Database Development 2

Database Development 2(DADE2)	
Assignment Number	1
Assignment Name	Formative Assessment
NQF Level	5
Credits	9
Due Date	
Marks	Total marks = 80 Formative assessments through the semester contribute towards the student's module mark and are used to assess progress and identify areas for improvement. This formative assessment will contribute 25% towards final mark. Take note of the following with regards to late submissions: a. One (1) day late (-5%) b. Two (2) days late (-10%) c. Three (3) days late (-15%)
Individual / Group Assignment	Individual
Lecturer Information	
Lecturer	
Lecturer E-mail	

Learning Objective:

Formative assessment 1 will cover the following concepts:

a. Database creation using T_SQL
b. Views
c. Stored procedures
d. Inserts
e. Code comments
f. Database backup

Attributes/Competencies Assessed:

a. 114048 - Create database access for a computer application using structured query language

Scope:

The scope of this formative assessment is based on a solid knwoledge coding databases using T_SQL.

Technical Aspects:

The number of pages for this formative assessment is ***16*** and the following font and size should be used in your report:

a. Font: Arial
b. Size: 12 and 14 for headings
c. Font colour: Black

Save and upload the report as a .PDF(No backgrounds) with the following naming convention:

a. Student no_StudentName_StudentSurname_ModuleCode_FA1(No ZIP folder uploads)

Ensure adequate referencing is used when using information from either books or internet. Plagiarism is a serious offecne and can result in 0% for the assessment when excessive work is copied without proper referencing.

Please complete the following and sign as requested for Portfolio of Evidence (POE)

a. Save code with screeshots of each question and upload when completed
b. Pre-Assessment agreement (Save, sign and submit as PDF)
c. Assessment Feedback Agreement (Save, sign and submit as PDF)

Mark allocation for report

See Mark allocation sheet below

Unit standard	*Specific outcome*	*Assessment criterion*
114048	*1*	*1*
	1	*2*
	1	*3*
	1	*4*
	2	*1*
	2	*2*
	2	*3*
	2	*4*
	2	*5*
	2	*6*
	3	*1*
	3	*2*
	3	*3*
	3	*4*
	3	*5*
	3	*6*
	3	*7*
	3	*8*
	4	*1*
	4	*2*
	4	*3*
	4	*4*

	4	*5*
	5	*1*
	5	*2*

Scenario

Suzi's Yoga Studio

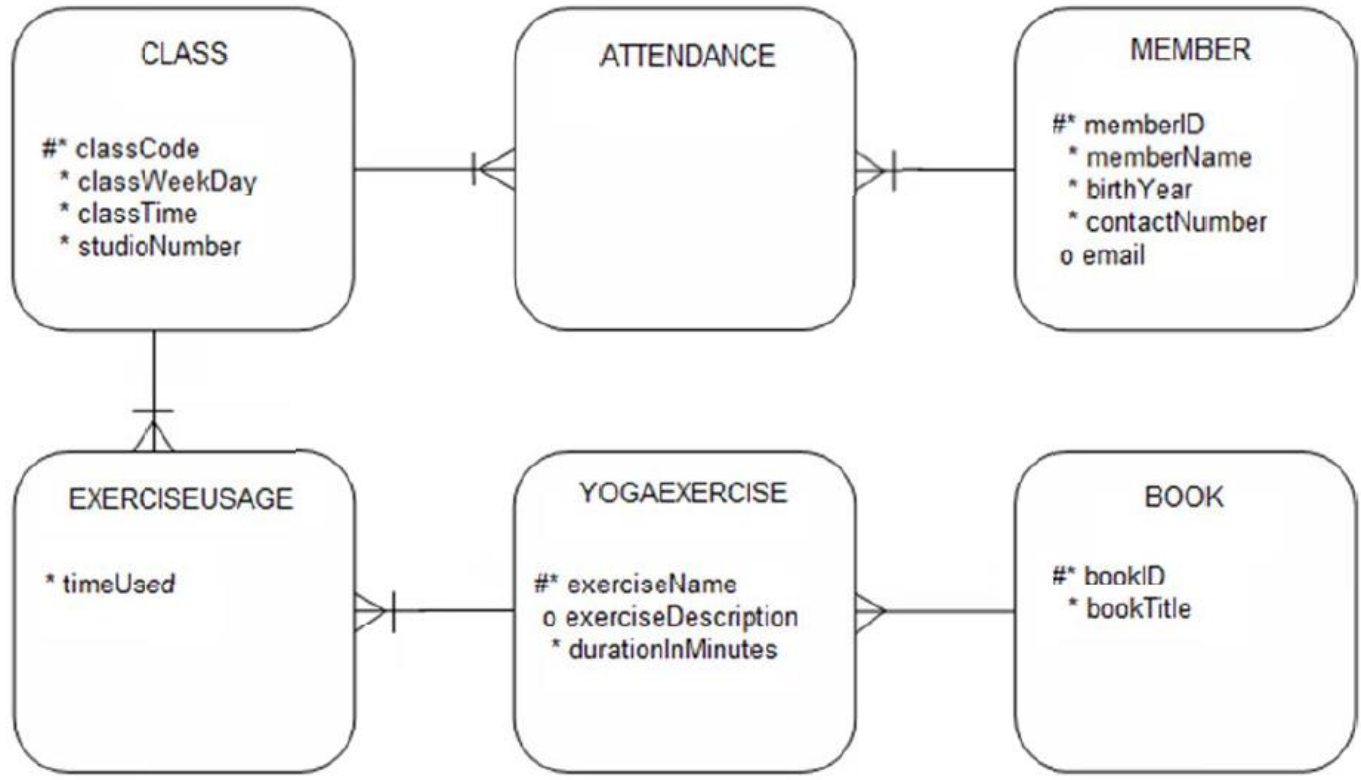
Suzi's family started their own yoga school and enlarged their house to accommodate four studios. Suzi has recruited you to be their database designer.

The database will store information on their members, yoga classes/sessions, and the various yoga exercises.

- People must be 16 years of age or older in order to become a member and they are welcome to assign themselves to more than one class/session per week.
- If a class is cancelled, Suzi must be able to contact members via telephone. Members who provide their email address will also be notified via email.
- Fixed yoga sessions are scheduled: for example, every Monday morning at 07:00 studio #1 is used and every Tuesday at 18:00 studio #4 is used. Each class/session record must contain sufficient information to indicate which weekday, time, and studio is reserved for it.
- Suzi references each exercise from a particular book and she would like to keep track of the book IDs and titles in case she needs to do more research on a particular exercise.
- Suzi needs to keep track of the different exercises performed at the sessions. She specifically wants to know the name, description, and the length of each exercise in terms of minutes.
- Suzi also needs to specify how many times an exercise must be performed per session/class.

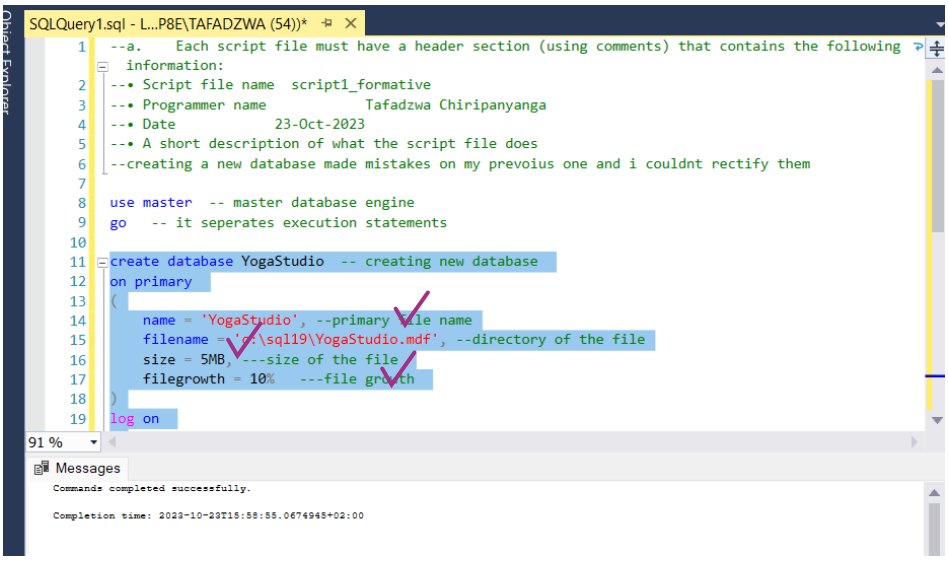
The following ER Diagram should be used to answer all questions that follow:

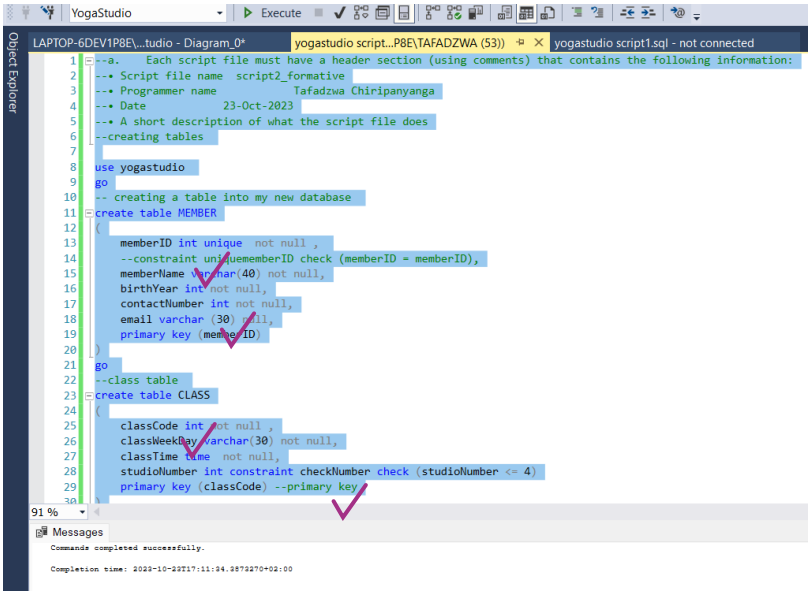
ERD from 3NF



Question 1 (16)

Create a script file that includes the code for creating the database and all of the tablesrequired in the project. PRIMARY KEY, FOREIGN KEY, UNIQUE, and CHECK constraintsmust be included. Make use of the IDENTITY property, DEFAULT values, and constraints. (16)





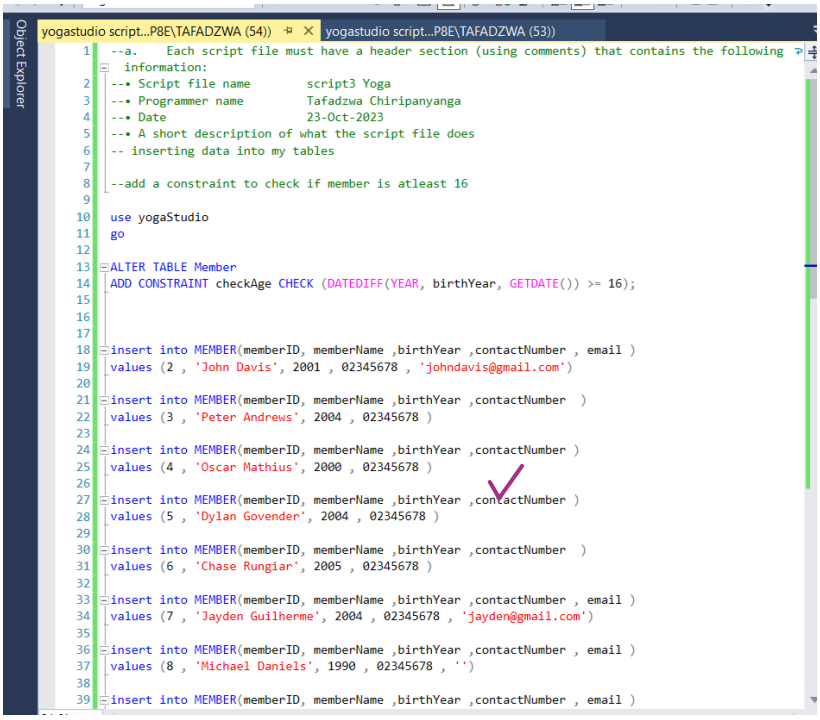
```
--a.    Each script file must have a header section (using comments) that contains the following information:
--• Script file name  script2_formative
--• Programmer name          Tafadzwa Chiripanyanga
--• Date              23-Oct-2023
--• A short description of what the script file does
--creating tables

use yogastudio
go
-- creating a table into my new database
create table MEMBER
(
    memberID int unique  not null ,
    --constraint uniquememberID check (memberID = memberID),
    memberName varchar(40) not null,
    birthYear int not null,
    contactNumber int not null,
    email varchar (30) null,
    primary key (memberID)
)
go
--class table
create table CLASS
(
    classCode int not null ,
    classWeekDay varchar(30) not null,
    classTime time  not null,
    studioNumber int constraint checkNumber check (studioNumber <= 4)
    primary key (classCode) --primary key
)
```

```
Commands completed successfully.

Completion time: 2023-10-23T17:11:24.2872270+02:00
```

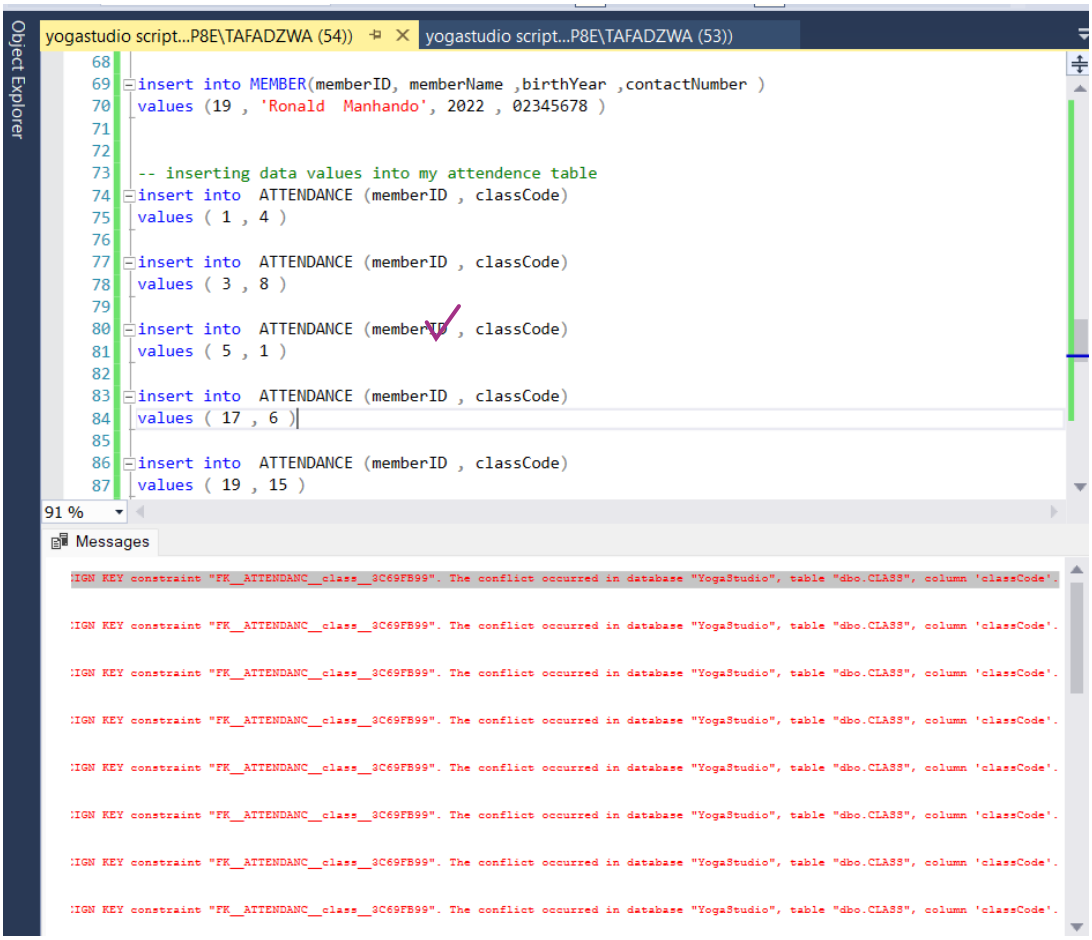
not all tables DDL shown: -8

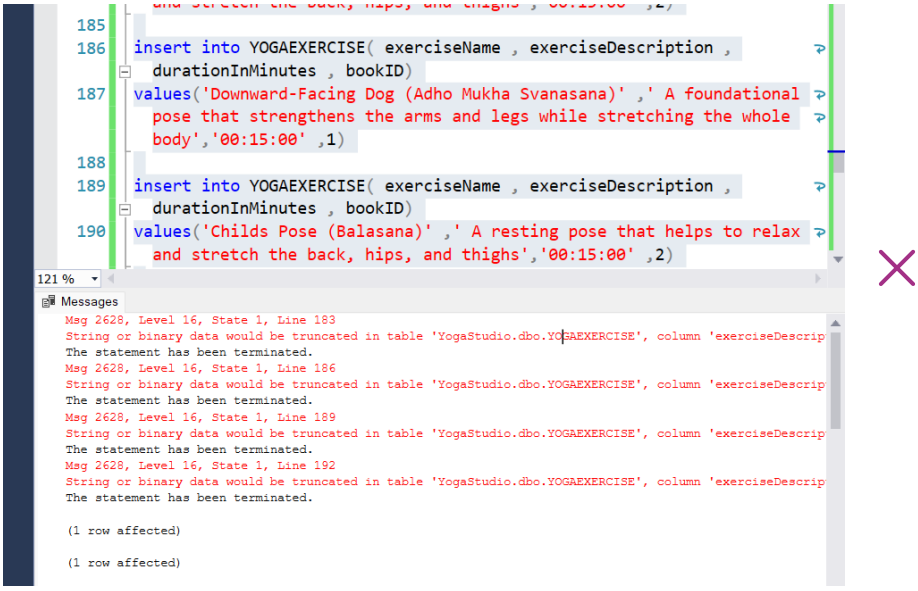
Question 2 (6)

Create a script file to insert data into your user created tables. Add at least four records for each table without foreign keys, and seven records for each table that does contain a foreign key. (6)



```
--a.    Each script file must have a header section (using comments) that contains the following
 information:
--• Script file name        script3 Yoga
--• Programmer name         Tafadzwa Chiripanyanga
--• Date                    23-Oct-2023
--• A short description of what the script file does
-- inserting data into my tables

--add a constraint to check if member is atleast 16

use yogaStudio
go

ALTER TABLE Member
ADD CONSTRAINT checkAge CHECK (DATEDIFF(YEAR, birthYear, GETDATE()) >= 16);



insert into MEMBER(memberID, memberName ,birthYear ,contactNumber , email )
values (2 , 'John Davis', 2001 , 02345678 , 'johndavis@gmail.com')

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber  )
values (3 , 'Peter Andrews', 2004 , 02345678 )

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber )
values (4 , 'Oscar Mathius', 2000 , 02345678 )

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber )
values (5 , 'Dylan Govender', 2004 , 02345678 )

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber  )
values (6 , 'Chase Rungiar', 2005 , 02345678 )

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber , email )
values (7 , 'Jayden Guilherme', 2004 , 02345678 , 'jayden@gmail.com')

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber , email )
values (8 , 'Michael Daniels', 1990 , 02345678 , '')

insert into MEMBER(memberID, memberName ,birthYear ,contactNumber , email )
```


Object Explorer
yogastudio script...P8E\TAFADZWA (54))
yogastudio script...P8E\TAFADZWA (53))
insert into MEMBER(memberID, memberName ,birthYear ,contactNumber)
values (19 , 'Ronald Manhando', 2022 , 02345678)
-- inserting data values into my attendence table
insert into ATTENDANCE (memberID , classCode)
values (1 , 4)
insert into ATTENDANCE (memberID , classCode)
values (3 , 8)
insert into ATTENDANCE (memberID , classCode)
values (5 , 1)
insert into ATTENDANCE (memberID , classCode)
values (17 , 6)
insert into ATTENDANCE (memberID , classCode)
values (19 , 15)
91 %
Messages
:IGN KEY constraint "FK__ATTENDANC__class__3C69FB99". The conflict occurred in database "YogaStudio", table "dbo.CLASS", column 'classCode'.


insert into YOGAEXERCISE(exerciseName , exerciseDescription , durationInMinutes , bookID)
values('Downward-Facing Dog (Adho Mukha Svanasana)' ,' A foundational pose that strengthens the arms and legs while stretching the whole body','00:15:00' ,1)
insert into YOGAEXERCISE(exerciseName , exerciseDescription , durationInMinutes , bookID)
values('Childs Pose (Balasana)' ,' A resting pose that helps to relax and stretch the back, hips, and thighs','00:15:00' ,2)
121 %
Messages
Msg 2628, Level 16, State 1, Line 183
String or binary data would be truncated in table 'YogaStudio.dbo.YOGAEXERCISE', column 'exerciseDescrip
The statement has been terminated.
Msg 2628, Level 16, State 1, Line 186
String or binary data would be truncated in table 'YogaStudio.dbo.YOGAEXERCISE', column 'exerciseDescrip
The statement has been terminated.
Msg 2628, Level 16, State 1, Line 189
String or binary data would be truncated in table 'YogaStudio.dbo.YOGAEXERCISE', column 'exerciseDescrip
The statement has been terminated.
Msg 2628, Level 16, State 1, Line 192
String or binary data would be truncated in table 'YogaStudio.dbo.YOGAEXERCISE', column 'exerciseDescrip
The statement has been terminated.
(1 row affected)
(1 row affected)

```
74  -- inserting data values into my attendence table
75  insert into  ATTENDANCE (memberID , classCode)
76  values ( 1 , 4 )
77
78  insert into  ATTENDANCE(memberID , classCode)
79  values ( 3 , 8 )
80
81  insert into  ATTENDANCE (memberID , classCode)
82  values ( 5 , 1 )
83
```

121 %

Messages

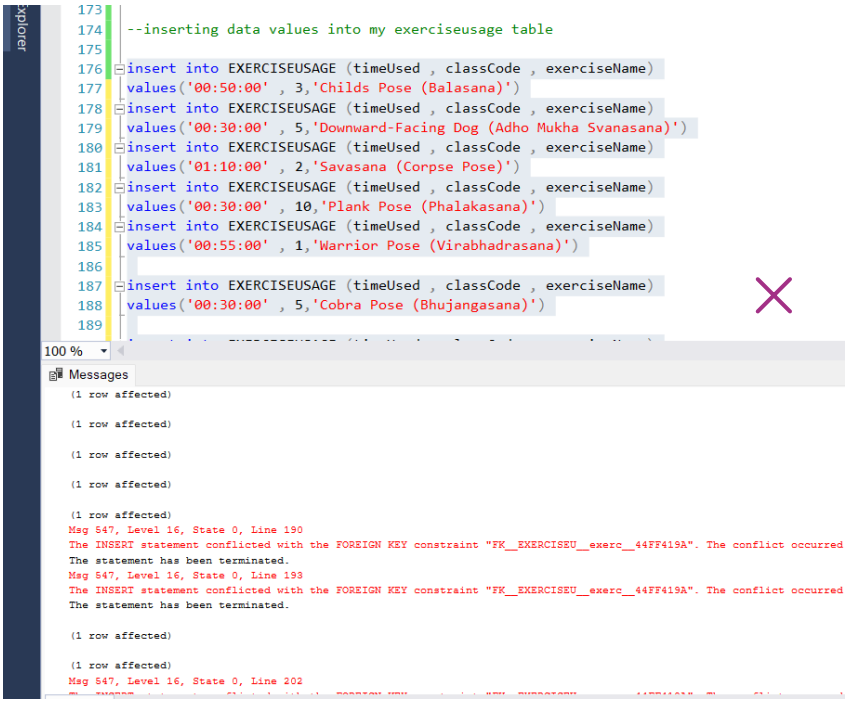
```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

```
220  --inserting data values into my book Table
221  insert into BOOK(bookID , bookTitle)
222  values(1 , 'Light on Yoga')
223  insert into BOOK(bookID , bookTitle)
224  values(2 , 'The Heart of Yoga')
225  insert into BOOK(bookID , bookTitle)
226  values(3 , 'The Key Muscles of Yoga')
227  insert into BOOK(bookID , bookTitle)
228  values(4 , 'The Yoga Bible" by Christina Brown')
229  insert into BOOK(bookID , bookTitle)
230  values(5 , 'The Secret Power of Yoga')
231  insert into BOOK(bookID , bookTitle)
232  values(6 , 'Yoga Anatomy')
233  insert into BOOK(bookID , bookTitle)
```

21 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```



```
173
174  --inserting data values into my exerciseusage table
175
176  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
177  values('00:50:00' , 3,'Childs Pose (Balasana)')
178  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
179  values('00:30:00' , 5,'Downward-Facing Dog (Adho Mukha Svanasana)')
180  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
181  values('01:10:00' , 2,'Savasana (Corpse Pose)')
182  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
183  values('00:30:00' , 10,'Plank Pose (Phalakasana)')
184  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
185  values('00:55:00' , 1,'Warrior Pose (Virabhadrasana)')
186
187  insert into EXERCISEUSAGE (timeUsed , classCode , exerciseName)
188  values('00:30:00' , 5,'Cobra Pose (Bhujangasana)')
189
```

100 %

Messages

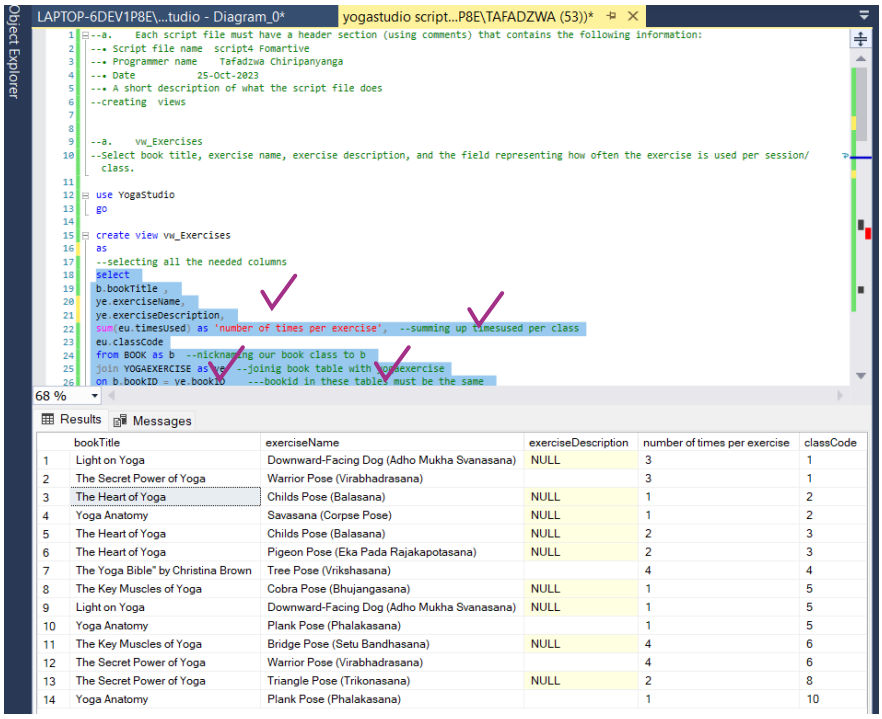
```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
Msg 547, Level 16, State 0, Line 190
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__EXERCISEU__exerc__44FF419A". The conflict occurred
The statement has been terminated.
Msg 547, Level 16, State 0, Line 193
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__EXERCISEU__exerc__44FF419A". The conflict occurred
The statement has been terminated.

(1 row affected)

(1 row affected)
Msg 547, Level 16, State 0, Line 202
```

Question 3 (16)

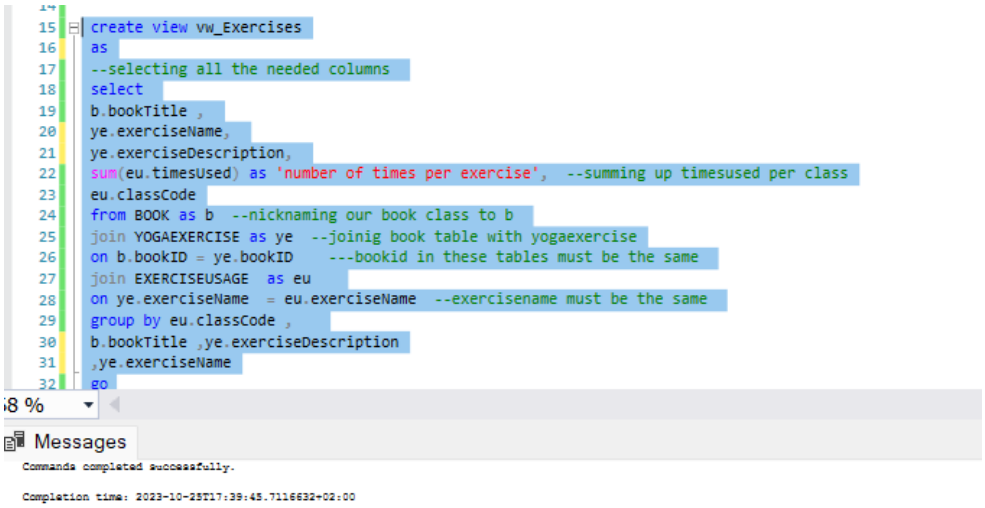
Create a script file to create the following views: (16)

a. vw_Exercises

Select book title, exercise name, exercise description, and the field representing how often the exercise is used per session/class.



	bookTitle	exerciseName	exerciseDescription	number of times per exercise	classCode
1	Light on Yoga	Downward-Facing Dog (Adho Mukha Svanasana)	NULL	3	1
2	The Secret Power of Yoga	Warrior Pose (Virabhadrasana)		3	1
3	The Heart of Yoga	Childs Pose (Balasana)	NULL	1	2
4	Yoga Anatomy	Savasana (Corpse Pose)	NULL	1	2
5	The Heart of Yoga	Childs Pose (Balasana)	NULL	2	3
6	The Heart of Yoga	Pigeon Pose (Eka Pada Rajakapotasana)	NULL	2	3
7	The Yoga Bible" by Christina Brown	Tree Pose (Vrikshasana)		4	4
8	The Key Muscles of Yoga	Cobra Pose (Bhujangasana)	NULL	1	5
9	Light on Yoga	Downward-Facing Dog (Adho Mukha Svanasana)	NULL	1	5
10	Yoga Anatomy	Plank Pose (Phalakasana)		1	5
11	The Key Muscles of Yoga	Bridge Pose (Setu Bandhasana)	NULL	4	6
12	The Secret Power of Yoga	Warrior Pose (Virabhadrasana)		4	6
13	The Secret Power of Yoga	Triangle Pose (Trikonasana)	NULL	2	8
14	Yoga Anatomy	Plank Pose (Phalakasana)		1	10



b. vw_ClassAttendance

Select class code, class week day, class time, studio number, and the total number of people attending each class. Only show the records where less than four people attend the class.

```sql
-- b.   vw_ClassAttendance
--Select class code, class week day, class time, studio number, and the total number of people attending each class.
  Only show the records where less than four people attend the class.

select   --selecting the required columns
c.classCode,
c.classTime,
c.studioNumber,
Count(distinct m.memberID) as total,  -- function used to count unique member
c.classWeekDay
from MEMBER  as m
join  ATTENDANCE as a
on m.memberID = a.memberID
join Class as c
on c.classCode = a.classCode
group by
c.studioNumber,
c.classCode,
c.classTime,
c.classWeekDay
having count(distinct m.memberID) < 4  --filtering the results
go
```

75 %

Results | Messages

	classCode	classTime	studioNumber	total	classWeekDay
1	1	15:00:00.0000000	2	2	Tuesday
2	3	17:30:00.0000000	4	1	Thursday
3	4	15:00:00.0000000	3	2	Monday
4	6	16:00:00.0000000	1	2	Wednesday
5	7	17:30:00.0000000	4	2	Thursday
6	8	15:00:00.0000000	3	1	Monday
7	10	16:00:00.0000000	1	2	Wednesday
8	11	17:30:00.0000000	4	1	Thursday
9	15	17:30:00.0000000	4	1	Thursday

```sql
create view vw_ClassAttendance
as
select   --selecting the required columns
c.classCode,
c.classTime,
c.studioNumber,
Count(distinct m.memberID) as total,  -- function used to count unique member
c.classWeekDay
from MEMBER  as m
join  ATTENDANCE as a
on m.memberID = a.memberID
join Class as c
on c.classCode = a.classCode
group by
c.studioNumber,
c.classCode,
c.classTime,
c.classWeekDay
having count(distinct m.memberID) < 4  --filtering the results
go
```

%

Messages

```
Commands completed successfully.

Completion time: 2023-10-25T18:19:22.0722904+02:00
```

c. vw_ExercisesUsed

Select the exercise name, description, duration, and how often each exercise is used. Only show the four most common exercises (use the alias 'total times performed').

```sql
select  top 4   -- selects top 4 rows
ye.exerciseName,
ye.exerciseDescription,
ye.durationInMinutes,
count( distinct eu.timesUsed) as 'times used on exercise'
from YOGAEXERCISE as ye
join EXERCISEUSAGE as eu
on  eu.exerciseName = ye.exerciseName
group by ye.exerciseName, ye.durationInMinutes,ye.durationInMinutes ,ye.exerciseDescription
order by count(distinct eu.timesUsed) desc;   --odering in descending order

go
```

```sql
create view  vw_ExercisesUsed
as
select  top 4   -- selects top 4 rows
ye.exerciseName,
ye.exerciseDescription,
ye.durationInMinutes,
count( distinct eu.timesUsed) as 'times used on exercise'
from YOGAEXERCISE as ye
join EXERCISEUSAGE as eu
on  eu.exerciseName = ye.exerciseName
group by ye.exerciseName, ye.durationInMinutes,ye.durationInMinutes ,ye.exerciseDescription
order by count(distinct eu.timesUsed) desc;   --odering in descending order

go
```

```
Commands completed successfully.

Completion time: 2023-10-25T21:19:43.5893924+02:00
```

d. vw_TotalClassesPerMember

Select all member names as well as the total number of classes that each member will attend. HINT: Use the COUNT function.

```sql
select m.memberName,    --selecting required columns
a.classCode,
count (c.classCode) as 'total number of classes a member has'
from MEMBER as m
join ATTENDANCE as a
on a.memberID = m.memberID
join CLASS as c
on c.classCode = a.classCode
group by m.memberName  , m.memberID ,a.classCode


--alter table EXERCISEUSAGE
--ADD CONSTRAINT  classCode foreign key (classCode) references CLASS(classCode)
```

Results Messages

	memberName	classCode	total number of classes a member has
1	Dylan Govender	1	1
2	Syden Syc		
3	Chase Rungiar	3	1
4	Tafadzwa Chiripanyanga	4	1
5	Jayden Guilherme	4	1
6	Peter Andrews	6	2
7	Aweh Aweh	6	1
8	Ubert Albert	7	1
9	Faith Mathews	7	1

Click to select the whole column

```sql
--d.    vw_TotalClassesPerMember
--Select all member names as well as the total number of classes that each member will attend. HINT: Use the COUNT
  function.

create view vw_TotalClassesPerMember
as
select m.memberName,    --selecting required columns
a.classCode,
count (c.classCode) as 'total number of classes a member has'
from MEMBER as m
join ATTENDANCE as a
on a.memberID = m.memberID
join CLASS as c
on  c.classCode = a.classCode
group by m.memberName  , m.memberID ,a.classCode
```

Messages

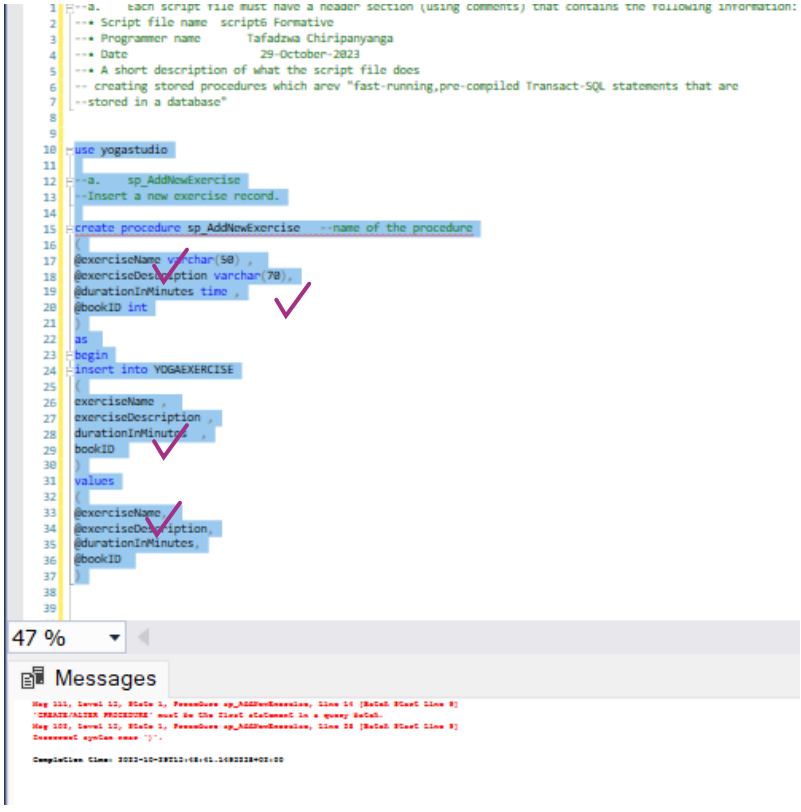
```
ommands completed successfully.

ompletion time: 2023-10-25T21:36:58.0202720+02:00
```

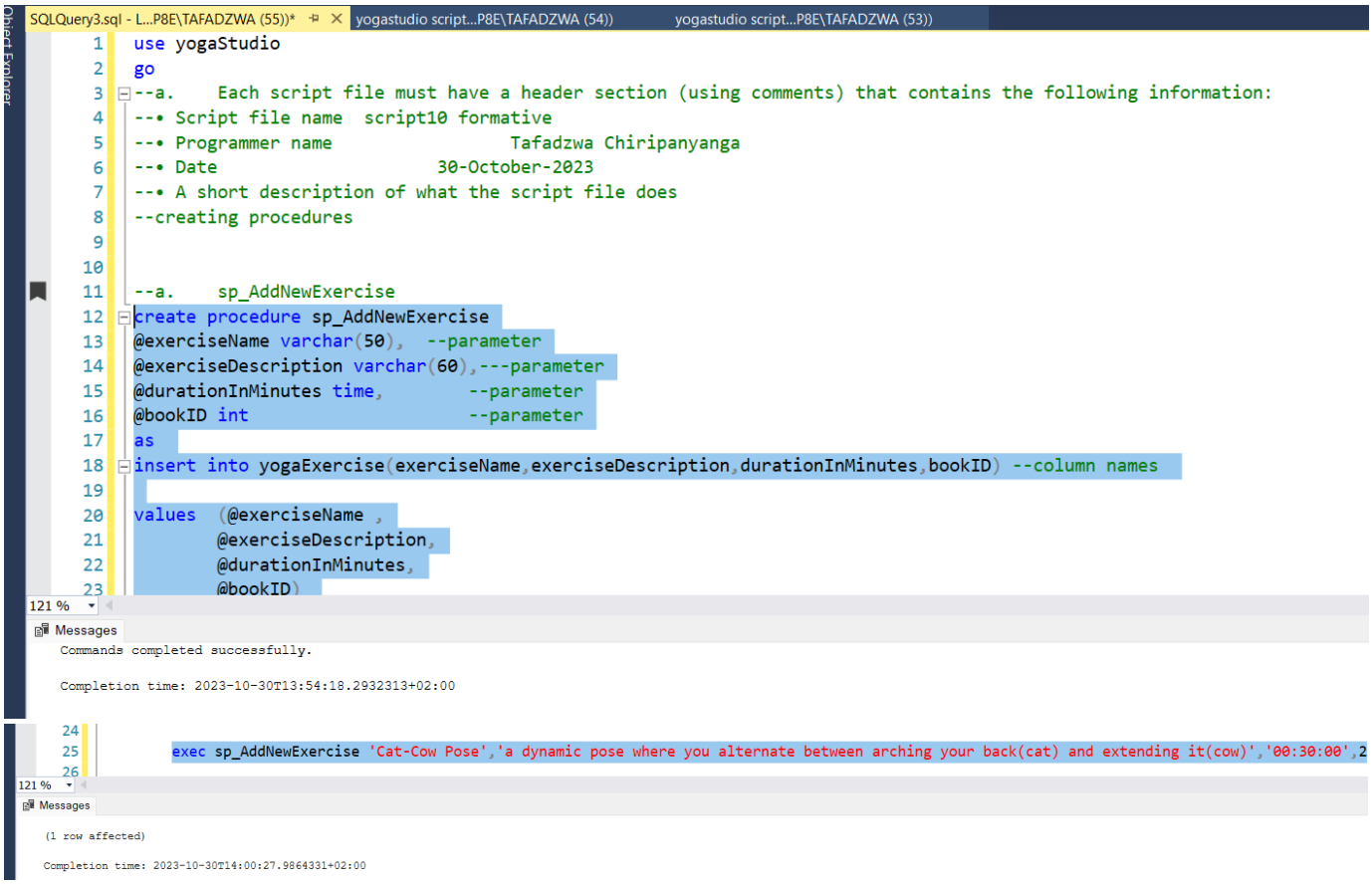
Question 4 (16)

Create a script file to create the following stored procedures: (16)

a. sp_AddNewExercise

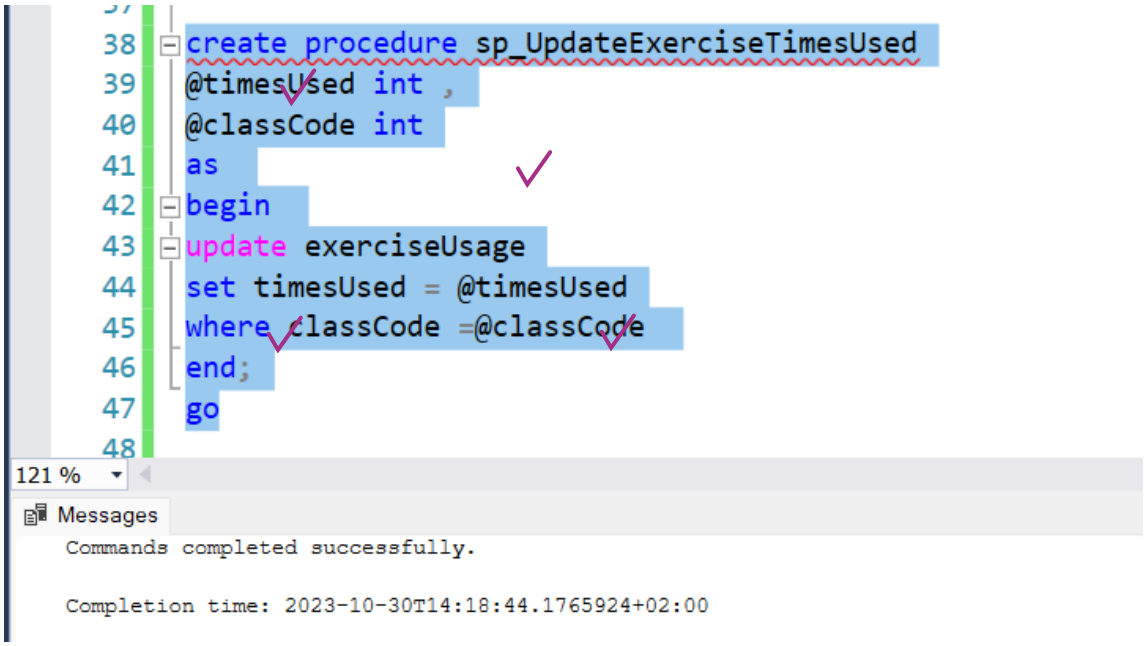
Insert a new exercise record.





b. sp_UpdateExerciseTimesUsed

Update the field which specifies how many times anexercise will be performed in each class.



```
48
49  exec sp_UpdateExerciseTimesUsed 5 , 1
50
51
52
53
54
55
```

```
21 %
Messages

(2 rows affected)

Completion time: 2023-10-30T14:21:46.6005597+02:00
```

c. sp_DeleteBook

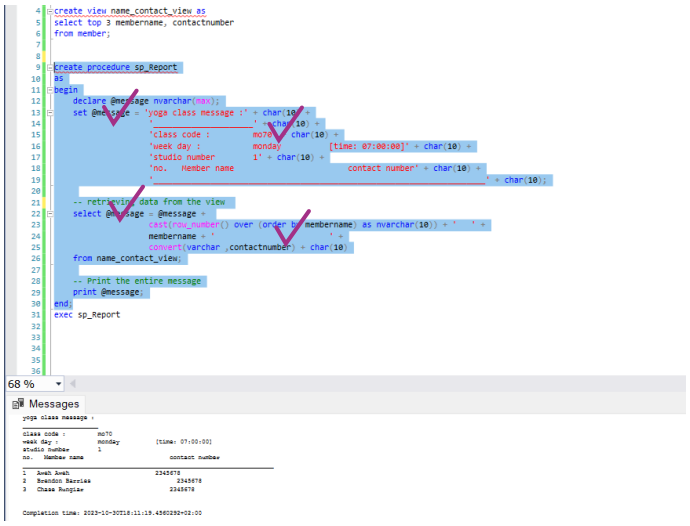
Delete a book record. A book may only be deleted if it is not contained in the vw_Exercises view.

```
51  --c.    sp_DeleteBook
52  --Delete a book record. A book may only be deleted if it is not
      contained in the vw_Exercises view.
53  create procedure sp_DeleteBook
54    @bookTitle varchar(40)
55  as
56  begin
57    if not exists (select 1 from vw_Exercises where bookTitle =
        @bookTitle)
58    begin
59      delete from book
60      where bookTitle = @bookTitle;
61    end
62   else
63    begin
64     print 'The book is contained in a view so cannot be deleted.';
65    end
66  end;
67
68  select *
69  from
```

```
121 %
Messages
Commands completed successfully.

Completion time: 2023-10-30T14:52:02.1157185+02:00
```

d. sp_Report

Print the details of a specified class and each member's name and contact number assigned to the class. Your report's output should have exactly the same format as shown below:

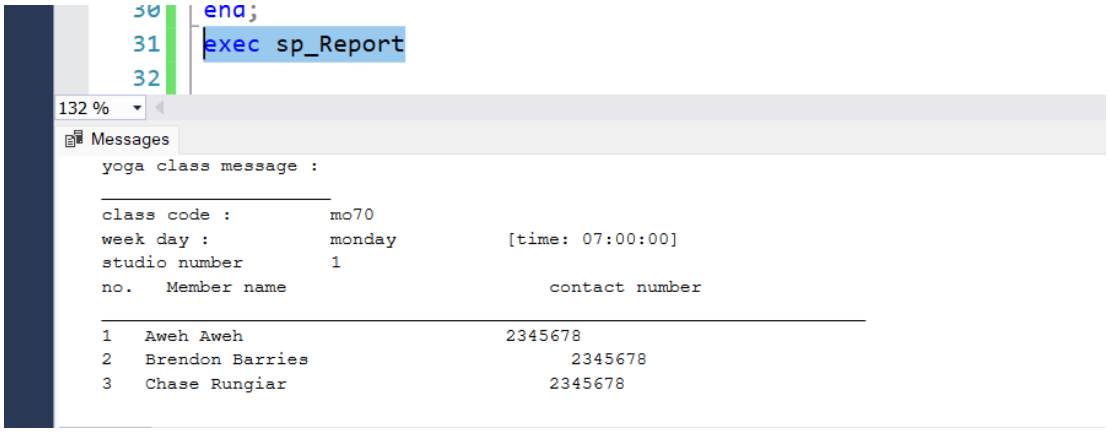


```
create view name_contact_view as
select top 3 membername, contactnumber
from member;

create procedure sp_Report
as
begin
    declare @message nvarchar(max);
    set @message = 'yoga class message :' + char(10) +
                   '_____' + char(10) +
                   'class code :        mo70' + char(10) +
                   'week day :          monday          [time: 07:00:00]' + char(10) +
                   'studio number       1' + char(10) +
                   'no.   Member name                          contact number' + char(10) +
                   '_____' + char(10);

    -- retrieving data from the view
    select @message = @message +
                   cast(row_number() over (order by membername) as nvarchar(10)) + '   ' +
                   membername + '          ' +
                   convert(varchar ,contactnumber) + char(10)
    from name_contact_view;

    -- Print the entire message
    print @message;
end;
exec sp_Report
```

68 %

Messages

```
yoga class message :
_____
class code :        mo70
week day :          monday          [time: 07:00:00]
studio number       1
no.   Member name                          contact number
_____
1   Aweh Aweh                       2345678
2   Brendon Barries                 2345678
3   Chase Rungiar                   2345678

Completion time: 2023-10-30T16:11:19.4560292+02:00
```



```
end;
exec sp_Report
```

132 %

Messages

```
yoga class message :
_____
class code :        mo70
week day :          monday          [time: 07:00:00]
studio number       1
no.   Member name                          contact number
_____
1   Aweh Aweh                       2345678
2   Brendon Barries                 2345678
3   Chase Rungiar                   2345678
```

```
YOGA CLASS REPORT:

_____

Class code:     mo70
Week day:       Monday    [Time:    07:00:00]
Studio Number   1

No. Member name                 Contact number

_____
1.  Andren du Preez             +27 83 562 3953
2.  Jenny Ritchie               (051) 861 2571
3.  Tom Edwards                 (021) 914 8000
```

Question 5 (4)

Create a script file to create at least two triggers for your project. There are no specifications for what types of triggers you must create. You should apply your knowledge and create triggers that you think would be appropriate. (4)

```
--a.    Each script file must have a header section (using comments) that contains the following information:
--• Script file name        script6 formative
--• Programmer name         Tafadzwa Chiripanyanga
--• Date                    29-October-2023
--• A short description of what the script file does
--creating triggers


--Create a script file to create at least two triggers for your project. There are no specifications for what types of triggers you must create. You should
  apply your knowledge and create triggers that you think would be appropriate.

create trigger add_trigger  --name of the trigger
on MEMBER   -- inserting the trigger on member table
after insert  -- after inserting a new value
as
print 'new member has been added succesfully ' -- this will be shown after inserting a new member
go
insert into MEMBER(memberID, memberName , birthyear , contactNumber, email) --column names
```

```
Messages
Commands completed successfully.

Completion time: 2023-10-29T16:41:25.7922562+02:00
```

```
insert into MEMBER(memberID, memberName , birthyear , contactNumber, email) --column names
values (20,'Spring Bokke ',1995,0813456873,'springbokke2023@gmail.com') --inserting values
```

```
Messages
new member has been added succesfully

(1 row affected)

Completion time: 2023-10-29T16:45:39.5404573+02:00
```

```
create trigger updateMember_trigger
on MEMBER
instead of insert , delete , update
as
print 'you can not edit this table'
go
```

```
Messages
Commands completed successfully.

Completion time: 2023-10-29T16:52:51.7720716+02:00
```

```
21  create trigger update_Member_trigger
22  on MEMBER
23  instead of insert , delete , update
24  as
25  raiserror('you cannot edit this table',2,15 );
26  go
27
```

```
Messages
Commands completed successfully.

Completion time: 2023-10-29T17:02:21.0825792+02:00
```

```
29  insert into MEMBER(memberID, memberName , birthyear , contactNumber,
      email) --column names
30  values (22,'Aweh bro ',2000,0813456873,'awehbro@gmail.com') --
      inserting values
31
```

```
Messages
you cannot edit this table
Msg 50000, Level 2, State 15

(1 row affected)

Completion time: 2023-10-29T17:03:45.2634964+02:00
```

Question 6 (4)

Create a script file to delete the database that you create. (4)

```
use yogaStudio
go
--a.    Each script file must have a header section (using comments)
  that contains the following information:
--• Script file name  script8 formative
--• Programmer name                Tafadzwa Chiripanyanga
--• Date                      30-October-2023
--• A short description of what the script file does
--script to delete my database



--Create a script file to delete the database that you create.

--deleting my yogastudio database
if exists (select name  from master.dbo.sysdatabases
where name = 'yogaStudio')
drop database yogaStudio  --to drop is to delete
go
```

Question 7 (4)

Create two (2) appropriate indices on your tables. (4)

SQLQuery2.sql - L...P8E\TAFADZWA (54))* | yogastudio script...P8E\TAFADZWA (53))

```
  that contains the following information:
--• Script file name  script9 formative
--• Programmer name                Tafadzwa Chiripanyanga
--• Date                      30-October-2023
--• A short description of what the script file does
--creating indices   which are used  To accelerate data access,


--Create two (2) appropriate indices on your tables.
create index memberID_index  --name of index
on MEMBER(memberID)  --table im indexing and its column
go
```

121 %

Messages

```
Commands completed successfully.

Completion time: 2023-10-30T13:10:21.9096136+02:00
```

```
14  go
15  create index exerciseName_index   --index name
16  on EXERCISEUSAGE(exerciseName) --table name and its columns
17  go
18
19
```

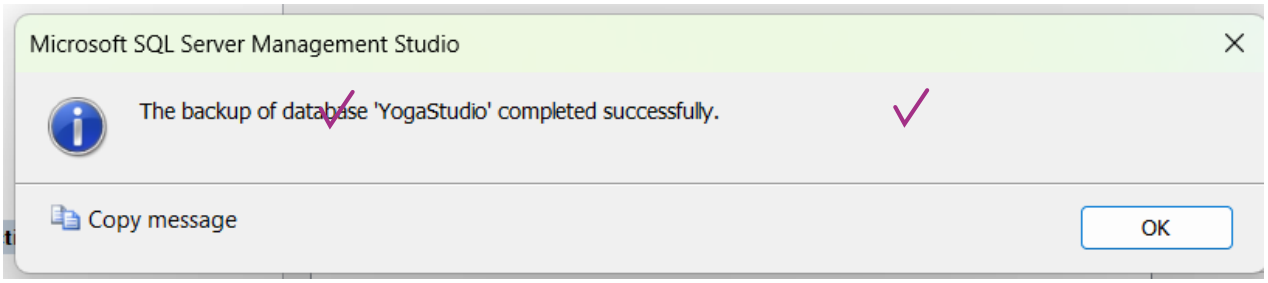
121 %

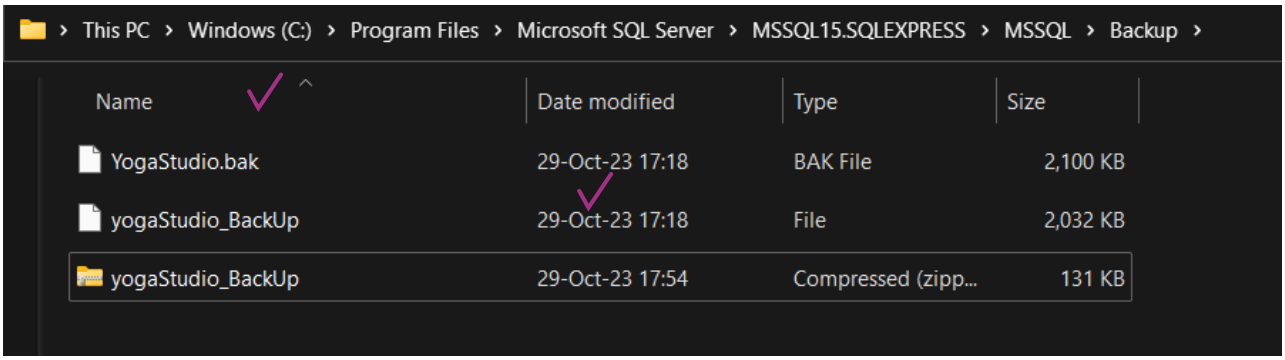
Messages

```
Commands completed successfully.

Completion time: 2023-10-30T13:26:43.4994440+02:00
```

Question 8 (4)

Using the GUI, create a full backup of your database and zip it with WinZip. (4)





Question 9 (10)

a. Each script file must have a header section (using comments) that contains the following information:
   - Script file name
   - Programmer name
   - Date
   - A short description of what the script file does

b. Ensure in code comments are added (10)

[TOTAL: 80]

Mark allocation for student			
Section	Sub-section	Maximum Mark	Learner mark
Body of the report	Question 1	16	8
	Question 2	6	4
	Question 3	16	
	Question 4	16	
	Question 5	4	
	Question 6	4	
	Question 7	4	
	Question 8	4	
	Question 9	10	

Deductions	1 day late	-5	
	2 days late	-10	
	3 days late	-15	
	Total:	80	70

PRE-ASSESSMENT AGREEMENT

Assessment Preparation: Preparing the Candidate

Student name and surname	Tafadzwa Chiripanyanga	Date	30-October-2023
		Time	6pm
Assessor name and surname		Venue	online

How to prepare the candidate	Document Requirements	Agree (tick)	Action Required
Explain to the candidate why you are meeting and the purpose of the assessment.	Assessment Policy Assessment process	✓	
Discuss the assessment plan in detail.	Assessment strategy	✓	
Explain assessment process, show assessment instruments to candidate and describe assessment conditions.	Assessment instruments	✓	
Identify the role-players	Assessors	✓	

during assessment.	Moderator		
Describe the evidence required to be declared competent.	Examples of evidence	✓	
Explain how evidence will be judged.	Mark allocation explained	✓	
Explain to the candidate how to prepare: Give candidate assessment task description.	Assessment task description	✓	
Confirm with the candidate what he/she should bring to the assessment.	Detailed briefing on exact requirements to be given to candidate in writing	✓	
Ensure that candidate understands the procedures of all assessment practices.	Appeals Policy Appeals procedure Assessment Policy Assessment Procedure Moderation Policy Moderation procedure Verification Policy Verification Procedure	✓	
Ask the candidate if he/she foresees any problems or identify any special needs.	List needs	✓	

Agreed Assessment Plan			
Student name and surname:		Tafadzwa Chiripanyanga	
Assessor name and surname:			
Module name:		Databased Development 2	
Unit Standard/s:		US114048	
Type of Assessment i.e. Formative assignment, Formative test, Formative Practical, Summative etc.		Formative Assessment 1	
Special Assessment Requirements:		N/A	
Event	**Date, time and location**	**Resources required**	**Evidence to be generated**
Assessments due date		Assessments	Completed documentation
Complete activity on MyAIE and upload to			Completed Portfolio

MyAIE			of Evidence
Submit Portfolio of Evidence			

Assessor Roles and Responsibility	
Roles	Assessor Guide Feedback Agent Reviewer
Responsibilities	Consult candidate re-assessment, assessment process and plan. Agree assessment process and plan with candidate. Forward documentation to candidate: plan, guide and assessment instruments. Assess candidate with the use of different instruments. Provide feedback on assessment findings. Support candidate through assessment process. Source feedback from candidate on assessment process. Review assessment process and outcome. Use assessment process as opportunity to transform assessment activities and outcomes.

Candidate Roles and Responsibility	
Roles	• Leaner • Feedback agent • Reviewer
Responsibilities	• Be available for assessment. • Be actively involved in the consultative process. • Learn from the assessment process. • Provide feedback to the assessor in terms of the assessment as learning activity. • Provide feedback to the assessor on the efficacy of the assessment process. • Review own role and assessor role in the assessment process.
Assessment Instruments	• Portfolio of Evidence • Questionnaire • Report • Presentation • Reflexive questions

	• Work sample • Practical's • Group Activity • Research activities

Assessment Process	
• Evaluation of POE addressing Essential Embedded Knowledge in unit standards. • Evaluation of Research Projects and other evidence addressing specific unit standards. • Consultation: assessment plan and assessment activities and instruments. Pre-assessment moderation and interviews conducted at this stage. • Observation: feedback on assessment against specific outcomes, critical outcomes in unit standards. • Feedback: to candidate regarding sufficiency of evidence and possible interview to gain supplementary evidence. • Feedback to candidate regarding assessment findings as well as review process.	
Feedback	Written feedback to be given to all stakeholders at the end of the assessment process, as well as verbal feedback to the candidate during assessment activities.
Recording Process	Process and findings to be recorded and submitted for record keeping purposes as well as moderation and verification.
Review Process	The review process is the responsibility of the assessor and the candidate. Joint reviewing will take place after feedback has been given to the candidate.
Right to appeal	The candidate must be advised of the right to appeal.
Resources Required	Assignments • POE • Assessments • Guides

I confirm that:

- I have been consulted on and have agreed to the training and assessment process as detailed in the assessment guide.
- I have been advised of my right to appeal against any assessment that is unfair, unreliable, invalid or impracticable.
- I have read and understood the appeal procedure.
- I know that assessments may be moderated or verified by an external party.
- The purpose of the assessment has been clearly explained to me.
- The criteria have been discussed with me, and I know I will be assessed against these criteria.
- I know when and where I will be assessed, and I was given fair notice.
- I know how the assessment will be done, and any other requirements related to the assessment.

Signed: ___Tafadzwa_____ Date: 30-October-2023

Overall Assessment Decision	Competent	C	Not yet competent	
Student's Signature	Tafadzwa		Date:	30-October-2023
Assessor's Signature			Date:	
Moderator's Signature			Date:	

ASSESSMENT FEEDBACK AGREEMENT

Assessment feedback: Feedback to learner

Qualification Name:	
Qualification SAQA Number:	
Subject Name:	Databased Development 2
Subject Code:	DADE2
Assessment Name:	Formative Assessment 1
Assessment Code:	DADE2_FA1
Assessment Type:	Formative

Feedback report	1st Attempt	2nd Attempt

	C	NYC	C	NYC
Unit standard Number(s)				
US114048	C			
SO1,AC1				
SO1,AC2				
SO1,AC3				
SO1,AC4				
SO2,AC1				
SO2,AC2				
SO2,AC3				
SO2,AC4				
SO2,AC5				
SO2,AC6				
SO3,AC1				
SO3,AC2				
SO3,AC3				
SO3,AC4				
SO3,AC5				
SO3,AC6				
SO3,AC7				
SO3,AC8				
SO4,AC1				
SO4,AC2				

SO4,AC3				
SO4,AC4				
SO4,AC5				
SO5,AC1				
SO5,AC2				

General feedback to learner (Attempt 1) Supply comprehensive feedback why learner is found NYC
student found competent

Learner Number:	258196		
Learner name and surname:	Tafadzwa Chiripanyanga	Date:	30-October-2023
Learner Signature:	*Tafadzwa*		
Lecturer name and surname:	Daniel van Deventer	Date:	10/11/23
Lecturer Signature:	DVD		
Assessor name and surname:		Date:	
Assessor Signature:			

Moderator name and surname:		Date:	
Moderator Signature:			

Note to learner
Review the feedback provided by your lecturer to check that you have been found competent in this assessment. If there are any areas where you have been found not yet competent, you must redo those parts of the assessment and resubmit within the stipulated time frame. The section below will only be completed in cases where the learner was asked to resubmit parts of the assessment where they were found not yet competent.

General feedback to learner (Attempt 2) Supply comprehensive feedback why learner is found NYC

Learner Number:			
Learner name and surname:		Date:	
Learner Signature:			
Lecturer name and surname:		Date:	
Lecturer Signature:			

Assessor name and surname:		Date:	
Assessor Signature:			
Moderator name and surname:		Date:	
Moderator Signature:			