

<https://www.halvorsen.blog>



DB DESIGNER

Database Design and Modelling using DB Designer

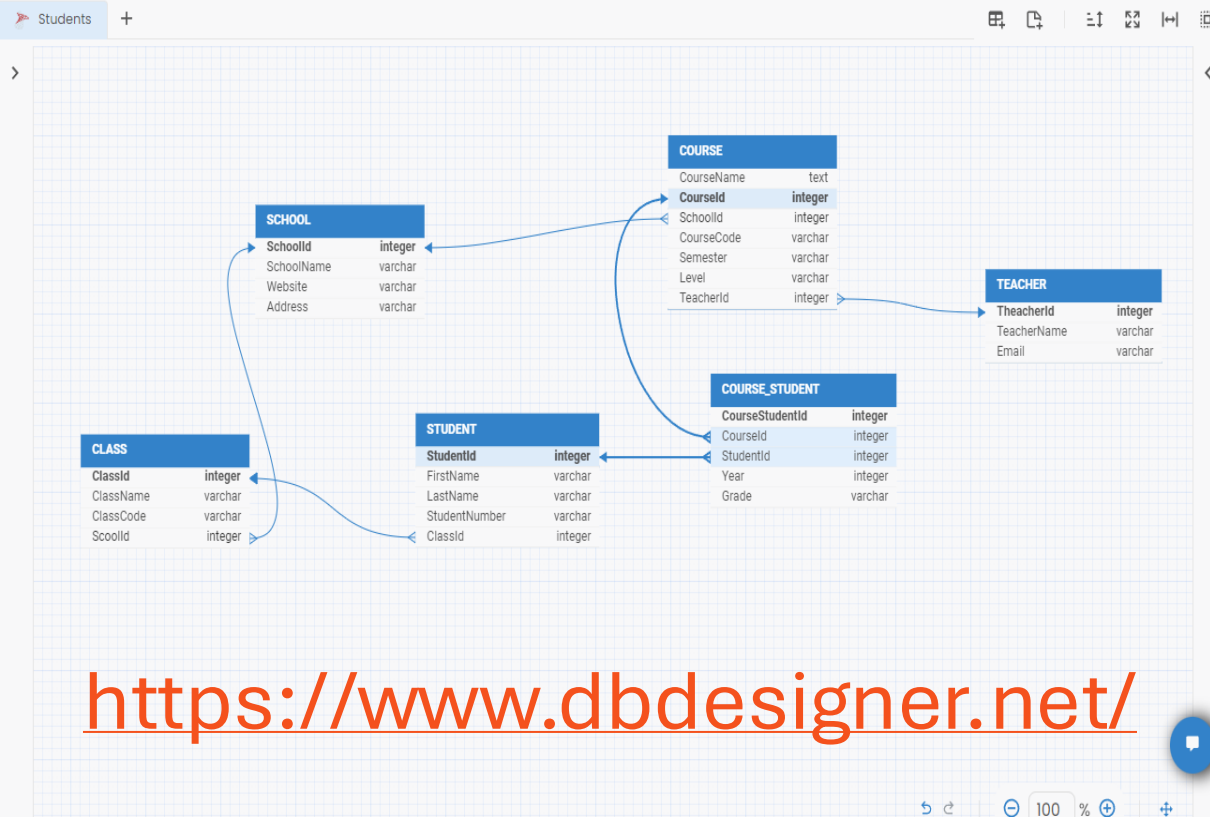
<https://www.dbdesigner.net/>

Hans-Petter Halvorsen





DB Designer



Dashboard + NEW PROJECT

+ NEW PROJECT

Dashboard

Team

My Exports

Logout

My Projects

Shared Projects

Project Name	DB Type	Last Updated
Students	Mssql	3 minutes ago

Project Name	DB Type	Last Updated
Books	Mysql	4 days ago

Project Name	DB Type	Last Updated
Project	Mssql	4 days ago

Project Name	DB Type	Last Updated
Test	Mssql	4 days ago

Showing 1 - 4 of 4 < 1 > Per page 10

<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/>

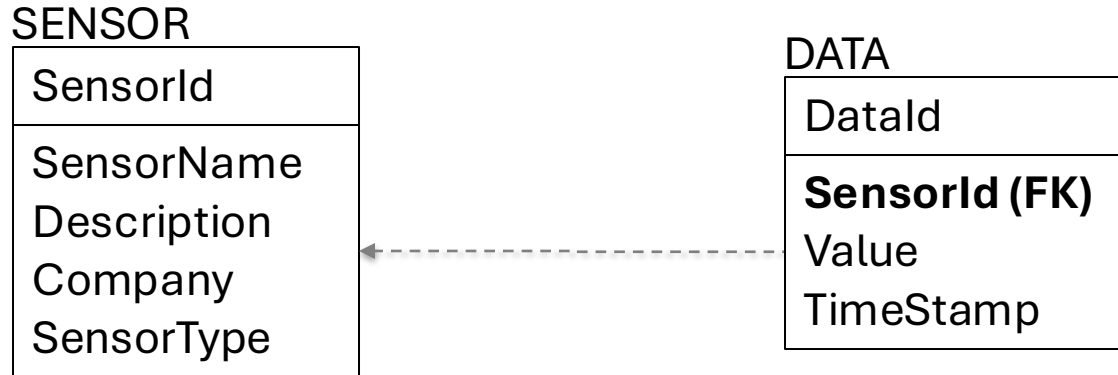
<https://www.halvorsen.blog>

Database Design Fundamentals

Hans-Petter Halvorsen



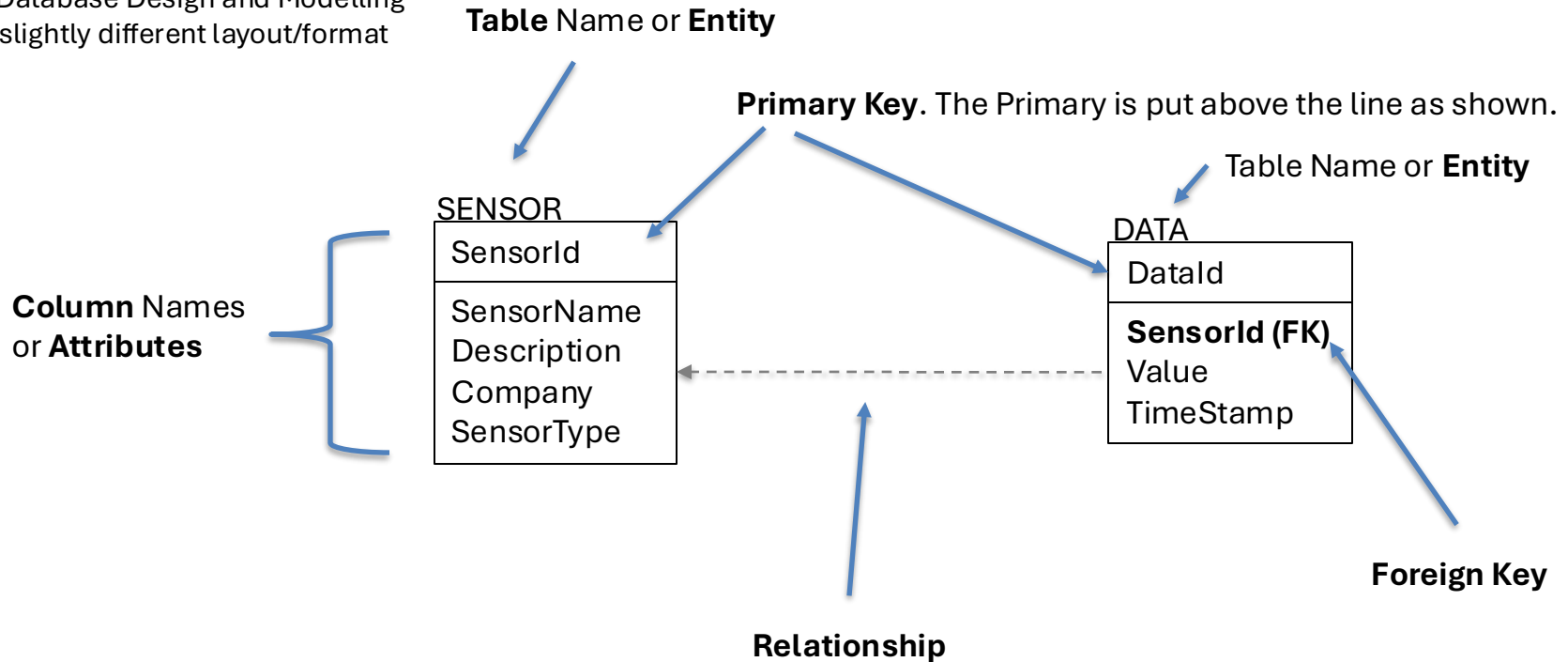
Basic Example



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

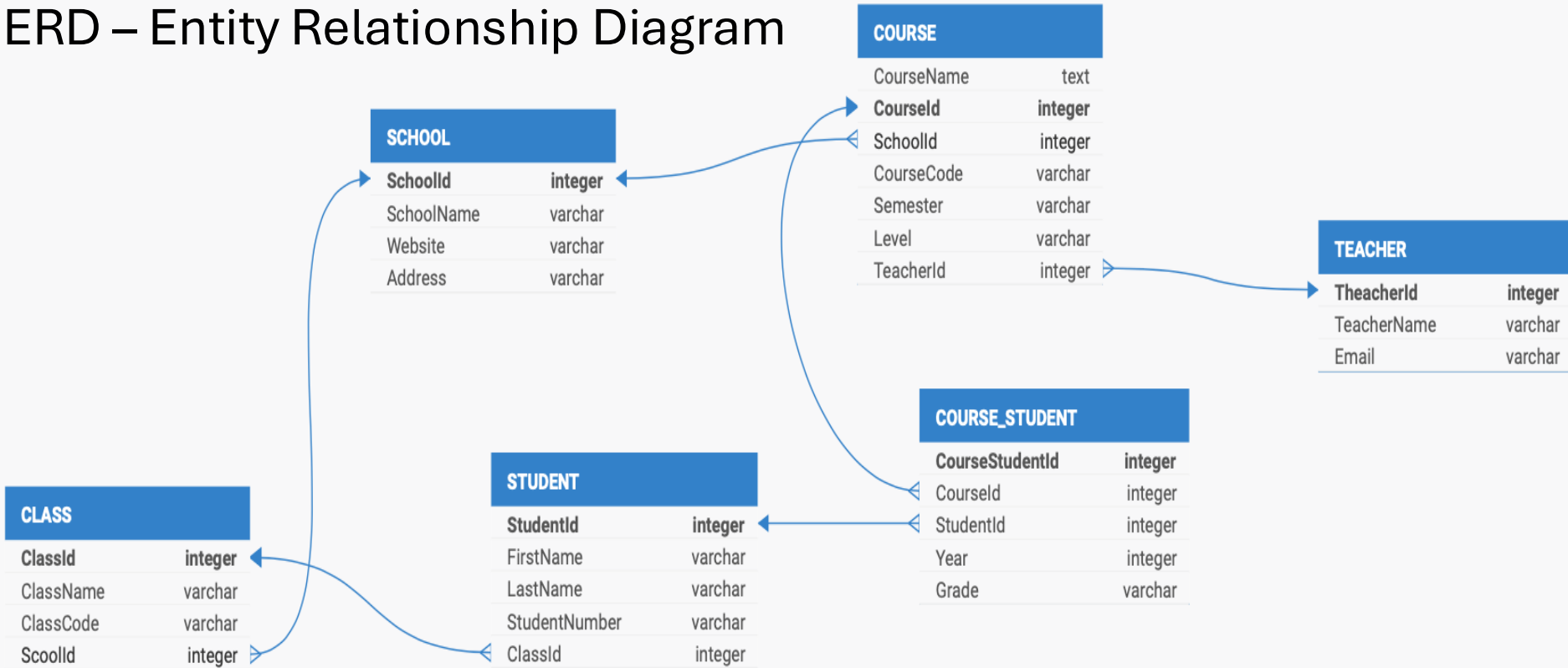
Relationships illustrate the association between two entities

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



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

It is recommended that you use **Pascal notation** for Column names

Primary Keys are just unique running numbers that increase with one for each record. In SQL Server we can use IDENTITY(1,1)

SCHOOL	
SchoolId	integer
SchoolName	varchar
Website	varchar
Address	varchar

COURSE	
CourseName	text
CourseId	integer
SchoolId	integer
CourseCode	varchar
Semester	varchar
Level	varchar
TeacherId	integer

Don't use too many different **datatypes**, stick to a few like int, varchar, bool, date

TEACHER	
TheacherId	integer
TeacherName	varchar
Email	varchar

COURSE_STUDENT	
CourseStudentId	integer
CourseId	integer
StudentId	integer
Year	integer
Grade	varchar

CLASS	
ClassId	integer
ClassName	varchar
ClassCode	varchar
SchoolId	integer

STUDENT	
StudentId	integer
FirstName	varchar
LastName	varchar
StudentNumber	varchar
ClassId	integer

Use descriptive names ("TeacherName"), not just "Name". Many tables containing the same name can be confusing and more difficult when making SQL queries

<https://www.halvorsen.blog>



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/>

Individual Plans

Academic & Non-Profit Plans

The image displays three pricing plans for dbdesigner.net, arranged horizontally. Each plan is presented in a card-like format with a title, a large 'Free' label, a list of features, and a 'START DESIGNING' button.

- Academic Free:** This plan is highlighted with a red circle. It offers unlimited models and tables, and is available only to current students and educators. The card has a light pink background.
- Non-Profit Free:** This plan offers unlimited models and tables, and is available only to non-profit organizations. The card has a blue background.
- Open Source Free:** This plan offers unlimited models and tables, and is available only to contributors of open source projects. The card has a white background.

<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

```
1 SCHOOL {
2   SchoolId integer pk
3   Increments unique
4   SchoolName varchar(100)
5     unique
6   Address varchar(100)
7   Website varchar(100)
8 }
9 COURSE {
10  CourseName text
11  CourseId integer pk
12  Increments unique
13  SchoolId integer *-> SCHOOL
14  SchoolId
15  CourseCode varchar(10)
16  unique
17  Semester varchar(10)
18  Level varchar(10)
19  TeacherId integer *->
20  TEACHER.TeacherId
21 }
22 STUDENT {
23  StudentId integer pk
24  Increments unique
25  FirstName varchar(50)
26  LastName varchar(100)
27  StudentNumber varchar(20)
28  unique
29  ClassId integer *-> CLASS
30  .ClassId
31 }
32 CLASS {
33  ClassId integer pk
34  Increments unique
35  ClassName varchar(100)
36  ClassCode varchar(50)
37  unique
38  SchoolId integer *-> SCHOOL
39  .SchoolId
40 }
41 COURSE_STUDENT {
42  CourseStudentId integer pk
43  Increments unique
44  CourseId integer *-> COURSE
45  .CourseId
46  StudentId integer *->
47  STUDENT.StudentId
48  Year integer
49  Grade varchar(1) null
50 }
51 TEACHER {
52  TeacherId integer pk
53  Increments unique
54  TeacherName varchar(100)
55  Email varchar(100) unique
56 }
```

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

TeacherId	integer
TeacherName	varchar
Email	varchar

STUDENT

StudentId	integer
FirstName	varchar
LastName	varchar
StudentNumber	varchar
ClassId	integer

CLASS

ClassId	integer
ClassName	varchar
ClassCode	varchar
SchoolId	integer

COURSE_STUDENT

CourseStudentId	integer
CourseId	integer
StudentId	integer
Year	integer
Grade	varchar

Schema Properties

Properties

Schema Name: Students

Database type is MSSQL

6 total tables

Schema Explorer

Project Colors

Search schema for tables

- SCHOOL
- COURSE
- STUDENT

Comments

No comments yet. Be the first to comment

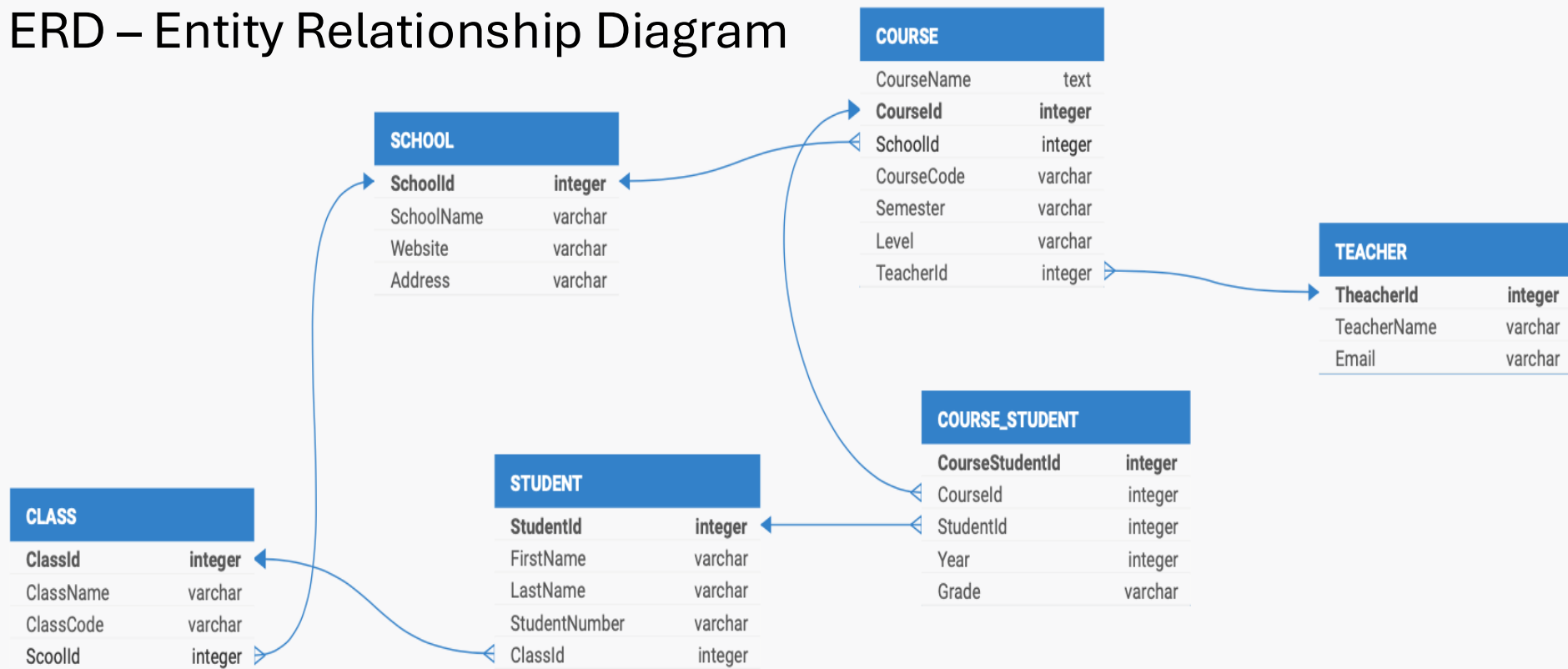
Enter your comment here

SAVE

106 %

Let's create this ERD Example

ERD – Entity Relationship Diagram



Export to SQL

```
Students-1730197595.sql
Users > halvorsen > Downloads > Students-1730197595.sql
1 CREATE TABLE [SCHOOL] (
2     [SchoolId] int IDENTITY(1,1) NOT NULL UNIQUE,
3     [SchoolName] nvarchar(100) NOT NULL UNIQUE,
4     [Address] nvarchar(100) NOT NULL,
5     [Website] nvarchar(100) NOT NULL,
6     PRIMARY KEY ([SchoolId])
7 );
8
9 CREATE TABLE [COURSE] (
10    [CourseName] nvarchar(max) NOT NULL,
11    [CourseId] int IDENTITY(1,1) NOT NULL UNIQUE,
12    [SchoolId] int NOT NULL,
13    [CourseCode] nvarchar(10) NOT NULL UNIQUE,
14    [Semester] nvarchar(10) NOT NULL,
15    [Level] nvarchar(10) NOT NULL,
16    [TeacherId] int NOT NULL,
17    PRIMARY KEY ([CourseId])
18 );
19
20 CREATE TABLE [STUDENT] (
21    [StudentId] int IDENTITY(1,1) NOT NULL UNIQUE,
22    [FirstName] nvarchar(50) NOT NULL,
23    [LastName] nvarchar(100) NOT NULL,
24    [StudentNumber] nvarchar(20) NOT NULL UNIQUE,
25    [ClassId] int NOT NULL,
26    PRIMARY KEY ([StudentId])
27 );
28
29 CREATE TABLE [CLASS] (
30    [ClassId] int IDENTITY(1,1) NOT NULL UNIQUE,
31    [ClassName] nvarchar(100) NOT NULL,
32    [ClassCode] nvarchar(50) NOT NULL UNIQUE,
33    [SchoolId] int NOT NULL,
34    PRIMARY KEY ([ClassId])
35 );
36
37 CREATE TABLE [COURSE_STUDENT] (
38    [CourseStudentId] int IDENTITY(1,1) NOT NULL UNIQUE,
39    [CourseId] int NOT NULL,
40    [StudentId] int NOT NULL,
41    [Year] int NOT NULL,
42    [Grade] nvarchar(1),
43    PRIMARY KEY ([CourseStudentId])
```

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

×


Import Sql

Import Engine

Use DB Designer's Engine Use Ai

Sql ...

OR

 Drag n drop or upload .sql file.

Please send a bug report if you encounter errors. Imported SQL should only include DDL statements.

IMPORT CANCEL

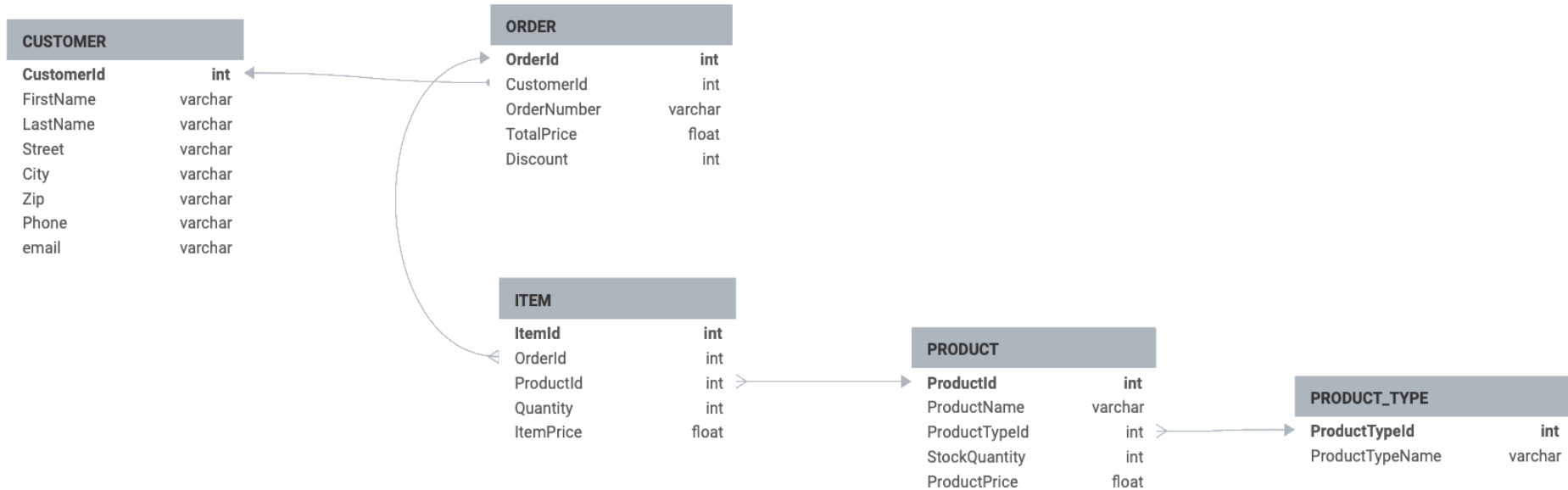
You can also Import SQL Files and generate ERD based on that SQL Script

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>

