

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%)**

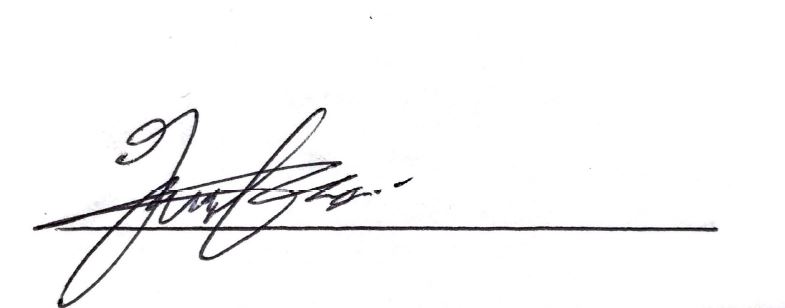
**Student Name: Justin Lam**

**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.*

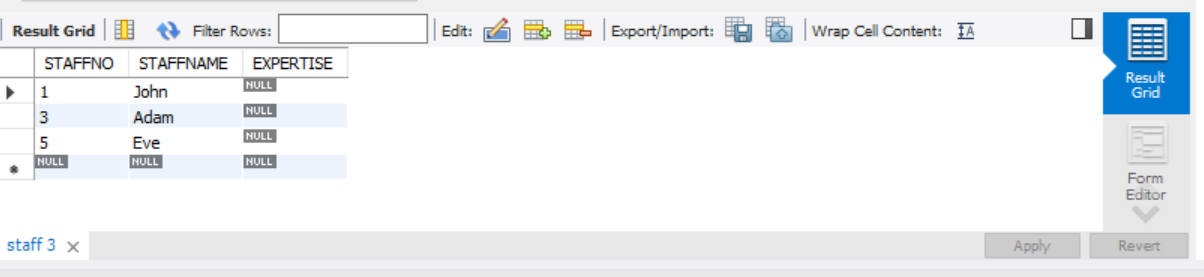
Student Signature:



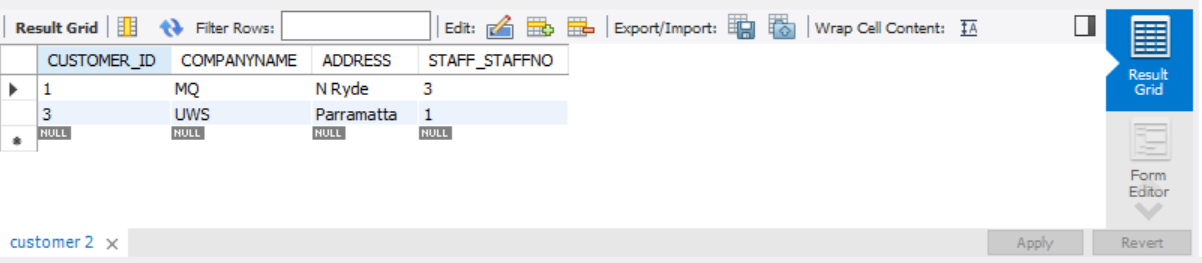
Student Name and Date: Justin Lam (23/10/2019)

1. **Initial State of the database**

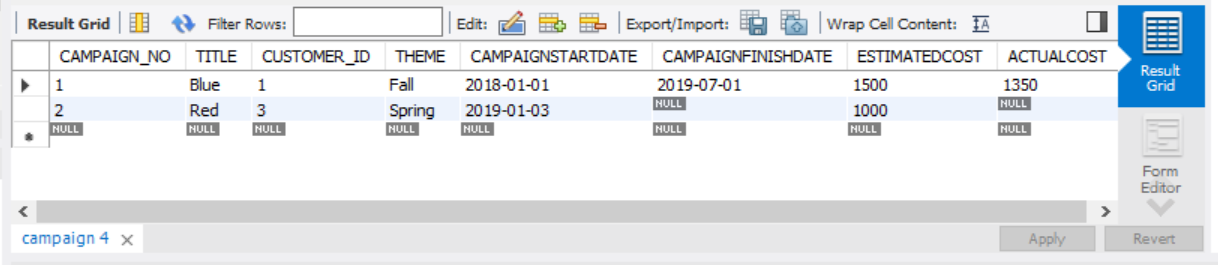
Staff Table



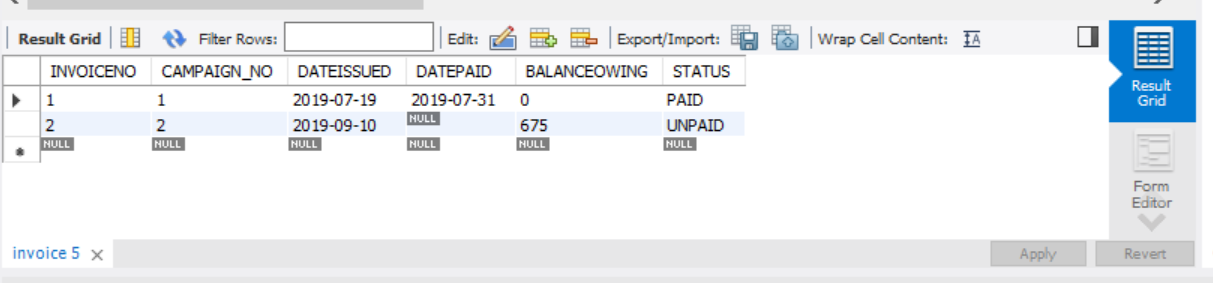
Customer Table



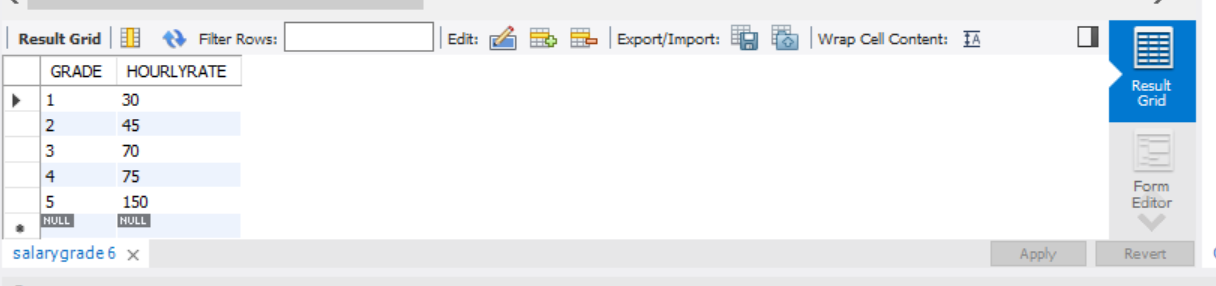
Campaign Table



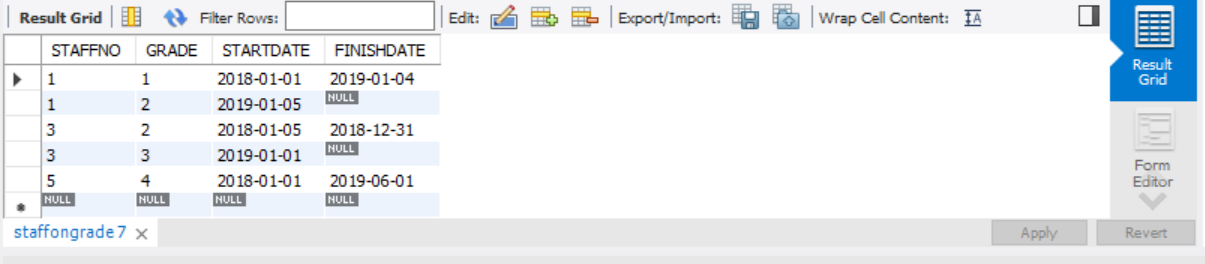
Invoice Table



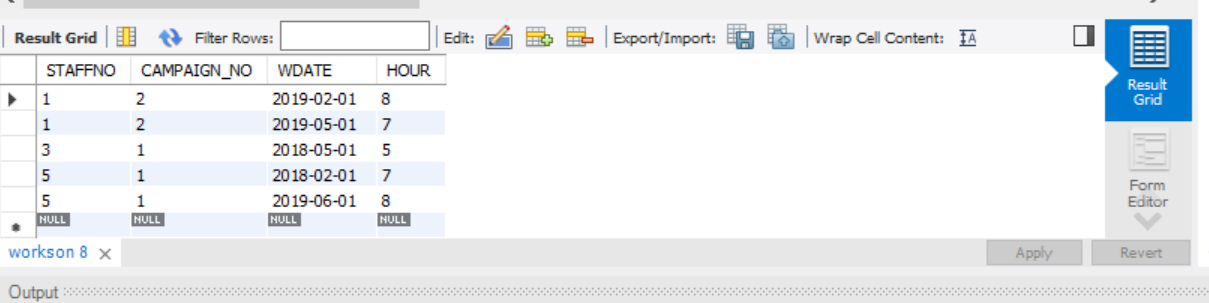
SalaryGrade Table



StaffonGrade Table

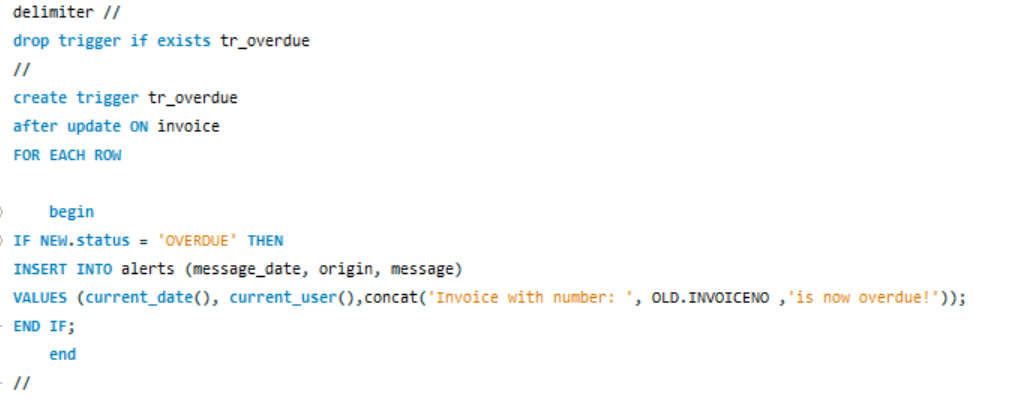


Works on Table



1. **Stored Programs.**

**Trigger 1 (tr\_overdue) code**

**

**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()

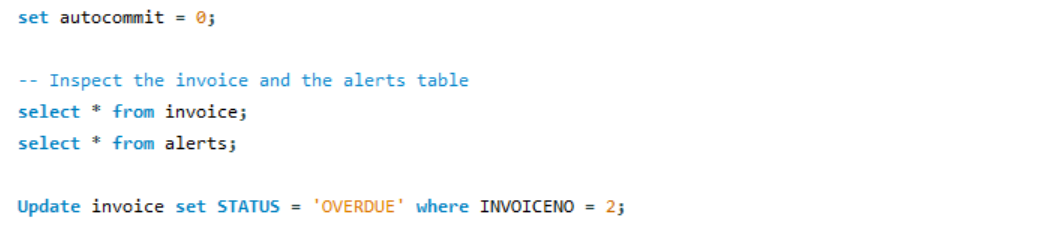
AND invoice.DATEPAID IS NULL;

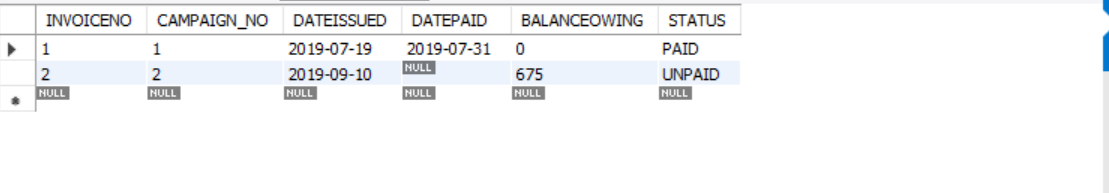
SELECT ROW\_COUNT() AS 'Number of invoices updated.';

END //

DELIMITER ;

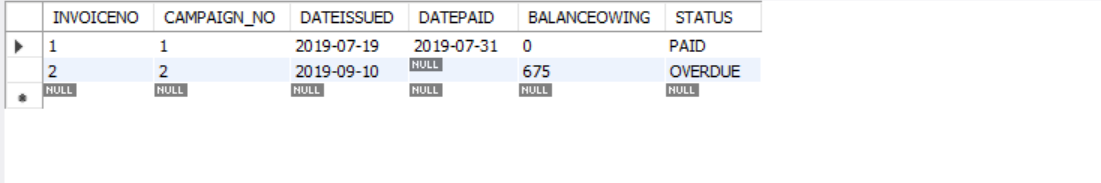
1. **Required Testing against Sample Database.**



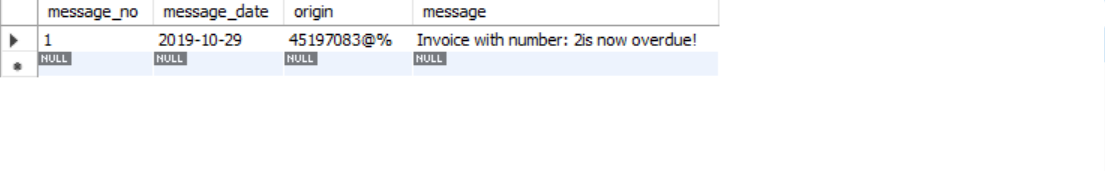


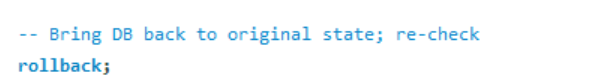
****

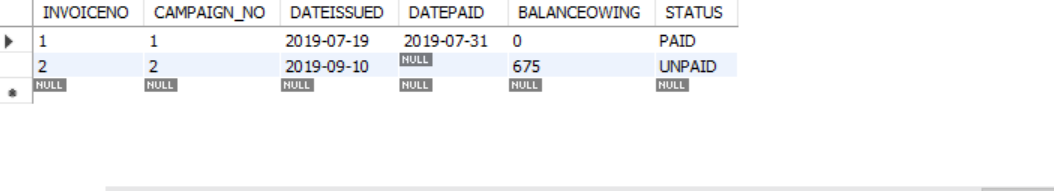




****

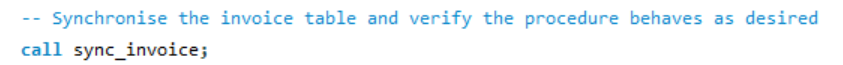
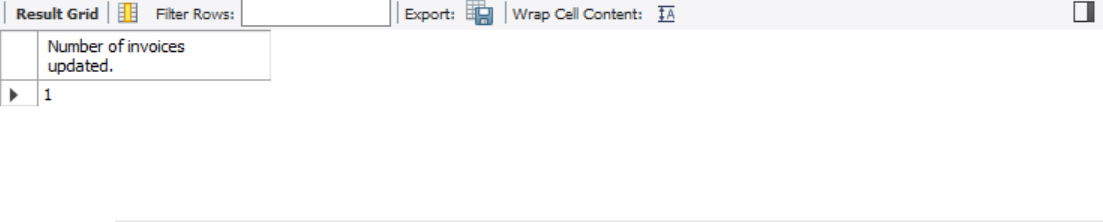
****



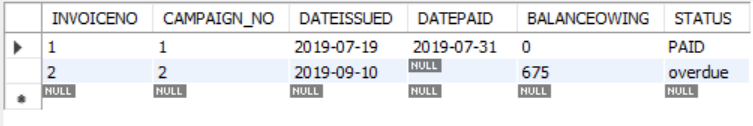


****

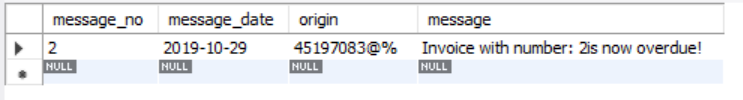
****

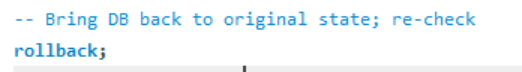
****

****

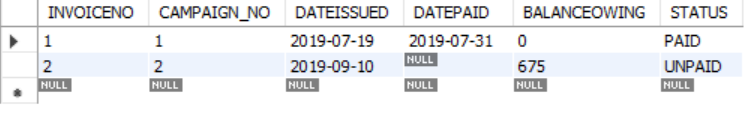
****

****

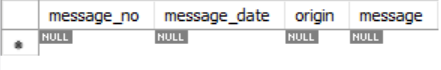
****

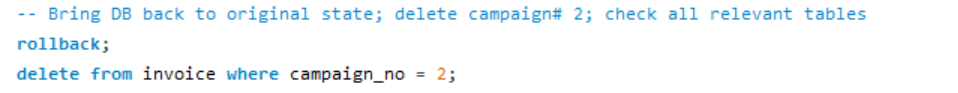
****

****

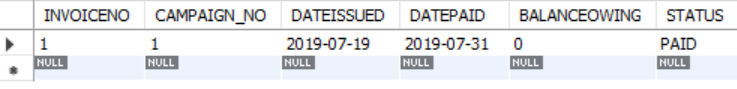
****

****

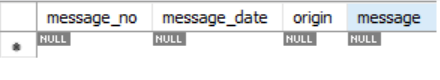
****

****

****

****

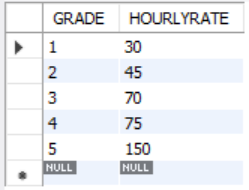




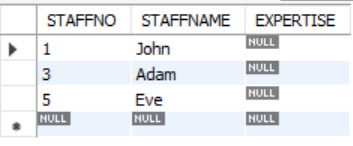
****

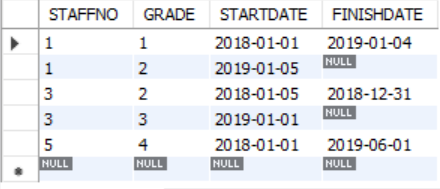
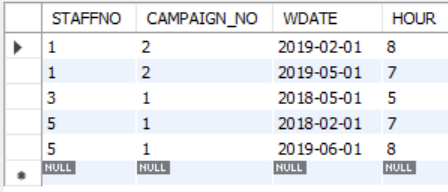
****

****

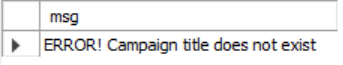
****

****

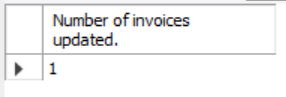
****

****

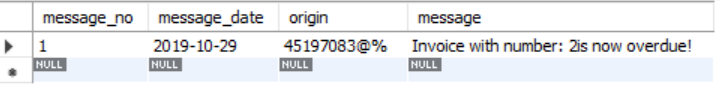
****

****

****

****

****



1. **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).*



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