Context:

This document contains assignments to be completed as part of the hands-on for the DB Advanced day1 topic.

Assignment 1: Declaring variables

Estimated time: 10 Mins

Problem description: Table tblStudent is given in below diagram:

Column Name	Data Type	Allow Nulls
id	int	☐
StudentName	varchar(50)	☐
Age	int	☐
ClassTeacherId	int	☐

Declare a variable @name of type varchar(50)
Select student name into the variable from tblStudent for a student id.
Print the variable

How can you declare the above code within a batch?
Hint: Use GO keyword

Can you access the variable @name outside the batch?

Assignment 2: SQL query

Estimated time: 25 Mins

Objective:

- How to write complex queries in SQLServer

Problem description: Given table with the following attributes

students		
STUDENT_NO	SURNAME	FORENAME
20060101	Dickens	Charles
20060102	ApGwilym	Dafydd
20060103	Zola	Emile
20060104	Mann	Thomas
20060105	Stevenson	Robert

modules	
MODULE_CODE	MODULE_NAME
CM0001	Databases
CM0003	Operating Systems
CM0004	Graphics

marks		
STUDENT_NO	MODULE_CODE	MARK
20060101	CM0001	80
20060101	CM0002	65
20060101	CM0003	50
20060102	CM0001	75
20060102	CM0003	45
20060102	CM0004	70
20060103	CM0001	60

20060103	CM0002	75
20060103	CM0004	60
20060104	CM0001	55
20060104	CM0002	40
20060104	CM0003	45
20060105	CM0001	55
20060105	CM0002	50
20060105	CM0004	65

Execute following statements in SSMS:

```
create table students ( student_no varchar(10), surname varchar(20), forename varchar(20));

create table modules ( module_code varchar(8), module_name varchar(25));
create table marks ( student_no varchar(10), module_code varchar(8), mark integer);

insert into students values ('20060101','Dickens','Charles');
insert into students values ('20060102','ApGwilym','Dafydd');
insert into students values ('20060103','Zola','Emile');
insert into students values ('20060104','Mann','Thomas');
insert into students values ('20060105','Stevenson','Robert');

insert into modules values ('CM0001', 'Databases');
insert into modules values ('CM0002', 'Programming Languages');
insert into modules values ('CM0003', 'Operating Systems');
insert into modules values ('CM0004', 'Graphics');

insert into marks values ('20060101', 'CM0001', 80);
insert into marks values ('20060101', 'CM0002', 65);
insert into marks values ('20060101', 'CM0003', 50);
insert into marks values ('20060102', 'CM0001', 75);
insert into marks values ('20060102', 'CM0003', 45);
insert into marks values ('20060102', 'CM0004', 70);
insert into marks values ('20060103', 'CM0001', 60);
insert into marks values ('20060103', 'CM0002', 75);
insert into marks values ('20060103', 'CM0004', 60);
insert into marks values ('20060104', 'CM0001', 55);
insert into marks values ('20060104', 'CM0002', 40);
insert into marks values ('20060104', 'CM0003', 45);
```

```
insert into marks values ('20060105', 'CM0001', 55);
insert into marks values ('20060105', 'CM0002', 50);
insert into marks values ('20060105', 'CM0004', 65);
```

Write SQL statement for following query:

1. Increase the marks by 5 for the module operating system.

2. Update the surname of the students by adding the text “mind” before their surname. Add only the string “mind”, if it has a null value

3. Delete all the records from the database whose surname ends with the letter ‘n’

4. Fetch top 3 scores from table marks

5. Write a query which prints score of each student in each module in percentage format.

For example:

STUDENT_NO	MODULE_CODE	MARK
20060101	CM0001	80%
20060101	CM0002	65%

Calculate the percentage based on following formula: (Marks/100)*100

Assignment 3: Stored procedure

Estimated time: 40 Mins

Objective:

- How to write stored procedures in SQL Server

Problem description: Create following table called 'employee'

```
create table employee (
empId int primary key,
empName varchar(20),
empSalary int,
grade char(1)
);
```

1. Create stored procedure to perform following operation on employee table:
    a. Return name from EMP table for given employee id
    b. Delete the employee id from the table
2. Create a stored procedure called 'sp_Get_Grade' which selects grade from EMP table for given employee id. If grade not found then assign character 'X' to grade.
3. Create a stored procedure called 'sp_Check_Grade' which invokes above stored procedure. If returned grade value is equal to 'X' then print 'Grade not found' else print the retrieved value
4. Create a stored procedure which increments salary of all employees by 15%

Assignment 4: Working with inbuilt functions

Estimated time: 10 Mins

Objective:

- How to use inbuilt functions

Problem description: Refer the 'employee' table created for previous question.

Create a stored procedure which takes delimited string in the below form as input parameter : empId|empName|empSalary|grade|

eg 1003|John|11000|A

The procedure retrieves column values from the delimited string and inserts it as a row to the table

Assignment 5: User-defined functions

Estimated time: 20 Mins

Objective:

How to write functions in SQL Server

Problem description: Refer the 'employee' table created for previous question.

1. Write a function to return average salary of all the employees
2. Write function which accepts empid and return an integer as explained below
    - 1, if grade is 'M'
    - 2, if grade is 'L'
    - 3, if grade is 'T'
    - 4, if grade is 'X'
    - 0, for any exception

Assignment 6: Transactions

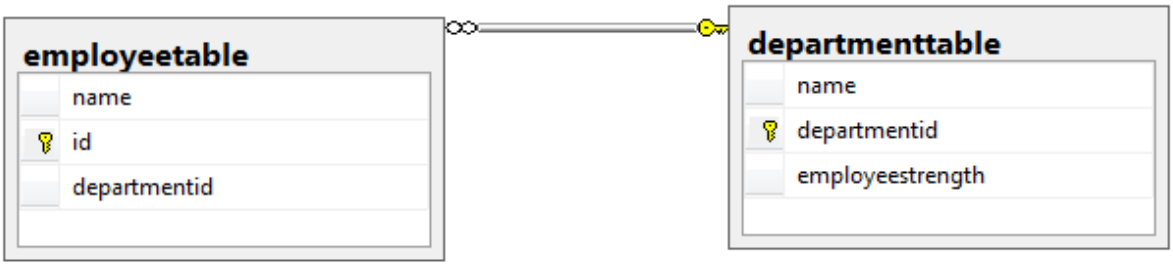
Estimated time: 30 Mins

Objective:

How to work with transactions in SQL Server

Problem description:

Create following two tables:



Above two tables have a foreign key association between “departmentid” column of “departmenttable” and “departmentid” column of “employeetable”.

Write a stored procedure as given below:

Input parameters: id , new department id

Create a transaction within the stored procedure to move the employee with given id to new department id. The department table should get updated for the new employee strength for previous and new department id

The stored procedure returns 1 if the transaction is successful else 0

Assignment 7: SQL query

Estimated time: 20 Mins

Objective:

- How to write complex queries in SQL Server

Problem description:

Refer the following tables:

Category

ID	CategoryName
1	Clothes
2	Kitchen
3	Sports
4	Computer

SubCategory

SubCatID	SubCategoryName	CatID
1	ChildClothes	1
2	Mens Wear	1
3	CookWare	2
4	Dinner	2
5	Football	3
6	Cricket	3

Product

PID	ProductName	SalesAmount	SubCatId
1	winter dress	1000	1
2	summer dress	2000	1
3	night dress	1500	2
4	casuals	3000	2
5	spoons	1000	3
6	tea cups	2500	3
7	non-sticks	5000	3
8	pan	2000	3
9	Bat	1500	6
10	Ball	2000	6

1. Write a SQL query to get the results as show below

Total Sales of a product with result columns as (SubCategoryName,TotalSales)]

SubCategoryName	TotalSales
ChildClothes	3000
Men's Wear	4500
CookWare	10500
Cricket	3500

Dinner	0
Football	0

2. Write a SQL query to get the results as show below

[Total sales of a category with result column as (CategoryName,TotalSales)]

CategoryName	**TotalSales**
Clothes	7500
Kitchen	10500
Sports	3500
Computer	0

Assignment 8: SQL query

Estimated time: 20 Mins

Objective:

- How to write complex queries in MySQL

Problem description:

Refer the following tables:

Country table

CountryID	CountryName
1	India
2	Italy
3	Sri Lanka
4	Australia
5	South Africa

Player table

PlayerID	CountryID	PlayerName	RunsMade	WicketTaken
1	1	Dhoni	25	5
2	1	Tendulakar	100	15
3	2	John Walt	20	13
4	2	David K	500	16
5	2	Alice M	800	17
6	3	Surya	200	20
7	3	Mithun	600	30
8	4	Gale	80	6
9	4	Kwisk	90	9
10	4	Marget	95	8

Description:

Select the player name, country the player belongs to and runs made, where the runs made by the player is more than average runs of all players within his country

Refer the following output:

PlayerName	CountryName	RunsMade
Tendulakar	India	100
David K	Italy	500
Alice M	Italy	800
Mithun	Sri Lanka	600
Kwisk	Australia	90
Marget	Australia	95

Assignment 9: SQL query

Estimated time: 30 Mins

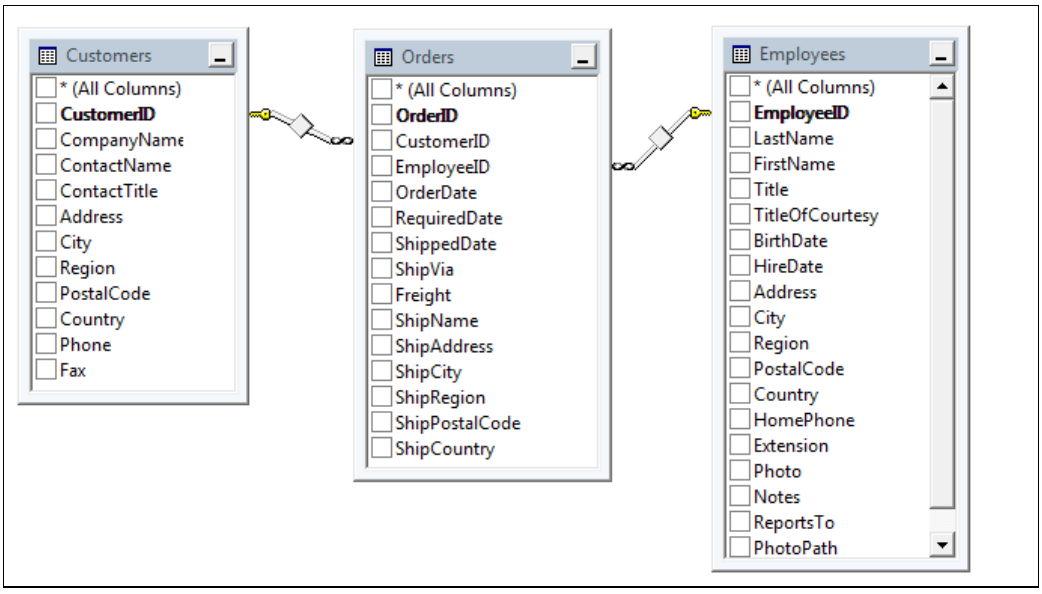
Objective:

- How to write complex queries in SQL Server

Problem description:

Create a report showing the Order ID, the name of the company that placed the order, and the first and last name of the associated employee. Only show orders placed after January 1, 1998 that shipped after they were required. Sort by Company Name.

The structure of the tables is as follows:



Summary of assignments:

You have learnt

- Concept of batch and local variables
- How to write complex queries in SQLServer?
- How to write stored procedures?
- How to process table data inside stored procedures?
- How to write functions in SQLServer?