Pune Vidyarthi Griha's
College of Engineering and Technology & G. K. Pate
(Wani) Institute of Management
Approved by AICTE, DTE (Code: 6274) | Affiliated to SPPU, Pune | NAAC Second Cycle 'A' Grade

Designing and Implementing a Relational DBMS on

Fuel Flow Management System

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

By

Omkar Tilekar(S190078577)

Aakanksha Thakar(S190078575)

Harish Bapat(S190078508)

Ayush Suryawanshi(S190078572)

Under the guidance of

Prof. N.R.Sonawane

Department Of Information Technology

PVG's College of Engineering and Technology & G. K. Pate (Wani)
Institute of Management, Pune-09.

2020-2021

CERTIFICATE

This is to certify that the mini project report entitled **"Fuel Flow Management System"** being submitted by a record of bonafide work by Omkar Tilekar (S190078577), Aakanksha Thakar (S190078575), Harish Bapat (S190078508), Ayush Suryawanshi (S190078572) carried out by him/her under the supervision and guidance of Prof. N.R.Sonawane in partial fulfillment ofthe requirement for **SE (Information Technology Engineering) – 2019course**of Savitribai Phule Pune University, Pune in the academic year 2020-2021.

Date:

Place: Pune

Project Guide Head of the Department Principal

This Mini Project report has been examined by us as per the Savitribai Phule PuneUniversity, Pune requirements atPVG's College of Engineering and Technology & G. K. Pate (Wani) Institute of Management, Pune-09on . . . . . .

Internal Examiner External Examiner

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who gave us the opportunity to complete this project. We want to thank the Department of Information Technology ,PVG's COET & GKPIOM for giving us the inspiration and motivation in order to complete this project in virtual mode in this covid – pandemic situation.

We express our sincere thanks to Prof N.R Sonawane sir who acted as the source of inspiration and guided us throughout the process and being a class-teacher for providing valuable guidelines to complete this course. We would also like to thank our family members for supporting and guiding us throughout the course presentation. Last but not least we would also like to give special thanks to our SE – IT batchmates for their help in group learning process.

Omkar Tilekar

Aakanksha Thakar

Harish Bapat

Ayush Suryawanshi

CONTENTS

ABSTRACT

Fuel is the biggest expenditure that fleet managers need to take into account for efficient fleet management. To ace this aspect, one can use fuel management systems that leverage telematics technology to provide valuable insights into your fuel spending and optimize the same.

They used to maintain, control and monitor fuel consumption and stock in any type of industry that uses transport, including rail, road, water and air, as a means of business. Fuel-management systems are designed to effectively measure and manage the use of fuel within the transportation and construction industries. They are typically used for fleets of vehicles, including railway vehicles and aircraft, as well as any vehicle that requires fuel to operate. They employ various methods and technologies to monitor and track fuel inventories, fuel purchases and fuel dispensed. This information can be then stored in computerized systems and reports generated with data to inform management practices. Online fuel management is provided through the use of web portals to provide detailed fueling data, usually via the back end of an automated fuel-management system. This enables consumption control, cost analysis and tax accounting for fuel purchases.

There are several types of fuel-management systems. Card-based fuel-management systems typically track fuel transactions based on a fueling credit card and the associated driver PIN. Reports can then be generated based on fuel consumption by the driver, and data can be directly downloaded. On-site fuel-management systems may employ fleet refueling services or bulk fuel tanks at the site. Fuel is tracked as it is pumped into vehicles, and on-site storage levels can be managed.

1. INTRODUCTION

- **Fuel Flow MANAGEMENT SYSTEM**

Fuel-management systems are used to maintain, control and monitor fuel consumption and stock in any type of industry that uses transport, including rail, road, water and air, as a means of business. Fuel-management systems are designed to effectively measure and manage the use of fuel within the transportation and construction industries. They are typically used for fleets of vehicles, including railway vehicles and aircraft, as well as any vehicle that requires fuel to operate. They employ various methods and technologies to monitor and track fuel inventories, fuel purchases and fuel dispensed. This information can be then stored in computerized systems and reports generated with data to inform management practices. Online fuel management is provided through the use of web portals to provide detailed fueling data, usually via the back end of an automated fuel-management system. This enables consumption control, cost analysis and tax accounting for fuel purchases.

- **HISTORY OF DATABASE MANAGEMENT SYSTEM**

E. F. Codd introduced the term in his seminal paper "A Relational Model of Data for Large Shared Data Banks", published in 1970. In this paper and later papers he defined what he meant by relational. One well-known definition of what constitutes a relational database system is Codd's 12 rules. However, many of the early implementations of the relational model did not conform to all of Codd's rules, so the term gradually came to describe a broader class of database systems. At a minimum, these systems: • presented the data to the user as relations (a presentation in tabular form, i.e. as a collection of tables with each table consisting of a set of rows and columns, can satisfy this

property) • provided relational operators to manipulate the data in tabular form The first systems that were relatively faithful implementations of the relational model were from the University of Michigan; Micro DBMS (1969) and from IBM UK Scientific Center at Peterlee; IS1 (1970–72) and its follow on PRTV (1973–79). The first system sold as an RDBMS was Multics Relational

Data Store, first sold in 1978. Others have been Berkeley Ingres QUEL and IBM BS12. The most popular definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules.[1] A second, theory-based school of thought argues that if a database does not implement all of Codd's rules (or the current understanding on the relational model, as expressed by Christopher J Date, Hugh Darwen and others), it is not relational. This view, sharedby many theorists and other strict adherents to Codd's principles, would disqualify most DBMSs as not relational. For clarification, they often refer to some RDBMSs as Truly-Relational Database Management Systems (TRDBMS), naming others Pseudo-Relational Database Management Systems (PRDBMS). As of now, all commercial relational DBMSs employ SQL as their query language. Alternative query languages have been proposed and implemented, notably the pre-1996 implementation of Berkeley Ingres QUEL. With standardization of theSQL, both commercial and open source DBMSes have adopted some degree of standards compliance.

- **History of SQL (A Research Project Conducted by IBM)**

The origins of the SQL language date back to a research project conducted by IBM at their research laboratories in San Jose, California in the early 1970s. The aim of the project was to develop an experimental RDBMS which would eventually lead to a marketable product. At that time, there was a lot of interest in the relational model for databases at the academic level, in conferences and seminars. IBM, which already had a large share of the commercial database market with hierarchical and network model DBMSs, realized quite quickly that the relational model would figure prominently in future database products. The project at IBM's San Jose labs was started in 1974 and was named System R. A language called Sequel (for Structured English QUEry Language) was chosen as the relational database language for System R. In the project, Sequel was abbreviated to SQL. This is the reason why SQL is still generally pronounced as see-quel. In the first phase of the System R project, researchers concentrated on developing a basic version of the RDBMS. The main aim at this stage was to verify that the theories of the relationalmodel could be translated into a working, commercially viable product. This first phase

was successfully completed by the end of 1975, and resulted in a rudimentary, single-user DBMS based on the relational model. The subsequent phases of System R concentrated on further developing the DBMS from the first phase. Additional features were added, multi-user capability was implemented, and by 1978, a completed RDBMS was ready for user evaluation. The SystemR project was finally completed in 1979. During this time, the SQL language was modified and added to as the needs of the System RDBMS dictated. During the development of System R andSQL/DS, other companies were also at work creating their own relational database management systems. Some of them, Oracle being a prime example, even implemented SQL as the relational database language for their DBMSs concurrently with IBM. Today, the SQL language has gained ANSI (American National Standards Institute) and ISO (International Standards Organization) certification. A version of SQL is available for almost any hardware platform from CRAY supercomputers to IBM PC microcomputers. In recent years, there has been a marked trend for software manufacturers to move away from proprietary database languages and settle on the SQL standard. The microcomputer platform especially has seen a proliferation of previously proprietary packages that have implemented SQL functionality. Even spreadsheet and word processing packages have added options which allow data to be sent to and retrieved from SQL based databases via a Local Area or a Wide Area network connection.[3]

1.1 Problem Statement.

Design a database management system of any organization (back end only).

Create a requirements document consisting of ER model, Relational database design and Relational Tables. Also perform queries on created tables.

1.2 Motivation

- Data redundancy and inconsistency - The file system cannot control redundancy of data as each user defines and maintains the needed files for a specific application to run.Whereas DBMS controls redundancy by maintaining a single repository of data that is defined once and is accessed by many users. As there is no or less redundancy, data remains consistent.
- Data sharing: - File system does not allow sharing of data or sharing is too complex. Whereas in DBMS, data can be shared easily due to a centralized system.
- Data concurrency: - Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occur when changes made by one user gets lost because of changes made by another user. File system does not provide any procedure to stop anomalies. Whereas DBMS provides a locking system to stop anomalies from occurring.
- Data searching: - For every search operation performed on a file system, a different application program has to be written. While DBMS provides inbuilt searching operations. Users only have to write a small query to retrieve data from database.
- Data integrity :- There may be cases when some constraints need to be applied on the data before inserting it in the database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS maintains data integrity by enforcing user defined constraints on data by itself.
- System crashing :-In some cases,systems might have crashes due to various reasons. It is a bane in the case of file systems because once the system crashes, there will be no recovery of the data that's been lost. A DBMS will have the recovery manager which retrieves the data making it another advantage over file systems.
- Data security :- A file system provides a password mechanism to protect the database but how longer can the password be protected? No one can guarantee that. This doesn't happen in the case of DBMS. DBMS has specialized features that help provide shielding to its data.

1.3. Objectives

• This project aims to simplify the task of maintaining records of the Fuel Flow Management System.

• To develop a well designed database to store Fuel flow management System information.

• Provides full functional reports to management of Fuel Flow Management System.

• Helps in maintaining the computerized Fuel Flow Management System details

2. Data Types

When you create a table or add a field to a table in the geodatabase, fields are created as a specific data. type. Data types are classifications that identify possible values for and operations that can be done on the data, as well as the way the data in that field is stored in the database.

When computer programs store data in variables, each variable must be assigned a specific data type. Some common data types include integers, floating point numbers, characters, strings, and arrays. They may also be more specific types, such as dates, timestamps, boolean values, and varchar (variable character) formats.

The following are the datatypes used for the development of our application database:

SR_NO	DATA_TYPE	DESCRIPTION
1	INT(INTEGER)	It is used to specify an integer value
2	CHAR(CHARACTER)	It has a maximum length of 8000 characters. It contains Fixed

3	VARCHAR(VARIABLE + CHARACTER)	Varchar is used to store variable length value as a string It denotes the string length in bytes and it can go up to 8000 character
4	FLOAT	parameter specifies the total number of digits. The number of digits after the decimal point is specified by d parameter.
5	DATE	This data type accepts date values. No parameters are required when declaring a DATE data-type. Date value should be specified in the form of YYYY-MM-DD.

*3.*Data Modeling using ER Model

3.1 REQUIREMENTS COLLECTION AND ANALYSIS

The details of Fuel is stored into the Fuel tables respective with all tables .Each entity (Gulf countries,Export country,Oil companies,State Government,Owner,User) contains primary key and unique keys .The entity , Gulf Countries has bonded with Export Country entity with foreign key .There is one-to-one and one-to-many relationships available between Gulf Countries,Export Country,Oil Companies.All the entities are normalized and reduce duplicity of records .We have implemented indexing on each tables of Fuel Flow Management System tables for fast query execution.

3.1.1 Entity Types, Entity Sets, Attributes, and Keys

1.Gulf countries : Attributes of Gulf countries are
Country_code,Country_Name,Loc_id,Export_cCode,Quantity,Product,Rate .

2.Export Country : Attributes of Export Country are

Export_cCode,Quantity,Product,Rate,Tax,Company_id .

3.Oil Companies: Attributes of Oil Companies are

Company_id, State_code,Quantity,Product,Rate.

4.State Government: Attributes of State Government are

State_code, Company_id,Quantity,Product,Rate,Tax.

5. Owner: Attributes of Owner are

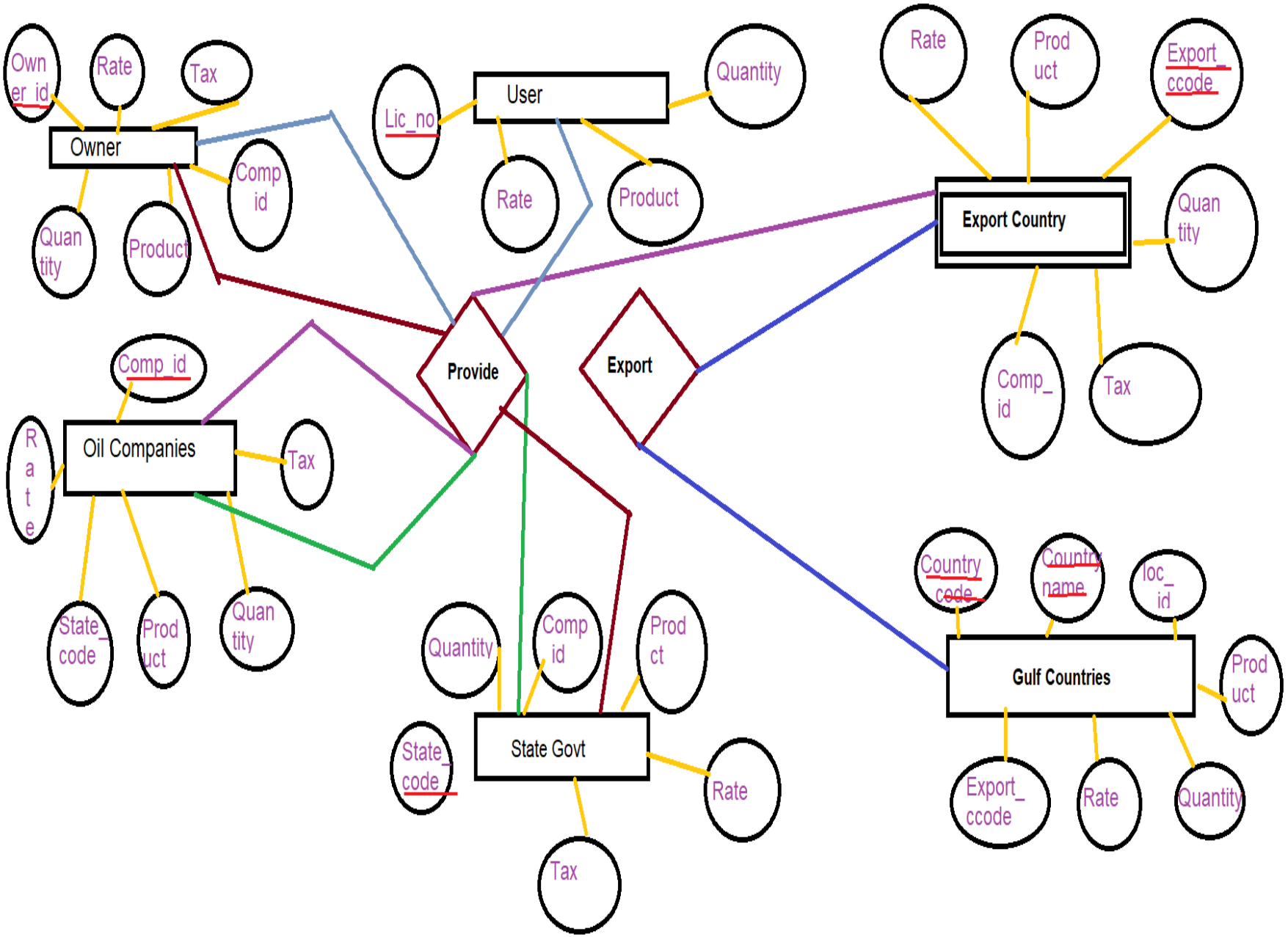
Owner_id,Quantity,Product,Rate,Tax,Company_id.

6. User: Attributes of User are

Lincence_no,Quantity,Product,Rate.

E-R Diagram

This ER (Entity Relationship) Diagram represents the model of Fuel Flow Management System Entity. The entity-relationship diagram of the Fuel Flow Management System shows all the database tables and the relations between Export Country,Gulf Countries, users, Oil Companies etc. It used structure data and to define the relationships between structured data groups of Fuel Flow Management System functionalities. The main Entities in ER Diagram are Owner,Oil Companies,Export Country,User,State Government and Gulf Countries


Owner
Owner_id
Rate
Tax
Quantity
Product
Comp id
User
Lic_no
Quantity
Rate
Product
Export Country
Rate
Product
Export_ccode
Quantity
Comp_id
Tax
Provide
Export
Oil Companies
Comp_id
Rate
Tax
State_code
Product
Quantity
State Govt
State_code
Quantity
Comp id
Product
Rate
Tax
Gulf Countries
Country code
Country name
loc_id
Product
Export_ccode
Rate
Quantity

4.2.2 Relational Database Design Using ER-to-Relational Mapping

1)Gulf Country

Country_ code	Country_ name	loc_id	Export_ cCode	Quantity	Product	Rate

2)Export Country

Export_ cCode	Quantity	Rate	Product	Comp_id	Tax	

3) Oil companies

Comp_id	Rate	State_code	Quantity	Tax	Product

4)State Govt

State_code	Tax	Rate	Product	Comp_id	Quantity

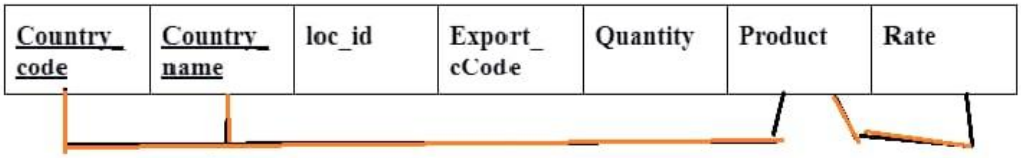
5)Owner

Owner_id	Tax	Rate	Quantity	Product

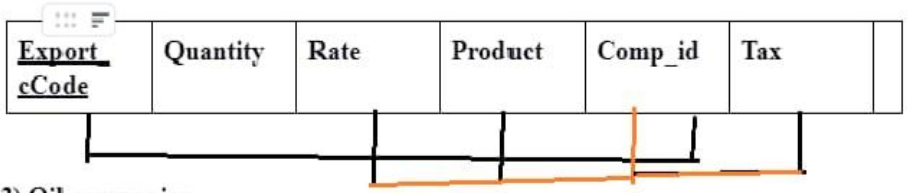
6)User

Linc_no	Rate	Product	Quantity

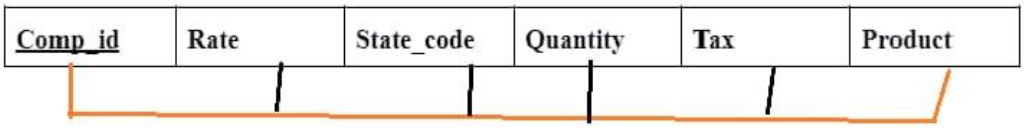
1)Gulf Country



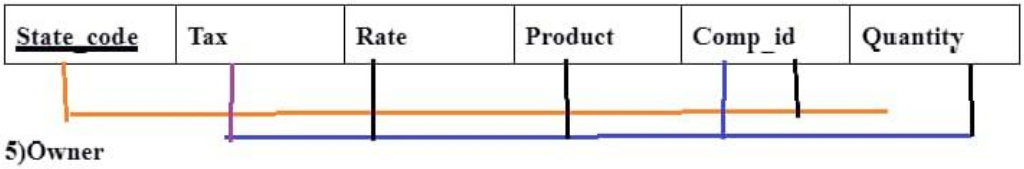
2)Export Country



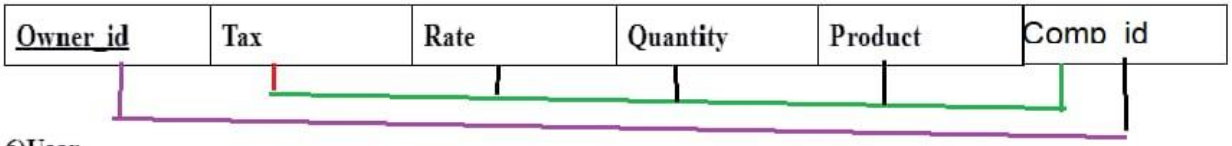
3) Oil companies



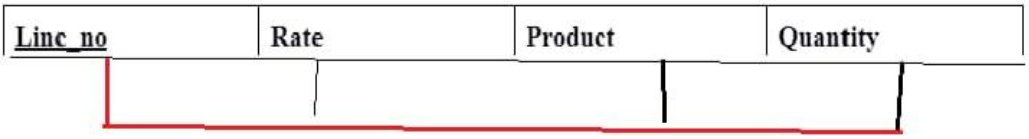
4)State Govt



5)Owner



6)User

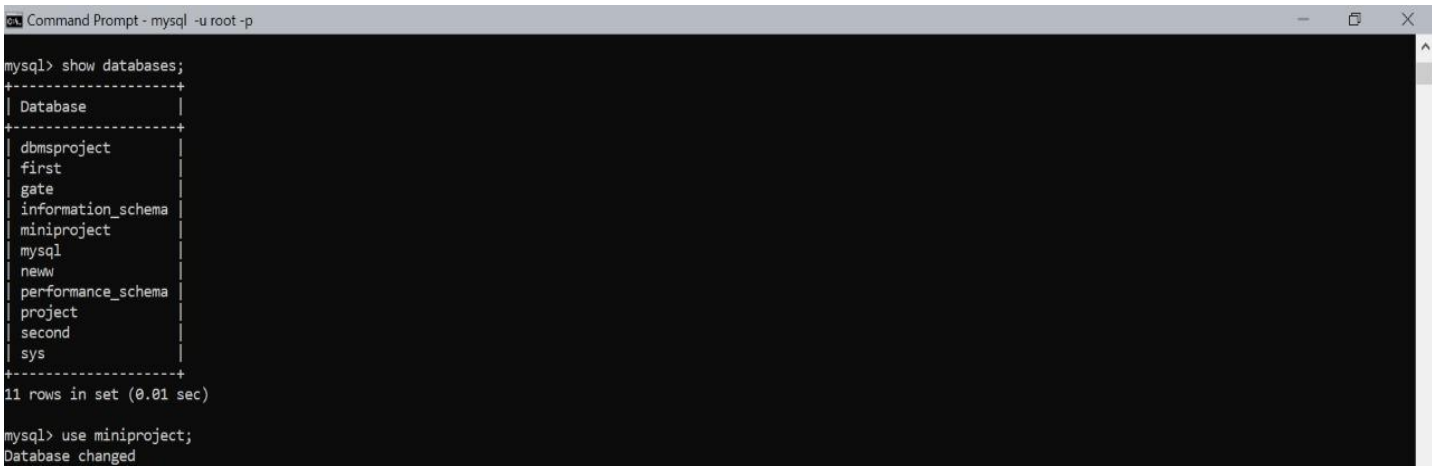


4. CREATING DATABASE USING MYSQL

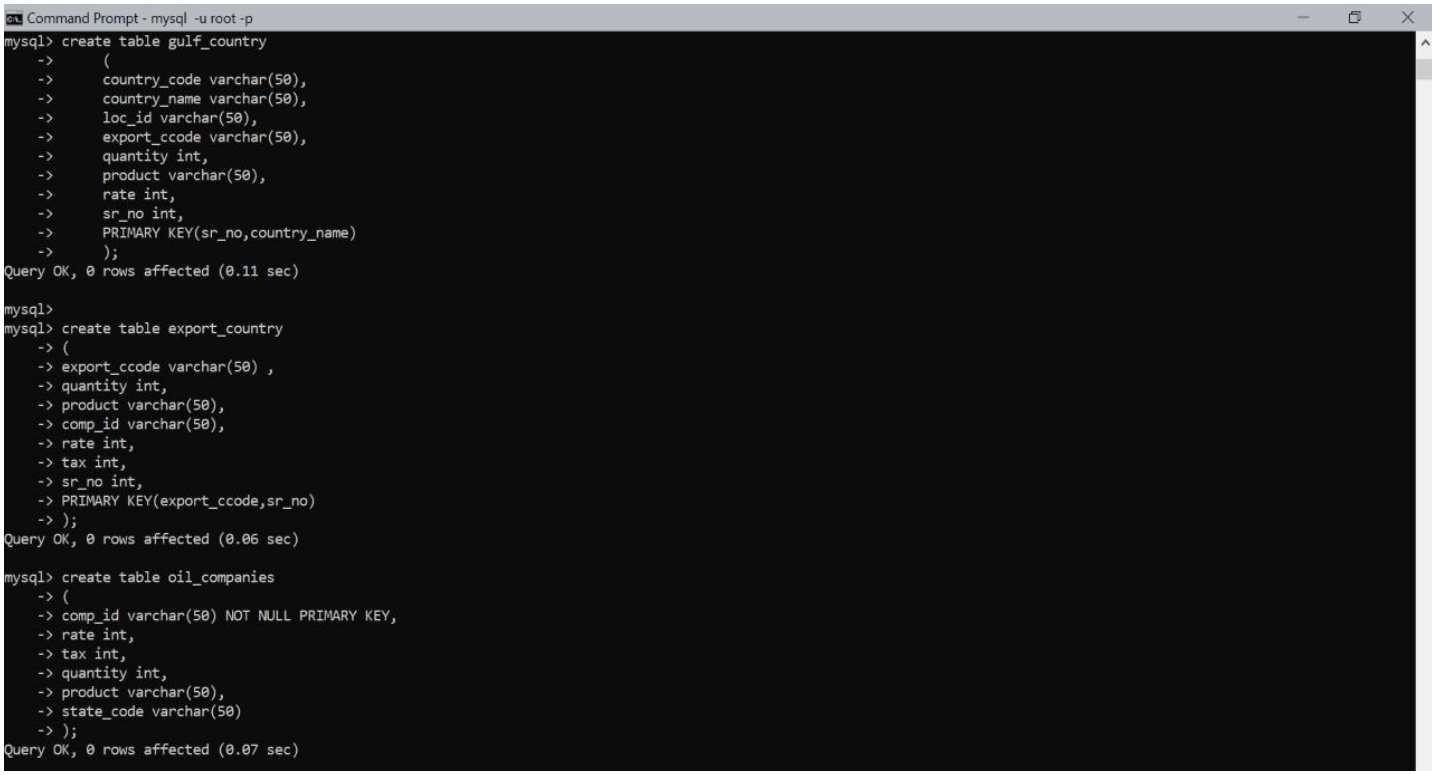
An SQL schema is identified by a schema name, and includes an authorization identifier to indicate the user or account who owns the schema, as well as descriptors for each element in the schema. Schema elements include tables, constraints, views, domains, and other constructs (such as authorization grants) that describe the schema. A schema is created via the CREATE SCHEMA

statement, which can include all the schema elements' definitions. Alternatively, the schema can be assigned a name and authorization identifier, and the elements can be defined later. The CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values, and any attribute constraints, such as NOT NULL. The key, entity integrity, and referential integrity constraints can be specified within the CREATE TABLE statement after the attributes are declared, or they can be added later using the ALTER TABLE command.

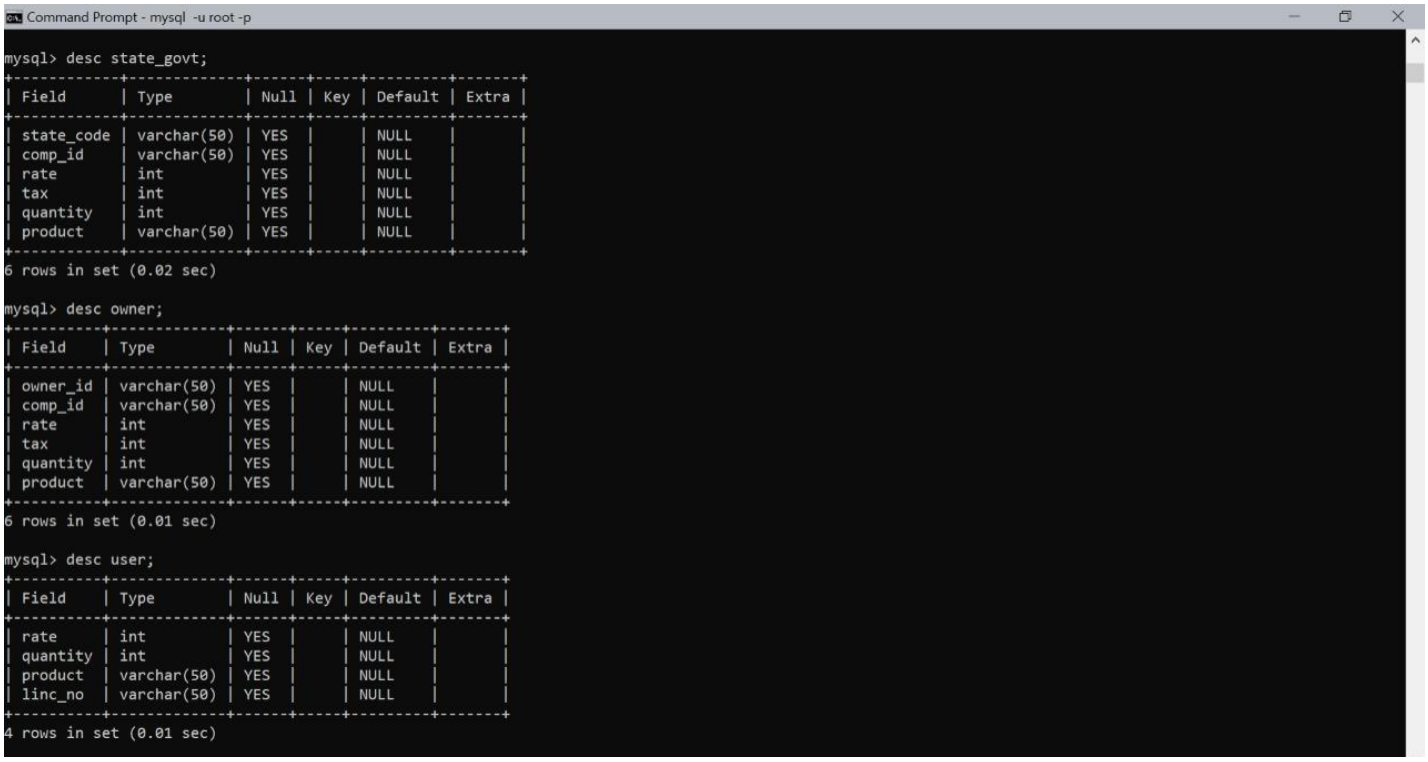
Created database



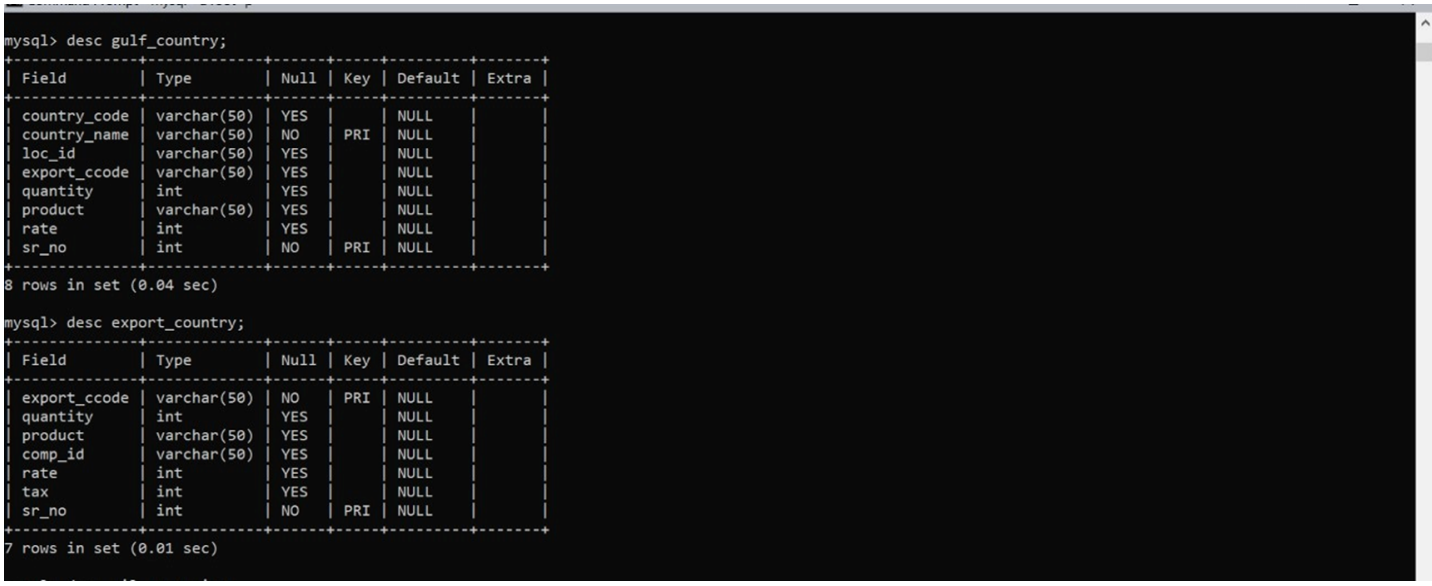
Create Tables



Describe Tables

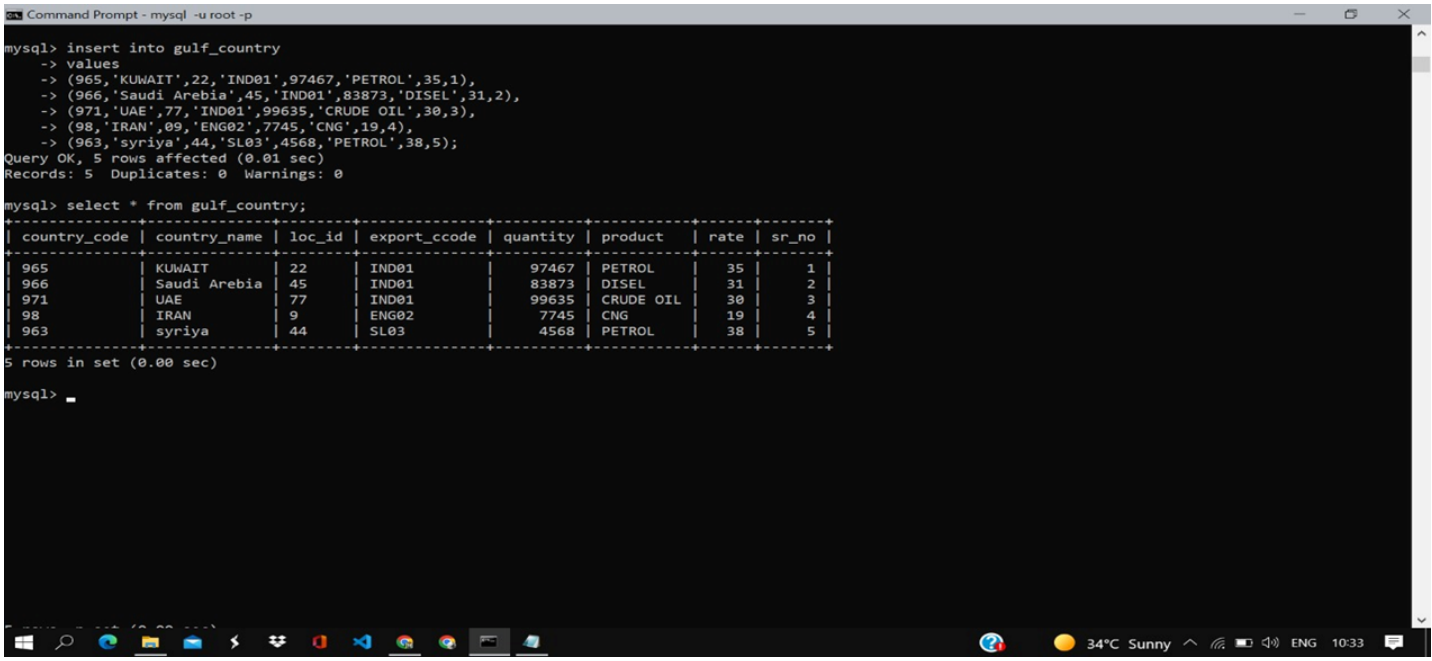



```
mysql> desc state_govt;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| state_code | varchar(50) | YES  |     | NULL    |       |
| comp_id    | varchar(50) | YES  |     | NULL    |       |
| rate       | int         | YES  |     | NULL    |       |
| tax        | int         | YES  |     | NULL    |       |
| quantity   | int         | YES  |     | NULL    |       |
| product    | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)

mysql> desc owner;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| owner_id | varchar(50) | YES  |     | NULL    |       |
| comp_id  | varchar(50) | YES  |     | NULL    |       |
| rate     | int         | YES  |     | NULL    |       |
| tax      | int         | YES  |     | NULL    |       |
| quantity | int         | YES  |     | NULL    |       |
| product  | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> desc user;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| rate     | int         | YES  |     | NULL    |       |
| quantity | int         | YES  |     | NULL    |       |
| product  | varchar(50) | YES  |     | NULL    |       |
| linc_no  | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

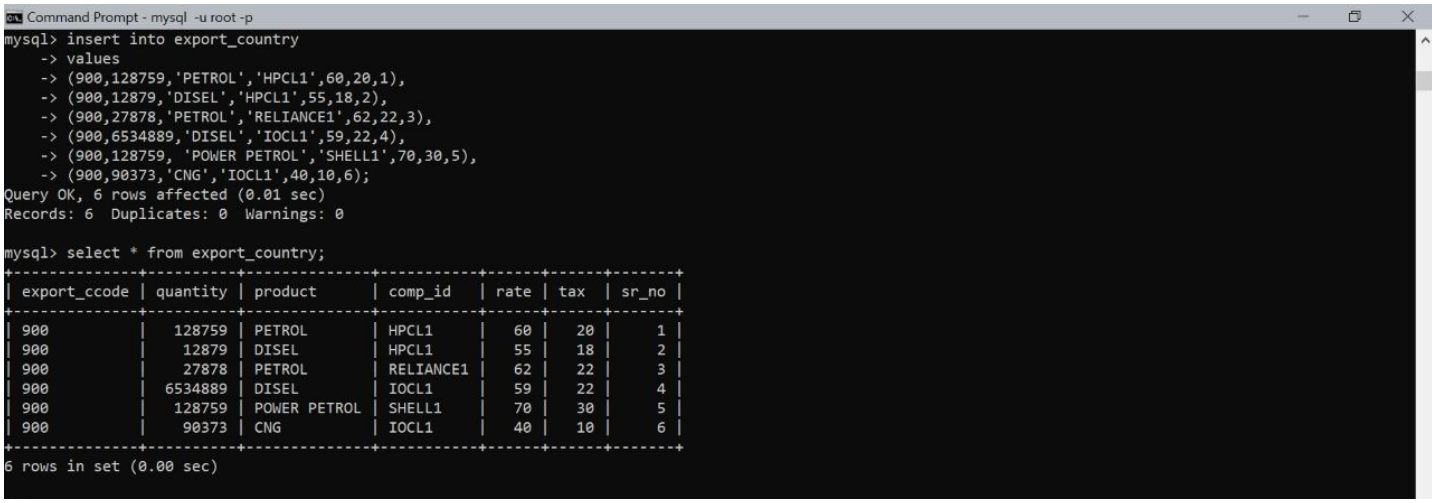


```
mysql> desc gulf_country;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| country_code | varchar(50) | YES  |     | NULL    |       |
| country_name | varchar(50) | NO   | PRI | NULL    |       |
| loc_id       | varchar(50) | YES  |     | NULL    |       |
| export_ccode | varchar(50) | YES  |     | NULL    |       |
| quantity     | int         | YES  |     | NULL    |       |
| product      | varchar(50) | YES  |     | NULL    |       |
| rate         | int         | YES  |     | NULL    |       |
| sr_no        | int         | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.04 sec)

mysql> desc export_country;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| export_ccode | varchar(50) | NO   | PRI | NULL    |       |
| quantity     | int         | YES  |     | NULL    |       |
| product      | varchar(50) | YES  |     | NULL    |       |
| comp_id      | varchar(50) | YES  |     | NULL    |       |
| rate         | int         | YES  |     | NULL    |       |
| tax          | int         | YES  |     | NULL    |       |
| sr_no        | int         | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

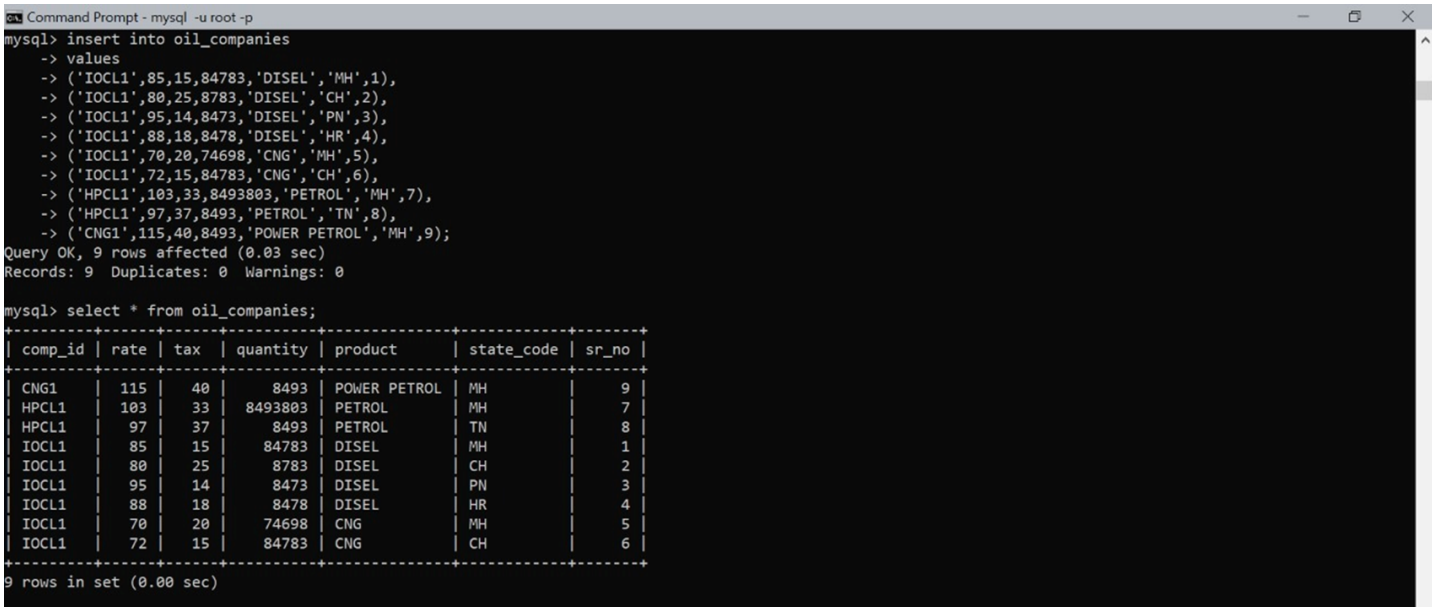
Insert records into tables

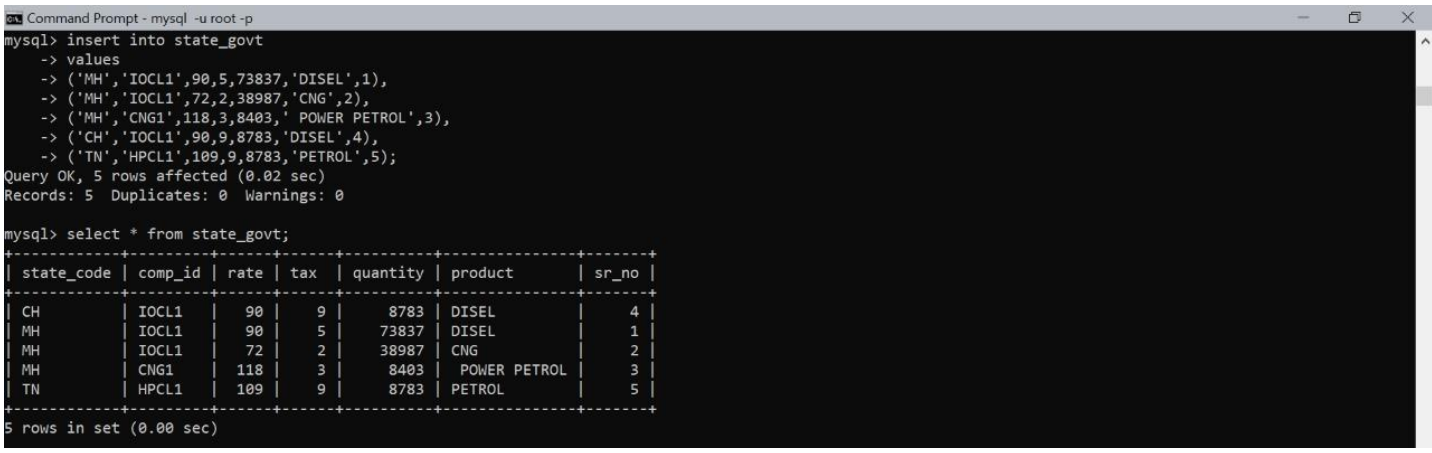


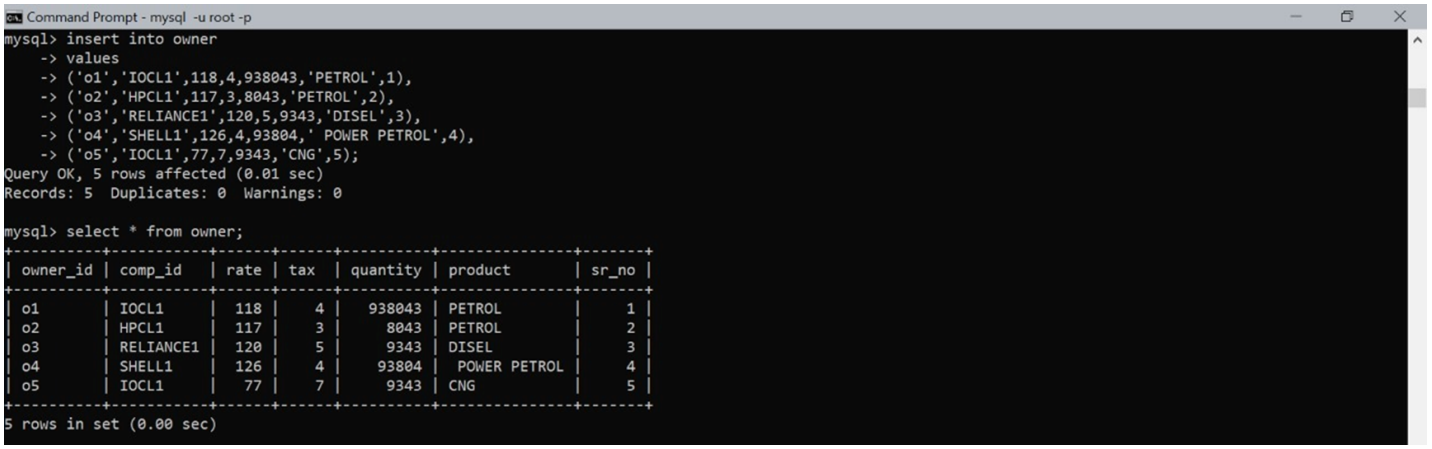
```
Command Prompt - mysql -u root -p

mysql> insert into gulf_country
    -> values
    -> (965,'KUWAIT',22,'IND01',97467,'PETROL',35,1),
    -> (966,'Saudi Arebia',45,'IND01',83873,'DISEL',31,2),
    -> (971,'UAE',77,'IND01',99635,'CRUDE OIL',30,3),
    -> (98,'IRAN',09,'ENG02',7745,'CNG',19,4),
    -> (963,'syriya',44,'SL03',4568,'PETROL',38,5);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from gulf_country;
+--------------+--------------+--------+--------------+----------+-----------+------+-------+
| country_code | country_name | loc_id | export_ccode | quantity | product   | rate | sr_no |
+--------------+--------------+--------+--------------+----------+-----------+------+-------+
| 965          | KUWAIT       | 22     | IND01        |    97467 | PETROL    |   35 |     1 |
| 966          | Saudi Arebia | 45     | IND01        |    83873 | DISEL     |   31 |     2 |
| 971          | UAE          | 77     | IND01        |    99635 | CRUDE OIL |   30 |     3 |
| 98           | IRAN         | 9      | ENG02        |     7745 | CNG       |   19 |     4 |
| 963          | syriya       | 44     | SL03         |     4568 | PETROL    |   38 |     5 |
+--------------+--------------+--------+--------------+----------+-----------+------+-------+
5 rows in set (0.00 sec)

mysql>
```

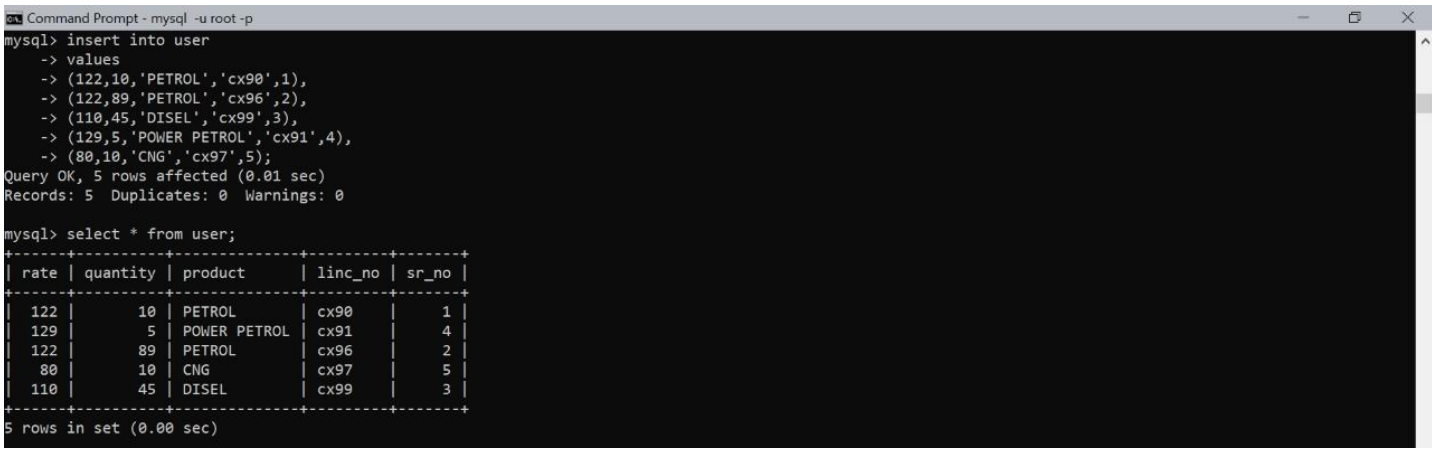


```
Command Prompt - mysql -u root -p
mysql> insert into export_country
    -> values
    -> (900,128759,'PETROL','HPCL1',60,20,1),
    -> (900,12879,'DISEL','HPCL1',55,18,2),
    -> (900,27878,'PETROL','RELIANCE1',62,22,3),
    -> (900,6534889,'DISEL','IOCL1',59,22,4),
    -> (900,128759, 'POWER PETROL','SHELL1',70,30,5),
    -> (900,90373,'CNG','IOCL1',40,10,6);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from export_country;
+--------------+----------+--------------+-----------+------+------+-------+
| export_ccode | quantity | product      | comp_id   | rate | tax  | sr_no |
+--------------+----------+--------------+-----------+------+------+-------+
| 900          |   128759 | PETROL       | HPCL1     |   60 |   20 |     1 |
| 900          |    12879 | DISEL        | HPCL1     |   55 |   18 |     2 |
| 900          |    27878 | PETROL       | RELIANCE1 |   62 |   22 |     3 |
| 900          |  6534889 | DISEL        | IOCL1     |   59 |   22 |     4 |
| 900          |   128759 | POWER PETROL | SHELL1    |   70 |   30 |     5 |
| 900          |    90373 | CNG          | IOCL1     |   40 |   10 |     6 |
+--------------+----------+--------------+-----------+------+------+-------+
6 rows in set (0.00 sec)
```

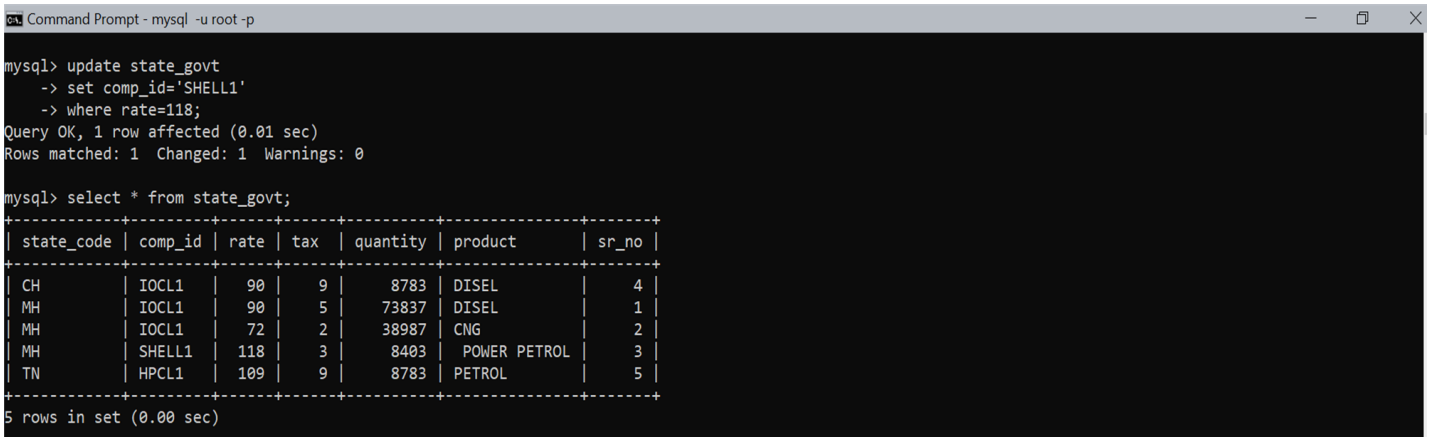

Command Prompt - mysql -u root -p
mysql> insert into oil_companies
-> values
-> ('IOCL1',85,15,84783,'DISEL','MH',1),
-> ('IOCL1',80,25,8783,'DISEL','CH',2),
-> ('IOCL1',95,14,8473,'DISEL','PN',3),
-> ('IOCL1',88,18,8478,'DISEL','HR',4),
-> ('IOCL1',70,20,74698,'CNG','MH',5),
-> ('IOCL1',72,15,84783,'CNG','CH',6),
-> ('HPCL1',103,33,8493803,'PETROL','MH',7),
-> ('HPCL1',97,37,8493,'PETROL','TN',8),
-> ('CNG1',115,40,8493,'POWER PETROL','MH',9);
Query OK, 9 rows affected (0.03 sec)
Records: 9 Duplicates: 0 Warnings: 0
mysql> select * from oil_companies;
comp_id	rate	tax	quantity	product	state_code	sr_no
CNG1	115	40	8493	POWER PETROL	MH	9
HPCL1	103	33	8493803	PETROL	MH	7
HPCL1	97	37	8493	PETROL	TN	8
IOCL1	85	15	84783	DISEL	MH	1
IOCL1	80	25	8783	DISEL	CH	2
IOCL1	95	14	8473	DISEL	PN	3
IOCL1	88	18	8478	DISEL	HR	4
IOCL1	70	20	74698	CNG	MH	5
IOCL1	72	15	84783	CNG	CH	6
9 rows in set (0.00 sec)


Command Prompt - mysql -u root -p
mysql> insert into state_govt
-> values
-> ('MH','IOCL1',90,5,73837,'DISEL',1),
-> ('MH','IOCL1',72,2,38987,'CNG',2),
-> ('MH','CNG1',118,3,8403,' POWER PETROL',3),
-> ('CH','IOCL1',90,9,8783,'DISEL',4),
-> ('TN','HPCL1',109,9,8783,'PETROL',5);
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> select * from state_govt;
state_code	comp_id	rate	tax	quantity	product	sr_no
CH	IOCL1	90	9	8783	DISEL	4
MH	IOCL1	90	5	73837	DISEL	1
MH	IOCL1	72	2	38987	CNG	2
MH	CNG1	118	3	8403	POWER PETROL	3
TN	HPCL1	109	9	8783	PETROL	5
5 rows in set (0.00 sec)

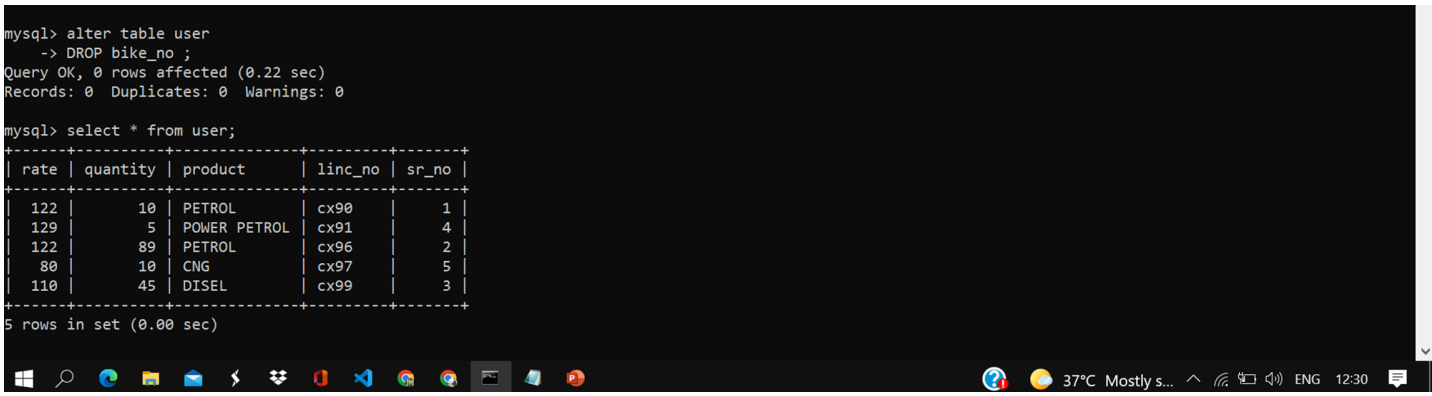

Command Prompt - mysql -u root -p
mysql> insert into owner
-> values
-> ('o1','IOCL1',118,4,938043,'PETROL',1),
-> ('o2','HPCL1',117,3,8043,'PETROL',2),
-> ('o3','RELIANCE1',120,5,9343,'DISEL',3),
-> ('o4','SHELL1',126,4,93804,' POWER PETROL',4),
-> ('o5','IOCL1',77,7,9343,'CNG',5);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> select * from owner;
owner_id	comp_id	rate	tax	quantity	product	sr_no
o1	IOCL1	118	4	938043	PETROL	1
o2	HPCL1	117	3	8043	PETROL	2
o3	RELIANCE1	120	5	9343	DISEL	3
o4	SHELL1	126	4	93804	POWER PETROL	4
o5	IOCL1	77	7	9343	CNG	5
5 rows in set (0.00 sec)


Command Prompt - mysql -u root -p
mysql> insert into user
-> values
-> (122,10,'PETROL','cx90',1),
-> (122,89,'PETROL','cx96',2),
-> (110,45,'DISEL','cx99',3),
-> (129,5,'POWER PETROL','cx91',4),
-> (80,10,'CNG','cx97',5);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> select * from user;
rate	quantity	product	linc_no	sr_no
122	10	PETROL	cx90	1
129	5	POWER PETROL	cx91	4
122	89	PETROL	cx96	2
80	10	CNG	cx97	5
110	45	DISEL	cx99	3
5 rows in set (0.00 sec)

DML UPDATE



DDL ALTER DROP



DDL ALTER MODIFY

```
mysql> alter table user
    -> MODIFY bike_no varchar(20);
Query OK, 5 rows affected (0.19 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from user
    -> ;
+------+----------+--------------+---------+-------+---------+
| rate | quantity | product      | linc_no | sr_no | bike_no |
+------+----------+--------------+---------+-------+---------+
|  122 |       10 | PETROL       | cx90    |     1 | NULL    |
|  129 |        5 | POWER PETROL | cx91    |     4 | NULL    |
|  122 |       89 | PETROL       | cx96    |     2 | NULL    |
|   80 |       10 | CNG          | cx97    |     5 | NULL    |
|  110 |       45 | DISEL        | cx99    |     3 | NULL    |
+------+----------+--------------+---------+-------+---------+
5 rows in set (0.00 sec)
```

DDL ALTER ADD

```
mysql> alter table user
    -> ADD bike_no int;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from user;
+------+----------+--------------+---------+-------+---------+
| rate | quantity | product      | linc_no | sr_no | bike_no |
+------+----------+--------------+---------+-------+---------+
|  122 |       10 | PETROL       | cx90    |     1 |    NULL |
|  129 |        5 | POWER PETROL | cx91    |     4 |    NULL |
|  122 |       89 | PETROL       | cx96    |     2 |    NULL |
|   80 |       10 | CNG          | cx97    |     5 |    NULL |
|  110 |       45 | DISEL        | cx99    |     3 |    NULL |
+------+----------+--------------+---------+-------+---------+
5 rows in set (0.00 sec)
```

RELATIONAL OPERATIONS

- Display all the information of state_govt who is having DIESEL as product

```
mysql> select * from state_govt where product='DISEL';
+------------+---------+------+------+----------+---------+-------+
| state_code | comp_id | rate | tax  | quantity | product | sr_no |
+------------+---------+------+------+----------+---------+-------+
| CH         | IOCL1   |   90 |    9 |     8783 | DISEL   |     4 |
| MH         | IOCL1   |   90 |    5 |    73837 | DISEL   |     1 |
+------------+---------+------+------+----------+---------+-------+
2 rows in set (0.00 sec)
```

- Display the Linc_No who is taken PETROL as product

```
mysql> select linc_no from user where product='PETROL';
+---------+
| linc_no |
+---------+
| cx90    |
| cx96    |
+---------+
2 rows in set (0.00 sec)
```

- Display the product of SHELL1 company

```
mysql> select product from owner where comp_id='SHELL1';
+---------------+
| product       |
+---------------+
|  POWER PETROL |
+---------------+
1 row in set (0.00 sec)
```

- Display the count of the MH State_Code

```
mysql> select count(state_code) from state_govt where state_code='MH';
+-------------------+
| count(state_code) |
+-------------------+
|                 3 |
+-------------------+
1 row in set (0.00 sec)
```

BOOLEAN OPERATIONS

- Use of AND

```
mysql> select tax,rate,product from oil_companies where comp_id='IOCL1' AND state_code='MH';
+------+------+---------+
| tax  | rate | product |
+------+------+---------+
|   15 |   85 | DISEL   |
|   20 |   70 | CNG     |
+------+------+---------+
2 rows in set (0.00 sec)
```

- Use of OR

```
mysql> select tax,rate,product from oil_companies where comp_id='IOCL1' OR state_code='MH';
+------+------+--------------+
| tax  | rate | product      |
+------+------+--------------+
|   40 |  115 | POWER PETROL |
|   33 |  103 | PETROL       |
|   15 |   85 | DISEL        |
|   25 |   80 | DISEL        |
|   14 |   95 | DISEL        |
|   18 |   88 | DISEL        |
|   20 |   70 | CNG          |
|   15 |   72 | CNG          |
+------+------+--------------+
8 rows in set (0.00 sec)
```

- Use of BETWEEN

```
mysql> select rate,product from state_govt where rate BETWEEN 85 and 110;
+------+---------+
| rate | product |
+------+---------+
|   90 | DISEL   |
|   90 | DISEL   |
|  109 | PETROL  |
+------+---------+
3 rows in set (0.00 sec)

mysql>
```

PATTERN MATCHING

- Display all the info about company starts with R and end with 1

```
mysql> select * from oil_companies where comp_id like 'H%1';
+---------+------+------+----------+---------+------------+-------+
| comp_id | rate | tax  | quantity | product | state_code | sr_no |
+---------+------+------+----------+---------+------------+-------+
| HPCL1   |  103 |   33 |  8493803 | PETROL  | MH         |     7 |
| HPCL1   |   97 |   37 |     8493 | PETROL  | TN         |     8 |
+---------+------+------+----------+---------+------------+-------+
2 rows in set (0.01 sec)
```

- Display the count of export_ccode where name starts from I and display the sum of quantity

```
mysql> select count(export_ccode),sum(quantity) from gulf_country where export_ccode like 'I%';
+---------------------+---------------+
| count(export_ccode) | sum(quantity) |
+---------------------+---------------+
|                   3 |        280975 |
+---------------------+---------------+
1 row in set (0.00 sec)

mysql>
```

ARITHMETIC OPERATIONS

- Arithmetic operation ADD

```
mysql> select rate+2,comp_id as new_rate from owner ;
+--------+-----------+
| rate+2 | new_rate  |
+--------+-----------+
|    120 | IOCL1     |
|    119 | HPCL1     |
|    122 | RELIANCE1 |
|    128 | SHELL1    |
|     79 | IOCL1     |
+--------+-----------+
5 rows in set (0.00 sec)
```

- Arithmetic operation modulo

```
mysql> select quantity%1000 as in_liters from owner;
+-----------+
| in_liters |
+-----------+
|        43 |
|        43 |
|       343 |
|       804 |
|       343 |
+-----------+
5 rows in set (0.00 sec)
```

BUILTIN FUNCTIONS

- BUILTIN function lower

```
mysql> select LOWER(country_name) from gulf_country;
+---------------------+
| LOWER(country_name) |
+---------------------+
| kuwait              |
| saudi arebia        |
| uae                 |
| iran                |
| syriya              |
+---------------------+
5 rows in set (0.00 sec)
```

- BUILT IN function REVERSE

```
mysql> select REVERSE(product) from user;
+------------------+
| REVERSE(product) |
+------------------+
| LORTEP           |
| LORTEP REWOP     |
| LORTEP           |
| GNC              |
| LESID            |
+------------------+
5 rows in set (0.00 sec)
```

- Built in function length

```
mysql> select length
    -> (product) from gulf_country where rate=30;
+------------------+
| length
(product) |
+------------------+
|                9 |
+------------------+
1 row in set (0.00 sec)
```

- Built in function min

```
mysql> select min(rate),product from state_govt;
+-----------+---------+
| min(rate) | product |
+-----------+---------+
|        72 | DISEL   |
+-----------+---------+
1 row in set (0.00 sec)
```

GROUP FUNCTIONS

- Use of avg

```
+-----------+
| avg(rate) |
+-----------+
|  112.6000 |
+-----------+
1 row in set (0.00 sec)
```

- Use of distinct

```
mysql> select distinct comp_id from oil_companies;
+---------+
| comp_id |
+---------+
| CNG1    |
| HPCL1   |
| IOCL1   |
+---------+
3 rows in set (0.00 sec)
```

DATE AND TIME FUNCTION

```
mysql> select now();
+---------------------+
| now()               |
+---------------------+
| 2022-05-07 13:27:18 |
+---------------------+
1 row in set (0.00 sec)
```

```
mysql> select sysdate();
+---------------------+
| sysdate()           |
+---------------------+
| 2022-05-07 13:28:42 |
+---------------------+
1 row in set (0.00 sec)
```

CLAUSES

- Order BY clause

```
mysql> select * from oil_companies order by sr_no;
+---------+------+------+----------+--------------+------------+-------+
| comp_id | rate | tax  | quantity | product      | state_code | sr_no |
+---------+------+------+----------+--------------+------------+-------+
| IOCL1   |   85 |   15 |    84783 | DISEL        | MH         |     1 |
| IOCL1   |   80 |   25 |     8783 | DISEL        | CH         |     2 |
| IOCL1   |   95 |   14 |     8473 | DISEL        | PN         |     3 |
| IOCL1   |   88 |   18 |     8478 | DISEL        | HR         |     4 |
| IOCL1   |   70 |   20 |    74698 | CNG          | MH         |     5 |
| IOCL1   |   72 |   15 |    84783 | CNG          | CH         |     6 |
| HPCL1   |  103 |   33 |  8493803 | PETROL       | MH         |     7 |
| HPCL1   |   97 |   37 |     8493 | PETROL       | TN         |     8 |
| CNG1    |  115 |   40 |     8493 | POWER PETROL | MH         |     9 |
+---------+------+------+----------+--------------+------------+-------+
9 rows in set (0.00 sec)
```

- Group By clause

```
mysql> select state_code from state_govt group by state_code;
+------------+
| state_code |
+------------+
| CH         |
| MH         |
| TN         |
+------------+
3 rows in set (0.00 sec)
```

6. Conclusion

The project “Fuel Flow Management System” is for computerizing the physical world of the fuel flow database system. This project allows organizer to store all the data related to the fuel .It allows organizer to retrieve the necessary details and also provides the facility to ALTER,MODIFY,DELETE and UPDATE the database .The system takes care of all the technical details and is capable to provide easy and effective storage of information related the gulf countries,export countries,state government,user,owner .All the export countries,owner have unique ID for fast searching and transaction .All the entities are normalized in order to reduce the duplicity of records. According to implemented test case queries , it leads to the conclusion that the system helps for efficient management of data, reduces the access time and is able to perform desired operations.