



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: Chris Jerylle Vargas Oidem

Student Number: 45476624

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

Chris Oidem 29th October 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

Workson 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

Alerts Table

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

2. Stored Programs.

```

/***** Trigger TR_OVERDUE *****/

delimiter //
drop trigger if exists tr_overdue
//

create trigger tr_overdue
after update on invoice
for each row

begin
declare msgdt date;
declare origin VARCHAR(50);
declare alertmsg VARCHAR(100);

set msgdt = curdate();
set origin = user();
set alertmsg = CONCAT('Invoice with number:', ' ', OLD.INVOICENO, ' ', 'is now overdue!');

if NEW.STATUS = "OVERDUE" then
insert into alerts(message_date, origin, message)
values (msgdt, origin, alertmsg);
end if;

end //
delimiter ;
```

```
/****** Procedure SP_FINISH_CAMPAIGN *****/
```

```
delimiter //
```

```
drop function if exists total_cost //
```

```
create function total_cost(title varchar(20))
```

```
returns double
```

```
DETERMINISTIC
```

```
begin
```

```
declare t_cost double;
```

```
select sum(b.HOUR*e.HOURLYRATE) into t_cost
```

```
from campaign a, workson b, staff c, staffongrade d, salarygrade e
```

```
where a.CAMPAIGN_NO = b.CAMPAIGN_NO
```

```
and c.STAFFNO = b.STAFFNO
```

```
and b.STAFFNO = d.STAFFNO
```

```
and d.GRADE = e.GRADE
```

```
and b.WDATE between d.STARTDATE and ifnull(d.FINISHDATE,now())
```

```
and a.TITLE = title;
```

```
return (t_cost);
```

```
end //
```

```
delimiter ;
```

```
delimiter //
```

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

```
create procedure sp_finish_campaign (in c_title varchar(30))
```

```
begin
```

```
declare finish_date date;
```

```
declare found int default 0;
```

```
set finish_date = curdate();
```

```
select 1 into found from campaign where TITLE = a_title;
```

```
if found = 1 then
```

```
update campaign
```

```
set CAMPAIGNFINISHDATE = f_date,
```

```
ACTUALCOST = t_cost(a_title)
```

```
where campaign.TITLE = a_title;
```

```
else
```

```
select 'ERROR! Campaign title does not exist' as ' Output';
```

```
end if;
```

```
end //
```

```
delimiter ;
```

```
/***** Procedure SYNC_INVOICE *****/
```

```
delimiter //
```

```
drop procedure if exists sync_invoice //
```

```
CREATE procedure sync_invoice()
```

```
begin
```

```
declare finished int default 0;
```

```
declare u_invoice int;
```

```
declare u_status varchar(10);
```

```
declare u_date date;
```

```
declare unpaid cursor for select INVOICENO, STATUS, DATEISSUED  
from invoice
```

```
where STATUS = 'UNPAID' and (now() - DATEISSUED) > 30;
```

```
declare continue handler for not found set finished=1;
```

```
open unpaid;
```

```
while finished !=1 do
```

```
fetch unpaid into u_invoice, u_status, u_date;
```

```
update invoice set STATUS = 'OVERDUE'
```

```
where INVOICENO = u_invoice and STATUS = u_status;
```

```
end while;
```

```
close unpaid;
```

```
end //
```

```
delimiter ;
```

3. Required Testing against Sample Database.

-- Turn off autocommit so you can have better control on what you are doing by rolling back transactions

set autocommit = 0;

-- Inspect the invoice and the alerts table

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

-- Update the invoice table

update invoice set STATUS = 'OVERDUE' where INVOICENO = 2;

-- Verify that the trigger you implemented works

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	2019-10-25	45476624@ash.science.mq.edu.au	Invoice with number: 2 is 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

-- Synchronise the invoice table and verify the procedure behaves as desired

call sync_invoice;

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	2019-10-25	45476624@ash.science.mq.edu.au	Invoice with number: 2 is now overdue!
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

-- Finish the campaign titled RED. Verify that it behaves as desired.

```
call sp_finish_campaign('RED');
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	2019-10-25	1000	675
NULL	NULL	NULL	NULL	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	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL

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

Output
ERROR! Campaign title does not exist

-- Synchronise the invoice table and verify the procedure behaves as desired

```
call sync_invoice;
```

```
select * from alerts;
```

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

```
rollback;
```

4. More Extensive Testing.

-- should SIGNAL error condition

```
call sp_finish_campaign('BLACK');
```

Output

ERROR! Campaign title does not exist

-- insert new records for testing purposes:

-- I added a 5th Grade to Staff No 5 due to a salary increase.

```
INSERT INTO `staffongrade` (`STAFFNO`, `GRADE`, `STARTDATE`)
VALUES ('5', '5', '2019-06-02');
```

-- insert new records for testing purposes:

-- A new campaign with the title Black and theme Winter is created for customer 3.

```
INSERT INTO `campaign` (`CAMPAIGN_NO`, `TITLE`, `CUSTOMER_ID`, `THEME`,
`CAMPAIGNSTARTDATE`, `ESTIMATEDCOST`)
VALUES ('3', 'Black', '3', 'Winter', '2019-05-01', '2000');
```

-- insert new records for testing purposes:

-- The following records show the Staff working on Campaign 3 and clocking in the number of hours spent working on the dates

```
INSERT INTO `workson` (`STAFFNO`, `CAMPAIGN_NO`, `WDATE`, `HOUR`)
VALUES ('1', '3', '2019-06-03', '9');
INSERT INTO `workson` (`STAFFNO`, `CAMPAIGN_NO`, `WDATE`, `HOUR`)
VALUES ('3', '3', '2019-06-05', '8');
INSERT INTO `workson` (`STAFFNO`, `CAMPAIGN_NO`, `WDATE`, `HOUR`)
VALUES ('5', '3', '2019-06-09', '9');
```

-- insert new records for testing purposes:

-- Invoice number has been issued and the status remains unpaid for customer 3

```
INSERT INTO `invoice` (`INVOICENO`, `CAMPAIGN_NO`, `DATEISSUED`,
`BALANCEOWING`, `STATUS`)
VALUES ('3', '3', '2019-09-10', '2315', 'UNPAID');
```

-- check all relevant tables

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
5	5	2019-06-02	NULL
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
3	Black	3	Winter	2019-05-01	NULL	2000	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from workson;

STAFFNO	CAMPAIGN_NO	WDATE	HOUR
1	2	2019-02-01	8
1	2	2019-05-01	7
1	3	2019-06-03	9
3	1	2018-05-01	5
3	3	2019-06-05	8
5	1	2018-02-01	7
5	1	2019-06-01	8
5	3	2019-06-09	9
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	UNPAID
3	3	2019-09-10	NULL	2315	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL

-- Update the invoice table

update invoice set STATUS = 'OVERDUE' where INVOICENO = 2;

update invoice set STATUS = 'OVERDUE' where INVOICENO = 3;

-- Verify that the trigger you implemented works

```
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
3	3	2019-09-10	NULL	2315	OVERDUE
NULL	NULL	NULL	NULL	NULL	NULL

```
select * from alerts;
```

message_no	message_date	origin	message
1	2019-10-26	45476624@ash.science.mq.edu.au	Invoice with number: 2 is now overdue!
2	2019-10-26	45476624@ash.science.mq.edu.au	Invoice with number: 3 is 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
3	3	2019-09-10	NULL	2315	UNPAID
NULL	NULL	NULL	NULL	NULL	NULL

```
select * from invoice;
```

message_no	message_date	origin	message
NULL	NULL	NULL	NULL

-- Finish the campaign titled *BLACK*. Verify that it behaves as desired.

```
call sp_finish_campaign('BLACK');
```

```
select * from campaign;
```

[illegible]

-- Finish the campaign titled RED. Verify that it behaves as desired.

call sp_finish_campaign('RED');

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	2019-10-26	1000	675
3	Black	3	Winter	2019-05-01	2019-10-26	2000	2315
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- Synchronise the invoice table and verify the procedure behaves as desired

call sync_invoice;

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
3	3	2019-09-10	NULL	2315	OVERDUE
NULL	NULL	NULL	NULL	NULL	NULL

select * from alerts;

message_no	message_date	origin	message
1	2019-10-26	45476624@ash.science.mq.edu.au	Invoice with number: 2 is now overdue!
2	2019-10-26	45476624@ash.science.mq.edu.au	Invoice with number: 3 is now overdue!
NULL	NULL	NULL	NULL

-- should SIGNAL error condition

call sp_finish_campaign('YELLOW');

Output
ERROR! Campaign title does not exist

5. Notes (optional).

- I went for consultation with my teacher to ask about the differences between stored functions and stored procedures.
- The practical and tutorials were very useful, but I had to do my own research about the codes to use as there were insufficient trigger examples.