

Afleveringsopgave SQL Transaktioner og Normalformer

Programkode:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;

namespace SQL_Transaktion_Department
{
    class Program
    {
        static void Main(string[] args)
        {
            string myConnectionString = @"User id =DESKTOP-A7EQPC0\Patrick;

            " +
            "Password=; " +
            "Persist Security Info=False;" +
            "Integrated Security=SSPI;" +
            "Database=SQL Transaktioner;" +
            "Server=DESKTOP-A7EQPC0;" +
            "Connect Timeout=40";

            SqlConnection conn = new SqlConnection(myConnectionString);

            try
            {
                conn.Open();
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }

            Console.WriteLine("Read and Update");
            Console.ReadLine();

            //SET TRANSACTION ISOLATION LEVEL
```

Afleveringsopgave

SQL Transaktioner og Normalformer

```
//READ UNCOMMITTED
//READ COMMITTED
//REPEATABLE READ
//SNAPSHOT
//SERIALIZABLE

// Set Transaction ISOLATION LEVEL
String sSQL = "SET TRANSACTION ISOLATION LEVEL READ
COMMITTED";
Console.WriteLine(sSQL);
SqlCommand command = new SqlCommand(sSQL, conn);
command.ExecuteNonQuery();
Console.ReadLine();

//Begin Transaction
Console.WriteLine("BEGIN TRANSACTION");
sSQL = "BEGIN TRANSACTION";
command = new SqlCommand(sSQL, conn);
command.ExecuteNonQuery();
Console.ReadLine();

//Read
sSQL = "SELECT * FROM Department;";
command = new SqlCommand(sSQL, conn);
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
    Console.WriteLine("{0}\t\t{1}\t\t", reader.GetInt32(0),
reader.GetString(1));
reader.Close();

// Update
Console.WriteLine("UPDATE Department, Write new name for
department Nr. 1.");
string newName = Console.ReadLine();
sSQL = "UPDATE Department";
sSQL += " SET name =" + newName + "'";
sSQL += " WHERE dno = 1;";
command = new SqlCommand(sSQL, conn);
command.ExecuteNonQuery();

// Update
Console.WriteLine("UPDATE department, Write new name for
```

Afleveringsopgave SQL Transaktioner og Normalformer

```
department Nr. 2.");
    newName = Console.ReadLine();
    sSQL = "UPDATE Department";
    sSQL += " SET name ='" + newName + "'";
    sSQL += " WHERE dno = 2;";
    command = new SqlCommand(sSQL, conn);
    command.ExecuteNonQuery();

    Console.WriteLine("Hit Enter to Commit.");
    Console.ReadLine();
    // Commit or Rollback
    Console.WriteLine("COMMIT");
    sSQL = "COMMIT"; // or ROLLBACK
    command = new SqlCommand(sSQL, conn);
    command.ExecuteNonQuery();

    try
    {
        conn.Close();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
}
}
```

Kommentarer til programkode:

P1 = Serializable, P2 = Read Uncommitted

Kører man en update på P1 kan man godt lave en read på P2. P2 læser dog en uncommitted værdi. P2 kan ikke få lov til at lave en update hvis P1 er i gang med en transaction.

P1 = Serializable, P2 = Serializable

Kører man en update på P1 kan man ikke lave en read på P2. P2 venter til P1 er færdig med transaction.

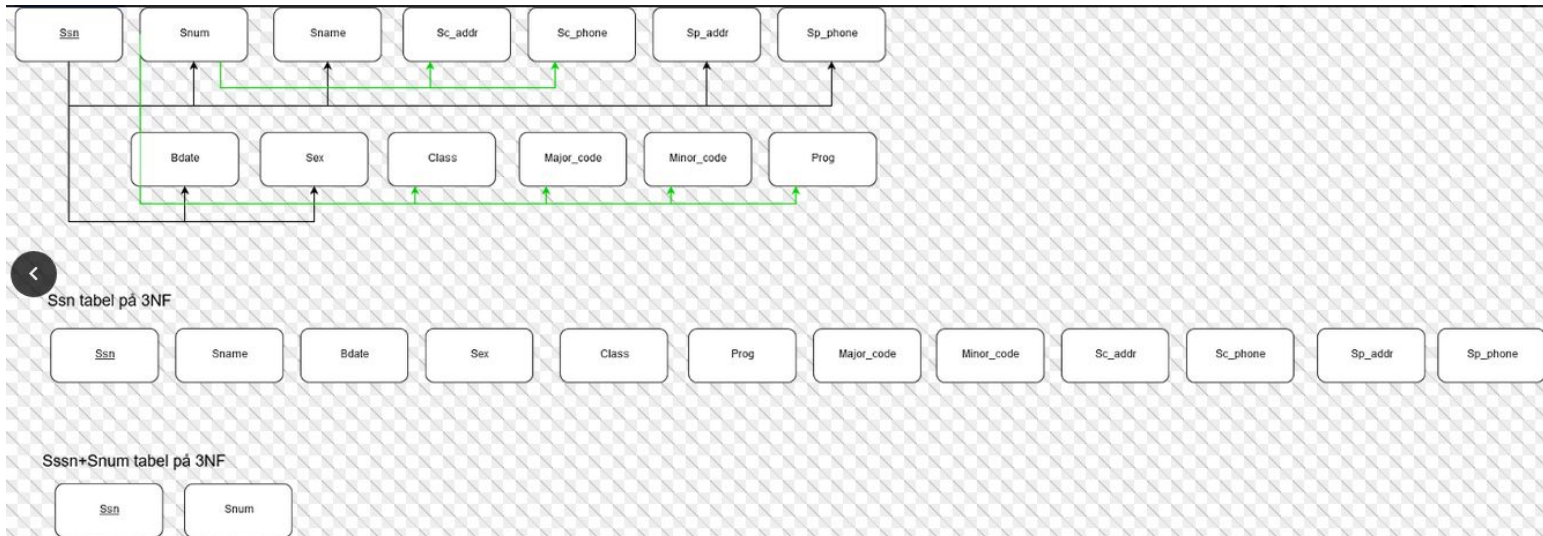
P1 = Serializable, P2 = Read Committed

P2 venter på at P1 committer før den får lov til at læse tabellen.

Afleveringsopgave

SQL Transaktioner og Normalformer

14.19:



14.20:

I a) Emp_Dept er der en update anomalitet pga. Dnumber. Hvis man ændrer Dnumber, er man nødt til også at ændre Dname og Dmgr_ssn. Jeg ville fjerne Dnumber og lave en {Ssn, Dnumber} tabel.

I b) Emp_Proj skaber Pnumber en update anomali, som ovenfor. Hvis Pnumber ændres, skal Pname ændres og muligvis Plocation.

14.21:

1. Z er PK.
2. Y er PK.
3. Y er PK.