Database Design
CS 6360.002: Project Report: Dry Cleaner

Due on Wednesday December 7, 2016 at 11:59pm

Nurcan Yuruk

Lizhong Zhang (lxz160730) Yikai Gao (yxg131830)

Contents

1 Requirement

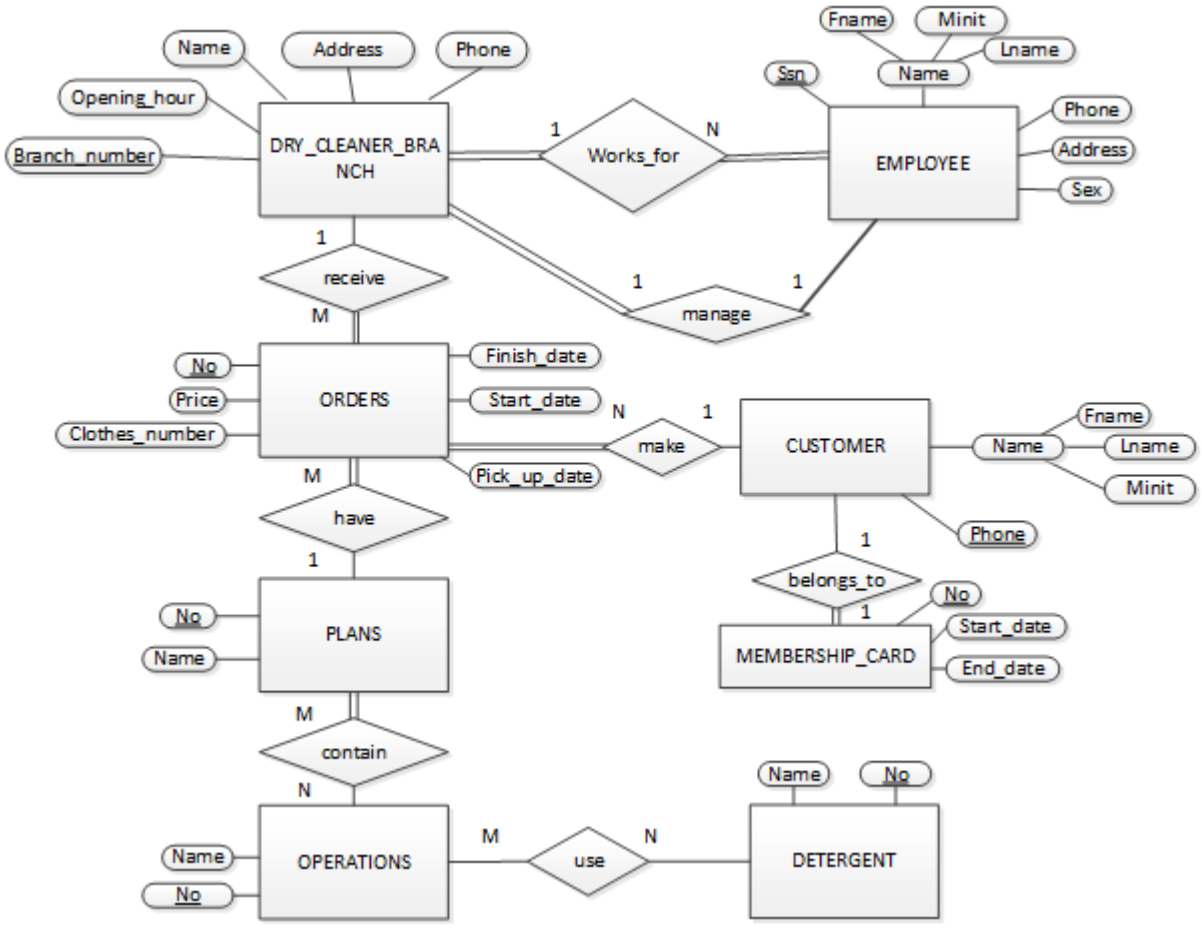
The Dry Cleaner shall include a whole operation and service system recording some necessary operations that dry cleaner needs to finish orders and handling business. The requirements of Dry Cleaner system shall include the following assumptions and functions:

1. The system shall have many branches of Dry Cleaner, each of which is in the same database system and has same system organization. And Dry Cleaner shall have several attributions including Branch_number, Opening_hour, Name, Address, Phone and Manager_ssn.
2. Every Dry Cleaner shall have at least one employees, and every employee works for only one Dry Cleaner. Dry Cleaner shall keep every employees information including Ssn, name, Phone, Address, Sex and Branch_no.
3. Every branch shall have a manager who manages only one Dry Cleaner.
4. Every Dry Cleaner shale receive many orders, and every order shall belong to only one Dry Cleaner. Orders shall have many attributions which includes No, Price, Clothes_number, Start_date, Plan_no, Finish_date, Pickup_date, Customer_phone.
5. Each order shall belong to only one customer, but each customer could place at least one orders. Each customer shall leave their own information including Phone, name in the Dry Cleaner.
6. Every customer could apply at most one membership card that belongs to only one customer. And each membership card shall have No, Start_date, End_date and Customer_phone.
7. There shall be many different plans to meet the needs of customers. And every order owns only one plan, however one plan could exist in disparate order.
8. Each plan has a plan number and a name, at the same time, same plan has a fixed time to finished a series of operations.
9. Each plan in Dry Cleaner needs at least one operations which include dry-clean, wash, ironing, repair, care and maintenance. In the meantime, each operation could exist in many plans.
10. Every operation could utilize at least one cleaners which include laundry detergent, powder detergent, detergent pacs & tablets, fabric softener, bleach, starch & anti-static sprays, lint removal, fabric deodorizer and stain removal. And each of these cleaners could be used in different operations.

2 ER Diagram

The ER diagram is as follow.

Figure 1: ER Diagram

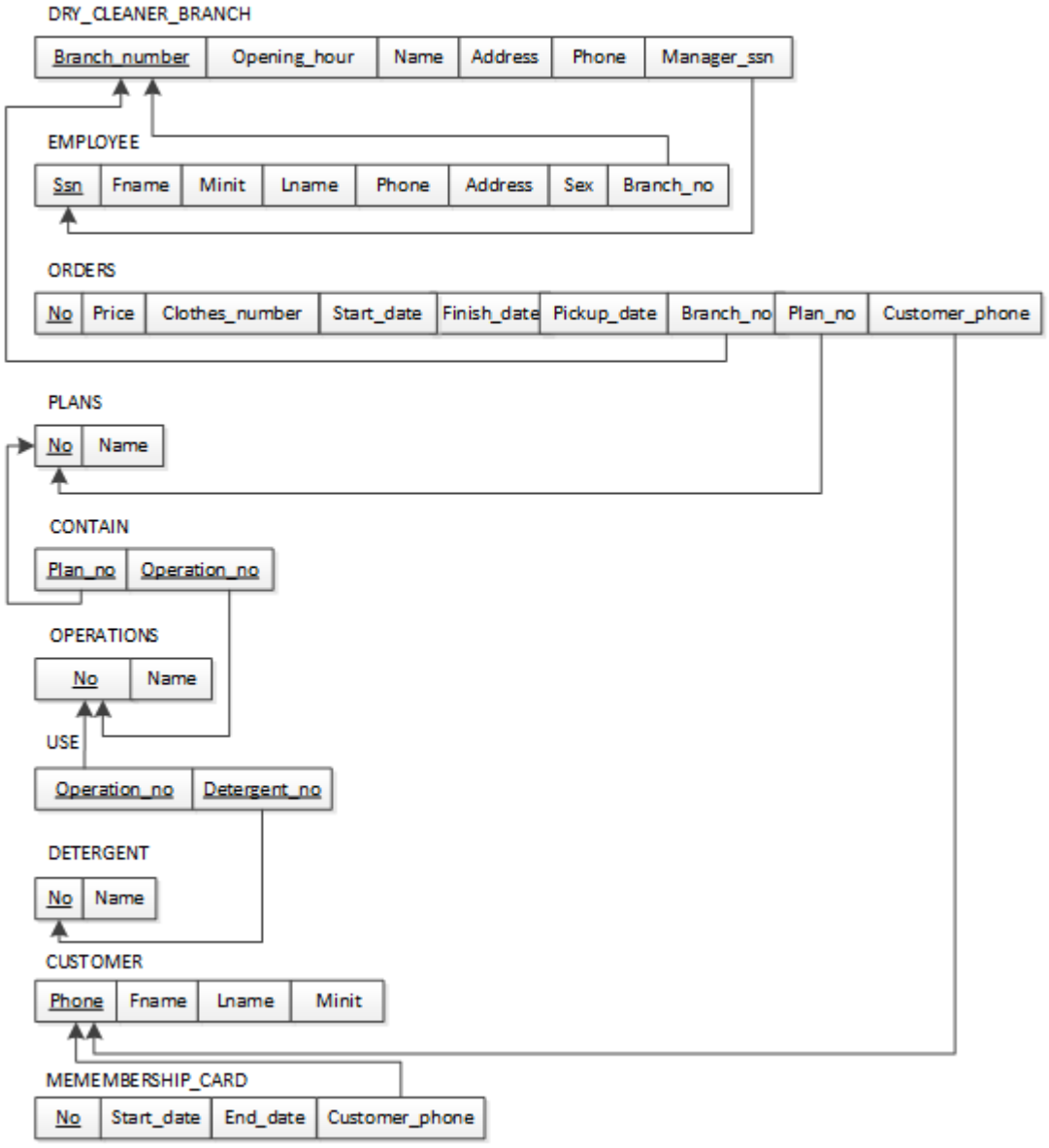


3 Mapping and Normalization

3.1 First Draft of Relational Schema

First draft of relational schema after mapping is as follow.

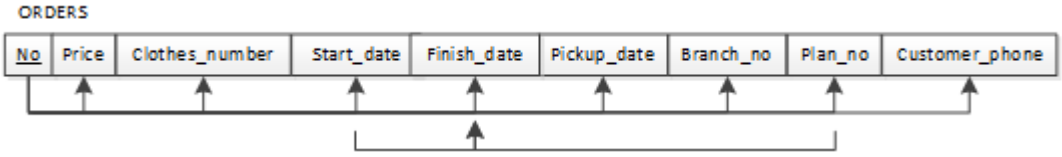
Figure 2: Relational Schema



3.2 Normalization

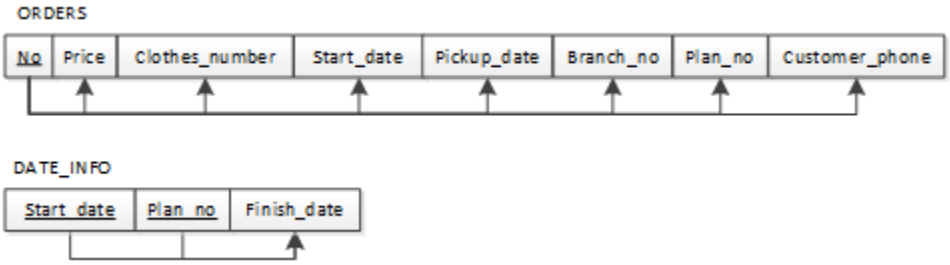
In this schema, we have nine tables that are in 3NF. Only ORDERS relation schema is not in 3NF. ORDER table has following relations:

Figure 3: Violate 3NF



Thus this violates 3NF, we normalize ORDERS into 3NF:

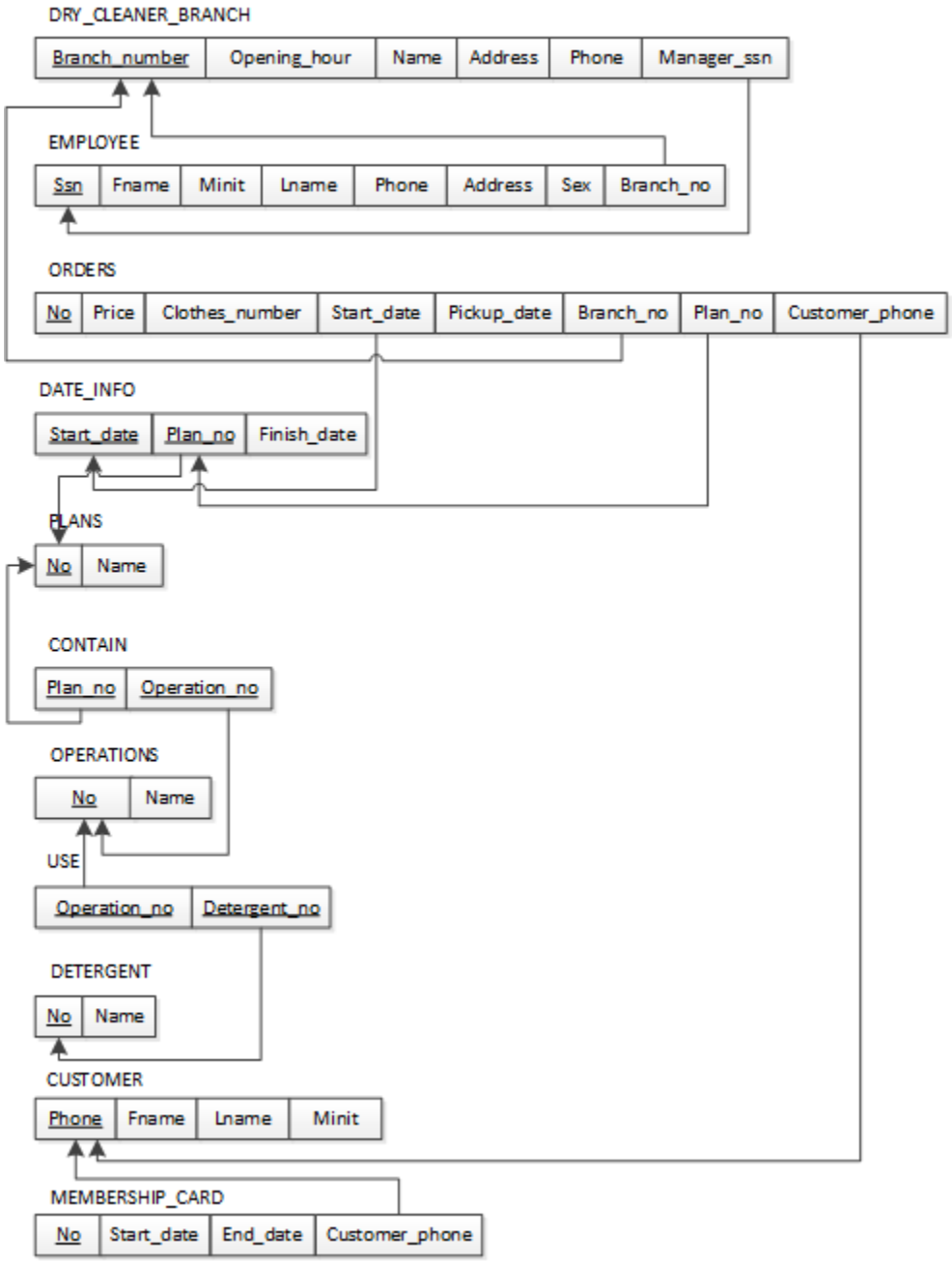
Figure 4: Normalize to 3NF



3.3 Final Schema

The final schema is as follow:

Figure 5: Final Relational Schema



4 SQL

SQL statements to create tables are as follow:

```
1  create table OPERATIONS
2  ( No  varchar(10),
3    Name  varchar(30)  not null,
4    primary key (No));
5
6  create table PLANS
7  ( No  varchar(10),
8    Name  varchar(15)  not null,
9    primary key (No));
10
11 create table DETERGENT
12 ( No  varchar(10),
13   Name  varchar(50)  not null,
14   primary key (No));
15
16 create table CUSTOMER
17 ( Phone  varchar(10),
18   Fname  varchar(15)  not null,
19   Minit  varchar(15)  not null,
20   Lname  varchar(15)  not null,
21   primary key (Phone));
22
23 create table MEMBERSHIP_CARD
24 ( No  varchar(10),
25   Start_date  date  not null,
26   End_date  date  not null,
27   Customer_phone  varchar(10)  not null,
28   primary key (No),
29   foreign key (Customer_phone) references CUSTOMER (Phone)
30     on delete cascade);
31
32 create table USE
33 ( Operation_no  varchar(10),
34   Detergent_no  varchar(10),
35   primary key (Operation_no, Detergent_no),
36   foreign key (Operation_no) references OPERATIONS (No)
37     on delete cascade,
38   foreign key (Detergent_no) references DETERGENT (No)
39     on delete cascade);
```

```
40
41 create table CONTAIN
42 ( Plan_no  varchar(10),
43   Operation_no  varchar(10)  not null,
44   primary key (Plan_no, Operation_no),
45   foreign key (Plan_no) references PLANS (No)
46     on delete cascade,
47   foreign key (Operation_no) references OPERATIONS (No)
48     on delete cascade);
49
50 create table DRY_CLEANER_BRANCH
51 ( Branch_number  varchar(10),
52   Opening_hour  varchar(20)  not null,
53   Name  varchar(15)  not null,
54   Address  varchar(30)  not null,
55   Phone  varchar(10)  not null,
56   Manager_ssn  varchar(9)  not null,
57   primary key (Branch_number));
58
59 create table EMPLOYEE
60 ( Ssn  varchar(9),
61   Fname  varchar(15)  not null,
62   Minit  varchar(15)  not null,
63   Lname  varchar(15)  not null,
64   Phone  varchar(10)  not null,
65   Address  varchar(30)  not null,
66   Sex  varchar(1)  not null,
67   Branch_no  varchar(10)  not null,
68   primary key (Ssn),
69   foreign key (Branch_no) references DRY_CLEANER_BRANCH (Branch_number)
70     on delete cascade);
71
72 create table DATE_INFO
73 ( Start_date  date,
74   Plan_no  varchar(10),
75   Finish_date  date  not null,
76   primary key (Start_date, Plan_no),
77   foreign key (Plan_no) references PLANS (No)
78     on delete cascade);
79
80 create table ORDERS
81 ( No  varchar(10),
82   Price  number  not null,
```

```
83   Clothes_number  varchar(10)  not null,
84   Start_date  date not null,
85   Pickup_date  date default null,
86   Branch_no  varchar(10)  not null,
87   Plan_no  varchar(10) not null,
88   Customer_phone  varchar(10)  not null,
89   primary key (No),
90   foreign key (Start_date, Plan_no) references DATE_INFO (Start_date,
        Plan_no)
91     on delete cascade,
92   foreign key (Branch_no) references DRY_CLEANER_BRANCH (Branch_number)
93     on delete cascade,
94   foreign key (Customer_phone) references CUSTOMER (Phone)
95     on delete cascade);
96
97 ------------------------------------------------------------------------
98 --After data insertion to existed table, execute this instruction.-
99 ------------------------------------------------------------------------
100 alter table DRY_CLEANER_BRANCH
101   add constraint mgr_ssn_constraint
102   foreign key (Manager_ssn) references EMPLOYEE (Ssn)
103   on delete cascade;
```

5 PL/SQL for Procedures & Triggers

```
1  ------------------------------------------------------------
2  --------------------------PROCEDURE 1-----------------------
3  --Select all orders' No whose finish_date is after sysdate.----
4  ------------------------------------------------------------
5  create or replace procedure check_order as
6  cursor order_information is
7  select ORDERS.No, DATE_INFO.Finish_date
8  from ORDERS, DATE_INFO
9  where ORDERS.Start_date = DATE_INFO.Start_date
10   AND ORDERS.Plan_no = DATE_INFO.Plan_no;
11 order_info  order_information%rowtype;
12 begin
13   open order_information;
14   loop
15     fetch order_information into order_info;
16     exit when order_information%NOTFOUND;
17     if (to_date(order_info.Finish_date,'DD-MON-YY') >
18       to_date(SYSDATE,'DD-MON-YY')) then
19       DBMS_OUTPUT.PUT_LINE ('Order No: ' || order_info.No);
20     end if;
21   end loop;
22   close order_information;
23 end;
24 /
25 ------------------------------------------------------------
26 --------------------------PROCEDURE 2-----------------------
27 --Select all names and phones of customers whose---------------
28 --membership cards would expire next month.--------------------
29 ------------------------------------------------------------
30 create or replace procedure check_card as
31 cursor card_information is
32 select CUSTOMER.Phone, CUSTOMER.Fname, CUSTOMER.Minit,
33   CUSTOMER.Lname, MEMBERSHIP_CARD.End_date
34 from CUSTOMER, MEMBERSHIP_CARD
35 where MEMBERSHIP_CARD.Customer_phone = CUSTOMER.Phone;
36 card_info  card_information%rowtype;
37 begin
38   open card_information;
39   loop
40     fetch card_information into card_info;
41     exit when card_information%NOTFOUND;
```

```
42      if (to_date(card_info.End_date, 'DD-MON-YY') <
43        to_date(SYSDATE,'DD-MON-YY') + 30) then
44        DBMS_OUTPUT.PUT_LINE ('Customer Phone: ' || card_info.Phone);
45        DBMS_OUTPUT.PUT_LINE ('Customer Fname: ' || card_info.Fname);
46        DBMS_OUTPUT.PUT_LINE ('Customer Minit: ' || card_info.Minit);
47        DBMS_OUTPUT.PUT_LINE ('Customer Lname: ' || card_info.Lname);
48      end if;
49    end loop;
50    close card_information;
51  end;
52  /
53
54  ------------------------------------------------------------
55  ----------------------TRIGGER 1-----------------------------
56  -------------Add a new log when a new order comes.----------
57  ------------------------------------------------------------
58  drop table Order_log;
59  /
60  create table Order_log
61  ( No  varchar(10),
62    Start_date  date,
63    Log_date  date,
64    primary key (No));
65  /
66  create or replace trigger new_log
67    after insert on ORDERS
68    for each row
69  begin
70    insert into Order_log(No, Start_date, Log_date) values
71      (:new.No, :new.Start_date, sysdate);
72  end;
73  /
74  ------------------------------------------------------------
75  ----------------------TRIGGER 2-----------------------------
76  --Make a new log to record the number of the-----------------
77  --same customer's orders.-------------------------------------
78  ------------------------------------------------------------
79  drop table Customer_log;
80  /
81  create table Customer_log
82  ( Phone  varchar(10),
83    Number_of_orders  number,
84    primary key (Phone));
```

```
85 /
86 create or replace trigger customer_number_log
87   after insert on ORDERS
88   for each row
89 begin
90   merge into Customer_log using dual on(Phone = :new.Customer_Phone)
91   when not matched then
92   insert (Phone, Number_of_orders) values (:new.Customer_Phone, 1)
93   when matched then
94   update set Number_of_orders = Number_of_orders + 1;
95 end;
96 /
```

6 CHECK constraint and PL/SQL for Business Rules

```
1  ------------------------------------------------------------------------
2  --This rule is to check if order's pick_up date is before start _date,
3  --if it is, then output an error: this time is a wrong time.----------
4  ------------------------------------------------------------------------
5  create or replace trigger rule_1
6    after insert or update of Pickup_date
7      on ORDERS
8    for each row
9  begin
10   if :new.Pickup_date < :new.Start_date then
11     DBMS_OUTPUT.PUT_LINE ('Error: ' || :new.Pickup_date || 'is a wrong
         time.');
12   end if;
13 end;
14 /
15
16 ------------------------------------------------------------------------
17 --This rule is to check if an employee's Ssn's length is 9.-----------
18 ------------------------------------------------------------------------
19 ALTER TABLE EMPLOYEEE
20 ADD CONSTRAINT constraint_Ssn_format CHECK (LENGTH(Ssn) = 9);
21 /
```

7 Sample Data

```
insert into OPERATIONS
values ('1', 'dry-clean');
insert into OPERATIONS
values ('2', 'wash');
insert into OPERATIONS
values ('3', 'care and maintenance');

insert into PLANS
values ('1', 'Plan A');
insert into PLANS
values ('2', 'Plan B');
insert into PLANS
values ('3', 'Plan C');
insert into PLANS
values ('4', 'Plan D');
insert into PLANS
values ('5', 'Plan E');


insert into DETERGENT
values ('1', 'laundry detergent');
insert into DETERGENT
values ('2', 'powder detergent');
insert into DETERGENT
values ('3', 'detergent pacs and tablets');
insert into DETERGENT
values ('4', 'fabric softener');
insert into DETERGENT
values ('5', 'bleach');
insert into DETERGENT
values ('6', 'starch and anti-static sprays');
insert into DETERGENT
values ('7', 'lint removal');
insert into DETERGENT
values ('8', 'fabric deodorizer');
insert into DETERGENT
values ('9', 'stain removal');

insert into CUSTOMER
values ('4695629999', 'Aaa', 'Wss', 'D');
insert into CUSTOMER
```

```
values ('4695628888', 'Baa', 'Ess', 'B');
insert into CUSTOMER
values ('4695627777', 'Caa', 'Rss', 'W');
insert into CUSTOMER
values ('4695626666', 'Daa', 'Sss', 'T');

insert into MEMBERSHIP_CARD
values ('11111', to_date('02-AUG-16','DD-MON-RR'),
        to_date('02-AUG-19','DD-MON-RR'), '4695629999');
insert into MEMBERSHIP_CARD
values ('22222', to_date('25-MAY-16','DD-MON-RR'),
        to_date('25-MAY-19','DD-MON-RR'), '4695628888');
insert into MEMBERSHIP_CARD
values ('33333', to_date('15-JUL-15','DD-MON-RR'),
        to_date('15-JUL-18','DD-MON-RR'), '4695627777');

insert into USE
values ('1', '3');
insert into USE
values ('2', '1');
insert into USE
values ('2', '2');
insert into USE
values ('2', '4');
insert into USE
values ('3', '5');
insert into USE
values ('3', '6');
insert into USE
values ('3', '7');
insert into USE
values ('3', '8');
insert into USE
values ('3', '9');

insert into CONTAIN
values ('1', '1');
insert into CONTAIN
values ('2', '2');
insert into CONTAIN
values ('3', '3');
insert into CONTAIN
values ('4', '1');
```

```
insert into CONTAIN
values ('4', '3');
insert into CONTAIN
values ('5', '2');
insert into CONTAIN
values ('5', '3');

insert into DRY_CLEANER_BRANCH
values ('1', '9:00-18:00', 'Dry Cleaner1', 'Plano',
        '4695621111', '123456789');
insert into DRY_CLEANER_BRANCH
values ('2', '9:00-18:00', 'Dry Cleaner2', 'Richardson',
        '4695622222', '234567890');

insert into EMPLOYEE
values ('111111111', 'Sgffg', 'F', 'Egfdfg',
        '4424565555', 'Plano', 'M', '1');
insert into EMPLOYEE
values ('222222222', 'Wgd', 'W', 'Fgf',
        '4424566666', 'Plano', 'F', '1');
insert into EMPLOYEE
values ('123456789', 'Egdf', 'B', 'Efdgfd',
        '4424562222', 'Plano', 'F', '1');
insert into EMPLOYEE
values ('999999999', 'Ssdd', 'F', 'Egdfbb',
        '4424562442', 'Plano', 'F', '2');
insert into EMPLOYEE
values ('234567890', 'Mfdf', 'H', 'Uddff',
        '4424561444', 'Richarsdon', 'M', '2');

insert into DATE_INFO
values (to_date('30-NOV-16','DD-MON-RR'), '1',
        to_date('2-DEC-16','DD-MON-RR'));
insert into DATE_INFO
values (to_date('12-NOV-16','DD-MON-RR'), '2',
        to_date('14-DEC-16','DD-MON-RR'));

insert into ORDERS
values ('1', '10.00', '110', to_date('30-NOV-16','DD-MON-RR'),
        null, '1', '1', '4695629999');
insert into ORDERS
values ('2', '12.22', '111', to_date('12-NOV-16','DD-MON-RR'),
        to_date('15-NOV-16','DD-MON-RR'), '2', '2', '4695628888');
```