Views

1. Customer_v – for each customer, indicate his or her name as well as the customer type (prospect, steady or premier) as well as the number of years that customer has been with us.

```
CREATE VIEW Customer_v AS
select NAME, YEARS, 'PREMIER' as "TYPE"
from CUSTOMERS
NATURAL JOIN PREMIER
union
select NAME, YEARS, 'STEADY' as "TYPE"
from CUSTOMERS
NATURAL JOIN STEADY
union
select NAME, YEARS, 'PROSPECT' as "TYPE"
from CUSTOMERS
NATURAL JOIN PROSPECTIVE
```

2. Customer_addresses_v – for each customer, indicate whether they are an individual or a corporate account, and display all of the addresses that we are managing for that customer.

```
CREATE VIEW Customer_addresses_v AS
select NAME, ADDRESS AS ADDRESS, 'Corporate' as "Acconut TYPE"
from CUSTOMERS
NATURAL JOIN ADDRESS
union
select NAME, MAILINGADD AS ADDRESS, 'Individual' as "Acconut TYPE"
from CUSTOMERS
NATURAL JOIN INDIVIDUAL
```

3. Mechanic_mentor_v – reports all of the mentor/mentee relationships at Dave's, sorted by the name of the mentor, then the name of the mentee.

```
CREATE VIEW Mechanic_mentor_v AS
SELECT A.FNAME AS MENTOR_FNAME, A.LNAME AS MENTOR_LNAME, B.FNAME AS MENTEE_
FNAME, B.LNAME AS MENTEE_LNAME
FROM EMPLOYEE A, EMPLOYEE B
WHERE A.EMPID IN
(SELECT MENTORID FROM MENTORSHIP)
AND
B.EMPID IN
(SELECT MENTEEID FROM MENTORSHIP)
ORDER BY MENTOR_FNAME, MENTOR_LNAME, MENTEE_FNAME, MENTEE_LNAME
```

4. Premier_profits_v – On a year by year basis, show the premier customer's outlay versus what they would have been charged for the services which they received had they merely been steady customers.

```
CREATE VIEW Premier_profits_v AS
select DISTINCT CUSID, NAME, ANNUALFEE, SUM(PRICE) AS TOTAL_SPENT, (ANNUALF
EE - SUM(PRICE)) AS PROFIT
from REPAIRINSTANCE
inner join ORDERLINE using (PACKAGEID)
inner join ORDERS using (ORDERID)
inner join VEHICLE using (VIN)
inner join CUSTOMERS using (CUSID)
inner join PREMIER using (CUSID)
WHERE CUSID IN
(SELECT CUSID FROM PREMIER)
GROUP BY CUSID, ANNUALFEE, NAME
```

5. Prospective_resurrection_v – List all of the prospective customers who have had three or more contacts, and for whom the most recent contact was more than a year ago. They might be ripe for another attempt.

```
CREATE VIEW Prospective_resurrection_v AS
SELECT A.NAME AS "3 OR MORE CONTACTS", B.NAME AS "LAST YEAR CONTACTED"
FROM CUSTOMERS A, CUSTOMERS B
WHERE A.CUSID IN
(select CUSID
from SPECIALINSTANCE
GROUP BY CUSID
HAVING COUNT(CUSID) >= 3)
AND
B.CUSID IN
(select CUSID
from SPECIALINSTANCE
GROUP BY CUSID, LASTDATECONTACTED
HAVING LASTDATECONTACTED <= '2016-05-11')
```

Queries

1. List the customers. For each customer, indicate which category he or she fall into, and his or her contact information.

```
select NAME, 'PREMIER' as "CATEGORY", PHONE, email
from CUSTOMERS
NATURAL JOIN PREMIER
union
select NAME, 'STEADY' as "CATEGORY", PHONE, email
from CUSTOMERS
NATURAL JOIN STEADY
union
select NAME, 'PROSPECT' as "CATEGORY", PHONE, email
from CUSTOMERS
NATURAL JOIN PROSPECTIVE;
```

2. For each service visit, list the total cost to the customer for that visit.

```
select NAME, (PRICE * numRepItem) AS "TOTAL COST"
from REPAIRINSTANCE
inner join ORDERLINE using (PACKAGEID)
inner join ORDERS using (ORDERID)
inner join VEHICLE using (VIN)
inner join CUSTOMERS using (CUSID)
WHERE CUSID IN
(SELECT CUSID FROM CUSTOMERS);
```

3. List the top three customers in terms of their net spending for the past two years, and the total that they have spent in that period.

```
SELECT DISTINCT CUS.NAME AS "NAME", (repI.PRICE * oli.numRepItem) AS TOTAL_
SPENDING --, SUM(repI.PRICE * oli.numRepItem) AS "NET SPENDING"
FROM CUSTOMERS CUS
    NATURAL JOIN VEHICLE
    NATURAL JOIN ORDERS ord
    inner join ORDERLINE oli on(ord.ORDERID = oli.ORDERID)
    inner join MAINTENANCEPACK map on(oli.PACKAGEID = map.PACKAGEID)
    inner join REPAIRINSTANCE repI on (map.PACKAGEID = repI.PACKAGEID)
    where ord.ORDERDATE between '2014-05-11' and '2017-05-11'
    GROUP BY CUS.NAME
    ORDER BY TOTAL_SPENDING DESC
    LIMIT 3;
```

4. Find all of the mechanics who have three or more skills.

```
SELECT
    EMPID, FNAME, LNAME, COUNT(*) AS "NUMBER OF SKILLS"
FROM
    SKILLMECHANIC
INNER JOIN
    MECHANIC USING (EMPID)
GROUP BY
    EMPID
HAVING
    COUNT(*) > 3;
```

5. Find all of the mechanics who have three or more skills in common.

```
CREATE VIEW MECHANIC_SKILL AS
SELECT * FROM SKILLMECHANIC
INNER JOIN MECHANIC USING (EMPID)
WHERE EMPID IN
(SELECT EMPID
FROM SKILLMECHANIC
INNER JOIN
    MECHANIC USING (EMPID)
GROUP BY EMPID
HAVING COUNT(*) > 3);

SELECT A.FNAME, A.LNAME, COUNT(A.SKILLID) AS COMMON_SKILLS_NUMBERS, B.FNAM
E, B.LNAME
FROM MECHANIC_SKILL A, MECHANIC_SKILL B
WHERE A.SKILLID = B.SKILLID
AND A.EMPID < B.EMPID
GROUP BY A.FNAME, A.LNAME, B.FNAME, B.LNAME
HAVING COUNT(A.SKILLID) >= 3
ORDER BY A.FNAME, A.LNAME, B.FNAME, B.LNAME;
```

6. For each maintenance package, list the total cost of the maintenance package, as well as a list of all of the maintenance items within that package.

```
SELECT PACKAGENAME AS PACKAGES ,NAMITEM AS ITEMS, PRICE AS TOTAL_COST
FROM MAINTENANCEPACK A
NATURAL JOIN REPAIRINSTANCE
WHERE PACKAGEID IN
(SELECT PACKAGEID FROM MAINTENANCEPACK)

UNION

(SELECT PACKAGENAME AS PACKAGES, ' ' AS ITEMS, SUM(PRICE) AS TOTAL_COST
FROM MAINTENANCEPACK
NATURAL JOIN REPAIRINSTANCE
WHERE PACKAGEID IN
(SELECT PACKAGEID FROM MAINTENANCEPACK)
GROUP BY PACKAGENAME)
ORDER BY PACKAGES, TOTAL_COST
```

7. Find all of those mechanics who have one or more maintenance items that they lacked one or more of the necessary skills.

```
SELECT EMPID, FNAME, LNAME, SKILLID FROM SKILLREPAIR
INNER JOIN REPAIRINSTANCE USING (NAMITEM)
INNER JOIN ORDERLINE USING (PACKAGEID)
INNER JOIN EMPLOYEE ON (EMPID = ASSIGNTO)
WHERE ASSIGNTO IN
(SELECT EMPID FROM MECHANIC)

EXCEPT

SELECT EMPID, FNAME, LNAME, SKILLID  FROM SKILLMECHANIC
NATURAL JOIN EMPLOYEE
WHERE EMPID IN
(SELECT ASSIGNTO FROM ORDERLINE)
ORDER BY EMPID, SKILLID
```

8. List the customers, sorted by the number of loyalty points that they have, from largest to smallest.

```
SELECT CUSID, NAME, LOYALPOINT
FROM STEADY
INNER JOIN CUSTOMERS USING (CUSID)
ORDER BY LOYALPOINT DESC
```

9. The premier customers and the difference between what they have paid in the past year, versus the services that they actually used during that same time. List from the customers with the largest difference to the smallest.

```
select DISTINCT CUSID, NAME, ANNUALFEE, SUM(PRICE) AS TOTAL_SPENT, (ANNUALF
EE - SUM(PRICE)) AS DIFFERENCE
from REPAIRINSTANCE
inner join ORDERLINE using (PACKAGEID)
inner join ORDERS using (ORDERID)
inner join VEHICLE using (VIN)
inner join CUSTOMERS using (CUSID)
inner join PREMIER using (CUSID)
WHERE CUSID IN
(SELECT CUSID FROM PREMIER)
GROUP BY CUSID, ANNUALFEE, NAME
ORDER BY DIFFERENCE DESC
```

10. Report on the steady customers based on the net profit that we have made from them over the past year, and the dollar amount of that profit, in order from the greatest to the least.

```
select DISTINCT CUSID, NAME, SUM(PRICE) AS NET_PROFIT
from REPAIRINSTANCE
inner join ORDERLINE using (PACKAGEID)
inner join ORDERS using (ORDERID)
inner join VEHICLE using (VIN)
inner join CUSTOMERS using (CUSID)
WHERE CUSID IN
(SELECT CUSID FROM STEADY)
AND ORDERDATE BETWEEN '2016-01-01' AND '2016-12-30'
GROUP BY CUSID, NAME
ORDER BY NET_PROFIT DESC
```

11. List the three services that we have performed the most in the last year and how many times they were performed.

```
SELECT CUSID, SUM(ANNUALFEE) AS TOTAL_SPENDING FROM PREMIER
GROUP BY CUSID
ORDER BY TOTAL_SPENDING DESC
LIMIT 3;
```

12. List the three services that have brought in the most money in the last year along with that amount of money.

```
SELECT MODEL, MAKE, MODYEAR FROM VEHICLE
WHERE VIN IN
(SELECT VIN FROM ORDERS
WHERE ORDERDATE BETWEEN '2014-01-01' AND '2016-12-30'
GROUP BY VIN
ORDER BY COUNT(VIN) DESC)
LIMIT 5;
```

13. Find the mechanic who is mentoring the most other mechanics. List the skills that the mechanic is passing along to the other mechanics.

```
SELECT FNAME, LNAME, SKILLNAME
FROM MECHANIC
INNER JOIN SKILLMECHANIC USING (EMPID)
INNER JOIN SKILL USING (SKILLID)
WHERE EMPID IN

(SELECT MENTORID
FROM MENTORSHIP
GROUP BY MENTORID
HAVING COUNT(MENTORID) in

(SELECT COUNT(*) AS MENTEE_NUMBER
FROM MENTORSHIP
GROUP BY MENTORID
ORDER BY MENTEE_NUMBER DESC
LIMIT 1;
```

14. Find the three skills that have the fewest mechanics who have those skills.

```
SELECT SKILLID, COUNT(SKILLID) AS RESULT
FROM SKILLMECHANIC
GROUP BY SKILLID
ORDER BY RESULT ASC
LIMIT 3;
```

15. List the employees who are both service technicians as well as mechanics.

```
SELECT EMPID, FNAME, LNAME
FROM EMPLOYEE
WHERE EMPID IN
(SELECT EMPID FROM  MECHANIC
WHERE EMPID IN
(SELECT EMPID FROM TECHNICIAN
WHERE EMPID IN
(SELECT EMPID FROM EMPLOYEE)));
```

16. Additional queries

a. List the corresponding skill that the mechanic has on the order that mechanic is working on.

```
SELECT FNAME, LNAME, SKILLNAME FROM SKILLMECHANIC
INNER JOIN SKILL USING (SKILLID)
INNER JOIN MECHANIC USING (EMPID)
WHERE EMPID IN
(SELECT ASSIGNTO FROM ORDERLINE)
ORDER BY FNAME, LNAME
```

b. A mechanic should have at least one skill in an individual maintenance responsible for.

```
select distinct EMPLOYEE.FNAME, EMPLOYEE.LNAME, SKILL.skillname From EMPLOY
EE
    inner join MECHANIC on EMPLOYEE.EMPID = MECHANIC.EMPID
inner join SKILL on MECHANIC.EMPID = SKILL.SkillID
```

c. A customer can only be a premier customer if he/she meets a credit score requirement.

```
select CUSTOMERS.NAME, PREMIER.annualFee * 10 as Current_Credit, 1000 as Cr
edit_Quirement, 'permier' as Customer_Type from CUSTOMERS
    inner join EXISTINGCUSTOMER on CUSTOMERS.CUSID = EXISTINGCUSTOMER.CUSID
    inner join PREMIER on EXISTINGCUSTOMER.CUSID = PREMIER.CUSID
```