

*Advanced SQL*

Training Assignments

|  |  |
| --- | --- |
| Program Code | ASQL |
| Version | 3.1 |
| Effective Date | 01/11/2016 |

**Hanoi, 11/2016**

RECORD OF CHANGES

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Changes | A\* M, D | Contents | Version |
| 14-Oct-2016 | Create | A | Add the new assignments. | v1.0 |
| 14-Oct-2018 | Update | M | Template. | v1.1 |
| 01-Jun-2019 | Update | M | Update Objective | v1.2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[For the following assignments: 4](#_Toc17276629)

[Day 2. Unit 2: Common Database Objects 4](#_Toc17276630)

[Assignment 2\_Opt2: Employee Management System 4](#_Toc17276631)

|  |  |
| --- | --- |
|  | **CODE: ASQL\_Assignment2\_Opt2**  **TYPE: Medium**  **LOC: n/a**  **DURATION: 180 MINUTES** |

# For the following assignments:

* Print out respectively the screenshots to show the query results.
* Pack screenshots and SQL scripts or your answers into the zip file named ASQL\_Assignment2\_AccountName.zip (for instance: ASQL\_Assignment2\_NamNT.zip) then handle to the evaluator via email ([XYZ@fsoft.com.vn](mailto:XYZ@fsoft.com.vn) ) or follow the guidance of the class admin.

# Day 2. Unit 2: Common Database Objects

## Assignment 2\_Opt2: Employee Management System

**Barems**: a - 15%, b - 15%, c - 15%, d - 15%, e - 15%, f – 25%

**Objectives**: H5SD - SQL skills

**Problem Descriptions**:

Login SQL Server and create database name EMS, run below script to generate tables:

CREATE TABLE [dbo].[Employee](

[EmpNo] [int] NOT NULL,

[EmpName] [nchar](30) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NOT NULL,

[BirthDay] [datetime] NOT NULL,

[DeptNo] [int] NOT NULL,

[MgrNo] [int] NOT NULL,

[StartDate] [datetime] NOT NULL,

[Salary] [money] NOT NULL,

[Status] [int] NOT NULL,

[Note] [nchar](100) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NULL,

[Level] [int] NOT NULL

) ON [PRIMARY]

GO

ALTER TABLE Employee

ADD CONSTRAINT PK\_Emp PRIMARY KEY (EmpNo)

GO

ALTER TABLE [dbo].[Employee] ADD CONSTRAINT [chk\_Level] CHECK (([Level]=(7) OR [Level]=(6) OR [Level]=(5) OR [Level]=(4) OR [Level]=(3) OR [Level]=(2) OR [Level]=(1)))

GO

ALTER TABLE [dbo].[Employee] ADD CONSTRAINT [chk\_Status] CHECK (([Status]=(2) OR [Status]=(1) OR [Status]=(0)))

GO

ALTER TABLE [dbo].[Employee]

ADD Email NCHAR(30)

GO

ALTER TABLE [dbo].[Employee]

ADD CONSTRAINT chk\_Email CHECK (Email IS NOT NULL)

GO

ALTER TABLE [dbo].[Employee] ADD CONSTRAINT chk\_Email1 UNIQUE(Email)

GO

ALTER TABLE Employee

ADD CONSTRAINT DF\_EmpNo DEFAULT 0 FOR EmpNo

GO

ALTER TABLE Employee

ADD CONSTRAINT DF\_Status DEFAULT 0 FOR Status

GO

CREATE TABLE [dbo].[Skill](

[SkillNo] [int] IDENTITY(1,1) NOT NULL,

[SkillName] [nchar](30) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NOT NULL,

[Note] [nchar](100) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE Skill

ADD CONSTRAINT PK\_Skill PRIMARY KEY (SkillNo)

GO

CREATE TABLE [dbo].[Department](

[DeptNo] [int] IDENTITY(1,1) NOT NULL,

[DeptName] [nchar](30) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NOT NULL,

[Note] [nchar](100) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE Department

ADD CONSTRAINT PK\_Dept PRIMARY KEY (DeptNo)

GO

CREATE TABLE [dbo].[Emp\_Skill](

[SkillNo] [int] NOT NULL,

[EmpNo] [int] NOT NULL,

[SkillLevel] [int] NOT NULL,

[RegDate] [datetime] NOT NULL,

[Description] [nchar](100) COLLATE

SQL\_Latin1\_General\_CP1\_CI\_AS NULL

) ON [PRIMARY]

GO

ALTER TABLE Emp\_Skill

ADD CONSTRAINT PK\_Emp\_Skill PRIMARY KEY (SkillNo, EmpNo)

GO

ALTER TABLE Employee ADD CONSTRAINT [FK\_1] FOREIGN KEY([DeptNo])

REFERENCES Department (DeptNo)

**Question:**

1. Write SQL statements for following activities & print out respectively the screenshots to show test data (the table data that you create to test each query) & query results:
2. Add at least 8 records into each created tables.
3. Write a stored procedure (without parameter) to update employee level to 2 of the employees that has employee level = 1 and has been working at least three year ago (from starting date). Print out the number updated records.
4. Write a stored procedure (with EmpNo parameter) to print out employee’s name, employee’s email address and department’s name of employee that has been out.
5. Write a user function named Emp\_Tracking (with EmpNo parameter) that return the salary of the employee has been working.
6. Write the trigger(s) to prevent the case that the end user to input invalid employees information (level = 1 and salary >10.000.000).

**-- THE END --**