

Project Report for Database Management System
(UCS310) On

DRUG DISTRIBUTION SYSTEM

(COE 15)

By

Manas Khatri	101703317
Manik Mangla	101703324
Maninder Singh	101703325

Submitted to-
Dr. Anil Vashisht



DEPARTMENT OF COMPUTER ENGINEERING
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, PATIALA,
PUNJAB
INDIA

January-May (2019)

INDEX

S.No	Topic	Page Number
1.	Introduction of Project	3
2.	Introduction to SQL & PL/SQL	3
3.	ER Diagram	4
4.	Conversion of ER Diagram to Table	5
5.	Normalized Tables	6
6.	Snap Shots	7
7.	PL/SQL codes	11
8.	References	14

INTRODUCTION

Introduction to the Project:

The project entitled Drug Distribution System is developed for a detailed storage of records for various pharmacies(shops), their contract with Drug manufacturing companies, the drugs they sell.

Various shops have contract with various pharmacy companies and they sell various drugs. Pharmacy companies make the drugs and transport them to the corresponding shops. All these relations(tables) have their associated attributes, which is illustrated in the ER diagram show below.

Technique Used:

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java. This tutorial will give you great understanding on PL/SQL to proceed with Oracle database and other advanced RDBMS concepts.

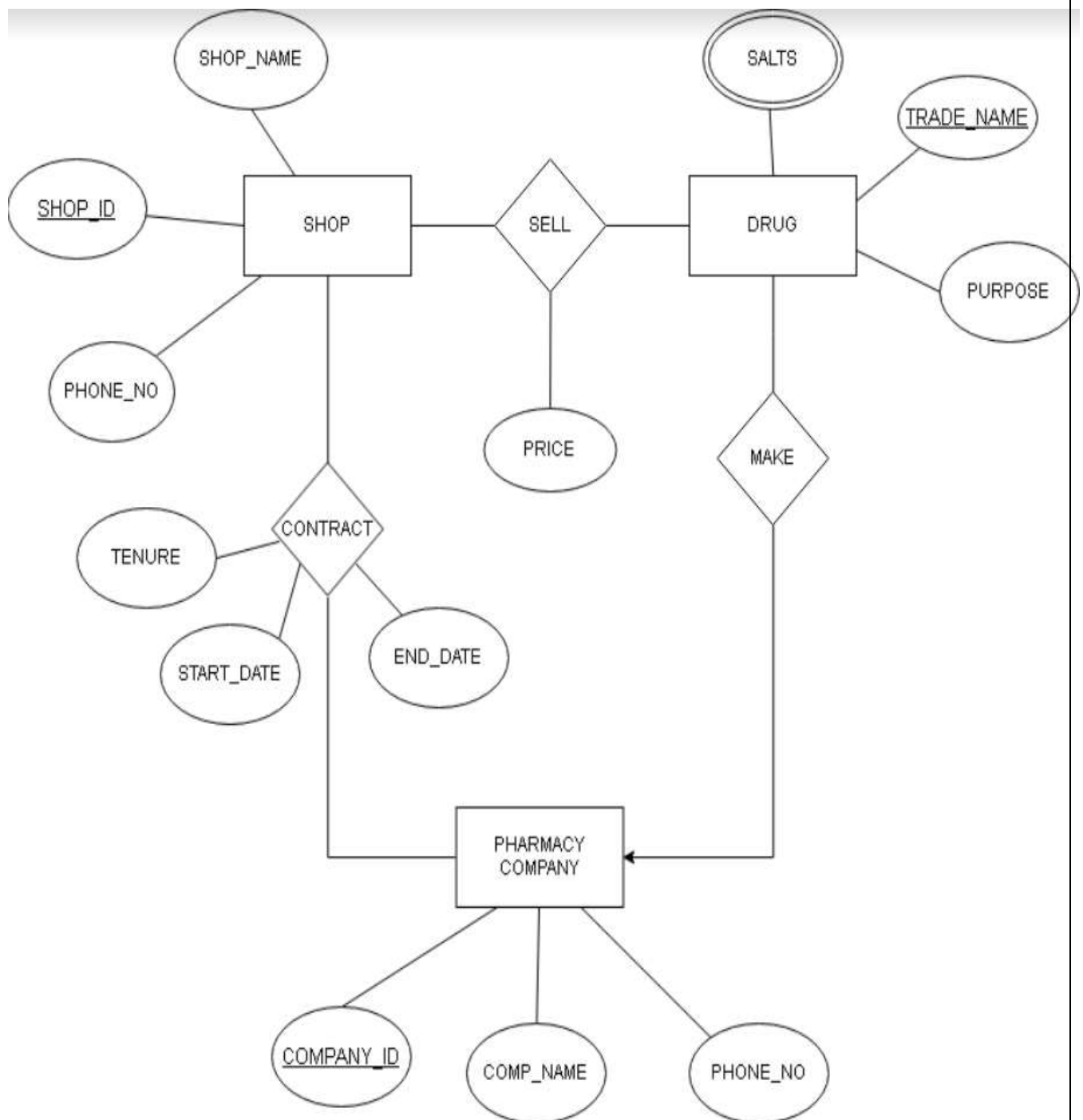
PL/SQL allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.

PL/SQL gives high productivity to programmers as it can query, transform, and update data in a database.

PL/SQL provides support for Object-Oriented Programming. It also provides support for developing Web Applications and Server Pages.

PL/SQL saves time on design and debugging by strong features, such as exception handling, encapsulation, data hiding, and object-oriented data types.

ER DIAGRAM



CONVERSION OF ER DIAGRAM TO TABLES:

SHOP TABLE:

<u>SHOP ID</u>	NAME	PHONE_NO
----------------	------	----------

DRUG TABLE:

<u>TRADE NAME</u>	PURPOSE	SALTS	COMP_ID
-------------------	---------	-------	---------

PHARMACY TABLE:

<u>COMP ID</u>	NAME	PHONE NO
----------------	------	----------

CONTRACT TABLE:

<u>SHOP ID</u>	<u>COMP ID</u>	TENURE	END DATE	START DATE
----------------	----------------	--------	----------	------------

SELL TABLE:

<u>SHOP ID</u>	<u>TRADE NAME</u>	PRICE
----------------	-------------------	-------

Normalized Tables

DRUG TABLE :

TRADE NAME	PURPOSE	SALTS	COMP_ID
T1	P1	S1	C1
T1	P1	S2	C1
T2	P2	S2	C2
T2	P3	S3	C2
T3	P1	S1	C3

This drug table contains the multivalued attribute “SALTS” and hence this table is not in 1 NF. To normalize this we decomposed drug table into two tables : Drug and DRUG_SALTS .

DRUG:

TRADE NAME	PURPOSE	COMP_ID
T1	P1	C1
T2	P2	C2
T3	P3	C3

DRUG_SALTS:

TRADE NAME	SALTS
T1	S1
T1	S2
T2	S2
T2	S3
T3	S1

CONTRACT TABLE:

<u>SHOP ID</u>	<u>COMP ID</u>	TENURE	<u>END DATE</u>	<u>START DATE</u>
<u>S1</u>	<u>C1</u>	T1	Date1	Date2
<u>S1</u>	<u>C2</u>	T2	Date3	Date4
<u>S2</u>	<u>C1</u>	T3	Date5	Date6

The combination of(Shop_id,comp_id,end_date,start_date) is made candidate key and tenure is derived from this combination which is also Super key and hence contract table is in BCNF.

SNAPSHOTS OF TABLES :

Snapshot for shop table :

It includes Shop_id,name ,Phone_no. shop_id is the primary key.

This table is in 1NF as each column contains singular values, each column contains a value that are of same type, each column has a unique name and finally the order in which data is saved doesn't matter. This table is also in 2NF as there are no partial dependencies(name and phone_no both can be determined from shop_id).this is in 3NF and BCNF because shop_id is a super key.

```
SQL> desc shop;
```

Name	Null?	Type
SHOP_ID	NOT NULL	NUMBER(5)
NAME		VARCHAR2(10)
PHONE_NO		NUMBER(12)

```
SQL> select constraint_name,column_name from user_cons_columns where table_name='SHOP';
```

CONSTRAINT_NAME	COLUMN_NAME
SYS_C007128	SHOP_ID

Snapshots for drug table:

Its attributes are trade_name and purpose. Trade_name is the primary key for this table. This table is in 1NF as all the conditions for 1st normal form are satisfied. This is in BCNF also because trade_name is a super key and purpose can be derived from trade_name.

```
SQL> desc drug;
Name                                         Null?    Type
-----
TRADE_NAME                                  NOT NULL VARCHAR2(20)
PURPOSE                                     VARCHAR2(40)
COMP_ID                                     NUMBER(5)
```

```
SQL> select constraint_name,column_name from user_cons_columns where table_name='DRUG';

CONSTRAINT_NAME      COLUMN_NAME
-----
SYS_C007132          TRADE_NAME
ID_FK2                COMP_ID
```

Snapshots for pharmacy table:

This table is in 1NF as the data is in singular form. It is also in 2NF as there is no partial dependency. Comp_id is the primary key which derives both name and phone_no which are non prime attributes. This is in 3NF and BCNF because comp_id is a super key (name and phone_no can be determined by comp_id).

```
SQL> select * from pharmacy;

COMP_ID  NAME          PHONE_NO
-----
12  RK Pharma    8.9083E+10
69  manimed      9.2929E+10
70  manglamed    3.7550E+10
71  manik_69     6.9697E+11
72  singh_med    4847596743
```



```
SQL> desc pharmacy;
```

Name	Null?	Type
COMP_ID	NOT NULL	NUMBER(5)
NAME		VARCHAR2(10)
PHONE_NO		NUMBER(12)

```
SQL> select constraint_name,column_name from user_cons_columns where table_name='PHARMACY';
```

CONSTRAINT_NAME	COLUMN_NAME
SYS_C007121	COMP_ID

Snapshots for contract table:

Contract table contains attributes such as shop_id, comp_id, tenure, end_date, start_date. Shop_id and comp_id are foreign key referenced to the tables shop(shop_id) and pharmacy(comp_id) respectively.

```
SQL> desc contract;
```

Name	Null?	Type
SHOP_ID	NOT NULL	NUMBER(5)
COMP_ID	NOT NULL	NUMBER(5)
END_DATE		DATE
START_DATE		DATE
TENURE		NUMBER(5)

SHOP_ID	COMP_ID	TENURE	END_DATE	START_DAT
1	12	395	22-DEC-21	22-NOV-20
2	12	761	22-DEC-21	22-NOV-19
3	69	761	22-DEC-21	22-NOV-19
3	70	761	22-DEC-21	22-NOV-19
3	71	761	22-DEC-21	22-NOV-19
3	72	761	22-DEC-21	22-NOV-19
1	72	761	22-DEC-21	22-NOV-19
1	70	761	22-DEC-21	22-NOV-19
4	69	761	22-DEC-21	22-NOV-19
4	70	761	22-DEC-21	22-NOV-19
4	71	761	22-DEC-21	22-NOV-19

Snapshot for salts drugs table:

Trade_name is a super key and salt is derived from trade_name hence it is in BCNF.

```
SQL> select constraint_name,column_name from user_cons_columns where table_name='SALTS_DRUGS';
```

CONSTRAINT_NAME	COLUMN_NAME
PK_DS	TRADE_NAME
PK_DS	SALT
SYS_C007146	TRADE_NAME

Snapshots for sel table:

This table is in BCNF because price is determined by combination of shop_id and trade_name(combination of these keys form a super key), combination of which are unique.so this table is in BCNF.

```
SQL> desc sel;
```

Name	Null?	Type
SHOP_ID	NOT NULL	NUMBER(5)
TRADE_NAME	NOT NULL	VARCHAR2(20)
PRICE		NUMBER(10)

CONSTRAINT_NAME	COLUMN_NAME
PK_ST	SHOP_ID
PK_ST	TRADE_NAME
T_FK	TRADE_NAME
S_FK	SHOP_ID

```
SQL> select * from sel;
```

SHOP_ID	TRADE_NAME	PRICE
1	aliloc	200
1	nici plus	100
1	glycocin	150
2	glycocin	152
2	betaden	152
2	nici plus	102
3	nici plus	102
3	betaden	162
3	crocin	162
4	crocin	152
4	aliloc	192

PL/SQL CODE:

Procedures:

A Stored Procedure is created which adds the drugs into drug table and correspondingly into the make table (relation table).

```
SQL> create or replace procedure add_drug( TEMP_NAME drug.TRADE_NAME%TYPE,TEMP_PURPOSE drug.PURPOSE%TYPE,TEMP_COMP make.COMP_ID%TYPE)
2 as
3 begin
4 insert into drug values(TEMP_NAME,TEMP_PURPOSE);
5
6 insert into make values(TEMP_NAME,TEMP_COMP);
7
8 end;
9 /
```

Triggers:

This trigger is created to check the date before insertion or deletion or updation of end_date or start_date. It will be triggered if the end date is greater than the start date.

```
SQL> create or replace trigger check_date
  2  before insert or update of end_date,start_date on contract
  3  for each row
  4
  5  begin
  6  if(:new.end_date<:new.start_date) then
  7  RAISE_APPLICATION_ERROR(-20003,'END DATE NEEDS TO GREATER THAN START DATE');
  8  end if;
  9  end;
 10  /

Trigger created.

SQL>
```

This trigger is created to check the price of the medicine before insertion or deletion or updation. An application error is raised if the new price is greater than 1.05 times the old price.

```
SQL> create or replace trigger check_price
  2  before insert or update  on sel
  3  for each row
  4
  5  begin
  6  if(:new.price>1.05*:old.price) then
  7  RAISE_APPLICATION_ERROR(-20003,'PRICES CANT BE INCREASED BY MORE THAN 5');
  8  end if;
  9  end;
 10  /

Trigger created.
```

This trigger updates the tenure in contract table by putting the difference of new end_date and new start_date in the tenure column.

```
SQL> create or replace trigger update_ten
  2  before insert or update of end_date,start_date on contract
  3  for each row
  4  declare
  5  i number;
  6
  7  begin
  8  :new.tenure:=:new.end_date-:new.start_date;
  9
 10  end;
 11  /
```

Trigger created.

```
SQL>
```

References

Tutorialspoint:

https://www.tutorialspoint.com/plsql/plsql_overview.htm

geeksforgeeks:

<https://www.geeksforgeeks.org/plsql-introduction/>