**SQL Server Queries**

/\* nth Highest \*/

USE [master];

GO

IF EXISTS(SELECT name FROM sys.databases WHERE NAME = N'empdb')

DROP DATABASE empdb;

GO

CREATE DATABASE empdb;

GO

USE [empdb];

GO

IF EXISTS(SELECT name FROM sys.tables WHERE NAME = N'Employee')

DROP TABLE dbo.Employee;

GO

CREATE TABLE dbo.Employee

(ID int, Name NVARCHAR(50), Salary Money);

GO

INSERT INTO dbo.Employee

VALUES

(1, 'A', 10000), --4th

(2, 'B', 8000), --5th

(3, 'C', 8000),

(4, 'D', 6000), --6th

(5, 'E', 6000),

(6, 'F', 6000),

(7, 'G', 5000), --7th

(8, 'H', 5000),

(9, 'I', 5000),

(10, 'J', 5000),

(11, 'K', 4000), --8th

(12, 'L', 4000),

(13, 'M', 3000), --9th

(14, 'N', 3000),

(15, 'O', 1000), --10th

(16, 'P', 14000), --2nd

(17, 'Q', 14000),

(18, 'R', 12000), --3rd

(19, 'S', 12000),

(20, 'T', 16000), --1st

(21, 'U', 16000),

(22, 'V', 16000),

(23, 'W', 14000),

(24, 'X', 12000),

(25, 'Y', 12000),

(26, 'Z', 10000);

GO

SELECT \* FROM Employee;

GO

-- 16000 --1st

-- 14000 --2nd

-- 12000 --3rd

-- 10000 --4th

-- 8000 --5th

-- 6000 --6th

-- 5000 --7th

-- 4000 --8th

-- 3000 --9th

-- 1000 --10th

-- nth Highest Salary

-- n = 5

SELECT TOP 1 Salary FROM

(

SELECT DISTINCT TOP 5 Salary FROM Employee

ORDER BY Salary DESC

)

A ORDER BY Salary;

GO

--Better

;WITH CTE AS

(

SELECT Salary, rnk = DENSE\_RANK() OVER (ORDER BY Salary DESC) FROM Employee

)

SELECT DISTINCT Salary FROM CTE WHERE rnk = 5;

GO

--Best

--TOP 1 just grabs one of them, while DISTINCT does imply a sort and therefore might be more costly without any obvious benefit

;WITH CTE AS

(

SELECT Salary, rnk = DENSE\_RANK() OVER (ORDER BY Salary DESC) FROM Employee

)

SELECT TOP 1 Salary FROM CTE WHERE rnk = 5;

GO

/\* Delete Duplicate records \*/

USE [master];

GO

IF EXISTS(SELECT name FROM sys.databases WHERE NAME = N'temporarydb')

DROP DATABASE temporarydb;

GO

CREATE DATABASE temporarydb;

GO

USE [temporarydb];

GO

IF EXISTS(SELECT name FROM sys.tables WHERE NAME = N'DuplicateRcordTable')

DROP TABLE DuplicateRcordTable;

GO

/\* Create Table with 7 entries - 3 are duplicate entries \*/

CREATE TABLE DuplicateRcordTable (Col1 INT, Col2 INT)

INSERT INTO DuplicateRcordTable

SELECT 1, 1

UNION ALL

SELECT 1, 1 --duplicate

UNION ALL

SELECT 1, 1 --duplicate

UNION ALL

SELECT 1, 2

UNION ALL

SELECT 1, 2 --duplicate

UNION ALL

SELECT 1, 3

UNION ALL

SELECT 1, 4

GO

/\* It should give you 7 rows \*/

SELECT \* FROM DuplicateRcordTable;

GO

/\* Delete Duplicate records \*/

WITH CTE (COl1,Col2, DuplicateCount)

AS

(

SELECT COl1,Col2,

ROW\_NUMBER() OVER(PARTITION BY COl1,Col2 ORDER BY Col1) AS DuplicateCount

FROM DuplicateRcordTable

)

DELETE

FROM CTE

WHERE DuplicateCount > 1

GO

/\* It should give you Distinct 4 records \*/

SELECT \* FROM DuplicateRcordTable;

GO

/\* Check if database or table exists \*/

USE [master];

IF EXISTS (SELECT \* FROM sys.databases WHERE name = N'Testdb')

DROP DATABASE [Testdb];

GO

IF DB\_ID (N'Testdb') IS NOT NULL

DROP DATABASE [Testdb];

GO

CREATE DATABASE [Testdb];

GO

USE [Testdb];

GO

IF OBJECT\_ID (N'Test', N'U') IS NOT NULL

DROP TABLE [Test];

GO

IF EXISTS(SELECT name FROM sys.databases WHERE name = N'Test')

DROP TABLE [Test];

IF EXISTS(SELECT name FROM sys.tables WHERE name = N'Test')

DROP TABLE [Test];

GO