

Professor Leavitt

Database 331 Project 2

Daymon Wu

Normalization:

TABLE 1: SUMMONED VEHICLE OWNER

```
CREATE TABLE Summoned_Vehicle_Owner(  
    SUMMONS_NUMBER VARCHAR2(36 Byte),  
    PLATE VARCHAR2(26 Byte)  
);  
  
INSERT INTO Summoned_Vehicle_Owner(SUMMONS_NUMBER, PLATE)  
  
SELECT  
  
    SUMMONS_NUMBER,  
  
    PLATE  
  
FROM VIOLATIONS  
  
--Alter summons_number to be primary key  
  
ALTER TABLE Summoned_Vehicle_Owner  
  
ADD PRIMARY KEY (SUMMONS_NUMBER);
```

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script with the following content:

```
SELECT  
    SUMMONS_NUMBER,  
    PLATE  
FROM VIOLATIONS  
  
COMMIT  
  
SELECT*  
FROM Summoned_Vehicle_Owner  
--Alter summons_number to be primary key  
ALTER TABLE Summoned_Vehicle_Owner  
ADD PRIMARY KEY (SUMMONS_NUMBER);
```

The script is executed, and the results are shown in the 'Query Result' pane. The results are as follows:

SUMMONS_NUMBER	PLATE
1481867179	V44LBT
1481867349	HJP4896
1481871018	LNF3804
1481873738	FWB2961
1481874743	C92MJW
1480348521	KGS7230
1480348557	JMW1426
1480348569	AG72204
1480348715	KJY5703
1480349010	JRC4364
1480349422	GYM6418
1480349471	JRC7360

The status bar at the bottom indicates 'Line: 189 Column: 28 | Insert | Modified | Windows: C'.

PRINTOUT OF TABLE 1:

SUMMONS_NUMBER	PLATE
1481867179	V44LBT
1481867349	HJP4896
1481871018	LNF3804
1481873738	FWB2961
1481874743	C92MJW
1480348521	KGS7230
1480348557	JMW1426
1480348569	AG72204
1480348715	KJY5703
1480349010	JRC4364
1480349422	GYM6418
1480349471	JRC7360
1480349495	AY31507
1480349501	T682740C
1480349562	HYS4820
1480349604	JJM2468
1480349630	HCL1992
1472224887	KGG3094
1480349690	JRW1845
1480350266	934YTM
1480351430	KKG3236
1480351441	HVY8308
1480351570	KEH3049
1480359580	JGZ9947
1480359634	JNB1833
1480359646	KKT1206

TABLE 2: SUMMONS

```
CREATE TABLE Summons(
    SUMMONS_NUMBER VARCHAR2(36 Byte),
    ISSUE_DATE DATE,
    VIOLATION_TIME VARCHAR2(26 Byte),
    VIOLATION VARCHAR2(128 Byte),
    ISSUING_AGENCY VARCHAR2(40 Byte),
    VIOLATION_STATUS VARCHAR2(128 Byte)
```

);

```
INSERT INTO
Summons(SUMMONS_NUMBER,ISSUE_DATE,VIOLATION_TIME,VIOLATION,ISSUING_AGENCY,VIOLATIO
N_STATUS)
```

```
SELECT
```

```
    SUMMONS_NUMBER,
```

```
    ISSUE_DATE,
```

```
    VIOLATION_TIME,
```

```
    VIOLATION,
```

```
    ISSUING_AGENCY,
```

```
    VIOLATION_STATUS
```

```
FROM
```

```
    VIOLATIONS
```

```
ALTER TABLE Summons
```

```
ADD PRIMARY KEY (SUMMONS_NUMBER);
```

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script with the following content:

```
FROM
VIOLATIONS

SELECT*
FROM Summons

ALTER TABLE Summons
ADD PRIMARY KEY (SUMMONS_NUMBER);
```

The query results are displayed in a table with the following columns: SUMMONS_NUMBER, ISSUE_DATE, VIOLATION_TIME, VIOLATION, ISSUING_AGENCY, and VIOLATION_STATUS. The results show 21 rows of data, including various traffic violations such as "FAIL TO DSPLY MUNI METER RECPT", "NO PARKING-DAY/TIME LIMITS", "NO STANDING-BUS STOP", "DOUBLE PARKING", "FRONT OR BACK PLATE MISSING", "NO PARKING-DAY/TIME LIMITS", "NO STANDING-DAY/TIME LIMITS", "NO PARKING-STREET CLEANING", "NO PARKING-DAY/TIME LIMITS", "INSP. STICKER-EXPIRED/MISSING", "FIRE HYDRANT", and "NO STANDING-DAY/TIME LIMITS".

SUMMONS_NUMBER	ISSUE_DATE	VIOLATION_TIME	VIOLATION	ISSUING_AGENCY	VIOLATION_STATUS
21897 8848174346	15-FEB-21	9:19 AM	FAIL TO DSPLY MUNI METER RECPT	TRAFFIC	(null)
21898 8848316153	13-JAN-21	5:23 PM	NO PARKING-DAY/TIME LIMITS	TRAFFIC	(null)
21899 8848318654	05-FEB-21	2:18 PM	NO STANDING-BUS STOP	TRAFFIC	(null)
21900 8848318680	05-FEB-21	3:39 PM	DOUBLE PARKING	TRAFFIC	(null)
21901 8848318691	05-FEB-21	3:44 PM	FRONT OR BACK PLATE MISSING	TRAFFIC	(null)
21902 8848319038	10-FEB-21	3:22 PM	NO PARKING-DAY/TIME LIMITS	TRAFFIC	(null)
21903 8848544708	08-FEB-21	3:56 PM	NO STANDING-DAY/TIME LIMITS	TRAFFIC	(null)
21904 8848721450	12-JAN-21	8:46 AM	NO PARKING-STREET CLEANING	TRAFFIC	(null)
21905 8848723391	22-JAN-21	6:03 AM	NO PARKING-DAY/TIME LIMITS	TRAFFIC	(null)
21906 8848775068	04-JAN-21	2:11 PM	INSP. STICKER-EXPIRED/MISSING	TRAFFIC	HEARING HELD-GUILTY
21907 8848776395	07-JAN-21	2:11 PM	INSP. STICKER-EXPIRED/MISSING	TRAFFIC	HEARING HELD-NOT GUILTY
21908 8848776607	07-JAN-21	6:11 PM	FIRE HYDRANT	TRAFFIC	(null)
21909 8848776693	08-JAN-21	1:24 PM	INSP. STICKER-EXPIRED/MISSING	TRAFFIC	(null)
21910 8848782218	04-FEB-21	4:14 PM	NO STANDING-DAY/TIME LIMITS	TRAFFIC	HEARING HELD-GUILTY
21911 8848782231	04-FEB-21	4:36 PM	NO STANDING-DAY/TIME LIMITS	TRAFFIC	HEARING HELD-GUILTY
21912 8848782954	10-FEB-21	4:15 PM	NO STANDING-DAY/TIME LIMITS	TRAFFIC	HEARING HELD-GUILTY

PRINTOUT OF TABLE 2:

SUMMONS _NUMBER	ISSUE _DATE	VIOLATIO N_TIME	VIOLATION	ISSUING_AGENCY	VIOLATION_STATU S
147178593 2	8- Mar- 21	6:02 AM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	
146614762 3	15- Jan-21	10:01 AM	NO STANDING-EXC. AUTH. VEHICLE	OTHER/UNKNOWN AGENCIES	
147235825 9	5-Feb- 21	3:13 AM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	HEARING HELD- GUILTY
147236262 7	10- Feb- 21	3:00 AM	STORAGE-3HR COMMERCIAL	POLICE DEPARTMENT	
147240893 7	10- Feb- 21	10:25 PM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	
147243578 3	9-Feb- 21	5:39 AM	FIRE HYDRANT	POLICE DEPARTMENT	
147998109 6	16- Feb- 21	7:38 AM	NO PARKING-STREET CLEANING	DEPARTMENT OF SANITATION	
147610881 0	6-Feb- 21	8:20 AM	TRAFFIC LANE	POLICE DEPARTMENT	
147149968 6	2-Jan- 21	2:05 AM	FIRE HYDRANT	POLICE DEPARTMENT	
147785256 6	22- Feb- 21	4:10 AM	FIRE HYDRANT	DEPARTMENT OF SANITATION	
147872839 5	16- Feb- 21	4:12 AM	NO PARKING- DAY/TIME LIMITS	OTHER/UNKNOWN AGENCIES	
147177151 9	22- Feb- 21	7:30 PM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	
147247051 5	7-Feb- 21	8:58 AM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	
147249244 4	10- Feb-	7:13 PM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	

	21				
1476723989	2-Mar-21	4:22 PM	NO STANDING-DAY/TIME LIMITS	POLICE DEPARTMENT	
1471720494	3-Jan-21	8:26 PM	STORAGE-3HR COMMERCIAL	POLICE DEPARTMENT	
1471547887	13-Jan-21	6:21 AM	FIRE HYDRANT	POLICE DEPARTMENT	HEARING HELD-NOT GUILTY
1482394339	9-Mar-21	7:12 PM	NO STANDING-BUS STOP	POLICE DEPARTMENT	
1471775069	4-Jan-21	6:55 PM	NO STANDING-BUS STOP	POLICE DEPARTMENT	HEARING HELD-NOT GUILTY
1482394390	10-Mar-21	5:35 PM	NO STANDING-BUS STOP	POLICE DEPARTMENT	HEARING HELD-NOT GUILTY
1473222552	4-Jan-21	11:30 AM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	
1472301882	1-Jan-21	5:42 PM	INSP. STICKER-EXPIRED/MISSING	POLICE DEPARTMENT	
1472335193	3-Jan-21	10:45 PM	OBSTRUCTING DRIVEWAY	POLICE DEPARTMENT	

TABLE 3: FINES

CREATE TABLE Fines(

SUMMONS_NUMBER VARCHAR2(36 Byte),

PLATE VARCHAR2(26 Byte),

FINE_AMOUNT NUMBER(38,0),

PENALTY_AMOUNT NUMBER(38,0),

INTEREST_AMOUNT NUMBER(38,2),

REDUCTION_AMOUNT NUMBER(38,2),

PAYMENT_AMOUNT NUMBER(38,2),

AMOUNT_DUE NUMBER(38,0)

);

INSERT INTO

Fines(SUMMONS_NUMBER,PLATE,FINE_AMOUNT,PENALTY_AMOUNT,INTEREST_AMOUNT,REDUCTION_AMOUNT,PAYMENT_AMOUNT,AMOUNT_DUE)

SELECT

SUMMONS_NUMBER,

```

PLATE,

TO_CHAR(FINE_AMOUNT,'$999,999,999.99') "FINE_AMOUNT",

TO_CHAR(PENALTY_AMOUNT,'$999,999,999.99') "PENALTY_AMOUNT",

TO_CHAR(INTEREST_AMOUNT,'$999,999,999.99') "INTEREST_AMOUNT",

TO_CHAR(REDUCTION_AMOUNT,'$999,999,999.99') "REDUCTION_AMOUNT",

TO_CHAR(PAYMENT_AMOUNT,'$999,999,999.99') "PAYMENT_AMOUNT",

TO_CHAR(AMOUNT_DUE,'$999,999,999.99') "AMOUNT_DUE"

FROM

VIOLATIONS

ALTER TABLE Summons

ADD PRIMARY KEY (SUMMONS_NUMBER);

```

Oracle SQL Developer: C:\Users\Daymon\Desktop\Daymon.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

Daymon

Tables (Filtered)

- FINES
- PRECINCT
- SMALLVIOLATIONS
- SUMMONED_VEHICLE_OW
- SUMMONS
- VIOLATIONS
- VIOLATIONS1
- VIOLATIONSBYCOUNTY
- VIOLATIONSPOP

Views

Reports

All Reports

- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Worksheet

```

TO_CHAR(REDUCTION_AMOUNT,'$999,999,999.99') "REDUCTION_AMOUNT",
TO_CHAR(PAYMENT_AMOUNT,'$999,999,999.99') "PAYMENT_AMOUNT",
TO_CHAR(AMOUNT_DUE,'$999,999,999.99') "AMOUNT_DUE"
FROM
VIOLATIONS
SELECT*
FROM Fines

```

Script Output x Query Result x

SQL | Fetched 10,750 rows in 0.163 seconds

SUMMONS_NUMBER	PLATE	FINE_AMOUNT	PENALTY_AMOUNT	INTEREST_AMOUNT	REDUCTION_AMOUNT	PAYMENT_AMOUNT	AMOUNT_DUE
10687	8871508506 FZA4718	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10688	8871508531 GTW6087	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10689	8871509250 568AE1	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10690	1418918842 FXY1894	\$95.00	\$0.00	\$0.00	\$0.00	\$95.00	\$0.00
10691	8871855840 I778870C	\$115.00	\$0.00	\$0.00	\$0.00	\$115.00	\$0.00
10692	8871855863 JLF1757	\$115.00	\$0.00	\$0.00	\$0.00	\$115.00	\$0.00
10693	8879757659 FHM2841	\$35.00	\$0.00	\$0.00	\$0.00	\$35.00	\$0.00
10694	8879757921 EJJ28750	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10695	8879758196 AJK6752	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10696	8879758202 KKB6175	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10697	8879758366 I774993C	\$65.00	\$0.00	\$0.00	\$0.00	\$65.00	\$0.00
10698	8879758652 JLA9724	\$35.00	\$0.00	\$0.00	\$0.00	\$35.00	\$0.00
10699	8879900122 JFU1562	\$35.00	\$0.00	\$0.00	\$0.00	\$35.00	\$0.00
10700	8879900316 BOPARAI	\$35.00	\$0.00	\$0.00	\$35.00	\$0.00	\$0.00
10701	8879900365 HMX5087	\$35.00	\$0.00	\$0.00	\$0.00	\$35.00	\$0.00

Click on an identifier with the Control key down to perform "Go to Declaration"

Line: 252 Column: 1 | Insert | Modified | Windows: C

PRINTOUT OF TABLE 3:

SUMMONS_ NUMBER	PLATE	FINE_A MOUNT	PENALTY_A MOUNT	INTEREST_ AMOUNT	REDUCTION_ AMOUNT	PAYMENT_ AMOUNT	AMOUN T_DUE
1481867179	V44LBT	\$95.00	\$0.00	\$0.00	\$0.00	\$95.00	\$0.00
1481867349	HJP489	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00

	6						
1481871018	LNF3804	\$95.00	\$0.00	\$0.00	\$0.00	\$95.00	\$0.00
1481873738	FWB2961	\$115.00	\$0.00	\$0.00	\$0.00	\$115.00	\$0.00
1481874743	C92MJW	\$115.00	\$0.00	\$0.00	\$0.00	\$115.00	\$0.00
1480348521	KGS7230	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480348557	JMW1426	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480348569	AG72204	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480348715	KJY5703	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349010	JRC4364	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349422	GYM6418	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349471	JRC7360	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349495	AY31507	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349501	T682740C	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349562	HYS4820	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349604	JJM2468	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480349630	HCL1992	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1472224887	KGG3094	\$95.00	\$0.00	\$0.00	\$0.00	\$95.00	\$0.00
1480349690	JRW1845	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480350266	934YTM	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480351430	KKG3236	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480351441	HVY8308	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480351570	KEH3049	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480359580	JGZ9947	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480359634	JNB1833	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00

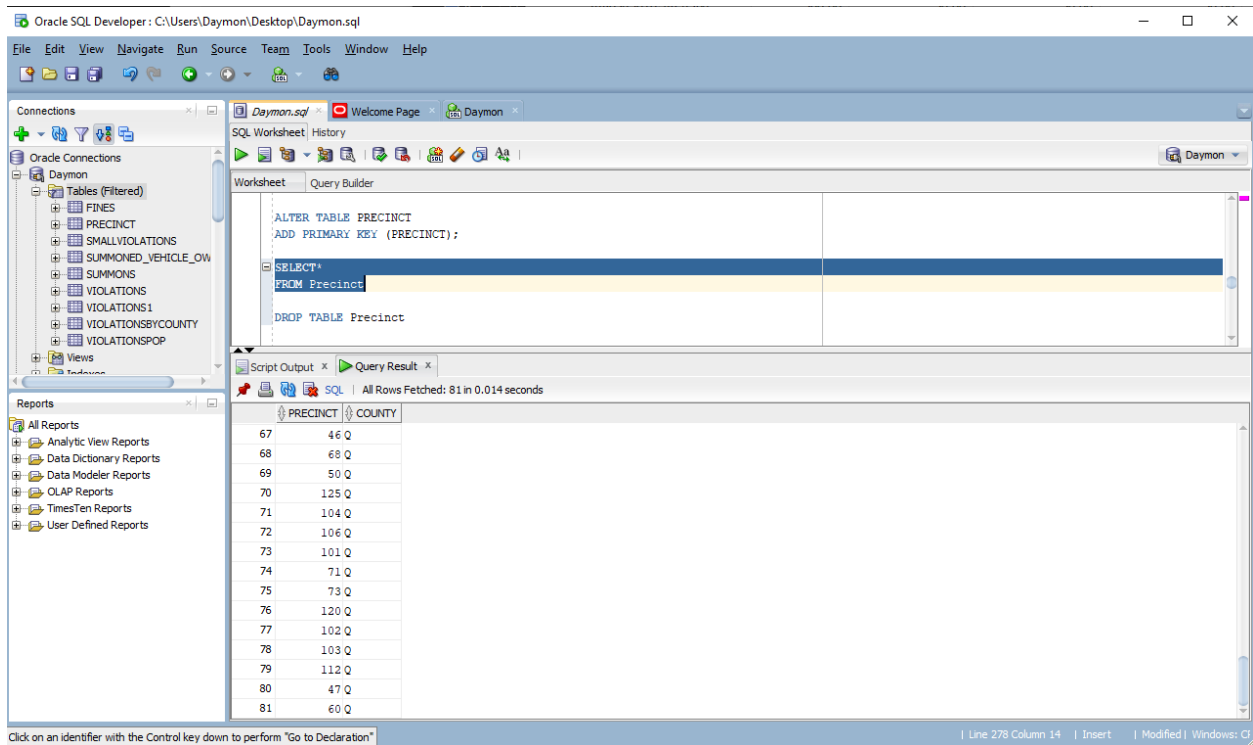
1480359646	KKT120 6	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480359853	FER132 0	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480360843	JCJ6076	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480360934	FPA155 1	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480360983	JMZ847 2	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1480365312	JRU223 0	\$60.00	\$0.00	\$0.00	\$0.00	\$60.00	\$0.00
1307574671	KFM55 58	\$95.00	\$0.00	\$0.00	\$0.00	\$95.00	\$0.00

TABLE 4: PRECINCT

```

CREATE TABLE Precinct(
    PRECINCT NUMBER(38,0),
    COUNTY VARCHAR2(26 BYTE)
);
INSERT INTO Precinct(PRECINCT,COUNTY)
SELECT
DISTINCT
    PRECINCT,
    COUNTY
FROM VIOLATIONS
ALTER TABLE PRECINCT
ADD PRIMARY KEY (PRECINCT);

```

PRINTOUT OF TABLE 4:

PRECINCT	COUNTY
108	Q
107	Q
61	Q
119	Q
9	Q
100	Q
75	Q
14	Q
170	Q
26	Q
72	Q
105	Q
83	Q
116	Q
44	Q
174	Q
19	Q
0	Q
25	Q

165	Q
114	Q
118	Q
70	Q
40	Q
63	Q
77	Q

TABLE 5: VEHICLE_INFO

```

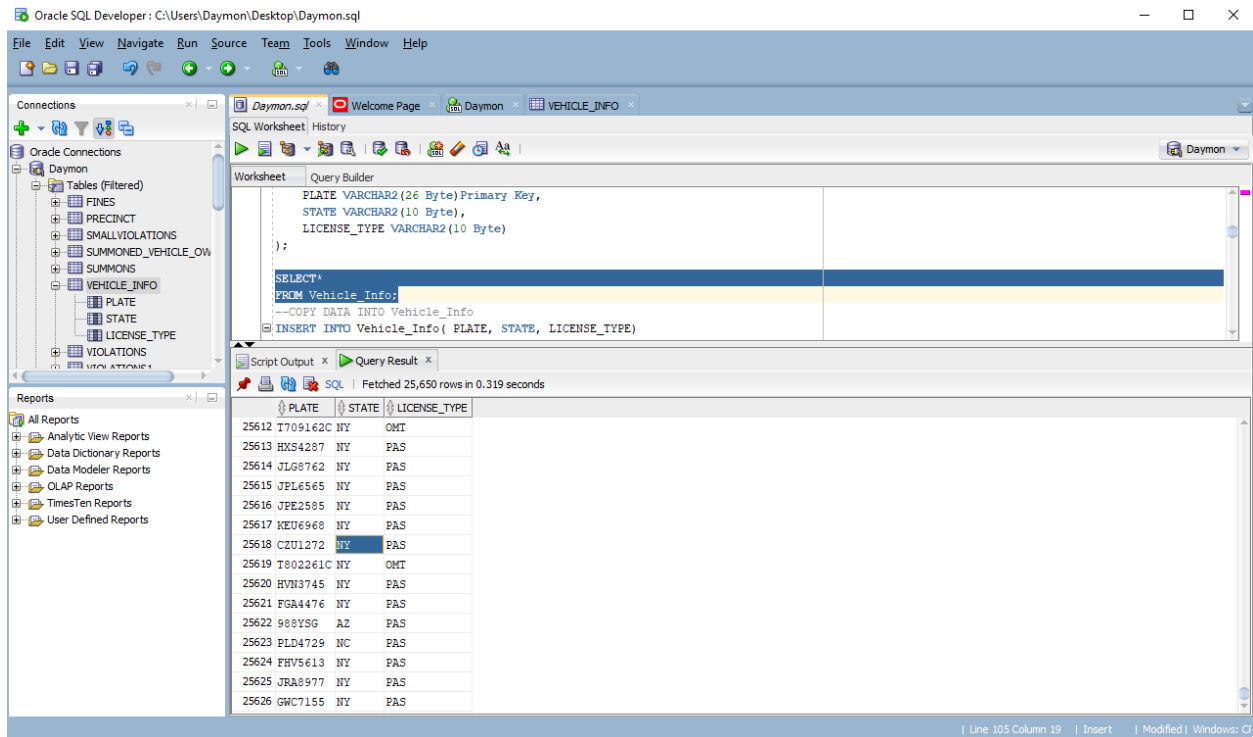
CREATE TABLE Vehicle_Info(
    PLATE VARCHAR2(26 Byte)Primary Key,
    STATE VARCHAR2(10 Byte),
    LICENSE_TYPE VARCHAR2(10 Byte)
);

INSERT INTO Vehicle_Info( PLATE, STATE, LICENSE_TYPE)

SELECT
DISTINCT
    PLATE,
    STATE,
    LICENSE_TYPE
FROM
    VIOLATIONS

ALTER TABLE Vehicle_Info
ADD PRIMARY KEY(PLATE);

```



PRINTOUT OF TABLE 5:

PLATE	STATE	LICENSE_TYPE
JJM6670	NY	PAS
JEJ4142	NY	PAS
JDZ7865	NY	PAS
JMU5567	NY	PAS
JPD8137	NY	PAS
LLW8518	PA	PAS
T765242C	NY	OMT
GPK9309	NY	PAS
GTY1542	NY	PAS
KDD1813	NY	PAS
JRC7603	NY	PAS
M068821	OH	PAS
JAU8279	NY	PAS
GXF9181	NY	PAS
DCS8240	NY	PAS
HZK4891	NY	PAS
F70LYM	NJ	PAS
ELL4422	NY	PAS
HVY7042	NY	PAS
HAC2627	NY	PAS

T709493C	NY	OMT
JRC7027	NY	PAS
JFG7449	NY	PAS
EAZ8950	NY	PAS
HYX9917	NY	PAS
JFZ8187	NY	PAS

QUESTIONS:

1. Identify the number of violations for a 6-month period. Display 1 row with the number of violations.

SELECT

EXTRACT(MONTH FROM ISSUE_DATE) AS month,

COUNT(1) AS count

FROM VIOLATIONS

GROUP BY

EXTRACT(MONTH FROM ISSUE_DATE)

ORDER BY 1 ASC

FETCH FIRST 6 ROWS ONLY

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
--ANSWER--ANSWER--ANSWER--ANSWER--ANSWER--ANSWER
SELECT
  --EXTRACT(YEAR FROM ISSUE_DATE) AS year,
  EXTRACT(MONTH FROM ISSUE_DATE) AS month,
  COUNT(1) AS count
FROM VIOLATIONS
GROUP BY
  --EXTRACT(YEAR FROM ISSUE_DATE),
  EXTRACT(MONTH FROM ISSUE_DATE)
ORDER BY 1 ASC
FETCH FIRST 6 ROWS ONLY

--ANSWER--ANSWER--ANSWER--ANSWER--ANSWER--ANSWER
SELECT (MAX(ISSUE_DATE)) "TIME", COUNT(1) "Number Of Violations"
FROM VIOLATIONS
WHERE ISSUE_DATE >= add_months(MAX(ISSUE_DATE), -6)
```

The Results window at the bottom shows the output of the query, which is a table with two columns: MONTH and COUNT. The data is as follows:

MONTH	COUNT
1	81811
2	50250
3	90890
4	55001
5	36797
6	48546

MONTH COUNT

1	81811
2	50250
3	90890
4	55001
5	36797
6	48546

2. Identify the violations by county. Display 3 columns: county, number of violations, total fine amount. The county with the largest number of violations is displayed first. Display 1 row for each distinct county.

```

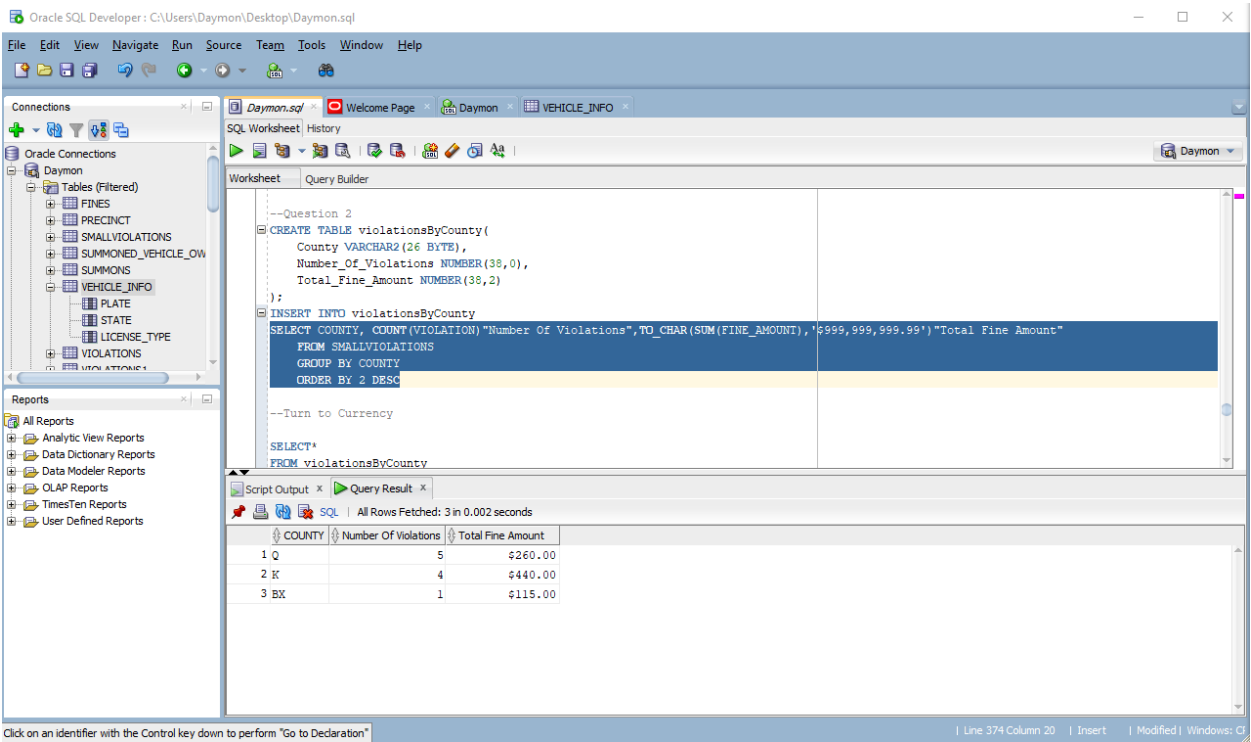
SELECT COUNTY, COUNT(VIOLATION)"Number Of
Violations",TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999.99')"Total Fine Amount"

FROM SMALLVIOLATIONS

GROUP BY COUNTY

ORDER BY 2 DESC

```



COUNTY	Number Of Violations	Total Fine Amount
Q	5	\$260.00
K	4	\$440.00
BX	1	\$115.00

3. Identify the violations by license plate. Display 4 columns: License plate, license state, number of violations and total cost. The license plate with the most violations is displayed first. Display 1 row for each distinct license plate.

```
SELECT PLATE "License Plate", "STATE", TO_CHAR(COUNT(1), '999,999,999') "Number Of Violations",
       TO_CHAR(SUM(FINE_AMOUNT), '$999,999,999') "Total Fine Amount"

FROM VIOLATIONS

GROUP BY PLATE, "STATE"

ORDER BY 4 DESC
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
--Question 3
--CREATE TABLE violationsByPlate(License_Plate,License_State,Number_Of_Violations,Total_Cost)
--Missing State
SELECT PLATE "License Plate", "STATE", TO_CHAR(COUNT(1), '999,999,999') "Number Of Violations",
       TO_CHAR(SUM(FINE_AMOUNT), '$999,999,999') "Total Fine Amount"
FROM VIOLATIONS
GROUP BY PLATE, "STATE"
ORDER BY 4 DESC
```

The Query Results tab shows the following data:

License Plate	STATE	Number Of Violations	Total Fine Amount
1 BLANKPLATE	99	1,164	\$98,628
2 12125MJ	NY	386	\$41,830
3 86145MM	NY	282	\$30,720
4 83460MH	NY	275	\$30,155
5 85884MD	NY	261	\$27,115
6 41950JX	NY	228	\$23,695
7 92255JJ	NY	198	\$21,745
8 91291MD	NY	190	\$20,110
9 56210MG	NY	182	\$18,425
10 56211MG	NY	157	\$17,185
11 30353NA	NY	160	\$16,060
12 30329NA	NY	148	\$14,790
13 29148ML	NY	144	\$14,715
14 47975KA	NY	156	\$14,610

License Plate	STATE	Number Of Violations	Total Fine Amount
BLANKPLATE	99	1,164	\$98,628
12125MJ	NY	386	\$41,830
86145MM	NY	282	\$30,720
83460MH	NY	275	\$30,155
85884MD	NY	261	\$27,115

41950JX	NY	228	\$23,695
92259JJ	NY	198	\$21,745
91291MD	NY	190	\$20,110
56210MG	NY	182	\$18,425
56211MG	NY	157	\$17,185
30353NA	NY	160	\$16,060
30329NA	NY	148	\$14,790
29148ML	NY	144	\$14,715
47975KA	NY	156	\$14,610
92253JJ	NY	145	\$14,085
75142MB	NY	147	\$13,985
66788ME	NY	131	\$13,245
41962JX	NY	121	\$11,940
44184MK	NY	124	\$11,890
95492MD	NY	119	\$11,285
12039MJ	NY	103	\$10,125
88502JU	NY	127	\$10,125
62189MM	NY	96	\$9,840
28559MH	NY	101	\$9,790
85849MD	NY	93	\$9,695
66783ME	NY	115	\$9,565
62191MM	NY	90	\$9,490
41825JX	NY	88	\$9,075
41958JX	NY	91	\$9,030
12922JB	NY	87	\$8,860
22215MM	NY	82	\$8,660
44119MK	NY	94	\$8,570
62227MM	NY	79	\$8,530
26302TC	NY	98	\$8,500
44139MK	NY	80	\$8,270
30248NA	NY	80	\$8,225
60304MD	NY	79	\$8,015
88356JX	NY	70	\$7,400
80635JW	NY	88	\$7,395
59907MJ	NY	80	\$7,385
58915MM	NY	75	\$7,065
11761ME	NY	115	\$6,785
58912MM	NY	67	\$6,705
52511JZ	NY	74	\$6,695
12349JB	NY	62	\$6,680
29159ML	NY	69	\$6,670
12343MG	NY	60	\$6,345
22171MM	NY	73	\$6,205
12341MG	NY	64	\$6,200
89365MJ	NY	64	\$6,160
44141MK	NY	60	\$6,120
12121MJ	NY	63	\$6,115
86008JV	NY	57	\$6,105

22212MM	NY	58	\$6,055
86615MG	NY	66	\$6,020
18420JU	NY	60	\$6,020
67655MC	NY	58	\$6,020
62273MM	NY	64	\$6,010

4. Identify the violations by license type. Display 3 columns: license type, number of violations and total fine amount. The license type with the most violations is displayed first. Display 1 row for each distinct license type.

```

SELECT LICENSE_TYPE, TO_CHAR(COUNT(1),'999,999,999')"Number Of Violations",
      TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"
FROM VIOLATIONS
GROUP BY LICENSE_TYPE
ORDER BY 2 DESC

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows a connection to 'Daymon'. The 'Script Output' pane at the bottom displays the results of the query. The results are as follows:

	LICENSE_TYPE	Number Of Violations	Total Fine Amount
1	PAS	411,907	\$29,381,961
2	COM	49,187	\$3,885,706
3	OMT	15,677	\$1,152,998
4	SRF	6,079	\$400,640
5	MOT	3,318	\$218,669
6	OMS	3,071	\$212,465
7	999	3,055	\$241,666
8	ORG	1,162	\$76,625
9	TRC	1,151	\$102,765
10	APP	1,109	\$96,945
11	SPO	788	\$53,855
12	IMB	606	\$43,245
13	MED	419	\$23,310
14	RGL	380	\$25,310

LICENSE_TYPE	Number Of Violations	Total Fine Amount
PAS	411,907	\$29,381,961
COM	49,187	\$3,885,706
OMT	15,677	\$1,152,998
SRF	6,079	\$400,640
MOT	3,318	\$218,669
OMS	3,071	\$212,465

	999	3,055	\$241,666
ORG		1,162	\$76,625
TRC		1,151	\$102,765
APP		1,109	\$96,945
SPO		788	\$53,855
LMB		606	\$43,245
MED		419	\$23,310
RGL		380	\$25,310
OMR		362	\$22,845
TRL		248	\$13,670
CMB		247	\$17,595
TOW		241	\$18,355
ITP		139	\$9,350
DLR		138	\$9,020
OML		129	\$9,770
HIS		104	\$6,900
SCL		87	\$4,465
VAS		61	\$3,885
TRA		61	\$5,190
SRN		53	\$3,395
IRP		50	\$5,770
MCL		37	\$2,560
MCD		25	\$1,715
OMV		14	\$685
SEM		12	\$590
STG		9	\$545
HAM		8	\$565
AGR		8	\$615
PHS		8	\$560
BOB		7	\$575
SPC		6	\$570
LTR		5	\$245
NLM		4	\$260
AYG		4	\$310
ARG		4	\$310
SOS		3	\$225
RGC		3	\$160
NYC		2	\$230
NYS		2	\$130
CHC		2	\$180
BOT		1	\$65
CSP		1	\$115
LUA		1	\$115

AMB	1	\$35
CMH	1	\$65
LMC	1	\$65
AGC	1	\$65

5. Identify violations by license plate state. Display 3 columns: state, number of violations and total fine amount. The state with the most violations is displayed first. Display 1 row for each distinct state.

```

SELECT STATE,TO_CHAR(COUNT(1),'999,999,999')"Number Of Violations",
        TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"

FROM VIOLATIONS

GROUP BY STATE

ORDER BY 2 DESC

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'Daymon' connection. The 'Worksheet' pane contains the following SQL query:

```

--Question 5
--CREATE TABLE violationsByLicenseState(License_State,Number_Of_Violations>Total_Fine_Amount)
SELECT STATE,TO_CHAR(COUNT(1),'999,999,999')"Number Of Violations",
        TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"
FROM VIOLATIONS
GROUP BY STATE
ORDER BY 2 DESC

```

The 'Query Result' pane shows the following results:

STATE	Number Of Violations	Total Fine Amount
1 NY	425,371	\$30,282,080
2 NJ	21,821	\$1,642,415
3 PA	9,429	\$729,680
4 FL	7,308	\$550,780
5 CT	5,214	\$392,465
6 TX	3,489	\$296,580
7 MA	2,672	\$199,580
8 99	2,471	\$220,518
9 VA	2,335	\$182,795
10 IN	2,082	\$181,040
11 MD	1,863	\$136,945
12 NC	1,704	\$126,747
13 CA	1,505	\$130,300
14 ME	1,449	\$105,745

STATE	Number Of Violations	Total Fine Amount
NY	425,371	\$30,282,080
NJ	21,821	\$1,642,415
PA	9,429	\$729,680
FL	7,308	\$550,780
CT	5,214	\$392,465
TX	3,489	\$296,580
MA	2,672	\$199,580
99	2,471	\$220,518
VA	2,335	\$182,795

IN	2,082	\$181,040
MD	1,863	\$136,945
NC	1,704	\$126,747
CA	1,505	\$130,300
ME	1,449	\$105,745
IL	1,392	\$115,795
GA	1,262	\$96,435
OH	949	\$73,665
AZ	855	\$67,505
TN	759	\$55,465
SC	673	\$53,095
MI	556	\$41,795
WI	471	\$40,210
DE	436	\$33,565
RI	363	\$26,685
WA	363	\$27,950
AL	294	\$22,820
NH	281	\$20,530
OK	232	\$18,175
MN	230	\$18,055
VT	207	\$14,795
CO	190	\$14,790
MO	187	\$14,915
KY	168	\$12,835
OR	153	\$13,040
LA	135	\$10,270
ON	114	\$8,115
MS	100	\$7,175
IA	89	\$6,820
AR	88	\$6,600
WV	88	\$6,645
SD	82	\$7,355
UT	63	\$5,280
KS	59	\$4,520
NE	58	\$4,475
DC	57	\$4,500
NV	54	\$4,655
NM	43	\$3,350
DP	39	\$2,960
MT	39	\$3,290
QB	34	\$2,350
ID	26	\$2,105
WY	25	\$2,060
ND	21	\$1,535
HI	13	\$870
AK	11	\$960
AB	7	\$650
PR	5	\$350

NB	5	\$375
GV	3	\$245
BC	3	\$345
FO	2	\$70
NT	2	\$160
NS	1	\$115

6. Identify FIRE HYDRANT violations by JAN. Display 2 columns: JAN and number of violations. Order by JAN.

SELECT VIOLATION,ISSUE_DATE

FROM VIOLATIONS

WHERE VIOLATION like '%FIRE HYDRANT'

AND ISSUE_DATE like '%JAN%'

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows a connection to 'Daymon'. The 'Worksheet' pane in the center contains the following SQL query:

```
--Question 6
--CREATE TABLE doubleParkingViolations(Month_Of_Violation,Number_Of_Violations)
--Monthly
SELECT VIOLATION,ISSUE_DATE
FROM VIOLATIONS
WHERE VIOLATION like '%FIRE HYDRANT'
AND ISSUE_DATE like '%JAN%'

--Question 7
--CREATE TABLE lastSixMonths(Violation,Number_Of_Violations,Total_Cost)
--FIRST VIOLATION TO CHAR(CONCAT('1 1999 999 999'))"Number_Of_Violations" TO CHAR(CONCAT('1 1999 999 999'))"Total_Fine_Amount"
```

The 'Query Result' pane at the bottom shows the results of the query, fetched in 0.018 seconds. The results are as follows:

VIOLATION	ISSUE_DATE
1 FIRE HYDRANT	01-JAN-21
2 FIRE HYDRANT	23-JAN-21
3 FIRE HYDRANT	22-JAN-21
4 FIRE HYDRANT	14-JAN-21
5 FIRE HYDRANT	16-JAN-21
6 FIRE HYDRANT	16-JAN-21
7 FIRE HYDRANT	10-JAN-21
8 FIRE HYDRANT	10-JAN-21
9 FIRE HYDRANT	17-JAN-21
10 FIRE HYDRANT	03-JAN-21
11 FIRE HYDRANT	17-JAN-21
12 FIRE HYDRANT	02-JAN-21
13 FIRE HYDRANT	13-JAN-21
14 FIRE HYDRANT	09-JAN-21

VIOLATION	ISSUE_DATE
FIRE HYDRANT	1-Jan-21
FIRE HYDRANT	23-Jan-21
FIRE HYDRANT	22-Jan-21
FIRE HYDRANT	14-Jan-21
FIRE HYDRANT	16-Jan-21
FIRE HYDRANT	16-Jan-21
FIRE HYDRANT	10-Jan-21
FIRE HYDRANT	10-Jan-21

FIRE HYDRANT	17-Jan-21
FIRE HYDRANT	3-Jan-21
FIRE HYDRANT	17-Jan-21
FIRE HYDRANT	2-Jan-21
FIRE HYDRANT	13-Jan-21
FIRE HYDRANT	9-Jan-21
FIRE HYDRANT	22-Jan-21
FIRE HYDRANT	17-Jan-21
FIRE HYDRANT	6-Jan-21
FIRE HYDRANT	3-Jan-21
FIRE HYDRANT	7-Jan-21
FIRE HYDRANT	7-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	18-Jan-21
FIRE HYDRANT	20-Jan-21
FIRE HYDRANT	31-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	3-Jan-21
FIRE HYDRANT	6-Jan-21
FIRE HYDRANT	7-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	27-Jan-21
FIRE HYDRANT	7-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	27-Jan-21
FIRE HYDRANT	2-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	1-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	23-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	5-Jan-21
FIRE HYDRANT	3-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	2-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	4-Jan-21
FIRE HYDRANT	2-Jan-21

FIRE HYDRANT
FIRE HYDRANT
FIRE HYDRANT

5-Jan-21
6-Jan-21
6-Jan-21

7. Identify violations for the last 6 months. Display 3 columns: violation, number of violations and total cost. The most violations will be displayed first. Display 1 row for each distinct violation.

```
SELECT VIOLATION, TO_CHAR(COUNT(1),'999,999,999')"Number Of Violations",  
TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"
```

FROM VIOLATIONS

GROUP BY VIOLATION

ORDER BY 2 DESC

The screenshot shows the Oracle SQL Developer interface. The 'Query Result' window displays the following data:

VIOLATION	Number Of Violations	Total Fine Amount
1 NO PARKING-STREET CLEANING	92,038	\$5,982,290
2 FAIL TO DSPLY MUNI METER RECPT	87,479	\$3,061,765
3 INSP. STICKER-EXPIRED/MISSING	41,349	\$2,687,685
4 FIRE HYDRANT	40,271	\$4,631,165
5 NO PARKING-DAY/TIME LIMITS	35,213	\$2,112,790
6 NO STANDING-DAY/TIME LIMITS	35,035	\$4,029,025
7 REG. STICKER-EXPIRED/MISSING	29,315	\$1,905,475
8 EXPIRED MUNI METER	26,996	\$944,860
9 NO STANDING-BUS STOP	16,591	\$1,907,965
10 FRONT OR BACK PLATE MISSING	15,665	\$1,018,225
11 OBSTRUCTING DRIVEWAY	14,170	\$1,345,601
12 DOUBLE PARKING	10,671	\$1,227,165
13 NO STANDING-EXC. TRUCK LOADING	8,181	\$777,195
14 CROSSWALK	7,149	\$822,135

VIOLATION	Number Of Violations	Total Fine Amount
NO PARKING-STREET CLEANING	92,038	\$5,982,290
FAIL TO DSPLY MUNI METER RECPT	87,479	\$3,061,765
INSP. STICKER-EXPIRED/MISSING	41,349	\$2,687,685
FIRE HYDRANT	40,271	\$4,631,165
NO PARKING-DAY/TIME LIMITS	35,213	\$2,112,790
NO STANDING-DAY/TIME LIMITS	35,035	\$4,029,025
REG. STICKER-EXPIRED/MISSING	29,315	\$1,905,475
EXPIRED MUNI METER	26,996	\$944,860

NO STANDING-BUS STOP	16,591	\$1,907,965
FRONT OR BACK PLATE MISSING	15,665	\$1,018,225
OBSTRUCTING DRIVEWAY	14,170	\$1,345,601
DOUBLE PARKING	10,671	\$1,227,165
NO STANDING-EXC. TRUCK LOADING	8,181	\$777,195
CROSSWALK	7,149	\$822,135
NGHT PKG ON RESID STR-COMM VEH	5,511	\$358,215
SIDEWALK	4,623	\$531,386
NO STANDING-EXC. AUTH. VEHICLE	3,729	\$354,255
NO STANDING-COMM METER ZONE	2,900	\$333,500
BIKE LANE	1,822	\$209,510
SAFETY ZONE	1,534	\$176,210
STORAGE-3HR COMMERCIAL	1,453	\$94,445
WRONG WAY	1,355	\$60,975
PLTFRM LFTS LWRD POS COMM VEH	1,222	\$54,990
DETACHED TRAILER	1,173	\$52,780
INSP STICKER-MUTILATED/C'FEIT	1,161	\$75,465
NON-COMPLIANCE W/ POSTED SIGN	1,116	\$66,960
NO MATCH-PLATE/STICKER	1,092	\$70,977
NIGHTTIME STD/ PKG IN A PARK	1,090	\$103,488
PEDESTRIAN RAMP	1,025	\$169,095
COMML PLATES-UNALTERED VEHICLE	1,011	\$116,265
NO STANDING-TAXI STAND	898	\$103,270
NO STOPPING-DAY/TIME LIMITS	852	\$97,873
ANGLE PARKING	666	\$29,970
OTHER	637	\$48,895
NO STANDING-BUS LANE	525	\$60,375
OVERNIGHT TRACTOR TRAILER PKG	465	\$120,250
FAIL TO DISP. MUNI METER RECPT	464	\$16,240
BEYOND MARKED SPACE	372	\$16,797
NO PARKING-EXC. HOTEL LOADING	308	\$35,370
NO PARKING-EXC. HNDICAP PERMIT	296	\$53,280
FEEDING METER	281	\$18,265
REG STICKER-MUTILATED/C'FEIT	263	\$17,095
PARKED BUS-EXC. DESIG. AREA	239	\$10,755
TRAFFIC LANE	202	\$23,230
IMPROPER REGISTRATION	181	\$11,765
NO PARKING-EXC. AUTH. VEHICLE	146	\$8,760
MISSING EQUIPMENT	139	\$6,255
DIVIDED HIGHWAY	136	\$15,640
VEHICLE FOR SALE(DEALERS ONLY)	133	\$5,985
NO STANDING-COMMUTER VAN STOP	128	\$14,720
VIN OBSCURED	110	\$7,145

INTERSECTION	94	\$10,810
OVERTIME PKG-TIME LIMIT POSTED	89	\$5,320
OBSTRUCTING TRAFFIC/INTERSECT	72	\$8,240
NO STANDING-HOTEL LOADING	59	\$6,785
	49	\$1,540
NO STANDING-FOR HIRE VEH STOP	47	\$5,405
EXPIRED MUNI MTR-COMM MTR ZN	43	\$1,505
UNAUTHORIZED BUS LAYOVER	35	\$4,025
EXCAVATION-VEHICLE OBSTR TRAFF	26	\$2,440
IDLING	22	\$2,530
RAILROAD CROSSING	17	\$1,529
UNALTERED COMM VEHICLE	14	\$910
WASH/REPAIR VEHCL-REPAIR ONLY	7	\$315
VEH-SALE/WSHNG/RPRNG/DRIVEWAY	7	\$175
OVERTIME STDG D/S	7	\$434
VACANT LOT	6	\$270
SELLING/OFFERING MCHNDSE-METER	5	\$175
ANGLE PARKING-COMM VEHICLE	5	\$575
BUS PARKING IN LOWER MANHATTAN	4	\$420
NO STANDING EXCP D/S	3	\$225
NO PARKING-TAXI STAND	2	\$120
PKG IN EXC. OF LIM-COMM MTR ZN	1	\$35
NO STANDING-OFF-STREET LOT	1	\$115
BUS LANE VIOLATION	1	\$115
EXPIRED METER-COMM METER ZONE	1	\$35
NO STANDING-SNOW EMERGENCY	1	\$115
EXPIRED METER	1	\$35

8. Identify total fines for double parking. Display the total fines.

```
SELECT VIOLATION,TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"
FROM VIOLATIONS
WHERE VIOLATION like '%DOUBLE PARKING%'
GROUP BY VIOLATION
```

The screenshot displays the Oracle SQL Developer interface. The 'Connections' pane on the left shows a connection to 'Daymon'. The 'SQL Worksheet' pane contains the following SQL query:

```
ORDER BY 2 DESC
--Question 8
--CREATE TABLE totalFines(Total_Fines)
SELECT VIOLATION,TO_CHAR(SUM(FINE_AMOUNT),'$999,999,999')"Total Fine Amount"
FROM VIOLATIONS
WHERE VIOLATION like '%DOUBLE PARKING%'
GROUP BY VIOLATION
--Question 9
```

The 'Query Result' pane at the bottom shows the results of the query, which has fetched 1 row in 0.079 seconds. The results are as follows:

VIOLATION	Total Fine Amount
1 DOUBLE PARKING	\$1,227,165

VIOLATION	Total Fine Amount
DOUBLE PARKING	\$1,227,165

9. Perform an analysis of your own choosing.

Identify the violations with keyword “parking” in OCTOBER. DISPLAY two columns: the Violation and number of violations.

```
SELECT VIOLATION, TO_CHAR(COUNT(1),'999,999,999') "Number Of Violations"
FROM VIOLATIONS
WHERE VIOLATION like '%NO PARKING%'
AND ISSUE_DATE like '%OCT%'
GROUP BY VIOLATION
ORDER BY 2 DESC
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'Daymon' selected, and the 'Tables (Filtered)' list showing 'VIOLATIONS'. The main window shows a SQL worksheet with the following query:

```
--Question 9
SELECT VIOLATION, TO_CHAR(COUNT(1),'999,999,999') "Number Of Violations"
FROM VIOLATIONS
WHERE VIOLATION like '%NO PARKING%'
AND ISSUE_DATE like '%OCT%'
GROUP BY VIOLATION
ORDER BY 2 DESC

SELECT
--EXTRACT(YEAR FROM ISSUE_DATE) AS year,
EXTRACT(MONTH FROM ISSUE_DATE) AS month,
COUNT(1) AS count
```

The 'Query Result' pane shows the following data:

VIOLATION	Number Of Violations
1 NO PARKING-STREET CLEANING	2,974
2 NO PARKING-DAY/TIME LIMITS	1,190
3 NO PARKING-EXC. HOTEL LOADING	12
4 NO PARKING-EXC. HANDICAP PERMIT	7
5 NO PARKING-EXC. AUTH. VEHICLE	6

VIOLATION	Number Of Violations
NO PARKING-STREET CLEANING	2,974
NO PARKING-DAY/TIME LIMITS	1,190
NO PARKING-EXC. HOTEL LOADING	12
NO PARKING-EXC. HANDICAP PERMIT	7
NO PARKING-EXC. AUTH. VEHICLE	6

10. Display the structure of ALL tables using SQL Describe.

TABLE 1: SUMMONED_VEHICLE_OWNER

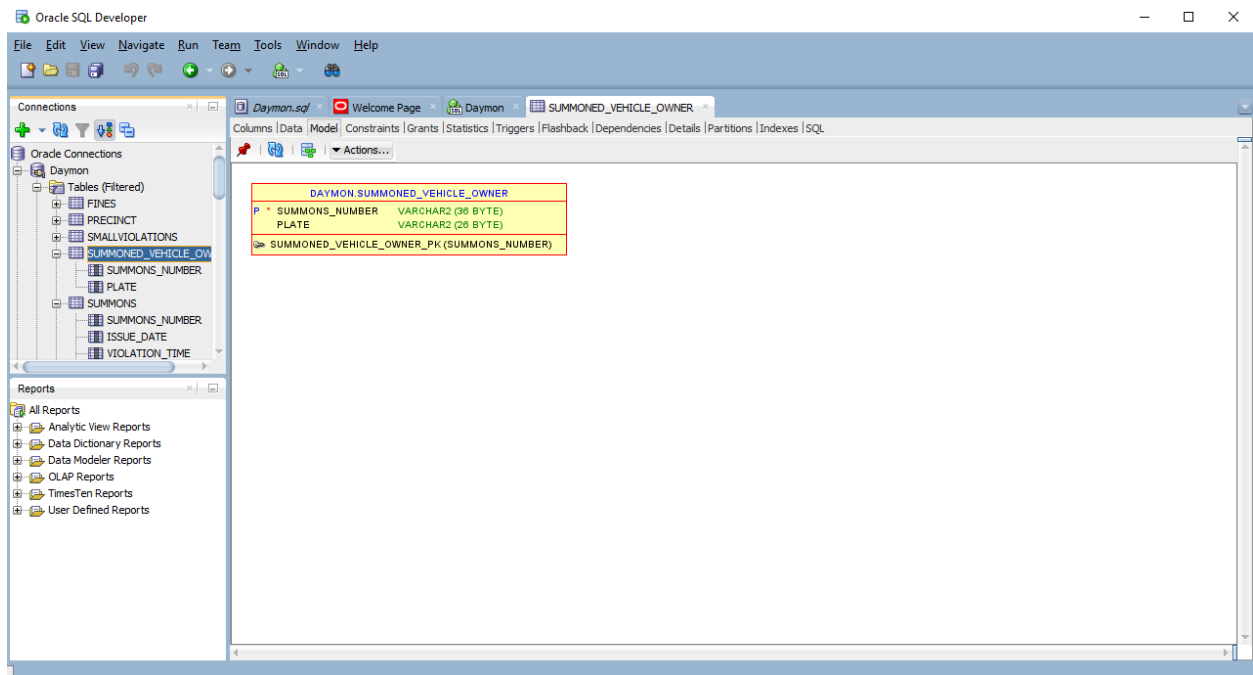
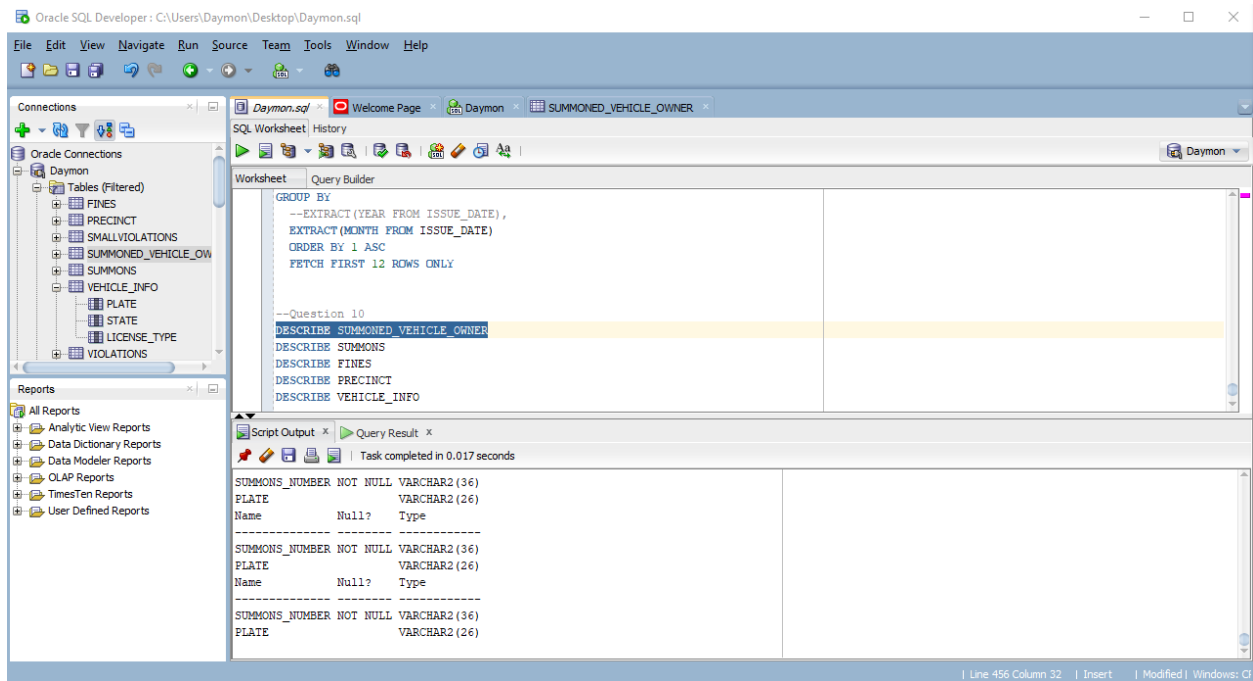


TABLE 2: SUMMONS

Oracle SQL Developer : C:\Users\Daymon\Desktop\Daymon.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections
Daymon
Tables (Filtered)
FINES
PRECINCT
SMALLVIOLATIONS
SUMMONED_VEHICLE_OW
SUMMONS
SUMMONS_NUMBER
ISSUE_DATE
VIOLATION_TIME
VIOLATION
ISSUING_AGENCY

Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports

Daymon.sql

Welcome Page Daymon SUMMONS

SQL Worksheet History

Worksheet Query Builder

```
FETCH FIRST 12 ROWS ONLY

--Question 10
DESCRIBE SUMMONED_VEHICLE_OWNER
DESCRIBE SUMMONS
DESCRIBE FINES
DESCRIBE PRECINCT
DESCRIBE VEHICLE_INFO

--Question 11
SELECT * FROM v$version;
```

Script Output Query Result

Task completed in 0.343 seconds

ISSUING_AGENCY	VAR
VIOLATION_STATUS	VAR
Name	Null? Type
SUMMONS_NUMBER	NOT NULL VAR
ISSUE_DATE	DATE
VIOLATION_TIME	VAR
VIOLATION	VAR
ISSUING_AGENCY	VAR
VIOLATION_STATUS	VAR

| Line 457 Column 17 | Insert | Modified | Windows: C

Oracle SQL Developer : Table DAYMON.SUMMONS@Daymon

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections
Daymon
Tables (Filtered)
FINES
PRECINCT
SMALLVIOLATIONS
SUMMONED_VEHICLE_OW
SUMMONS
SUMMONS_NUMBER
ISSUE_DATE
VIOLATION_TIME
VIOLATION
ISSUING_AGENCY

Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports

Daymon.sql

Welcome Page Daymon SUMMONS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

DAYMON.SUMMONS

P	SUMMONS_NUMBER	VAR
	ISSUE_DATE	DATE
	VIOLATION_TIME	VAR
	VIOLATION	VAR
	ISSUING_AGENCY	VAR
	VIOLATION_STATUS	VAR
	SUMMONS_PK (SUMMONS_NUMBER)	

TABLE 3:FINES

Oracle SQL Developer : C:\Users\Daymon\Desktop\Daymon.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

Daymon

Tables (Filtered)

FINES

SUMMONS_NUMBER

PLATE

FINE_AMOUNT

PENALTY_AMOUNT

INTEREST_AMOUNT

REDUCTION_AMOUNT

PAYMENT_AMOUNT

AMOUNT_DUE

PRECINCT

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Daymon.sql Welcome Page Daymon FINES

SQL Worksheet History

Worksheet Query Builder

FETCH FIRST 12 ROWS ONLY

--Question 10

DESCRIBE SUMMONED_VEHICLE_OWNER

DESCRIBE SUMMONS

DESCRIBE FINES

DESCRIBE PRECINCT

DESCRIBE VEHICLE_INFO

--Question 11

SELECT * FROM v\$version;

Script Output x Query Result x

Task completed in 0.311 seconds

Name	Null?	Type
SUMMONS_NUMBER	NOT NULL	VARCHAR2 (36)
PLATE		VARCHAR2 (26)
FINE_AMOUNT		NUMBER (38)
PENALTY_AMOUNT		NUMBER (38)
INTEREST_AMOUNT		NUMBER (38)
REDUCTION_AMOUNT		NUMBER (38)
PAYMENT_AMOUNT		NUMBER (38)
AMOUNT_DUE		NUMBER (38)

| Line 458 Column 15 | Insert | Modified | Windows: C

Oracle SQL Developer : Table DAYMON.FINES@Daymon

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

Daymon

Tables (Filtered)

FINES

SUMMONS_NUMBER

PLATE

FINE_AMOUNT

PENALTY_AMOUNT

INTEREST_AMOUNT

REDUCTION_AMOUNT

PAYMENT_AMOUNT

AMOUNT_DUE

PRECINCT

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Daymon.sql Welcome Page Daymon FINES

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

DAYMON.FINES		
P	SUMMONS_NUMBER	VARCHAR2 (36 BYTE)
	PLATE	VARCHAR2 (26 BYTE)
	FINE_AMOUNT	NUMBER (38)
	PENALTY_AMOUNT	NUMBER (38)
	INTEREST_AMOUNT	NUMBER (38)
	REDUCTION_AMOUNT	NUMBER (38)
	PAYMENT_AMOUNT	NUMBER (38)
	AMOUNT_DUE	NUMBER (38)
	FINES_PK (SUMMONS_NUMBER)	

TABLE 4:PRECINCT

Oracle SQL Developer : C:\Users\Daymon\Desktop\Daymon.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

FINES

SUMMONS_NUMBER

PLATE

FINE_AMOUNT

PENALTY_AMOUNT

INTEREST_AMOUNT

REDUCTION_AMOUNT

PAYMENT_AMOUNT

AMOUNT_DUE

PRECINCT

PRECINCT

COUNTY

SMALLVIOLATIONS

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Daymon.sql

Welcome Page

Daymon

PRECINCT

SQL Worksheet

History

Daymon

Worksheet

Query Builder

FETCH FIRST 12 ROWS ONLY

--Question 10

DESCRIBE SUMMONED_VEHICLE_OWNER

DESCRIBE SUMMONS

DESCRIBE FINES

DESCRIBE PRECINCT

DESCRIBE VEHICLE_INFO

--Question 11

SELECT * FROM v\$version;

Script Output

Query Result

Task completed in 0.22 seconds

FINE_AMOUNT	NUMBER (38)
PENALTY_AMOUNT	NUMBER (38)
INTEREST_AMOUNT	NUMBER (38)
REDUCTION_AMOUNT	NUMBER (38)
PAYMENT_AMOUNT	NUMBER (38)
AMOUNT_DUE	NUMBER (38)
Name	Null? Type
PRECINCT	NOT NULL NUMBER (38)
COUNTY	VARCHAR2 (26)

Line 459 Column 18

Insert

Modified | Windows: C

Oracle SQL Developer : Table DAYMON.PRECINCT@Daymon

File Edit View Navigate Run Source Team Tools Window Help

Connections

FINES

SUMMONS_NUMBER

PLATE

FINE_AMOUNT

PENALTY_AMOUNT

INTEREST_AMOUNT

REDUCTION_AMOUNT

PAYMENT_AMOUNT

AMOUNT_DUE

PRECINCT

PRECINCT

COUNTY

SMALLVIOLATIONS

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Daymon.sql

Welcome Page

Daymon

PRECINCT

Columns

Data

Model

Constraints

Grants

Statistics

Triggers

Flashback

Dependencies

Details

Partitions

Indexes

SQL

Actions...

DAYMON.PRECINCT

P

PRECINCT

NUMBER (38)

C

COUNTY

VARCHAR2 (26 BYTE)

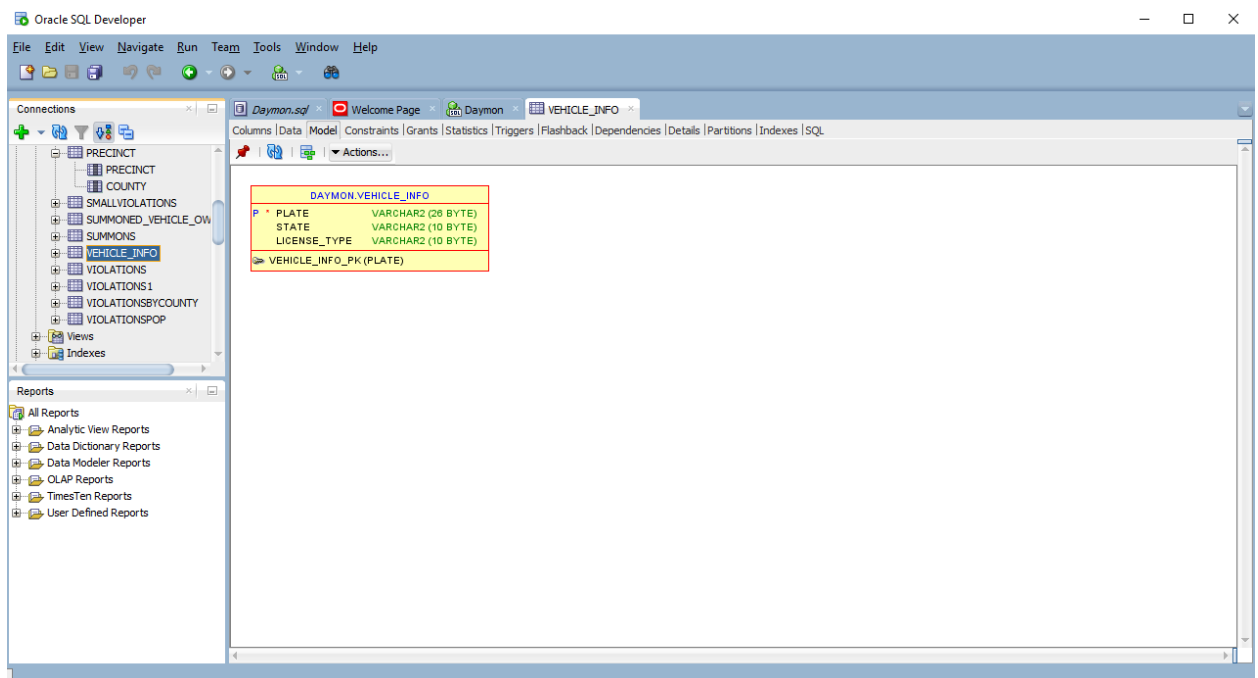
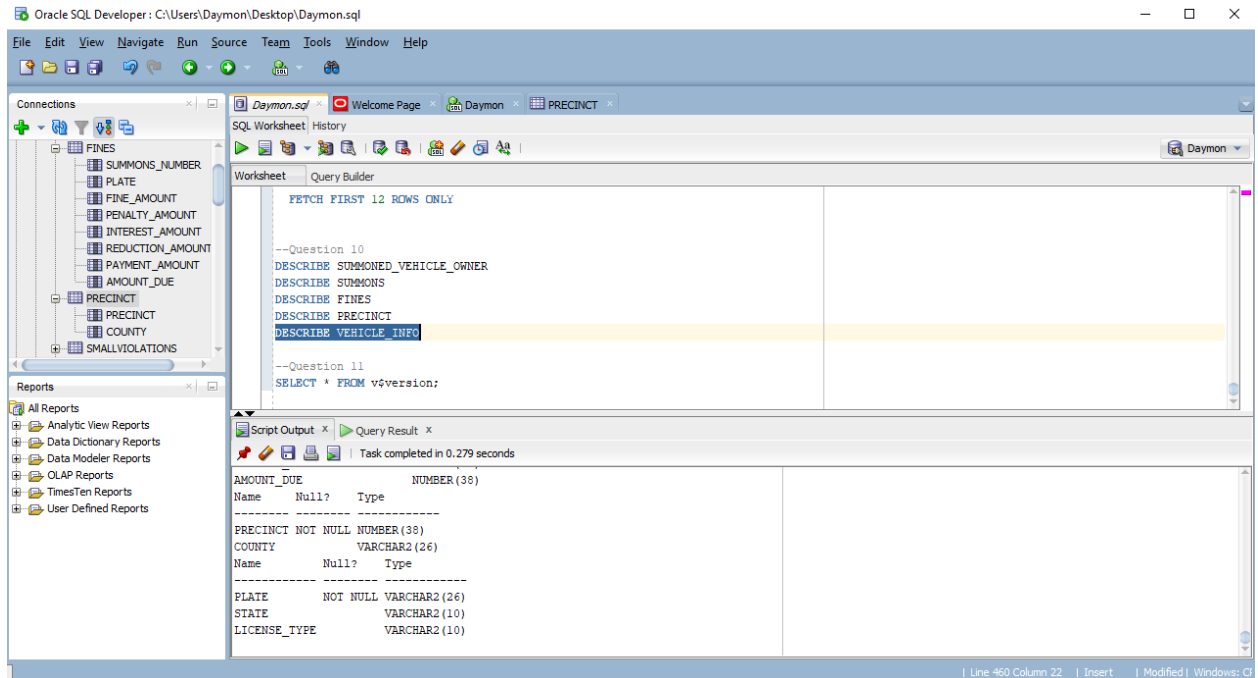
PRECINCT_PK (PRECINCT)

Line 459 Column 18

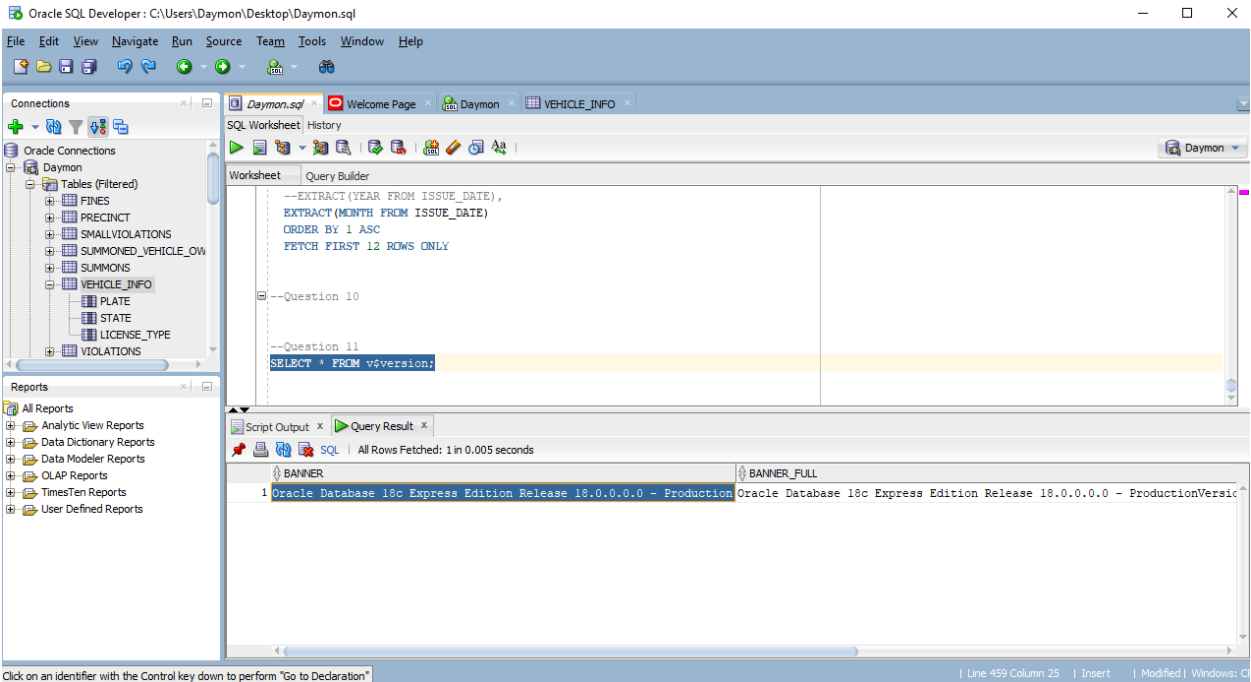
Insert

Modified | Windows: C

TABLE 5: VEHICLE_INFO



11. Display the version of Oracle. Enter:



BANNER	BANNER_FULL	BANNER_LEGACY	CON_ID
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production	Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production		0