INSERT, UPDATE, DELETE)

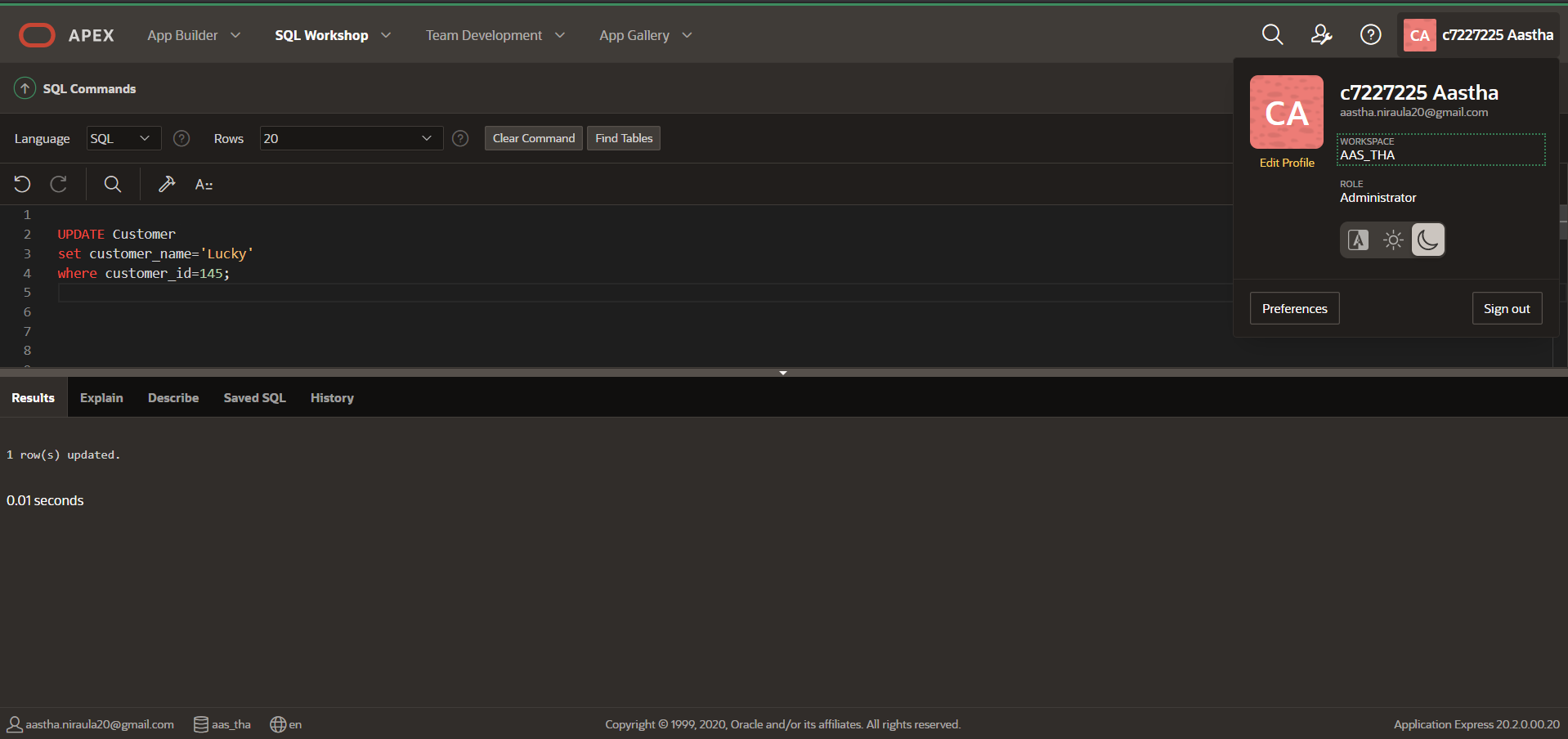


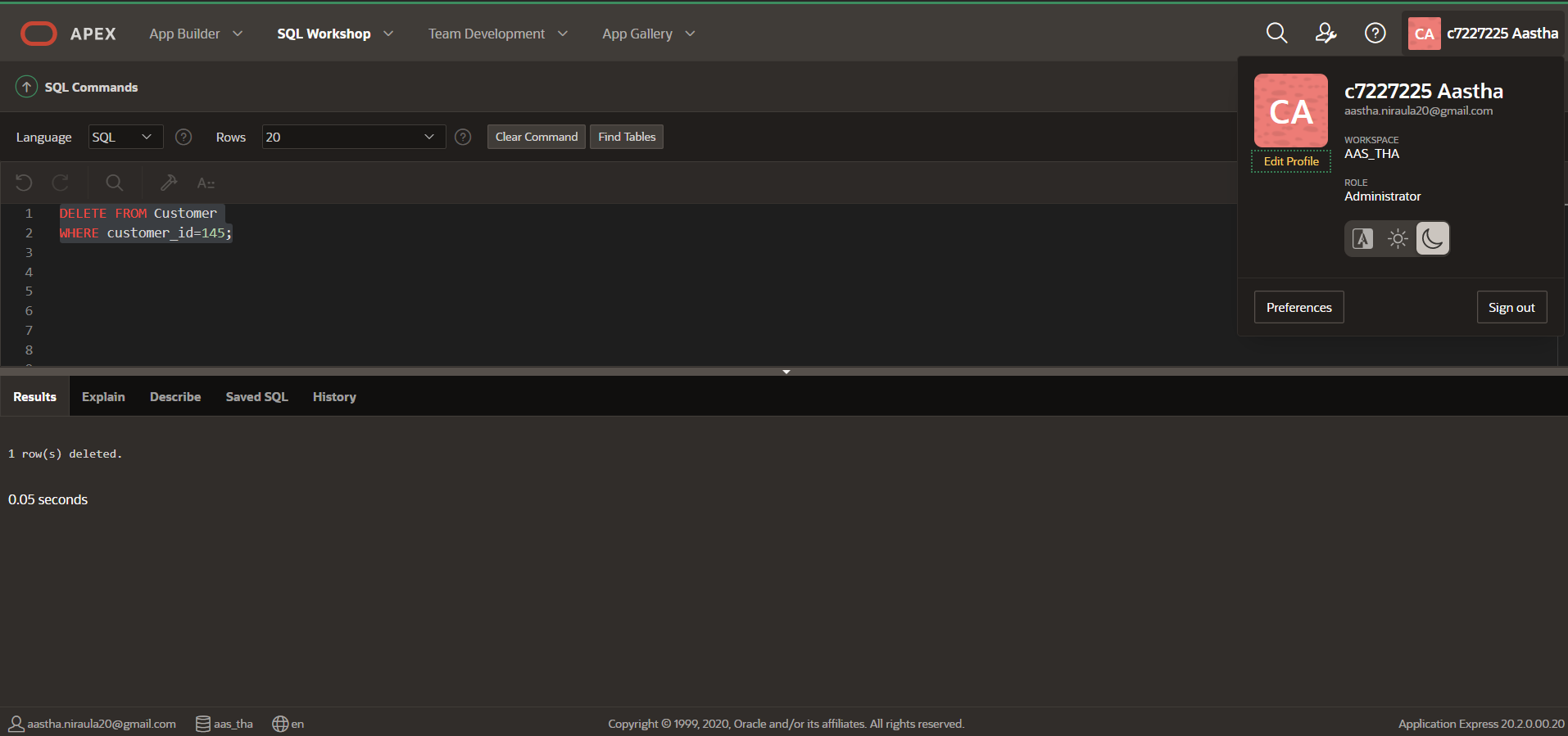


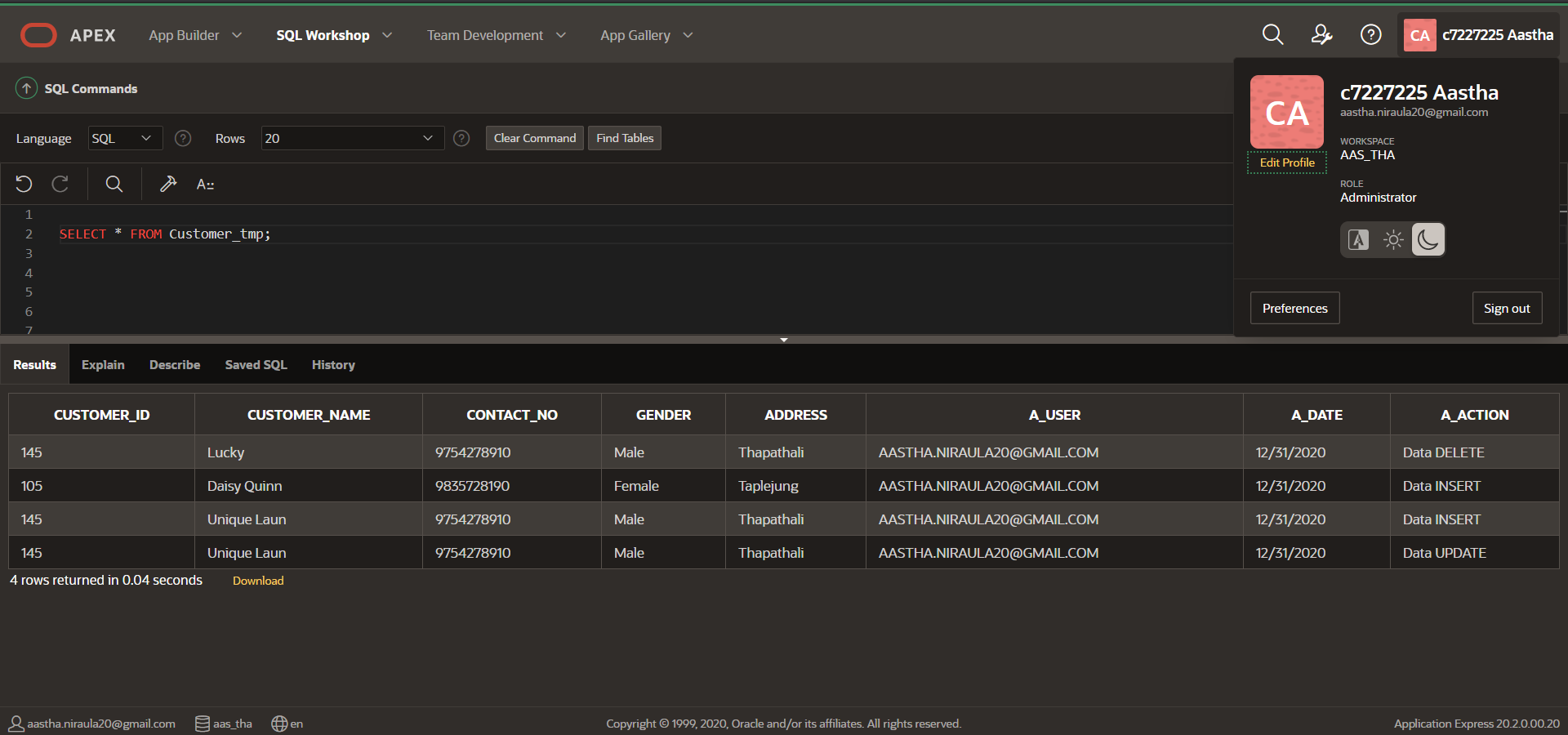


## 1.2. Testing of Trigger 1



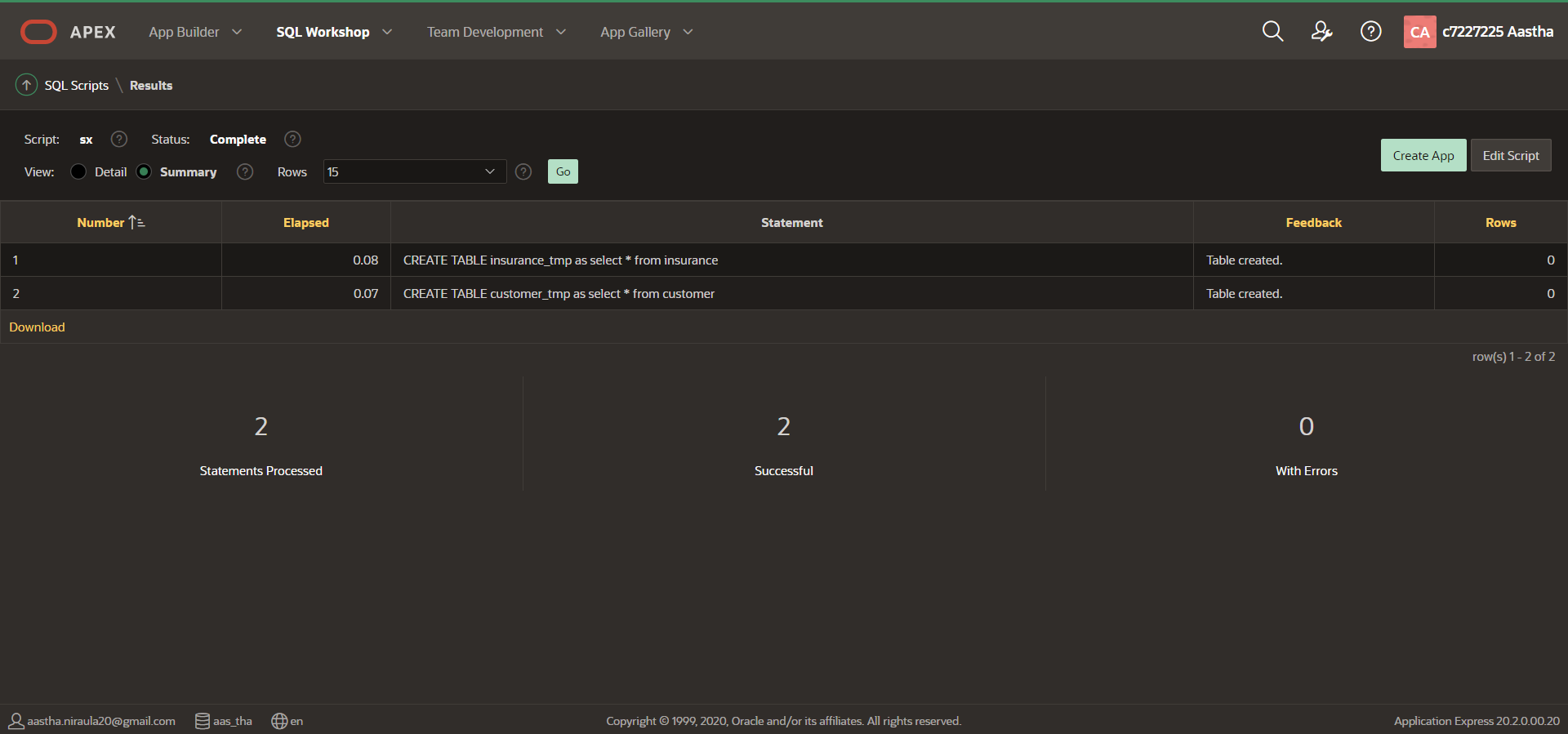


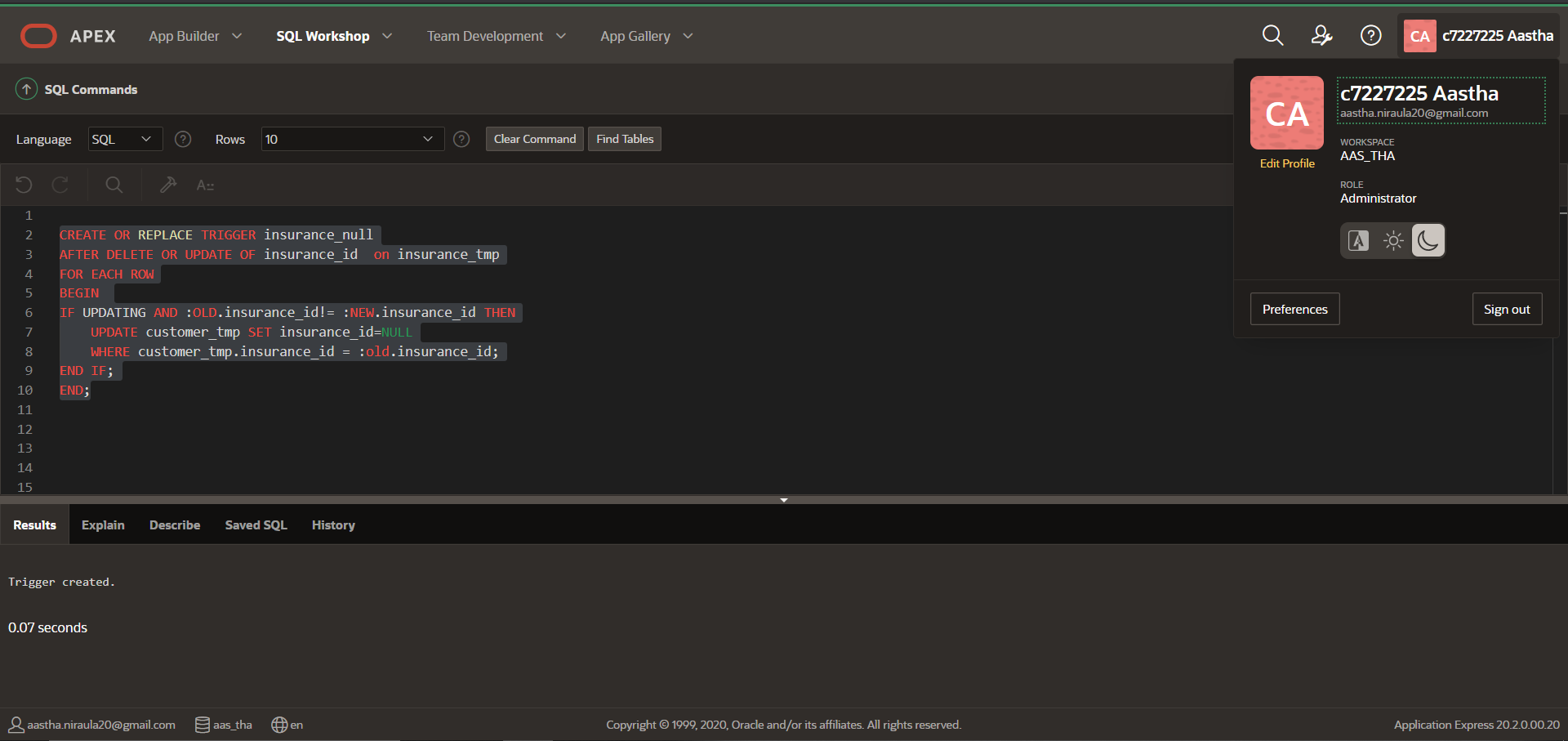




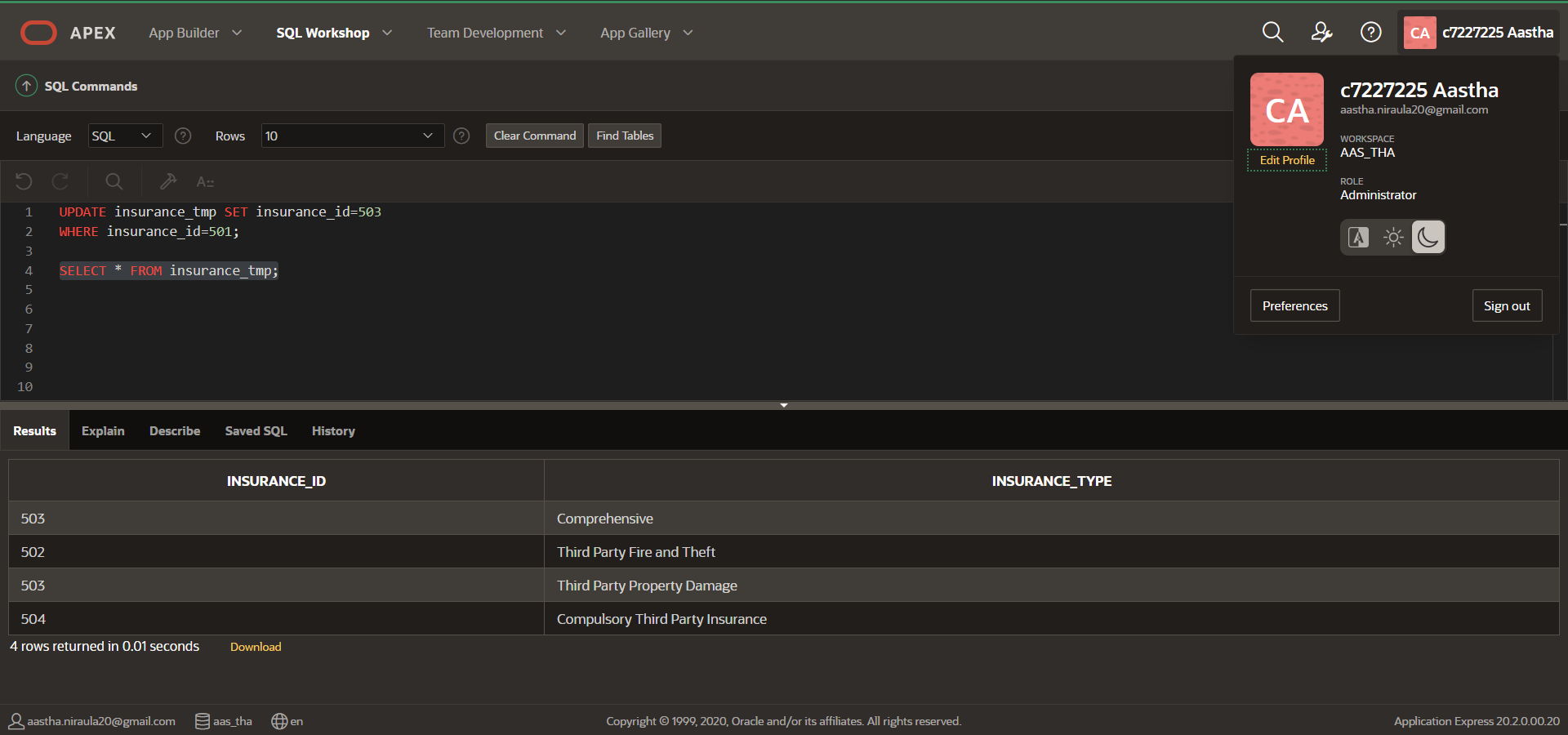
## 1.3 Additional Trigger for Customer and Insurance table

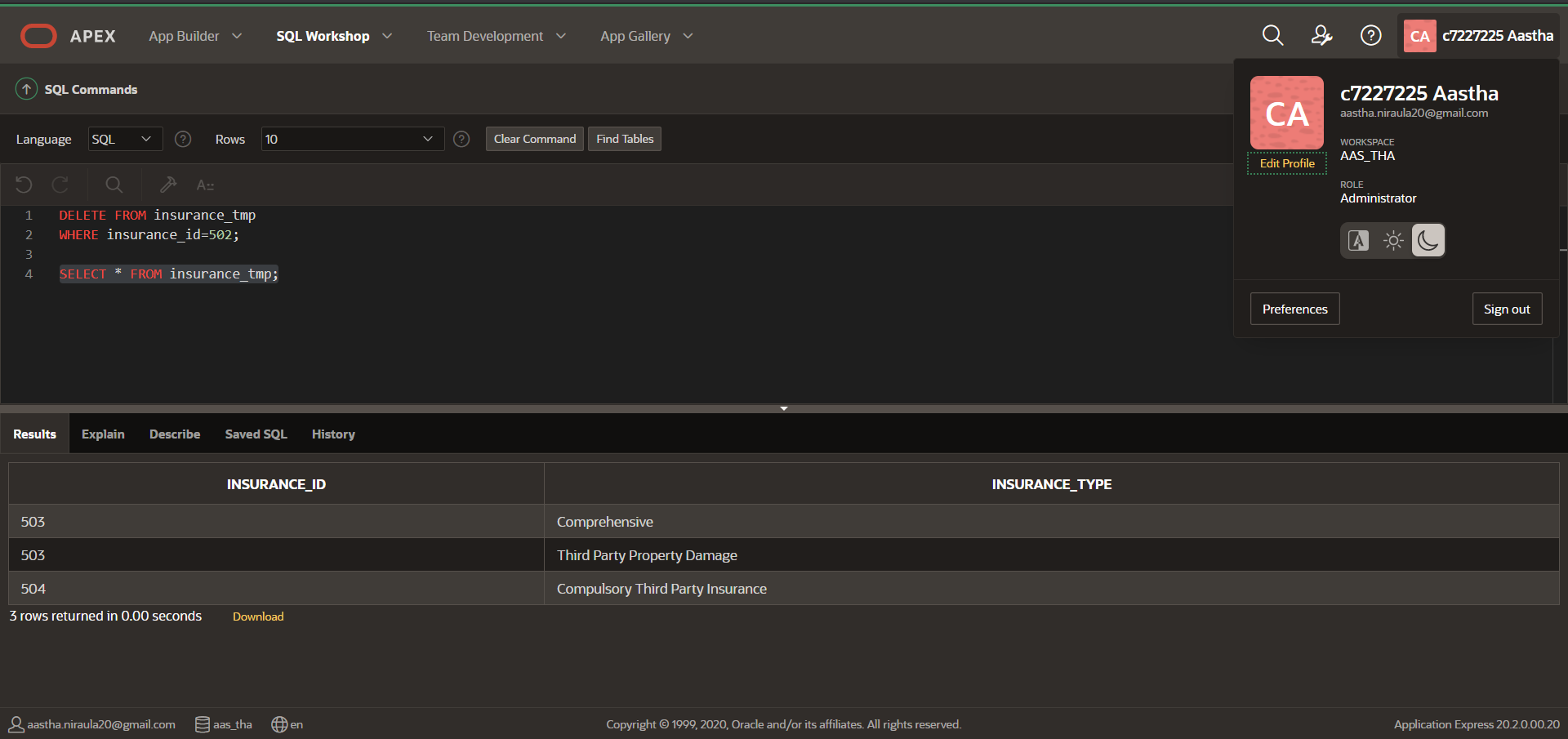
### Maintaining Data Integrity of Constraints

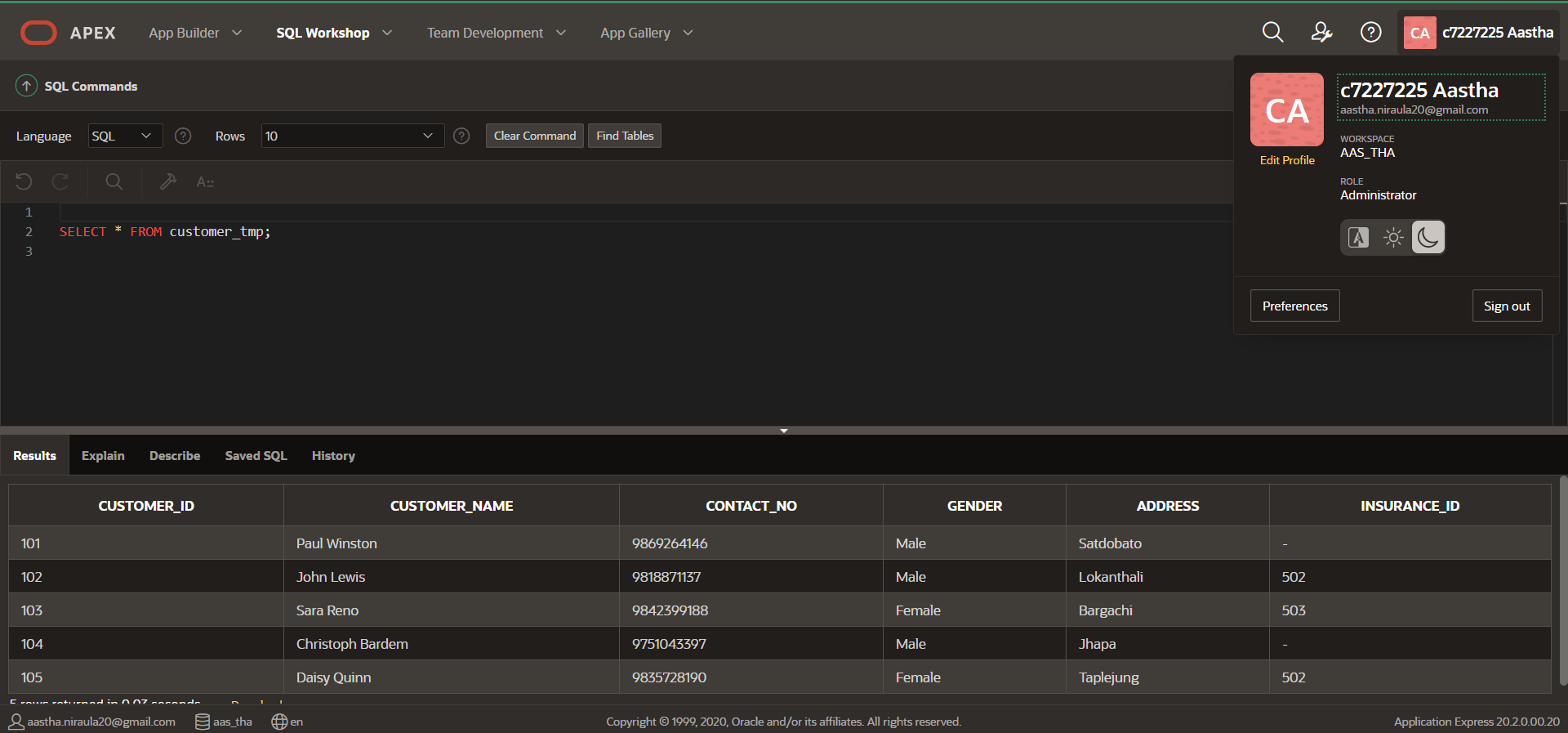




### Testing of Additional Trigger

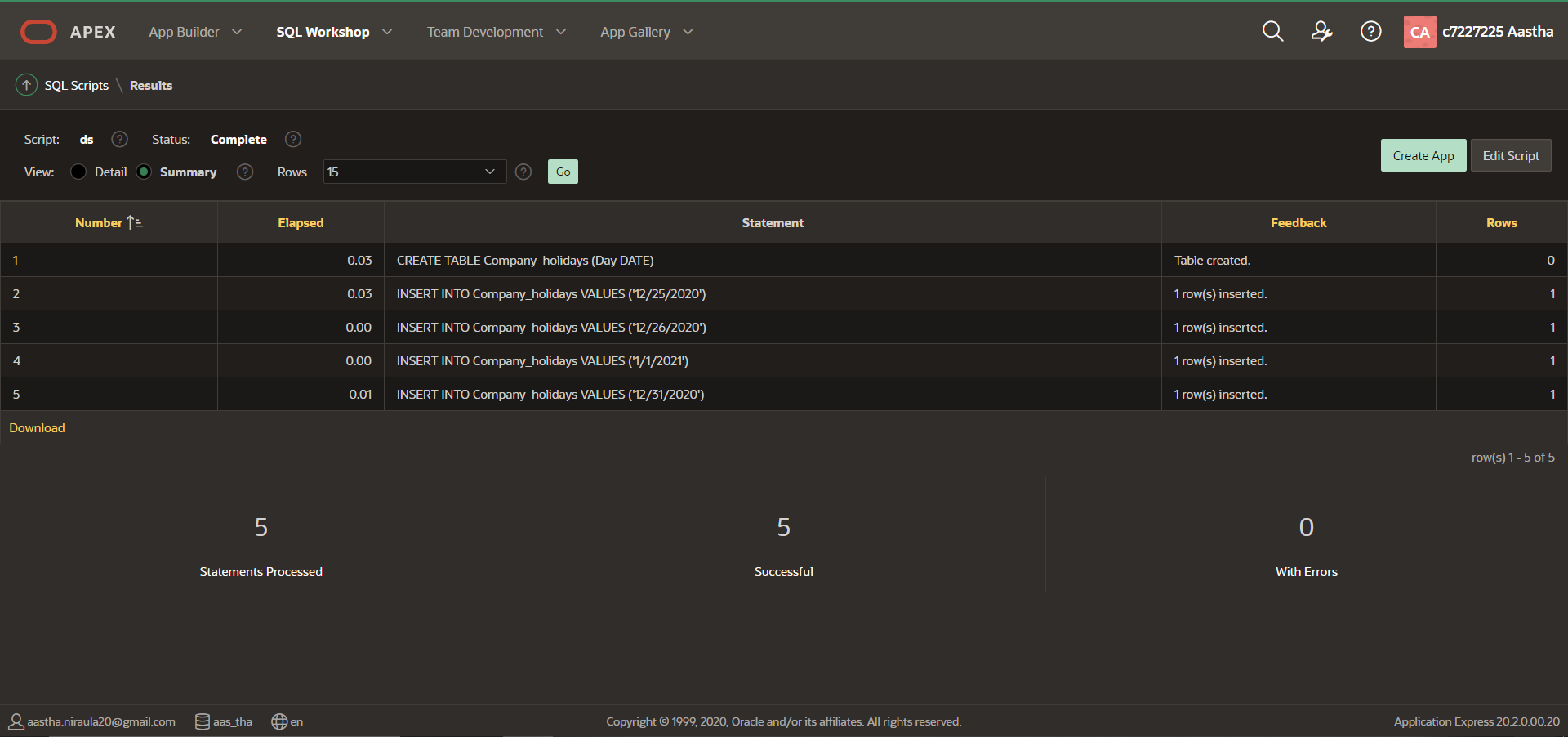


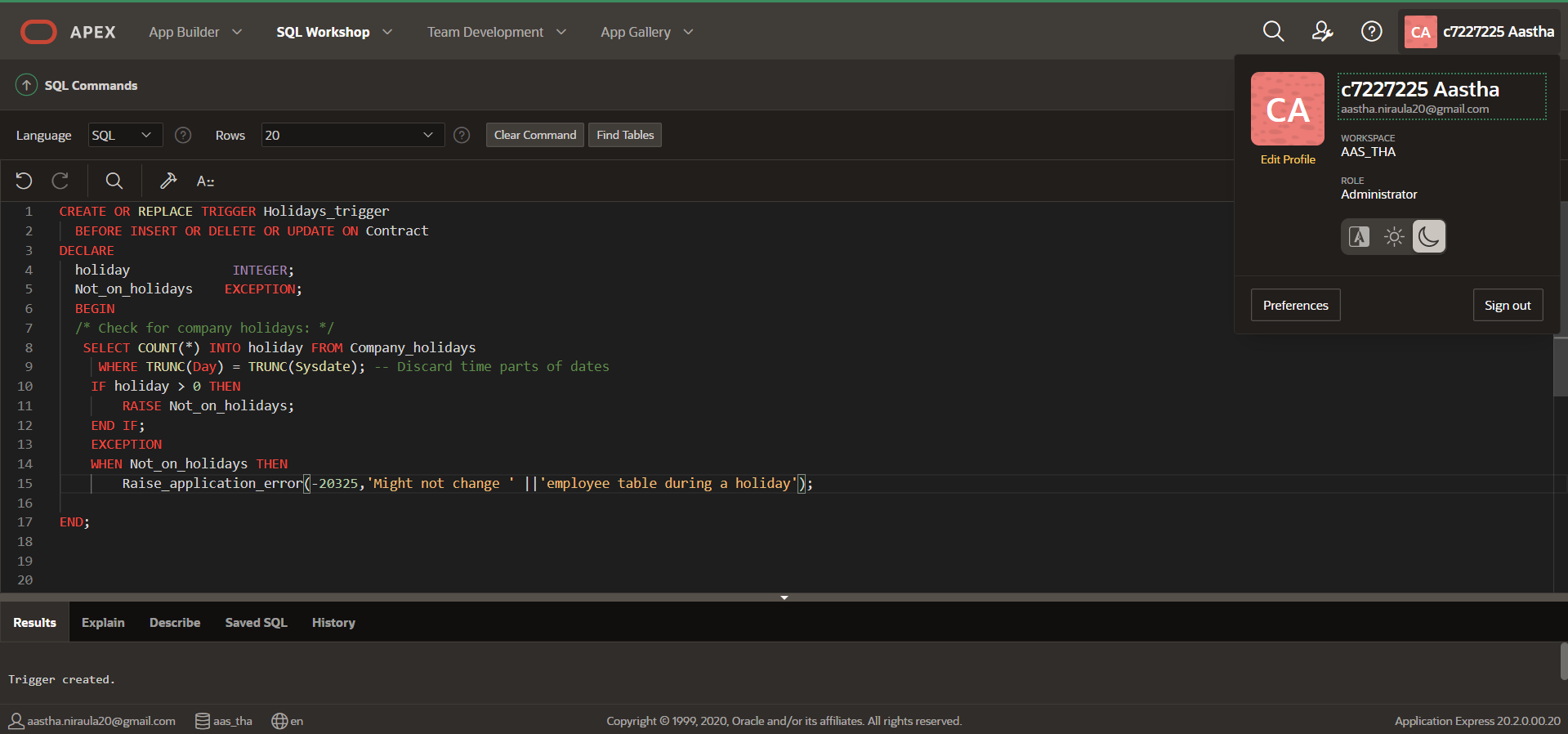




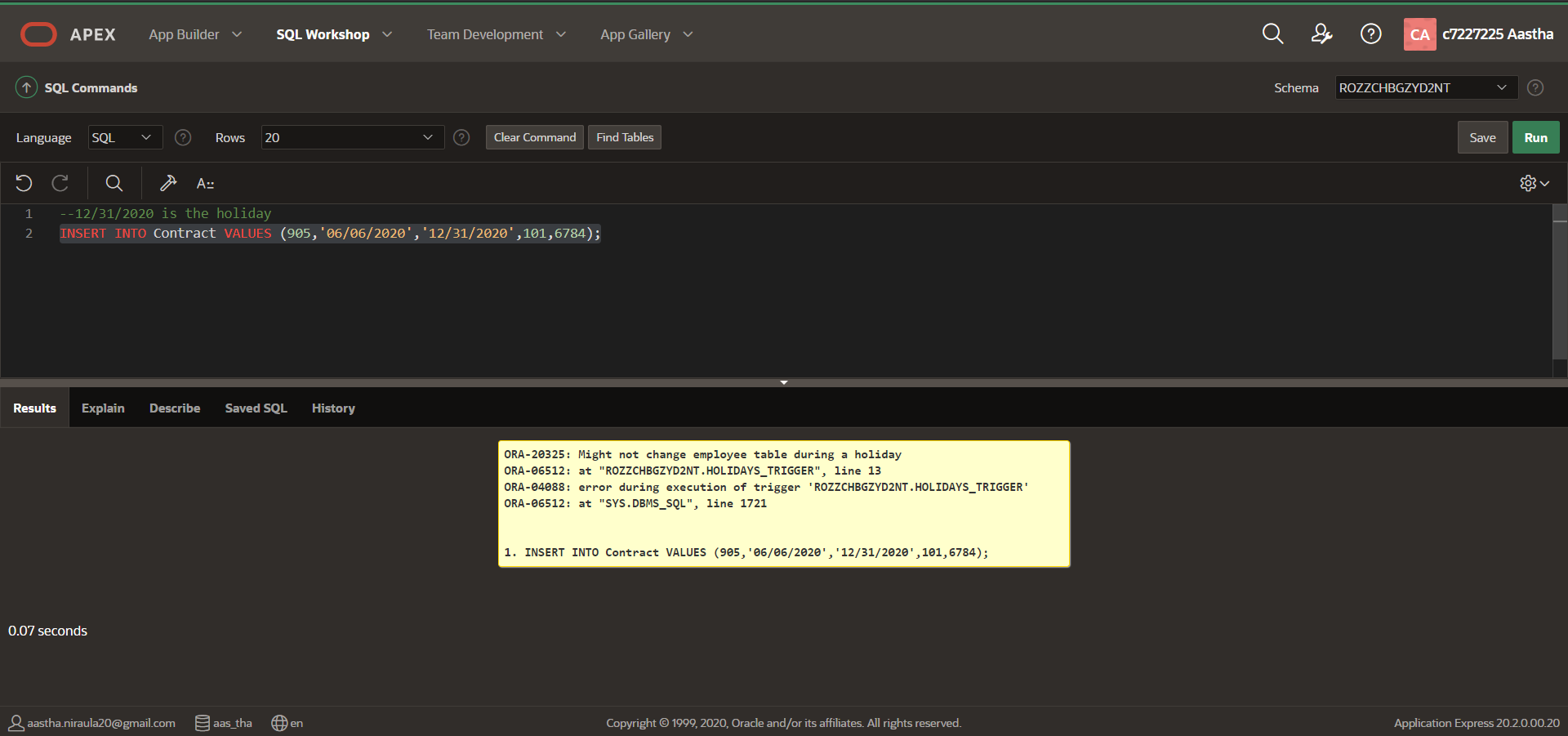
## 1.4 Implementation of trigger 2

### Trigger for signing contract on Company Holidays





## 1.5 Testing of trigger 2



# Task 2. Application Builder Report

## 1. SQL Apex Report Number: 1

The report has included two tables Customer and Hire which takes the details of Customer who has hired a vehicle more than once and the details are ordered based on the hire days.

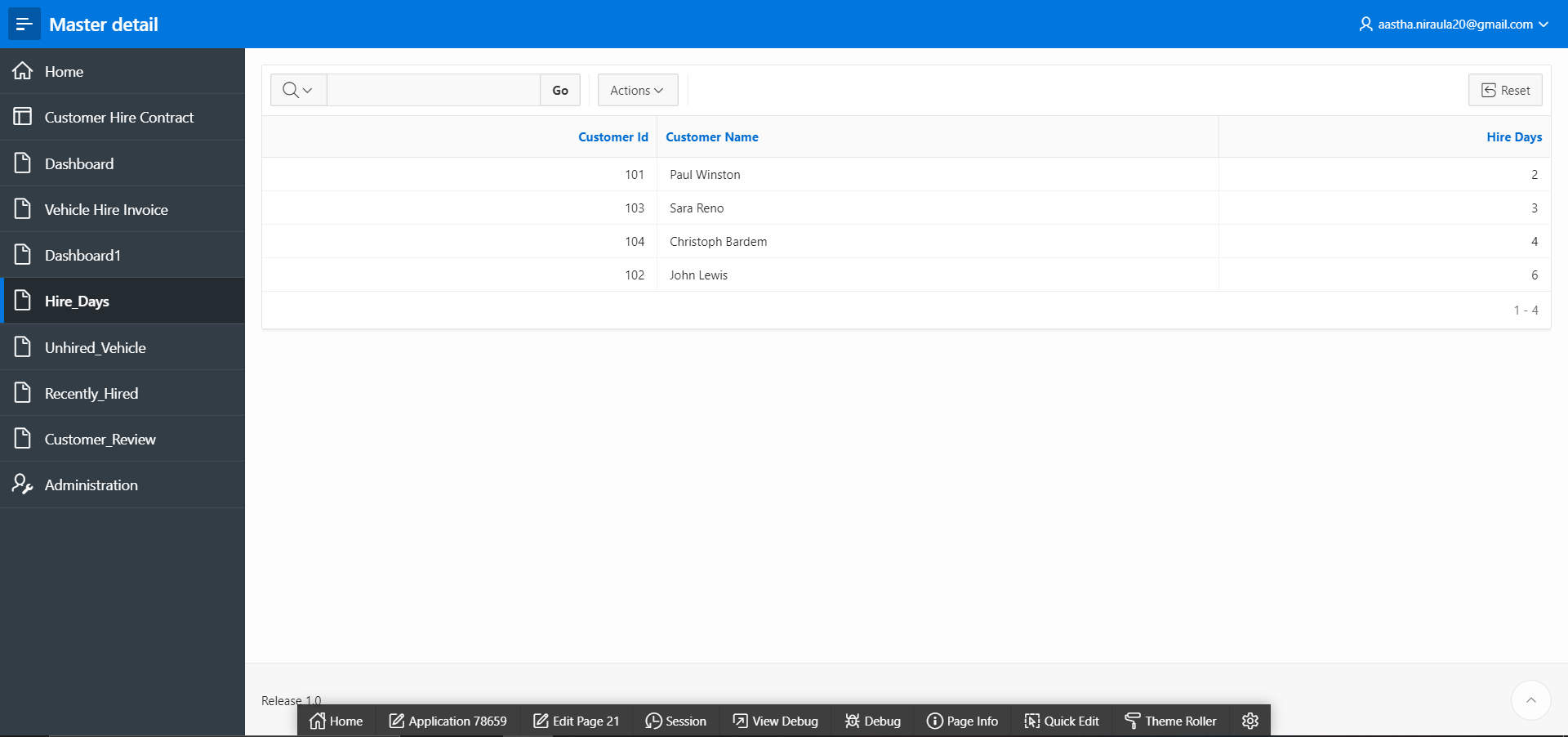
SELECT c.Customer\_ID, c.Customer\_name, h.hire\_days

FROM Customer c, Hire h

WHERE c.customer\_id = h.customer\_id

AND hire\_days >1

ORDER BY Hire\_days ;



## 2. SQL Apex Report Number: 2

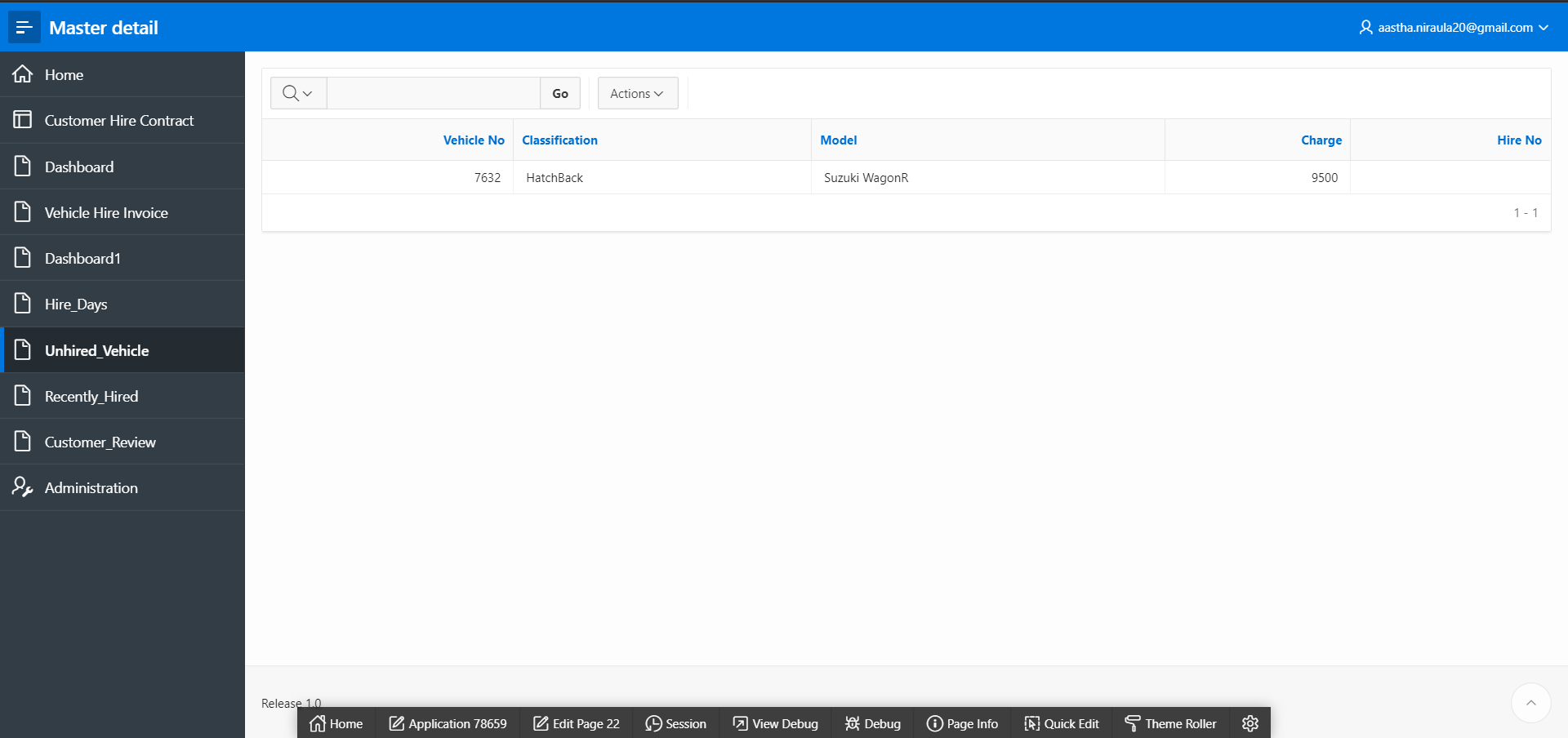
The report includes two tables’ vehicles and hire and applies the condition that if the vehicles have not been hired even once then the list is displayed for the vehicles that are not hired.

SELECT v.vehicle\_no, v.classification, v.model, v.charge, h.hire\_no

FROM vehicles v, hire h

WHERE h.hire\_no (+) = v.hire\_no

AND v.hire\_no is NULL;



## 3. SQL Apex Report Number: 3

The report gives information about the vehicle that has been recently hired.

SELECT contract\_no, start\_date, customer\_id

FROM Contract

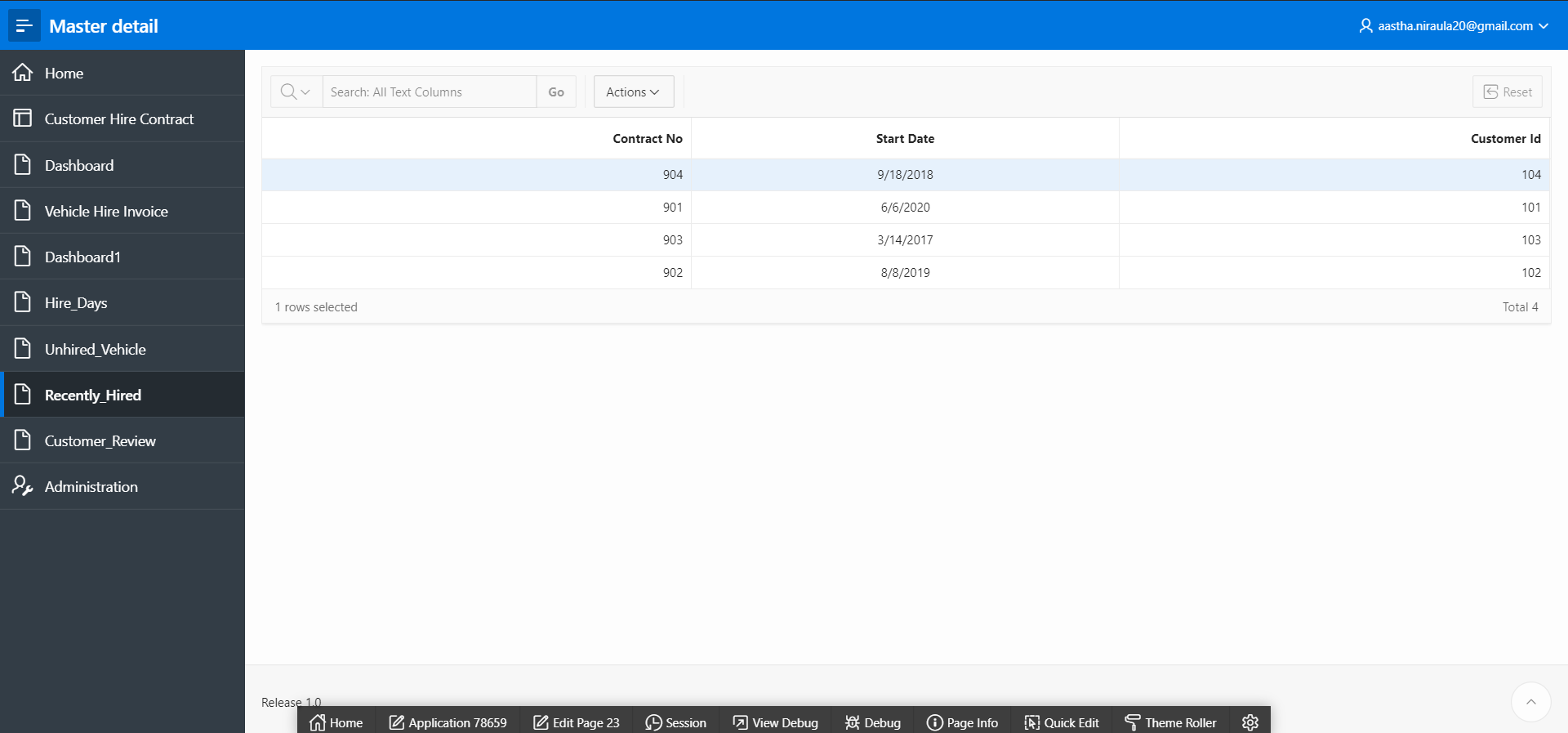
WHERE (start\_date, customer\_id)

IN

(SELECT MAX(start\_date), customer\_id

FROM Contract

GROUP BY customer\_id);



## 4. SQL Apex Report Number: 4

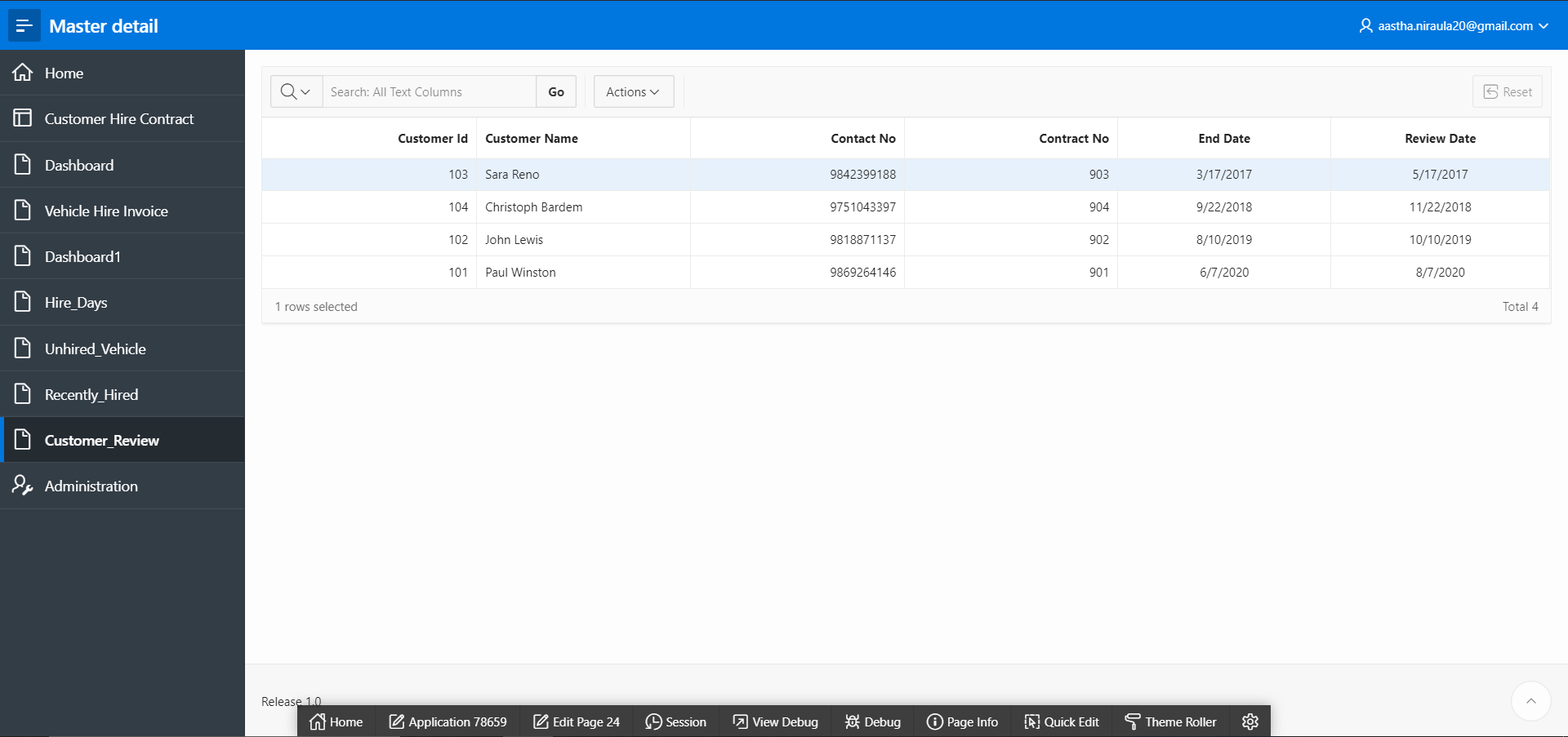
The report merges two tables Contract and Customer which helps in taking the review from the customers. The review date is recorded 2 months prior to the contract ending date.

SELECT ct.customer\_id, c.customer\_name, c.contact\_no, ct.Contract\_no, ct.end\_date, add\_months(end\_date, 2) review\_date

FROM contract ct INNER JOIN customer c

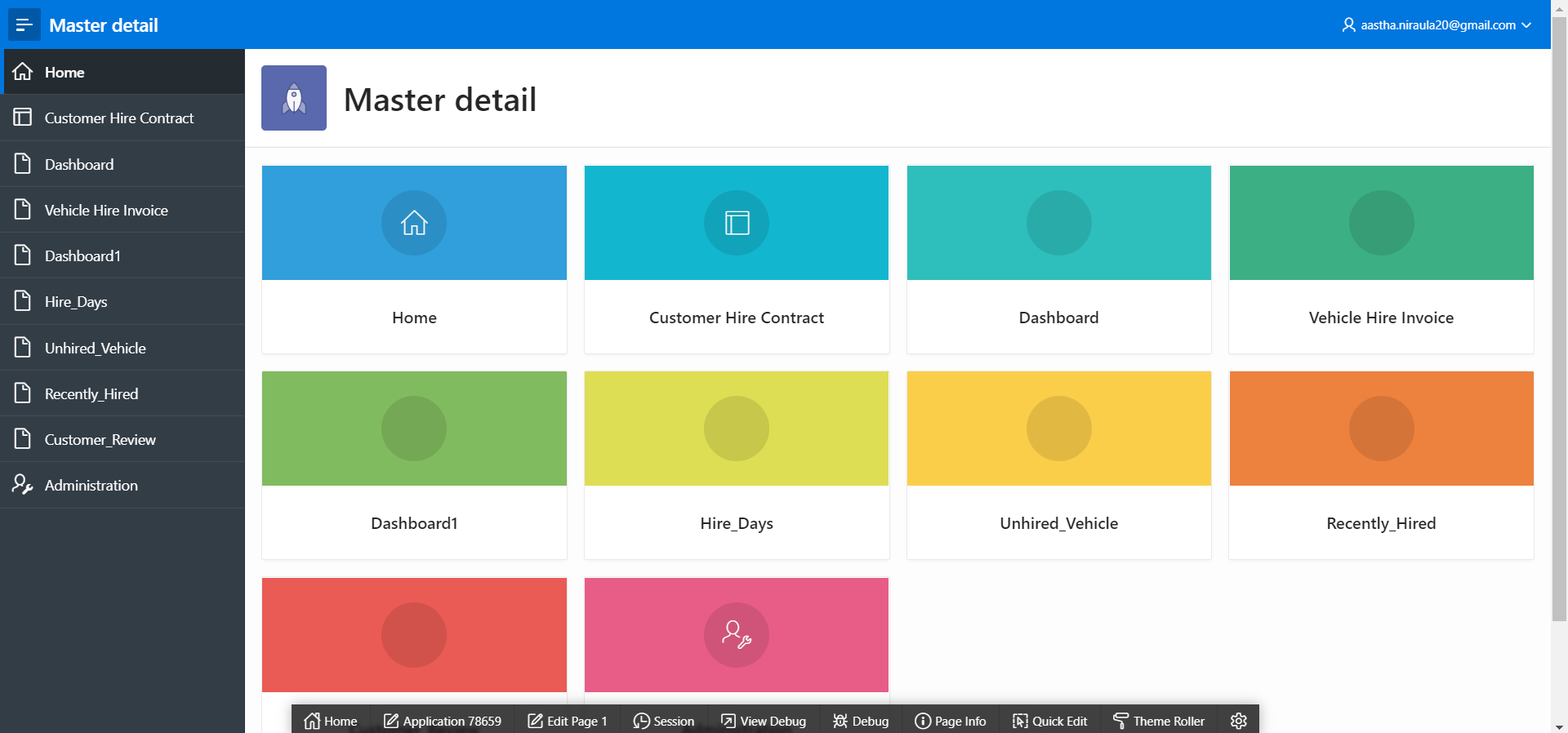
ON ct.customer\_id=c.customer\_id

ORDER BY add\_months(end\_date,2);

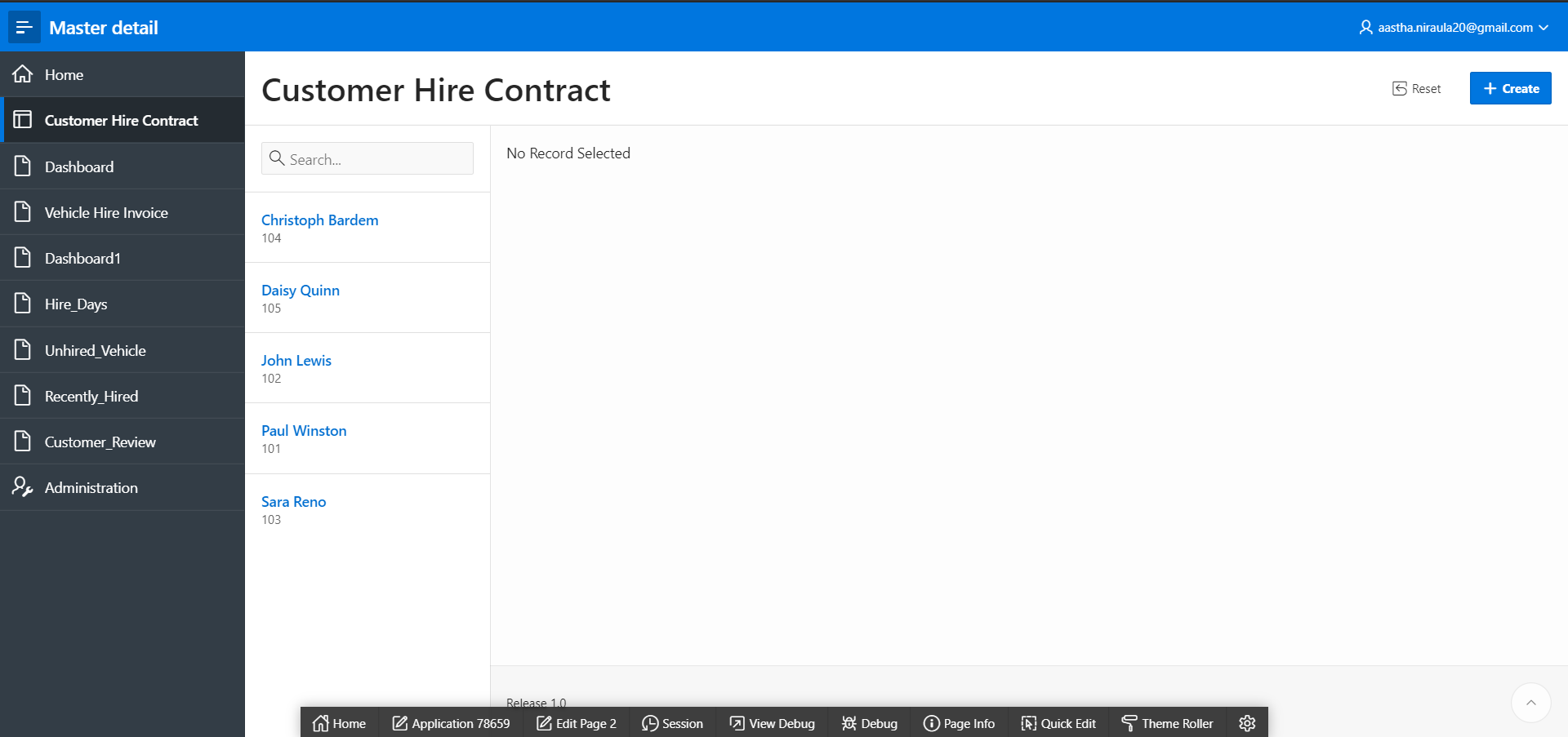


# Task 3. Application Builder Forms

## Home Page

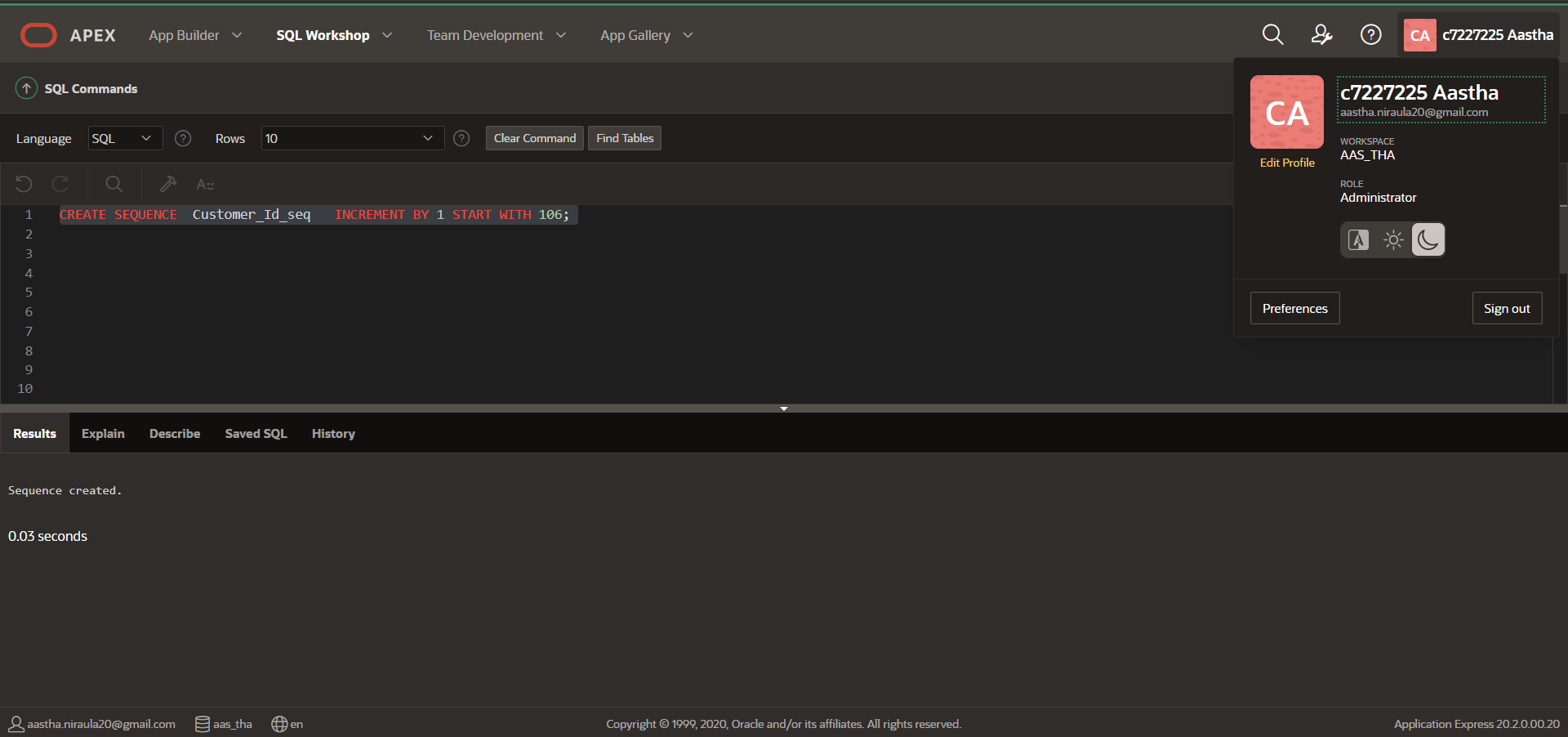


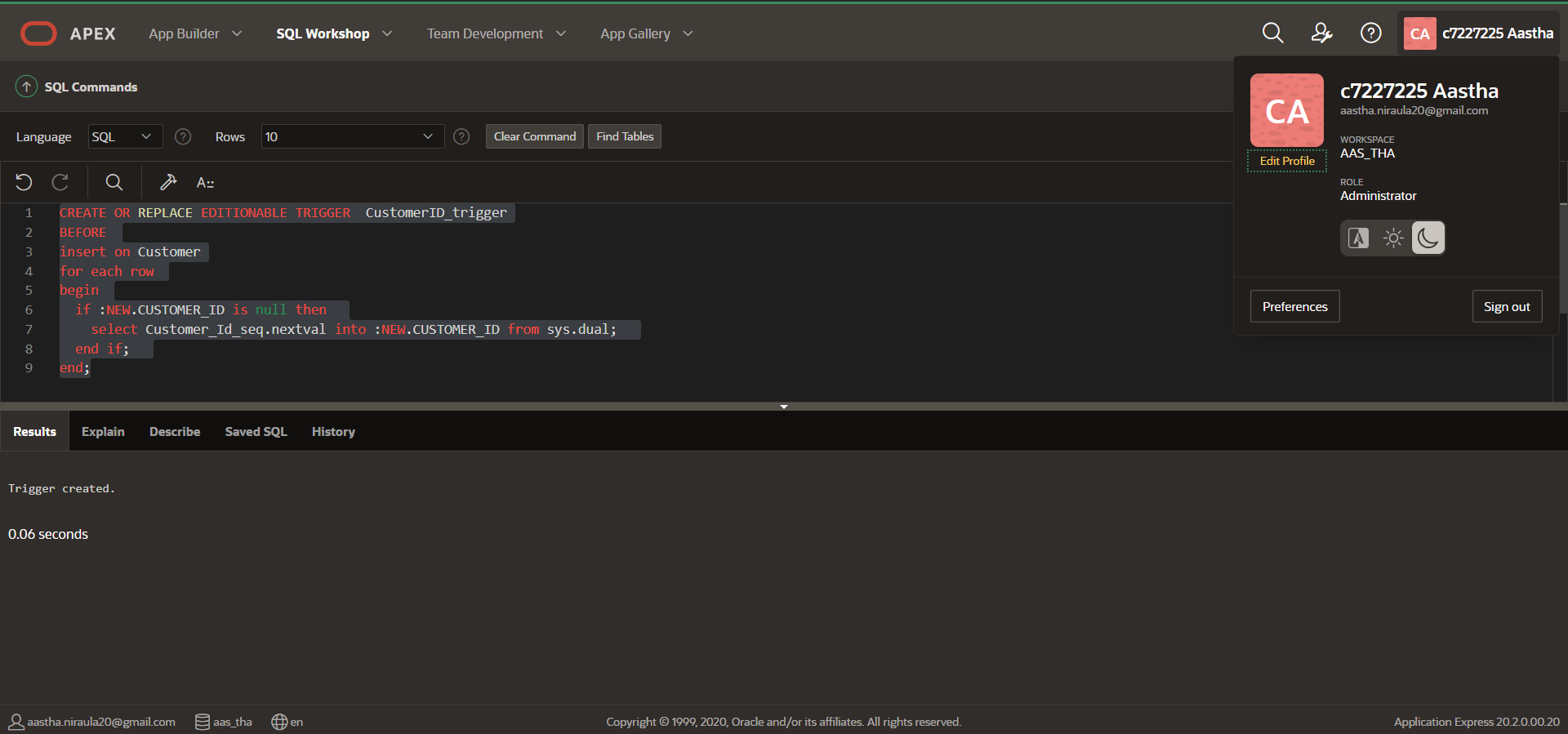
## 1. Master Detail for Customer, Hire and Contract



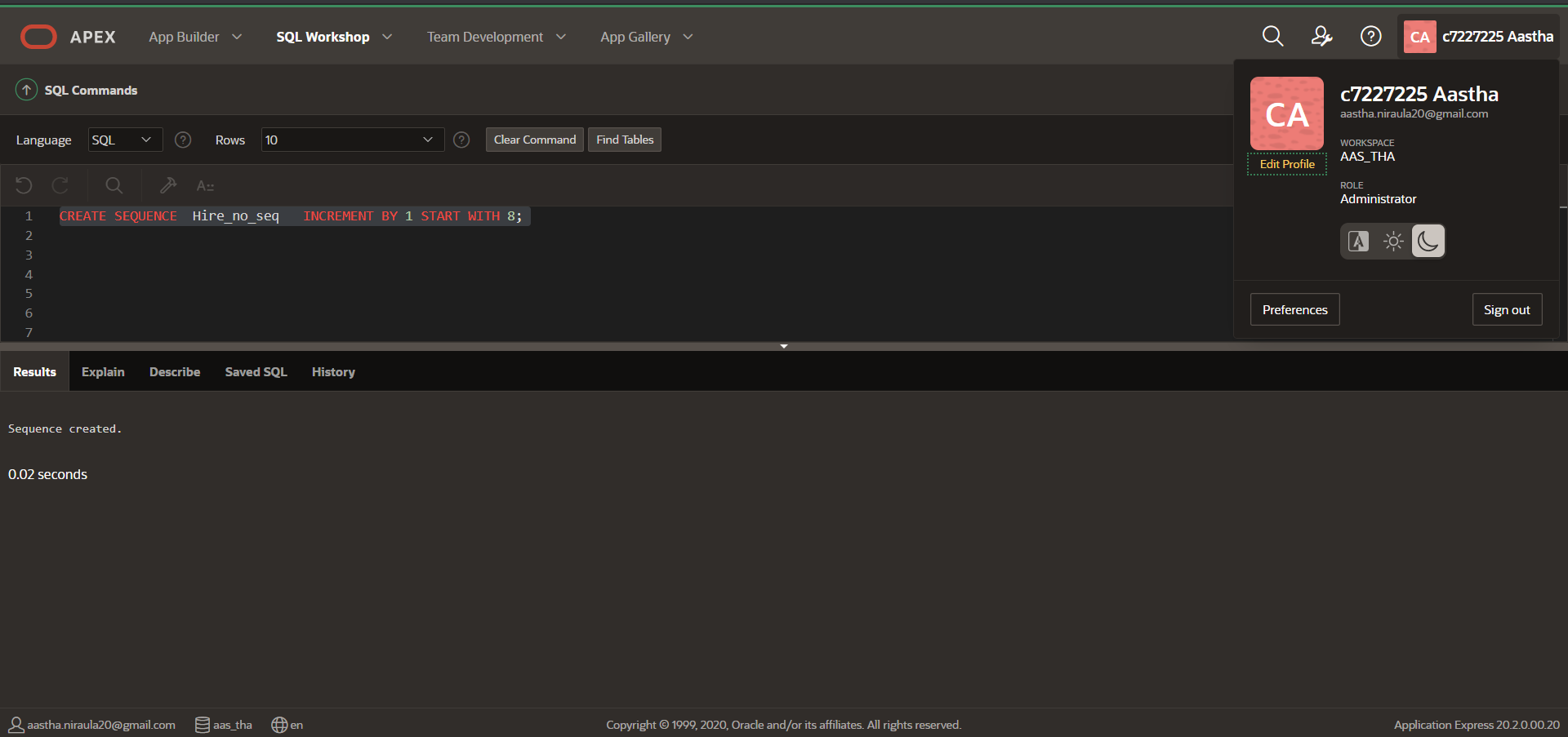
## Creating Sequence and Trigger

### Sequence and trigger for Customer ID





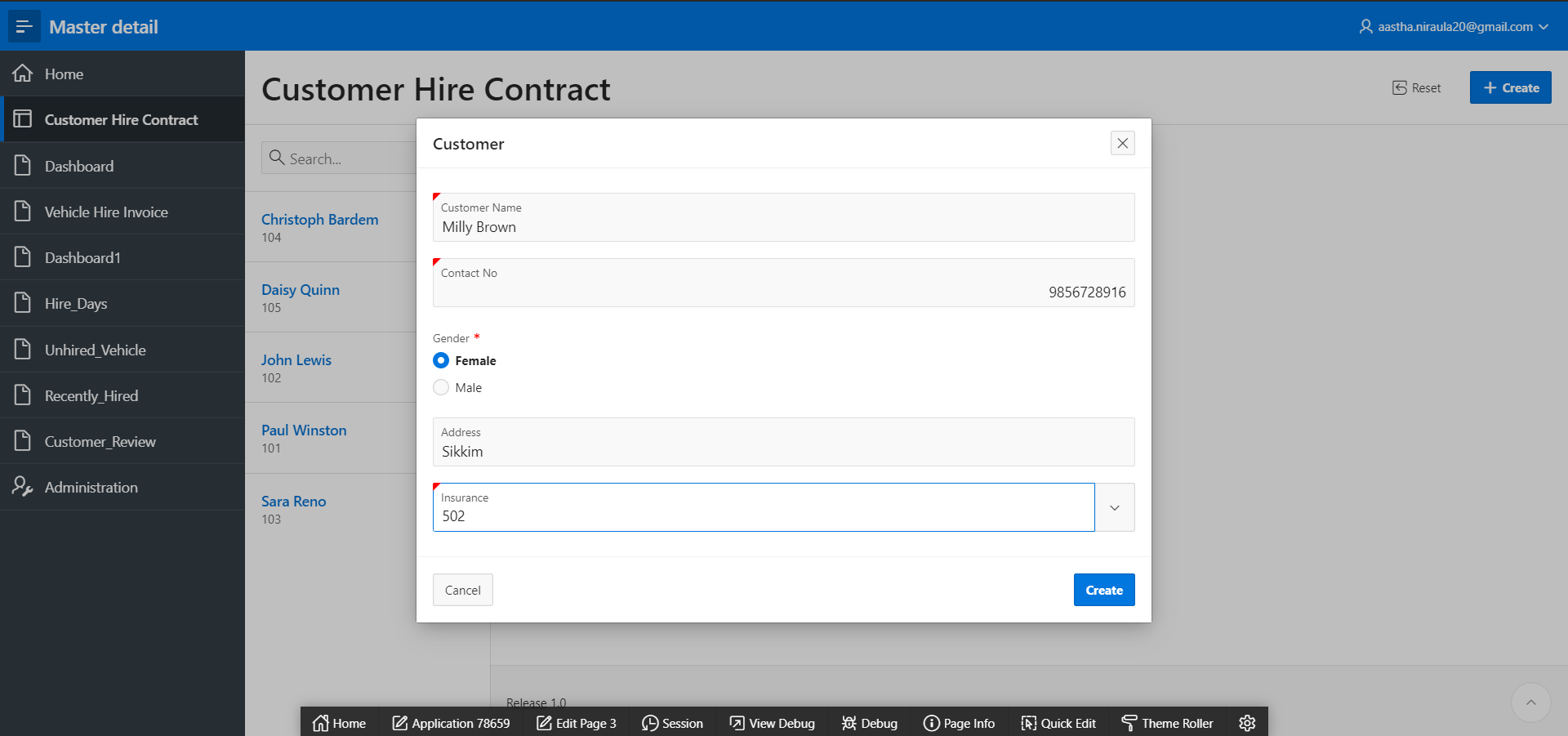
### Sequence and trigger for Hire Number



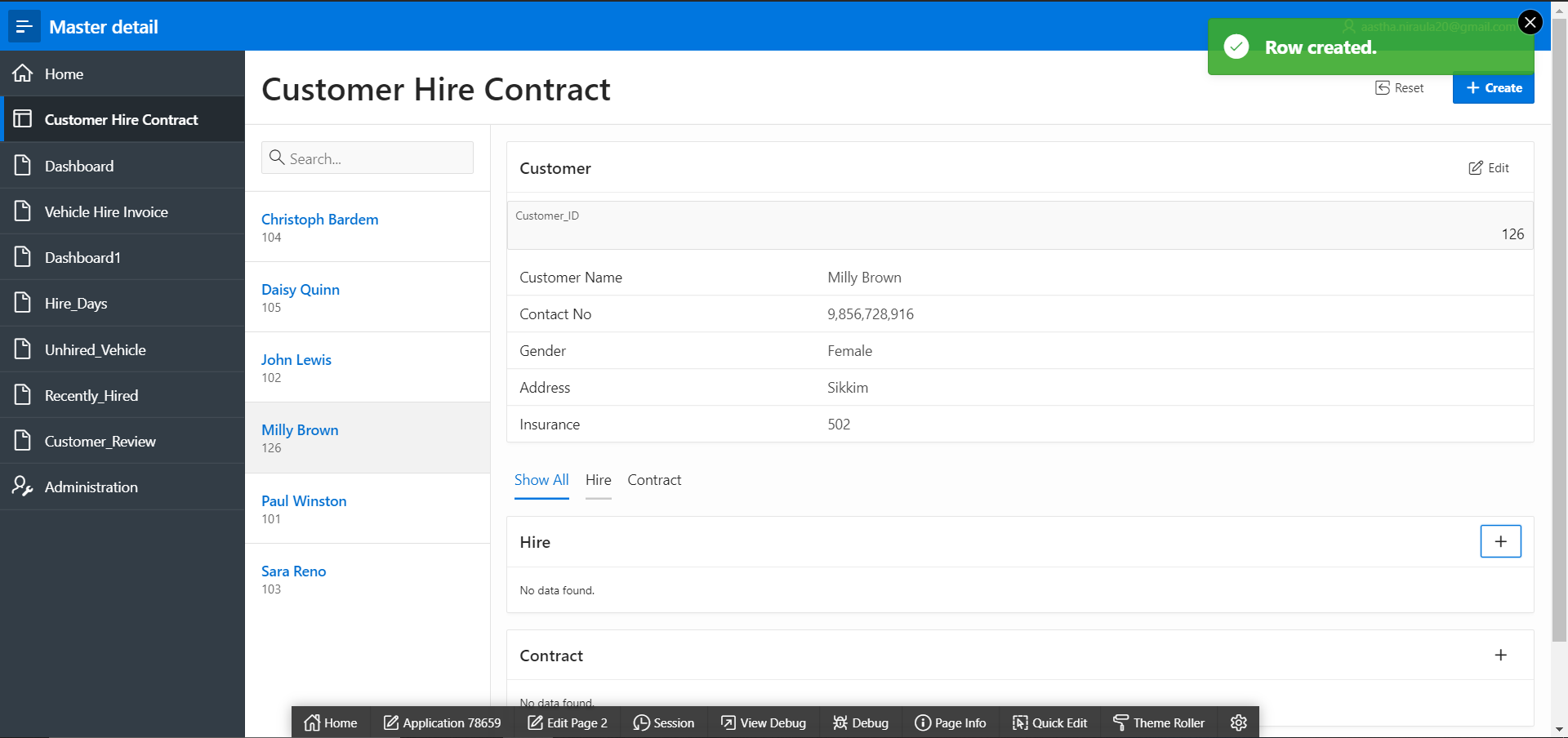


## CRUD Functionality

#### CREATE

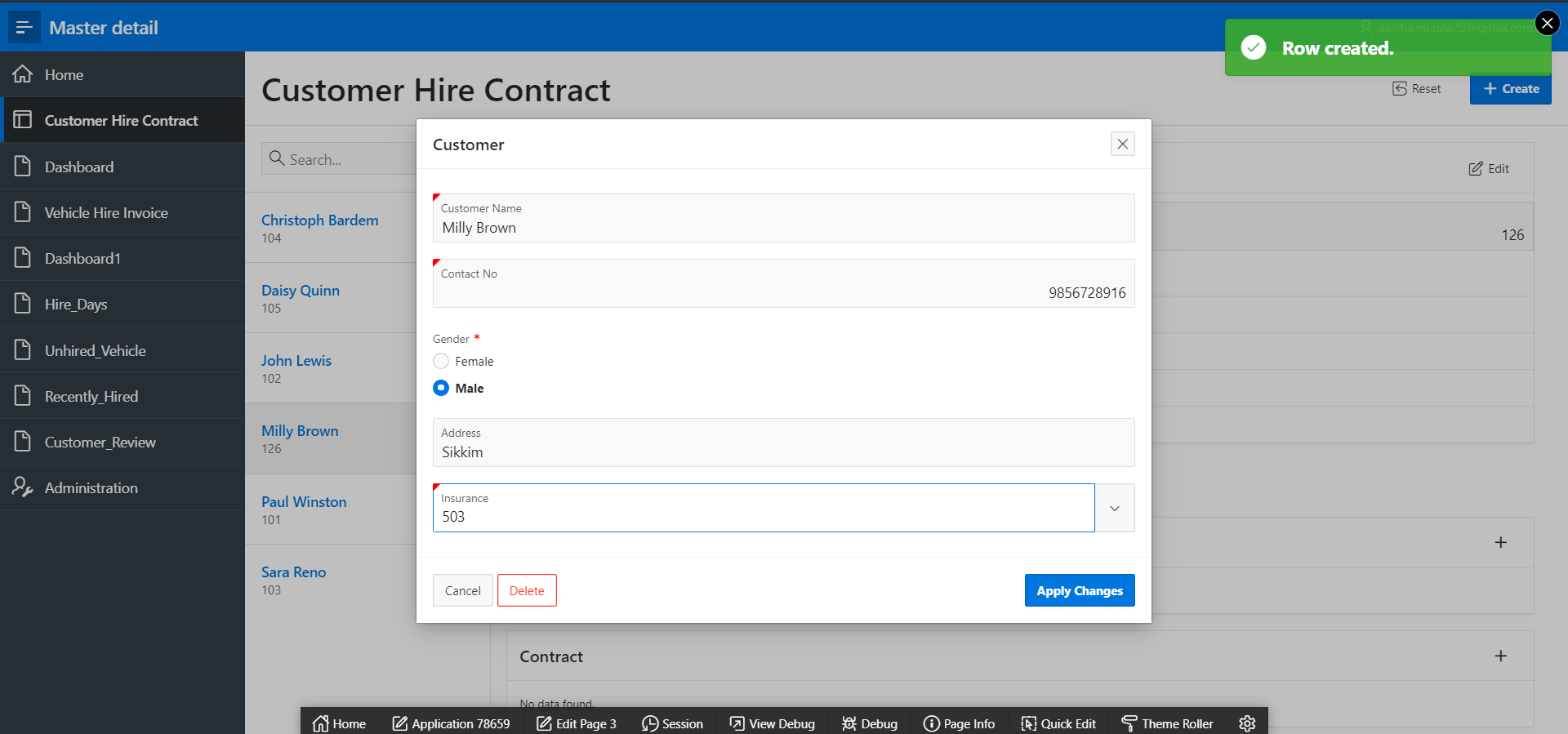


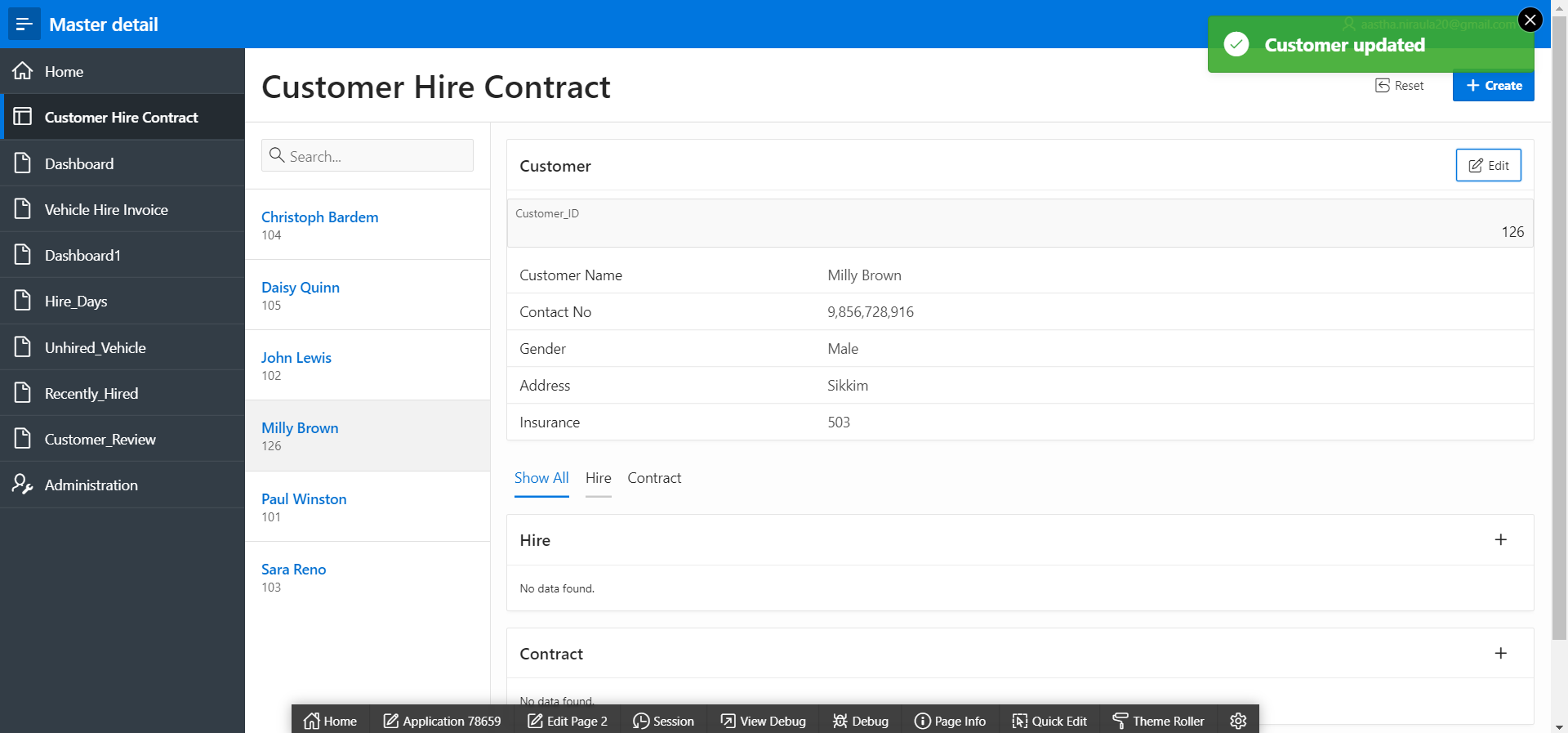
#### READ



#### UPDATE

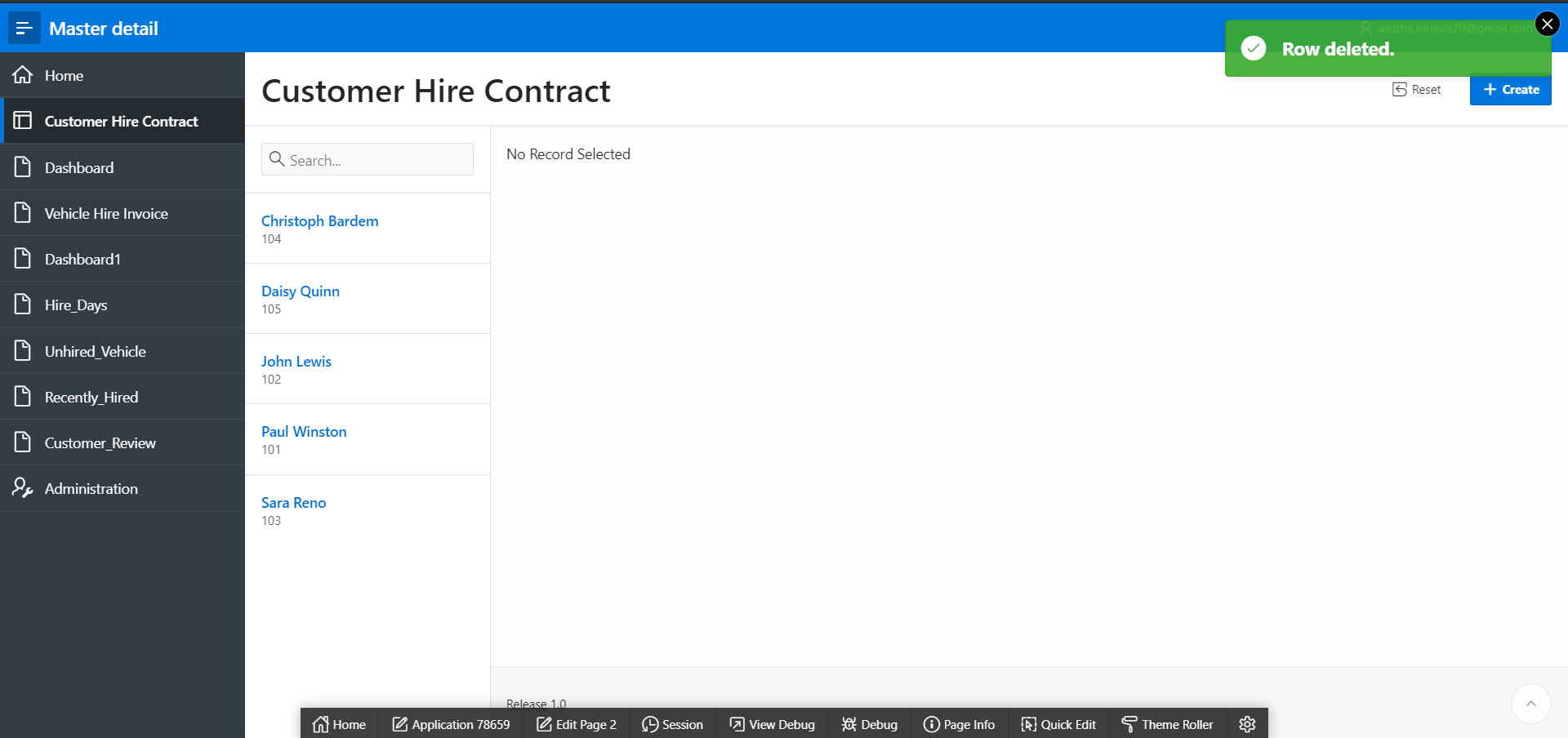
Changed insurance number and Gender of the customer



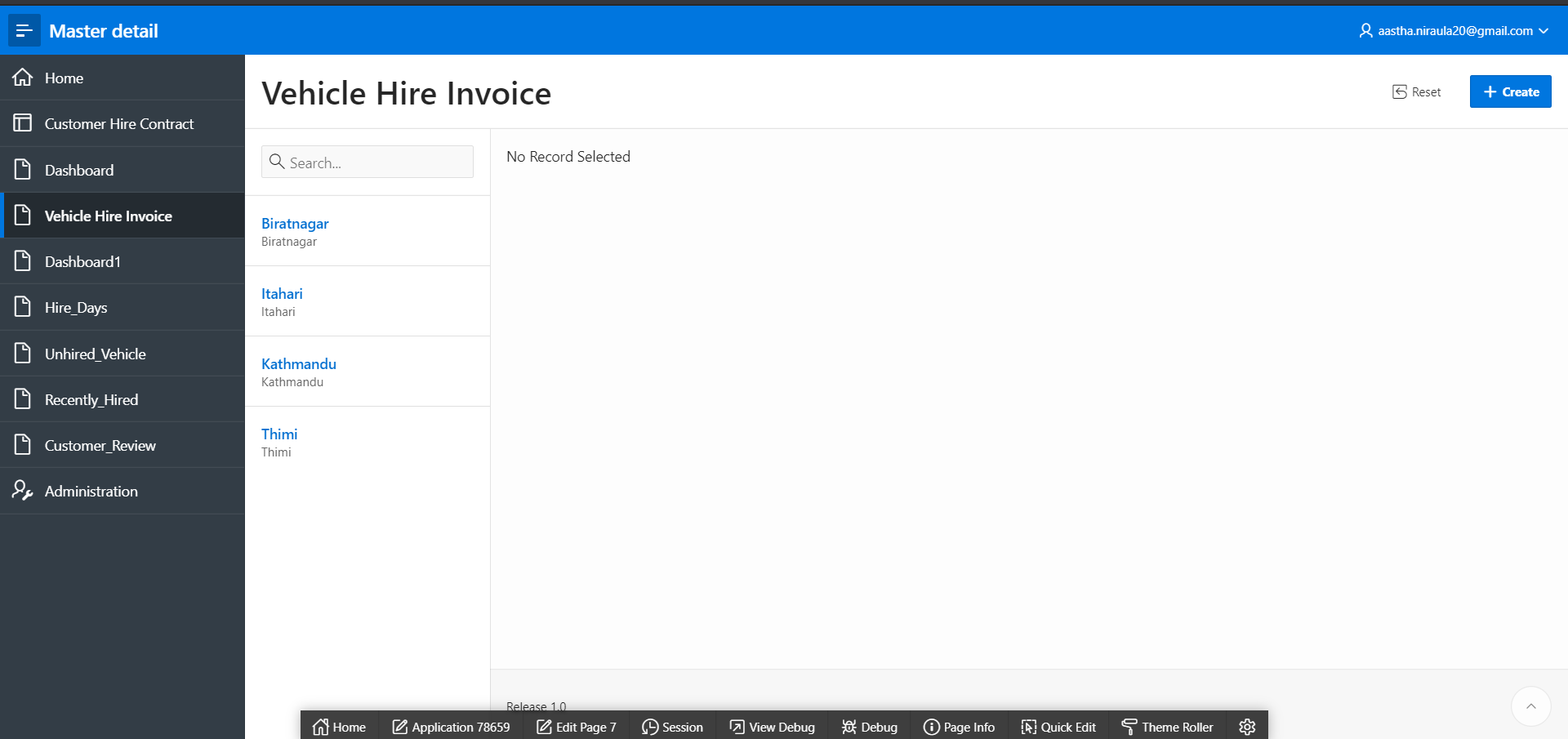


#### DELETE

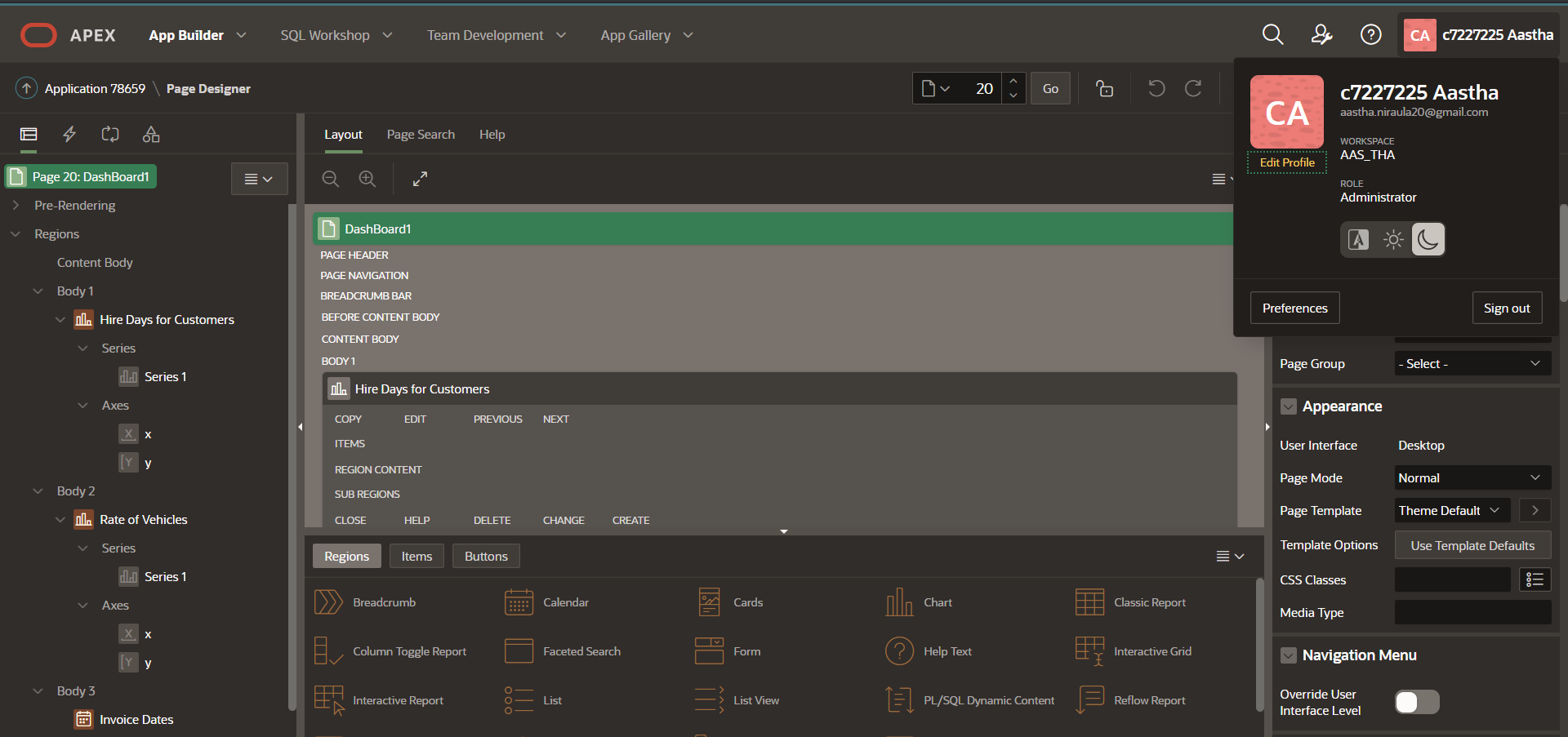


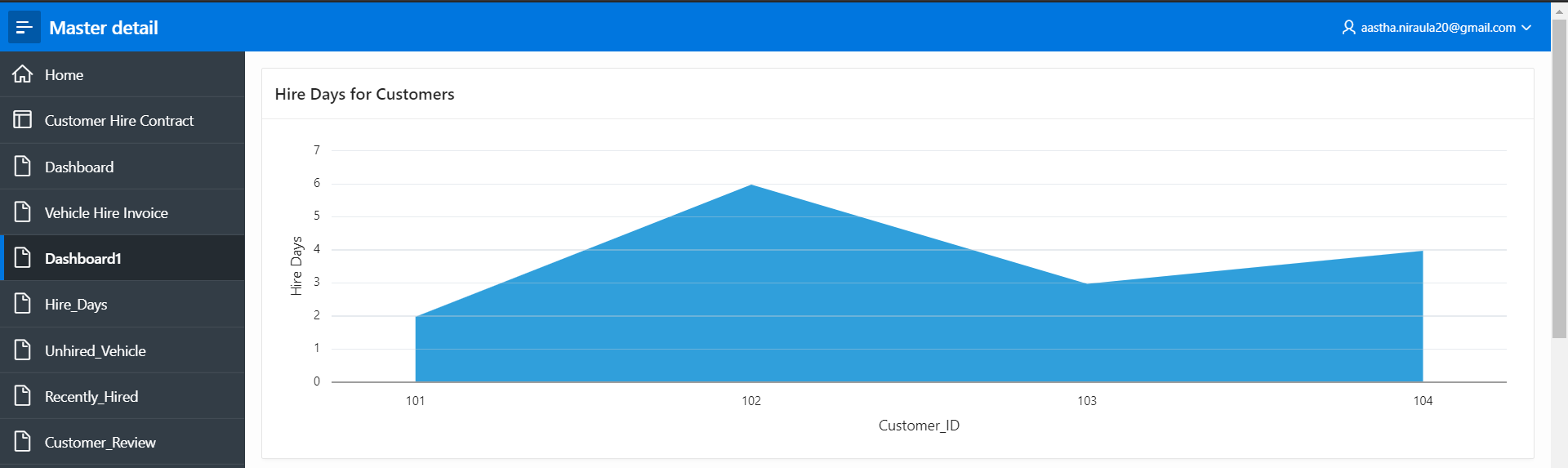


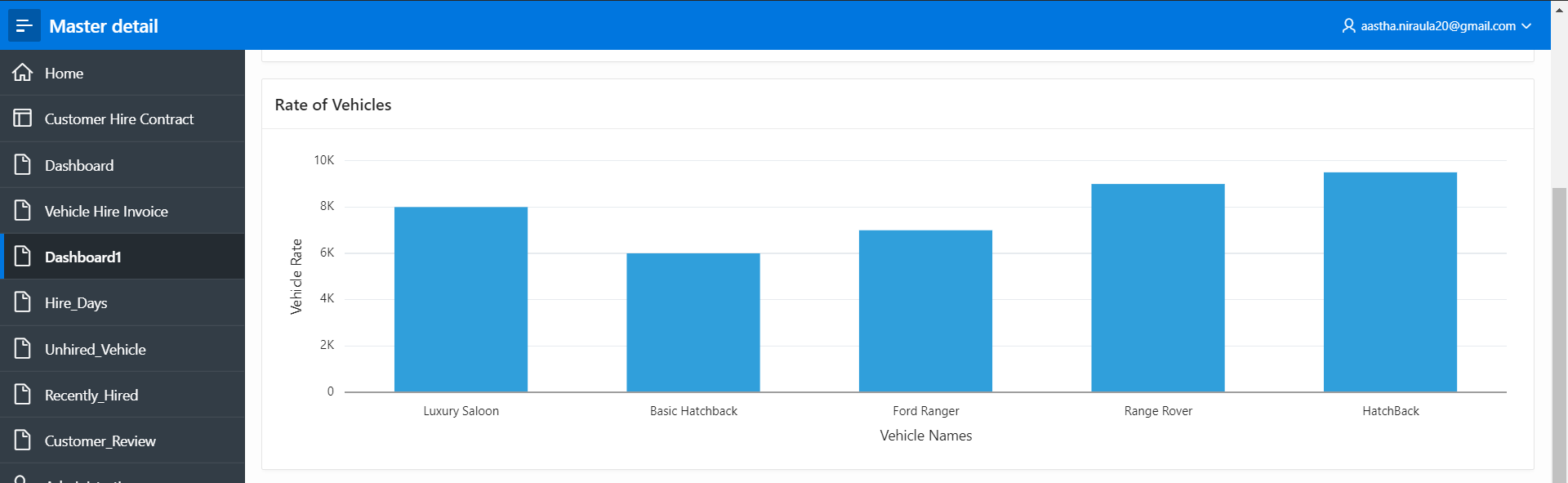
## 2. Master Detail for Vehicle, Hire and Invoice



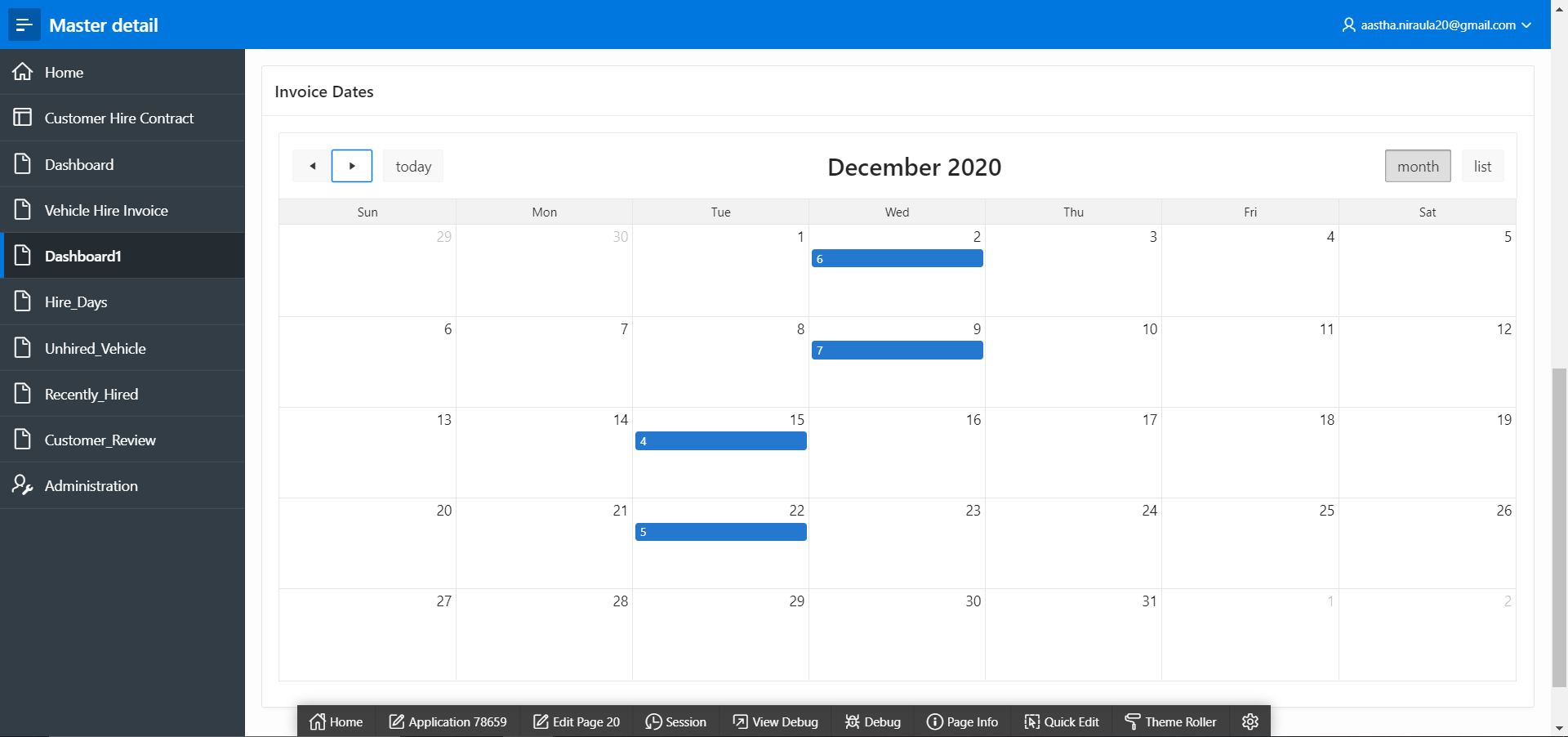
## 3. Dashboard for Vehicle, Hire and Invoice





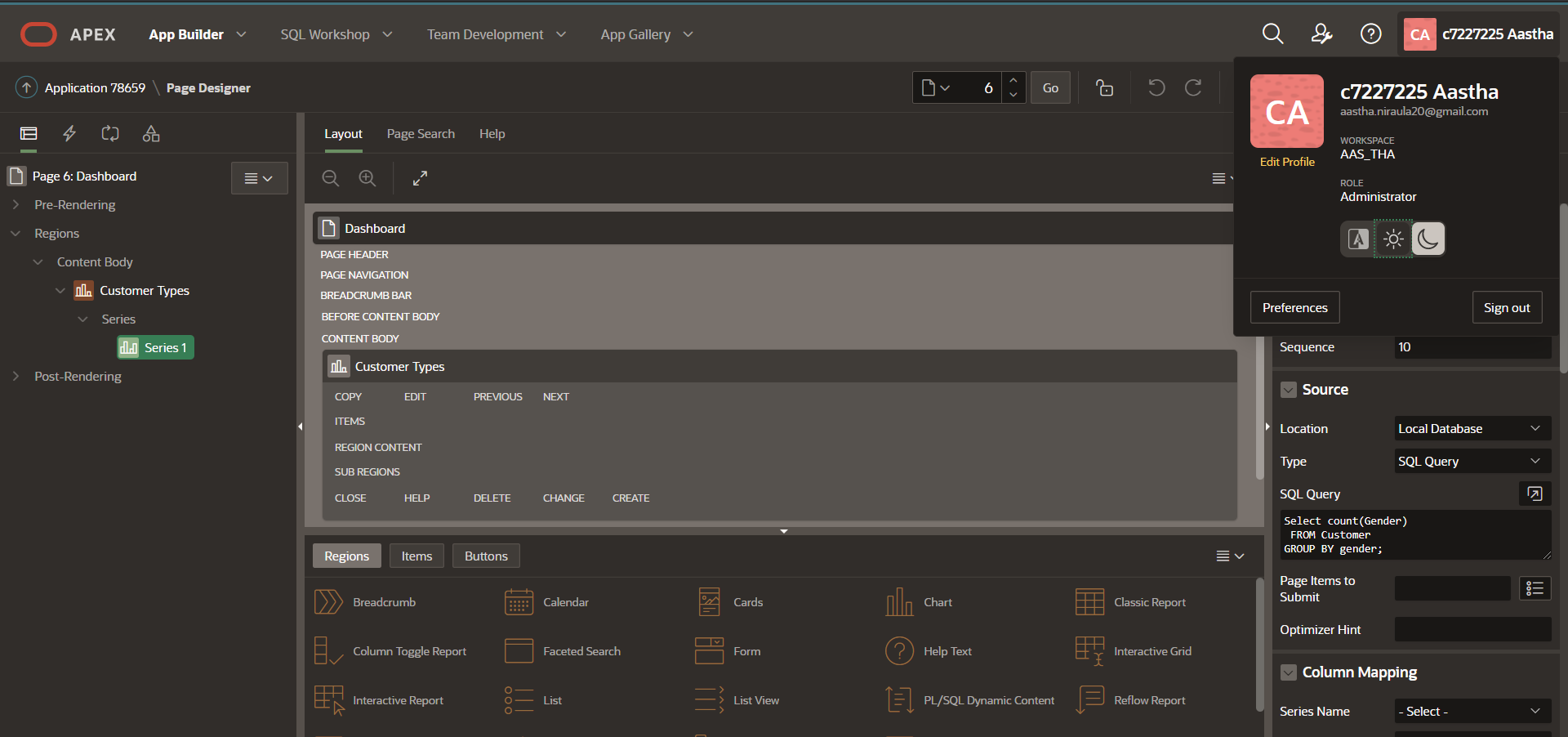


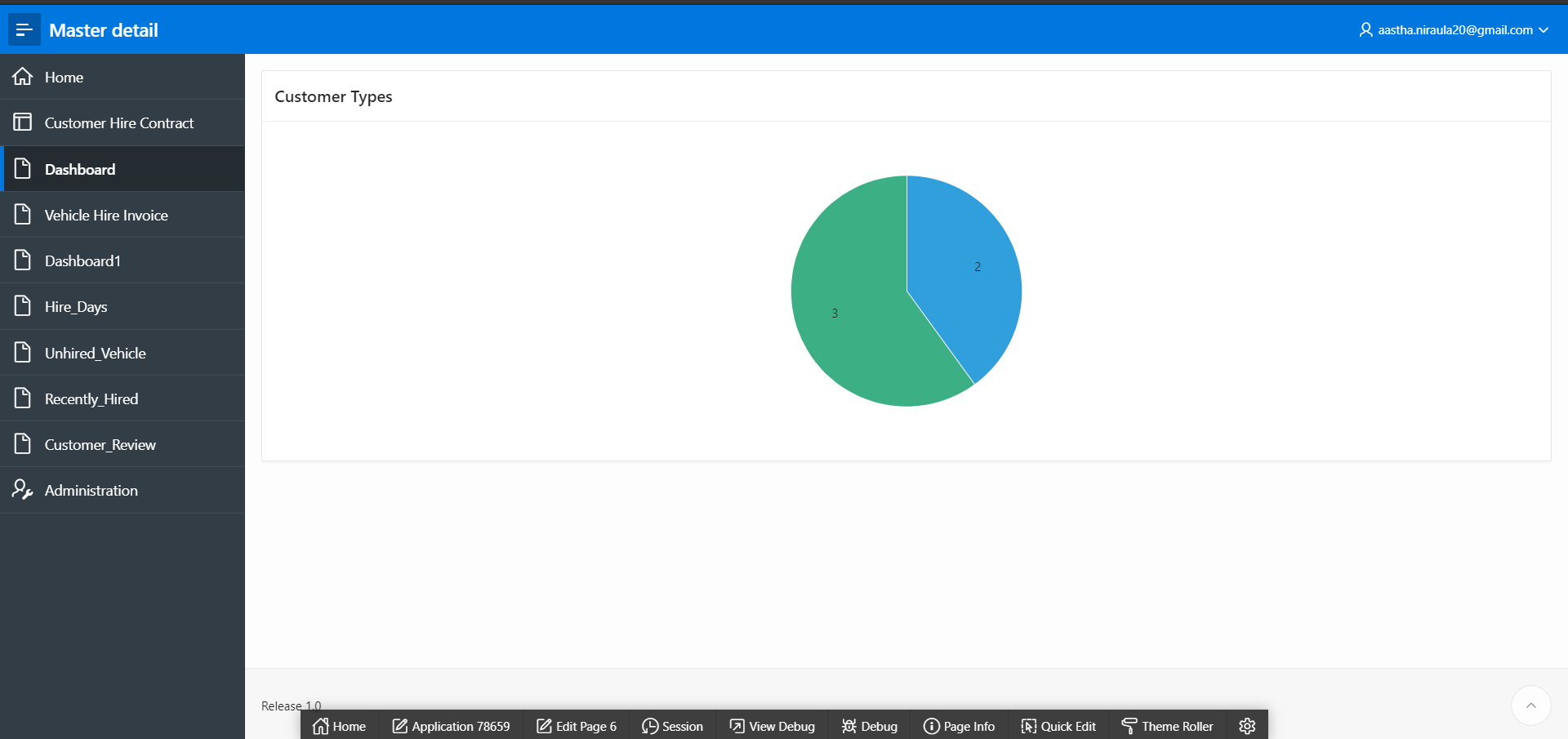
The calendar shows the hire number based on hire dates in the calendar.



## 4. Dashboard for Customer based on Gender

Male: Female (3:2)



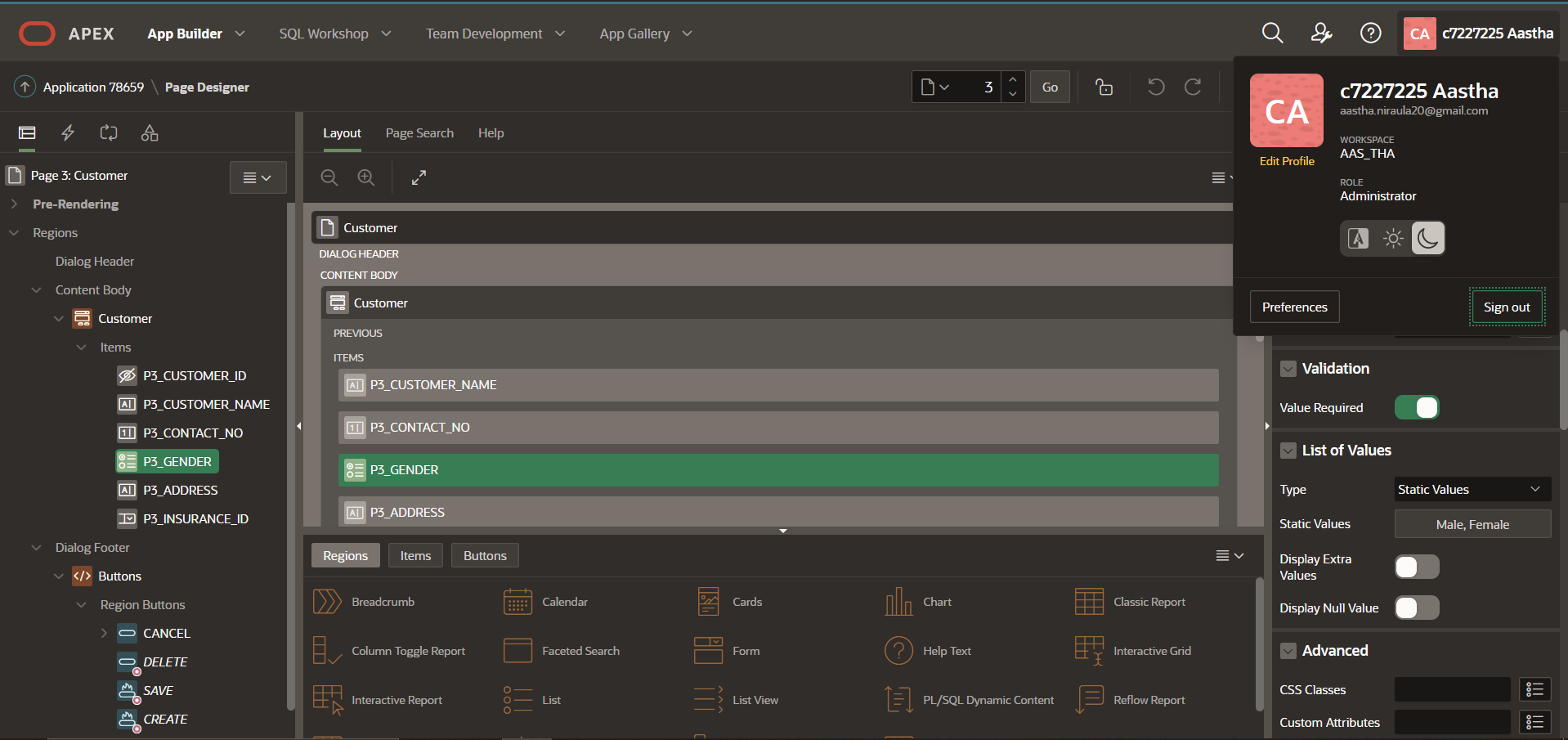


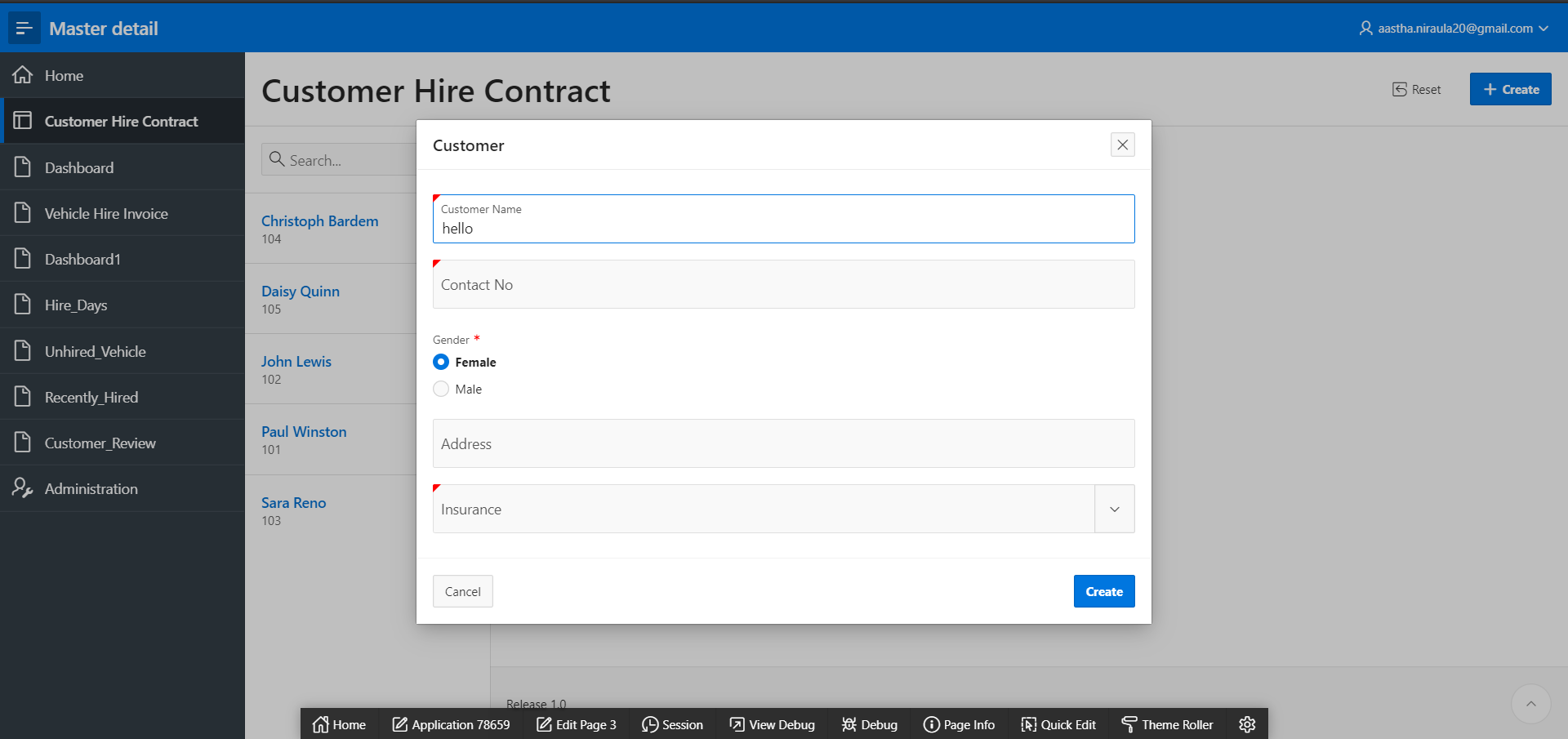
## 

## 5. Using APEX Functionality

### a. Using Radio Group

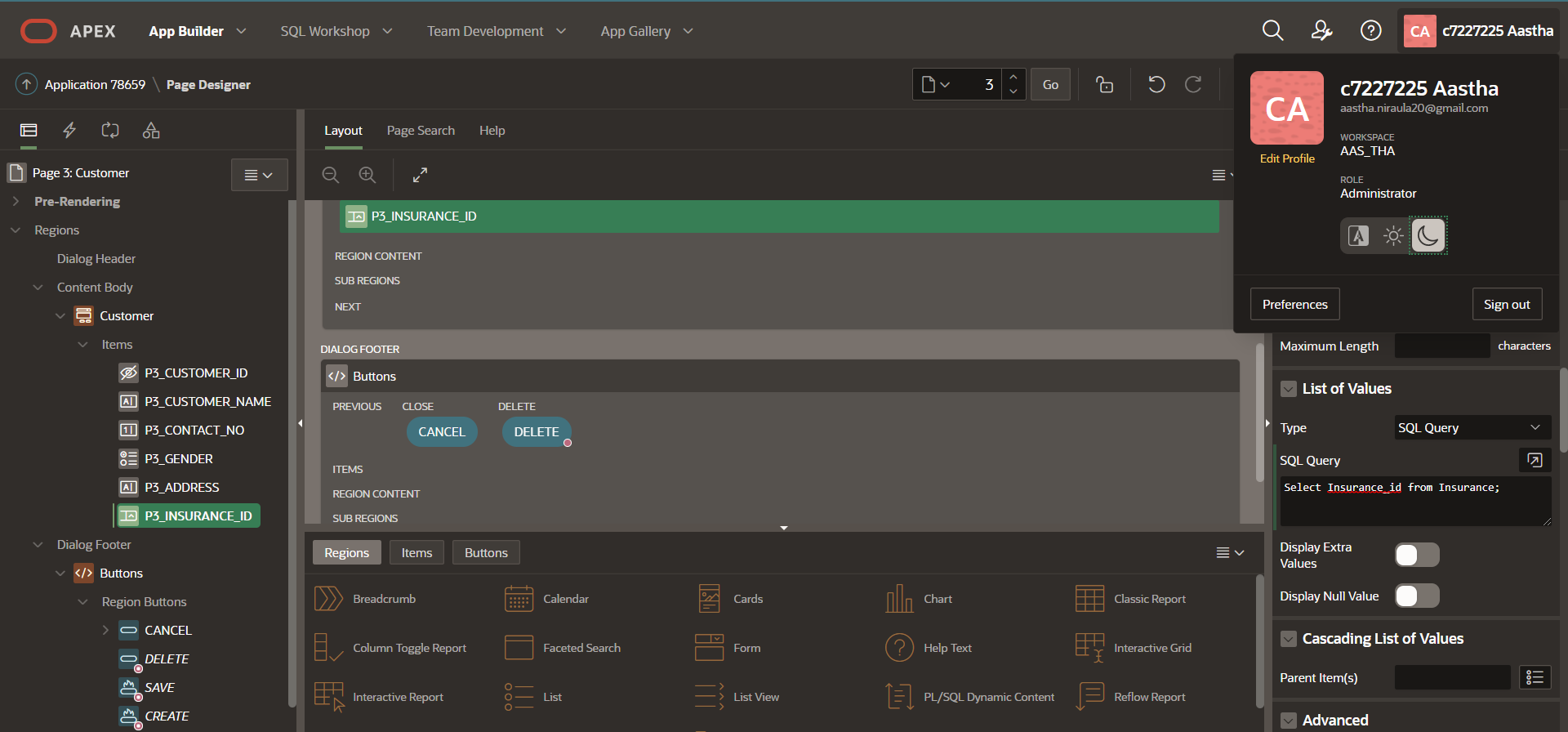
For customer’s gender in customer table:

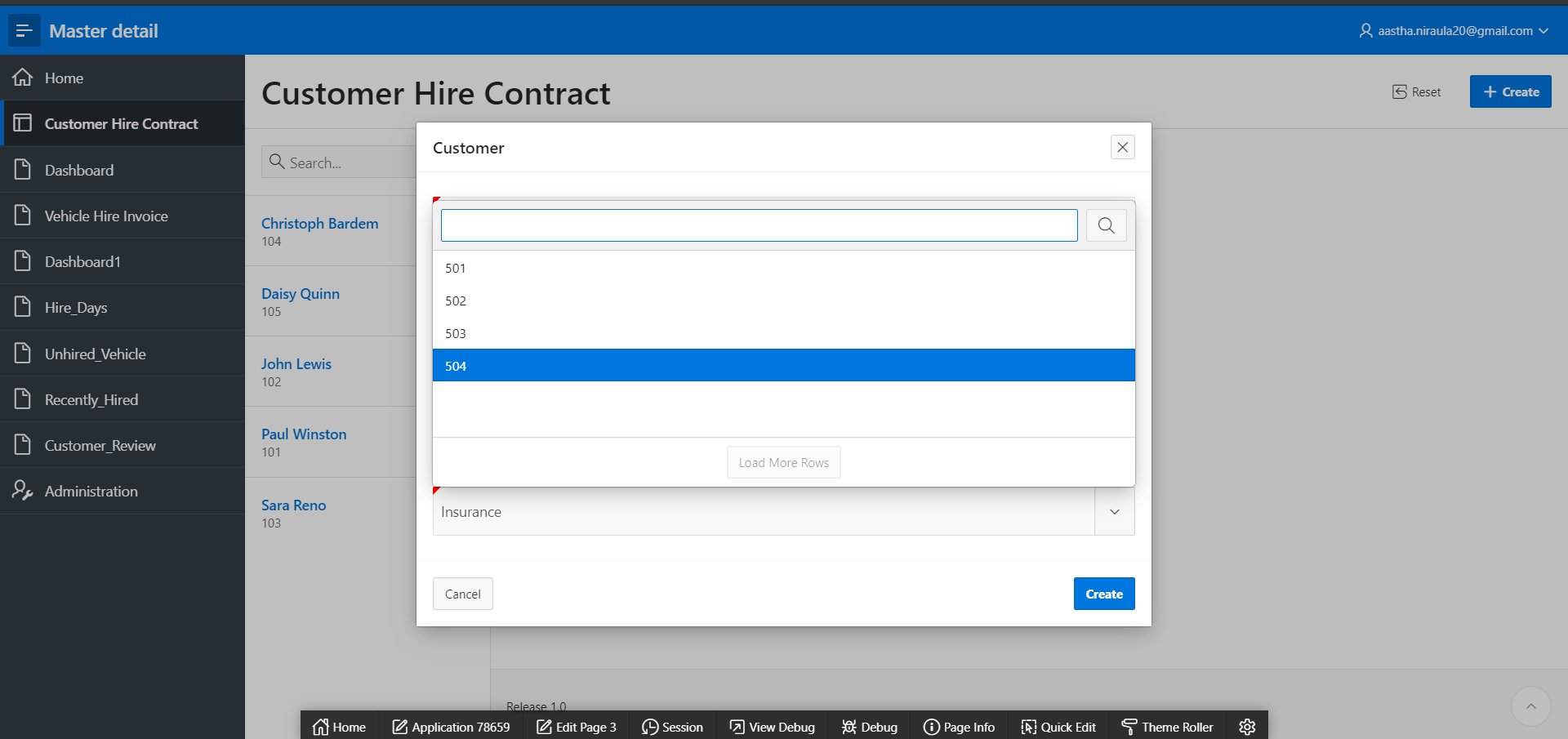




### b. Using Popup LOV and adding SQL command

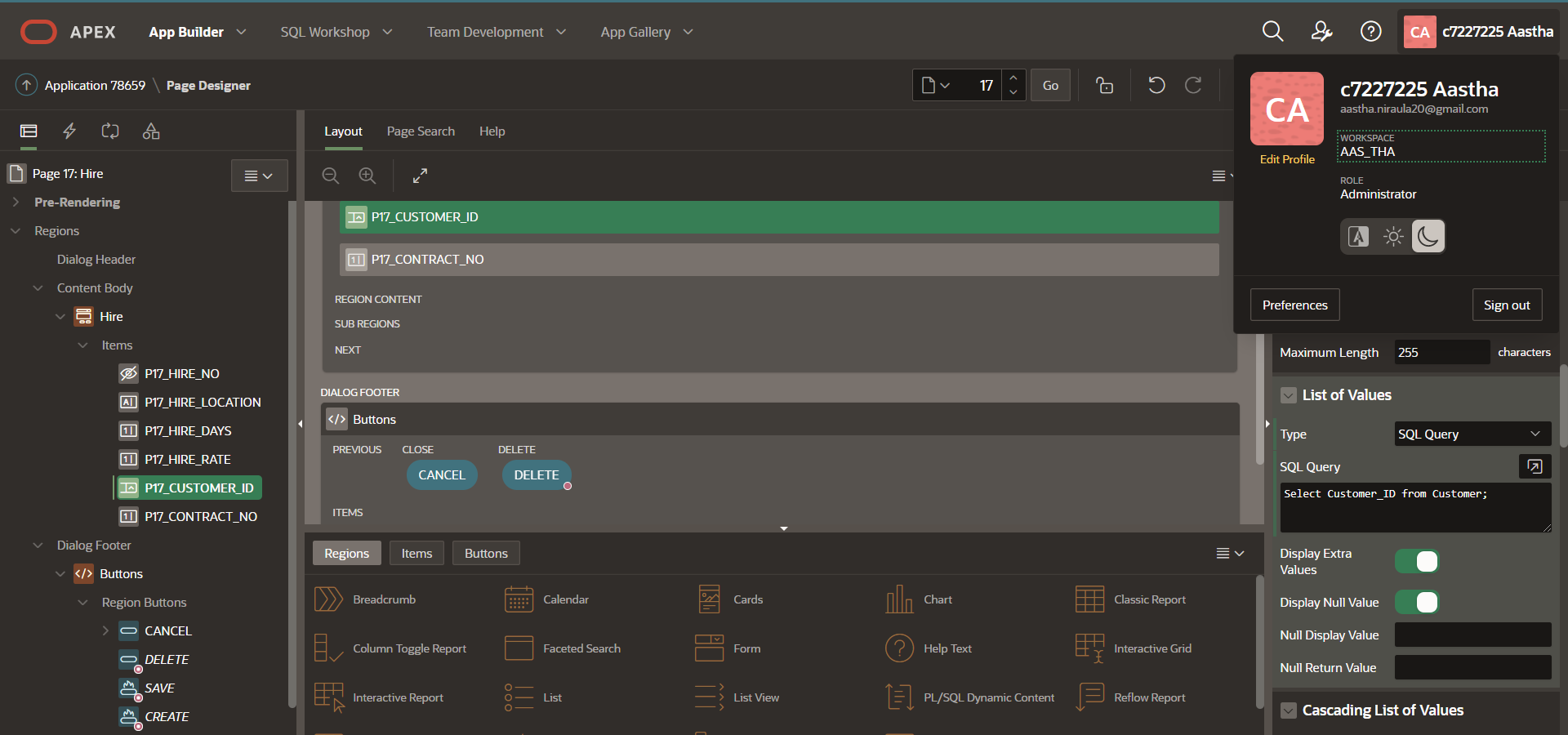
#### For insurance:

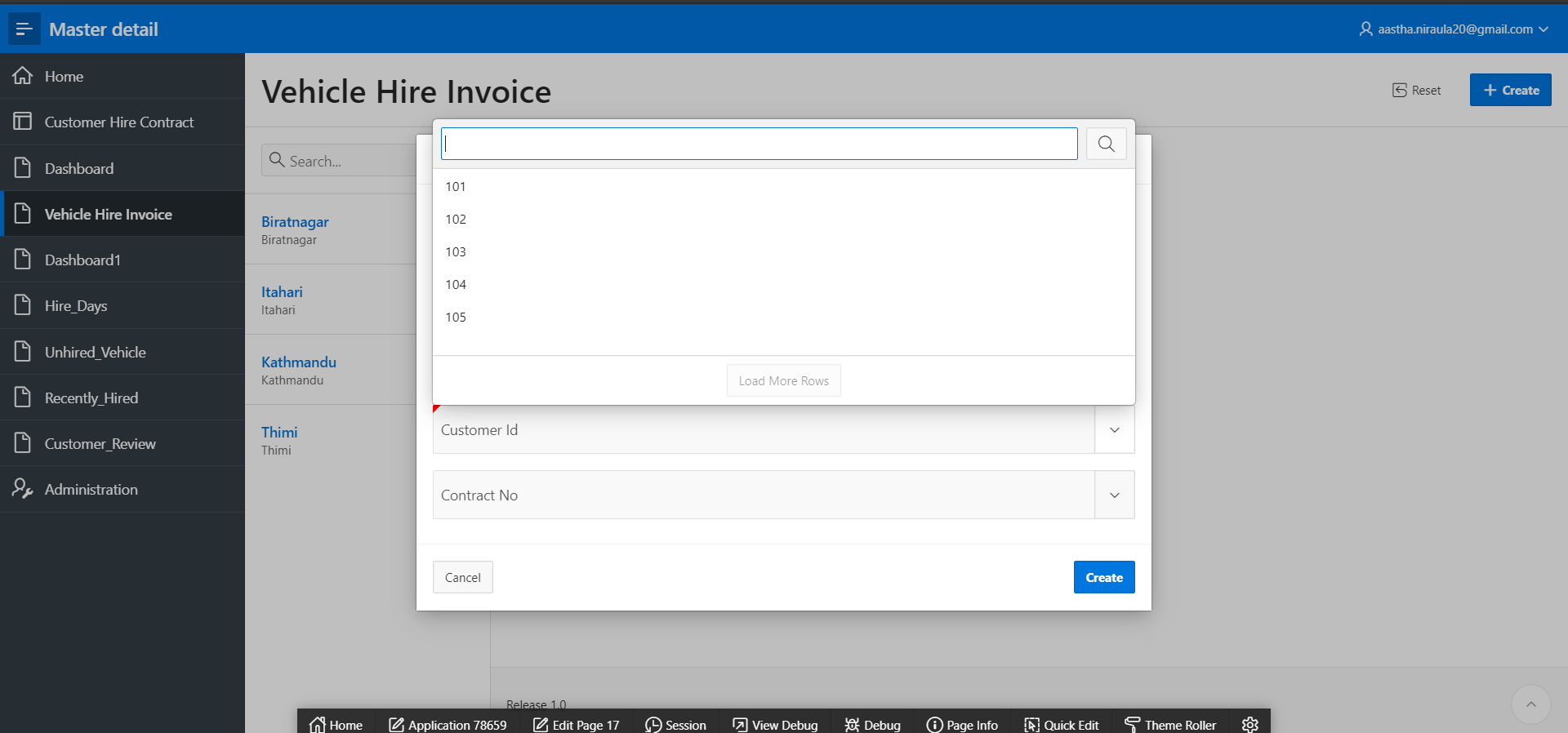




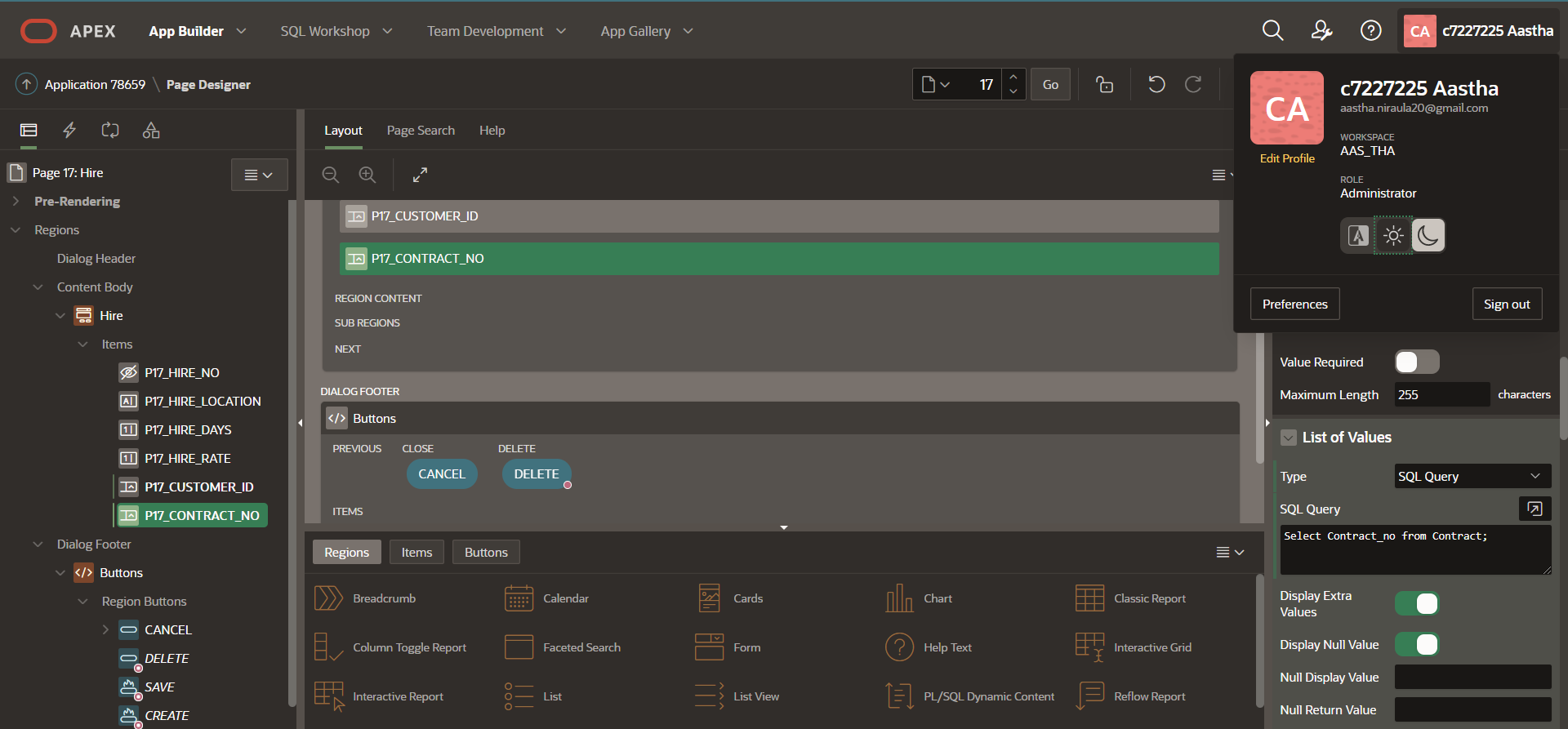
#### For Hire:

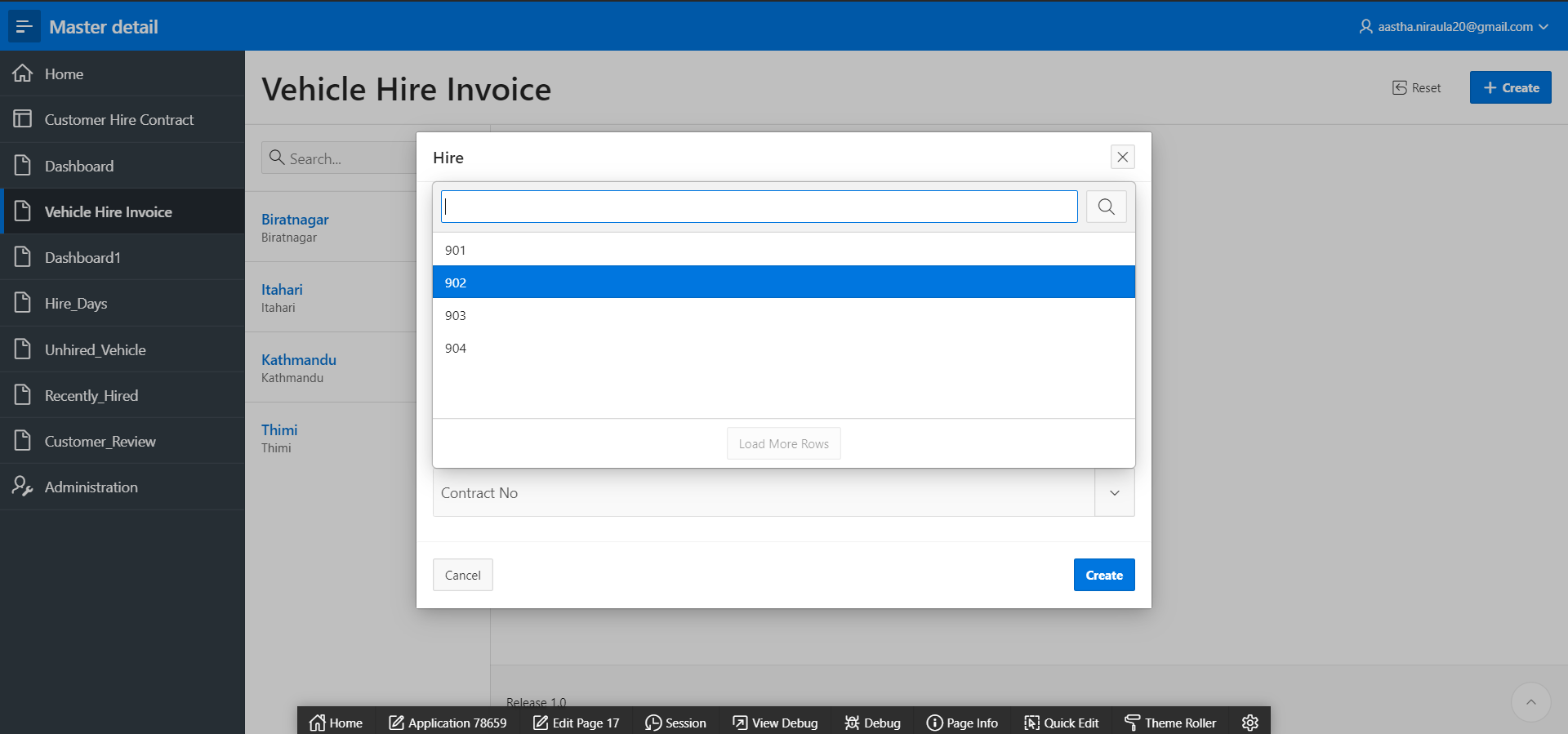
Customer ID



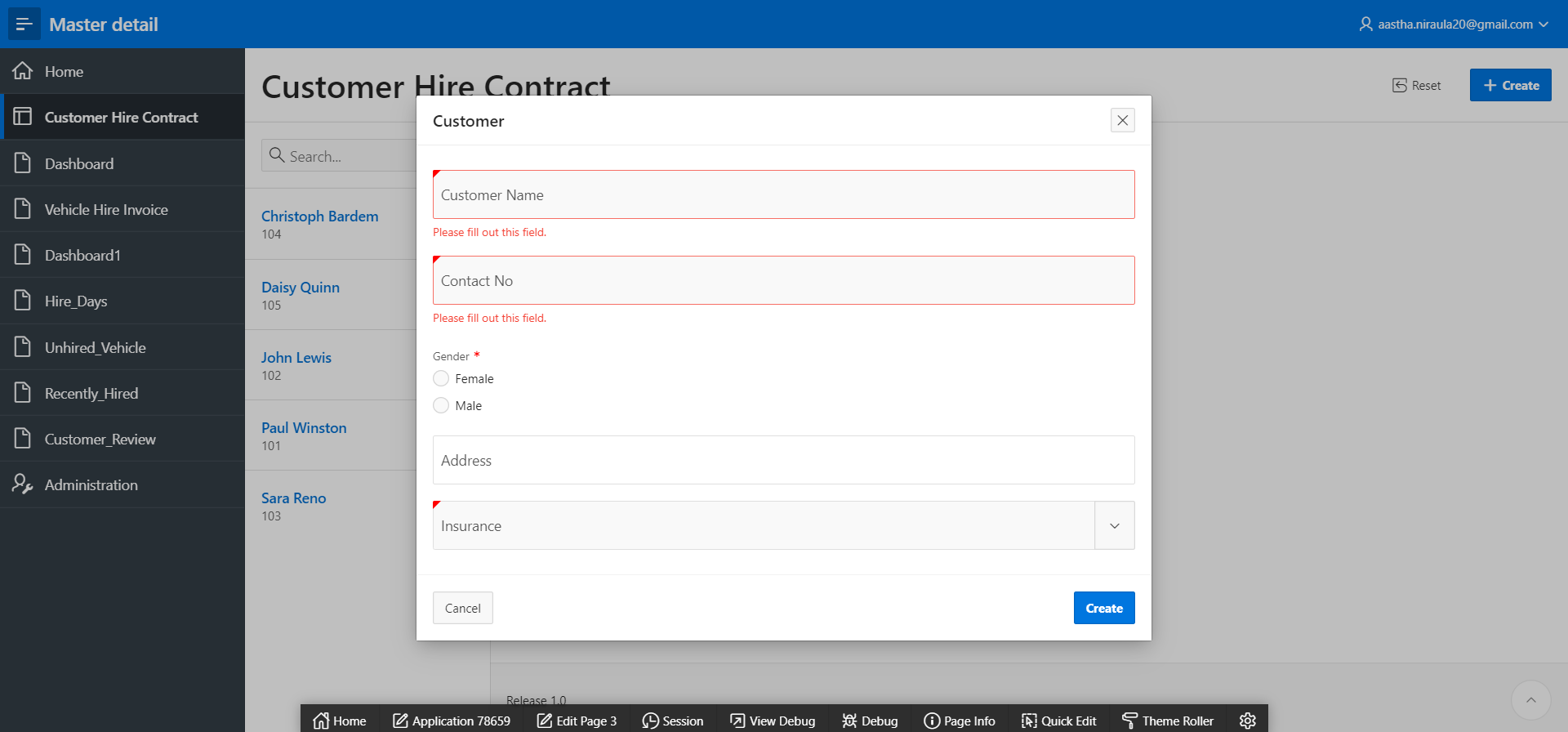


Contract number:



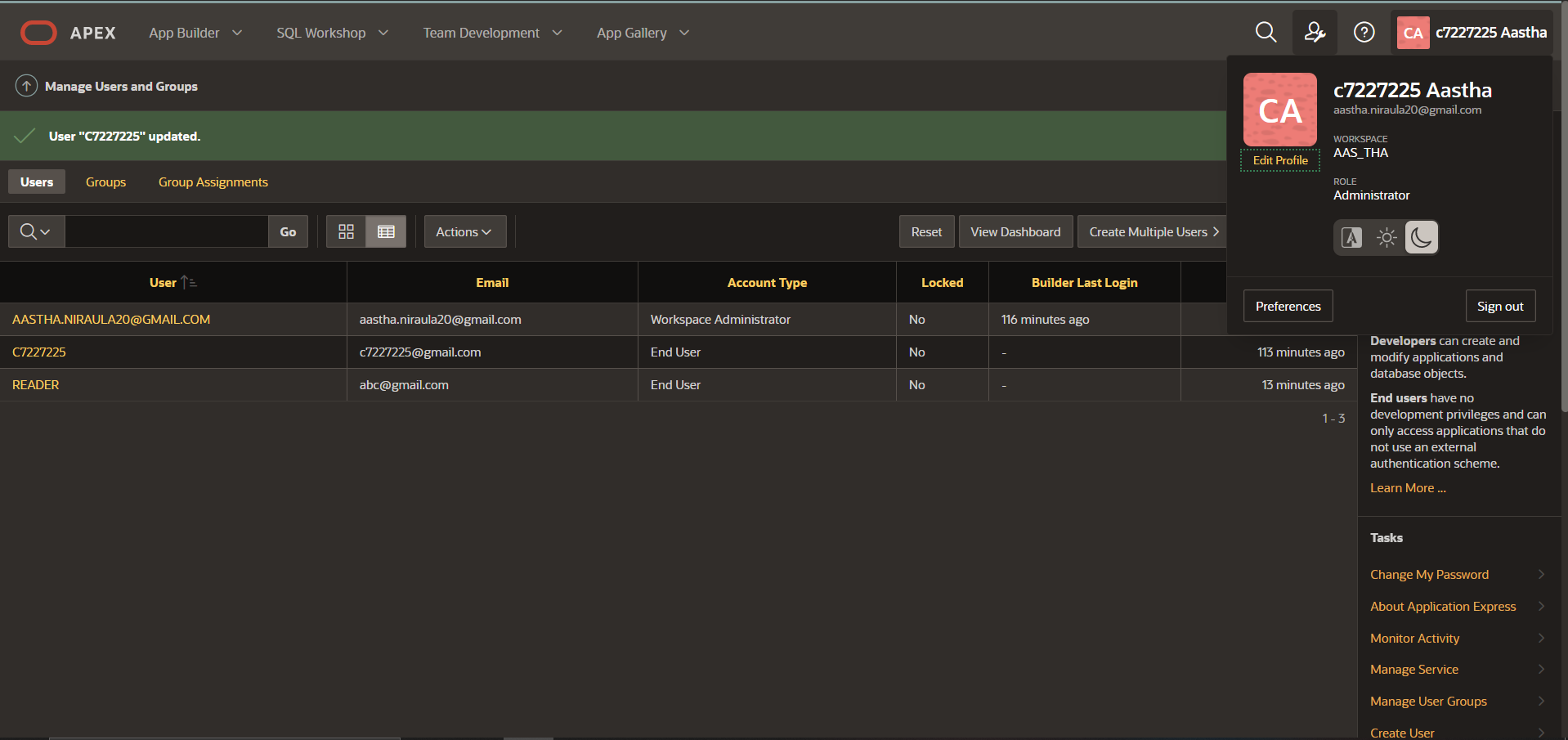


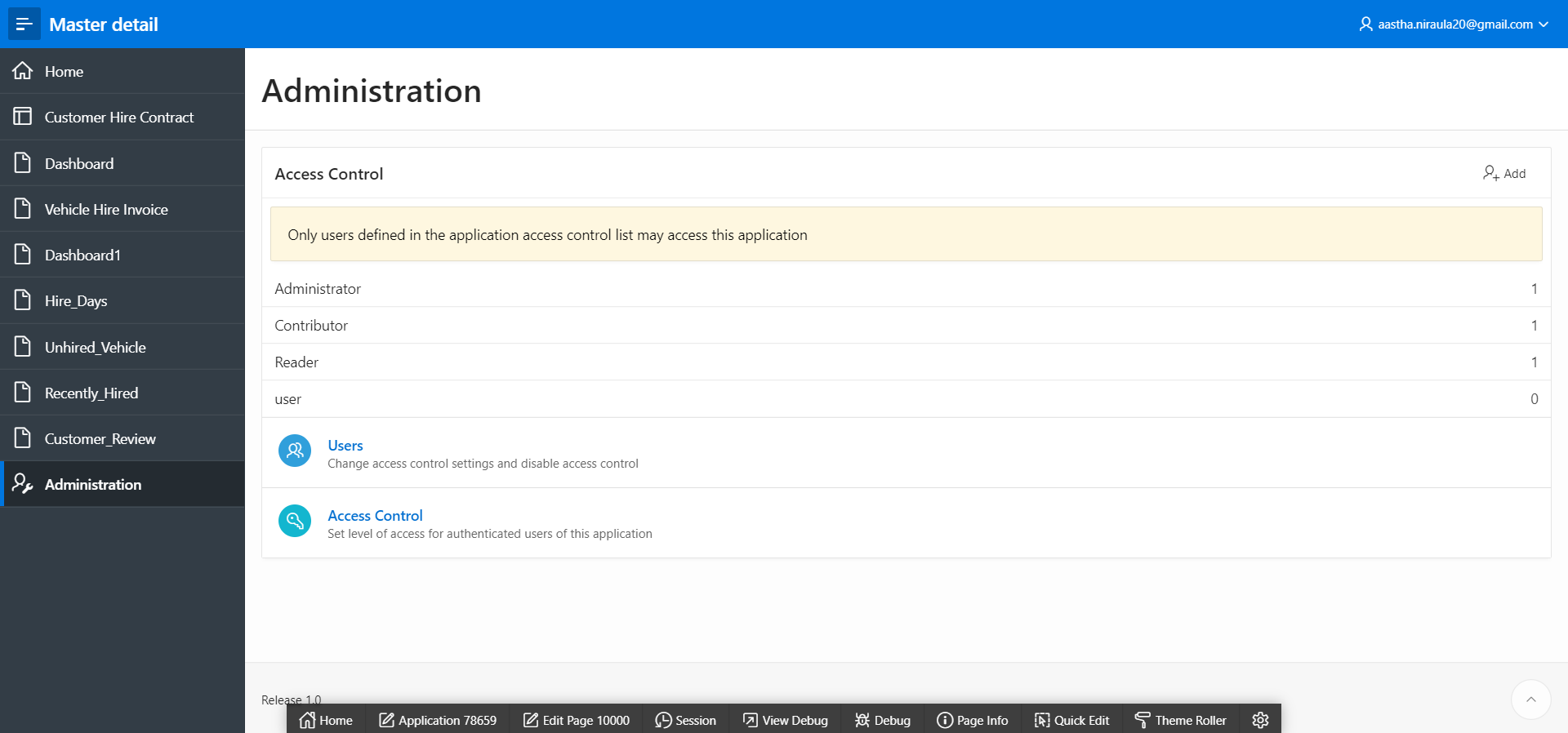
## 6. Required Fields



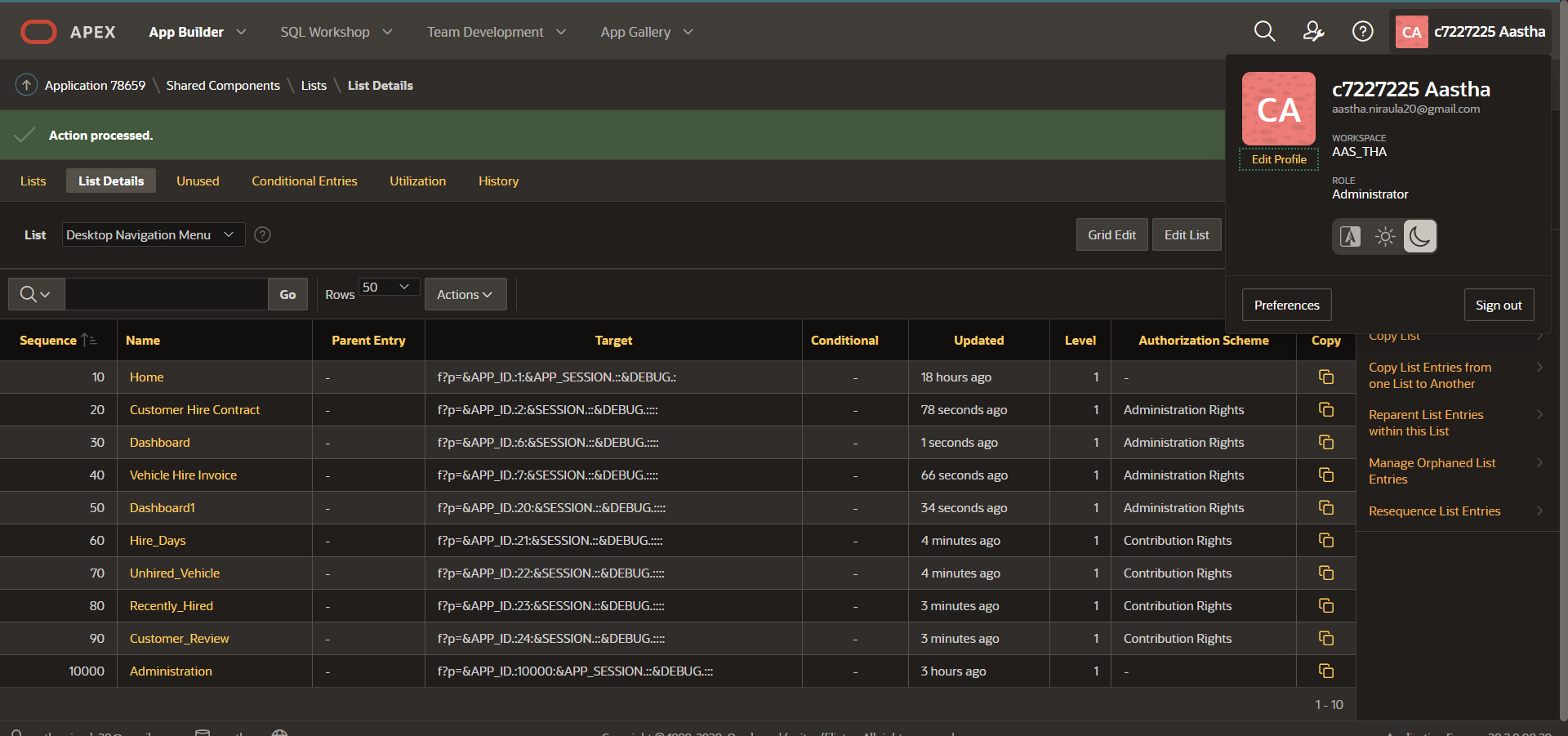
## 7. User login

### Creating Users

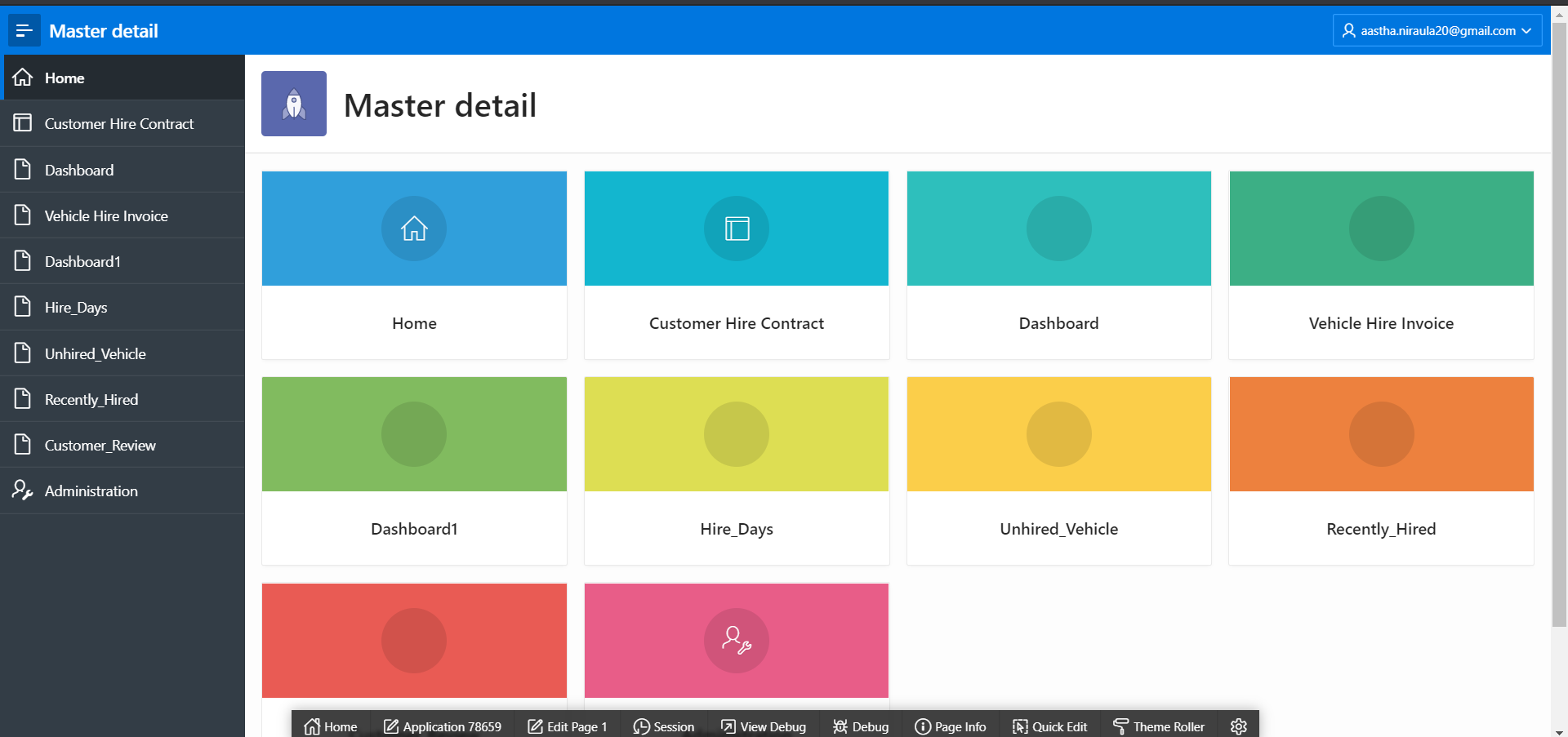




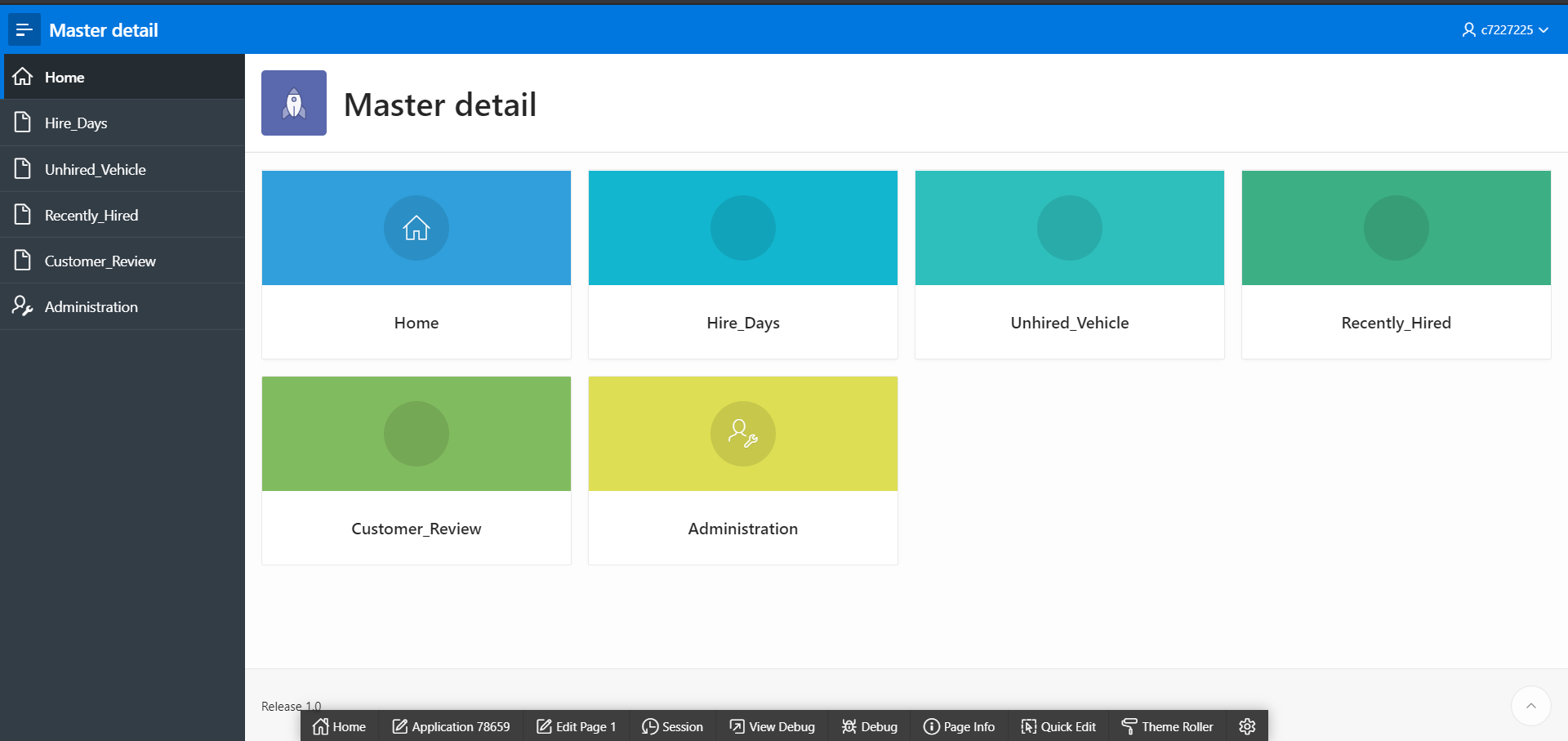
### Editing the Actions



#### For Administrator



#### For Contributor



# Bibliography

Connolly, Thomas M (2015). *Database systems: a practical approach to design, implementation and management.* Sixth Edition. England: Pearson Education Limited.

Oracle(2020) *Database Documentation* [Online]. Available at:

<https://docs.oracle.com/en/database/> (Accessed:29th December 2020)

# Appendixes

## TASK1:

### Trigger 1

DROP TABLE Customer\_tmp CASCADE CONSTRAINTS;

CREATE TABLE Customer\_tmp (

Customer\_ID NUMBER(10),

Customer\_Name VARCHAR(30) ,

Contact\_no NUMBER(20),

Gender CHAR(7),

Address VARCHAR(30),

a\_user VARCHAR2(30),

a\_date DATE,

a\_action VARCHAR2(30)

);

DROP TRIGGER Customer\_trigger;

CREATE OR REPLACE TRIGGER Customer\_trigger

AFTER INSERT OR DELETE OR UPDATE ON Customer

FOR EACH ROW

DECLARE

--variable for declaring update, delete or insert

vc\_trigger\_action VARCHAR2(30);

BEGIN

IF updating

THEN

-- incase of updating

vc\_trigger\_action := 'Data UPDATE';

ELSIF inserting

THEN

-- incase of inserting

vc\_trigger\_action := 'Data INSERT';

ELSIF deleting

THEN

-- incase of deleting

vc\_trigger\_action := 'Data DELETE';

END IF;

IF vc\_trigger\_action IN ('Data DELETE','Data UPDATE') THEN

-- if vc\_trg\_action is DELETE or UPDATE then insert old table values

INSERT INTO Customer\_tmp

(Customer\_ID,Customer\_Name,Contact\_no,Gender,Address,a\_user ,a\_date,a\_action)

VALUES

(:OLD.Customer\_ID, :OLD.Customer\_Name, :OLD.Contact\_no,:OLD.Gender,:OLD.Address,v('APP\_USER'), SYSDATE, vc\_trigger\_action);

ELSE

-- if vc\_trg\_action is INSERT then insert new table values

INSERT INTO Customer\_tmp

(Customer\_ID,Customer\_Name,Contact\_no,Gender,Address,a\_user ,a\_date,a\_action)

VALUES

(:NEW.Customer\_ID, :NEW.Customer\_Name, :NEW.Contact\_no,:NEW.Gender,:NEW.Address,UPPER(v('APP\_USER')), SYSDATE, vc\_trigger\_action);

END IF;

END customer\_trigger;

### Testing

INSERT INTO CUSTOMER VALUES (105,'Daisy Quinn',9835728190 ,'Female','Taplejung',502);

INSERT INTO CUSTOMER VALUES (145,'Unique Laun',9754278910,'Male','Thapathali',501);

UPDATE Customer

set customer\_name='Lucky'

where customer\_id=145;

DELETE FROM Customer

WHERE customer\_id=145;

SELECT \* FROM Customer\_tmp;

### Additional Trigger

CREATE TABLE insurance\_tmp as select \* from insurance;

CREATE TABLE customer\_tmp as select \* from customer;

CREATE OR REPLACE TRIGGER insurance\_null

AFTER DELETE OR UPDATE OF insurance\_id on insurance\_tmp

FOR EACH ROW

BEGIN

IF UPDATING AND :OLD.insurance\_id!= :NEW.insurance\_id THEN

UPDATE customer\_tmp SET insurance\_id=NULL

WHERE customer\_tmp.insurance\_id = :old.insurance\_id;

END IF;

END;

### Testing

UPDATE insurance\_tmp SET insurance\_id=503

WHERE insurance\_id=501;

SELECT \* FROM insurance\_tmp;

DELETE FROM insurance\_tmp

WHERE insurance\_id=502;

SELECT \* FROM insurance\_tmp;

### Trigger 2

CREATE TABLE Company\_holidays (Day DATE);

INSERT INTO Company\_holidays VALUES ('12/25/2020');

INSERT INTO Company\_holidays VALUES ('12/26/2020');

INSERT INTO Company\_holidays VALUES ('1/1/2021');

INSERT INTO Company\_holidays VALUES ('12/31/2020');

CREATE OR REPLACE TRIGGER Holidays\_trigger

BEFORE INSERT OR DELETE OR UPDATE ON Contract

DECLARE

holiday INTEGER;

Not\_on\_holidays EXCEPTION;

BEGIN

/\* Check for company holidays: \*/

SELECT COUNT(\*) INTO holiday FROM Company\_holidays

WHERE TRUNC(Day) = TRUNC(Sysdate); -- Discard time parts of dates

IF holiday > 0 THEN

RAISE Not\_on\_holidays;

END IF;

EXCEPTION

WHEN Not\_on\_holidays THEN

Raise\_application\_error(-20325,'Might not change '

||'employee table during a holiday');

END;

### Testing

INSERT INTO Contract values (901,'06/06/2020','12/31/2020',101,6784);

## TASK 2:

### Report1

SELECT c.Customer\_ID, c.Customer\_name, h.hire\_days

FROM Customer c, Hire h

WHERE c.customer\_id=h.customer\_id

AND hire\_days>1

ORDER BY Hire\_days;

### Report2

SELECT v.vehicle\_no, v.classification, v.model, v.charge, h.hire\_no

FROM vehicles v, hire h

WHERE h.hire\_no (+) = v.hire\_no

AND v.hire\_no is NULL;

### Report3

SELECT contract\_no, start\_date, customer\_id

FROM contract

WHERE (start\_date,customer\_id)

IN

(SELECT MAX(start\_date), customer\_id FROM contract

GROUP BY customer\_id);

### Report4

SELECT ct.customer\_id,c.customer\_name,c.contact\_no, ct.Contract\_no, ct.end\_date, add\_months(end\_date,2) review\_date

FROM contract ct INNER JOIN customer c

ON ct.customer\_id=c.customer\_id

ORDER BY add\_months(end\_date,2);

## SEQUENCE AND TRIGGERS

CREATE SEQUENCE Customer\_Id\_seq INCREMENT BY 1 START WITH 106;

CREATE OR REPLACE EDITIONABLE TRIGGER CustomerID\_trigger

BEFORE

insert on Customer

for each row

begin

if :NEW.CUSTOMER\_ID is null then

select Customer\_Id\_seq.nextval into :NEW.CUSTOMER\_ID from sys.dual;

end if;

end;

CREATE SEQUENCE Hire\_no\_seq INCREMENT BY 1 START WITH 8;

CREATE OR REPLACE EDITIONABLE TRIGGER Hireno\_trigger

BEFORE

insert on Hire

for each row

begin

if :NEW.Hire\_no is null then

select Hire\_no\_seq.nextval into :NEW.Hire\_no from sys.dual;

end if;

end;