



Database Programming and Implementation (SQL)

This project aims to get the proper skills to use SQL trigger, procedure and function.

1. The Initial State of the Database

Branch Table:

select * from branch;

Result Grid   Filter Rows:			
	BranchID	ProductID	StockLevel
▶	AB	2	10000
◀	CD	3	10000
	EF	5	100000
◀	HH	4	20000
	MA	1	1000
◀	NULL	NULL	NULL

Campaign table:

select * from 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:

select * from 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:

select * from membership;

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶	1	John	Smith	johnsimth@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

Product:

select * from 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

2. Stored Programs

-- task 2

Drop TRIGGER if exists CHECK_MEMBERSHIP_UPDATE;

DELIMITER //

CREATE TRIGGER CHECK_MEMBERSHIP_UPDATE

BEFORE UPDATE ON Membership

FOR EACH ROW

BEGIN

 DECLARE msg VARCHAR(200);

 DECLARE currentDate DATE default now();

 Declare flag boolean;

 -- when try to change membership level with the values other than 'Silver', 'Gold', 'Platinum'

 set flag = new.MembershipLevel='Platinum' or new.MembershipLevel='Silver' or
new.MembershipLevel='Gold';

 if flag = false then

 SET msg="Membership level can not have values except Platinum, Gold,
Silver.";

 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;

 end if;

 IF new.MemberExpDate < currentDate and old.MembershipLevel <>
new.MembershipLevel THEN

 -- when nothing changed except MembershipLevel

 SET msg="Membership expired! Can't upgrade membership level.";

 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;

 else

 if old.MembershipLevel='Platinum' and new.MembershipLevel <>'Platinum'
then

 SET msg="There is no further upgrade for the PLATINUM members";

 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;

 elseif new.MembershipLevel='Platinum' and old.MembershipLevel='Silver'

then

 SET msg="Only the GOLD member can be upgraded to the PLATINUM
members";

 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;

 elseif new.MembershipLevel='Silver' and old.MembershipLevel<>'Silver' then

 SET msg="Silver is the lowest membership level";

 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =msg;

 end if;

 END IF;

END //

DELIMITER ;

-- task 3

drop procedure if exists BrandNameCampaign;

delimiter //

create procedure BrandNameCampaign(in brandName varchar(255))

begin

declare v_finished int default 0;

declare campaignNo int ;

declare campaignStart date;

declare campaignFinish date;

declare v_productID int;

declare discountProduct cursor for

select productID from product where brand=brandName order by price desc

limit 5;

declare continue handler for not found set v_finished = 1;

select campaignID into campaignNo from campaign order by campaignID desc limit 1;

set campaignNo = campaignNo +1;

set campaignStart = DATE_ADD(now(), INTERVAL 2 week);

set campaignFinish = DATE_ADD(now(), INTERVAL 6 week);

insert into campaign Values (campaignNo, campaignStart, campaignFinish);

open discountProduct;

repeat

fetch discountProduct into v_productID;

if not (v_finished =1) then

insert into discountDetails Values (v_productID,
campaignNo,'Silver',10);

insert into discountDetails Values (v_productID, campaignNo,'Gold',20);

insert into discountDetails Values (v_productID,
campaignNo,'Platinum',30);

end if;

until v_finished

end repeat;

close discountProduct;

end //

delimiter ;

3. Required Testing on the Sample Data - Task 4 Part (a)

-- task4 Part(a)

-- task 2 testing

-- a.2.1 when try to upgrade the membeshipLevel for the expired member

Update Membership set MembershipLevel='Gold' where MembershipID=2;

Update Membership set MembershipLevel='Platinum' where MembershipID=2;

Action Output				
	Time	Action	Response	Duration / Fetch Time
29	13:58:09	Update Membership set MembershipLevel='Gold' where MembershipID=2	Error Code: 1644. Membership expired! Can't upgrade membership level.	0.0049 sec
30	13:58:24	Update Membership set MembershipLevel='Platinum' where MembershipID=2	Error Code: 1644. Membership expired! Can't upgrade membership level.	0.00086 sec

-- failed since membership is expired

-- a.2.2 when try to change silver to platinum

Update Membership set MembershipLevel='Platinum' where MembershipID=5;

Action Output				
	Time	Action	Response	Duration / Fetch Time
24	18:05:39	Update Membership set MembershipLevel='Platinum' where MembershipID=5	Error Code: 1644. Only the GOLD member can be upgraded to the PLATINUM members	0.0012 sec

-- failed since only the 'Gold' member can be upgraded to 'Platinum' member

-- a.2.3 when try to change silver to gold (vaild)

Update Membership set MembershipLevel='Gold' where MembershipID=5;

Action Output				
	Time	Action	Response	Duration / Fetch Time
26	18:08:11	Update Membership set MembershipLevel='Gold' where MembershipID=5	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.00086 sec

MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
1	John	Smith	johnsimth@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	Gold	2021-12-31
NULL	NULL	NULL	NULL	NULL	NULL

-- success because 'Silver' member can be upgraded to 'Gold' member

-- a.2.4 when try to change gold to silver

Update Membership set MembershipLevel='Silver' where MembershipID=1;

Action Output				
	Time	Action	Response	Duration / Fetch Time
28	18:10:58	Update Membership set MembershipLevel='Silver' where MembershipID=1	Error Code: 1644. Silver is the lowest membership level	0.00085 sec

-- failed since 'Silver' is the lowest membership level

-- a.2.5 when try to change gold to platinum (vaild)

Update Membership set MembershipLevel='Platinum' where MembershipID=1;

Action Output						
	Time	Action		Response	Duration / Fetch Time	
29	18:11:41	Update Membership set MembershipLevel='Platinum' where MembershipID=3		1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.0017 sec	
MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate	
1	John	Smith	johnsimth@gmail.com	Platinum	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	Gold	2021-12-31	
NULL	NULL	NULL	NULL	NULL	NULL	

-- success because 'Gold' member can be upgraded to 'Platinum' member

-- a.2.6 when try to change platinum to silver or gold

Update Membership set MembershipLevel='Silver' where MembershipID=3;

Update Membership set MembershipLevel='Gold' where MembershipID=3;

Action Output						
	Time	Action		Response	Duration / Fetch Time	
32	18:13:11	Update Membership set MembershipLevel='Gold' where MembershipID=3		Error Code: 1644. There is no further upgrade for the PLATINUM members	0.00088 sec	

-- failed since there are no further upgrades for 'Platinum' member

-- test 3 testing

-- a.3.1 when call the procedure with 'Blonde'

call BrandNameCampaign('Blonde');

Result Grid				Filter Rows:	Search	Edit:
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	2021-11-09	2021-12-07				
NULL	NULL	NULL				

campaign table

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

DiscountDetails table

-- a.3.2 when call the procedure with 'Penfold'
call BrandNameCampaign('Penfold');

Action Output				
	Time	Action	Response	Duration / Fetch Time
36	18:19:10	call BrandNameCampaign('Penfold')	0 row(s) affected	0.0065 sec

	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	2021-11-09	2021-12-07
	7	2021-11-09	2021-12-07
	NULL	NULL	NULL

campaign table

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

discountDetails table

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

-- task 2 testing

-- b.2.1 Insert values to check what happens when trying to change the membership level of a row having today as an expiry date

```
insert into membership values (6, 'Noah', 'Kwon', 'ahweqwheq@gmail.com', 'Gold', date(now()));
```

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶ 1		John	Smith	johnsimth@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
6		Noah	Kwon	ahweqwheq@gmail.com	Gold	2021-10-27
	NULL	NULL	NULL	NULL	NULL	NULL

-- test when today is the expiry date

-- purpose : prove that relation between member expiry date and the current time is well set up

Update Membership set MembershipLevel='Platinum' where MembershipID=6;

	MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
▶ 1		John	Smith	johnsimth@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
6		Noah	Kwon	ahweqwheq@gmail.com	Platinum	2021-10-27
	NULL	NULL	NULL	NULL	NULL	NULL

-- the test was successful since the membership is not expired yet.

-- b.2.2 Other variables like FirstName, LastName can be changed even when the expiry date is passed since there are no given restrictions for that

-- purpose: prove that there are no expiry date restrictions for updating other variables except membershipLevel

Update Membership set FirstName='Bree' where MembershipID=2;

Update Membership set eMail='test@email'where MembershipID=2;

MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
1	John	Smith	johnsimth@gmail.com	Gold	2021-12-31
2	Bree	Cole	test@email	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
6	Noah	Kwon	ahweqwheq@gmail.com	Platinum	2021-10-27
NULL	NULL	NULL	NULL	NULL	NULL

-- the test was successful since there is no expiry date restriction when updating another variable

-- b.2.3 As an extension of test b.2.2 try to change two variables includes membership level when membership is expired

-- purpose: prove that membershipLevel can not be updated even with updating other variables when membership is expired

Update Membership set FirstName='quin', MembershipLevel='Gold' where MembershipID=2;

✖ 252 16:14:52 Update Membership set FirstName='quin', MembershipLevel='Gold' where Membe... Error Code: 1644. Membership expired! Can't upgrade membership level.

-- b.2.4 more than two other variables include membershipLevel can be changed if the update is valid

-- purpose: prove that more than two other variables can be changed if the new values are valid

Update Membership set eMail='GoldTest@google.com', membershipLevel='Platinum' where MembershipID=4;

✔ 270 16:35:01 Update Membership set eMail='GoldTest@google.com', membershipLevel='Platinu... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

MembershipID	FirstName	LastName	eMail	MembershipLevel	MemberExpDate
1	John	Smith	johnsimth@gmail.com	Gold	2021-12-31
2	quin	Cole	test@email	Silver	2021-10-15
3	Sarah	Goodman	Platinum@google.com	Platinum	2021-10-31
4	Christ	Gordon	GoldTest@google.com	Platinum	2021-11-28
5	Emma	Spring	espringh@gmail.com	Silver	2021-12-31
6	Noah	Kwon	ahweqwheq@gmail.com	Platinum	2021-10-27
NULL	NULL	NULL	NULL	NULL	NULL

-- other variables can be changed without any expiry date restriction

-- b.2.5 when trying to change MembershipLevel to an invalid value which is not 'Silver', 'Gold', 'Platinum'

-- purpose: prove that membershipLevel can not be any other values than 'Silver', 'Gold', 'Platinum'

Update Membership set MembershipLevel='test' where MembershipID=5;

Update Membership set MembershipLevel='test' where MembershipID=1;

Update Membership set MembershipLevel='test' where MembershipID=3;

Action Output			
	Time	Action	Response
259	16:16:38	Update Membership set MembershipLevel='test' where MembershipID=5	Error Code: 1644. Membership level can not have values except Platinum, Gold, Silver.
260	16:16:41	Update Membership set MembershipLevel='test' where MembershipID=1	Error Code: 1644. Membership level can not have values except Platinum, Gold, Silver.
261	16:16:43	Update Membership set MembershipLevel='test' where MembershipID=3	Error Code: 1644. Membership level can not have values except Platinum, Gold, Silver.

-- Test failed since there is no membership level other than 'Silver', 'Gold', 'Platinum'.

-- b.2.6 when try to change membershipLevel into same membershipLevel

-- purpose: prove that changing membershipLevel into the same value is valid

Update Membership set MembershipLevel='Silver' where MembershipID=5;

Update Membership set MembershipLevel='Gold' where MembershipID=1;

Update Membership set MembershipLevel='Platinum' where MembershipID=3;

Update Membership set MembershipLevel='Silver' where MembershipID=2;

81	14:09:15	Update Membership set MembershipLevel='Silver' where...	0 row(s) affected Rows matched:...	0.00078 sec
82	14:09:19	Update Membership set MembershipLevel='Gold' where M...	0 row(s) affected Rows matched:...	0.00076 sec
83	14:09:22	Update Membership set MembershipLevel='Platinum' whe...	0 row(s) affected Rows matched:...	0.00085 sec
84	14:09:24	Update Membership set MembershipLevel='Silver' where...	0 row(s) affected Rows matched:...	0.00079 sec

-- the test was successful since nothing the value has not changed at all logically

-- task 3 testing

-- b.3.1 insert more products which are brand is 'Penfold'

-- purpose: insert more 'Penfold' products to check if the sorting function in the procedure is working well. before there are only 4 default 'Penfold' products so they can not be tested.

insert into product values (6, 'Wine', 'Bottle', 2021, 550, 'Penfold');

insert into product values (7, 'Wine', 'Bottle', 2020, 530, 'Penfold');

insert into product values (8, 'Wine', 'Bottle', 2018, 320, 'Penfold');

insert into product values (9, 'Wine', 'Bottle', 2019, 3800, 'Penfold');

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	2021	550	Penfold
7	Wine	Bottle	2020	530	Penfold
8	Wine	Bottle	2018	320	Penfold
9	Wine	Bottle	2019	3800	Penfold
NULL	NULL	NULL	NULL	NULL	NULL

-- 6~9 are new inserted rows

-- b.3.2 call function with 'Penfold'

-- purpose: test that this procedure can offer the sale campaign for the top 5 expensive 'Penfold' products

call BrandNameCampaign('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	2021-11-10	2021-12-08

-- the campaign is successfully added

ProductID	CampaignID	MembershipLevel	Discount
1	1	Gold	21
1	1	Platinum	32
1	6	Gold	20
1	6	Platinum	30
1	6	Silver	10
2	1	Gold	13
2	2	Silver	50
2	6	Gold	20
2	6	Platinum	30
2	6	Silver	10
3	3	Platinum	5
6	6	Gold	20
6	6	Platinum	30
6	6	Silver	10
7	6	Gold	20
7	6	Platinum	30
7	6	Silver	10
9	6	Gold	20
9	6	Platinum	30
9	6	Silver	10

-- The discountDetail is successfully added since the top 5 expensive 'Penfold' products are like as shown in the table below(9,1,6,7,2).

ProductID	ProductType	PackageType	YearProduced	Price	Brand
9	Wine	Bottle	2019	3800	Penfold
1	Wine	Bottle	1990	1000	Penfold
6	Wine	Bottle	2021	550	Penfold
7	Wine	Bottle	2020	530	Penfold
2	Wine	Bottle	2000	400	Penfold

-- b.3.3 insert 3 products which are brand is 'XXXX'

-- purpose: test that this procedure can be performed for the new brand products which were not in the default table

insert into product values (10, 'Wine', 'Bottle',2019,30,'XXXX');

insert into product values (11, 'beer', 'Bottle',2020,15,'XXXX');

insert into product values (12, 'Wine', 'can',2020,35,'XXXX');

ProductID	ProductType	PackageType	YearProduced	Price	Brand
9	Wine	Bottle	2019	3800	Penfold
10	Wine	Bottle	2019	30	XXXX
11	beer	Bottle	2020	15	XXXX
12	Wine	can	2020	35	XXXX
NULL	NULL	NULL	NULL	NULL	NULL

-- the products are successfully added

-- b.3.4 call function with 'XXXX'

-- purpose: test that this procedure can offer the sale campaign for the top 3 expensive 'XXXX' products

call BrandNameCampaign('XXXX');

CampaignID	CampaignStartDate	CampaignEndDate
4	2021-08-02	2021-08-30
5	2021-09-02	2021-09-30
6	2021-11-10	2021-12-08
7	2021-11-10	2021-12-08

-- the campaign is successfully added

ProductID	CampaignID	MembershipLevel	Discount
1	1	Gold	21
1	1	Platinum	32
1	6	Gold	20
1	6	Platinum	30
1	6	Silver	10
2	1	Gold	13
2	2	Silver	50
2	6	Gold	20
2	6	Platinum	30
2	6	Silver	10
3	3	Platinum	5
6	6	Gold	20
6	6	Platinum	30
6	6	Silver	10
7	6	Gold	20
7	6	Platinum	30
7	6	Silver	10
9	6	Gold	20
9	6	Platinum	30
9	6	Silver	10
10	7	Gold	20
10	7	Platinum	30
10	7	Silver	10
11	7	Gold	20
11	7	Platinum	30
11	7	Silver	10
12	7	Gold	20
12	7	Platinum	30
12	7	Silver	10
NULL	NULL	NULL	NULL

-- The discountDetail is successfully added

-- b.3.5 call function with 'randomtest'

-- purpose: Test campaigns when there are no products with a given brand.

call BrandNameCampaign('randomtest');

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	2021-11-10	2021-12-08
7	2021-11-10	2021-12-08
8	2021-11-10	2021-12-08
NULL	NULL	NULL

-- The campaign is successfully added

ProductID	CampaignID	MembershipLevel	Discount
1	1	Gold	21
1	1	Platinum	32
1	6	Gold	20
1	6	Platinum	30
1	6	Silver	10
2	1	Gold	13
2	2	Silver	50
2	6	Gold	20
2	6	Platinum	30
2	6	Silver	10
3	3	Platinum	5
6	6	Gold	20
6	6	Platinum	30
6	6	Silver	10
7	6	Gold	20
7	6	Platinum	30
7	6	Silver	10
9	6	Gold	20
9	6	Platinum	30
9	6	Silver	10
10	7	Gold	20
10	7	Platinum	30
10	7	Silver	10
11	7	Gold	20
11	7	Platinum	30
11	7	Silver	10
12	7	Gold	20
12	7	Platinum	30
12	7	Silver	10
NULL	NULL	NULL	NULL

-- However, because there is no product with the brand 'randomtest', the discountDetail rows are not created.