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: Matthew De Masi

Student Number: 45585342

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: Matthew De Masi 23/10/2019

1. Initial State of the database

Salary Table

```
151  Select * from salarygrade;
152
```

100% 27:151

Result Grid Filter Rows: Search

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

Staffongrade Table

```
151  Select * from staffongrade;
```

100% 28:151

Result Grid Filter Rows: Search

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

Salary Table

```
151  Select * from invoice;
152
```

100% 23:151

Result Grid Filter Rows: Search Edit: Export/I

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

Workson Table

```
151  Select * from workson;
```

100% 23:151

Result Grid Filter Rows: Search

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

Staff Table

```
151  Select * from staff;
```

100% 21:151

Result Grid Filter Rows: Search

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

Salary Table

```
150
151   Select * from customer;
152
```

100% 24:151

Result Grid Filter Rows: Search Edit:

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

Campaign Table

```
150
151   Select * from campaign;
```

100% 24:151

Result Grid Filter Rows: Search Edit: Export/Import:

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

2. Stored Programs.

Trigger Overdue

```
delimiter //
drop trigger if exists tr_overdue
//

create trigger tr_overdue
-- type of trigger, etc
AFTER UPDATE ON invoice
FOR EACH ROW
    begin
-- implementation goes here
 IF NEW.status = 'OVERDUE' THEN
 INSERT INTO
 alerts (message_no,message_date,orign,message)
values  (current_date(),  current_user(),CONCAT('Invoice  with
number: ', OLD.INVOICENO ,'is now overdue!'));
 END IF;
    end
//
```

Rate on date procedure

```
drop function if exists rate_on_date //

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

DETERMINISTIC
begin
declare hour_rate float;
declare start_date date;
declare finish_date date;
```

```
declare hourly_rate int default 0;
declare v_finished int default 0;

declare hourly_work CURSOR FOR
select hourlyrate, startdate, finishdate
from salarygrade join staffongrade on staffongrade.grade =
salarygrade.grade
where staffongrade.staffno = staff_id;
declare continue handler for not found set v_finished = 1;

open hourly_work;
repeat fetch hourly_work into hourly_rate, start_date,
finish_date;

if(given_date between start_date and finish_date)
then set hour_rate = hourly_rate;

elseif given_date >= start_date and finish_date is null
then set hour_rate = hourly_rate;
end if;
until v_finished
end repeat;
close hourly_work;

return hour_rate;

end //
```

Cost of Campaign procedure

```
drop function if exists cost_of_campaign //

create function cost_of_campaign (camp_id int) returns float

DETERMINISTIC
begin
declare total_cost float;
select sum(rate_on_date(STAFFNO,WDATE)*hour)
into total_cost from workson
where camp_id = CAMPAIGN_NO;
return total_cost;
end //
```

Sp Finish Campaign procedure

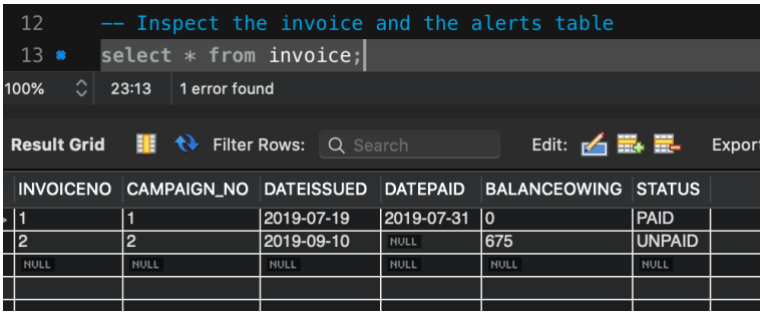
```
drop procedure if exists sp_finish_campaign //
create procedure sp_finish_campaign (in c_title varchar(30))

begin
declare B varchar(30);

select count(c_title) into B
from campaign where c_title = TITLE;

if (B != 1) then signal sqlstate '45000'

set message_text = 'ERROR! Campaign title does not exist!';
end if;

update campaign
set ACTUALCOST = cost_of_campaign (CAMPAIGN_NO)
where c_title = TITLE;
update campaign
set CAMPAIGNFINISHDATE = current_date() where c_title = TITLE;

end //
Delimiter ;
```

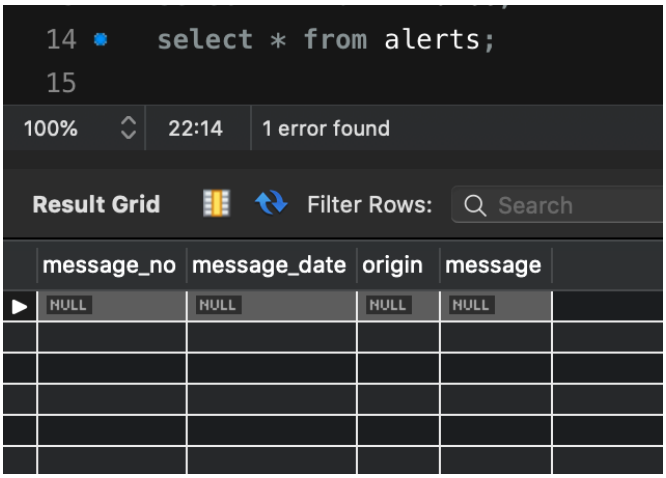
Sync Invoice Table

```
drop procedure if exists sync_invoice //

CREATE procedure sync_invoice()
begin
declare dDate date;
declare stat varchar(20);
declare d_finished int default 0;
declare  d_array  cursor  for  select  DATEISSUED,  STATUS  from
invoice;
declare continue handler for not found set d_finished = 1;
open d_array;

repeat
fetch d_array into dDate, stat;
if (datediff(current_date(), dDate)> 30 )then
```

```
Update invoice Set STATUS = 'OVERDUE' Where datediff(current_date(), DATEISSUED) > 30 And STATUS = 'UNPAID';
end if;

until d_finished
end repeat;

close d_array;
end
//
delimiter ;
```

3. Required Testing against Sample Database.





```
-- Update the invoice table
UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 2;

-- Verify that the trigger you implemented works
```

100% 59:13

Action Output

	Time	Action
1	17:36:40	UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 2

```
-- Update the invoice table
UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 2;

-- Verify that the trigger you implemented works
select * from invoice;
select * from alerts;

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

100% 49:15

Result Grid Filter Rows: Search Edit: Export/Impo

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

```
-- Verify that the trigger you implemented works
select * from invoice;
select * from alerts;
```

00% 22:21

Result Grid Filter Rows: Search Edit:

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

```
-- Synchronise the invoice table and verify the proce
call sync_invoice;
select * from invoice;
select * from alerts;
```

100% 23:22

Result Grid Filter Rows: Search Edit: Exp

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

```
-- Synchronise the invoice ta
call sync_invoice;
select * from invoice;
select * from alerts;
```

100% 22:27

Result Grid Filter Rows: Search

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

```
call sp_finish_campaign('RED');
select * from campaign;
select * from invoice;
```

100% 32:45

Action Output

	Time	Action
1	02:13:23	call sp_finish_campaign('RED')

```
-- Finish the campaign titled RED. Verify that it be
call sp_finish_campaign('RED');
select * from campaign;
select * from invoice;
call sp_finish_campaign('GREEN'); -- should SIGNAL e
```

100% 23:47

Result Grid Filter Rows: Search Edit: E

INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
2	2	2019-09-10	NULL	675	UNPAID

```
-- Finish the campaign titled RED. Verify that it behaves as desired.
call sp_finish_campaign('RED');
select * from campaign;
select * from invoice;
call sp_finish_campaign('GREEN'); -- should SIGNAL error condition
```

100% 24:46

Result Grid Filter Rows: Search Edit: Export/Import:

CAMPAIGN_NO	TITLE	CUSTOMER_ID	THEME	CAMPAIGNSTARTDATE	CAMPAIGNFINISHDATE	ESTIMATEDCOST	ACTUALCOST
2	Red	3	Spring	2019-01-03	2019-10-30	1000	675
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
call sp_finish_campaign('GREEN'); -- should SIGNAL error condition
```

100% 68:48

Action Output

	Time	Action	Response
134	02:07:51	delete from invoice wh...	1 row(s) affected
135	02:07:54	select * from invoice LI...	1 row(s) returned
136	02:07:57	select * from alerts LIM...	0 row(s) returned
137	02:07:58	select * from campaign...	2 row(s) returned
138	02:07:59	select * from salarygra...	5 row(s) returned
139	02:08:02	select * from staff LIMI...	3 row(s) returned
140	02:08:04	select * from staffongr...	5 row(s) returned
141	02:08:06	select * from workson L...	5 row(s) returned
142	02:08:12	call sp_finish_campaign...	Error Code: 1175. You are using safe update mode and you trie
143	02:08:28	call sp_finish_campaign...	Error Code: 1175. You are using safe update mode and you trie
144	02:08:35	select * from campaign...	2 row(s) returned
145	02:09:41	select * from invoice LI...	1 row(s) returned
146	02:09:58	call sp_finish_campaign...	Error Code: 1644. ERROR! Campaign title does not exist!

```
-- Synchronise the invoice table and verify the procedure behaves as desired
call sync_invoice;
select * from alerts;
rollback;
```

100% 22:49

Result Grid Filter Rows: Search Edit: Export/Import:

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

4. More Extensive Testing.

Additional Records applied

```
INSERT INTO `customer` (`CUSTOMER_ID`, `COMPANYNAME`, `ADDRESS`,  `STAFF_STAFFNO`) VALUES ('2', 'ANU', 'Eastwood',  '3');
INSERT INTO `customer` (`CUSTOMER_ID`, `COMPANYNAME`, `ADDRESS`,  `STAFF_STAFFNO`) VALUES ('4', 'UNSW', 'Sydney',  '5');
```

```
INSERT INTO `campaign` (`CAMPAIGN_NO`, `TITLE`, `CUSTOMER_ID`, `THEME`, `CAMPAIGNSTARTDATE`, `CAMPAIGNFINISHDATE`, `ESTIMATEDCOST`, `ACTUALCOST`)
    VALUES ('3', 'Green', '3', 'Summer', '2018-07-01', '2019-08-10', '1600', '1650');

INSERT INTO `campaign` (`CAMPAIGN_NO`, `TITLE`, `CUSTOMER_ID`, `THEME`, `CAMPAIGNSTARTDATE`, `ESTIMATEDCOST`)
    VALUES ('4', 'Black', '4', 'Winter', '2019-03-10', '2000');
```

```
INSERT INTO `invoice` (`INVOICENO`, `CAMPAIGN_NO`, `DATEISSUED`, `DATEPAID`, `BALANCEOWING`, `STATUS`)
    VALUES ('3', '3', '2019-09-19',  '2019-09-30', '980', 'UNPAID');

INSERT INTO `invoice` (`INVOICENO`, `CAMPAIGN_NO`, `DATEISSUED`, `BALANCEOWING`, `STATUS`)
    VALUES ('4', '4', '2019-10-10', '675', 'UNPAID');
```

Testing the additional records

```
-- Update the invoice table
UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 1;

-- Verify that the trigger you implemented works
select * from invoice;
```

100% 59:13

Result Grid Filter Rows: Search Edit: Export/Import:

message_no	message_date	origin	message
alerts 5			

Action Output

	Time	Action	Response
1	10:29:16	UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

```
-- Update the invoice table
UPDATE invoice set STATUS = 'OVERDUE' where INVOICENO = 1;

-- Verify that the trigger you implemented works
```

100% 23:16

Result Grid Filter Rows: Search Edit: Export/Import:

INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	OVERDUE
2	2	2019-09-10	NULL	675	UNPAID
3	3	2019-09-19	2019-09-30	980	UNPAID
4	4	2019-10-10	NULL	675	UNPAID

alerts 5 invoice 22

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

-- Update the invoice table
UPDATE invoice set STATUS = 'O

-- Verify that the trigger you
```

100% 22:10

Result Grid Filter Rows: Search

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

```
-- Synchronise the invoice table and verify the proced
call sync_invoice;
select * from invoice;
select * from alerts;

-- Bring DB back to original state; delete campaign# 2
```

100% 23:27

Result Grid Filter Rows: Search Edit: Expo

INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	OVERDUE
2	2	2019-09-10	NULL	675	OVERDUE
3	3	2019-09-19	2019-09-30	980	OVERDUE
4	4	2019-10-10	NULL	675	UNPAID

```
-- Synchronise the invoice table
call sync_invoice;
select * from invoice;
select * from alerts;

-- Bring DB back to original sta
rollback;
delete from invoice where campai
select * from invoice;
```

100% 22:28

Result Grid Filter Rows: Search

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

```
-- Bring DB back to original state; delete campaign# 2;
rollback;
delete from invoice where campaign_no = 2;
select * from invoice;
select * from alerts;
select * from campaign;
select * from salarygrade;
select * from staff;
select * from staffongrade;
```

100% 23:33

Result Grid Filter Rows: Search Edit: Export/

INVOICENO	CAMPAIGN_NO	DATEISSUED	DATEPAID	BALANCEOWING	STATUS
1	1	2019-07-19	2019-07-31	0	PAID
3	3	2019-09-19	2019-09-30	980	UNPAID
4	4	2019-10-10	NULL	675	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL

```
-- Finish the campaign titled RED. Verify that it behaves as desired.
call sp_finish_campaign('BLUE');
select * from campaign;
select * from invoice;
call sp_finish_campaign('GREEN'); -- should SIGNAL error condition
```

100% 24:43

Result Grid Filter Rows: Search Edit: Export/Import:

CAMPAIGN_NO	TITLE	CUSTOMER_ID	THEME	CAMPAIGNSTARTDATE	CAMPAIGNFINISHDATE	ESTIMATEDCOST	ACTUALCOST
1	Blue	1	Fall	2018-01-01	2019-10-30	1500	1350
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
41    -- Finish the campaign titled RED. Verify that it behaves as desired.
42    call sp_finish_campaign('BLUE');
43    select * from campaign;
44    select * from invoice;
45    call sp_finish_campaign('GREEN'); -- should SIGNAL error condition
46
47
```

100% 24:43

Result Grid Filter Rows: Search Edit: Export/Import:

CAMPAIGN_NO	TITLE	CUSTOMER_ID	THEME	CAMPAIGNSTARTDATE	CAMPAIGNFINISHDATE	ESTIMATEDCOST	ACT
3	Green	3	Summer	2018-07-01	2019-10-30	1600	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

alerts 5 | campaign 40 | campaign 42

Action Output

	Time	Action	Response
1	10:36:20	call sp_finish_campaign('GREEN')	1 row(s) affected
2	10:36:23	select * from campaign LIMIT 0, 1000	4 row(s) returned