Database Assignment

Submitted by Nishit Gala

# Contents

[Chapter 1 – Introduction 3](#_Toc69029364)

[Chapter 2 – ER Diagrams with Constraints and Assumptions 4](#_Toc69029365)

[Chapter 3 – Implementation 10](#_Toc69029366)

[Chapter 4 – Demonstration using SQL Queries 31](#_Toc69029367)

[Chapter 5 – Conclusion 45](#_Toc69029368)

# Chapter 1 – Introduction

This report demonstrates the day-to-day operations of a Wine-stopper Website that allows the customers to place online orders for Wine of their choice. Also, it will demonstrate the backend operations of a Distribution center stocking and ordering the wines from their respective warehouses when the stock needs to be replenished.

The Wine can be categorized as per Colour, Country, Sweetness Indicator. Additionally, there are various promotions on the offer with different types and delivery charge calculations. All of these factors are taken into consideration while designing the problem statement.

The objective of this report is to get a holistic view of Wine Stopper day-to day operations that includes following -

1. Customer adding Items into their shopping basket
2. Customer adding items into their shopping list
3. Customer placing an order to generate an invoice
4. Distribution center placing Standing Order for wine from the warehouse
5. Distribution center placing Special Order for wine from the warehouse
6. Promotions applied for shopping and its related calculations
7. Delivery charges calculation
8. Stock maintenance and updates at the distribution center
9. Setting the Expiry to the Shopping List (Wish List) and to clear it if it is more than 60 days

The first part of the report is mentioning about the ER Diagram that will be highlighted with the assumptions and constraints while designing the entity relationships.

This Diagram is classified into 3 main parts with corresponding entities designed as described below

|  |  |  |
| --- | --- | --- |
| Part |  | Entities (Tables) |
| Wine Entities | Main | WS\_WINE, WS\_WINE\_DETAILS |
| Supporting | WS\_WINE\_CATEGORY, WS\_WINE\_INDICATOR, WS\_COUNTRY |
| Customer Entities | Main | WS\_CUSTOMER, WS\_SHOPPING\_BASKET, WS\_SHOPPING\_LIST, WS\_INVOICE |
| Supporting | WS\_PROMOTION, WS\_CALCULATE\_DELIVERY\_CHARGE |
| Distribution Center Entities | Main | WS\_CENTER, WS\_WAREHOUSE, WS\_STOCK\_INFO, WS\_ORDER, WS\_ORDER\_DETAILS |
| Supporting | WS\_REGION, WS\_LOCATION, WS\_ORDER\_TYPE |

The second part consist of the Implementation that will include screenshots from Oracle Apex for DDL commands including CREATE, ALTER statements

The third part consist of the critical SQL’s that demonstrates the functionalities described above

The last part is the conclusion of the entire case study with critical evaluation, explaining high level flow and related assumptions

# Chapter 2 – ER Diagrams with Constraints and Assumptions

The ER diagram designed for the day-to-day operations of a Wine-Stopper Web page is as shown below

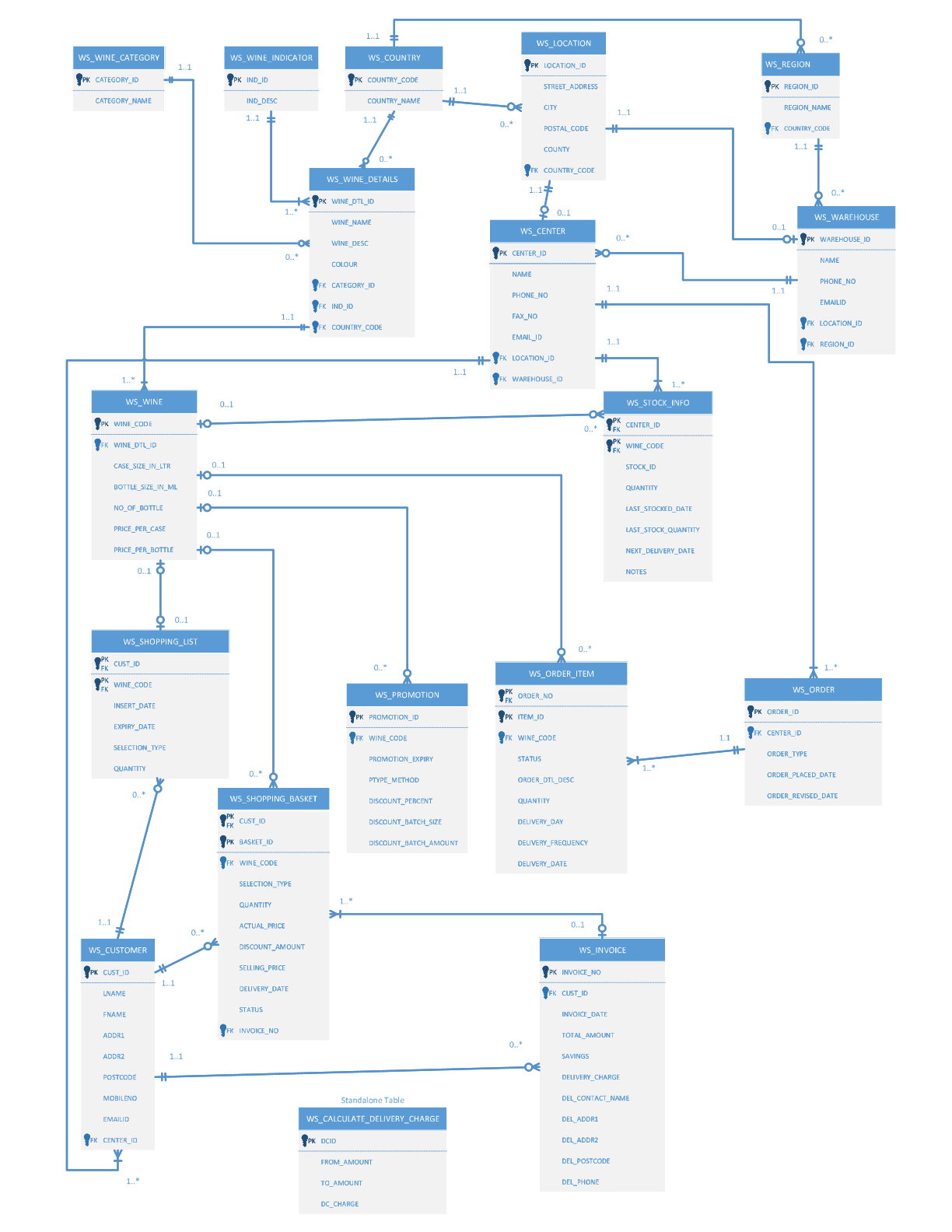
This is the complete ER Diagram as shown

Below is the PDF File attached for better viewing.



The screenshot of the ER Diagram is specified as below (next page)

***Note – For better visibility, please refer the PDF file attached above***

ER Diagram is as shown in the below image 

**Assumptions / Constraint Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Constraint Type / Assumption** | **Column Name** | **Comments** |
| **WS\_WINE\_DETAILS** | PK | WINE\_DTL\_ID | This table stores the information of Wine including Name, Description, Colour, Indicator, Country |
| Check | COLOUR | Colour in ('Red','White','Rose','Gold','Yellow','Orange') |
| FK, Assumption | CATEGORY\_ID | Wine is categorized into 3 types  1) Regular 2) Sparkling 3) Dessert  FK Reference - WS\_WINE\_CATEGORY(CATEGORY\_ID) |
| FK, Assumption | IND\_ID | Indicator of Sweetness of the Wine as below 1 - Very Dry, 2 - Medium Dry, 3 - Mild Dry 4 - Mild Sweet, 5 - Medium Sweet, 6 - Sweet  FK Reference - WS\_WINE\_INDICATOR(IND\_ID) |
| FK | COUNTRY\_CODE | FK Reference - WS\_COUNTRY(COUNTRY\_CODE) |
| **WS\_WINE** | PK | WINE\_CODE | One Wine (as per WINE\_DTL\_ID) can have 1 or more unique Wine Codes |
| FK | WINE\_DTL\_ID | This will reference the Wine Name that is available in multiple forms (wine\_codes) in the WINE table  FK Reference - WS\_WINE\_DETAILS(WINE\_DTL\_ID) |
| Assumption | CASE\_SIZE\_IN\_LTR | CASE Size is assumed to be in LTR |
| Assumption | BOTTLE\_SIZE\_IN\_ML | Bottle Size is assumed to be in ML |
| Auto Generated Column | NUM\_OF\_BOTTLE | auto calculated with formula - No Of Bottles = CASE\_SIZE\_IN\_LTR \* 1000 /BOTTLE\_SIZE\_IN\_ML |
| Check | PRICE\_PER\_CASE | PRICE\_PER\_CASE > 0 |
| Check | PRICE\_PER\_BOTTLE | PRICE\_PER\_BOTTLE > 0 |
| Check | CHK\_PRICE | PRICE\_PER\_BOTTLE < PRICE\_PER\_CASE |
| **WS\_PROMOTION** | Composite PK, Assumption | WINE\_CODE , START\_DATE,EXPIRY\_DATE,STATUS | Composite Primary Key to consider the promotion period and status of the promotion for the wine code Assumption - Promotion is on the Wine Codes and no generic promotions allowed as per design |
| Auto Generated Column | PROMOTION\_ID | Autogenerated column as the table has composite key |
| FK | WINE\_CODE | Reference - WS\_WINE(WINE\_CODE) |
| Check, Assumption | PTYPE\_METHOD | 3 methods taken into consideration 1) PRICE\_DISCOUNT - e.g. 15% OFF 2) QUANTITY\_DISCOUNT - e.g. Buy 4 For 3 3) REDUCED\_PRICE - e.g. Reduced Price 2.99 per bottle |
| Check | STATUS | Status in ('Active', 'Inactive', 'Expired') |
| Check | DISCOUNT\_PERCENT | DISCOUNT\_PERCENT BETWEEN 1 and 90 |
| Check | DISCOUNT\_QUANTITY\_BUY | DISCOUNT\_QUANTITY\_BUY > 0 |
| Check | DISCOUNT\_QUANTITY\_FOR | DISCOUNT\_QUANTITY\_FOR > 0 |
| Check | chck\_buy\_greater\_than\_for | DISCOUNT\_QUANTITY\_BUY > DISCOUNT\_QUANTITY\_FOR |
| Check | RED\_PRICE\_PER\_CASE | RED\_PRICE\_PER\_CASE > 0 |
| Check | RED\_PRICE\_PER\_BOTTLE | RED\_PRICE\_PER\_BOTTLE > 0 |
| Check | chck\_Reduced\_bottlePrice\_less\_CasePrice | RED\_PRICE\_PER\_BOTTLE < RED\_PRICE\_PER\_CASE |
| **WS\_CUSTOMER** | PK | CUST\_ID | Customer ID is the PK |
| Check | LNAME | Check that the Last Name contains only Alphabets REGEXP\_LIKE(LNAME,'^[[:alpha:]]+$') |
| Check | FNAME | Check that the First Name contains only AlphabetsREGEXP\_LIKE(FNAME,'^[[:alpha:]]+$') |
| Check | GENDER | GENDER in ('M','F','N') |
| Check | MOBILE\_NO | Only Numbers allowed REGEXP\_LIKE(MOBILE\_NO,'^[[:digit:]]+$') |
| Check | EMAIL\_ID | Email ID is in correct format  REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$' |
| FK | CENTER\_ID | A Center ID has to be allocated to Customer on registration using the Address information Reference - WS\_CENTER(CENTER\_ID) |
| **WS\_SHOPPING\_LIST** | FK | CUST\_ID | Reference - WS\_CUSTOMER(CUST\_ID) |
| FK | WINE\_CODE | Reference - WS\_WINE(WINE\_CODE) |
| Default | INSERT\_DATE | Default to Sysdate |
| Default | EXPIRY\_DATE | Default to Sysdate + 60 |
| Check | STATUS | STATUS IN ('WishList', 'Expired', 'UserDeleted') |
| Check | SELECTION\_TYPE | SELECTION\_TYPE in ('CASE','BOTTLE') |
| Composite PK | CUST\_ID, WINE\_CODE | Composite PK CUST\_ID, WINE\_CODE |
| **WS\_SHOPPING\_BASKET** | FK | CUST\_ID | Reference - WS\_CUSTOMER(CUST\_ID) |
| FK | WINE\_CODE | Reference - WS\_WINE(WINE\_CODE) |
| Check | SELECTION\_TYPE | SELECTION\_TYPE in ('CASE','BOTTLE') |
| Default, Assumption | DISCOUNT\_AMOUNT | Default to 0 Assumption -  When User proceeds to checkout and upon successful payment, the status wil change to Ordered and Discount calculations will be done using Promotion table and the Discount\_Amount column will be updated |
| Auto Generated Column | SELLING\_PRICE | ACTUAL\_PRICE - DISCOUNT\_AMOUNT |
| Check, Assumption | STATUS | Status IN ('Pending','Ordered','Shipped','Fullfilled') Assumptions:  1) When the User adds the item in the shopping basket, the status is Pending and Discount Amount is defaulted to 0 2) When User proceeds to checkout and upon successful payment, the status wil change to Ordered and Discount calculations will be done using Promotion table and the Discount\_Amount column will be updated |
| FK, Assumption | INVOICE\_NO | Reference - WS\_INVOICE(INVOICE\_NO )  Assumptions: Invoice No will be updated When User proceeds to checkout and upon successful payment |
| **WS\_CALCULATE\_DELIVERY\_CHARGE** | PK, Assumption | DC\_ID | Assumption - **This is a standlone table** and the value of the delivery charge will be picked up from this table and updated in WS\_INVOICE (DELIVERY\_CHARGE) |
| Check | FRM\_AMOUNT | FRM\_AMOUNT > 0 |
| Check | TO\_AMOUNT | TO\_AMOUNT > 0 |
| Check | chck\_ToAmt\_greater\_thn\_FrmAmt | TO\_AMOUNT  > FRM\_AMOUNT |
| **WS\_INVOICE** | PK | INVOICE\_NO |  |
| FK | CUST\_ID | Reference - WS\_CUSTOMER(CUST\_ID) |
| Check | DEL\_CONTACT\_NAME | Name contains only alphabets REGEXP\_LIKE(DEL\_CONTACT\_NAME,'^[[:alpha:]]+$') |
| Check | DEL\_PHONE | Phone no contains only digitsREGEXP\_LIKE(DEL\_PHONE,'^[[:digit:]]+$') |
| Assumption | TOTAL\_AMOUNT | TOTAL\_AMOUNT is sum of SELLING\_PRICE from WS\_INVOICE\_ITEM for all the items ordered |
| Assumption | SAVINGS | SAVINGS is sum of DISCOUNT\_AMOUNT from WS\_INVOICE\_ITEM for all the items ordered |
| Assumption | DELIVERY\_CHARGE | Will be updated on the basis of rules configured in CALCULATE\_DELIVERY\_CHARGE |
| **WS\_LOCATION** | PK | LOCATION\_ID |  |
| FK | COUNTRY\_CODE | REFERENCES WS\_COUNTRY(COUNTRY\_CODE) |
| **WS\_WAREHOUSE** | PK | WAREHOUSE\_ID |  |
| Check | EMAIL\_ID | Email ID is in correct format  REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$' |
| FK | LOCATION\_ID | REFERENCES WS\_LOCATION(LOCATION\_ID) |
| FK | REGION\_ID | REFERENCES WS\_REGION(REGION\_ID) |
| Check | PHONE\_NO | Phone no contains only digits REGEXP\_LIKE(PHONE\_NO,'^[[:digit:]]+$') |
| Check | FAX\_NO | Phone no contains only digits REGEXP\_LIKE(FAX\_NO,'^[[:digit:]]+$') |
| **WS\_CENTER** | PK | CENTER\_ID |  |
| Check | EMAIL\_ID | Email ID is in correct format  REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$' |
| FK | LOCATION\_ID | REFERENCES WS\_LOCATION(LOCATION\_ID) |
| FK | WAREHOUSE\_ID | REFERENCES WS\_WAREHOUSE(WAREHOUSE\_ID) |
| Check | PHONE\_NO | Phone no contains only digits REGEXP\_LIKE(PHONE\_NO,'^[[:digit:]]+$') |
| Check | FAX\_NO | Phone no contains only digits REGEXP\_LIKE(FAX\_NO,'^[[:digit:]]+$') |
| **WS\_STOCK\_INFO** | Auto Generated Column | STOCK\_ID | Table with Composite PK |
| FK | CENTER\_ID | REFERENCES WS\_CENTER(CENTER\_ID) |
| FK | WINE\_CODE | Reference - WS\_WINE(WINE\_CODE) |
| Composite PK | CENTER\_ID,WINE\_CODE |  |
| **WS\_ORDER** | PK | ORDER\_ID |  |
| FK | CENTER\_NO | REFERENCES WS\_CENTER(CENTER\_ID) |
| Check | ORDER\_TYPE | ORDER\_TYPE in ('ST','SP')) ST- Standing Order SP - Special order |
| **WS\_ORDER\_ITEM** | Auto Generated Column | ITEM\_ID |  |
| FK | ORDER\_ID | REFERENCES WS\_ORDER(ORDER\_ID) |
| FK | WINE\_CODE | Reference - WS\_WINE(WINE\_CODE) |
| Check | Status | Status - Active, Inactive |
| Check, Assumption | DELIVERY\_DAY | DELIVERY\_DAY IN ('Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday') Assumption - This field will be used for Standing Orders only |
| Check, Assumption | DELIVERY\_FREQUENCY | DELIVERY\_FREQUENCY IN ('Weekly','Fortnightly','Monthly','Quarterly') Assumption - This field will be used for Standing Order only |
| Assumption | DELIVERY\_DATE | Assumption - This field will be used for setting Delivery Date of Special Order |

# Chapter 3 – Implementation

1. **WINE TABLES**

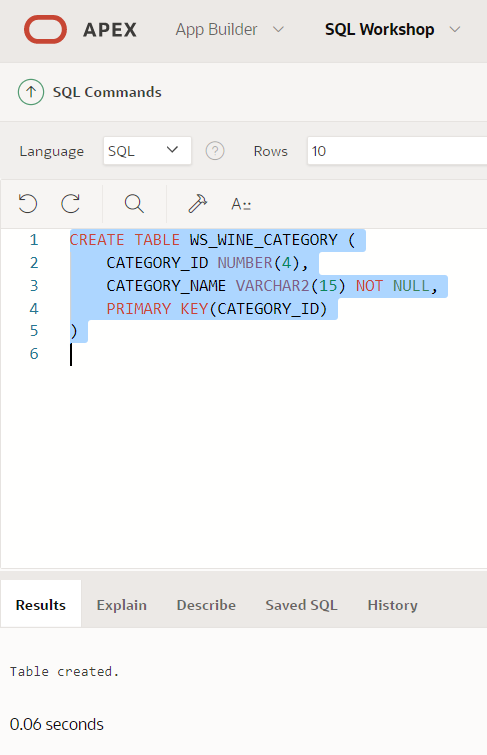
CREATE TABLE WS\_WINE\_CATEGORY (

    CATEGORY\_ID NUMBER(4),

    CATEGORY\_NAME VARCHAR2(15) NOT NULL,

    PRIMARY KEY(CATEGORY\_ID)

)



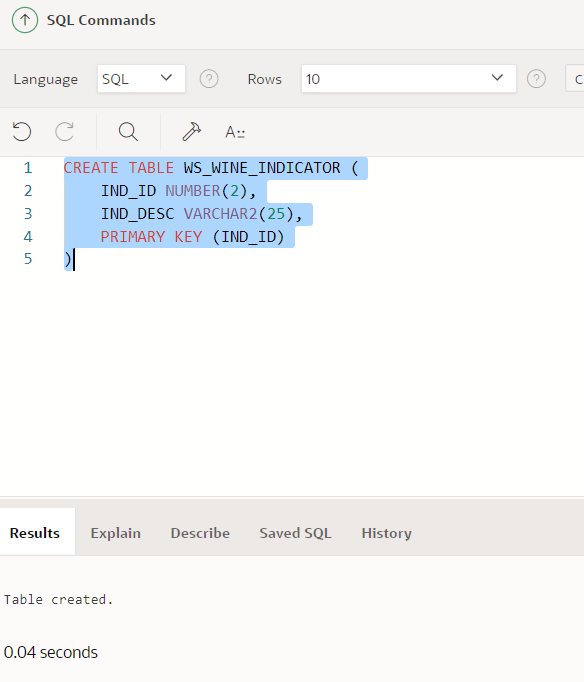
CREATE TABLE WS\_WINE\_INDICATOR (

    IND\_ID NUMBER(2),

    IND\_DESC VARCHAR2(25),

    PRIMARY KEY (IND\_ID)

)



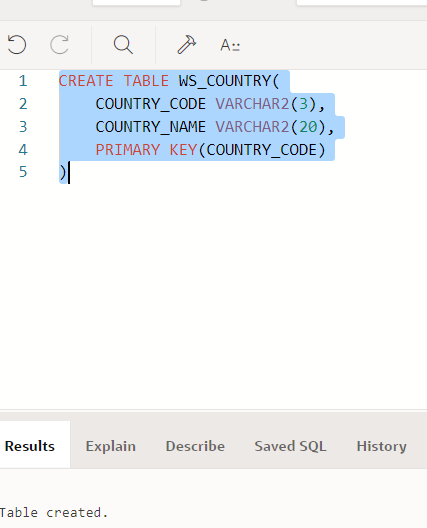
CREATE TABLE WS\_COUNTRY(

    COUNTRY\_CODE VARCHAR2(3),

    COUNTRY\_NAME VARCHAR2(20),

    PRIMARY KEY(COUNTRY\_CODE)

)



CREATE TABLE WS\_WINE\_DETAILS (

    WINE\_DTL\_ID NUMBER(9),

    WINE\_NAME VARCHAR2(20) NOT NULL,

    WINE\_DESC VARCHAR2(500) ,

    COLOUR VARCHAR2(10) CHECK(COLOUR in ('Red','White','Rose','Gold','Yellow','Orange') ),

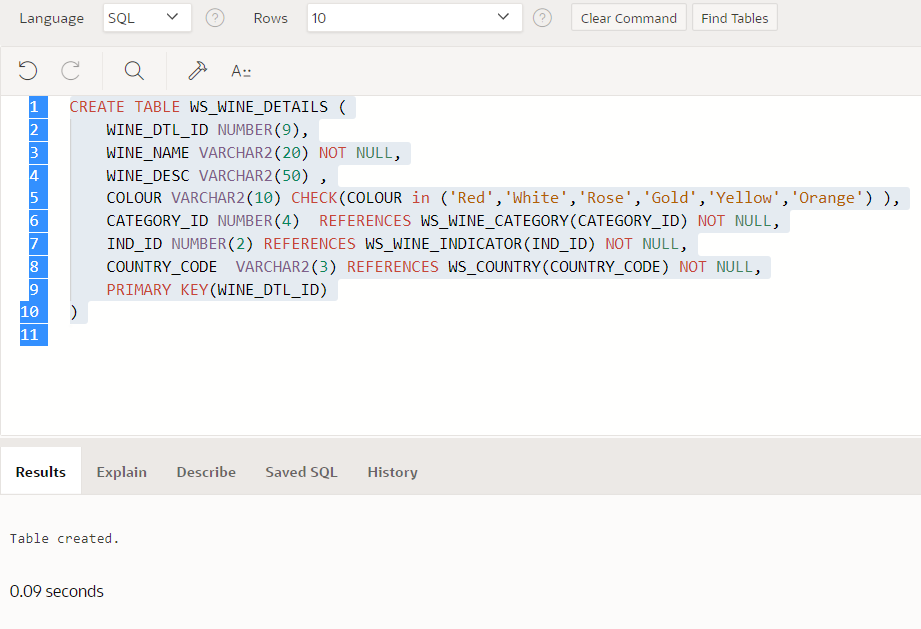
    CATEGORY\_ID NUMBER(4)  REFERENCES WS\_WINE\_CATEGORY(CATEGORY\_ID) NOT NULL,

    IND\_ID NUMBER(2) REFERENCES WS\_WINE\_INDICATOR(IND\_ID) NOT NULL,

    COUNTRY\_CODE  VARCHAR2(3) REFERENCES WS\_COUNTRY(COUNTRY\_CODE) NOT NULL,

    PRIMARY KEY(WINE\_DTL\_ID)

)



CREATE TABLE WS\_WINE (

    WINE\_CODE VARCHAR2(15) NOT NULL,

    WINE\_DTL\_ID NUMBER(9) REFERENCES WS\_WINE\_DETAILS(WINE\_DTL\_ID) NOT NULL,

    CASE\_SIZE\_IN\_LTR NUMBER(4,2) NOT NULL,

    BOTTLE\_SIZE\_IN\_ML NUMBER(4) NOT NULL,

    NUM\_OF\_BOTTLE NUMBER(4) GENERATED ALWAYS AS (round(CASE\_SIZE\_IN\_LTR \* 1000 /BOTTLE\_SIZE\_IN\_ML)) VIRTUAL,

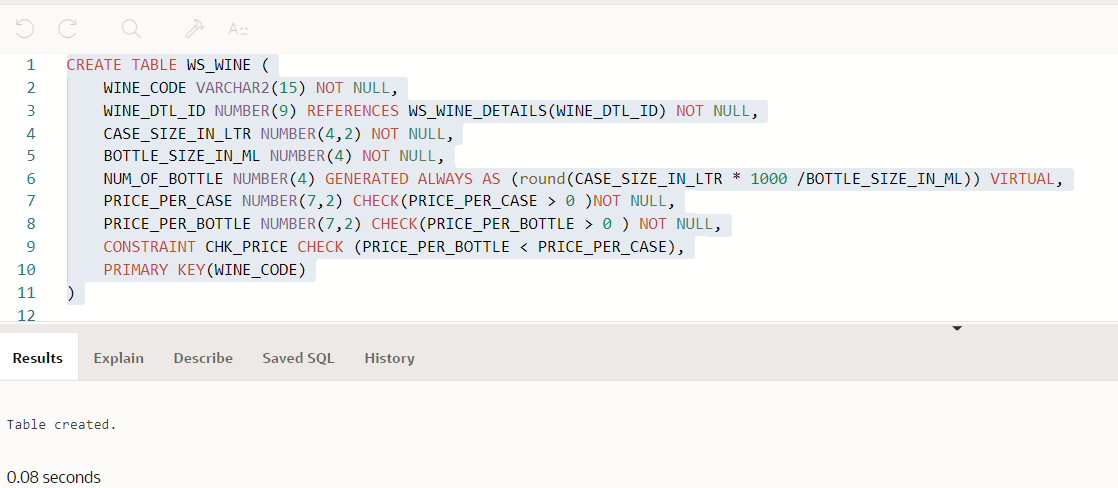
    PRICE\_PER\_CASE NUMBER(7,2) CHECK(PRICE\_PER\_CASE > 0 )NOT NULL,

    PRICE\_PER\_BOTTLE NUMBER(7,2) CHECK(PRICE\_PER\_BOTTLE > 0 ) NOT NULL,

    CONSTRAINT CHK\_PRICE CHECK (PRICE\_PER\_BOTTLE < PRICE\_PER\_CASE),

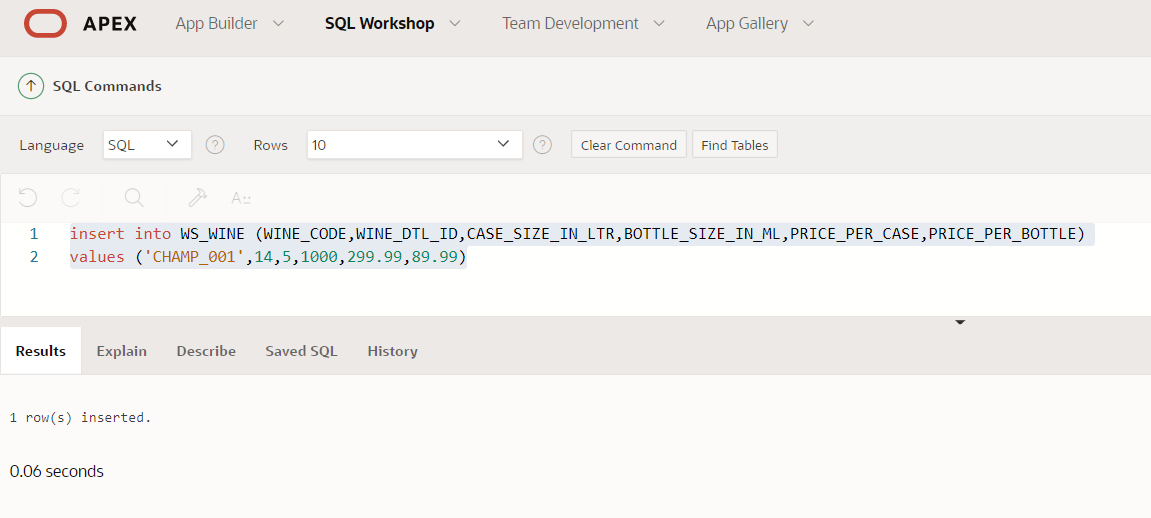
    PRIMARY KEY(WINE\_CODE)

)



TYPEWS\_WINE (WINE\_CODE,WINE\_DTL\_ID,CASE\_SIZE\_IN\_LTR,BOTTLE\_SIZE\_IN\_ML,PRICE\_PER\_CASE,PRICE\_PER\_BOTTLE)

values ('CHAMP\_001',14,5,1000,299.99,89.99)



create table WS\_PROMOTION (

    PROMOTION\_ID NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),

    WINE\_CODE VARCHAR2(15) REFERENCES WS\_WINE(WINE\_CODE) NOT NULL,

    PTYPE\_METHOD VARCHAR2(25) CHECK (PTYPE\_METHOD IN ('PRICE\_DISCOUNT','QUANTITY\_DISCOUNT')) NOT NULL,

    METHOD\_DESC VARCHAR2(100) NOT NULL,

    START\_DATE DATE NOT NULL,

    EXPIRY\_DATE  DATE NOT NULL,

    STATUS VARCHAR2(15) CHECK (STATUS IN ('Active','Inactive','Expired')) NOT NULL,

    DISCOUNT\_PERCENT NUMBER(2) CHECK (DISCOUNT\_PERCENT BETWEEN 1 and 90),

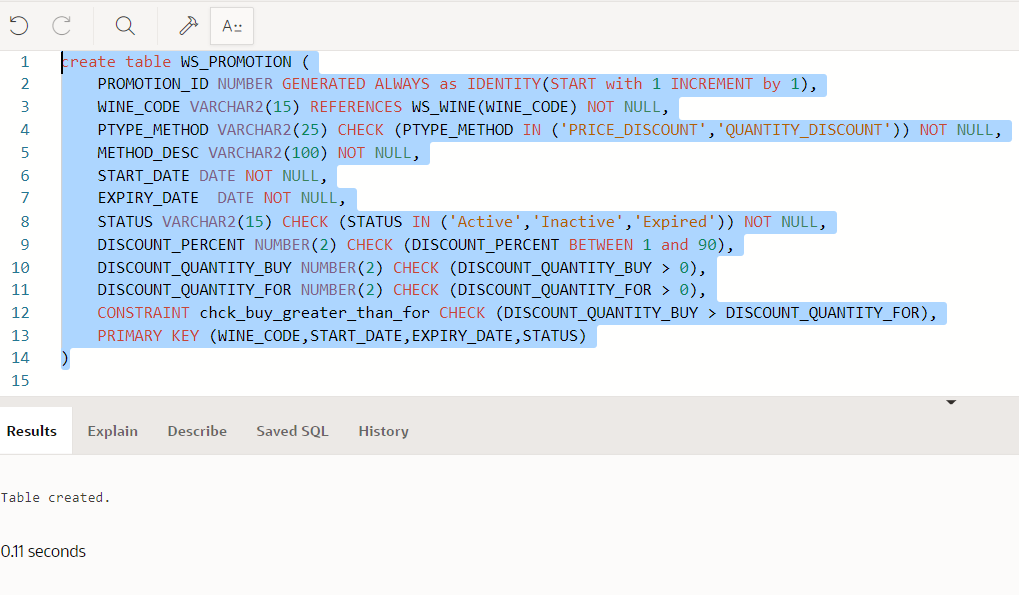
    DISCOUNT\_QUANTITY\_BUY NUMBER(2) CHECK (DISCOUNT\_QUANTITY\_BUY > 0),

    DISCOUNT\_QUANTITY\_FOR NUMBER(2) CHECK (DISCOUNT\_QUANTITY\_FOR > 0),

    CONSTRAINT chck\_buy\_greater\_than\_for CHECK (DISCOUNT\_QUANTITY\_BUY > DISCOUNT\_QUANTITY\_FOR),

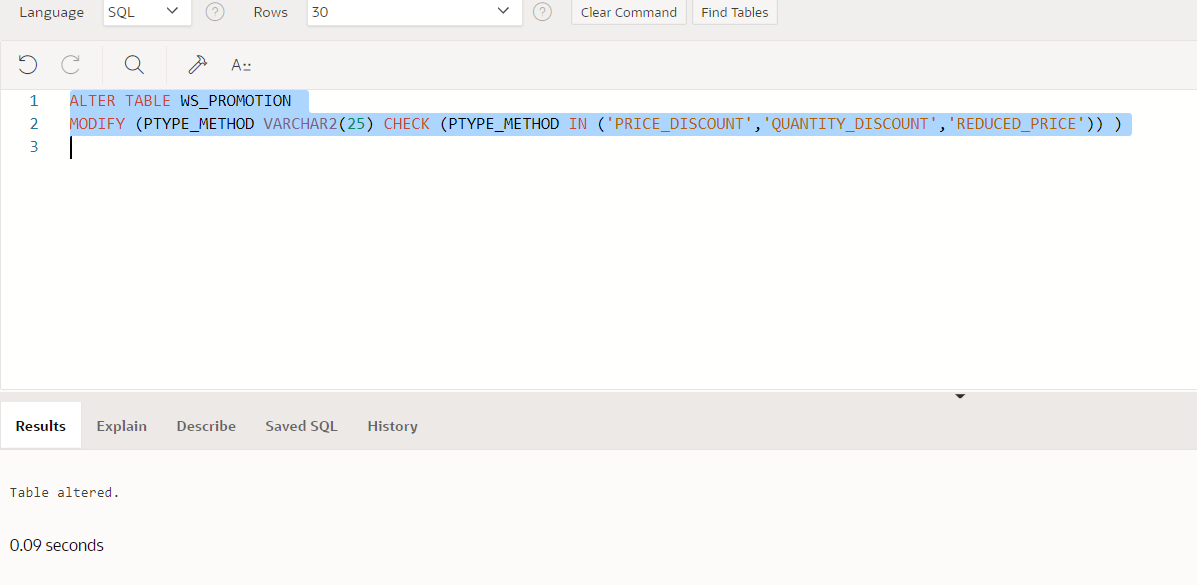
    PRIMARY KEY (WINE\_CODE,START\_DATE,EXPIRY\_DATE,STATUS)

)



ALTER TABLE WS\_PROMOTION

MODIFY (PTYPE\_METHOD VARCHAR2(25) CHECK (PTYPE\_METHOD IN ('PRICE\_DISCOUNT','QUANTITY\_DISCOUNT','REDUCED\_PRICE')) )



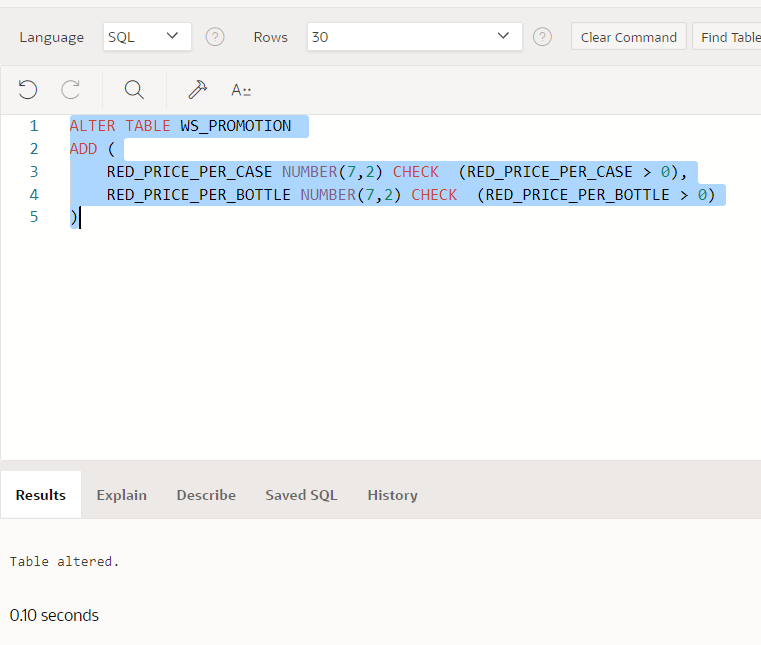
ALTER TABLE WS\_PROMOTION

ADD (

    RED\_PRICE\_PER\_CASE NUMBER(7,2) CHECK  (RED\_PRICE\_PER\_CASE > 0),

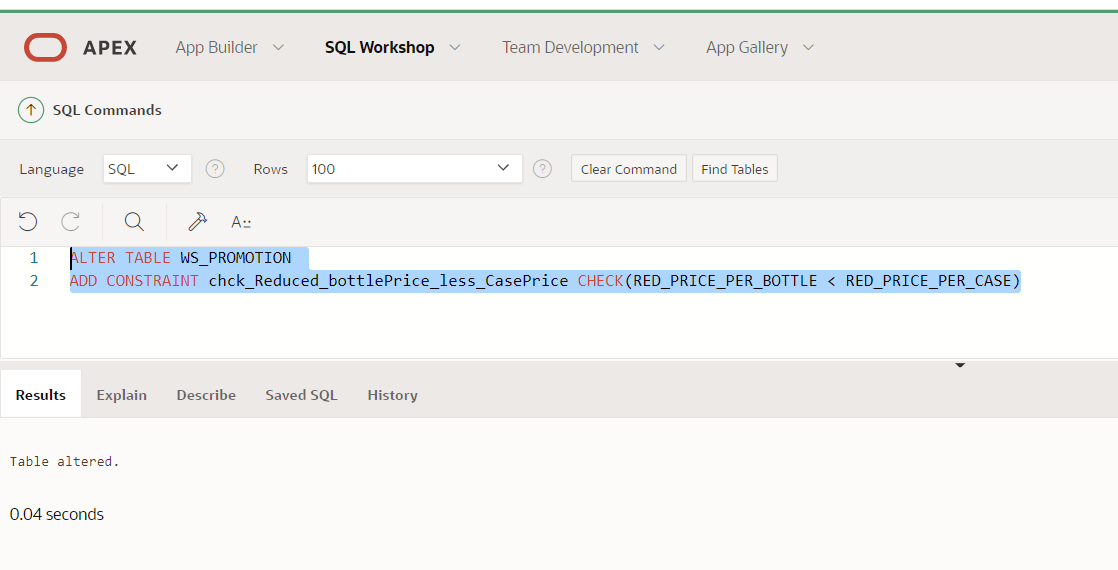
    RED\_PRICE\_PER\_BOTTLE NUMBER(7,2) CHECK  (RED\_PRICE\_PER\_BOTTLE > 0)

)



ALTER TABLE WS\_PROMOTION

ADD CONSTRAINT chck\_Reduced\_bottlePrice\_less\_CasePrice CHECK(RED\_PRICE\_PER\_BOTTLE < RED\_PRICE\_PER\_CASE)



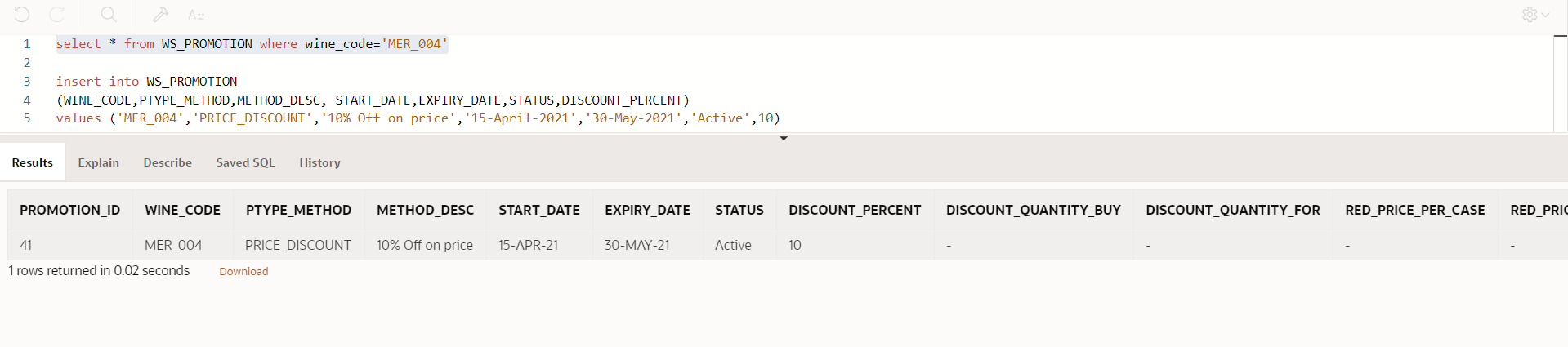
Record Insert:

1. Inserting a PRICE\_DISCOUNT (10% OFF)

insert into WS\_PROMOTION

(WINE\_CODE,PTYPE\_METHOD,METHOD\_DESC, START\_DATE,EXPIRY\_DATE,STATUS,DISCOUNT\_PERCENT)

values ('MER\_004','PRICE\_DISCOUNT','10% Off on price','15-April-2021','30-May-2021','Active',10)

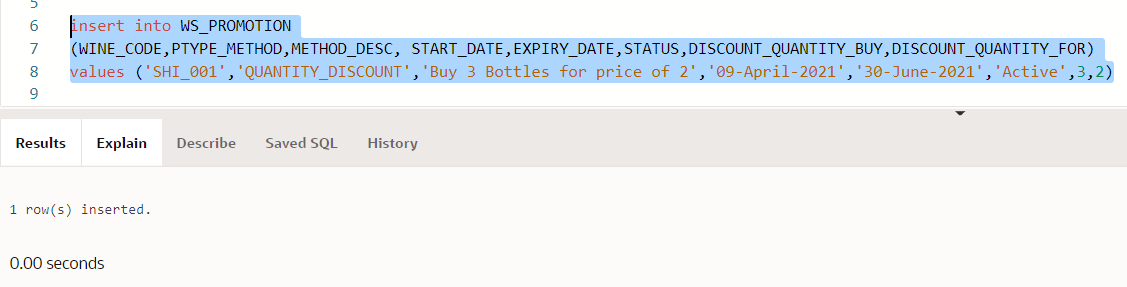


1. Inserting a QUANTITY\_DISCOUNT ( Buy 3 for price of 2)

insert into WS\_PROMOTION

(WINE\_CODE,PTYPE\_METHOD,METHOD\_DESC, START\_DATE,EXPIRY\_DATE,STATUS,DISCOUNT\_QUANTITY\_BUY,DISCOUNT\_QUANTITY\_FOR)

values ('SHI\_001','QUANTITY\_DISCOUNT','Buy 3 Bottles for price of 2','09-April-2021','30-June-2021','Active',3,2)

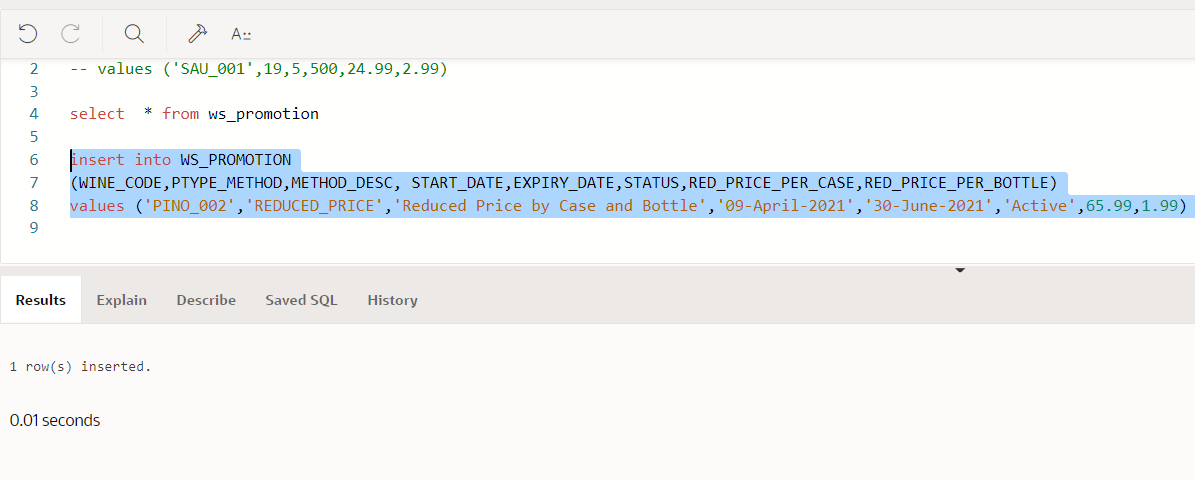


1. Inserting a Reduced price discount

insert into WS\_PROMOTION

(WINE\_CODE,PTYPE\_METHOD,METHOD\_DESC, START\_DATE,EXPIRY\_DATE,STATUS,RED\_PRICE\_PER\_CASE,RED\_PRICE\_PER\_BOTTLE)

values ('PINO\_002','REDUCED\_PRICE','Reduced Price by Case and Bottle','09-April-2021','30-June-2021','Active',65.99,1.99)



1. **CUSTOMER TABLE**

CREATE TABLE  WS\_CUSTOMER

   (

    CUST\_ID NUMBER NOT NULL,

    LNAME VARCHAR2(50) CHECK(REGEXP\_LIKE(LNAME,'^[[:alpha:]]+$')) NOT NULL,

    FNAME VARCHAR2(50) CHECK(REGEXP\_LIKE(FNAME,'^[[:alpha:]]+$')) NOT NULL,

    GENDER VARCHAR2(1) CHECK(GENDER in ('M','F','N')) NOT NULL,

    ADDR1 VARCHAR2(50) NOT NULL,

    ADDR2 VARCHAR2(50) NOT NULL,

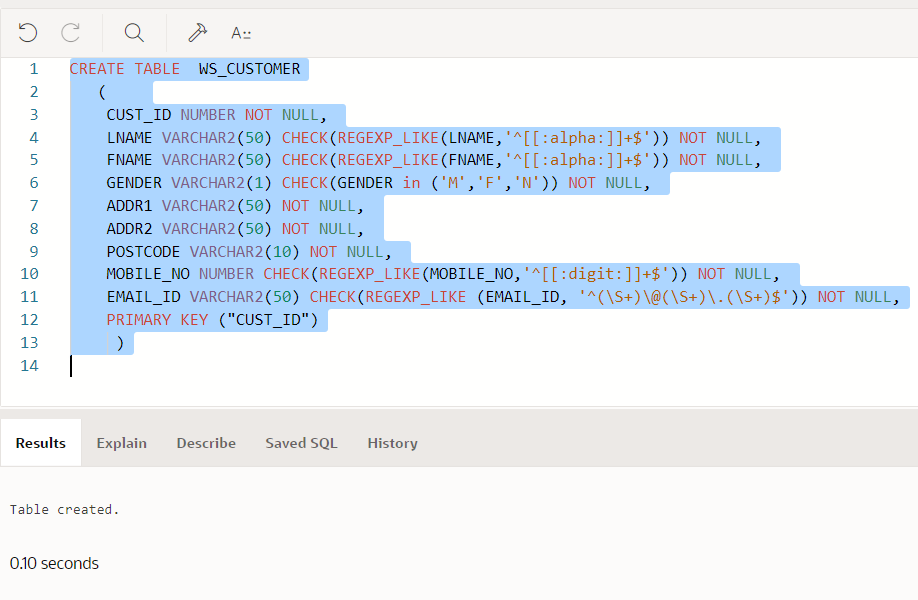
    POSTCODE VARCHAR2(10) NOT NULL,

    MOBILE\_NO NUMBER CHECK(REGEXP\_LIKE(MOBILE\_NO,'^[[:digit:]]+$')) NOT NULL,

    EMAIL\_ID VARCHAR2(50) CHECK(REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$')) NOT NULL,

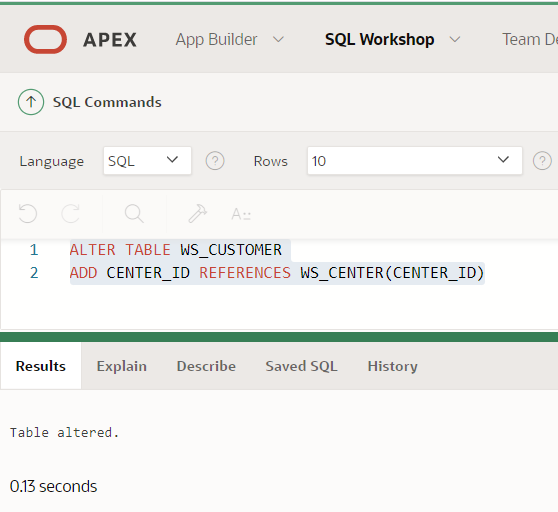
    PRIMARY KEY ("CUST\_ID")

     )



ALTER TABLE WS\_CUSTOMER

ADD CENTER\_ID REFERENCES WS\_CENTER(CENTER\_ID)



CREATE TABLE WS\_SHOPPING\_LIST

(

    CUST\_ID NUMBER REFERENCES WS\_CUSTOMER(CUST\_ID) NOT NULL,

    WINE\_CODE VARCHAR2(15) REFERENCES WS\_WINE(WINE\_CODE) NOT NULL,

    INSERT\_DATE DATE DEFAULT SYSDATE ,

    EXPIRY\_DATE DATE DEFAULT SYSDATE+60,

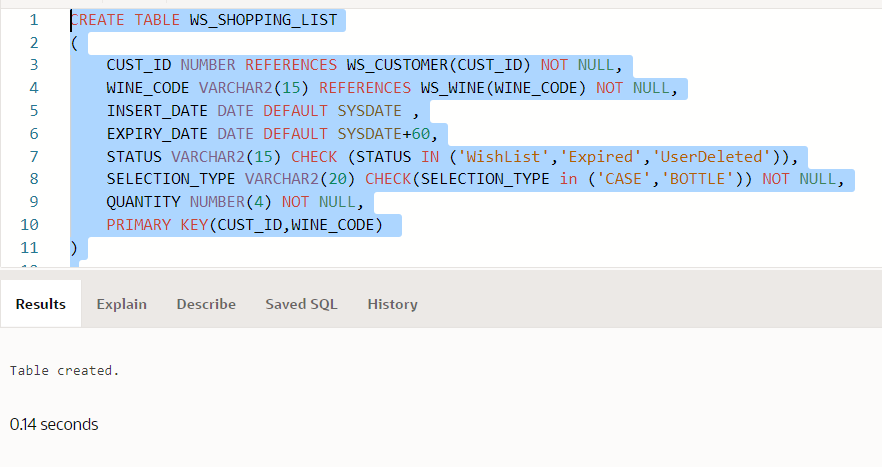
    STATUS VARCHAR2(15) CHECK (STATUS IN ('WishList','Expired','UserDeleted')),

    SELECTION\_TYPE VARCHAR2(20) CHECK(SELECTION\_TYPE in ('CASE','BOTTLE')) NOT NULL,

    QUANTITY NUMBER(4) NOT NULL,

    PRIMARY KEY(CUST\_ID,WINE\_CODE)

)



CREATE TABLE WS\_SHOPPING\_BASKET

(

    CUST\_ID NUMBER REFERENCES WS\_CUSTOMER(CUST\_ID) NOT NULL,

    BASKET\_ID NUMBER NOT NULL,

    WINE\_CODE VARCHAR2(15) REFERENCES WS\_WINE(WINE\_CODE) NOT NULL,

    SELECTION\_TYPE VARCHAR2(20) CHECK(SELECTION\_TYPE in ('CASE','BOTTLE')) NOT NULL,

    QUANTITY NUMBER(4) NOT NULL,

    ACTUAL\_PRICE NUMBER(7,2) NOT NULL,

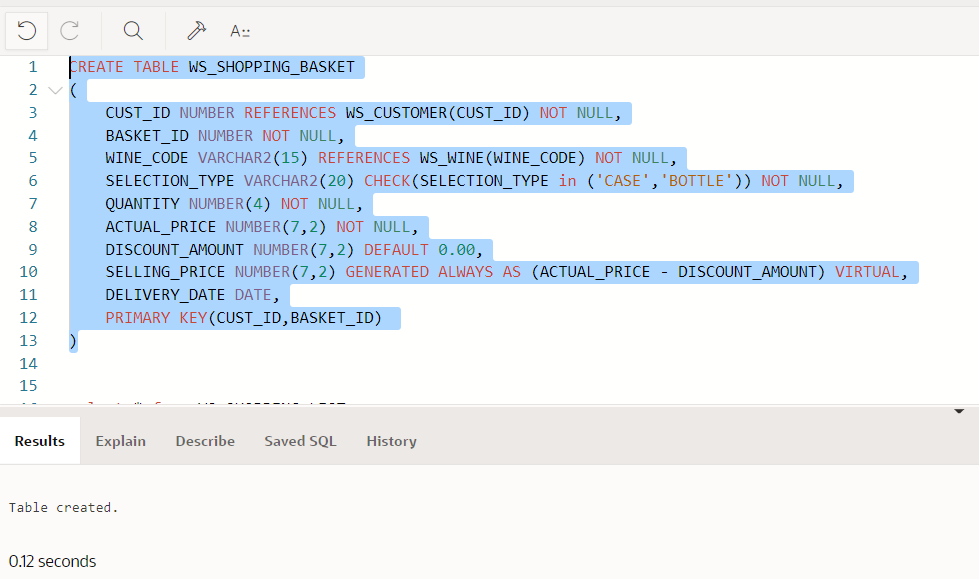
    DISCOUNT\_AMOUNT NUMBER(7,2) DEFAULT 0.00,

    SELLING\_PRICE NUMBER(7,2) GENERATED ALWAYS AS (ACTUAL\_PRICE - DISCOUNT\_AMOUNT) VIRTUAL,

    DELIVERY\_DATE DATE,

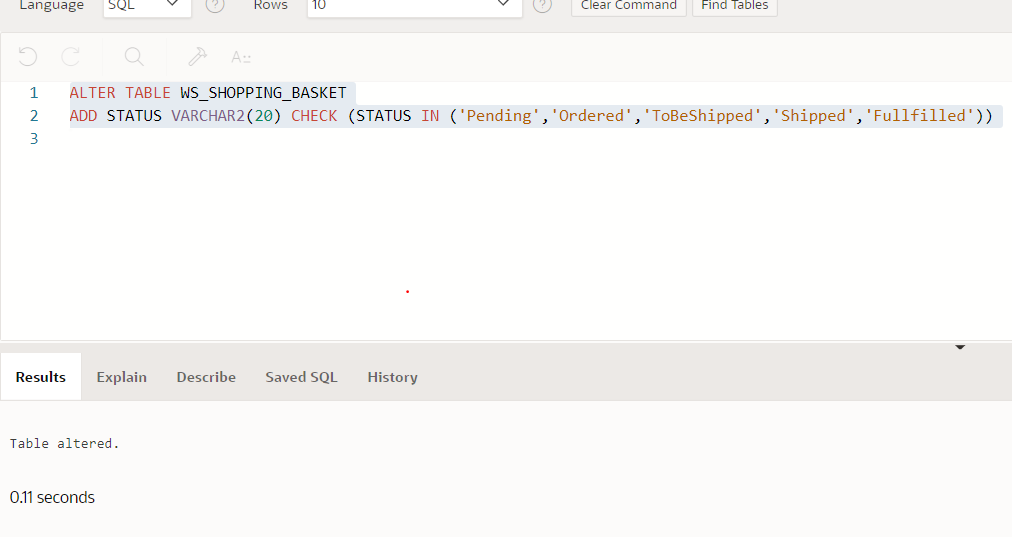
PRIMARY KEY(CUST\_ID,BASKET\_ID)

)



ALTER TABLE WS\_SHOPPING\_BASKET

ADD STATUS VARCHAR2(20) CHECK (STATUS IN ('Pending','Ordered','ToBeShipped','Shipped','Fullfilled'))

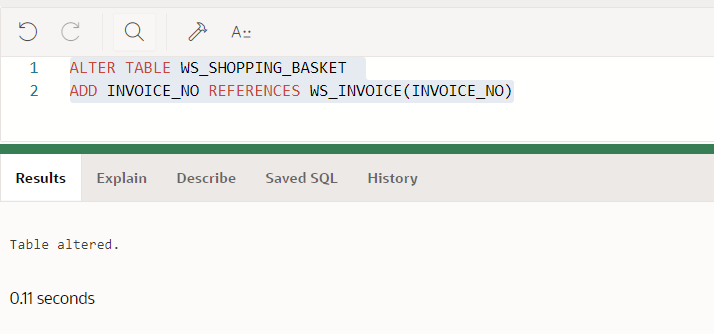


ALTER TABLE WS\_SHOPPING\_BASKET

MODIFY STATUS VARCHAR2(20) CHECK (STATUS IN ('Pending','Ordered','Shipped','Fullfilled'))

ALTER TABLE WS\_SHOPPING\_BASKET

ADD INVOICE\_NO REFERENCES WS\_INVOICE(INVOICE\_NO)



create table CALCULATE\_DELIVERY\_CHARGE(

    DC\_ID NUMBER NOT NULL,

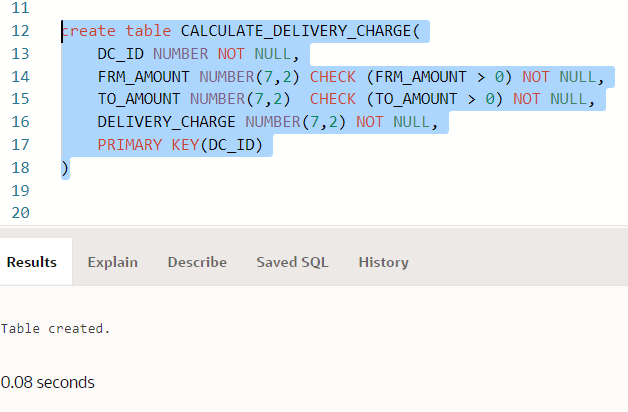
    FRM\_AMOUNT NUMBER(7,2) CHECK (FRM\_AMOUNT > 0) NOT NULL,

    TO\_AMOUNT NUMBER(7,2)  CHECK (TO\_AMOUNT > 0) NOT NULL,

    DELIVERY\_CHARGE NUMBER(7,2) NOT NULL,

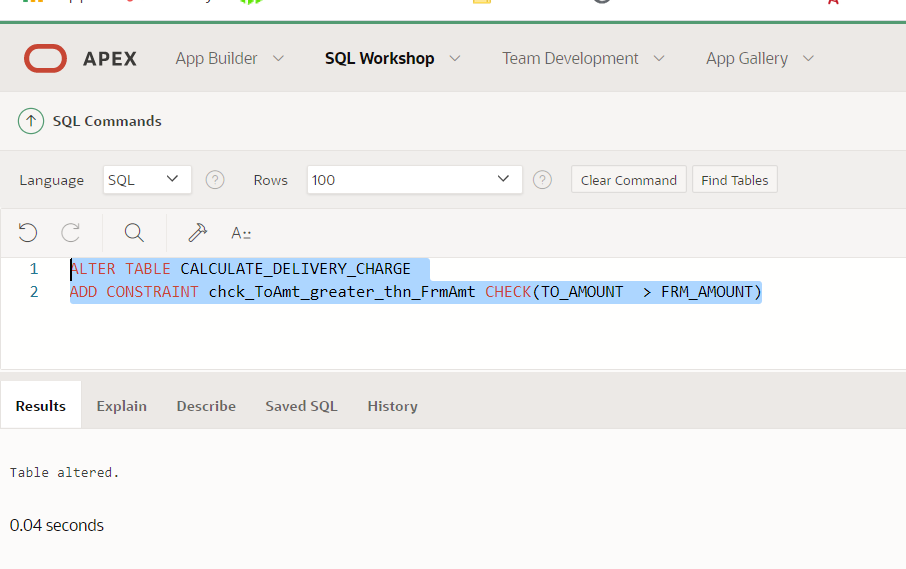
    PRIMARY KEY(DC\_ID)

)



ALTER TABLE CALCULATE\_DELIVERY\_CHARGE

ADD CONSTRAINT chck\_ToAmt\_greater\_thn\_FrmAmt CHECK(TO\_AMOUNT  > FRM\_AMOUNT)



create table WS\_INVOICE(

    INVOICE\_NO VARCHAR2(25) NOT NULL,

    CUST\_ID NUMBER REFERENCES WS\_CUSTOMER(CUST\_ID) NOT NULL,

    INVOICE\_DATE DATE NOT NULL,

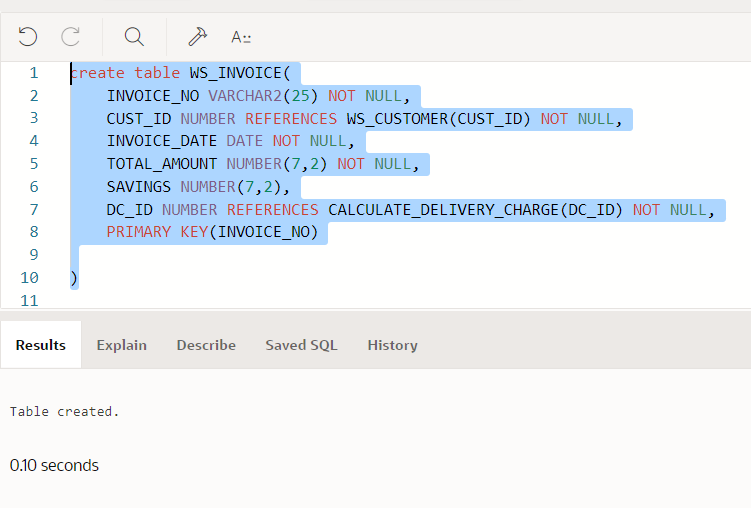
    TOTAL\_AMOUNT NUMBER(7,2) NOT NULL,

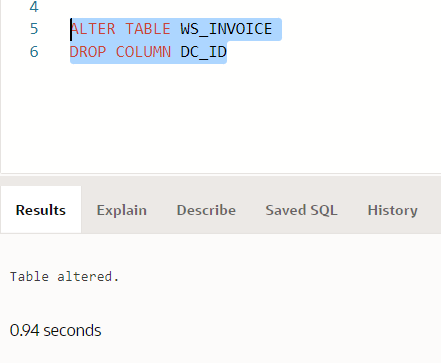
    SAVINGS NUMBER(7,2),

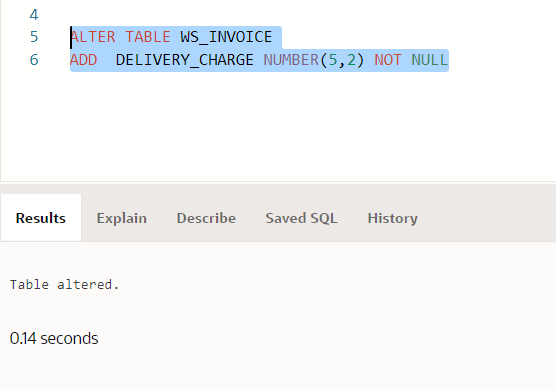
    DC\_ID NUMBER REFERENCES CALCULATE\_DELIVERY\_CHARGE(DC\_ID) NOT NULL,

    PRIMARY KEY(INVOICE\_NO)

)







ALTER TABLE WS\_INVOICE

ADD

(

 DEL\_CONTACT\_NAME VARCHAR2(50) CHECK(REGEXP\_LIKE(DEL\_CONTACT\_NAME,'^[[:alpha:]]+$')) NOT NULL,

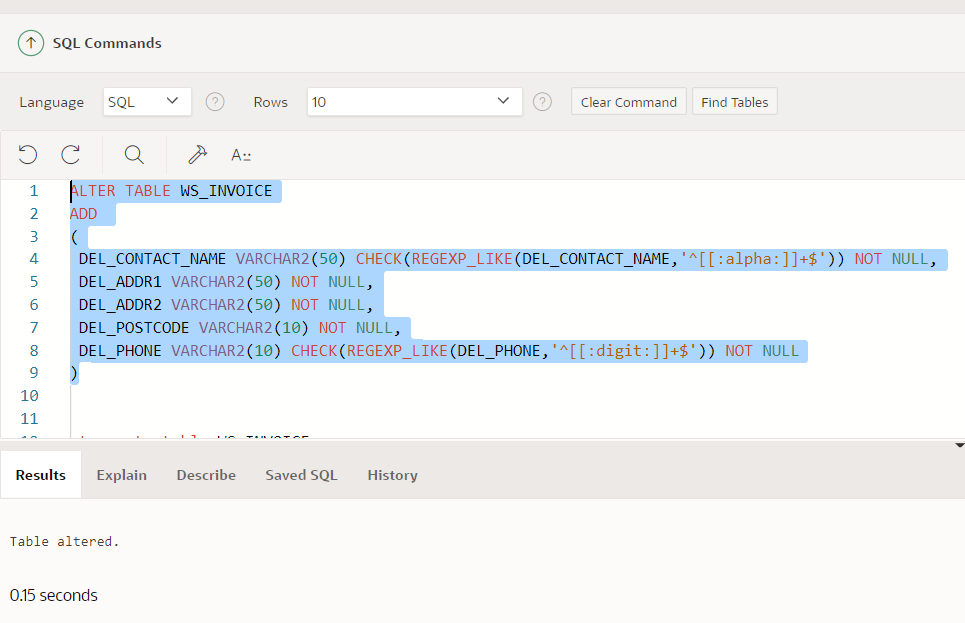
 DEL\_ADDR1 VARCHAR2(50) NOT NULL,

 DEL\_ADDR2 VARCHAR2(50) NOT NULL,

 DEL\_POSTCODE VARCHAR2(10) NOT NULL,

 DEL\_PHONE VARCHAR2(10) CHECK(REGEXP\_LIKE(DEL\_PHONE,'^[[:digit:]]+$')) NOT NULL

)



1. **WARESHOUSE and DISTRIBUTION CENTER TABLES**

WS\_REGION

CREATE TABLE WS\_REGION (

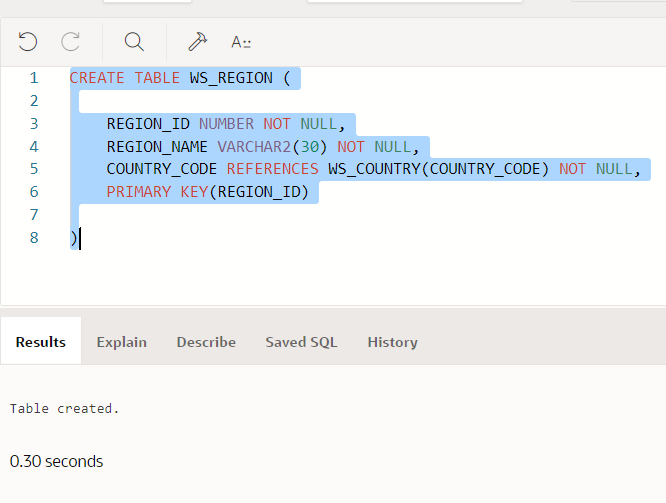
    REGION\_ID NUMBER NOT NULL,

    REGION\_NAME VARCHAR2(30) NOT NULL,

    COUNTRY\_CODE REFERENCES WS\_COUNTRY(COUNTRY\_CODE) NOT NULL,

    PRIMARY KEY(REGION\_ID)

)



CREATE TABLE WS\_LOCATION (

    LOCATION\_ID NUMBER NOT NULL,

    STREET\_ADDRESS VARCHAR2(50) NOT NULL,

    POSTAL\_CODE VARCHAR2(10) NOT NULL,

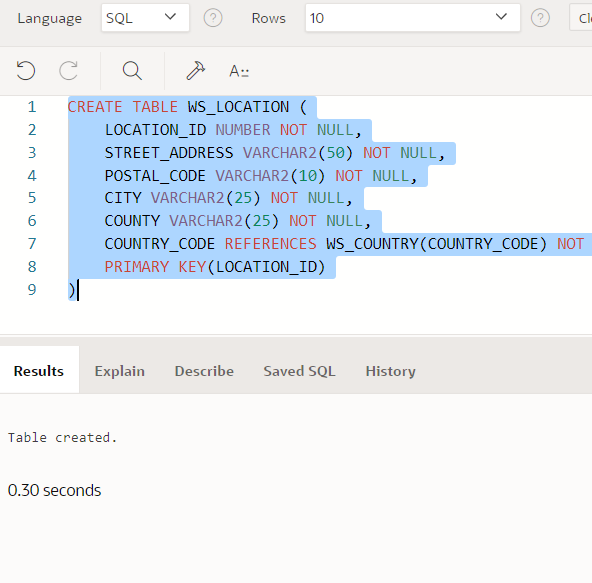
    CITY VARCHAR2(25) NOT NULL,

    COUNTY VARCHAR2(25) NOT NULL,

    COUNTRY\_CODE REFERENCES WS\_COUNTRY(COUNTRY\_CODE) NOT NULL,

    PRIMARY KEY(LOCATION\_ID)

)



CREATE TABLE WS\_WAREHOUSE (

    WAREHOUSE\_ID NUMBER NOT NULL,

    WAREHOUSE\_NAME VARCHAR2(50) NOT NULL,

    PHONE\_NO VARCHAR2(10) NOT NULL,

    FAX\_NO VARCHAR2(10) NOT NULL,

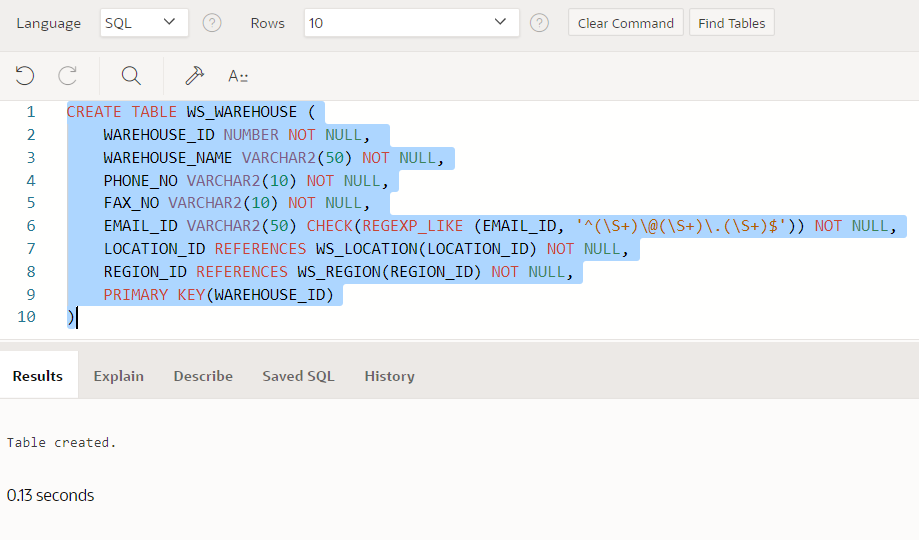
    EMAIL\_ID VARCHAR2(50) CHECK(REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$')) NOT NULL,

    LOCATION\_ID REFERENCES WS\_LOCATION(LOCATION\_ID) NOT NULL,

    REGION\_ID REFERENCES WS\_REGION(REGION\_ID) NOT NULL,

    PRIMARY KEY(WAREHOUSE\_ID)

)



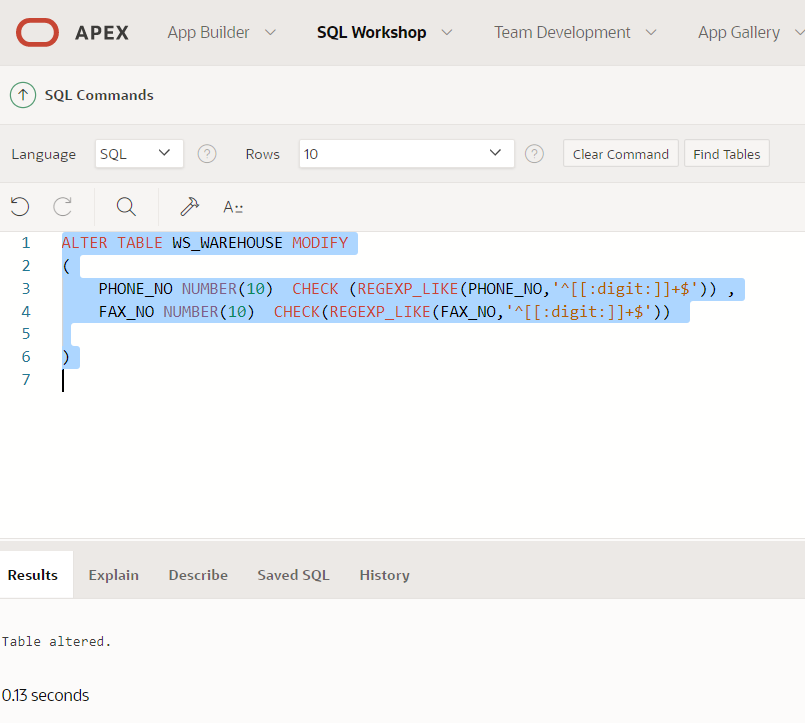
ALTER TABLE WS\_WAREHOUSE MODIFY

(

    PHONE\_NO NUMBER(10)  CHECK (REGEXP\_LIKE(PHONE\_NO,'^[[:digit:]]+$')) ,

    FAX\_NO NUMBER(10)  CHECK(REGEXP\_LIKE(FAX\_NO,'^[[:digit:]]+$'))

)



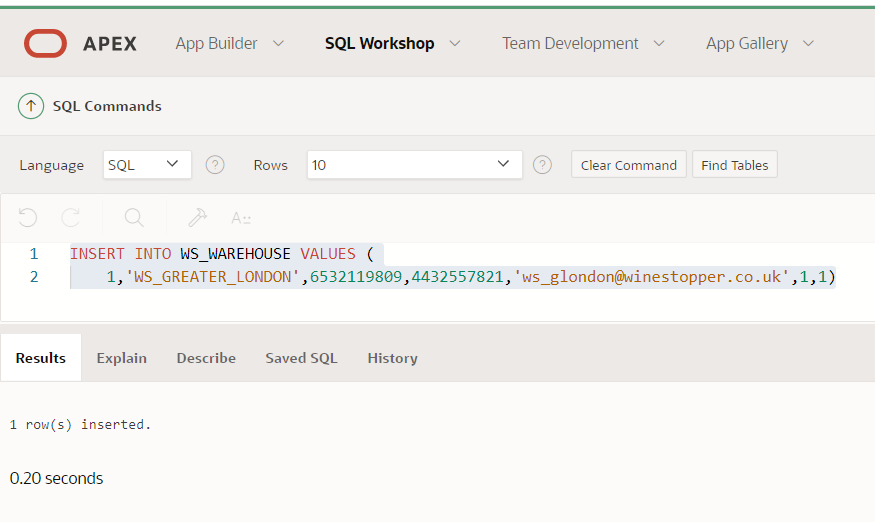
Inserting into table

INSERT INTO WS\_WAREHOUSE VALUES (

    1,'WS\_GREATER\_LONDON',6532119809,4432557821,'ws\_glondon@Wine-Stopper.co.uk',1,1)

INSERT INTO WS\_WAREHOUSE VALUES (

    2,'WS\_GREATER\_MANCHESTER',5320953781,4421879042,'ws\_gmanchester@Wine-Stopper.co.uk',2,2)



CREATE TABLE WS\_CENTER(

    CENTER\_ID NUMBER NOT NULL,

    CENTER\_NAME VARCHAR2(25) NOT NULL,

    PHONE\_NO VARCHAR2(10) NOT NULL,

    FAX\_NO VARCHAR2(10) NOT NULL,

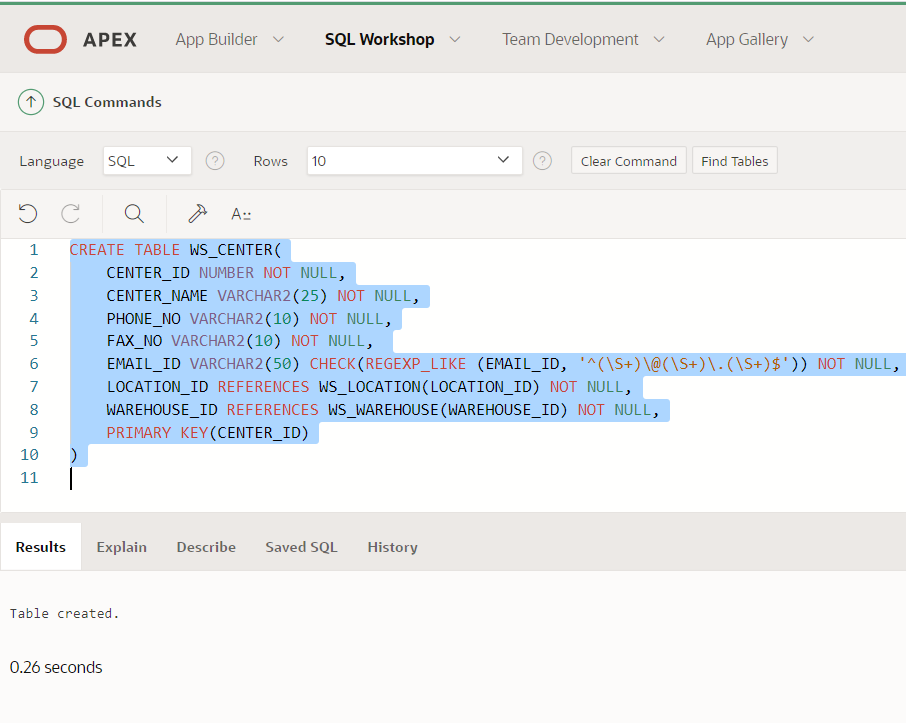
    EMAIL\_ID VARCHAR2(50) CHECK(REGEXP\_LIKE (EMAIL\_ID, '^(\S+)\@(\S+)\.(\S+)$')) NOT NULL,

    LOCATION\_ID REFERENCES WS\_LOCATION(LOCATION\_ID) NOT NULL,

    WAREHOUSE\_ID REFERENCES WS\_WAREHOUSE(WAREHOUSE\_ID) NOT NULL,

    PRIMARY KEY(CENTER\_ID)

)



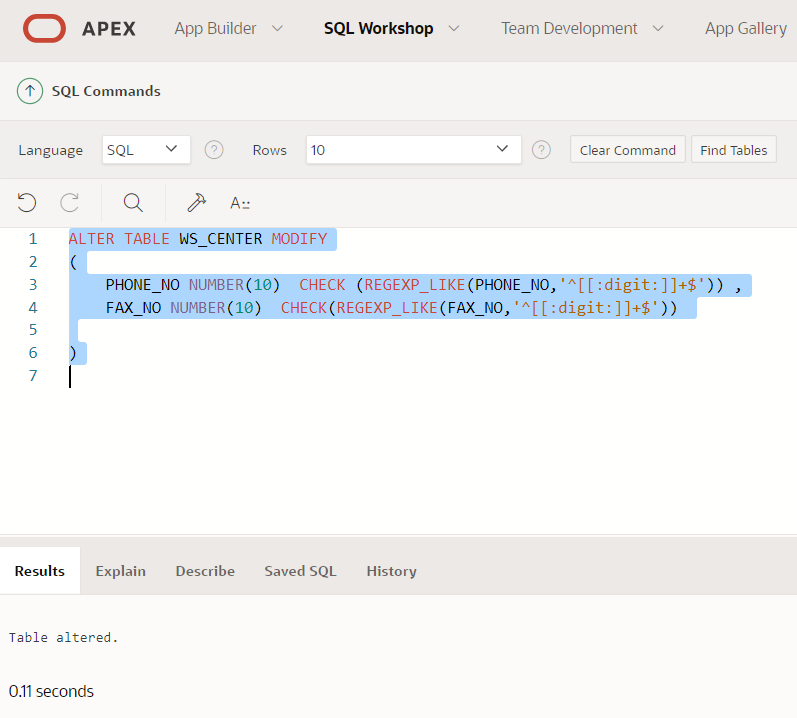
ALTER TABLE WS\_CENTER MODIFY

(

    PHONE\_NO NUMBER(10)  CHECK (REGEXP\_LIKE(PHONE\_NO,'^[[:digit:]]+$')) ,

    FAX\_NO NUMBER(10)  CHECK(REGEXP\_LIKE(FAX\_NO,'^[[:digit:]]+$'))

)



CREATE TABLE WS\_STOCK\_INFO (

    STOCK\_ID NUMBER GENERATED ALWAYS AS IDENTITY,

    CENTER\_ID REFERENCES WS\_CENTER(CENTER\_ID) NOT NULL,

    WINE\_CODE REFERENCES WS\_WINE(WINE\_CODE) NOT NULL,

    QUANTITY NUMBER NOT NULL,

    LAST\_STOCKED DATE,

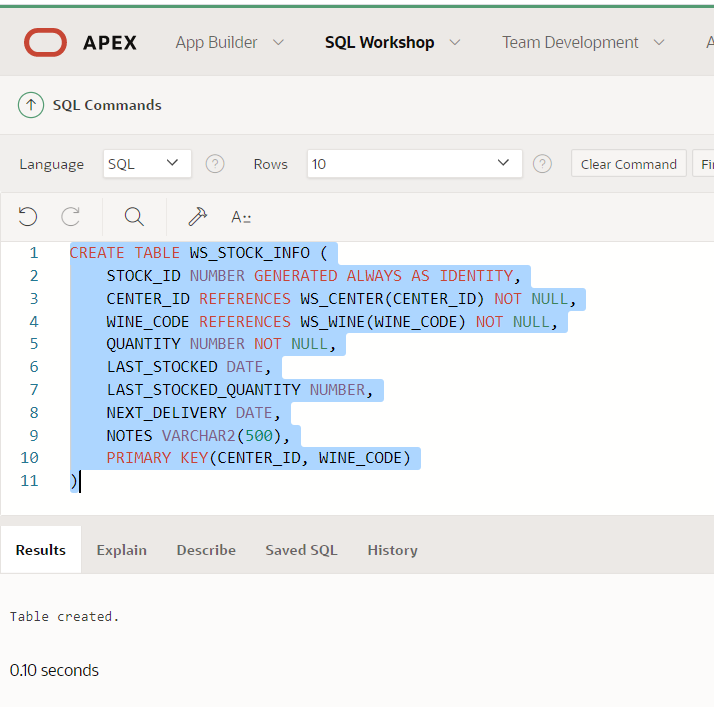
    LAST\_STOCKED\_QUANTITY NUMBER,

    NEXT\_DELIVERY DATE,

    NOTES VARCHAR2(500),

    PRIMARY KEY(CENTER\_ID, WINE\_CODE)

)



CREATE TABLE WS\_ORDER

(

    ORDER\_ID NUMBER NOT NULL,

    CENTER\_NO REFERENCES WS\_CENTER(CENTER\_ID) NOT NULL,

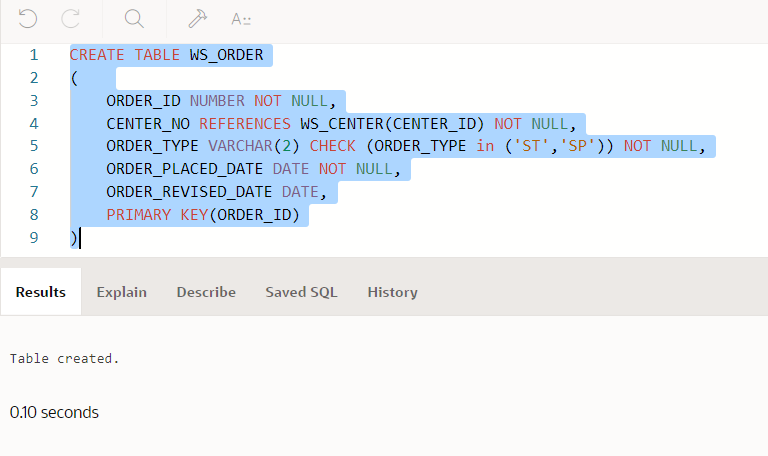
    ORDER\_TYPE VARCHAR(2) CHECK (ORDER\_TYPE in ('ST','SP')) NOT NULL,

    ORDER\_PLACED\_DATE DATE NOT NULL,

    ORDER\_REVISED\_DATE DATE,

    PRIMARY KEY(ORDER\_ID)

)



Inserting a Standing Order (ORDER\_TYPE=”ST”)

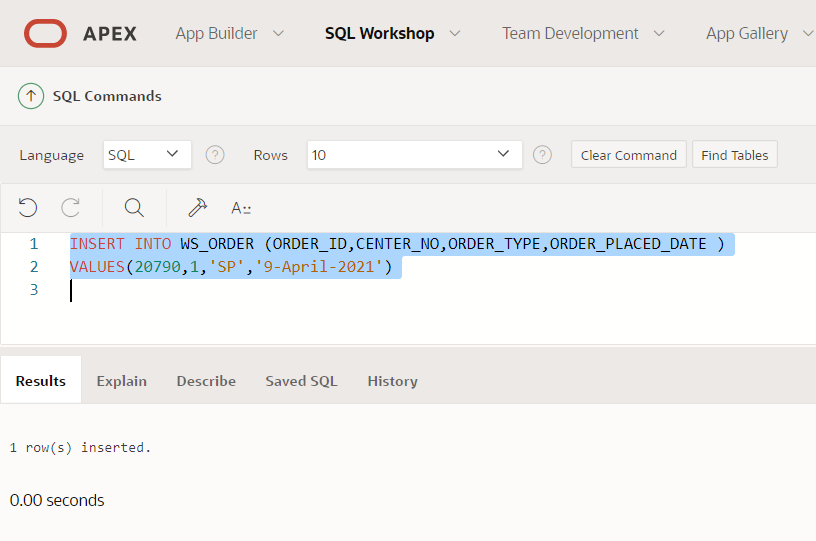
INSERT INTO WS\_ORDER (ORDER\_ID,CENTER\_NO,ORDER\_TYPE,ORDER\_PLACED\_DATE )

VALUES(20789,1,'ST','9-April-2021')

Inserting a SPECIAL Order (ORDER\_TYPE=”SP”)

INSERT INTO WS\_ORDER (ORDER\_ID,CENTER\_NO,ORDER\_TYPE,ORDER\_PLACED\_DATE )

VALUES(20790,1,'SP','9-April-2021')



CREATE TABLE WS\_ORDER\_ITEM

(

    ITEM\_ID NUMBER GENERATED ALWAYS AS IDENTITY,

    ORDER\_ID REFERENCES WS\_ORDER(ORDER\_ID) NOT NULL,

    WINE\_CODE REFERENCES WS\_WINE(WINE\_CODE) NOT NULL,

    STATUS VARCHAR2(15) CHECK(STATUS IN ('Active','Inactive')),

    ORDER\_DTL\_DESC VARCHAR2(200),

    QUANTITY NUMBER NOT NULL,

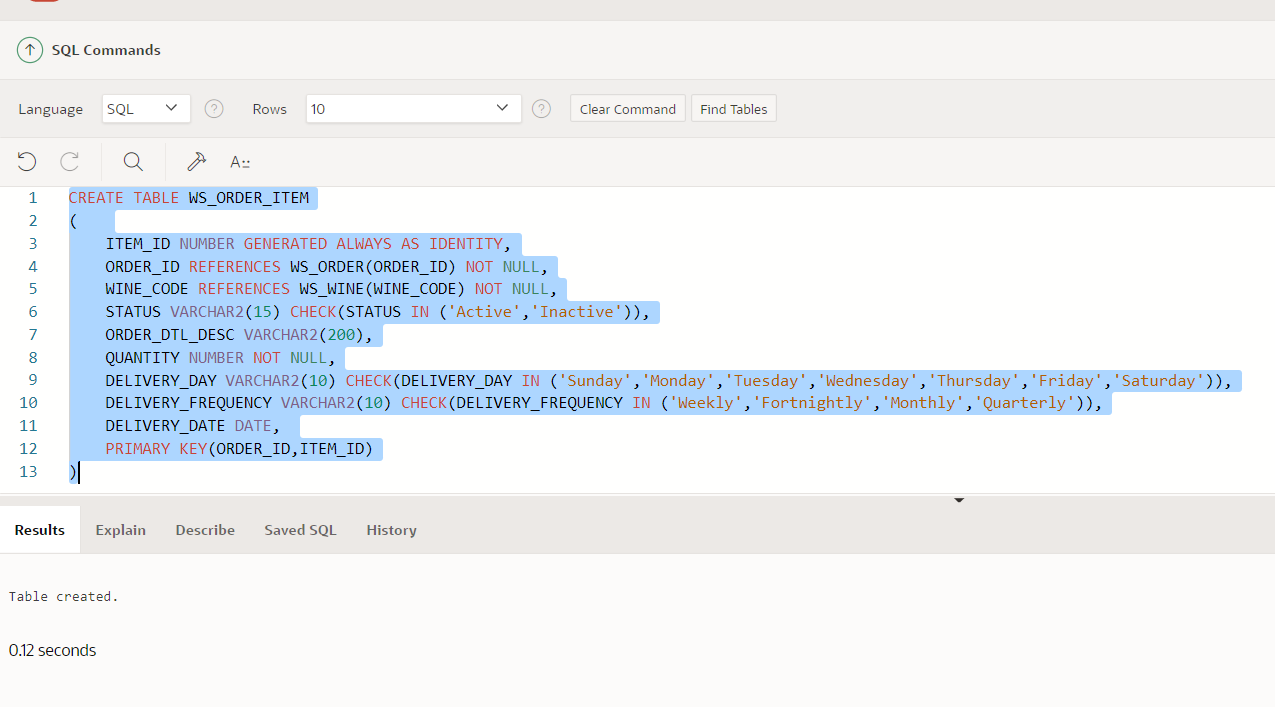
    DELIVERY\_DAY VARCHAR2(10) CHECK(DELIVERY\_DAY IN ('Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday')),

    DELIVERY\_FREQUENCY VARCHAR2(10) CHECK(DELIVERY\_FREQUENCY IN ('Weekly','Fortnightly','Monthly','Quarterly')),

    DELIVERY\_DATE DATE,

    PRIMARY KEY(ORDER\_ID,ITEM\_ID)

)



# Chapter 4 – Demonstration using SQL Queries

1. **Quantitative Analysis of Range of Wines on the Wine-Stopper Website**

The results are on the basis of Country, Colour, Category of Wine

SELECT WC.COUNTRY\_NAME, WD.COLOUR,WCAT.CATEGORY\_NAME , COUNT(WINE\_CODE)  NO\_OF\_WINES

FROM WS\_WINE W, WS\_COUNTRY WC, WS\_WINE\_DETAILS WD , WS\_WINE\_CATEGORY WCAT

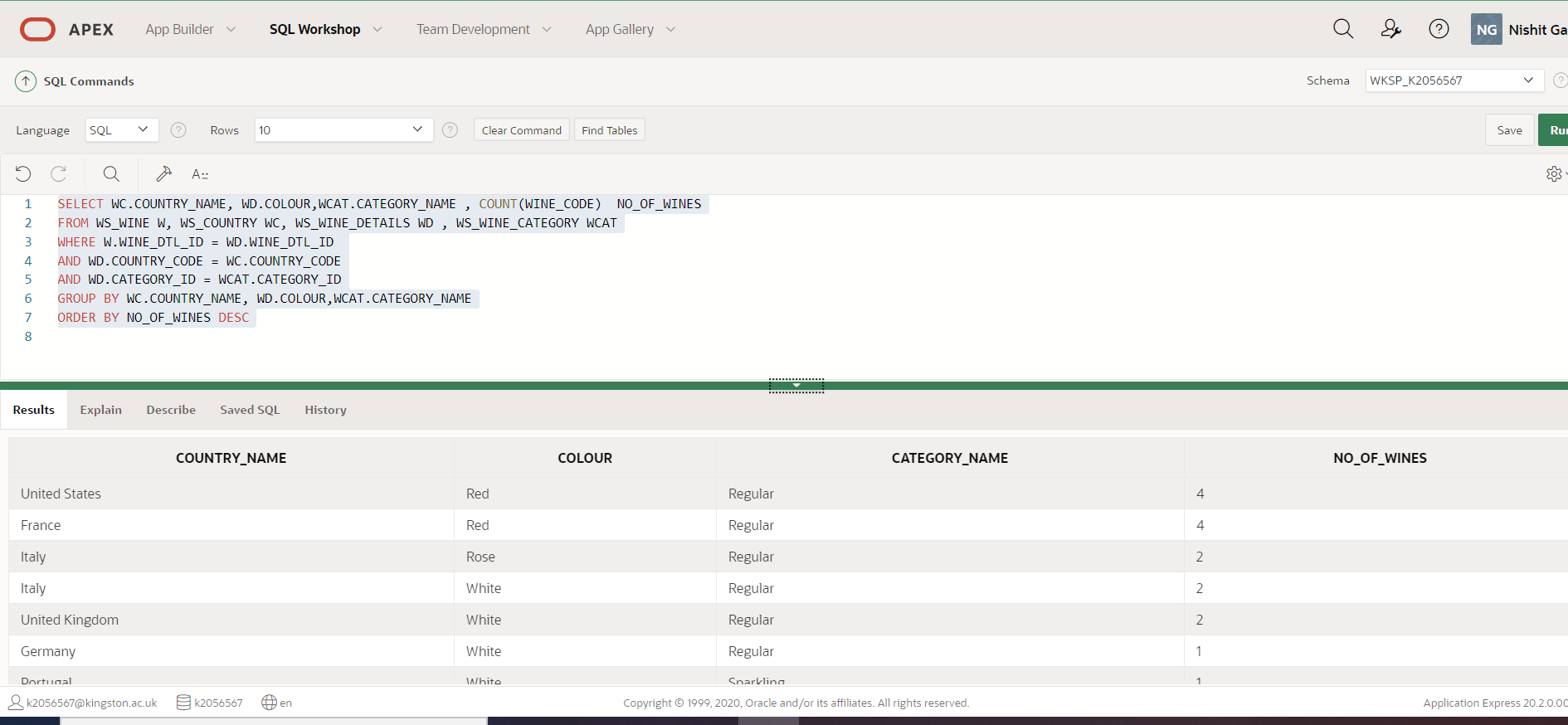
WHERE W.WINE\_DTL\_ID = WD.WINE\_DTL\_ID

AND WD.COUNTRY\_CODE = WC.COUNTRY\_CODE

AND WD.CATEGORY\_ID = WCAT.CATEGORY\_ID

GROUP BY WC.COUNTRY\_NAME, WD.COLOUR,WCAT.CATEGORY\_NAME

ORDER BY NO\_OF\_WINES DESC



1. **Selection of Wines with information about Special Offers, Availability**

User has an option to filter by Wine Category of General, Sparkling and Dessert Wines

User has an option to select the colour of the wine

User has an option to set the country of the wine

The Center ID has to be provided on the basis of User location

SELECT  W.WINE\_CODE, WD.WINE\_NAME || '  ' ||  W.CASE\_SIZE\_IN\_LTR  || ' LTR' AS "WINE\_NAME" ,

WD.WINE\_DESC AS WINE\_DESC , WI.IND\_DESC AS WINE\_INDICATOR, W.PRICE\_PER\_BOTTLE AS "BOTTLE(£)", W.PRICE\_PER\_CASE AS "CASE(£)",

W.NUM\_OF\_BOTTLE || ' X ' ||  W.BOTTLE\_SIZE\_IN\_ML AS "CASE CONTAINS",

CASE WHEN  WS.NEXT\_DELIVERY IS NOT NULL AND WS.QUANTITY <= 0 THEN  'AVAILABLE IN ' ||  TO\_CHAR(WS.NEXT\_DELIVERY - TRUNC(SYSDATE)) || ' DAYS'

     WHEN  WS.NEXT\_DELIVERY IS NULL AND WS.QUANTITY <= 0 THEN  'UNAVAILABLE'

     WHEN  WS.QUANTITY > 0 THEN  'IN STOCK'

END AS AVAILABILITY,

CASE

    WHEN  WP.DISCOUNT\_PERCENT > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

    THEN WP.DISCOUNT\_PERCENT || ' % OFF'

     WHEN WP.DISCOUNT\_QUANTITY\_BUY > 0 AND  WP.DISCOUNT\_QUANTITY\_FOR > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'BUY ' || WP.DISCOUNT\_QUANTITY\_BUY || ' FOR ' || DISCOUNT\_QUANTITY\_FOR

     WHEN WP.RED\_PRICE\_PER\_CASE > 0  AND   WP.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || WP.RED\_PRICE\_PER\_CASE ||   ', REDUCED PRICE PER BOTTLE (£): ' || WP.RED\_PRICE\_PER\_BOTTLE

     WHEN WP.RED\_PRICE\_PER\_CASE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || WP.RED\_PRICE\_PER\_CASE

     WHEN WP.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER BOTTLE (£): ' || WP.RED\_PRICE\_PER\_BOTTLE

END AS SPECIAL\_OFFER

FROM WS\_WINE W, WS\_WINE\_DETAILS WD, WS\_COUNTRY WC, WS\_WINE\_INDICATOR WI,

WS\_PROMOTION WP, WS\_WINE\_CATEGORY WCAT, WS\_STOCK\_INFO WS

WHERE W.WINE\_DTL\_ID = WD.WINE\_DTL\_ID

AND WD.COUNTRY\_CODE = WC.COUNTRY\_CODE

AND WD.IND\_ID = WI.IND\_ID

AND WD.CATEGORY\_ID = WCAT.CATEGORY\_ID

AND W.WINE\_CODE = WP.WINE\_CODE(+)

AND W.WINE\_CODE = WS.WINE\_CODE

AND UPPER(WCAT.CATEGORY\_NAME) LIKE COALESCE('','%') --  REGULAR, SPARKLING, DESSERT

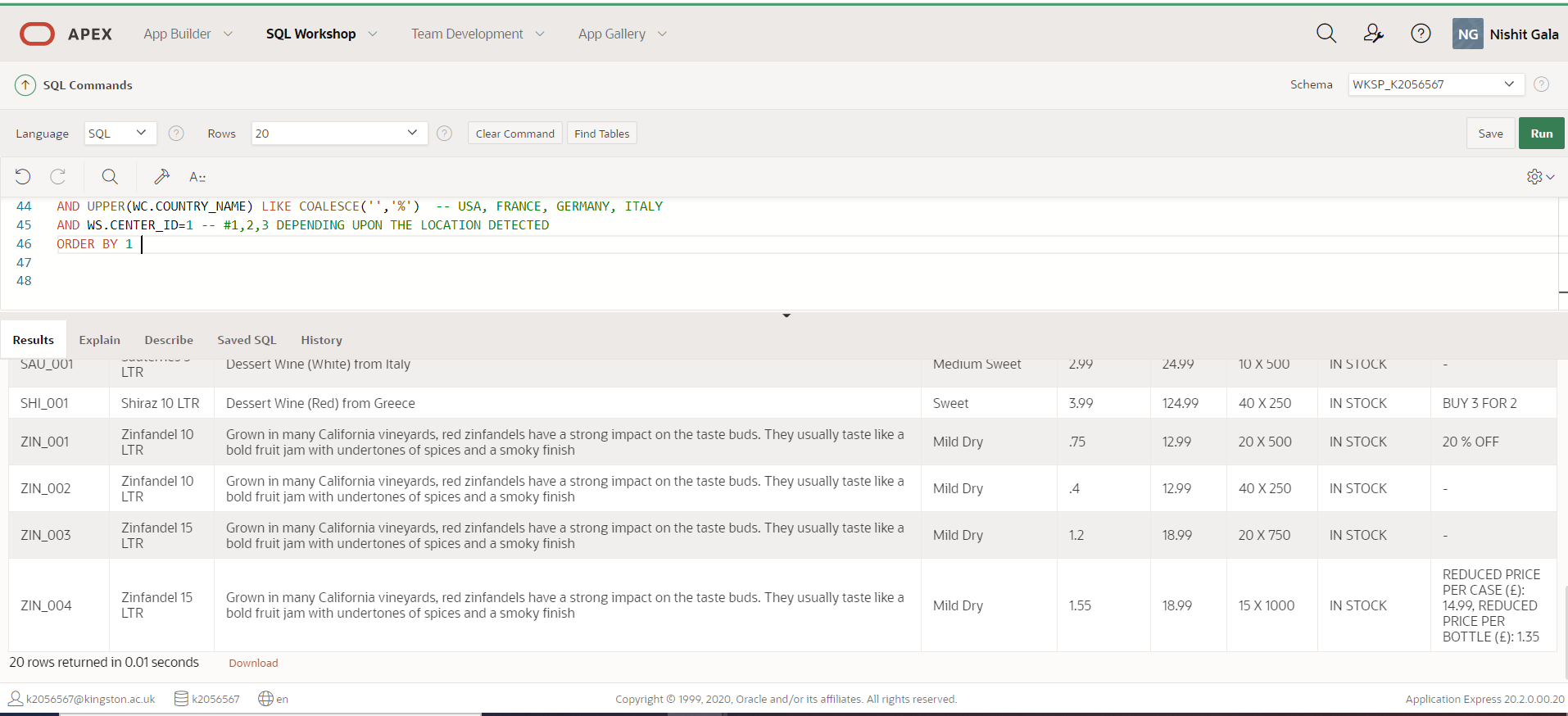
AND UPPER(WD.COLOUR) LIKE COALESCE('','%') -- RED, WHITE, GOLD, ROSE, YELLOW, ORANGE

AND UPPER(WC.COUNTRY\_NAME) LIKE COALESCE('','%')  -- UNITED STATES, FRANCE, GERMANY, ITALY

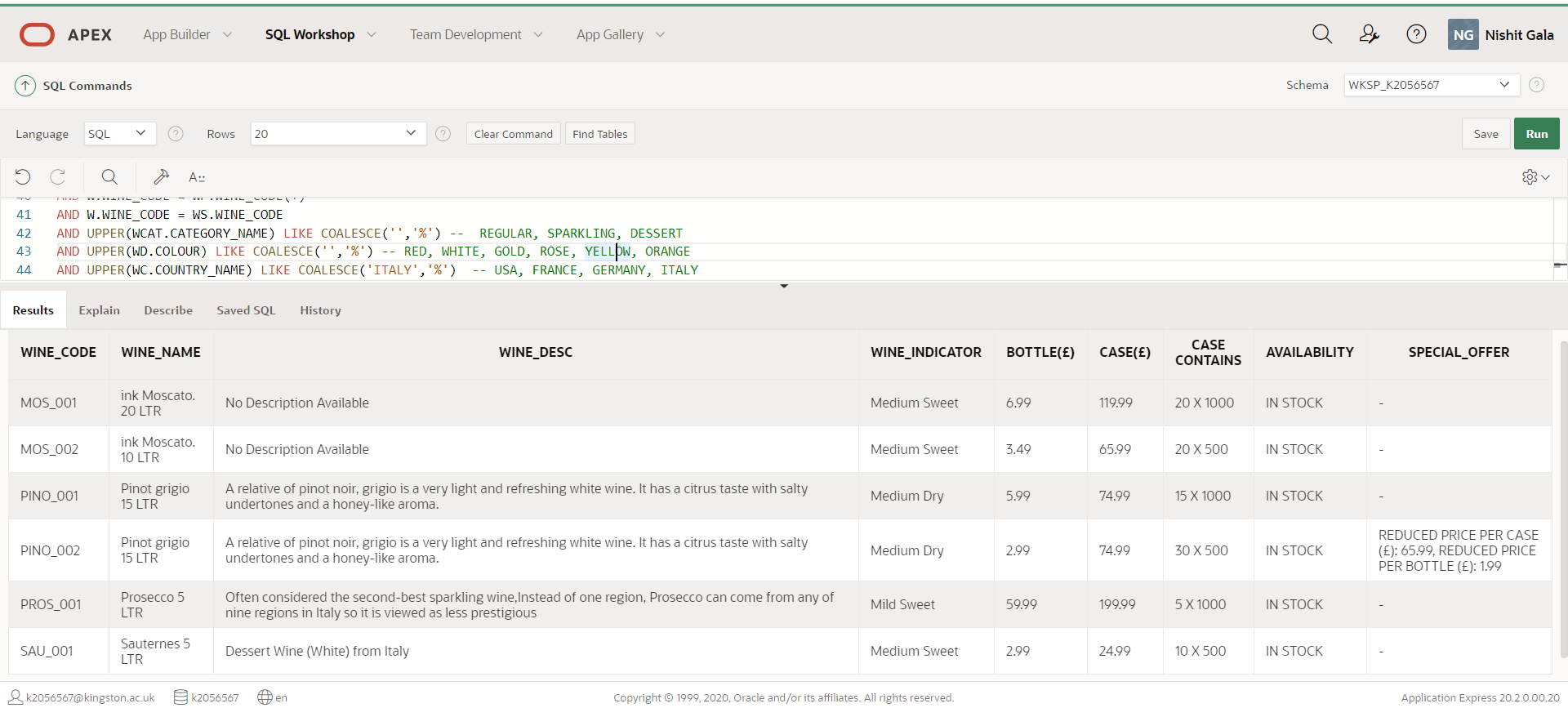
AND WS.CENTER\_ID=1 -- #1,2,3 DEPENDING UPON THE LOCATION DETECTED

ORDER BY 1

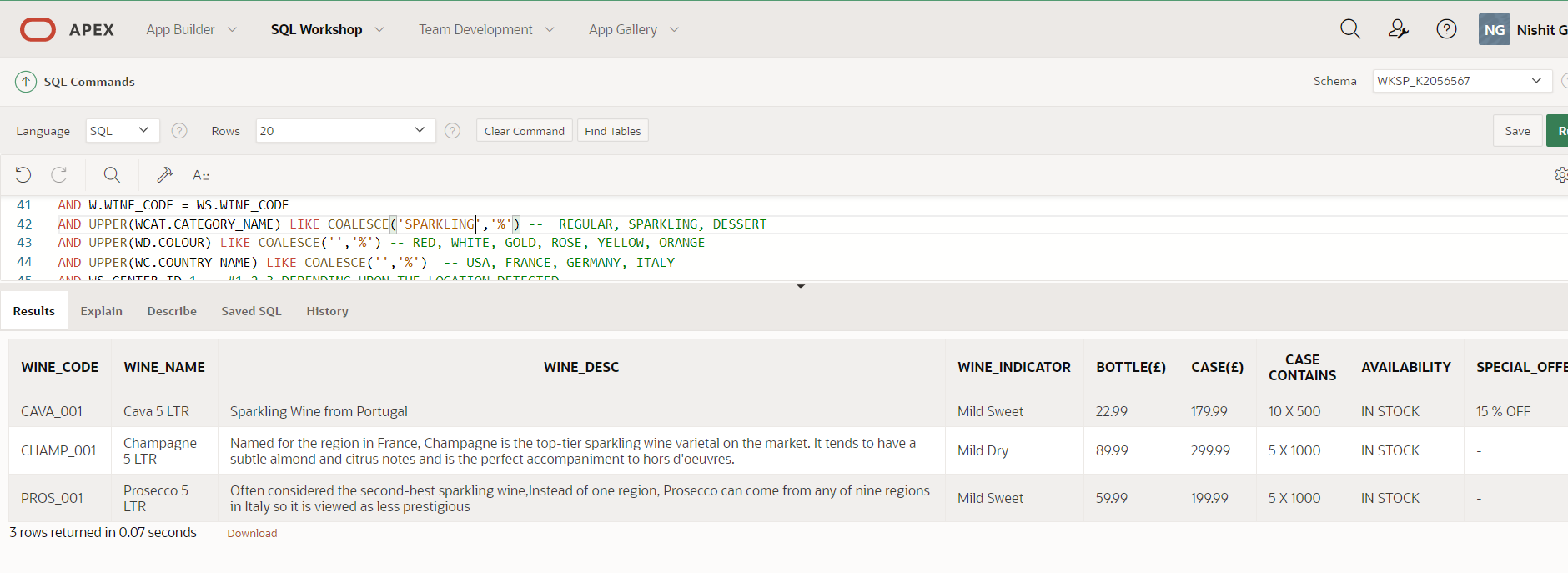
All Wines in CENTER 1 -



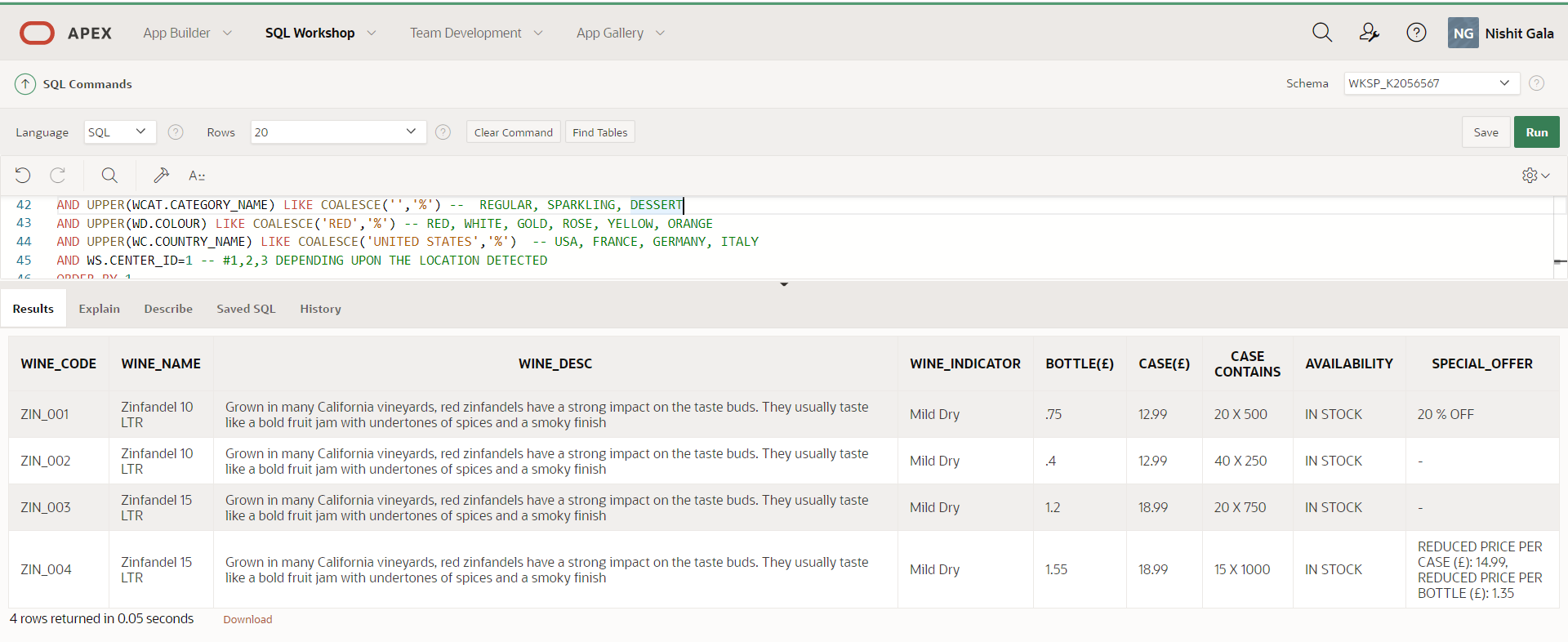
Italian Wine Example-



Sparkling Wine Example –



RED WINES of United States –



Output of Query for CENTER – 1 returning all wines:



1. On Promotion Wines

SELECT  W.WINE\_CODE, WD.WINE\_NAME || '  ' ||  W.CASE\_SIZE\_IN\_LTR  || ' LTR' AS "WINE\_NAME" ,

WD.WINE\_DESC AS WINE\_DESC , WI.IND\_DESC AS WINE\_INDICATOR, W.PRICE\_PER\_BOTTLE AS "BOTTLE(£)", W.PRICE\_PER\_CASE AS "CASE(£)",

W.NUM\_OF\_BOTTLE || ' X ' ||  W.BOTTLE\_SIZE\_IN\_ML AS "CASE CONTAINS",

CASE WHEN  WS.NEXT\_DELIVERY IS NOT NULL AND WS.QUANTITY <= 0 THEN  'AVAILABLE IN ' ||  TO\_CHAR(WS.NEXT\_DELIVERY - TRUNC(SYSDATE)) || ' DAYS'

     WHEN  WS.NEXT\_DELIVERY IS NULL AND WS.QUANTITY <= 0 THEN  'UNAVAILABLE'

     WHEN  WS.QUANTITY > 0 THEN  'IN STOCK'

END AS AVAILABILITY,

CASE

    WHEN  WP.DISCOUNT\_PERCENT > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

    THEN WP.DISCOUNT\_PERCENT || ' % OFF'

     WHEN WP.DISCOUNT\_QUANTITY\_BUY > 0 AND  WP.DISCOUNT\_QUANTITY\_FOR > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'BUY ' || WP.DISCOUNT\_QUANTITY\_BUY || ' FOR ' || DISCOUNT\_QUANTITY\_FOR

     WHEN WP.RED\_PRICE\_PER\_CASE > 0  AND   WP.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || WP.RED\_PRICE\_PER\_CASE ||   ', REDUCED PRICE PER BOTTLE (£): ' || WP.RED\_PRICE\_PER\_BOTTLE

     WHEN WP.RED\_PRICE\_PER\_CASE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || WP.RED\_PRICE\_PER\_CASE

     WHEN WP.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

          AND WP.STATUS = 'Active'

     THEN 'REDUCED PRICE PER BOTTLE (£): ' || WP.RED\_PRICE\_PER\_BOTTLE

END AS SPECIAL\_OFFER

FROM WS\_WINE W, WS\_WINE\_DETAILS WD, WS\_COUNTRY WC, WS\_WINE\_INDICATOR WI,

WS\_PROMOTION WP, WS\_WINE\_CATEGORY WCAT, WS\_STOCK\_INFO WS

WHERE W.WINE\_DTL\_ID = WD.WINE\_DTL\_ID

AND WD.COUNTRY\_CODE = WC.COUNTRY\_CODE

AND WD.IND\_ID = WI.IND\_ID

AND WD.CATEGORY\_ID = WCAT.CATEGORY\_ID

AND W.WINE\_CODE = WP.WINE\_CODE

AND W.WINE\_CODE = WS.WINE\_CODE

AND UPPER(WCAT.CATEGORY\_NAME) LIKE COALESCE('','%') --  REGULAR, SPARKLING, DESSERT

AND UPPER(WD.COLOUR) LIKE COALESCE('','%') -- RED, WHITE, GOLD, ROSE, YELLOW, ORANGE

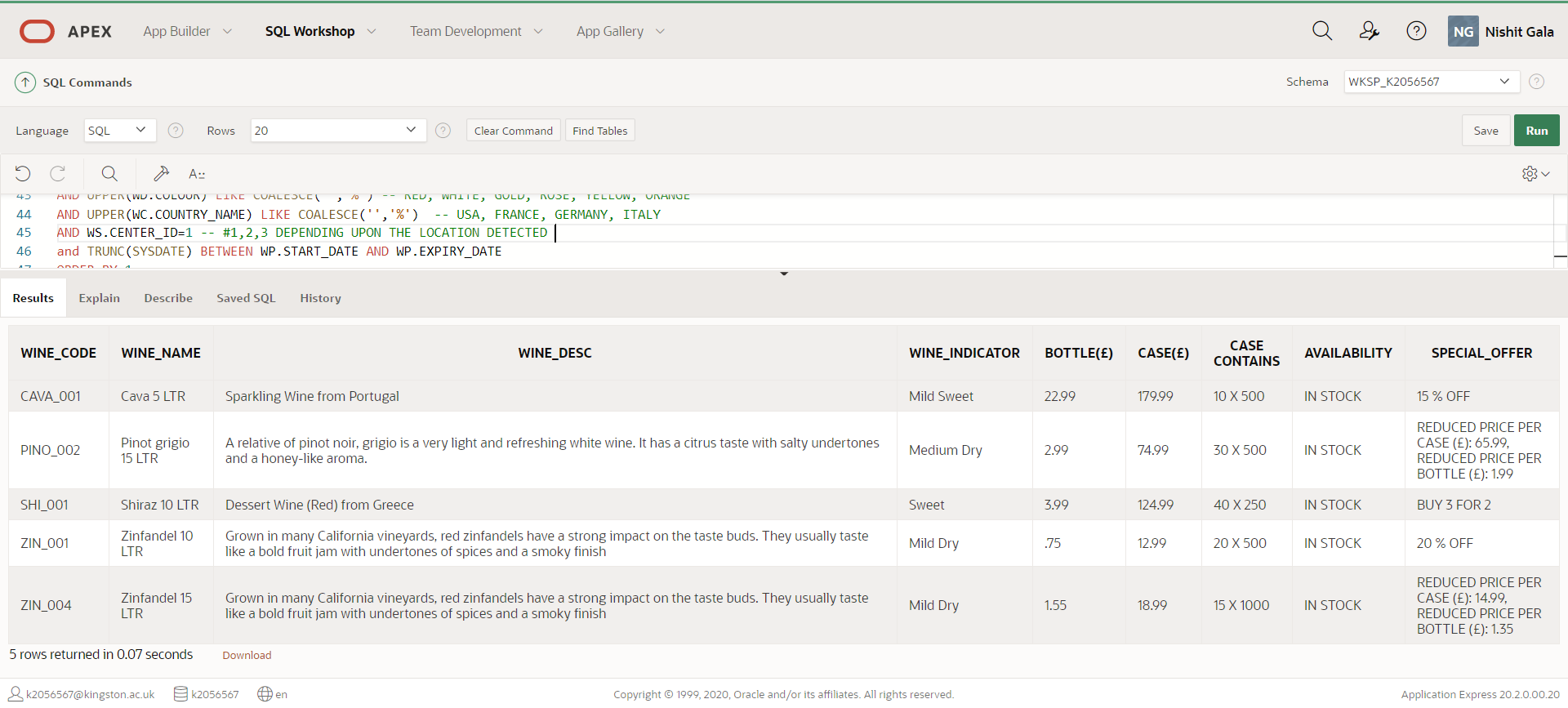
AND UPPER(WC.COUNTRY\_NAME) LIKE COALESCE('','%')  -- USA, FRANCE, GERMANY, ITALY

AND WS.CENTER\_ID=1 -- #1,2,3 DEPENDING UPON THE LOCATION DETECTED

and TRUNC(SYSDATE) BETWEEN WP.START\_DATE AND WP.EXPIRY\_DATE

ORDER BY 1

Wines on Prmotion



1. **Data shown in the User Shopping Basket before the purchase**

SELECT CUST.FNAME ,CUST.LNAME ,

WDET.WINE\_NAME || '  ' ||  WINE.CASE\_SIZE\_IN\_LTR  || ' LTR' AS "WINE\_NAME" , WINE.WINE\_CODE,

COUNTRY.COUNTRY\_NAME AS "COUNTRY",  WDET.COLOUR AS "COLOUR",

WINE.PRICE\_PER\_BOTTLE AS "BOTTLE(£)", WINE.PRICE\_PER\_CASE AS "CASE(£)",

CASE WHEN  WS.NEXT\_DELIVERY IS NOT NULL AND WS.QUANTITY <= 0 THEN  'AVAILABLE IN ' ||  TO\_CHAR(WS.NEXT\_DELIVERY - TRUNC(SYSDATE)) || ' DAYS'

     WHEN  WS.NEXT\_DELIVERY IS NULL AND WS.QUANTITY <= 0 THEN  'UNAVAILABLE'

     WHEN  WS.QUANTITY > 0 THEN  'IN STOCK'

END AS AVAILABILITY,

CASE

    WHEN  PROMO.DISCOUNT\_PERCENT > 0

          AND TRUNC(SYSDATE) BETWEEN PROMO.START\_DATE AND PROMO.EXPIRY\_DATE

          AND PROMO.STATUS = 'Active'

    THEN PROMO.DISCOUNT\_PERCENT || ' % OFF'

     WHEN PROMO.DISCOUNT\_QUANTITY\_BUY > 0 AND  PROMO.DISCOUNT\_QUANTITY\_FOR > 0

          AND TRUNC(SYSDATE) BETWEEN PROMO.START\_DATE AND PROMO.EXPIRY\_DATE

          AND PROMO.STATUS = 'Active'

     THEN 'BUY ' || PROMO.DISCOUNT\_QUANTITY\_BUY || ' FOR ' || DISCOUNT\_QUANTITY\_FOR

     WHEN PROMO.RED\_PRICE\_PER\_CASE > 0  AND   PROMO.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN PROMO.START\_DATE AND PROMO.EXPIRY\_DATE

          AND PROMO.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || PROMO.RED\_PRICE\_PER\_CASE ||   ', REDUCED PRICE PER BOTTLE (£): ' || PROMO.RED\_PRICE\_PER\_BOTTLE

     WHEN PROMO.RED\_PRICE\_PER\_CASE > 0

          AND TRUNC(SYSDATE) BETWEEN PROMO.START\_DATE AND PROMO.EXPIRY\_DATE

          AND PROMO.STATUS = 'Active'

     THEN 'REDUCED PRICE PER CASE (£): ' || PROMO.RED\_PRICE\_PER\_CASE

     WHEN PROMO.RED\_PRICE\_PER\_BOTTLE > 0

          AND TRUNC(SYSDATE) BETWEEN PROMO.START\_DATE AND PROMO.EXPIRY\_DATE

          AND PROMO.STATUS = 'Active'

     THEN 'REDUCED PRICE PER BOTTLE (£): ' || PROMO.RED\_PRICE\_PER\_BOTTLE

END AS "SPECIAL OFFER",

BASKET.SELECTION\_TYPE || ' : ' || BASKET.QUANTITY AS "QUANTITY REQUESTED"

FROM WS\_SHOPPING\_BASKET BASKET, WS\_CUSTOMER CUST, WS\_WINE WINE, WS\_WINE\_DETAILS WDET, WS\_COUNTRY COUNTRY, WS\_PROMOTION PROMO, WS\_STOCK\_INFO WS

WHERE BASKET.CUST\_ID = CUST.CUST\_ID

AND BASKET.WINE\_CODE = WINE.WINE\_CODE

AND WINE.WINE\_DTL\_ID = WDET.WINE\_DTL\_ID

AND WDET.COUNTRY\_CODE = COUNTRY.COUNTRY\_CODE

AND WINE.WINE\_CODE = PROMO.WINE\_CODE(+)

AND WS.WINE\_CODE= WINE.WINE\_CODE

AND CUST.CENTER\_ID = WS.CENTER\_ID

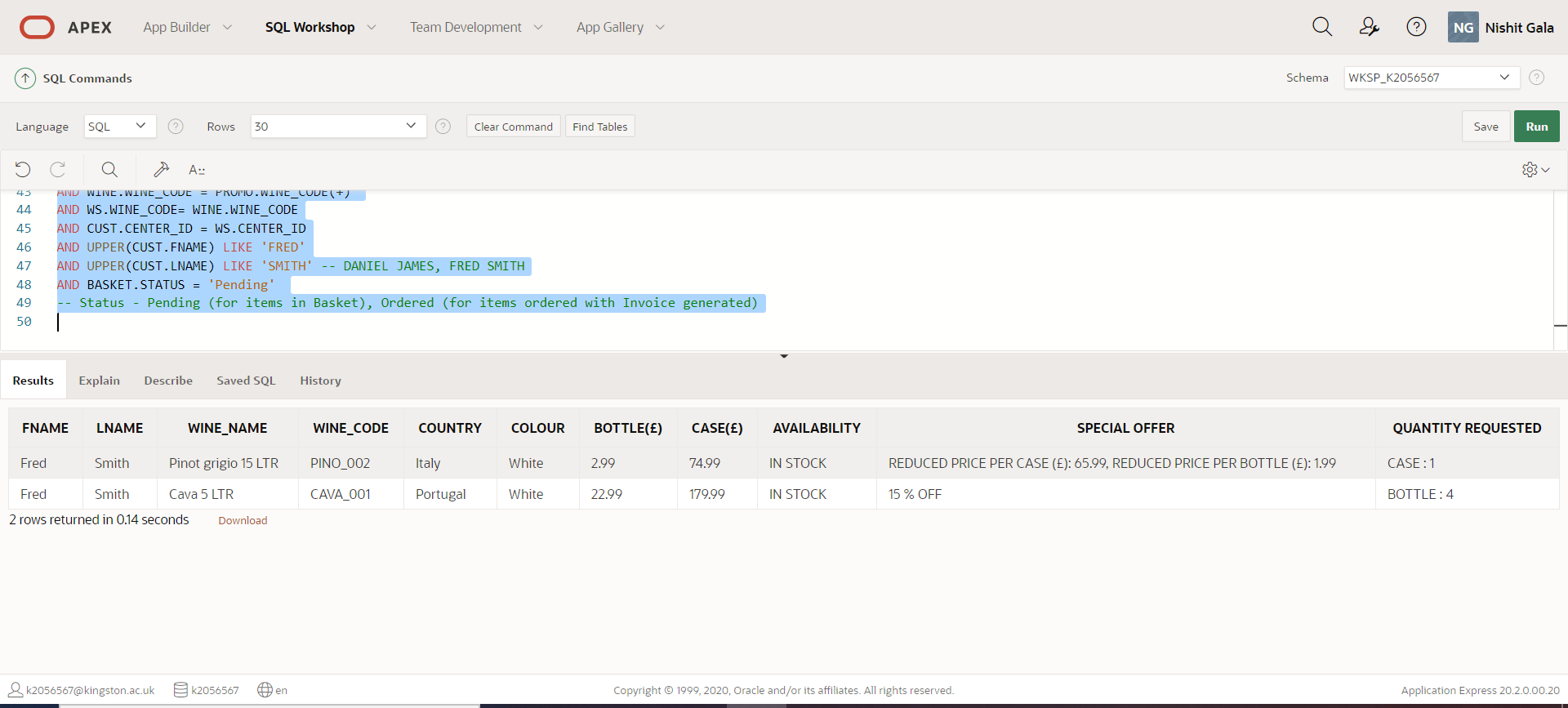
AND UPPER(CUST.FNAME) LIKE 'FRED'

AND UPPER(CUST.LNAME) LIKE 'SMITH' -- DANIEL JAMES, FRED SMITH

AND BASKET.STATUS = 'Pending'

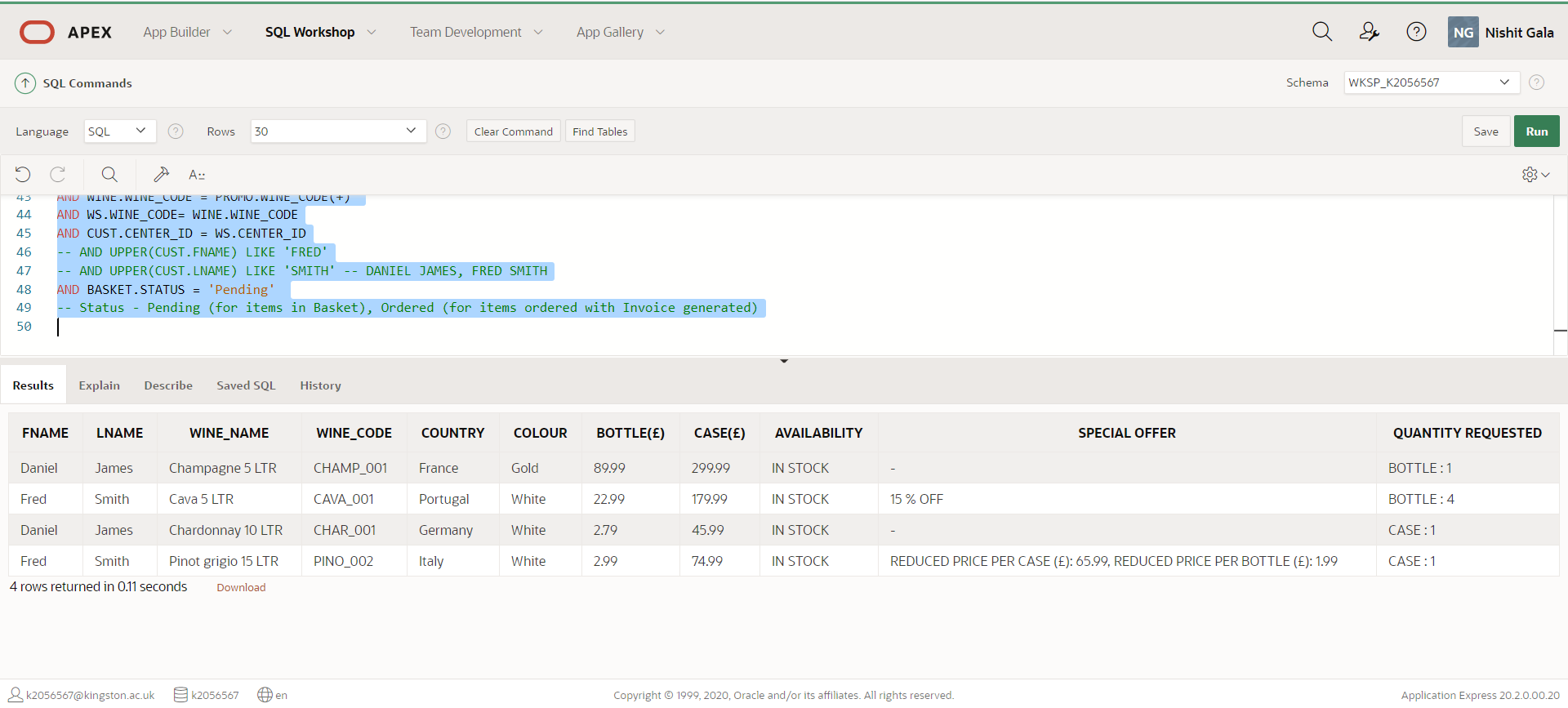
-- Status - Pending (for items in Basket), Ordered (for items ordered with Invoice generated)

FRED SMITH BASKET -



In Below example,

Note: Customer Name is not taken in where clause, so it will return all the customers which have items with Status = Pending in WS\_SHOPPING\_BASKET



1. **Invoice Data post the Purchase is made**

**Part A – Displaying the purchased items with selling price, discount, total price for a customer on basis of either customer name or order no**

SELECT   INV.INVOICE\_NO, INV.INVOICE\_DATE, CUST.FNAME ,CUST.LNAME ,

WDET.WINE\_NAME || '  ' ||  WINE.CASE\_SIZE\_IN\_LTR  || ' LTR' AS "WINE\_NAME",

BASKET.DELIVERY\_DATE,

BASKET.SELECTION\_TYPE || ' : ' || BASKET.QUANTITY AS "QUANTITY\_REQUESTED" ,

CASE WHEN BASKET.SELECTION\_TYPE = 'BOTTLE' THEN WINE.PRICE\_PER\_BOTTLE

     WHEN BASKET.SELECTION\_TYPE = 'CASE' THEN WINE.PRICE\_PER\_CASE END AS "ITEM\_PRICE (£)",

 '(' || BASKET.DISCOUNT\_AMOUNT || ')'  AS "DISCOUNT (£)",

BASKET.SELLING\_PRICE AS "TOTAL\_COST (£)"

FROM WS\_SHOPPING\_BASKET BASKET, WS\_CUSTOMER CUST, WS\_WINE WINE, WS\_WINE\_DETAILS WDET, WS\_INVOICE INV

WHERE BASKET.CUST\_ID = CUST.CUST\_ID

AND BASKET.WINE\_CODE = WINE.WINE\_CODE

AND WINE.WINE\_DTL\_ID = WDET.WINE\_DTL\_ID

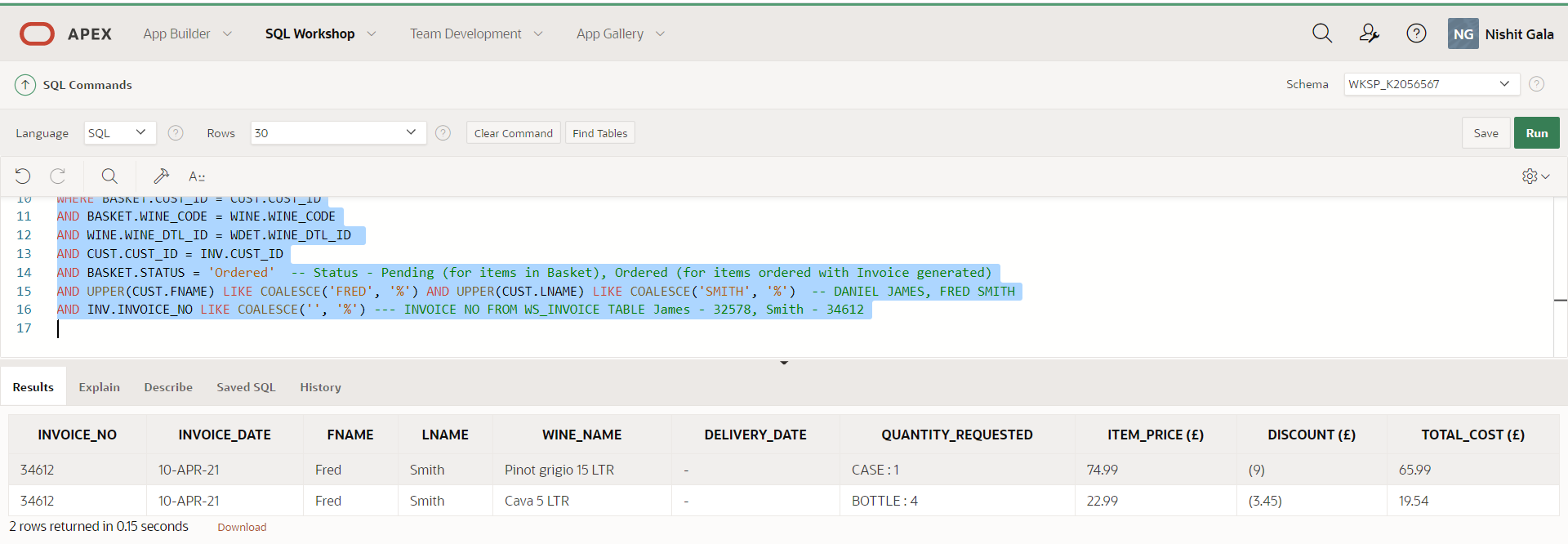
AND CUST.CUST\_ID = INV.CUST\_ID

AND BASKET.STATUS = 'Ordered'  -- Status - Pending (for items in Basket), Ordered (for items ordered with Invoice generated)

AND UPPER(CUST.FNAME) LIKE COALESCE('FRED', '%') AND UPPER(CUST.LNAME) LIKE COALESCE('SMITH', '%')  -- DANIEL JAMES, FRED SMITH

AND INV.INVOICE\_NO LIKE COALESCE('', '%') --- INVOICE NO FROM WS\_INVOICE TABLE James - 32578, Smith - 34612

Note – Status is Ordered when order invoice is generated



**Part B - Order Total, Delivery Charge will be populated using below query**

SELECT INVOICE\_NO, INVOICE\_DATE,

ACTUAL\_PRICE\_TOTAL as "ACTUAL\_PRICE\_TOTAL(£)",

SELLING\_PRICE\_TOTAL AS "SELLING\_TOTAL(£)",

 '(' || SAVINGS\_TOTAL  || ')' AS "SAVINGS\_TOTAL(£)",

DELIVERY\_CHARGE AS "DELIVERY\_CHARGE(£)",

SELLING\_PRICE\_TOTAL + DELIVERY\_CHARGE AS "ORDER\_TOTAL(£)" FROM (

SELECT ORD.INVOICE\_NO, ORD.INVOICE\_DATE,ORD.ACTUAL\_PRICE\_TOTAL, ORD.SELLING\_PRICE\_TOTAL , ORD.SAVINGS\_TOTAL,

( SELECT DELIVERY\_CHARGE FROM WS\_CALCULATE\_DELIVERY\_CHARGE

WHERE ORD.SELLING\_PRICE\_TOTAL BETWEEN FRM\_AMOUNT AND TO\_AMOUNT) AS DELIVERY\_CHARGE

FROM

(SELECT INV.INVOICE\_NO , INV.INVOICE\_DATE,

SUM(BASKET.ACTUAL\_PRICE) AS ACTUAL\_PRICE\_TOTAL,

SUM (BASKET.SELLING\_PRICE) AS SELLING\_PRICE\_TOTAL,

SUM(BASKET.DISCOUNT\_AMOUNT) as SAVINGS\_TOTAL

FROM WS\_SHOPPING\_BASKET BASKET, WS\_CUSTOMER CUST, WS\_WINE WINE, WS\_WINE\_DETAILS WDET,

WS\_INVOICE INV

WHERE BASKET.CUST\_ID = CUST.CUST\_ID

AND BASKET.WINE\_CODE = WINE.WINE\_CODE

AND WINE.WINE\_DTL\_ID = WDET.WINE\_DTL\_ID

AND CUST.CUST\_ID = INV.CUST\_ID

AND BASKET.STATUS = 'Ordered'  -- Status - Pending (for items in Basket), Ordered (for items ordered with Invoice generated)

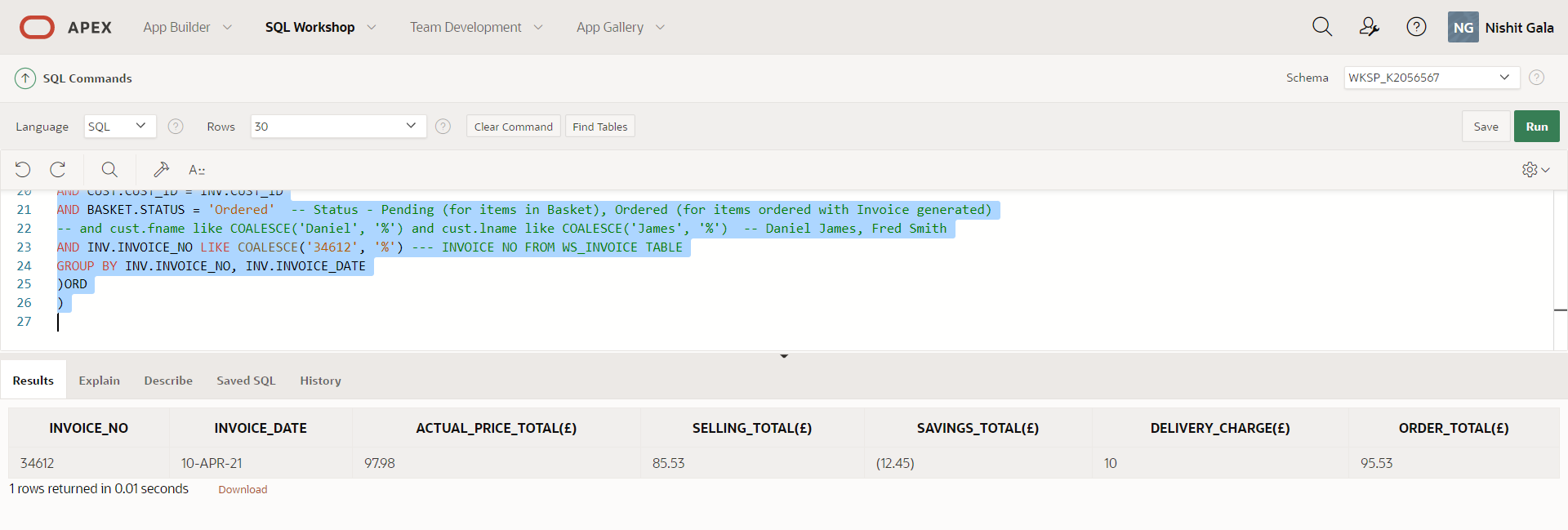
-- and cust.fname like COALESCE('Daniel', '%') and cust.lname like COALESCE('James', '%')  -- Daniel James, Fred Smith

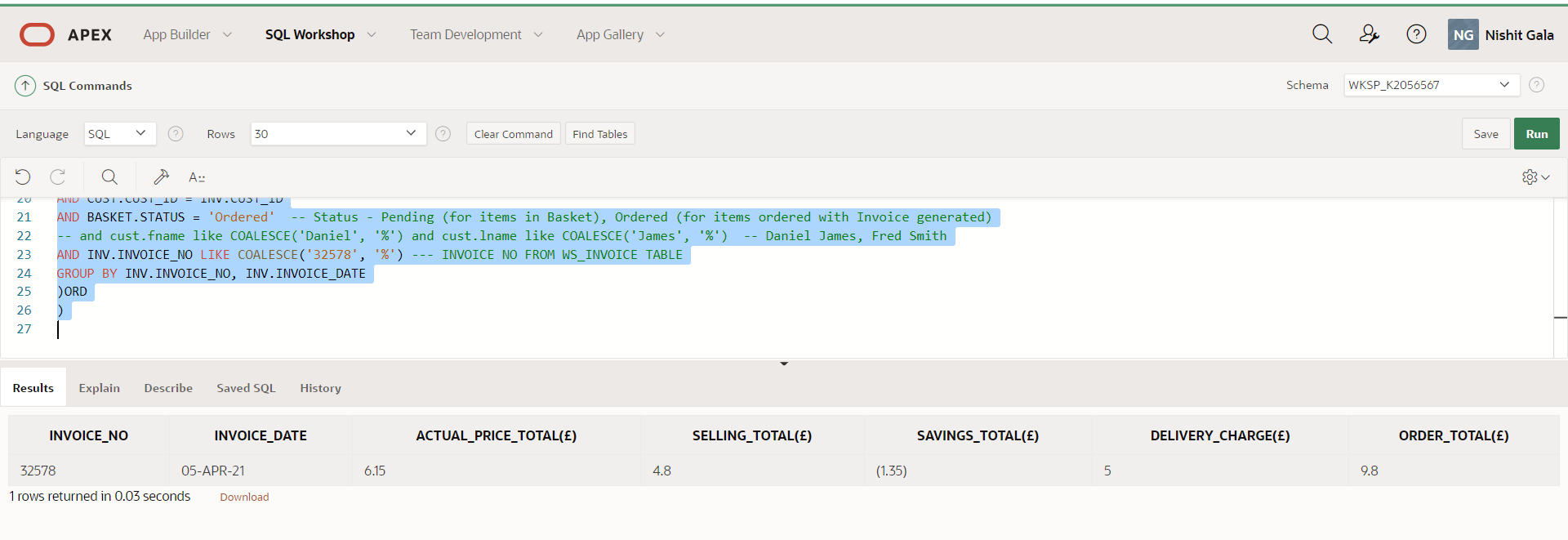
AND INV.INVOICE\_NO LIKE COALESCE('34612', '%') --- INVOICE NO FROM WS\_INVOICE TABLE

GROUP BY INV.INVOICE\_NO, INV.INVOICE\_DATE

)ORD

)





1. **Distribution Center placing Order**

**Part A – Special Order**

SELECT  ORD.ORDER\_ID, 'SPECIAL\_ORDER' AS "ORDER\_TYPE" ,

ORD.ORDER\_PLACED\_DATE, ORD.ORDER\_REVISED\_DATE, WINE.WINE\_CODE,

ITEM.DELIVERY\_DATE, ITEM.STATUS,

WDET.WINE\_DESC,ITEM.QUANTITY,

WL.STREET\_ADDRESS || '  '  || WL.CITY || '  ' ||   WL.POSTAL\_CODE || '  TEL:  '  || WC.PHONE\_NO AS "DISTRIBUTION\_CENTER",

WL2.STREET\_ADDRESS || '  '  || WL2.CITY || '  ' ||   WL2.POSTAL\_CODE || '  TEL:  '  || WH.PHONE\_NO AS "DISTRIBUTION\_CENTER",

WR.REGION\_NAME

FROM WS\_ORDER ORD, WS\_ORDER\_ITEM ITEM, WS\_WINE WINE, WS\_WINE\_DETAILS WDET,

WS\_CENTER WC, WS\_LOCATION WL, WS\_WAREHOUSE WH, WS\_LOCATION WL2, WS\_REGION WR

WHERE ORD.ORDER\_ID = ITEM.ORDER\_ID

AND WINE.WINE\_CODE = ITEM.WINE\_CODE

AND WDET.WINE\_DTL\_ID = WINE.WINE\_DTL\_ID

AND WC.CENTER\_ID = ORD.CENTER\_NO

AND WL.LOCATION\_ID = WC.LOCATION\_ID

AND WC.WAREHOUSE\_ID = WH.WAREHOUSE\_ID

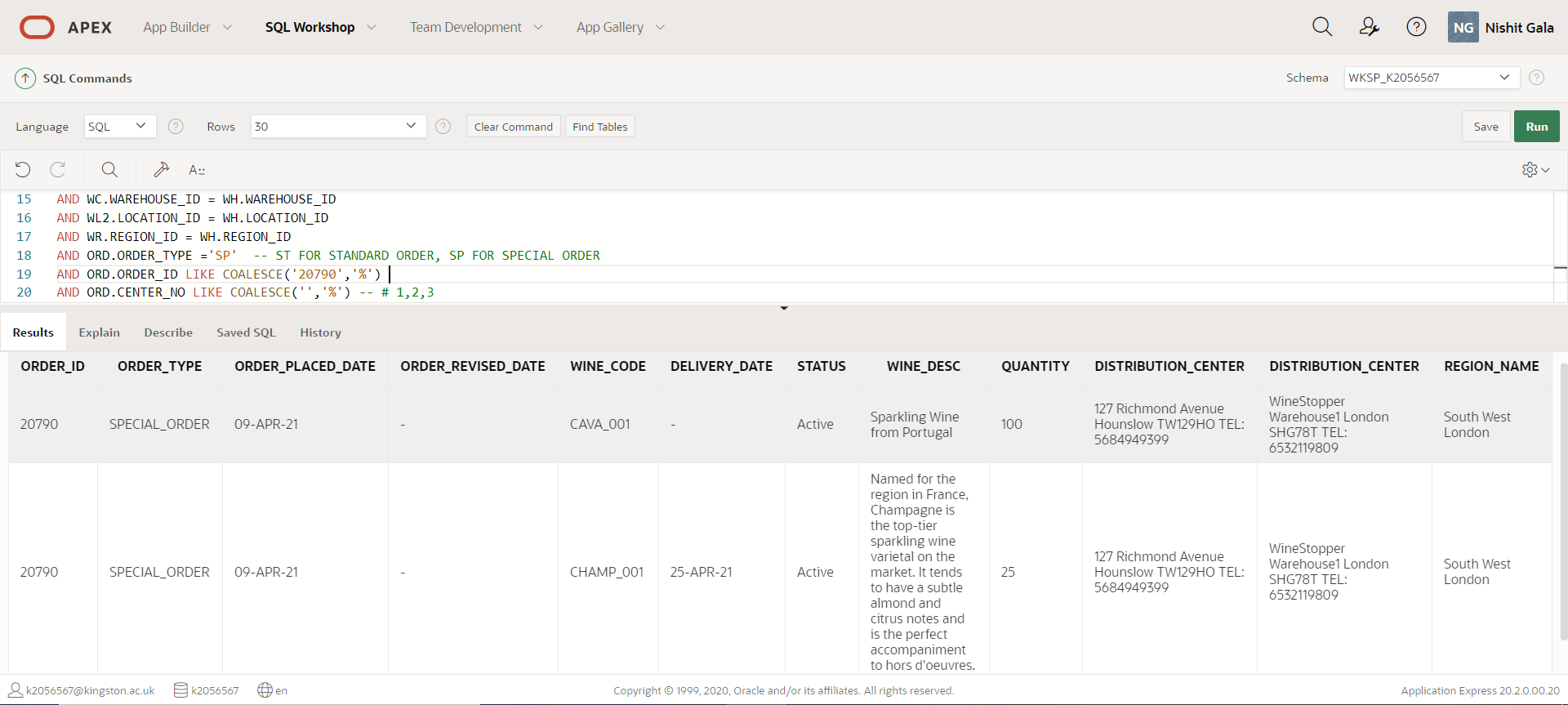
AND WL2.LOCATION\_ID = WH.LOCATION\_ID

AND WR.REGION\_ID = WH.REGION\_ID

AND ORD.ORDER\_TYPE ='SP'  -- ST FOR STANDARD ORDER, SP FOR SPECIAL ORDER

AND ORD.ORDER\_ID LIKE COALESCE('20790','%')

AND ORD.CENTER\_NO LIKE COALESCE('','%') -- # 1,2,3



**Part B – Standard Order**

SELECT  ORD.ORDER\_ID, 'STANDARD\_ORDER' AS "ORDER\_TYPE" ,

ORD.ORDER\_PLACED\_DATE, ORD.ORDER\_REVISED\_DATE, WINE.WINE\_CODE,

ITEM.DELIVERY\_DAY, ITEM.DELIVERY\_FREQUENCY, ITEM.STATUS,

WDET.WINE\_DESC,ITEM.QUANTITY,

WL.STREET\_ADDRESS || '  '  || WL.CITY || '  ' ||   WL.POSTAL\_CODE || '  TEL:  '  || WC.PHONE\_NO AS "DISTRIBUTION\_CENTER",

WL2.STREET\_ADDRESS || '  '  || WL2.CITY || '  ' ||   WL2.POSTAL\_CODE || '  TEL:  '  || WH.PHONE\_NO AS "DISTRIBUTION\_CENTER",

WR.REGION\_NAME

FROM WS\_ORDER ORD, WS\_ORDER\_ITEM ITEM, WS\_WINE WINE, WS\_WINE\_DETAILS WDET,

WS\_CENTER WC, WS\_LOCATION WL, WS\_WAREHOUSE WH, WS\_LOCATION WL2, WS\_REGION WR

WHERE ORD.ORDER\_ID = ITEM.ORDER\_ID

AND WINE.WINE\_CODE = ITEM.WINE\_CODE

AND WDET.WINE\_DTL\_ID = WINE.WINE\_DTL\_ID

AND WC.CENTER\_ID = ORD.CENTER\_NO

AND WL.LOCATION\_ID = WC.LOCATION\_ID

AND WC.WAREHOUSE\_ID = WH.WAREHOUSE\_ID

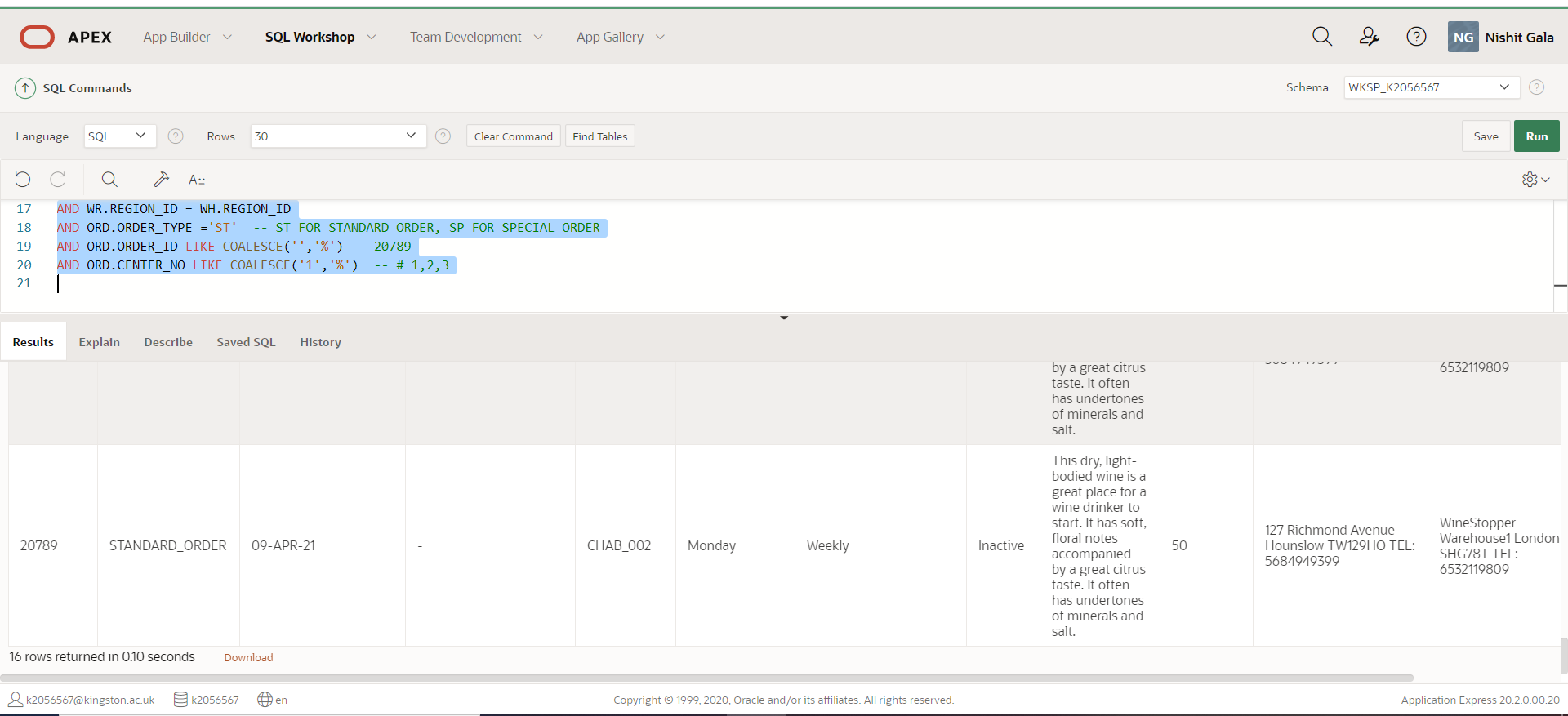
AND WL2.LOCATION\_ID = WH.LOCATION\_ID

AND WR.REGION\_ID = WH.REGION\_ID

AND ORD.ORDER\_TYPE ='ST'  -- ST FOR STANDARD ORDER, SP FOR SPECIAL ORDER

AND ORD.ORDER\_ID LIKE COALESCE('','%') -- 20789

AND ORD.CENTER\_NO LIKE COALESCE('1','%')  -- # 1,2,3



The Output is shown below in the tabular format as exported from Oracle Apex

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ORDER\_ID | ORDER\_TYPE | ORDER\_PLACED\_DATE | ORDER\_REVISED\_DATE | WINE\_CODE | DELIVERY\_DAY | DELIVERY\_FREQUENCY | STATUS | WINE\_DESC | QUANTITY | DISTRIBUTION\_CENTER | DISTRIBUTION\_CENTER | REGION\_NAME |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | MER\_001 | Friday | Monthly | Inactive | merlot is a softer and more accessible red wine. It typically tastes like rich, ripe fruit and can have subtle earthy undertones | 125 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | MER\_004 | Friday | Fortnightly | Active | merlot is a softer and more accessible red wine. It typically tastes like rich, ripe fruit and can have subtle earthy undertones | 100 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | ZIN\_001 | Monday | Monthly | Active | Grown in many California vineyards, red zinfandels have a strong impact on the taste buds. They usually taste like a bold fruit jam with undertones of spices and a smoky finish | 100 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | ZIN\_002 | Monday | Weekly | Inactive | Grown in many California vineyards, red zinfandels have a strong impact on the taste buds. They usually taste like a bold fruit jam with undertones of spices and a smoky finish | 50 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | CHAR\_001 | Friday | Monthly | Active | he more flavorful cousin to Chablis, chardonnay is a medium-to-full-bodied wine. It has strong fruity notes ranging from apple to papaya with hints of oak and vanilla. | 50 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | PINO\_001 | Friday | Monthly | Active | A relative of pinot noir, grigio is a very light and refreshing white wine. It has a citrus taste with salty undertones and a honey-like aroma. | 20 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | PINO\_002 | Friday | Quarterly | Active | A relative of pinot noir, grigio is a very light and refreshing white wine. It has a citrus taste with salty undertones and a honey-like aroma. | 100 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | MOS\_001 | Friday | Monthly | Active | No Description Available | 40 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | MOS\_002 | Monday | Weekly | Active | No Description Available | 25 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | CHAMP\_001 | Monday | Fortnightly | Active | Named for the region in France, Champagne is the top-tier sparkling wine varietal on the market. It tends to have a subtle almond and citrus notes and is the perfect accompaniment to hors d'oeuvres. | 25 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | PROS\_001 | Monday | Weekly | Active | Often considered the second-best sparkling wine,Instead of one region, Prosecco can come from any of nine regions in Italy so it is viewed as less prestigious | 50 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | SHI\_001 | Friday | Quarterly | Active | Dessert Wine (Red) from Greece | 100 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | SAU\_001 | Friday | Monthly | Active | Dessert Wine (White) from Italy | 75 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | CAVA\_001 | Monday | Weekly | Active | Sparkling Wine from Portugal | 100 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | CHAB\_001 | Monday | Weekly | Inactive | This dry, light-bodied wine is a great place for a wine drinker to start. It has soft, floral notes accompanied by a great citrus taste. It often has undertones of minerals and salt. | 25 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |
| 20789 | STANDARD\_ORDER | 9-Apr-21 |  | CHAB\_002 | Monday | Weekly | Inactive | This dry, light-bodied wine is a great place for a wine drinker to start. It has soft, floral notes accompanied by a great citrus taste. It often has undertones of minerals and salt. | 50 | 127 Richmond Avenue Hounslow TW129HO Tel: 5684949399 | Wine-Stopper Warehouse1 London SHG78T Tel: 6532119809 | South West London |

1. **Update Query to update the Shopping List post 60-day period**

Note – This update query needs to run via a CRON JOB on daily basis to clear the Shopping list and set the status as “Expired”

UPDATE WS\_SHOPPING\_LIST SET STATUS = 'Expired' WHERE TRUNC(EXPIRY\_DATE) <= TRUNC(SYSDATE) AND STATUS = 'WishList'

# Chapter 5 – Conclusion

The Wine-Stopper Website has some complex day-to-day operations, and the system is designed to keep in mind that critical points of this application are accurately designed and demonstrated

For this we have considered below assumptions:

1. The promotions considered are at the organization level and if the entry is made in WS\_PROMOTION table, it will be applied to all distribution centers
2. Employee should be mapped to a respective center upon registration by considering the address supplied in the registration process
3. The NEXT\_DELIVERY in WS\_STOCK\_INFO table will only be populated by the Warehouse for Special Orders; this field is going to be null for Standing orders.
4. In case, if the quantity of Stock =0, the above point will lead us to show the Available in X days (X = NEXT\_DELIVERY – SYSDATE) only for the Special Orders and it will be shown as Unavailable for Standing Orders
5. In the WS\_SHOPPING\_BASKET table, the record will be inserted with status as “PENDING” when they are added to basket and the default DISCOUNT will be 0.
6. When the customer makes a purchase, Invoice is generated and will be stored in the INVOICE\_NO column of WS\_SHOPPING\_BASKET table, this will also change the status as “ORDERED” and if there are any applicable discount as per WS\_PROMOTION table for respective WINE\_CODE, then they will be applied
7. The TOTAL\_AMOUNT column in the WS\_INVOICE table will be sum of SELLING\_PRICE from WS\_INVOICE\_ITEM for all the items ordered
8. The SAVINGS column in the WS\_INVOICE table will be sum of DISCOUNT\_AMOUNT from WS\_INVOICE\_ITEM for all the items ordered
9. The DELIVERY\_CHARGE in WS\_INVOICE table will be updated based on rules configured in CALCULATE\_DELIVERY\_CHARGE
10. The payment process is not designed as an Entity and the payment will be reflected upon the Status change in WS\_SHOPPING\_BASKET table once status is changed to “Ordered” and INVOICE\_NO generated

The implementation at high level is described as below:

1. The WS\_WINE\_DETAILS table captures the Name, Colour, Country, Category, Sweetness Indicator of the wine, whilst the WS\_WINE table captures the unique wine code and to cover the possibility of the same wine held in multiple sizes.
2. WS\_WINE table will store the multiple variants in terms of size of case, bottle with a unique Wine Code. Hence this table is related to WS\_WINE\_DETAILS and one Wine record in Wine Details can have multiple Wine codes in the Wine table
3. The customer details are captured in the WS\_CUSTOMER table when they register. This table is linked to the Shopping List (WS\_SHOPPING\_LIST) and Shopping basket (WS\_SHOPPING\_BASKET) along with the Invoice entity (WS\_INVOICE) as shown in the ER Diagram in below section
4. The Shopping List (Wishlist) is captured in the WS\_SHOPPING\_LIST table and there is an expiry date to this table that is defaulted to 60 days from the SYSDATE. It also has a Status column to capture the status as Wishlist, Expired and User Deleted.
5. The Status column of WS\_SHOPPING\_LIST should be updated to “Expired” by a CRON JOB that will run an update statement on daily basis to clear out the Wishlist items that are reaching a 60-day limit.
6. The Shopping Basket (WS\_SHOPPING\_BASKET) is also linked to the Invoice (WS\_INVOICE) table and is having multiple status that controls the flow of the order placed by the customer. This table also records the invoice number for an order placed by the customer.
7. The promotion table (WS\_PROMOTION) is linked to the Wine table and records the wine that are on discounted price. We have taken into consideration the 3 forms of discount as specified below
8. Price Discount (e.g. - 10 % OFF)
9. Quantity Discount (e.g. – Buy 3 Get 4)
10. Reduced Price (Original Price – 3.99$ per bottle, 39.99$ per Case

Reduced Price - Buy at $2.99 per bottle, at 29.99 per case)

1. The information of Wine distribution centers is captured by WS\_CENTER table whilst, the information about the Warehouses is captured in WS\_WAREHOUSE table. The center table is linked to the Warehouse table through the Warehouse ID.
2. The WS\_REGION table will supply the region ID to the WS\_WAREHOUSE table and the Country code to region table will come from WS\_COUNTRY table as defined.
3. The stock information of a distribution center is captured in the WS\_STOCK\_INFO table through the Center ID. The Stock Information table maintains the relationship with the Wine table using Wine Code. These relationships will ensure that each and every Distribution center is stocked as per the Wines configured in the Wine table.
4. The distribution center can place the orders of wine through Standard order also referred to as Standing Order and a special order to request goods that are in demand during a specific time of the year. The Ordering by a distribution center is captured in the WS\_ORDER table
5. The WS\_ORDER table captures both Standing and Specialized orders using a column called as Order type. This Order type column will have a constraint of having fixed values for Standing and Special Order.

The overall conclusion of this exercise is achieved through the list of assumptions provided above , however there is a coding required to set / update certain parameters in the table to correctly reflect the working on the functional flowof the Wine-Stopper day-to-day operations.

The Delivery date field in the stock table is one of the confusion area in this exercise as the statement provided for this field only covers for the special orders as warehouse will update this field, However for the standing orders this field is currently not being updated as this update needs to happen at the regular intervals (monthly, forthnightly, weekly etc). Hence the table needs to be monitored using a CRON Job or a Batch scheduler to achieve this functionality available. This has not been covered in the above solutioning.