MACQUARIE
University
SYDNEY·AUSTRALIA

MACQUARIE UNIVERSITY
Faculty of Science and Engineering
Department of Computing

ISYS224/ITEC624 Database Systems 2019 (Semester 2)

Assignment 2 (Report)

Database Programming and Implementation (worth 15%)

Student Name: Justin Lam

Student Number: 45197083

Student Declaration:

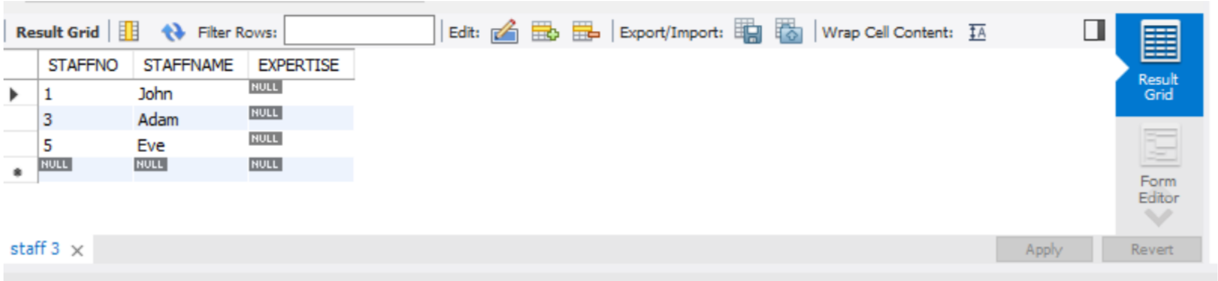
I declare that the work reported here is my own. Any help received, from any person, through discussion or other means, has been acknowledged in the last section of this report.

Student Signature:

Student Name and Date: Justin Lam (23/10/2019)

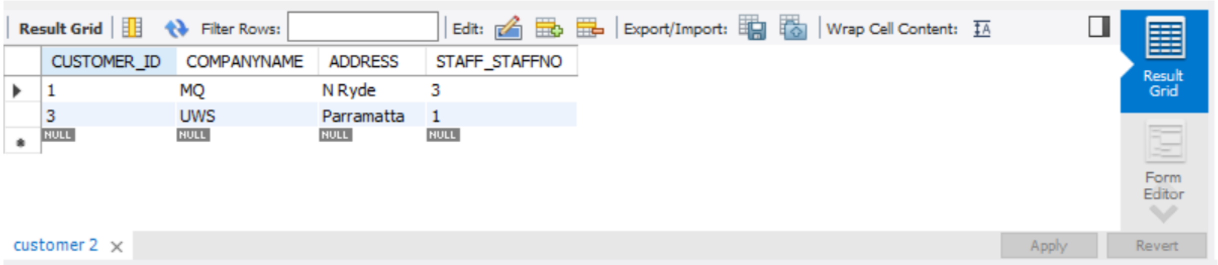
1. Initial State of the database

Staff Table



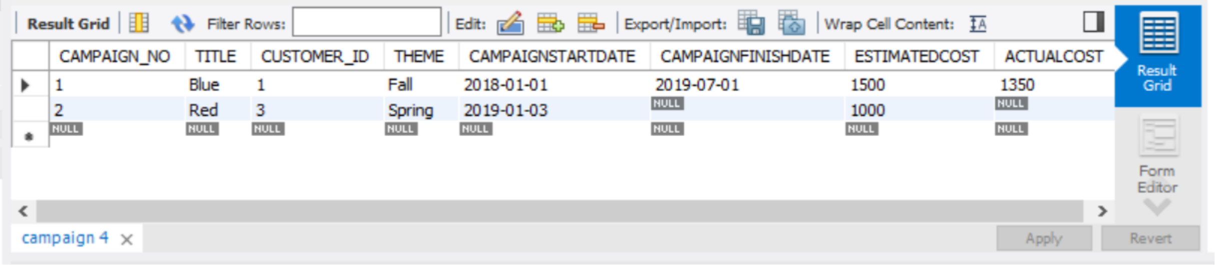
STAFFNO	STAFFNAME	EXPERTISE
1	John	NULL
3	Adam	NULL
5	Eve	NULL
NULL	NULL	NULL

Customer Table



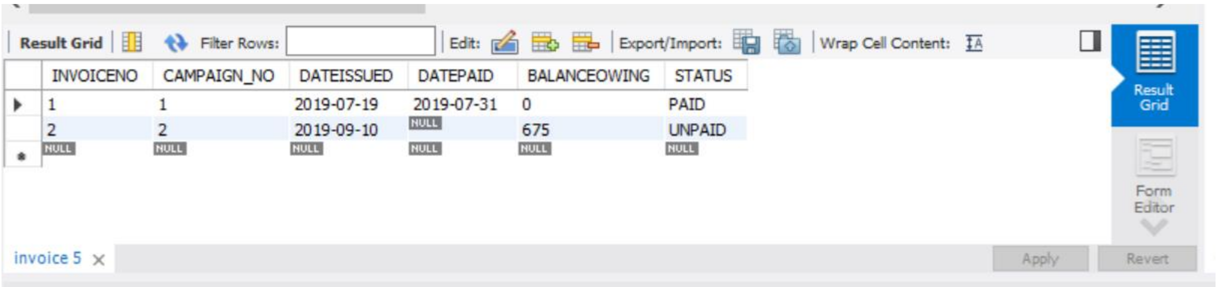
CUSTOMER_ID	COMPANYNAME	ADDRESS	STAFF_STAFFNO
1	MQ	N Ryde	3
3	UWS	Parramatta	1
NULL	NULL	NULL	NULL

Campaign Table



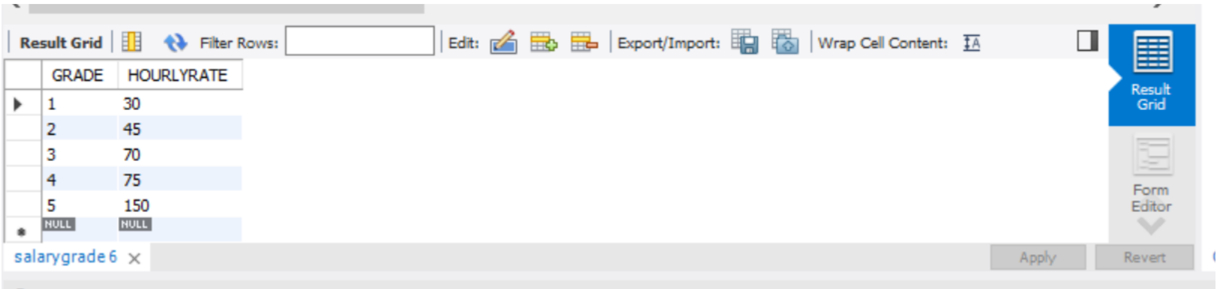
CAMPAIGN_NO	TITLE	CUSTOMER_ID	THEME	CAMPAIGNSTARTDATE	CAMPAIGNFINISHDATE	ESTIMATEDCOST	ACTUALCOST
1	Blue	1	Fall	2018-01-01	2019-07-01	1500	1350
2	Red	3	Spring	2019-01-03	NULL	1000	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Invoice Table



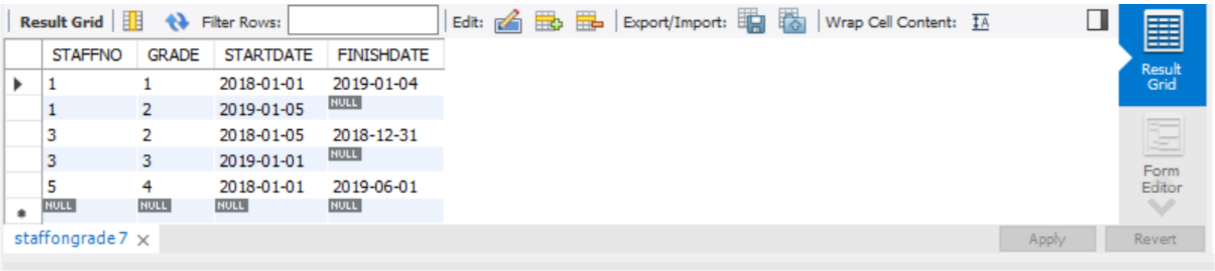
INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	PAID
2	2	2019-09-10	NULL	675	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL

SalaryGrade Table



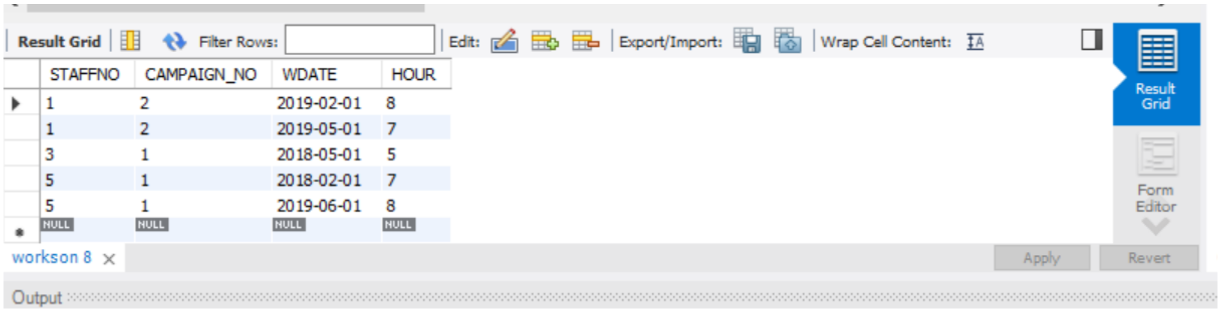
GRADE	HOURLYRATE
1	30
2	45
3	70
4	75
5	150
NULL	NULL

StaffonGrade Table



STAFFNO	GRADE	STARTDATE	FINISHDATE
1	1	2018-01-01	2019-01-04
1	2	2019-01-05	NULL
3	2	2018-01-05	2018-12-31
3	3	2019-01-01	NULL
5	4	2018-01-01	2019-06-01
NULL	NULL	NULL	NULL

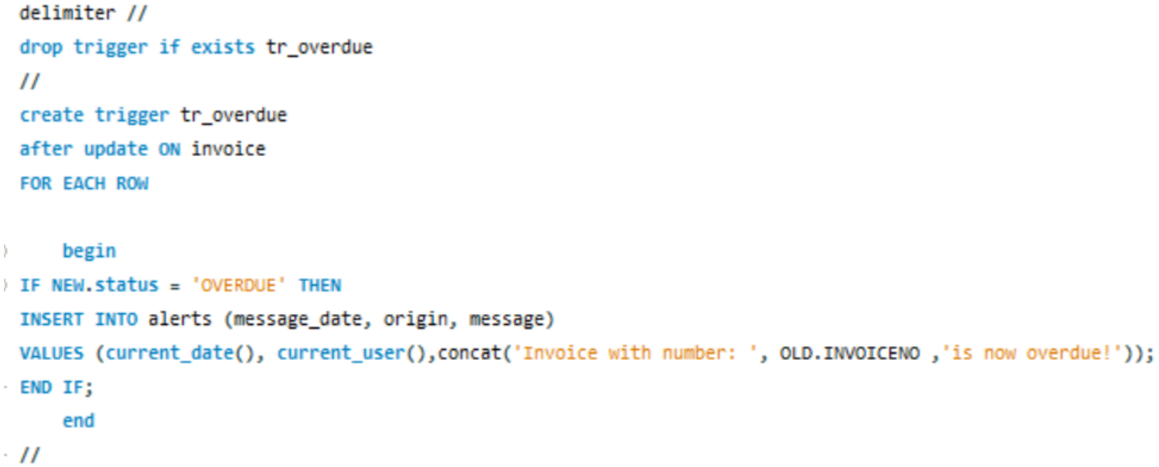
Works on Table



STAFFNO	CAMPAIGN_NO	WDATE	HOUR
1	2	2019-02-01	8
1	2	2019-05-01	7
3	1	2018-05-01	5
5	1	2018-02-01	7
5	1	2019-06-01	8
NULL	NULL	NULL	NULL

2. Stored Programs.

Trigger 1 (tr_overdue) code



```
delimiter //
drop trigger if exists tr_overdue
//
create trigger tr_overdue
after update ON invoice
FOR EACH ROW

    begin
IF NEW.status = 'OVERDUE' THEN
INSERT INTO alerts (message_date, origin, message)
VALUES (current_date(), current_user(),concat('Invoice with number: ', OLD.INVOICENO ,'is now overdue!'));
END IF;
    end
//
```

Function 1 (rate_on_date) Code

```
DELIMITER //
 drop function if exists rate_on_date //

create function rate_on_date(staff_id int, given_date DATE)
returns float
deterministic

BEGIN
            DECLARE hourly_rate float;

     DECLARE r_hour cursor for
            SELECT HOURLYRATE
            FROM staffongrade, salaryongrade
            WHERE STAFFONGRADE.GRADE = SALARYONGRADE.GRADE
            AND WORKSON.STAFFNO = staff_id
            AND given_date >= STARTDATE and (FINISHDATE is NULL or given_date
<= FINISHDATE);

open r_hour;
fetch r_hour into hourly_rate;
close r_hour;
RETURN hourly_rate;
    END //
```

Function 2 (cost_of_campaign) Code

```
drop function if exists cost_of_campaign //
create function cost_of_campaign (camp_id int)
returns float
DETERMINISTIC

begin
declare staff_id int;
declare work_date date;
declare hours float;

declare t_cost float;
declare hand int default 0;


declare c_cursor cursor for
select staffno, WDATE, 'Hours'
from workson
where CAMPAIGN_NO = camp_id;

declare continue handler for not found set hand = 1;

open c_cursor;
set t_cost = 0;
while hand != 1 DO
fetch c_cursor into staff_id, work_date, hours;
set t_cost = t_cost + (hours * rate_on_date (staff_id, work_date));
end while;
close c_cursor;
RETURN t_cost;
        END //
```

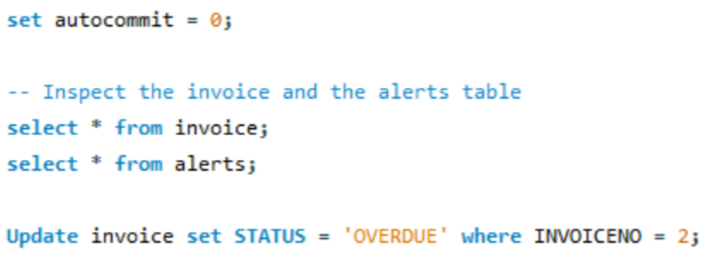
Stored Procedure 1 (sp_finish_campaign) Code

```
Delimiter //
drop procedure if exists sp_finish_campaign //


create procedure sp_finish_campaign (in c_title varchar(30))

begin
        declare number_of_campaigns int;

   select COUNT(campaign.CAMPAIGN_NO) into number_of_campaigns from campaign
WHERE campaign.TITLE = c_title;

        if number_of_campaigns = 0 then
                select 'ERROR! Campaign title does not exist' as 'msg';

   elseif number_of_campaigns = 1 then
        update campaign
        set
        campaign.CAMPAIGNFINISHDATE = CURRENT_DATE(),
        campaign.ACTUALCOST = cost_of_campaign(campaign.CAMPAIGN_NO)
   where campaign.TITLE = c_title;



        end if;

        end //
        Delimiter ;
```

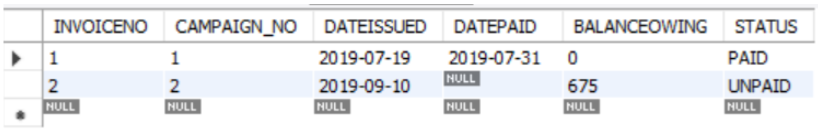
Stored Procedure 2 (sync_invoice) Code

```
Delimiter //
DROP PROCEDURE IF EXISTS sync_invoice //
--
CREATE PROCEDURE sync_invoice()
        BEGIN
                UPDATE invoice
                SET invoice.STATUS = 'overdue'
                WHERE invoice.DATEISSUED + 30 <= CURRENT_DATE()
                AND invoice.DATEPAID IS NULL;

                SELECT ROW_COUNT() AS 'Number of invoices updated.';
        END //
DELIMITER ;
```

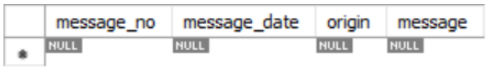
3. Required Testing against Sample Database.



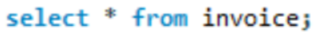
```
set autocommit = 0;

-- Inspect the invoice and the alerts table
select * from invoice;
select * from alerts;

Update invoice set STATUS = 'OVERDUE' where INVOICENO = 2;
```



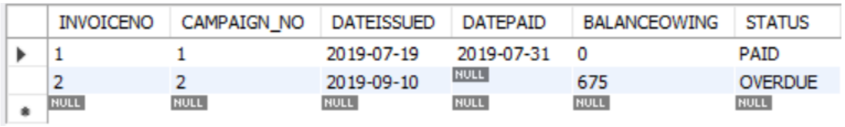
	INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
▶	1	1	2019-07-19	2019-07-31	0	PAID
	2	2	2019-09-10	NULL	675	UNPAID
*	NULL	NULL	NULL	NULL	NULL	NULL



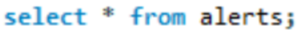
	message_no	message_date	origin	message
*	NULL	NULL	NULL	NULL



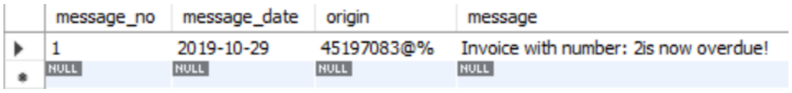
```
select * from invoice;
```



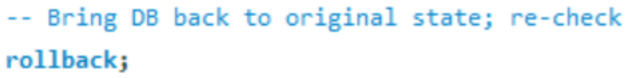
	INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
▶	1	1	2019-07-19	2019-07-31	0	PAID
	2	2	2019-09-10	NULL	675	OVERDUE
*	NULL	NULL	NULL	NULL	NULL	NULL



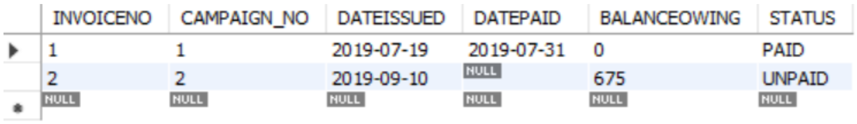
```
select * from alerts;
```



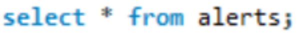
	message_no	message_date	origin	message
▶	1	2019-10-29	45197083@%	Invoice with number: 2is now overdue!
*	NULL	NULL	NULL	NULL



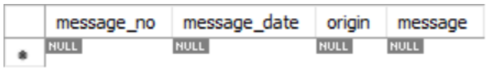
```
-- Bring DB back to original state; re-check
rollback;
```



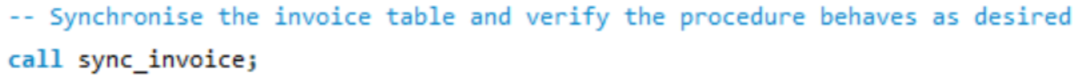
	INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
▶	1	1	2019-07-19	2019-07-31	0	PAID
	2	2	2019-09-10	NULL	675	UNPAID
*	NULL	NULL	NULL	NULL	NULL	NULL



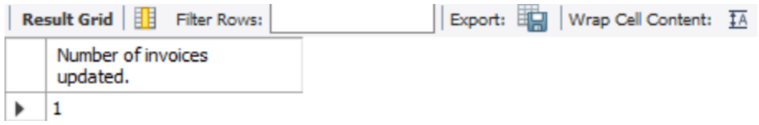
```
select * from alerts;
```



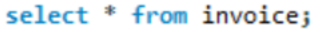
	message_no	message_date	origin	message
*	NULL	NULL	NULL	NULL



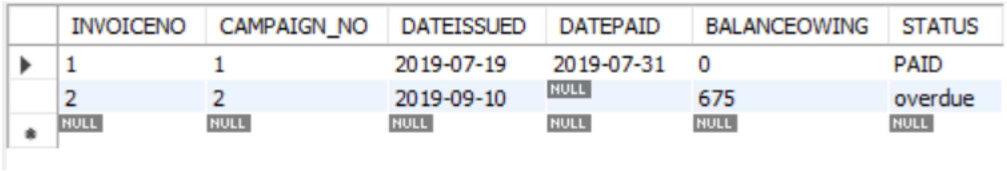
```
-- Synchronise the invoice table and verify the procedure behaves as desired
call sync_invoice;
```



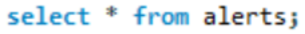
Number of invoices updated.
1



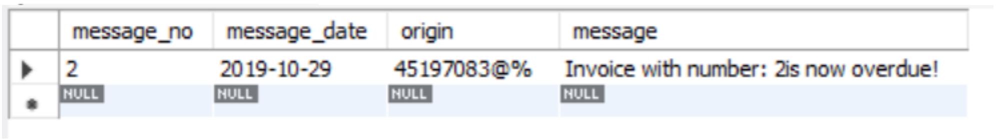
```
select * from invoice;
```



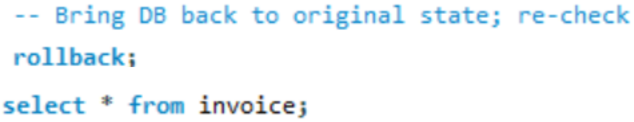
INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	PAID
2	2	2019-09-10	NULL	675	overdue
NULL	NULL	NULL	NULL	NULL	NULL



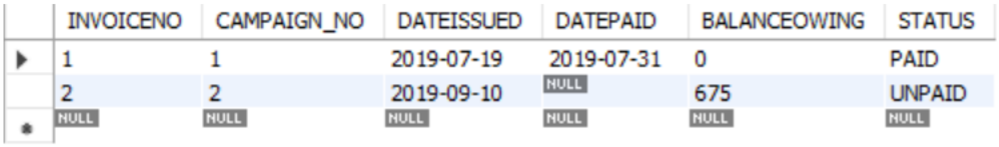
```
select * from alerts;
```



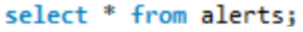
message_no	message_date	origin	message
2	2019-10-29	45197083@%	Invoice with number: 2is now overdue!
NULL	NULL	NULL	NULL



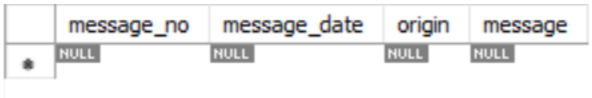
```
-- Bring DB back to original state; re-check
rollback;
select * from invoice;
```



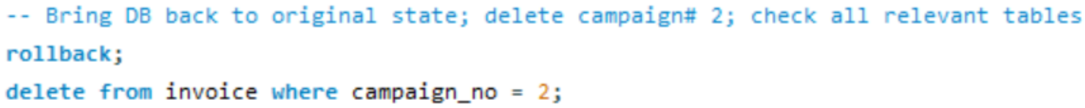
INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	PAID
2	2	2019-09-10	NULL	675	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL



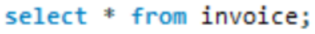
```
select * from alerts;
```



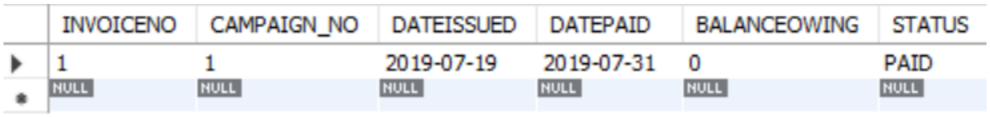
message_no	message_date	origin	message
NULL	NULL	NULL	NULL



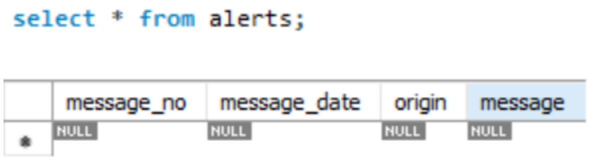
```
-- Bring DB back to original state; delete campaign# 2; check all relevant tables
rollback;
delete from invoice where campaign_no = 2;
```



```
select * from invoice;
```

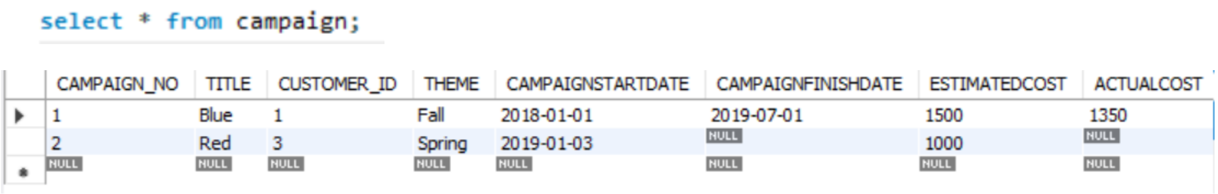


INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	PAID
NULL	NULL	NULL	NULL	NULL	NULL



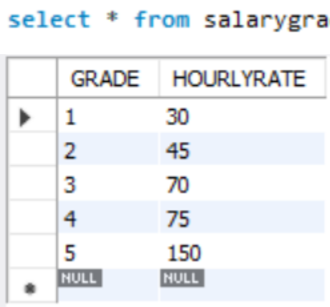
```
select * from alerts;
```

	message_no	message_date	origin	message
*	NULL	NULL	NULL	NULL



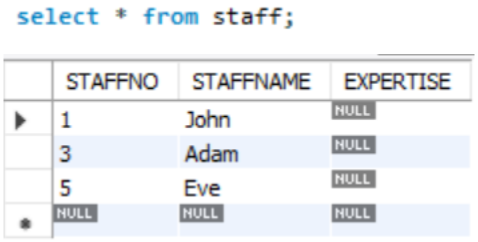
```
select * from campaign;
```

	CAMPAIGN_NO	TITLE	CUSTOMER_ID	THEME	CAMPAIGNSTARTDATE	CAMPAIGNFINISHDATE	ESTIMATEDCOST	ACTUALCOST
▶	1	Blue	1	Fall	2018-01-01	2019-07-01	1500	1350
	2	Red	3	Spring	2019-01-03	NULL	1000	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



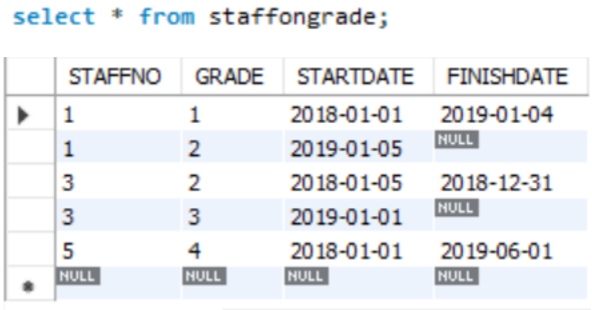
```
select * from salarygrade;
```

	GRADE	HOURLYRATE
▶	1	30
	2	45
	3	70
	4	75
	5	150
*	NULL	NULL



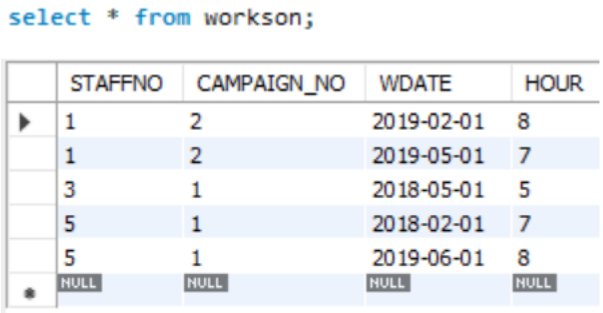
```
select * from staff;
```

	STAFFNO	STAFFNAME	EXPERTISE
▶	1	John	NULL
	3	Adam	NULL
	5	Eve	NULL
*	NULL	NULL	NULL



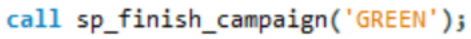
```
select * from staffongrade;
```

	STAFFNO	GRADE	STARTDATE	FINISHDATE
▶	1	1	2018-01-01	2019-01-04
	1	2	2019-01-05	NULL
	3	2	2018-01-05	2018-12-31
	3	3	2019-01-01	NULL
	5	4	2018-01-01	2019-06-01
*	NULL	NULL	NULL	NULL

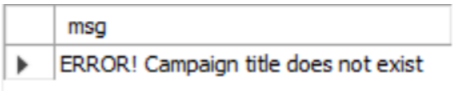


```
select * from workson;
```

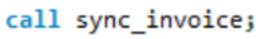
	STAFFNO	CAMPAIGN_NO	WDATE	HOUR
▶	1	2	2019-02-01	8
	1	2	2019-05-01	7
	3	1	2018-05-01	5
	5	1	2018-02-01	7
	5	1	2019-06-01	8
*	NULL	NULL	NULL	NULL



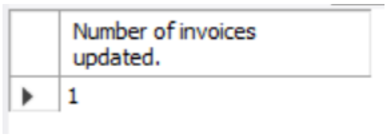
```
call sp_finish_campaign('GREEN');
```



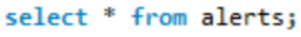
msg
ERROR! Campaign title does not exist



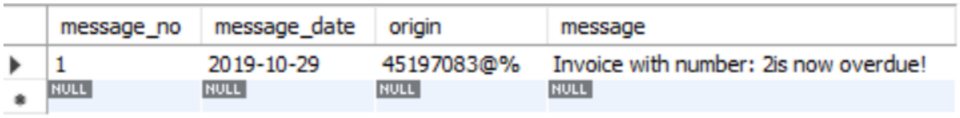
```
call sync_invoice;
```



Number of invoices updated.
1



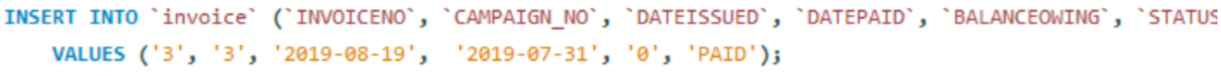
```
select * from alerts;
```



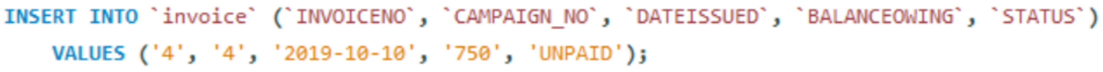
message_no	message_date	origin	message
1	2019-10-29	45197083@%	Invoice with number: 2is now overdue!
NULL	NULL	NULL	NULL

4. More Extensive Testing.

Explain what changes you made to which tables, what tests you ran, and why. Copy and paste from your file `<yourLastname_yourFirstname>_test_script.sql` *the DML statements you used for this purpose, followed by the screenshots of the records in those tables. Then copy and paste the procedure calls you made, and the screenshots of the records in the relevant tables (or the error messages).*



```
INSERT INTO `invoice` (`INVOICENO`, `CAMPAIGN_NO`, `DATEISSUED`, `DATEPAID`, `BALANCEOWING`, `STATUS
    VALUES ('3', '3', '2019-08-19',  '2019-07-31', '0', 'PAID');
```



```
INSERT INTO `invoice` (`INVOICENO`, `CAMPAIGN_NO`, `DATEISSUED`, `BALANCEOWING`, `STATUS`)
    VALUES ('4', '4', '2019-10-10', '750', 'UNPAID');
```

More data are more inserted to test out whether the trigger is working or not.

Acknowledgement

I would like to acknowledge the following people for their help through discussions.

- Matt De Masi
- Cheryl Tran
- Renáta Szabolcsik
- Isobel Ford