Advanced Database Management Systems

Traffic Violation Database Design

Submitted by:

Deepanshu Mahajan

Hariharan Murli Krishnan

Manoj Angane

Nikky Nalin Parashar

Prachi Gupta

Shaily Saigal

Table of Contents

1. INTRODUCTION 3

1.1 DATA SOURCE 3

1.2 BASIC REQUIREMENTS 3

2. LOGICAL DATABASE DESIGN 3

2.1 CONCEPTUAL DESIGN 3

3. CREATION AND INSERTION IN TABLE 4

3.1 CREATE AND INSERT TABLES 4

4. IMPORTING DATA 19

5. QUERIES 21

5.1 TOP 5 VEHICLE TYPE WITH MAXIMUM NUMBER OF ACCIDENTS 21

5.2 TOP 5 COMPANIES WITH MAX ACCIDENTS 22

5.3 NUMBERS VIOLATIONS WHERE THERE WAS AN ARREST 23

5.4 COUNT OF VIOLATION IN THE DATABASE 23

5.5 5 CITIES WITH MAXIMUM NUMBER OF ACCIDENTS 24

5.6 VIOLATIONS BASED ON GENDER 25

5.7 CAR MODELS WHICH HAVE HIGH FATAL DRESS 26

5.8 NUMBER OF ACCIDENTS IN PARTICULAR MONTH 26

6. PERFORMANCE IMPROVEMENT 27

6.1 INDEXING 27

6.2 TRIGGERING 28

1. INTRODUCTION

The database designed in this project is a traffic violation database. This database is designed in such a way that it can serve the important and basic requirements of all traffic violations and can be linked to a web page used by the traffic police. Traffic police can log in and update the details about the vehicle owner like their vehicle, violations and the officer issuing the violation details. They can log in to their accounts and check about the violations. We can draw insights from the database about the type of violations committed and take precautionary measures for it.

1.1 DATA SOURCE

The data for this project was acquired from the catalog.data.gov. This dataset contains traffic violation information from all electronic traffic violations issued in the County. Any information that can be used to uniquely identify the vehicle, the vehicle owner or the officer issuing the violation will not be published.

1.2 BASIC REQUIREMENTS

- The traffic department maintains all records pertaining to traffic citations issued to persons in county and all parking tickets issued by law enforcement agencies.
- The traffic violation database will have information about vehicle, manufacturing company, model, its color, state and city where their violation was committed, types of violations and arrest type details.

2. LOGICAL DATABASE DESIGN

This section includes the entity-relationship diagram (ERD) and data dictionaries of the traffic violation database design and the conceptual design behind this.

2.1 CONCEPTUAL DESIGN

Initially, based on requirements of the enterprise, a conceptual model is designed which is platform independent i.e. irrespective of the database management system version. The following is the conceptual diagram traffic violation database:

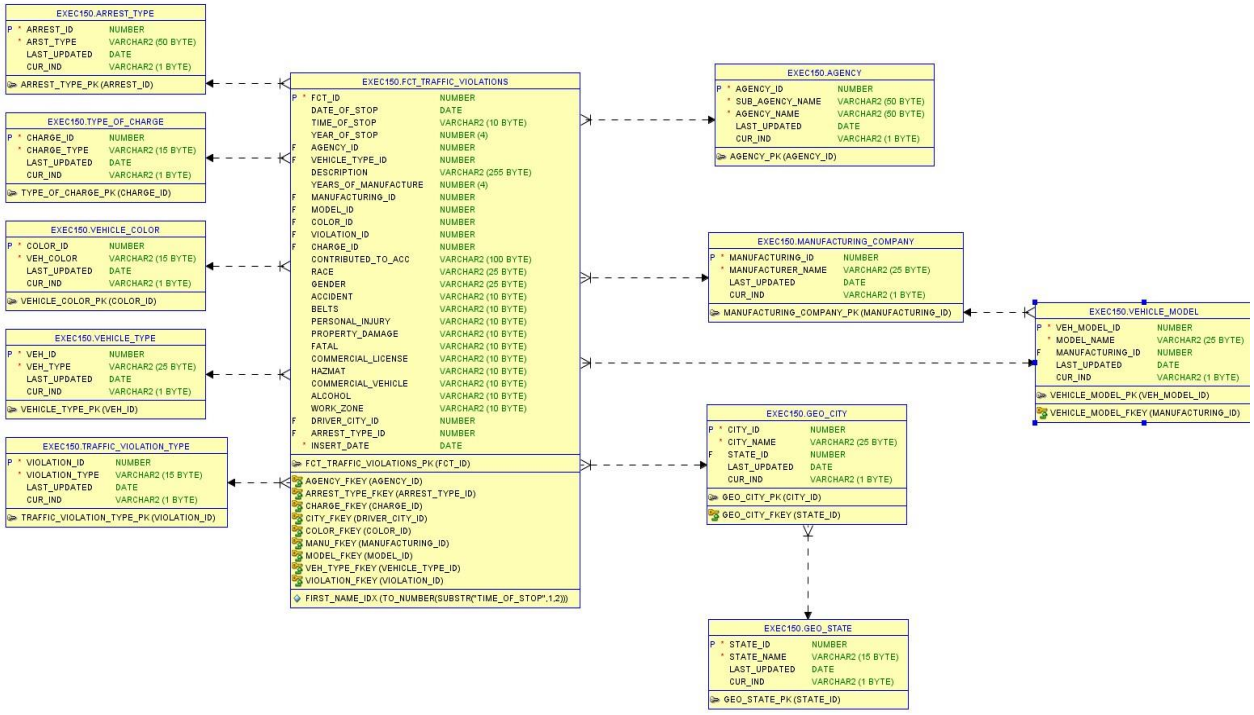


Figure 1: ERD of the database

3. CREATION AND INSERTION IN TABLE

Below are the sample queries we wrote to import the data into the tables we created and inserted the tables. The tables are created based on the ERD design of the project:

3.1 CREATE AND INSERT TABLES

Table Name: Vehicle_Type

Create Query:

```
CREATE TABLE vehicle_type (
    veh_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    veh_type     VARCHAR2(25) NOT NULL,
      "LAST_UPDATED" DATE default sysdate,
      "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert Query:

INSERT INTO vehicle_type(veh_type) SELECT DISTINCT SUBSTR(vehicletype,5) FROM testsh2;

INSERT INTO vehicle_type(veh_type) values ('XX');

Output:

SQL Tuning Advisor × Query Result ×

SQL | All Rows Fetched: 23 in 0.191 seconds

	VEH_ID	VEH_TYPE	LAST_UPDATED	CUR_IND
1	1	Motorcycle	17-NOV-18	Y
2	2	Boat Trailer	17-NOV-18	Y
3	3	Fire (Non-Emerg)	17-NOV-18	Y
4	4	Recreational Vehicle	17-NOV-18	Y
5	5	Light Duty Truck	17-NOV-18	Y
6	6	Truck/Road Tractor	17-NOV-18	Y
7	7	School Bus	17-NOV-18	Y
8	8	Other	17-NOV-18	Y
9	9	Moped	17-NOV-18	Y
10	10	Farm Vehicle	17-NOV-18	Y
11	11	Transit Bus	17-NOV-18	Y

Table Name: vehicle_color

Create:

```
CREATE TABLE vehicle_color (
    color_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    veh_color    VARCHAR2(15) NOT NULL,
            "LAST_UPDATED" DATE default sysdate,
            "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

INSERT INTO vehicle_color(veh_color) SELECT DISTINCT color FROM testsh2 WHERE color IS NOT NULL;

INSERT INTO vehicle_color(veh_color) values ('XX');

Output:

SQL Tuning Advisor | Query Result

SQL | All Rows Fetched: 28 in 0.062 seconds

	COLOR_ID	VEH_COLOR	LAST_UPDATED	CUR_IND
1	1	RED	17-NOV-18	Y
2	2	GRAY	17-NOV-18	Y
3	3	COPPER	17-NOV-18	Y
4	4	MULTICOLOR	17-NOV-18	Y
5	5	GREEN	17-NOV-18	Y
6	6	BLUE, DARK	17-NOV-18	Y
7	7	MAROON	17-NOV-18	Y
8	8	GREEN, DK	17-NOV-18	Y
9	9	TAN	17-NOV-18	Y
10	10	PURPLE	17-NOV-18	Y
11	11	BLUE, LIGHT	17-NOV-18	Y

Table Name: traffic_violation_type

Create:

```
CREATE TABLE traffic_violation_type (
    violation_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    violation_type     VARCHAR2(15) NOT NULL,
                "LAST_UPDATED" DATE default sysdate,
               "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
Insert into traffic_violation_type (violation_type) (select distinct upper(TRIM(violation_type)) from testsh2);

Insert into traffic_violation_type (violation_type) values('XX');
```

Output:

SQL Tuning Advisor | Query Result

SQL | All Rows Fetched: 4 in 0.06 seconds

	VIOLATION_ID	VIOLATION_TYPE	LAST_UPDATED	CUR_IND
1	1	ESERO	17-NOV-18	Y
2	2	CITATION	17-NOV-18	Y
3	3	WARNING	17-NOV-18	Y
4	4	XX	17-NOV-18	Y

Table Name: type_of_charge

Create:

```
CREATE TABLE type_of_charge (
    charge_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    charge_type     VARCHAR2(15) NOT NULL,
                "LAST_UPDATED" DATE default sysdate,
               "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
Insert into type_of_charge(charge_type) select distinct charge from testsh2;
Insert into type_of_charge(charge_type) values('XX');
```

Output:

SQL Tuning Advisor | Query Result

SQL | Fetched 50 rows in 0.407 seconds

	CHARGE_ID	CHARGE_TYPE	LAST_UPDATED	CUR_IND
1	254	16-301(d)	17-NOV-18	Y
2	255	21-401.1	17-NOV-18	Y
3	256	11-393.95(a)	17-NOV-18	Y
4	257	11-393.20(9d)	17-NOV-18	Y
5	258	11-393.78	17-NOV-18	Y
6	259	11-393.10(0b)	17-NOV-18	Y
7	260	16-301(g)	17-NOV-18	Y
8	261	21-503(a)	17-NOV-18	Y
9	262	24-106.2(a)	17-NOV-18	Y
10	263	16-101(al)	17-NOV-18	Y
11	264	21-1003(n)	17-NOV-18	Y

Table Name: arrest_type

Create:

```
CREATE TABLE arrest_type (
    arrest_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    arst_type    VARCHAR2(50) NOT NULL,
                 "LAST_UPDATED" DATE default sysdate,
                 "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
INSERT INTO arrest_type(arst_type) SELECT DISTINCT SUBSTR(arrest_type, 4) FROM testsh2;
insert INTO  arrest_type (arst_type) values ('XX');
```

Output:

SQL Tuning Advisor | Query Result

SQL | All Rows Fetched: 20 in 0.06 seconds

	ARREST_ID	ARST_TYPE	LAST_UPDATED	CUR_IND
1	1	Motorcycle	17-NOV-18	Y
2	2	Marked Moving Radar (Moving)	17-NOV-18	Y
3	3	Marked Moving Radar (Stationary)	17-NOV-18	Y
4	4	Unmarked Moving Radar (Stationary)	17-NOV-18	Y
5	5	Mounted Patrol	17-NOV-18	Y
6	6	Unmarked Moving Radar (Moving)	17-NOV-18	Y
7	7	Marked Stationary Radar	17-NOV-18	Y
8	8	Marked VASCAR	17-NOV-18	Y
9	9	Aircraft Assist	17-NOV-18	Y
10	10	Marked Patrol	17-NOV-18	Y
11	11	Marked (Off-Duty)	17-NOV-18	Y

Table Name: agency

Create:

```
CREATE TABLE agency (
    agency_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    sub_agency_name    VARCHAR2(50) NOT NULL,
                agency_name    VARCHAR2(50) NOT NULL,
                "LAST_UPDATED" DATE default sysdate,
                "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
INSERT INTO agency(sub_agency_name, agency_name) SELECT DISTINCT subagency, agency
FROM testsh2;

INSERT INTO agency (agency_name,sub_agency_name) values ('XX','XX');
```

Output:

SQL Tuning Advisor | Query Result

All Rows Fetched: 8 in 0.07 seconds

	AGENCY_ID	SUB_AGENCY_NAME	AGENCY_NAME	LAST_UPDATED	CUR_IND
1	1	6th district, Gaithersburg / Montgomery Village	MCP	17-NOV-18	Y
2	2	2nd district, Bethesda	MCP	17-NOV-18	Y
3	3	5th district, Germantown	MCP	17-NOV-18	Y
4	4	3rd district, Silver Spring	MCP	17-NOV-18	Y
5	5	Headquarters and Special Operations	MCP	17-NOV-18	Y
6	6	4th district, Wheaton	MCP	17-NOV-18	Y
7	7	1st district, Rockville	MCP	17-NOV-18	Y
8	8	XX	XX	17-NOV-18	Y

SQL Tuning Advisor | Query Result

All Rows Fetched: 8 in 0.07 seconds

	AGENCY_ID	SUB_AGENCY_NAME	AGENCY_NAME	LAST_UPDATED	CUR_IND
1	1	6th district, Gaithersburg / Montgomery Village	MCP	17-NOV-18	Y
2	2	2nd district, Bethesda	MCP	17-NOV-18	Y
3	3	5th district, Germantown	MCP	17-NOV-18	Y
4	4	3rd district, Silver Spring	MCP	17-NOV-18	Y
5	5	Headquarters and Special Operations	MCP	17-NOV-18	Y
6	6	4th district, Wheaton	MCP	17-NOV-18	Y
7	7	1st district, Rockville	MCP	17-NOV-18	Y
8	8	XX	XX	17-NOV-18	Y

Table Name: manufacturing_company

Create:

```
CREATE TABLE manufacturing_company (
    manufacturing_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
            manufacturer_name     VARCHAR2(25) NOT NULL,
            "LAST_UPDATED" DATE default sysdate,
            "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
Insert INTO manufacturing_company(manufacturer_name) select distinct TRIM(MAKE) FROM
testsh2;
```

Output:

SQL Tuning Advisor | Query Result

SQL | All Rows Fetched: 22 in 0.06 seconds

	MANUFACTURING_ID	MANUFACTURER_NAME	LAST_UPDATED	CUR_IND
1	1	HONDA	17-NOV-18	Y
2	2	SUZUKI	17-NOV-18	Y
3	3	FORD	17-NOV-18	Y
4	4	MERCEDES	17-NOV-18	Y
5	5	CHEVROLET	17-NOV-18	Y
6	6	INFINITY	17-NOV-18	Y
7	7	EAGLE	17-NOV-18	Y
8	8	BMW	17-NOV-18	Y
9	9	VOLKSWAGEN	17-NOV-18	Y
10	10	DODGE	17-NOV-18	Y
11	11	TESLA	17-NOV-18	Y

Table Name: vehicle_model

Create:

```
CREATE TABLE vehicle_model (
    veh_model_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    model_name     VARCHAR2(25) NOT NULL,
                manufacturing_id     NUMBER
    CONSTRAINT vehicle_model_fkey REFERENCES manufacturing_company
        (manufacturing_id),
            "LAST_UPDATED" DATE default sysdate,
            "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
INSERT INTO vehicle_model(model_name,manufacturing_id)
SELECT distinct trim(MODEL),(select manufacturing_id from manufacturing_company where
MANUFACTURER_NAME=MAKE) FROM testsh2
where trim(MODEL) is not null;
```

insert into vehicle_model(model_name,manufacturing_id) values('XX',17);

Output:

SQL Tuning Advisor | Query Result

SQL | Fetched 50 rows in 0.101 seconds

	VEH_MODEL_ID	MODEL_NAME	MANUFACTURING_ID	LAST_UPDATED	CUR_IND
1	256	JEEP WRANGLER	15	17-NOV-18	Y
2	257	FORD	5	17-NOV-18	Y
3	258	4DOOR	8	17-NOV-18	Y
4	259	TCQ	11	17-NOV-18	Y
5	260	CARAVAN	5	17-NOV-18	Y
6	261	E 350	4	17-NOV-18	Y
7	262	2S ECLIPSE	4	17-NOV-18	Y
8	263	TR	4	17-NOV-18	Y
9	264	EZERA	1	17-NOV-18	Y
10	265	CP	12	17-NOV-18	Y
11	266	EDGE (SUV)	3	17-NOV-18	Y

Table Name: geo_state

Create:

```
CREATE TABLE geo_state (
    state_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
            state_name    VARCHAR2(15) NOT NULL,
            "LAST_UPDATED" DATE default sysdate,
            "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
insert into geo_state (state_name) values ('AL');
insert into geo_state (state_name) values ('AK');
insert into geo_state (state_name) values ('AZ');
insert into geo_state (state_name) values ('AR');
insert into geo_state (state_name) values ('CA');
insert into geo_state (state_name) values ('CO');
```

```
insert into geo_state (state_name) values ('CT');
insert into geo_state (state_name) values ('DE');
insert into geo_state (state_name) values ('FL');
insert into geo_state (state_name) values ('GA');
insert into geo_state (state_name) values ('HI');
insert into geo_state (state_name) values ('ID');
insert into geo_state (state_name) values ('IL');
insert into geo_state (state_name) values ('IN');
insert into geo_state (state_name) values ('IA');
insert into geo_state (state_name) values ('KS');
insert into geo_state (state_name) values ('KY');
insert into geo_state (state_name) values ('LA');
insert into geo_state (state_name) values ('ME');
insert into geo_state (state_name) values ('MD');
insert into geo_state (state_name) values ('MA');
insert into geo_state (state_name) values ('MI');
insert into geo_state (state_name) values ('MN');
insert into geo_state (state_name) values ('MS');
insert into geo_state (state_name) values ('MO');
insert into geo_state (state_name) values ('MT');
insert into geo_state (state_name) values ('NE');
insert into geo_state (state_name) values ('NV');
insert into geo_state (state_name) values ('NH');
insert into geo_state (state_name) values ('NJ');
insert into geo_state (state_name) values ('NM');
insert into geo_state (state_name) values ('NY');
insert into geo_state (state_name) values ('NC');
insert into geo_state (state_name) values ('ND');
```

```
insert into geo_state (state_name) values ('OH');
insert into geo_state (state_name) values ('OK');
insert into geo_state (state_name) values ('OR');
insert into geo_state (state_name) values ('PA');
insert into geo_state (state_name) values ('RI');
insert into geo_state (state_name) values ('SC');
insert into geo_state (state_name) values ('SD');
insert into geo_state (state_name) values ('TN');
insert into geo_state (state_name) values ('TX');
insert into geo_state (state_name) values ('UT');
insert into geo_state (state_name) values ('VT');
insert into geo_state (state_name) values ('VA');
insert into geo_state (state_name) values ('WA');
insert into geo_state (state_name) values ('WV');
insert into geo_state (state_name) values ('WI');
insert into geo_state (state_name) values ('WY');
insert into geo_state (state_name) values ('XX');
```

Output:

SQL Tuning Advisor | Query Result

SQL | Fetched 50 rows in 0.128 seconds

	STATE_ID	STATE_NAME	LAST_UPDATED	CUR_IND
1	1	AL	17-NOV-18	Y
2	2	AK	17-NOV-18	Y
3	3	AZ	17-NOV-18	Y
4	4	AR	17-NOV-18	Y
5	5	CA	17-NOV-18	Y
6	6	CO	17-NOV-18	Y
7	7	CT	17-NOV-18	Y
8	8	DE	17-NOV-18	Y
9	9	FL	17-NOV-18	Y
10	10	GA	17-NOV-18	Y
11	11	HI	17-NOV-18	Y

Table Name: geo_city

Create:

```
CREATE TABLE geo_city (
    city_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    city_name    VARCHAR2(25) NOT NULL,
    state_id NUMBER
    CONSTRAINT geo_city_fkey REFERENCES geo_state
        (state_id),
            "LAST_UPDATED" DATE default sysdate,
            "CUR_IND" VARCHAR2(1 BYTE) default 'Y');
```

Insert:

```
INSERT INTO geo_city(city_name,state_id)
SELECT distinct trim(DRIVER_CITY),coalesce((select STATE_ID from GEO_STATE where
STATE_NAME=DRIVER_STATE),51) FROM testsh2
where trim(DRIVER_CITY) is not null;
INSERT INTO geo_city(city_name,state_id) values ('XX',51);
```

Output:

SQL Tuning Advisor | Query Result

SQL | Fetched 50 rows in 0.053 seconds

	CITY_ID	CITY_NAME	STATE_ID	LAST_UPDATED	CUR_IND
1	227	FAIRFAX STATION	46	17-NOV-18	Y
2	228	FLUSHING	32	17-NOV-18	Y
3	229	MOUNT AIRY	20	17-NOV-18	Y
4	230	DISTRICT HEIGHTS	20	17-NOV-18	Y
5	231	POOLESVILLE	20	17-NOV-18	Y
6	232	SPRINGFIELD	46	17-NOV-18	Y
7	233	MT RAINER	20	17-NOV-18	Y
8	234	JEFFERSON	20	17-NOV-18	Y
9	235	BOISE	20	17-NOV-18	Y
10	236	COCHRANVILLE	38	17-NOV-18	Y
11	237	PHOENIX	3	17-NOV-18	Y

Table Name: fct_traffic_violations

Create:

Create TABLE fct_traffic_violations (

fct_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

date_Of_Stop date,

time_Of_Stop varchar2(10),

year_of_Stop NUMBER(4),

agency_ID NUMBER CONSTRAINT agency_fkey REFERENCES agency (agency_id),

vehicle_Type_ID NUMBER CONSTRAINT veh_type_fkey REFERENCES vehicle_type (veh_id),

description VARCHAR2(255),

years_Of_Manufacture NUMBER(4),

manufacturing_ID NUMBER CONSTRAINT manu_fkey REFERENCES manufacturing_company (manufacturing_id),

model_ID NUMBER CONSTRAINT model_fkey REFERENCES vehicle_model (veh_model_id),

color_ID NUMBER CONSTRAINT color_fkey REFERENCES vehicle_color(color_id),

violation_ID NUMBER CONSTRAINT violation_fkey REFERENCES traffic_violation_type (violation_id),

```
charge_id NUMBER CONSTRAINT charge_fkey REFERENCES type_of_charge (charge_id),
contributed_To_Acc VARCHAR2(100),
race VARCHAR2(25),
gender VARCHAR2(25),
driver_City_ID NUMBER CONSTRAINT city_fkey REFERENCES geo_city (city_id),
arrest_Type_ID NUMBER CONSTRAINT arrest_type_fkey REFERENCES arrest_type (arrest_id),
Insert_Date date default sysdate not null)
partition by range (YEAR_OF_STOP)
(
    partition p1 values less than (2013),
    partition p2 values less than (2014),
    partition p3 values less than (2015),
    partition p4 values less than (2016),
            partition p5 values less than (2017),
    partition p6 values less than (2018),
  partition p7 values less than (MAXVALUE)
);
```

Insert:

```
insert into fct_traffic_violations
(DATE_OF_STOP,
TIME_OF_STOP,
YEAR_OF_STOP,
DESCRIPTION,
YEARS_OF_MANUFACTURE,
CONTRIBUTED_TO_ACC,
RACE,
GENDER,
```

```
DRIVER_CITY_ID,

ARREST_TYPE_ID,

AGENCY_ID,

VEHICLE_TYPE_ID,

MANUFACTURING_ID,

MODEL_ID,

COLOR_ID,

VIOLATION_ID,

charge_id)

(select to_date(DATE_OF_STOP,'MM/DD/YYYY'),TIME_OF_STOP,EXTRACT(YEAR FROM
to_date(DATE_OF_STOP,'MM/DD/YYYY')),DESCRIPTION,YEAR,CONTRIBUTED_TO_ACCIDE
NT,RACE,GENDER,

coalesce((select city_id from geo_city where city_name=DRIVER_CITY and geo_city.state_id=(select
state_id from geo_state where state_name=driver_state)),1773),

coalesce((select arrest_id from arrest_type where trim(arst_type)=trim(substr(arrest_type,5))),20),

coalesce((select agency_id from agency where agency_name=agency and
sub_agency_name=subagency),8),

coalesce((select veh_id from vehicle_type where veh_type=substr(vehicletype,5)),23),

coalesce((select manufacturing_id from manufacturing_company where
manufacturer_name=make),17),

coalesce((select veh_model_id from vehicle_model where model_name=Model and
manufacturing_id=(select manufacturing_id from manufacturing_company where
manufacturer_name=make)),2676),

coalesce((select color_id from vehicle_color where veh_color=color),28),

coalesce((select violation_id from traffic_violation_type where
violation_type=upper(testsh2.violation_type)),4),

coalesce((select charge_id from type_of_charge where charge_type=charge),612)

from testsh2);
```

Output:

SQL Tuning Advisor | Query Result

SQL | Fetched 50 rows in 0.312 seconds

	FCT_ID	DATE_OF_STOP	TIME_OF_STOP	YEAR_OF_STOP	AGENCY_ID	VEHICLE_TYPE_ID	DESCRIPTION
1	40749	05-MAY-12	00:53:00	2012	3	17	DRIVER ENTERING INTERSECTION AT FLASHING RED TRAFFIC SIGNAL WITHOUT ST
2	40750	28-JUN-12	01:49:00	2012	4	17	DRIVING MOTOR VEH. ON HWY. WEARING EARPLUGS IN BOTH EARS
3	40751	22-SEP-12	02:14:00	2012	1	17	FAILURE OF VEH. DRIVER IN ACCIDENT TO LOCATE AND NOTIFY OWNER OF UNATT
4	40752	24-JUL-12	00:55:00	2012	1	17	DRIVING TO DRIVE MOTOR VEHICLE ON HIGHWAY WITHOUT REQUIRED LICENSE AND
5	40753	06-JUL-12	22:12:00	2012	4	17	DRIVER FAILURE TO OBEY PROPERLY PLACED TRAFFIC CONTROL DEVICE INSTRUCT
6	40754	09-AUG-12	22:56:00	2012	4	17	DRIVING VEHICLE ON HIGHWAY WITH SUSPENDED REGISTRATION
7	40803	19-MAY-12	12:27:00	2012	2	17	OPERATOR NOT RESTRAINED BY SEATBELT
8	40804	31-AUG-12	23:02:00	2012	1	17	DRIVING WHILE IMPAIRED BY ALCOHOL
9	40859	17-MAY-12	23:09:00	2012	7	17	EXCEEDING MAXIMUM SPEED: 34 MPH IN A POSTED 25 MPH ZONE
10	40860	04-SEP-12	23:48:00	2012	4	17	RECKLESS DRIVING VEHICLE IN WANTON AND WILLFUL DISREGARD FOR SAFETY OF
11	40873	19-MAY-12	12:33:00	2012	2	17	OPERATOR NOT RESTRAINED BY SEATBELT
12	40878	04-SEP-12	10:28:00	2012	6	17	EXCEEDING MAXIMUM SPEED: 60 MPH IN A POSTED 45 MPH ZONE
13	40905	18-JAN-12	10:55:00	2012	7	17	PERSON DRIVING MOTOR VEHICLE ON HIGHWAY OR PUBLIC USE PROPERTY ON SUSP

4. IMPORTING DATA

Following is the MS Excel worksheet downloaded from data.gov website for Bank table of the database:

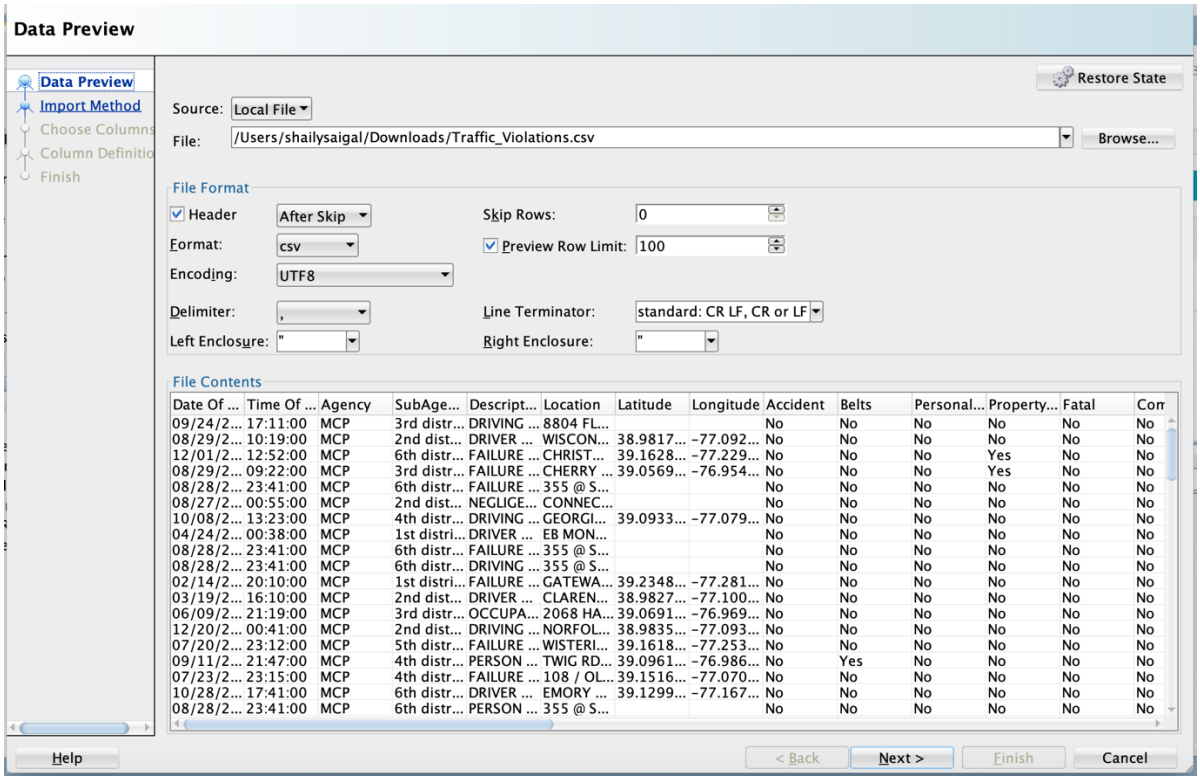
A1 | Date Of Stop

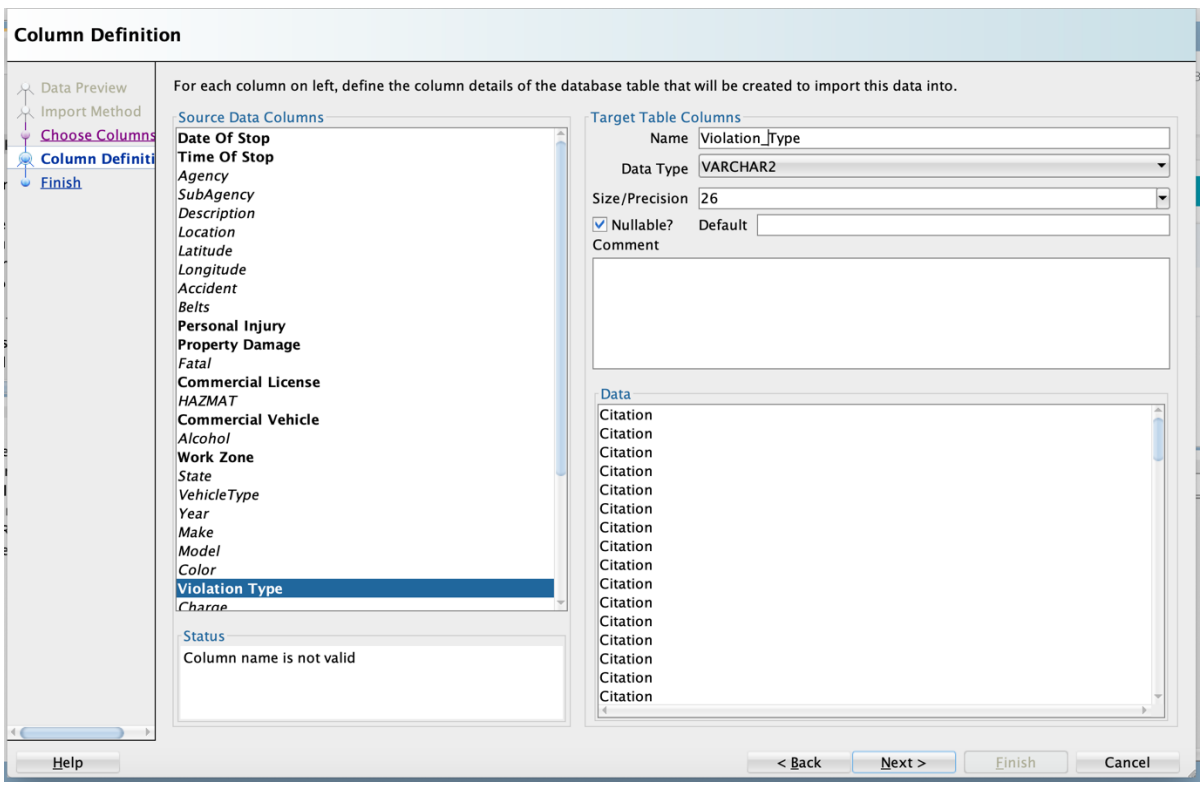
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Date Of Stop	Time Of Sto	Agency	SubAgency	Description	Location	Latitude	Longitude	Accident	Belts	Personal Inju	Property Dan	Fatal	Commercial	HAZMAT	Commercial	Alcohol	Work Zone	State	VehicleType	Year
2	9/24/13	17:11:00	MCP	3rd district,	DRIVING VEI	8804 FLOWER AVE			No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2008
3	8/29/17	10:19:00	MCP	2nd district,	DRIVER FAIL	WISCONSIN	38.981725	-77.092757	No	No	No	No	No	No	No	No	No	No	VA	02 - Automo	2001
4	12/1/14	12:52:00	MCP	6th district,	FAILURE STC	CHRISTOPHE	39.1628883	-77.229088	No	No	No	Yes	No	No	No	No	No	No	MD	02 - Automo	2001
5	8/29/17	9:22:00	MCP	3rd district,	FAILURE YIE	CHERRY HILL	39.056975	-76.954633	No	No	No	Yes	No	No	No	No	No	No	MD	02 - Automo	1998
6	8/28/17	23:41:00	MCP	6th district,	FAILURE OF	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
7	8/27/13	0:55:00	MCP	2nd district,	NEGLIGENT	CONNECTICUT/CHEVY CHASE LAKE			No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2013
8	10/8/13	13:23:00	MCP	4th district,	DRIVING VEI	GEORGIA AV	39.0933833	-77.079552	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	1993
9	4/24/15	0:38:00	MCP	1st district,	DRIVER FAIL	EB MONTROSE PKWY/EAST JEFFERSOI			No	No	No	No	No	No	No	No	No	No	VA	02 - Automo	2003
10	8/28/17	23:41:00	MCP	6th district,	FAILURE OF	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
11	8/28/17	23:41:00	MCP	6th district,	DRIVING VEI	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
12	2/14/14	20:10:00	MCP	1st district,	FAILURE TO	GATEWAY CI	39.2348434	-77.28154	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2005
13	3/19/12	16:10:00	MCP	2nd district,	DRIVER USIN	CLARENDON	38.9827307	-77.100755	No	No	No	No	No	No	No	No	No	No	VA	02 - Automo	1996
14	6/9/12	21:19:00	MCP	3rd district,	OCCUPANT U	2068 HARLE(	39.069143	-76.969678	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2004
15	12/20/12	0:41:00	MCP	2nd district,	DRIVING WH	NORFOLK AV	38.9835782	-77.093105	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2005
16	7/20/12	23:12:00	MCP	5th district,	FAILURE TO	WISTERIA DI	39.1618098	-77.253581	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2002
17	9/11/12	21:47:00	MCP	4th district,	PERSON DRI	TWIG RD AT	39.0961989	-76.986962	No	Yes	No	No	No	No	No	No	No	No	MD	02 - Automo	2009
18	7/23/13	23:15:00	MCP	4th district,	FAILURE TO	108 / OLD B/	39.1516659	-77.070628	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2000
19	10/28/13	17:41:00	MCP	6th district,	DRIVER FAIL	EMORY GRO	39.1299348	-77.16705	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2003
20	8/28/17	23:41:00	MCP	6th district,	PERSON DRI	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
21	12/17/15	13:15:00	MCP	5th district,	FAILURE TO	19916 CRYST	39.2227933	-77.25369	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2005
22	8/28/17	23:41:00	MCP	6th district,	PERSON DRI	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
23	5/12/15	8:26:00	MCP	1st district,	PERSON DRI	WOOTTON F	39.067535	-77.14941	No	No	No	No	No	No	No	No	No	No	VA	28 - Other	2002
24	8/28/17	23:41:00	MCP	6th district,	DRIVING TO	355 @ SOUTH	WESTLAND DRIVE		No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2015
25	10/25/15	11:17:00	MCP	3rd district,	DRIVING VEI	RT. 29/ STEV	39.048205	-76.984722	No	No	No	No	No	Yes	No	No	No	No	MD	02 - Automo	1991
26	8/29/17	1:56:00	MCP	6th district,	DRIVING MC	KENTLANDS	39.1232417	-77.232998	No	No	No	No	No	No	No	No	No	No	MD	05 - Light Du	2005
27	3/20/13	8:53:00	MCP	4th district,	EXCEEDING I	29/LOCKWO	39.0306767	-77.005105	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2004
28	1/12/13	8:53:00	MCP	2nd district,	DRIVER WHE	355 X SOUTH	38.9992782	-77.09701	No	No	No	No	No	Yes	No	Yes	No	No	MD	10 - Transit E	2012
29	5/22/13	14:04:00	MCP	6th district,	FAILURE TO	12413 PUEBLO ROAD			No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2005
30	12/6/12	12:34:00	MCP	Headquarter	PEDESTRIAN	FLOWER AVI	38.9984182	-77.003847	No	No	No	No	No	No	No	No	No	No	XX	29 - Unknown	
31	8/29/17	1:56:00	MCP	6th district,	FAILURE OF	KENTLANDS	39.1232417	-77.232998	No	No	No	No	No	No	No	No	No	No	MD	05 - Light Du	2005
32	8/28/17	23:09:00	MCP	2nd district,	DRIVING VEI	7301 WAVEF	38.9830317	-77.092455	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2008
33	8/28/17	23:09:00	MCP	2nd district,	DRIVING, AT	7301 WAVEF	38.9830317	-77.092455	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2008
34	8/28/17	23:01:00	MCP	6th district,	PASSENGER	WATKINS MI	39.1621717	-77.214045	No	No	No	No	No	No	No	No	No	No	XX	29 - Unknown	
35	9/29/13	21:43:00	MCP	1st district,	FAILURE OF	11717 TRAIL	39.1131833	-77.235485	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2011
36	8/28/17	23:14:00	MCP	4th district,	PERSON DRI	NEW HAMPS	39.0699367	-77.003092	No	No	No	No	No	No	No	No	No	No	MD	02 - Automo	2006

Traffic_Violations

Ready | 100%

We have traffic violations data in csv format. The data was imported to a temporary table “testsh2” using “Import data” option available in SQL developer. The temporary table is used to store the data we imported from the csv file which will be used to import data into the tables we have created as per the ERD design. Since the data we are using is not so big we didn’t use SQL loader utility for this project. We have imported the data from csv files to stage table and then from this stage table to data warehouse.





Number of Records:

Table	Count
Agency	8
Arrest_Type	20
FCT_Traffic_Violation	98212
Geo_City	1773
Geo_state	51
manufacturing_company	22
traffic_violation_type	4
type_of_charge	612
vehicle_color	28
vehicle_model	4294
vehicle_type	23

5. QUERIES

This section covers some useful and interesting queries which demonstrates some of the questions that can be answered by the database. These questions are executed by some internal users for statistical and data mining purposes.

5.1 TOP 5 VEHICLE TYPE WITH MAXIMUM NUMBER OF ACCIDENTS

```
Select veh_type, cnt from (

Select v.veh_type,count(*) as cnt

from FCT_Traffic_violations f

join Vehicle_type v on f.vehicle_type_id=v.veh_id

group by veh_type

order by count(*) desc)

where  rownum<=5;
```

```
Top 5 Vehicle type with maximum number of accidents
Select veh_type, cnt from (
Select v.veh_type,count(*) as cnt
from FCT_Traffic_violations f
join Vehicle_type v on f.vehicle_type_id=v.veh_id
group by veh_type
order by count(*) desc)
where  rownum<=5;

Top 5 Companies with max accidents
SELECT MANUFACTURER_NAME,cnt from(
select m.MANUFACTURER_NAME,COUNT(*) AS cnt
from fct_traffic_violations f
join MANUFACTURING_COMPANY m on f.MANUFACTURING_ID=m.
group by MANUFACTURER_NAME
```

uery Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 5 in 0.199 seconds

	VEH_TYPE	CNT
1	Automobile	89128
2	Light Duty Truck	3777
3	Station Wagon	1144
4	Motorcycle	1134
5	Other	998

5.2 TOP 5 COMPANIES WITH MAX ACCIDENTS

SELECT MANUFACTURER_NAME,cnt from(

select m.MANUFACTURER_NAME,COUNT(*) AS cnt

from fct_traffic_violations f

join MANUFACTURING_COMPANY m on f.MANUFACTURING_ID=m.MANUFACTURING_ID

group by MANUFACTURER_NAME

ORDER BY COUNT(*) DESC)

WHERE ROWNUM <=5

```
Top 5 Companies with max accidents
SELECT MANUFACTURER_NAME,cnt from(
select m.MANUFACTURER_NAME,COUNT(*) AS cnt
from fct_traffic_violations f
join MANUFACTURING_COMPANY m on f.MANUFACTURING_ID=m.MANUFACTURING_ID
group by MANUFACTURER_NAME
ORDER BY COUNT(*) DESC )
WHERE ROWNUM <=5
```

```
Numbers violations where there was an arrest
```

y Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 5 in 0.116 seconds

MANUFACTURER_NAME	CNT
HONDA	17465
TESLA	16964
CHEVROLET	10528
FORD	9553
NISSAN	8553

5.3 NUMBERS VIOLATIONS WHERE THERE WAS AN ARREST

Select violation_type,count(*) as arrests from

fct_traffic_violations f join arrest_type a on f.arrest_type_id = a.arrest_Id

join traffic_violation_type t on f.violation_id= t.violation_id

group by t.violation_type

```
Numbers violations where there was an arrest
Select violation_type,count(*) as arrests  from
fct_traffic_violations f join arrest_type a on f.arrest_type_id = a.arrest_Id
join traffic_violation_type t on f.violation_id= t.violation_id
group by t.violation_type
```

```
Count of violation in the database
```

y Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 3 in 0.133 seconds

VIOLATION_TYPE	ARRESTS
ESERO	2007
CITATION	78121
WARNING	18084

5.4 COUNT OF VIOLATION IN THE DATABASE

SELECT VIOLATION_TYPE,cnt from(

SELECT t.VIOLATION_TYPE,COUNT(*) AS cnt

FROM fct_traffic_violations f

JOIN TRAFFIC_VIOLATION_TYPE t on f.VIOLATION_ID=t.VIOLATION_ID

group by t.VIOLATION_TYPE

ORDER BY COUNT(*) DESC);

```
Count of violation in the database
SELECT VIOLATION_TYPE,cnt from(
SELECT t.VIOLATION_TYPE,COUNT(*) AS cnt
FROM fct_traffic_violations f
JOIN TRAFFIC_VIOLATION_TYPE t on f.VIOLATION_ID=t.VIOLATION_ID
group by t.VIOLATION_TYPE
ORDER BY COUNT(*) DESC )
 ;

5 cities with maximum number of accidents
```

uery Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 3 in 0.07 seconds

	VIOLATION_TYPE	CNT
1	CITATION	78121
2	WARNING	18084
3	ESERO	2007

5.5 5 CITIES WITH MAXIMUM NUMBER OF ACCIDENTS

Select city_name, cnt from (

Select c.city_name ,count(*) as cnt

from FCT_Traffic_violations f

join geo_city c on f.driver_city_id =c.city_id

group by city_name

order by count(*) desc)

where rownum<=5;

```
5 cities with maximum number of accidents
Select city_name, cnt from (
Select c.city_name ,count(*) as cnt
from FCT_Traffic_violations f
join geo_city c on f.driver_city_id =c.city_id
group by city_name
order by count(*) desc)
where rownum<=5;

Violations based on gender
select * from traffic_violation_type;
Select f.gender, violation_type,count(*)  from
fct_traffic_violations f join traffic_violation_type
```

Query Result 1 | Query Result 2

All Rows Fetched: 5 in 0.137 seconds

	CITY_NAME	CNT
1	SILVER SPRING	25590
2	GAITHERSBURG	9786
3	ROCKVILLE	8049
4	GERMANTOWN	8040
5	XX	3413

5.6 VIOLATIONS BASED ON GENDER

select * from traffic_violation_type;

Select f.gender, violation_type,count(*) from

fct_traffic_violations f join traffic_violation_type v

on f.violation_id = v.violation_id

group by v.violation_type,f.gender;

```
select * from traffic_violation_type;
Select f.gender, violation_type,count(*)  from
fct_traffic_violations f join traffic_violation_type v
on f.violation_id = v.violation_id
group by v.violation_type,f.gender;

Car companies have high fatal dress
Select model_name, cnt from (
Select v.model_name ,count(*) as cnt
```

Query Result 1 | Query Result 2 | Query Result 3

All Rows Fetched: 8 in 0.188 seconds

	GENDER	VIOLATION_TYPE	COUNT(*)
1	F	CITATION	22336
2	M	CITATION	55763
3	M	WARNING	11654
4	M	ESERO	1370
5	F	ESERO	637
6	U	CITATION	22
7	F	WARNING	6410
8	U	WARNING	20

5.7 CAR MODELS WHICH HAVE HIGH FATAL DRESS

Select model_name, cnt from (

Select v.model_name ,count(*) as cnt

from FCT_Traffic_violations f

join vehicle_model v on f.manufacturing_id = v.manufacturing_id where f.fatal = 'Yes'

group by v.model_name

order by count(*) desc);

```
Car companies have high fatal dress
Select model_name, cnt from (
Select v.model_name ,count(*) as cnt
from FCT_Traffic_violations f
join vehicle_model v on f.manufacturing_id = v.manufacturi
group by v.model_name
order by count(*) desc);
```

ry Result × | Query Result 1 × | Query Result 2 × | Query Result 3 × | Query Re

SQL | Fetched 50 rows in 0.12 seconds

	MODEL_NAME	CNT
1	4S	47
2	2S	47
3	SU	47
4	SUV	47
5	4DR	47
6	4D	47
7	SW	47
8	TK	47
9	4DOOR	46
10	VN	45
11	TRUCK	45
12	4 DOOR	44
13	SEDAN	44

5.8 NUMBER OF ACCIDENTS IN PARTICULAR MONTH

select HOUR, MONTH, CNT from (

select to_number(substr(time_of_stop,1,2)) Hour,extract(month from date_of_stop) month, count(*) cnt from fct_traffic_violations

group by to_number(substr(time_of_stop,1,2)),extract(month from date_of_stop)

order by count(*) desc) where rownum<=5;

```
select HOUR,MONTH,CNT from (
select to_number(substr(time_of_stop,1,2)) Hour,extract(month from date_of_stop) month, count(*) cnt from fct_traffic_violations
group by to_number(substr(time_of_stop,1,2)),extract(month from date_of_stop)
order by count(*) desc) where rownum<=5;
```

Query Result

SQL | All Rows Fetched: 5 in 0.118 seconds

	HOUR	MONTH	CNT
1	22	9	1775
2	23	9	1590
3	21	9	1350
4	0	9	1295
5	8	9	1221

6. PERFORMANCE IMPROVEMENT

6.1 INDEXING

```
select HOUR,MONTH,CNT from (
select to_number(substr(time_of_stop,1,2)) Hour,extract(month from date_of_stop) month, count(*) cnt from fct_traffic_violations
group by to_number(substr(time_of_stop,1,2)),extract(month from date_of_stop)
order by count(*) desc)
where rownum<=5;
```

Script Output | Query Result | Query Result 1 | Explain Plan

SQL | 0.209 seconds

OPERATION	OBJECT_NAME	OPTIONS	PARTITION_START	PARTITION_STOP	PARTITION_ID	CAR...	COST
SELECT STATEMENT						5	790
COUNT		STOPKEY					
Filter Predicates							
ROWNUM<=5							
VIEW						95779	790
SORT		ORDER BY STOPKEY				95779	790
Filter Predicates							
ROWNUM<=5							
HASH		GROUP BY				95779	790
PARTITION RANGE		ALL	1	7	5	95779	784
TABLE ACCESS	FCT_TRAFFIC_VIOLATIONS	FULL	1	7	5	95779	784

CREATE INDEX first_name_idx ON fct_traffic_violations (to_number(substr(time_of_stop,1,2)));

select HOUR,MONTH,CNT from (

select to_number(substr(time_of_stop,1,2)) Hour,extract(month from date_of_stop) month, count(*) cnt from fct_traffic_violations

group by to_number(substr(time_of_stop,1,2)),extract(month from date_of_stop)

order by count(*) desc)

where rownum<=5;

```
CREATE INDEX first_name_idx ON fct_traffic_violations (to_number(substr(time_of_stop,1,2)));
select HOUR,MONTH,CNT from (
select to_number(substr(time_of_stop,1,2)) Hour,extract(month from date_of_stop) month, count(*) cnt from fct_traffic_violations
group by to_number(substr(time_of_stop,1,2)),extract(month from date_of_stop)
order by count(*) desc)
where rownum<=5;
```

Script Output x | Query Result x | Query Result 1 x | Explain Plan x

SQL | 0.617 seconds

OPERATION	OBJECT_NAME	OPTIONS	PARTITION_START	PARTITION_STOP	PARTITION_ID	CAR...	COST
SELECT STATEMENT						5	790
COUNT		STOPKEY					
Filter Predicates							
ROWNUM<=5							
VIEW						95779	790
SORT		ORDER BY STOPKEY				95779	790
Filter Predicates							
ROWNUM<=5							
HASH		GROUP BY				95779	790
PARTITION RANGE		ALL	1	7	5	95779	784
TABLE ACCESS	FCT_TRAFFIC_VIOLATIONS	FULL	1	7	5	95779	784

6.2 TRIGGERING

```
CREATE OR REPLACE TRIGGER EXEC150.after_insert_city

AFTER INSERT

  ON EXEC150.geo_city

  DECLARE c_id INT;

    BEGIN

      SELECT max(city_id) INTO c_id FROM EXEC150.geo_city;

      UPDATE EXEC150.geo_city SET geo_city.LAST_UPDATED = (SELECT SYSDATE
FROM DUAL) WHERE city_id = c_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_state

AFTER INSERT

  ON EXEC150.geo_state

  DECLARE s_id INT;

    BEGIN

      SELECT max(state_id) INTO s_id FROM EXEC150.geo_state;

      UPDATE EXEC150.geo_state SET geo_state.LAST_UPDATED = (SELECT SYSDATE
FROM DUAL) WHERE state_id = s_id;
```

```
    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_vehicle

AFTER INSERT

  ON EXEC150.vehicle_model

  DECLARE v_id INT;

    BEGIN

      SELECT max(veh_model_id) INTO v_id FROM EXEC150.vehicle_model;

      UPDATE EXEC150.vehicle_model SET vehicle_model.LAST_UPDATED = (SELECT
SYSDATE FROM DUAL) WHERE veh_model_id = v_id;

    END;

CREATE OR REPLACE TRIGGER EXEC150.after_insert_manufacturing

AFTER INSERT

  ON EXEC150.manufacturing_company

  DECLARE m_id INT;

    BEGIN

      SELECT max(manufacturing_id) INTO m_id FROM EXEC150.manufacturing_company;

      UPDATE EXEC150.manufacturing_company SET
manufacturing_company.LAST_UPDATED = (SELECT SYSDATE FROM DUAL) WHERE
manufacturing_id = m_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_agency

AFTER INSERT

  ON EXEC150.agency

  DECLARE a_id INT;

    BEGIN

      SELECT max(agency_id) INTO a_id FROM EXEC150.agency;
```

```
        UPDATE EXEC150.agency SET agency.LAST_UPDATED = (SELECT SYSDATE FROM
DUAL) WHERE agency_id = a_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_arrest_type

AFTER INSERT

  ON EXEC150.arrest_type

  DECLARE ar_id INT;

    BEGIN

        SELECT max(arrest_id) INTO ar_id FROM EXEC150.arrest_type;

        UPDATE EXEC150.arrest_type SET arrest_type.LAST_UPDATED = (SELECT SYSDATE
FROM DUAL) WHERE arrest_id = ar_id;

    END;

CREATE OR REPLACE TRIGGER EXEC150.after_insert_charge

AFTER INSERT

  ON EXEC150.type_of_charge

  DECLARE ch_id INT;

    BEGIN

        SELECT max(charge_id) INTO ch_id FROM EXEC150.type_of_charge;

        UPDATE EXEC150.type_of_charge SET type_of_charge.LAST_UPDATED = (SELECT
SYSDATE FROM DUAL) WHERE charge_id = ch_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_violation

AFTER INSERT

  ON EXEC150.traffic_violation_type

  DECLARE vi_id INT;

    BEGIN

        SELECT max(violation_id) INTO vi_id FROM EXEC150.traffic_violation_type;
```

```
        UPDATE EXEC150.traffic_violation_type SET traffic_violation_type.LAST_UPDATED =
(SELECT SYSDATE FROM DUAL) WHERE violation_id = vi_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_color

AFTER INSERT

  ON EXEC150.vehicle_color

  DECLARE co_id INT;

    BEGIN

        SELECT max(color_id) INTO co_id FROM EXEC150.vehicle_color;

        UPDATE EXEC150.vehicle_color SET vehicle_color.LAST_UPDATED = (SELECT
SYSDATE FROM DUAL) WHERE color_id = co_id;

    END;


CREATE OR REPLACE TRIGGER EXEC150.after_insert_vehicle_type

AFTER INSERT

  ON EXEC150.vehicle_type

  DECLARE vt_id INT;

    BEGIN

        SELECT max(veh_id) INTO vt_id FROM EXEC150.vehicle_type;

        UPDATE EXEC150.vehicle_type SET vehicle_type.LAST_UPDATED = (SELECT
SYSDATE FROM DUAL) WHERE veh_id = vt_id;

    END;
```

TRIGGER EXEC150.after_insert_vehicle

On insertion of new row in *vehicle_model* table, a trigger is executed that updates the Last_Updated column with system's current date.

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM EXEC150.vehicle_model where model_name = 'RV 100'
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.065 seconds

VEH_MOD...	MODEL_N...	MANUFAC...	LAST_UP...	CUR_IND

```
insert into vehicle_model(model_name,manufacturing_id) values('RV 100',18);
```

```
1 row inserted.
```

```
SELECT * FROM EXEC150.vehicle_model where model_name = 'RV 100'

insert into vehicle_model(model_name,manufacturing_id) values('RV 100',18);
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.057 seconds

	VEH_MODEL_ID	MODEL_NAME	MANUFACTURING_ID	LAST_UPDATED	CUR_IND
1	4303	RV 100	18	20-11-18	Y

TRIGGER EXEC150.after_insert_color

On insertion of new row in *vehicle_color* table, a trigger is executed that updates the Last_Updated column with system's current date.

```
SELECT * FROM EXEC150.vehicle_color where veh_color = 'XY'

INSERT INTO vehicle_color(veh_color) values ('XY');
```

Script Output x | Query Result x

SQL | All Rows Fetched: 0 in 0.051 seconds

COLOR_ID	VEH_COLOR	LAST_UP...	CUR_IND

```
INSERT INTO vehicle_color(veh_color) values ('XY');
```

```
1 row inserted.
```

```
SELECT * FROM EXEC150.vehicle_color where veh_color = 'XY'

INSERT INTO vehicle_color(veh_color) values ('XY');
```

Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.051 seconds

	COLOR_ID	VEH_COLOR	LAST_UPDATED	CUR_IND
1	41	XY	20-11-18	Y

TRIGGER EXEC150.after_insert_type

On insertion of new row in *vehicle_type* table, a trigger is executed that updates the Last_Updated column with system's current date.

```
SELECT * FROM EXEC150.vehicle_type where veh_type = 'ZZ'

INSERT INTO vehicle_type(veh_type) values ('XY');
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.048 seconds

VEH_ID	VEH_TYPE	LAST_UP...	CUR_IND

```
INSERT INTO vehicle_type(veh_type) values ('ZZ');
```

```
1 row inserted.
```

```
SELECT * FROM EXEC150.vehicle_type where veh_type = 'ZZ'

INSERT INTO vehicle_type(veh_type) values ('ZZ');
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.047 seconds

	VEH_ID	VEH_TYPE	LAST_UPDATED	CUR_IND
1	41	ZZ	20-11-18	Y