https://www.halvorsen.blog

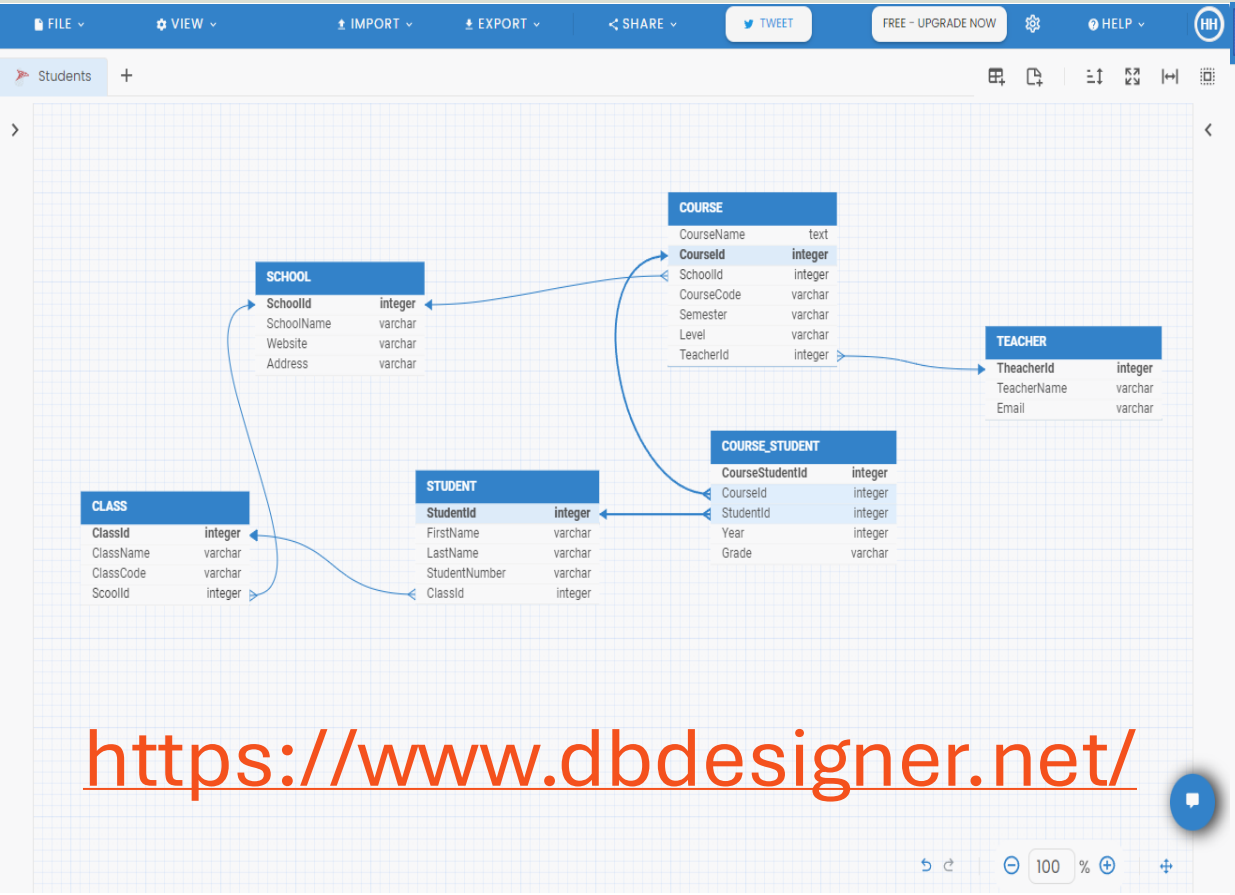
DB DESIGNER

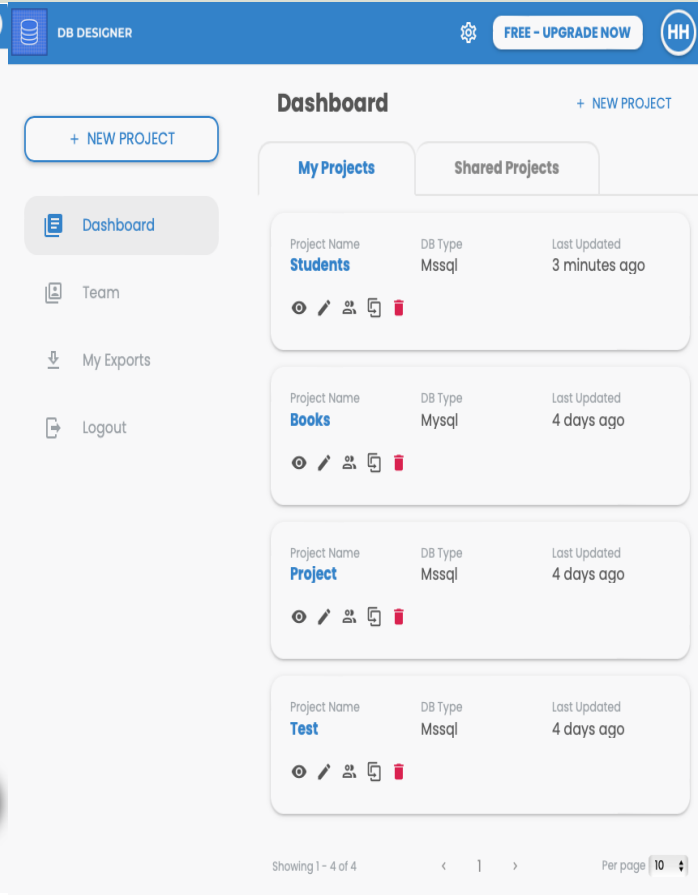
Database Design and Modelling using DB Designer

https://www.dbdesigner.net/

Hans-Petter Halvorsen

DB Designer





https://www.dbdesigner.net/

DB Designer

- DB Designer is an **online tool** for Database Design and Modelling for creating so-called ERDs, i.e., Entity Relationship Diagrams.
- You can **export** the ERD as **SQL Scripts** that supports many of the most popular database systems like SQL Server, MySQL, PostgreSQL, Oracle, etc.
- You can use it for free, but it also have paid options
- **Free for Academic use**, Non-profit and Open-Source projects.
- You can share and collaborate with others.
- Homepage: https://www.dbdesigner.net/

https://www.halvorsen.blog

Database Design Fundamentals

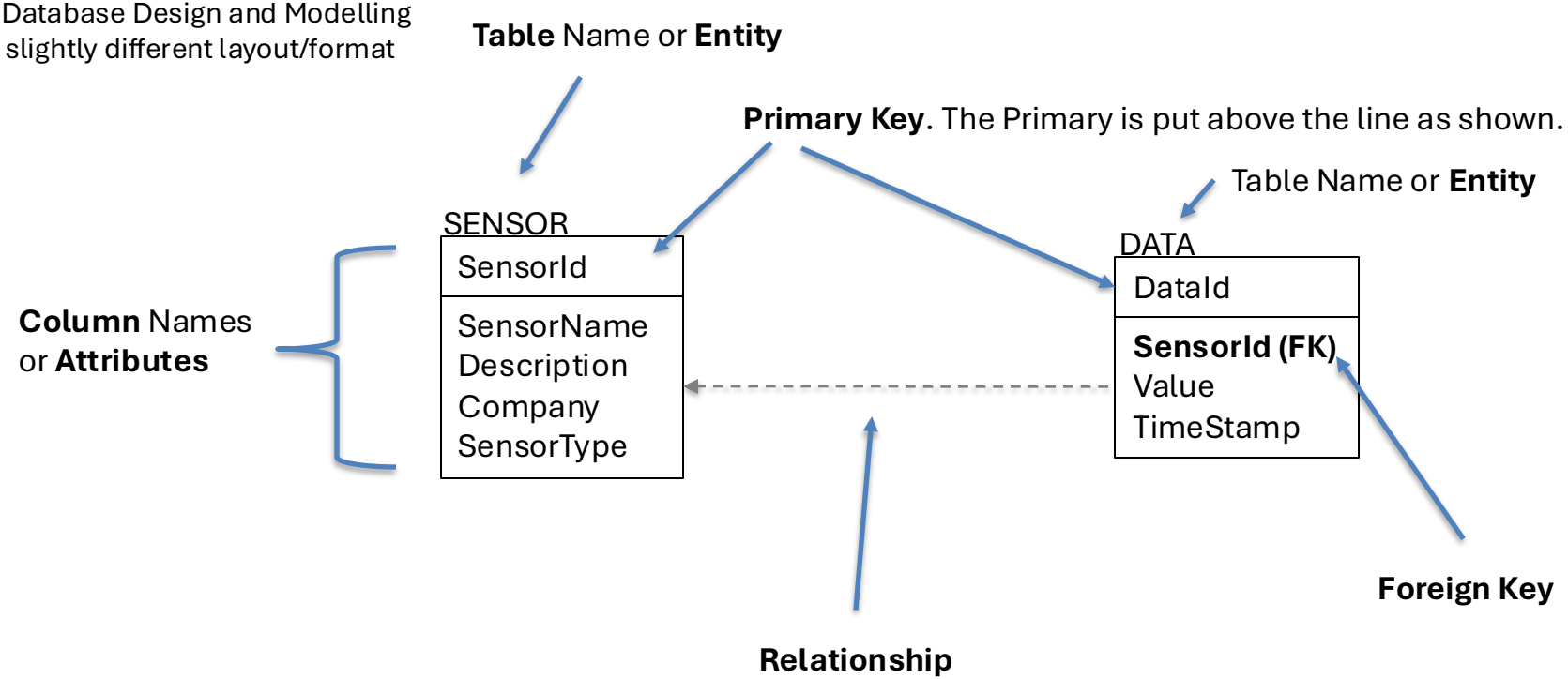
Hans-Petter Halvorsen

Basic Example

An **Entity Relationship Diagram** (**ERD** or just **ER** diagram) is a visual representation of a database that shows how the elements within are related. An ER diagram is made up of two object types, **entities** and **relationships**.

ERD Fundamentals

Each Database Design and Modelling has a slightly different layout/format



ERD – Entity Relationship Diagram

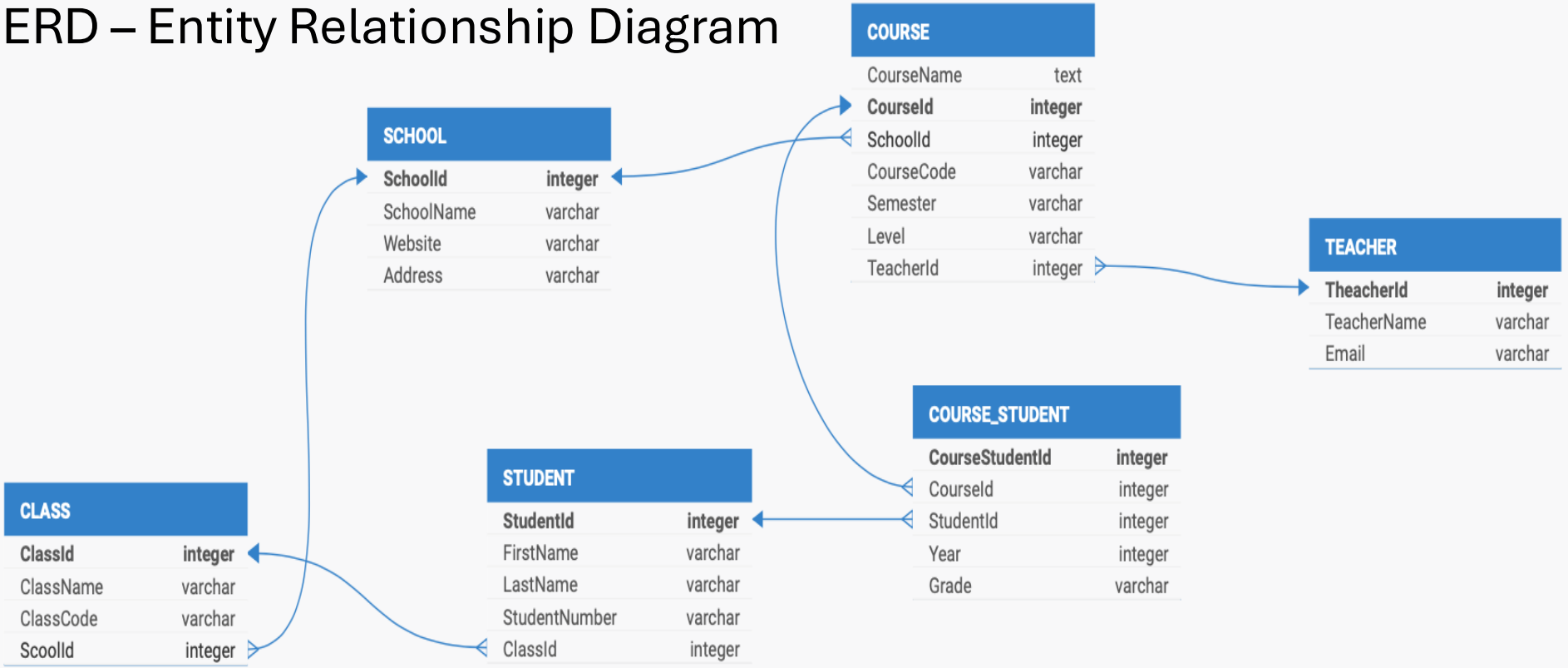
Relationships illustrate the association between two entities

Database Modelling Tools

- There exist hundreds of different tools for database design and modelling.
- Most of those cost a lot of money while others has some free alternatives.
- Some tools are specialized for Database Design and Modelling, etc.
- While other tools are just general-purpose diagram tools.
- The advantage with a specialized Database Design and Modelling tool is that you can generate SQL code for your tables.

ERD Example

ERD – Entity Relationship Diagram

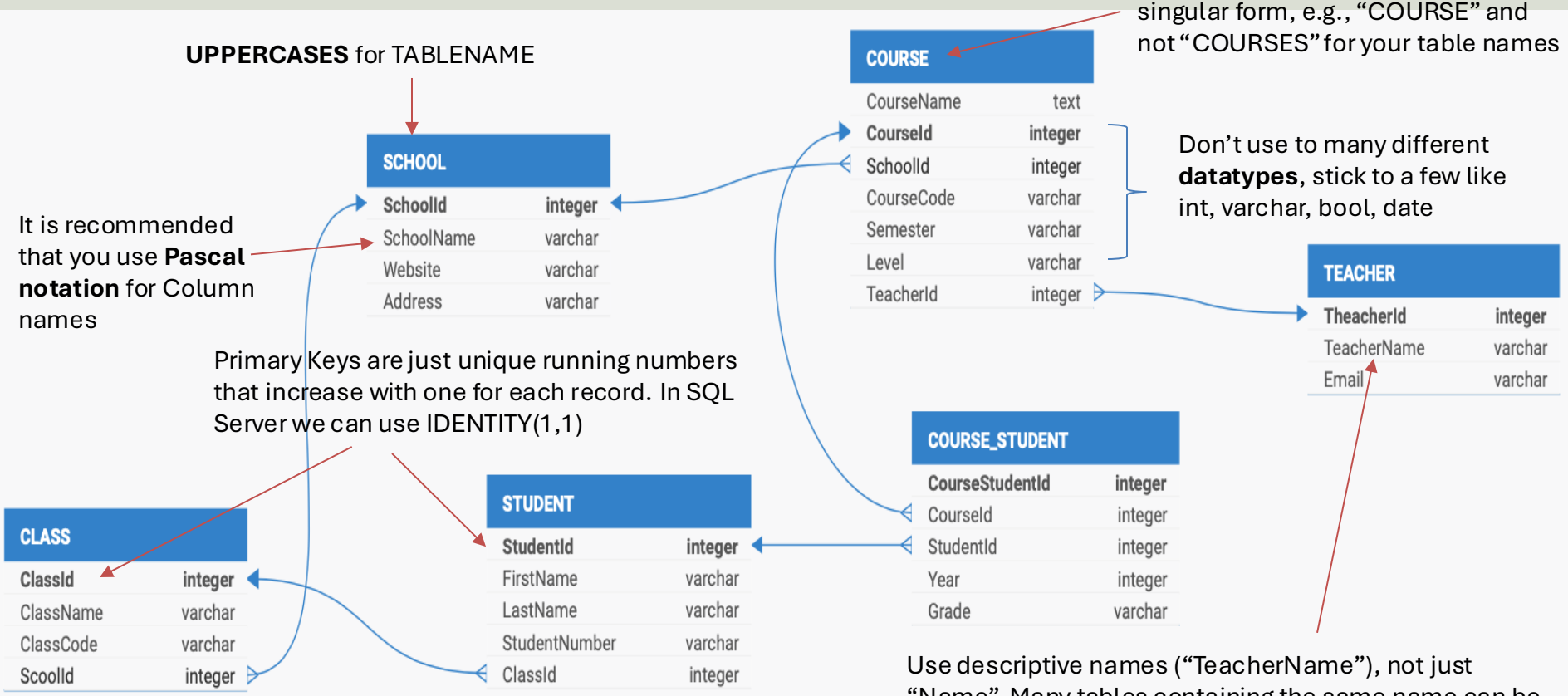


Created with DB Designer

Database Tips and Tricks

- It is recommended that you use **UPPERCASES** for TABLENAME.
- It is recommended that you use singular form, e.g., “CUSTOMER” and not “CUSTOMERS” for your table names.
- It is recommended that you use **Pascal notation** for Column names. Table names and column names should also be in English.
- I always prefer to use int for my **Primary keys** and that the Primary Keys are just numbers like 1, 2, 3, etc. In SQL Server you can use **IDENTITY(1,1)** or AUTO_INCREMENT in MySQL.
- Primary Key – Foreign Key relationships ensures that you don’t duplicate data, and you cannot fill in illegal data into the tables. In this case you can only use SensorIds that already exists in the SENSOR table.
- I like to use the same name for the Primary key and the Foreign Key, but it is not necessary.
- It might be useful to use “DirectorName”, “AgentName”, etc. instead of just “Name”, else you will have many tables containing the same name and that can be confusing and more difficult when making SQL queries.
- Stick to a few datatypes, like int, varchar(100), datetime and bit. Easier to remember when creating queries, Stored Procedures, etc,

ERD Example


It is recommended that you use singular form, e.g., “COURSE” and not “COURSES” for your table names
UPPERCASES for TABLENAME
COURSE
CourseName text
CourseId integer
SchoolId integer
CourseCode varchar
Semester varchar
Level varchar
TeacherId integer
Don’t use to many different datatypes, stick to a few like int, varchar, bool, date
SCHOOL
SchoolId integer
SchoolName varchar
Website varchar
Address varchar
It is recommended that you use Pascal notation for Column names
TEACHER
TheacherId integer
TeacherName varchar
Email varchar
Primary Keys are just unique running numbers that increase with one for each record. In SQL Server we can use IDENTITY(1,1)
COURSE_STUDENT
CourseStudentId integer
CourseId integer
StudentId integer
Year integer
Grade varchar
STUDENT
StudentId integer
FirstName varchar
LastName varchar
StudentNumber varchar
ClassId integer
CLASS
ClassId integer
ClassName varchar
ClassCode varchar
ScoolId integer
Use descriptive names (“TeacherName”), not just “Name”. Many tables containing the same name can be confusing and more difficult when making SQL queries

DB DESIGNER

DB Designer

https://www.dbdesigner.net/

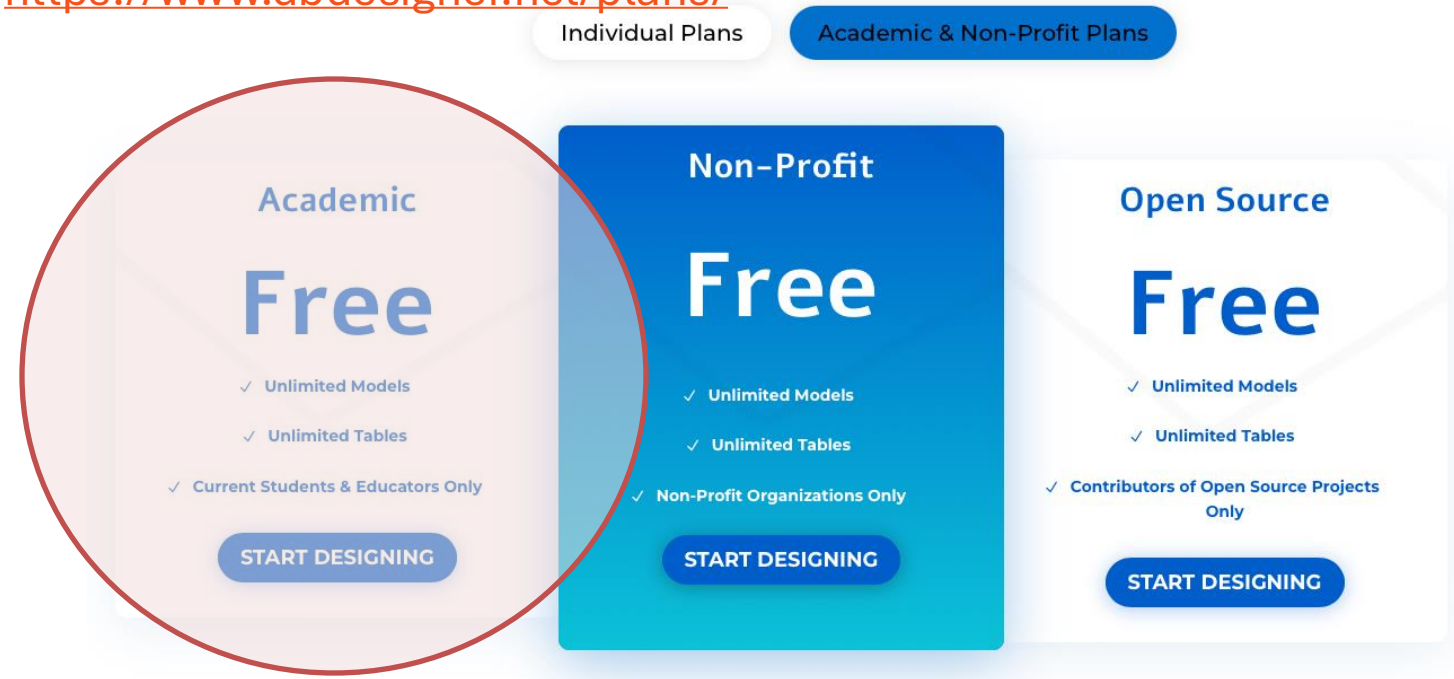
Hans-Petter Halvorsen

DB Designer

- DB Designer is an **online tool** for Database Design and Modelling for creating so-called ERDs, i.e., Entity Relationship Diagrams.
- You can **export** the ERD as **SQL Scripts** that supports many of the most popular database systems like SQL Server, MySQL, PostgreSQL, Oracle, etc.
- You can use it for free, but it also have paid options
- **Free for Academic use**, Non-profit and Open-Source projects.
- You can share and collaborate with others.
- Homepage: https://www.dbdesigner.net/

Free Plan for Academics and Non-Profit

https://www.dbdesigner.net/plans/



https://www.dbdesigner.net/opportunities-to-earn-free-upgrade-to-our-paid-plans/

Free Plan for Academics

Information given by DB Designer:

If you are a **Student**

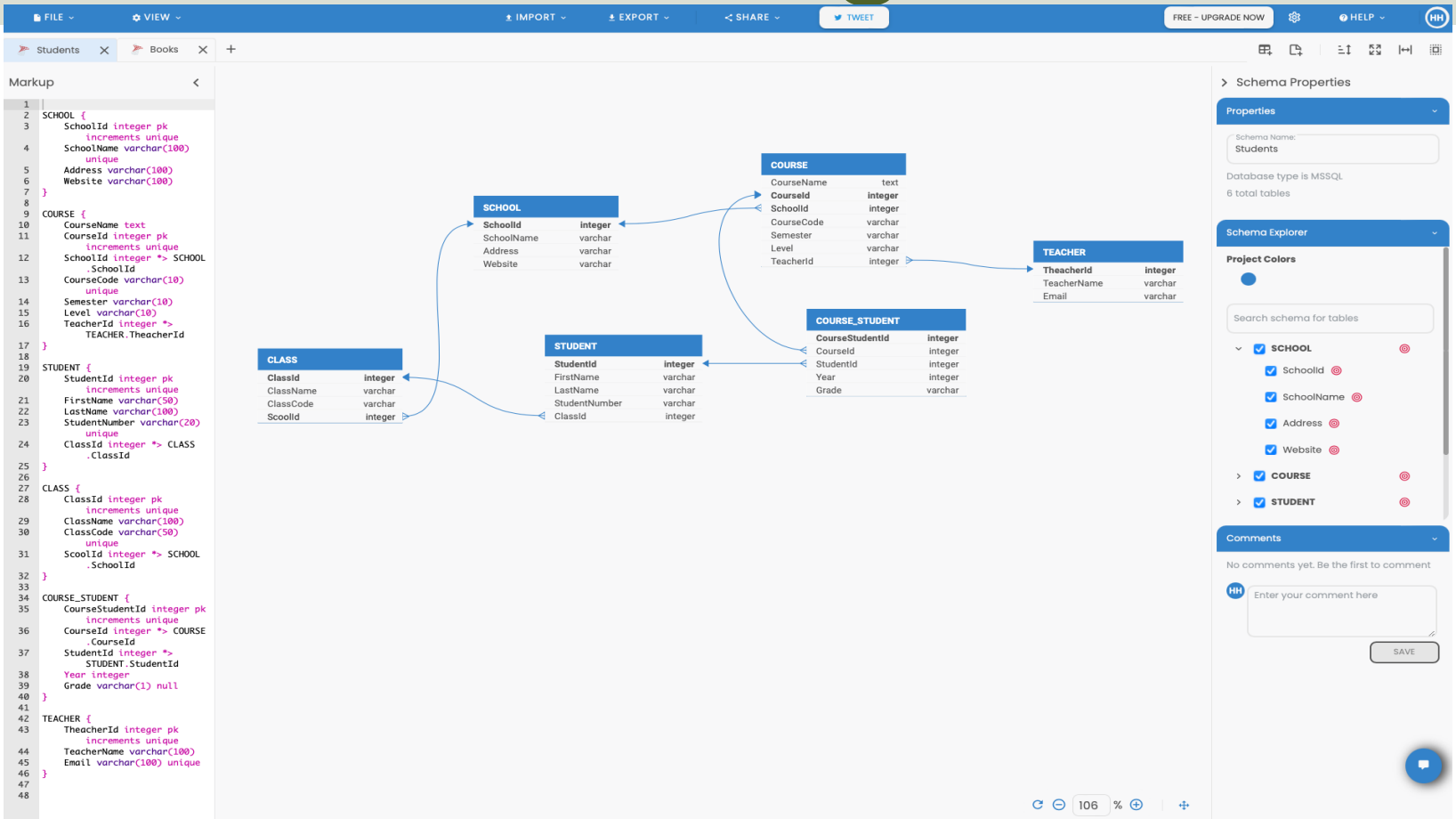
- If you are a student. Please register using **.edu** email and we'll upgrade you for free
- OR take a picture of your student ID and email us at **info@dbdesigner.net** with this **subject line "Student Request"** .
- We will upgrade your account plan absolutely free.

If you are a **Professor or Educator**

- Please contact us from your official email address requesting Educator access
- OR take a picture of your official staff ID and email us at **info@dbdesigner.net**.
- We will upgrade your account to "Unlimited" plan absolutely free.
- This way, you can always enjoy unlimited access to our application for free and you can encourage your students to request a student account from us for free.

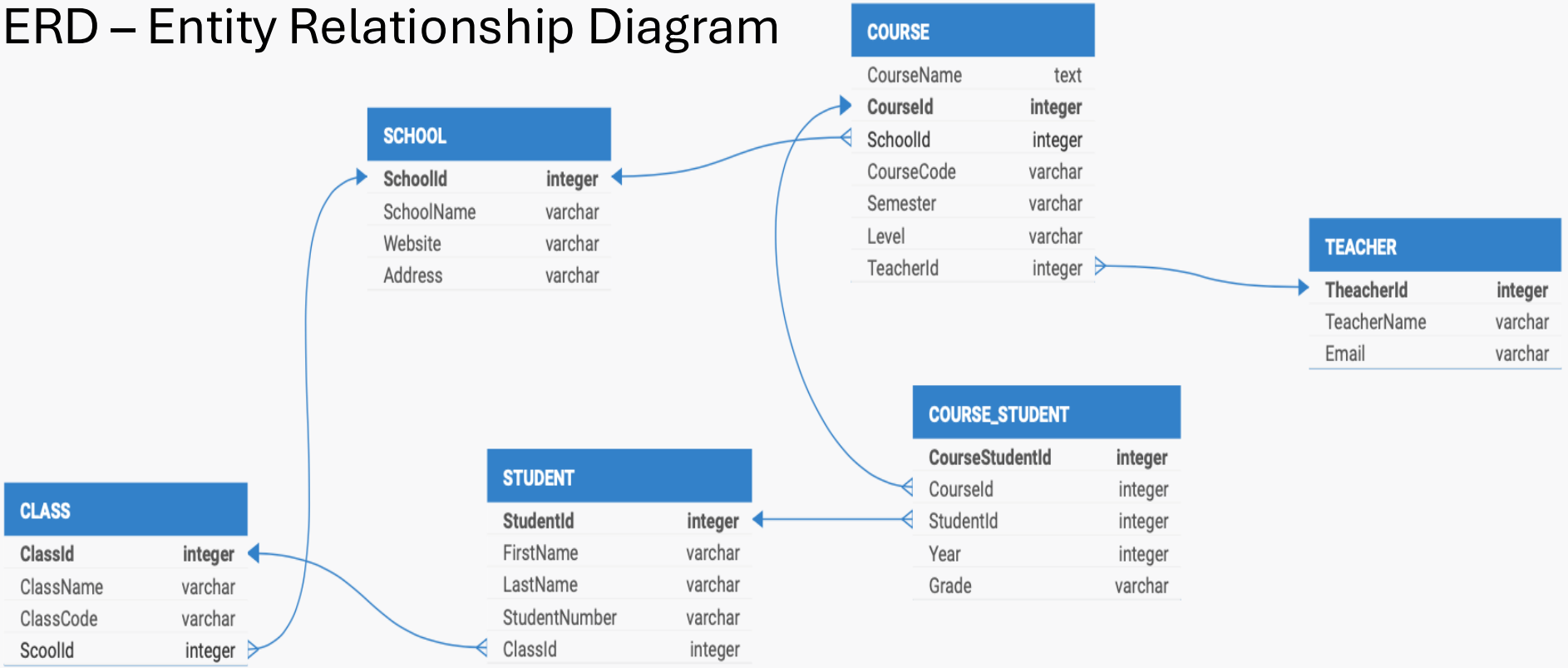
https://www.dbdesigner.net/opportunities-to-earn-free-upgrade-to-our-paid-plans/

DB Designer


FILE
VIEW
IMPORT
EXPORT
SHARE
TWEET
FREE - UPGRADE NOW
HELP
Students
Books
Markup
SCHOOL {
SchoolId integer pk increments unique
SchoolName varchar(100) unique
Address varchar(100)
Website varchar(100)
}
COURSE {
CourseName text
CourseId integer pk increments unique
SchoolId integer *> SCHOOL.SchoolId
CourseCode varchar(10) unique
Semester varchar(10)
Level varchar(10)
TeacherId integer *> TEACHER.TheacherId
}
STUDENT {
StudentId integer pk increments unique
FirstName varchar(50)
LastName varchar(100)
StudentNumber varchar(20) unique
ClassId integer *> CLASS.ClassId
}
CLASS {
ClassId integer pk increments unique
ClassName varchar(100)
ClassCode varchar(50) unique
ScoolId integer *> SCHOOL.SchoolId
}
COURSE_STUDENT {
CourseStudentId integer pk increments unique
CourseId integer *> COURSE.CourseId
StudentId integer *> STUDENT.StudentId
Year integer
Grade varchar(1) null
}
TEACHER {
TheacherId integer pk increments unique
TeacherName varchar(100)
Email varchar(100) unique
}
SCHOOL
SchoolId integer
SchoolName varchar
Address varchar
Website varchar
COURSE
CourseName text
CourseId integer
SchoolId integer
CourseCode varchar
Semester varchar
Level varchar
TeacherId integer
TEACHER
TheacherId integer
TeacherName varchar
Email varchar
CLASS
ClassId integer
ClassName varchar
ClassCode varchar
ScoolId integer
STUDENT
StudentId integer
FirstName varchar
LastName varchar
StudentNumber varchar
ClassId integer
COURSE_STUDENT
CourseStudentId integer
CourseId integer
StudentId integer
Year integer
Grade varchar
106 %
Schema Properties
Properties
Schema Name:
Students
Database type is MSSQL
6 total tables
Schema Explorer
Project Colors
Search schema for tables
SCHOOL
SchoolId
SchoolName
Address
Website
COURSE
STUDENT
Comments
No comments yet. Be the first to comment
Enter your comment here
SAVE

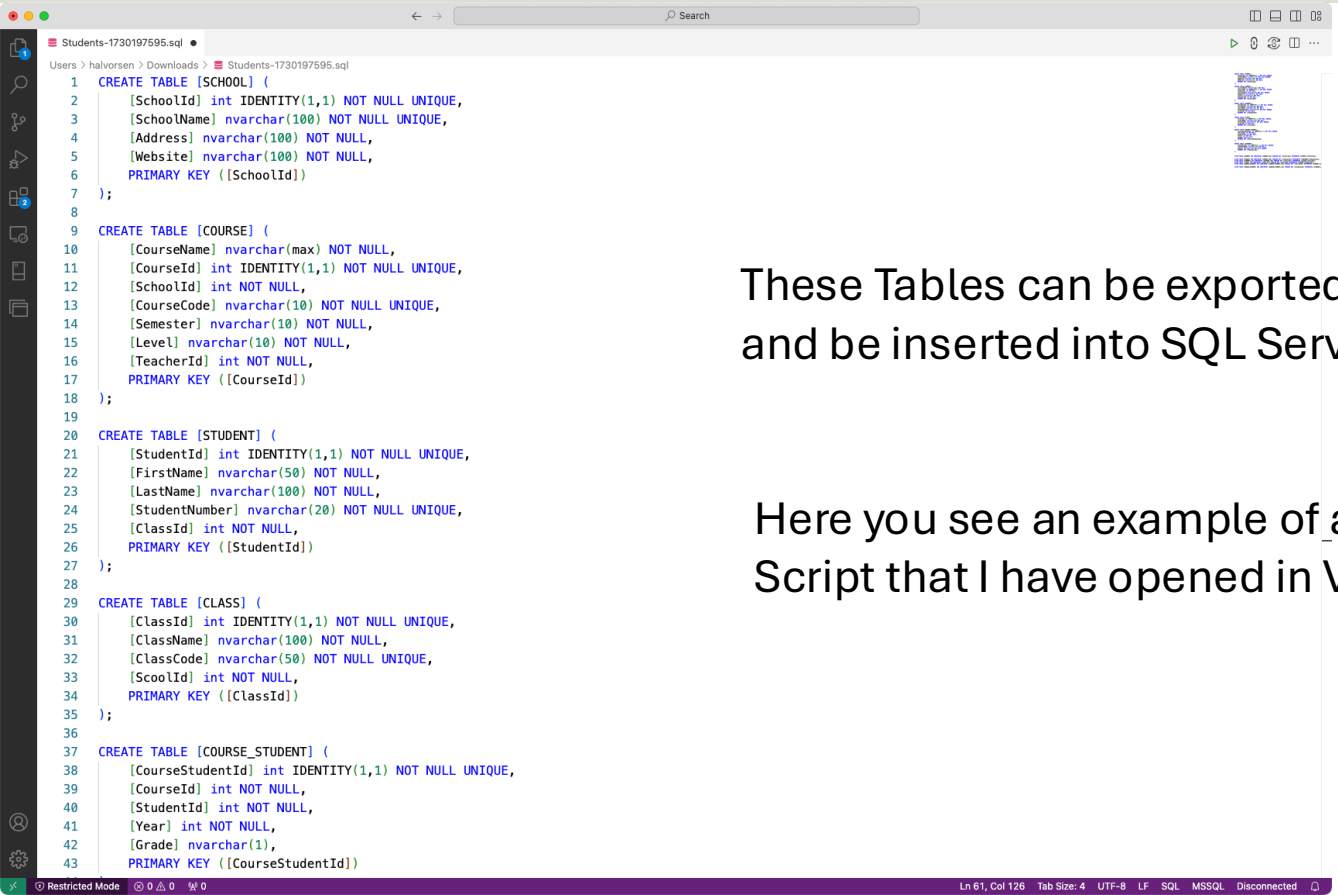
Let's create this ERD Example

ERD – Entity Relationship Diagram



Created with DB Designer

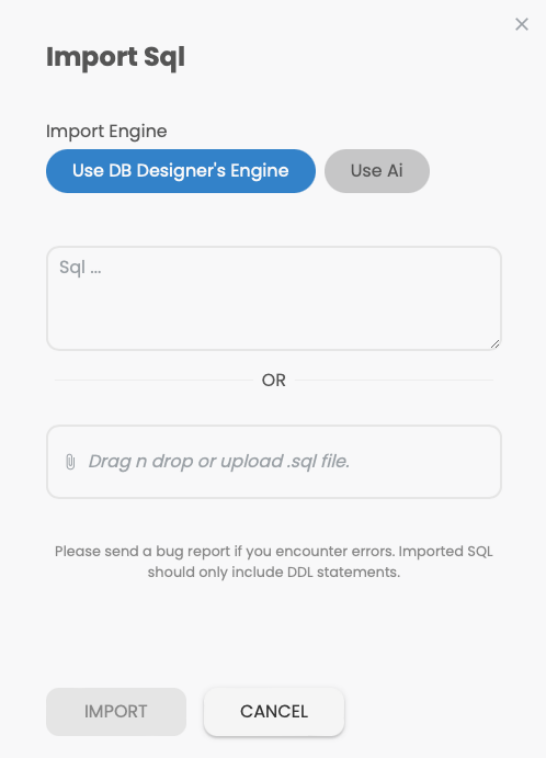
Export to SQL



These Tables can be exported as a SQL script and be inserted into SQL Server, MySQL, etc.

Here you see an example of a generated SQL Script that I have opened in Visual Studio Code

Import



You can also Import SQL Files and generate ERD based on that SQL Script

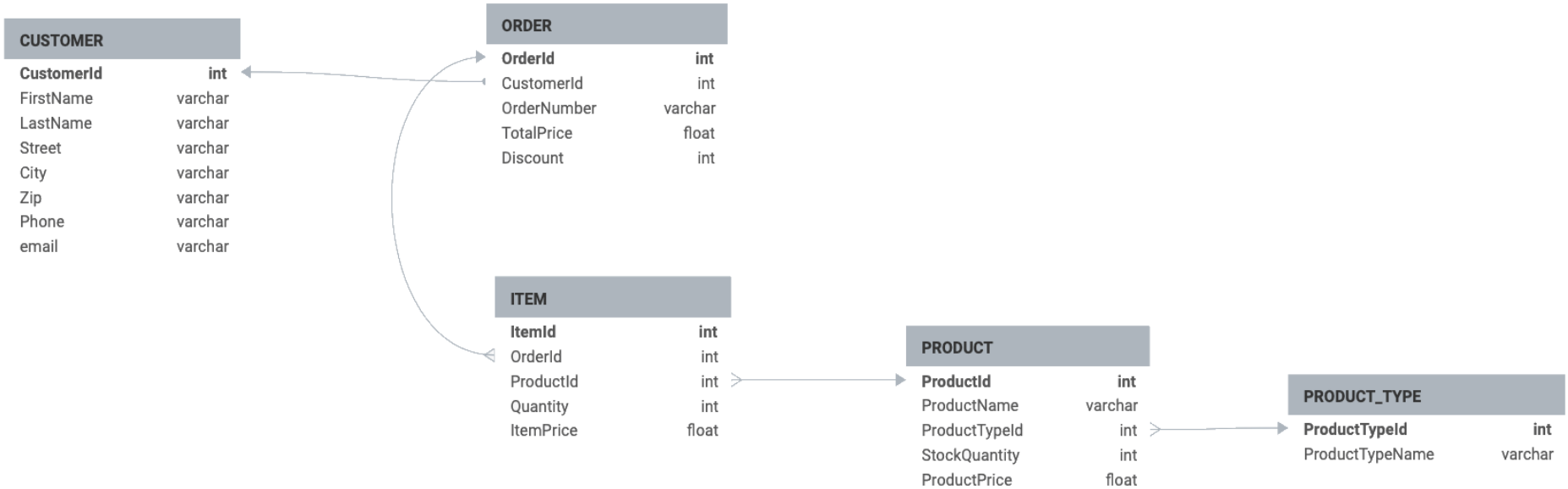
Order System Example

- Customers
- Products and Product Types
- Orders and Items in a specific Order
- ..

=> Can you make an ERD for such a system using DB Designer?

Order System Example

Here you see the final Order System with data types, etc.



These Tables can now be exported as a SQL script and be inserted into SQL Server

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog