



MACQUARIE
University
SYDNEY • AUSTRALIA

MACQUARIE UNIVERSITY
Faculty of Science and Engineering
Department of Computing

COMP2350/COMP6350 Database Systems 2021 (Semester 2)

Assignment 3 (Report)

Database Programming and Implementation (worth 10%)

Student Declaration:

I declare that the work reported here is my own. Any help received, from any person, through discussion or other means, has been acknowledged in the last section of this report.

Student Name: Runde Jia

Student Number: 44434065

费润德

Student Signature:

Date: 10/29/2021

1. The Initial State of the Database

Paste here the screen shots showing the provided sample data in the tables.

Product:

	ProductID	ProductType	PackageType	YearProduced	Price	Brand
▶	1	Wine	Bottle	1990	1000	Penfold
	2	Wine	Bottle	2000	400	Penfold
	3	Wine	Bottle	2010	200	Penfold
	4	Wine	Bottle	2020	50	Penfold
	5	Beer	Can	2020	10	Blonde
*	NULL	NULL	NULL	NULL	NULL	NULL

Campaign:

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
*	NULL	NULL	NULL

DiscountDetails:

	ProductID	CampaignID	MembershipLevel	Discount
▶	1	1	Gold	21
	1	1	Platinum	32
	2	1	Gold	13
	2	2	Silver	50
	3	3	Platinum	5
*	NULL	NULL	NULL	NULL

Membership:

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶	1	John	Smith	johnsmith@gmail.com	Gold	2021-12-31
	2	Peter	Cole	petercole@gmail.com	Silver	2021-10-15
	3	Sarah	Goodman	sgoodman@gmail.com	Platinum	2021-10-31
	4	Christ	Gordon	cgordon@gmail.com	Gold	2021-11-28
	5	Emma	Spring	espringh@gmail.com	Silver	2021-12-31
*	NULL	NULL	NULL	NULL	NULL	NULL

Branch:

	BranchID	ProductID	StockLevel
▶	AB	2	10000
	CD	3	10000
	EF	5	100000
	HH	4	20000
	MA	1	1000
*	NULL	NULL	NULL

2. Stored Programs

Paste here the contents of <yourLastname_yourFirstname>_programs.sql.

```
3. -- Student Name: Runde Jia
4. -- Student ID: 44434065
5.
6. /* USE magique; */
7. /* Task2 */
8. DELIMITER //
9.         DROP TRIGGER if exists CHECK_MEMBERSHIP_UPDATE;
10. //
11. DELIMITER //
12. CREATE TRIGGER CHECK_MEMBERSHIP_UPDATE
13.         BEFORE UPDATE ON Membership
14.     FOR EACH ROW
15. BEGIN
16.         DECLARE msg VARCHAR(255);
17.
18. -- Stop updating PK column value
19.     IF(NEW.MembershipID != OLD.MembershipID) THEN
20.         SET msg = 'Cannot update PK column value';
21.         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
22.     END IF;
23.
24. -- Check if upgrading MembershipLevel
25.         IF (NEW.MembershipID = OLD.MembershipID AND NEW.FirstName =
            OLD.FirstName AND NEW.LastName = OLD.LastName AND NEW.eMail =
            OLD.eMail
26.         AND          NEW.MembershipLevel=OLD.MembershipLevel          AND
            NEW.MemberExpDate = OLD.MemberExpDate) THEN
27.         SET msg = 'Nothing updated!';
28.         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
29.     END IF;
30.
31.     IF (NEW.MembershipLevel != OLD.MembershipLevel) THEN
32.
33. -- Check if the Membership has expired
34.     IF NEW.MemberExpDate < curdate() THEN
35.         SET msg = 'Membership expired!';
36.         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
37.     END IF;
38.
39. -- Only Silver can be upgraded to Gold
40.     IF (OLD.MembershipLevel != 'Silver' AND NEW.MembershipLevel = 'Gold')
```

```

THEN
41.          SET msg = 'Membership update failed, only Silver can upgrade to
Gold';
42.    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;
43.      END IF;
44.
45. -- Only Gold can be upgraded to Platinum
46.  IF (OLD.MembershipLevel != 'Gold' AND NEW.MembershipLevel = 'Platinum')
    THEN
47.          SET msg = 'Membership update failed, only Gold can upgrade to
Platinum';
48.    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;
49.      END IF;
50.
51.  END IF;
52. END //
53.
54. DELIMITER ;
55.
56. /* Task3 */
57. DELIMITER //
58.    DROP PROCEDURE IF EXISTS BrandNameCampaign;
59. //
60. DELIMITER //
61. CREATE PROCEDURE BrandNameCampaign(
62.          IN in_Brand varchar(255)
63.    )
64. BEGIN
65.    DECLARE v_finished INT DEFAULT 0;
66.    DECLARE c_ID INT DEFAULT null;
67.  DECLARE s_date date DEFAULT null;
68.  DECLARE e_date date DEFAULT null;
69.  DECLARE p_ID INT DEFAULT 0;
70.
71. -- Create cursor for Top 5 expensive product
72.  DECLARE campaign_rec CURSOR FOR
73.    SELECT ProductID
74.    FROM Product
75.    WHERE Product.Brand = in_Brand
76.    ORDER BY Product.Price DESC
77.    LIMIT 5;
78.    DECLARE CONTINUE HANDLER FOR NOT FOUND SET v_finished = 1;
79.

```

```

80. -- Create new campaign
81.  SELECT (COUNT(*) + 1)
82.  INTO c_ID
83.  FROM Campaign;
84.
85. -- Campaign starts in 2 weeks after the creation
86.      SELECT DATE_ADD(curdate(), INTERVAL 14 DAY)
87.  INTO s_date;
88.
89. -- The duration of the campaign is 4 weeks, 6 weeks after creation
90.  SELECT DATE_ADD(curdate(), INTERVAL 42 DAY)
91.  INTO e_date;
92.
93.  insert  into `Campaign`(`CampaignID`,`CampaignStartDate`,`CampaignEndDate`)
      values
94.  (c_ID, s_date, e_date);
95.
96. -- Insert cursor and other attributes to create discounts details for members
97.  OPEN campaign_rec;
98.  REPEAT
99.      FETCH campaign_rec INTO p_ID;
100.      IF not(v_finished = 1) THEN
101.          INSERT INTO
      `DiscountDetails`(`ProductID`,`CampaignID`,`MembershipLevel`,`Discount`) values
102.      (p_ID, c_ID, 'Silver', 10),
103.      (p_ID, c_ID, 'Gold', 20),
104.      (p_ID, c_ID, 'Platinum', 30);
105.      END IF;
106.  UNTIL v_finished
107.  END REPEAT;
108.  CLOSE campaign_rec;
109.
110.  END //

```

111. DELIMITER ;DELIMITER ;**Required Testing on the Sample Data - Task 4 Part (a)**

Paste into this section your SQL statements for the initial tests you ran (one by one) from your file <yourLastname_yourFirstname>_programs.sql, each followed by the corresponding results as screen shots.

```

-- Test update eMail
UPDATE Membership

```

```
SET eMail = 'johnsmith@gmail.com'
WHERE MembershipID = 1;
SELECT * FROM Membership;
```

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶	1	John	Smith	johnsmith@gmail.com	Gold	2021-12-31

```
-- Test upgrading silver to Gold
UPDATE Membership
SET MembershipLevel = 'Gold'
WHERE MembershipID = 5;
SELECT * FROM Membership;
```

	5	Emma	Spring	espringh@gmail.com	Silver	2021-12-31
	5	Emma	Spring	espringh@gmail.com	Gold	2021-12-31

```
-- Test upgrading expired membership
UPDATE Membership
SET MembershipLevel = 'Platinum'
WHERE MembershipID = 2;
SELECT * FROM Membership;
```

	2	Peter	Cole	petercole@gmail.com	Silver	2021-10-15
--	---	-------	------	---------------------	--------	------------

✖ 145 02:19:17 UPDATE Membership SET MembershipLevel = 'Platinum' WHERE MembershipID = 2

Error Code: 1644. Membership expired!

```
-- Test updating MembershipExpDate
UPDATE Membership
SET MemberExpDate = '2022-11-28'
WHERE MembershipID = 2;
SELECT * FROM Membership;
```

	2	Peter	Cole	petercole@gmail.com	Silver	2021-10-15
	2	Peter	Cole	petercole@gmail.com	Silver	2022-11-28

```
-- Test upgrading membershipLevel after refreshing valid MembershipExpDate
UPDATE Membership
SET MembershipLevel = 'Gold'
WHERE MembershipID = 2;
SELECT * FROM Membership;
```

	2	Peter	Cole	petercole@gmail.com	Silver	2022-11-28
	2	Peter	Cole	petercole@gmail.com	Gold	2022-11-28

```
-- Test Nothing changed
UPDATE Membership
SET MembershipLevel = 'Platinum'
```

```
WHERE MembershipID = 3;
SELECT * FROM Membership;
```

Error Code: 1644. Nothing updated!

-- Test updating PK column value

```
UPDATE Membership
SET MembershipID = '118'
WHERE MembershipID = '1';
```

Error Code: 1644. Cannot update PK column value

```
CALL BrandNameCampaign('Penfold');
```

```
SELECT * FROM Campaign;
```

```
SELECT * FROM DiscountDetails;
```

■ New campaign created start in 2 weeks from creation

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
*	NULL	NULL	NULL

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
	6	2021-11-15	2021-12-13
*	NULL	NULL	NULL

■ Top five products in the brand are in campaign(If fewer than 5, all products will be in the campaign)

	ProductID	ProductType	PackageType	YearProduced	Price	Brand
▶	1	Wine	Bottle	1990	1000	Penfold
	2	Wine	Bottle	2000	400	Penfold
	3	Wine	Bottle	2010	200	Penfold
	4	Wine	Bottle	2020	50	Penfold

	ProductID	CampaignID	MembershipLevel	Discount
▶	1	1	Gold	21
	1	1	Platinum	32
	2	1	Gold	13
	2	2	Silver	50
	3	3	Platinum	5
	1	6	Gold	20
	1	6	Platinum	30
	1	6	Silver	10
	2	6	Gold	20
	2	6	Platinum	30
	2	6	Silver	10
	3	6	Gold	20
	3	6	Platinum	30
	3	6	Silver	10
	4	6	Gold	20
	4	6	Platinum	30
	4	6	Silver	10
*	NULL	NULL	NULL	NULL

CALL BrandNameCampaign('Blonde');

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
	6	2021-11-15	2021-12-13
*	NULL	NULL	NULL

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
	6	2021-11-15	2021-12-13
	7	2021-11-15	2021-12-13
*	NULL	NULL	NULL

Products in campaign 7.

	ProductID	CampaignID	MembershipLevel	Discount
▶	1	1	Gold	21
	1	1	Platinum	32
	2	1	Gold	13
	2	2	Silver	50
	3	3	Platinum	5
	1	6	Gold	20
	1	6	Platinum	30
	1	6	Silver	10
	2	6	Gold	20
	2	6	Platinum	30
	2	6	Silver	10
	3	6	Gold	20
	3	6	Platinum	30
	3	6	Silver	10
	4	6	Gold	20
	4	6	Platinum	30
	4	6	Silver	10
*	NULL	NULL	NULL	NULL

ProductID	CampaignID	MembershipLevel	Discount
1	1	Gold	21
1	1	Platinum	32
2	1	Gold	13
2	2	Silver	50
3	3	Platinum	5
1	6	Gold	20
1	6	Platinum	30
1	6	Silver	10
2	6	Gold	20
2	6	Platinum	30
2	6	Silver	10
3	6	Gold	20
3	6	Platinum	30
3	6	Silver	10
4	6	Gold	20
4	6	Platinum	30
4	6	Silver	10
5	7	Gold	20
5	7	Platinum	30
5	7	Silver	10

112. More Extensive Testing - Task 4 Part (b)

First explain what changes you made to which tables, what tests you ran, and why. Paste from your file <yourLastname_yourFirstname>_testscript.sql, the SQL statements you used for extensive testing including the procedure calls you made for this purpose, each followed by the screenshots of the records in the affected tables (and/or the error/warning messages).

Insert more records into 'Membership' Table to check complete upgrading rules

```

insert      into      `Membership`(`MembershipID`,`FirstName`,`LastName`,`eMail`,
`MembershipLevel`,`MemberExpDate`) values
(6, 'Kelly', 'Smith', 'kellysimth@gmail.com','Silver', '2021-12-31'),
(7, 'Ben', 'Lue', 'benlue@gmail.com','Gold', '2022-1-31'),
(8, 'Steph', 'Curry', 'stephcurry@gmail.com','Platinum', '2022-1-31'),
(9, 'Lebron', 'James', 'lebronjames@gmail.com','Silver', '2020-1-31'),
(10, 'Kyrie', 'Irving', 'kyrieirving@gmail.com','Gold', '2020-12-31'),
(11, 'Joe', 'Biden', 'joebiden@gmail.com','Platinum', '2020-3-20');

```

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶	1	John	Smith	johnsmith@gmail.com	Gold	2021-12-31
	2	Peter	Cole	petercole@gmail.com	Gold	2022-11-28
	3	Sarah	Goodman	sgoodman@gmail.com	Platinum	2021-10-31
	4	Christ	Gordon	cgordon@gmail.com	Gold	2021-11-28
	5	Emma	Spring	espringh@gmail.com	Gold	2021-12-31
	6	Kelly	Smith	kellysimth@gmail.com	Silver	2021-12-31
	7	Ben	Lue	benlue@gmail.com	Gold	2022-01-31
	8	Steph	Curry	stephcurry@gmail.com	Platinum	2022-01-31
	9	Lebron	James	lebronjames@gmail.com	Silver	2020-01-31
	10	Kyrie	Irving	kyrieirving@gmail.com	Gold	2020-12-31
	11	Joe	Biden	joebiden@gmail.com	Platinum	2020-03-20
*	NULL	NULL	NULL	NULL	NULL	NULL

-- Test upgrading silver to platinum

```

UPDATE Membership
SET MembershipLevel = 'Platinum'
WHERE MembershipID = 6;
SELECT * FROM Membership;

```

213 02:54:51 UPDATE Membership SET MembershipLevel = 'Platinum' WHERE MembershipID = 6

Error Code: 1644. Membership update failed, only Gold can upgrade to Platinum

-- Test updating silver to silver

```

UPDATE Membership
SET MembershipLevel = 'Silver'
WHERE MembershipID = 6;
SELECT * FROM Membership;

```

✖ 214 02:56:38 UPDATE Membership SET MembershipLevel = 'Silver' WHERE MembershipID = 6

Error Code: 1644. Nothing updated!

-- Test updating silver to Gold

```

UPDATE Membership
SET MembershipLevel = 'Gold'
WHERE MembershipID = 6;
SELECT * FROM Membership;

```

✔ 215 02:57:29 UPDATE Membership SET MembershipLevel = 'Gold' WHERE MembershipID = 6

6	Kelly	Smith	kellysimth@gmail.com	Silver	2021-12-31
6	Kelly	Smith	kellysimth@gmail.com	Gold	2021-12-31

```
-- Test upgrade Gold to Gold
UPDATE Membership
SET MembershipLevel = 'Gold'
WHERE MembershipID = 7;
SELECT * FROM Membership;
```

✖ 217 03:04:02 UPDATE Membership SET MembershipLevel = 'Gold' WHERE MembershipID = 7
Error Code: 1644. Nothing updated!

```
-- Test upgrade Gold to Platinum
UPDATE Membership
SET MembershipLevel = 'Platinum'
WHERE MembershipID = 7;
SELECT * FROM Membership;
```

7	Ben	Lue	benlue@gmail.com	Gold	2022-01-31
7	Ben	Lue	benlue@gmail.com	Platinum	2022-01-31

```
-- Test expried Membership
UPDATE Membership
SET MembershipLevel = 'Gold'
WHERE MembershipID = 9;
SELECT * FROM Membership;
```

Error Code: 1644. Membership expired!

```
-- Test expired Membership
UPDATE Membership
SET MembershipLevel = 'Silver'
WHERE MembershipID = '10';
SELECT * FROM Membership;
```

Error Code: 1644. Membership expired!

```
-- Test updating MemberExpDate to valid date and upgrade
UPDATE Membership
SET MembershipLevel = 'Platinum', MemberExpDate = '2022-12-28'
WHERE MembershipID = '10';
SELECT * FROM Membership;
```

10	Kyrie	Irving	kyrieirving@gmail.com	Gold	2020-12-31
10	Kyrie	Irving	kyrieirving@gmail.com	Platinum	2022-12-28

```
insert into `Product`(`ProductID`,`ProductType`,`PackageType`,`YearProduced`,`Price`,
`Brand`) values
```

(6,'Wine', 'Bottle', '1977', 3000, 'Penfold'),
 (7,'Beer', 'Can', '2018', 2500, 'Penfold'),
 (8,'Wine', 'Can', '2020', 500, 'Penfold'),
 (9,'Wine', 'Bottle', '1927', 8000, 'Penfold'),
 (10,'Beer','Can', 2010, 50, 'Blonde'),
 (11,'Wine','Can', 2019, 20, 'Blonde'),
 (12,'Beer','Can', 2021, 20, 'Blonde'),
 (13,'Spirit','Can', 2010, 80, 'Blonde'),
 (14,'Beer','Can', 2020, 10, 'VB'),
 (15,'Wine','Bottle', 2000, 80, 'VB');

	ProductID	ProductType	PackageType	YearProduced	Price	Brand
▶	1	Wine	Bottle	1990	1000	Penfold
	2	Wine	Bottle	2000	400	Penfold
	3	Wine	Bottle	2010	200	Penfold
	4	Wine	Bottle	2020	50	Penfold
	5	Beer	Can	2020	10	Blonde
	6	Wine	Bottle	1977	3000	Penfold
	7	Beer	Can	2018	2500	Penfold
	8	Wine	Can	2020	500	Penfold
	9	Wine	Bottle	1927	8000	Penfold
	10	Beer	Can	2010	50	Blonde
	11	Wine	Can	2019	20	Blonde
	12	Beer	Can	2021	20	Blonde
	13	Spirit	Can	2010	80	Blonde
	14	Beer	Can	2020	10	VB
	15	Wine	Bottle	2000	80	VB
★	NULL	NULL	NULL	NULL	NULL	NULL

-- Check brand with more than 5 products

CALL BrandNameCampaign('Penfold');

1	Wine	Bottle	1990	1000	Penfold
2	Wine	Bottle	2000	400	Penfold
3	Wine	Bottle	2010	200	Penfold
4	Wine	Bottle	2020	50	Penfold
6	Wine	Bottle	1977	3000	Penfold
7	Beer	Can	2018	2500	Penfold
8	Wine	Can	2020	500	Penfold
9	Wine	Bottle	1927	8000	Penfold

	CampaignID	CampaignStartDate	CampaignEndDate
▶	1	2021-01-02	2021-01-30
	2	2021-02-02	2021-02-28
	3	2021-03-02	2021-03-30
	4	2021-08-02	2021-08-30
	5	2021-09-02	2021-09-30
	6	2021-11-15	2021-12-13
	7	2021-11-15	2021-12-13
★	NULL	NULL	NULL

CampaignID	CampaignStartDate	CampaignEndDate
1	2021-01-02	2021-01-30
2	2021-02-02	2021-02-28
3	2021-03-02	2021-03-30
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-15	2021-12-13
7	2021-11-15	2021-12-13
8	2021-11-15	2021-12-13

1	8	Gold	20
1	8	Platinum	30
1	8	Silver	10
6	8	Gold	20
6	8	Platinum	30
6	8	Silver	10
7	8	Gold	20
7	8	Platinum	30
7	8	Silver	10
8	8	Gold	20
8	8	Platinum	30
8	8	Silver	10
9	8	Gold	20
9	8	Platinum	30
9	8	Silver	10

-- Check brand with 5 products

CALL BrandNameCampaign('Blonde');

ProductID	ProductType	PackageType	YearProduced	Price	Brand
5	Beer	Can	2020	10	Blonde
10	Beer	Can	2010	50	Blonde
11	Wine	Can	2019	20	Blonde
12	Beer	Can	2021	20	Blonde
13	Spirit	Can	2010	80	Blonde

CampaignID	CampaignStartDate	CampaignEndDate
1	2021-01-02	2021-01-30
2	2021-02-02	2021-02-28
3	2021-03-02	2021-03-30
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-15	2021-12-13
7	2021-11-15	2021-12-13
8	2021-11-15	2021-12-13

CampaignID	CampaignStartDate	CampaignEndDate
1	2021-01-02	2021-01-30
2	2021-02-02	2021-02-28
3	2021-03-02	2021-03-30
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-15	2021-12-13
7	2021-11-15	2021-12-13
8	2021-11-15	2021-12-13
9	2021-11-15	2021-12-13
NULL	NULL	NULL

5	9	Gold	20
5	9	Platinum	30
5	9	Silver	10
10	9	Gold	20
10	9	Platinum	30
10	9	Silver	10
11	9	Gold	20
11	9	Platinum	30
11	9	Silver	10
12	9	Gold	20
12	9	Platinum	30
12	9	Silver	10
13	9	Gold	20
13	9	Platinum	30
13	9	Silver	10
NULL	NULL	NULL	NULL

-- Check brand less than 5 products

CALL BrandNameCampaign('VB');

14	Beer	Can	2020	10	VB
15	Wine	Bottle	2000	80	VB
NULL	NULL	NULL	NULL	NULL	NULL

CampaignID	CampaignStartDate	CampaignEndDate
1	2021-01-02	2021-01-30
2	2021-02-02	2021-02-28
3	2021-03-02	2021-03-30
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-15	2021-12-13
7	2021-11-15	2021-12-13
8	2021-11-15	2021-12-13
9	2021-11-15	2021-12-13
NULL	NULL	NULL

CampaignID	CampaignStartDate	CampaignEndDate
1	2021-01-02	2021-01-30
2	2021-02-02	2021-02-28
3	2021-03-02	2021-03-30
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-15	2021-12-13
7	2021-11-15	2021-12-13
8	2021-11-15	2021-12-13
9	2021-11-15	2021-12-13
10	2021-11-15	2021-12-13
NULL	NULL	NULL

14	10	Gold	20
14	10	Platinum	30
14	10	Silver	10
15	10	Gold	20
15	10	Platinum	30
15	10	Silver	10

113. Notes (optional)

Mention here anything worth noting, e.g., whether you faced any particular difficulty in completing any of these tasks, the nature and extent of any help you received from anyone, and why.

1. Updating other attributes for customer when they MembershipExpDate is expired. Initially thought it is not able to update Membership details when membership expired.
2. Updating information with no changes. This should give an error message.
3. Updating both MembershipExpDate and MembershipLevel. MembershipLevel is able to be upgraded once MembershipExpDate extended to the valid date
4. Nonmembership