# LærerVikaren 02

|  |  |
| --- | --- |
| **Læringsmål** | Du kan:   * **1Pf4:** anvende centrale metoder og teknikker til at realisere modeller i et databasesystem […] |
| **Forventet læringsudbytte [SOLO]** | Programmeringssporet:   * [Multistrukturel] Du kan beskrive flere kendetegn ved **database-design** * [Multistrukturel] Du kan beskrive flere kendetegn ved **database-implementering (SQL)** |
| **Din forberedelse** | Programmeringssporet:   * ’Kom i gang med database.FORELØBIG.v3.docx’ (fase B i del 1) * SQL:   + [ALTER TABLE](https://www.w3schools.com/sql/sql_alter.asp)   + [Named Constraints](https://www.youtube.com/watch?v=3_yro1G_4oc) (video: 3:41)   + [What is the purpose of constraint naming](https://stackoverflow.com/questions/1397440/what-is-the-purpose-of-constraint-naming) (se specielt svar fra Philip Kelley) * SQL Joins:   + [Joining tables](https://www.linkedin.com/learning/programming-foundations-databases-2/joining-tables) (video: 3:20)   + [SQL Join Overview](https://4.bp.blogspot.com/-_HsHikmChBI/VmQGJjLKgyI/AAAAAAAAEPw/JaLnV0bsbEo/s1600/sql%2Bjoins%2Bguide%2Band%2Bsyntax.jpg)   + [SQL Join](https://www.w3schools.com/sql/sql_join.asp)   + [SQL Inner Join](https://www.w3schools.com/sql/sql_join_inner.asp)   + [SQL Left Join](https://www.w3schools.com/sql/sql_join_left.asp)   + [SQL Right Join](https://www.w3schools.com/sql/sql_join_right.asp)   + [SQL Full Join](https://www.w3schools.com/sql/sql_join_full.asp)   + [SQL Self Join](https://www.w3schools.com/sql/sql_join_self.asp)   **Løs følgende øvelser fra SQLBolt:**   * [SQL Lesson 4: Filtering and sorting Query results](https://sqlbolt.com/lesson/filtering_sorting_query_results) * [SQL Lesson 6: Multi-table queries with JOINs](https://sqlbolt.com/lesson/select_queries_with_joins) * [SQL Lesson 7: OUTER JOINs](https://sqlbolt.com/lesson/select_queries_with_outer_joins) * Prøv denne interaktive demo: <https://joins.spathon.com/> |

Denne opgave er den næste i en serie af opgaver, hvor du skal arbejde videre med database­temaet primært med udgangspunkt i LærerVikar-casen, som du er introduceret til i systemudvikling.   
Løs den forrige opgave Ex45-LærerVikaren01 for en udvalgt iteration (use case + domæneklasser), før du går i gang med denne opgave. På samme måde skal denne Ex46-opgave også løses, før du fortsætter med den næste opgave (Ex47).

Denne opgave har fokus på den fase B: **implementering af en SQL-database**:

* Fase A: Database-design (fra domænemodel til UML-databasemodel via normaliserede relationsskemaer)
* **Fase B: Database-implementering (fra UML-databasemodel til konkret database i et DBMS via SQL DDL)**
* Fase C: Database-tilgang via C# og .NET

## Dagens ord:



# Øvelse 1: Terminologi

Del teamet op i to mindre grupper, og brug **Ordet rundt** til at reflektere over begreberne ”ALTER TABLE”, ”named constraints” og ”Joins”.  
Sørg for, at alle får mulighed for at tale.

*Tidsramme: 20 minutter*

# Øvelse 2: Fase A: Database-design

Hvis du ikke er blevet færdig med fase A i forrige opgave Ex45-LærerVikaren01 (iterativt), da sørg for at gøre dette nu. Du skulle gerne have etableret en UML-databasemodel for en udvalgt use case og tilhørende domæneklasser, inden du fortsætter.

**Benyt parprogrammering i alle følgende øvelser.**

# Øvelse 3: Fase B: Implementering af SQL-database

Med udgangspunkt i din UML-databasemodel fra forrige opgave, skal du nu i fase B af database­udviklings­processen realisere dit databasedesign i et R-DBMS, i dit tilfælde konkret i Microsoft SQL Server, du tidligere har arbejdet med.

Denne fase involverer følgende trin:

1. Skriv SQL-sætninger til at:
   1. oprette tabellerne for din database med alle attributter og restriktioner (en: constraints)
   2. fylde (’seede’) (test)data i databasen
   3. hente data fra databasen
2. Opret din konkrete database i et DBMS (MS SQL Server)

Med fokus på den iterative tilgang, da gennemfør disse trin som angivet i de følgende underøvelser for en eller to udvalgte tabeller i database­modellen.

## Øvelse 3.1: SQL-sætninger til oprettelse og håndtering af database

Du er nu ved at være klar til at realisere din planlagte database i Microsoft SQL Server. Men først skal du skrive alle SQL-sætninger på tekstform til at oprette databasen.

Udfør følgende:

* Med din UML-databasemodel i hånden, skriv de nødvendige SQL-sætninger (CREATE TABLE) til at oprette en database for LærerVikaren, dvs. alle iterativt udvalgte tabeller, kolonner, nøgler og restriktioner. Sørg for at anvende **navngivne restriktioner** denne gang, så man kan referere til dem. Sørg også for, at du som minimum for oprettet mindst 2 tabeller med en 1:M-relation mellem dem (du får brug for det i øvelse 3.4).
* Videndel dine SQL-sætninger med de andre i teamet, så du sikrer dig, at alt er med, og alle forstår sætningerne.

## Øvelse 3.2: Realisering af database i DBMS

Du er nu endelig klar til at realisere din database.

Udfør følgende:

* Åbn nu SQL Server Management Studio (husk at sikre en VPN-forbindelse, hvis du ikke er på skolen)
* Eksekver dine SQL-sætninger fra øvelse 3.1
* Tjek, at alle dine tabeller er oprettet korrekt. Ret, hvis noget er galt.

Ønsker man at lave ændringer til tabeller i en database, kan man ’droppe’ (slette) de tabeller, man vil ændre, og oprette dem igen med ændret opsætning. Dette fungerer rimeligt nemt, hvis ens system ikke er gået i produktion endnu, eller man nemt kan genetablere data. I en kørende database med mange produktionsdata og en vis kompleksitet, kan det være problematisk at justere databasen ved at slette tabeller og oprette dem igen. I SQL er der med ’ALTER TABLE’ mulighed for at foretage visse ændringer uden, at man sletter tabellen.

* Brug ALTER TABLE til at rette en eksisterende tabel. Sørg for, at du får prøvet det af.

## Øvelse 3.3: Opret og afprøv testdata til database

Udfør følgende:

* Benyt INSERT INTO i Management Studio til at oprette passende testdata i jeres database. Tag f.eks. udgangspunkt i data fra case-beskrivelsen, eller find på jeres egne data til jeres forskellige tabeller.
* Udfør forskellige SELECT-sætninger for at hente data fra hver enkelt tabel i databasen.

## Øvelse 3.4: Hent data med SQL Joins

Udfør følgende:

* Udfør dernæst SELECT-sætninger, hvor du ved hjælp af SQL JOIN henter data fra to eller flere tabeller samtidigt med brug af primær- og fremmednøgler. Prøv de forskellige variationer er SQL JOINS for at se og forstå forskellen.

# Øvelse 4: Iterationer

Er du blevet færdig med den udvalgte iteration, da udfør ovenstående ’fase B’-øvelser for en ny tabel i UML-databasemodellen (evt. også for fase A for en ny use case og tilhørende domæneklasser, hvis der mangler tabeller). Gentag dette, så længe du har tid i dag.

Du har nu afsluttet fase B af databaseudviklingen for LærerVikaren. Næste fase C er at implementere systemets tilgang til databasen via C#, så du kan bruge disse data i din applikation. Dette ser du på i den næste opgave Ex47-LærerVikaren03.