Lucidchart

Database Design and Modelling using Lucidchart

Hans-Petter Halvorsen

Lucidchart

- Lucidchart is an online multipurpose diagram tool.

- It can also be used to create Database diagrams.

- You can use it for free, but it also have paid options for more functionality and features.

ERD – Entity Relationship Diagram

- You can share and collaborate with others.

- It has a built in ERD template that you can use.

- Homepage: https://www.lucidchart.com

- Create an Entity Relationship Diagram in Lucidchart: https://help.lucid.co/hc/en-us/articles/1647565238292-Create-an-Entity-Relationship-Diagram-in-Lucidchart
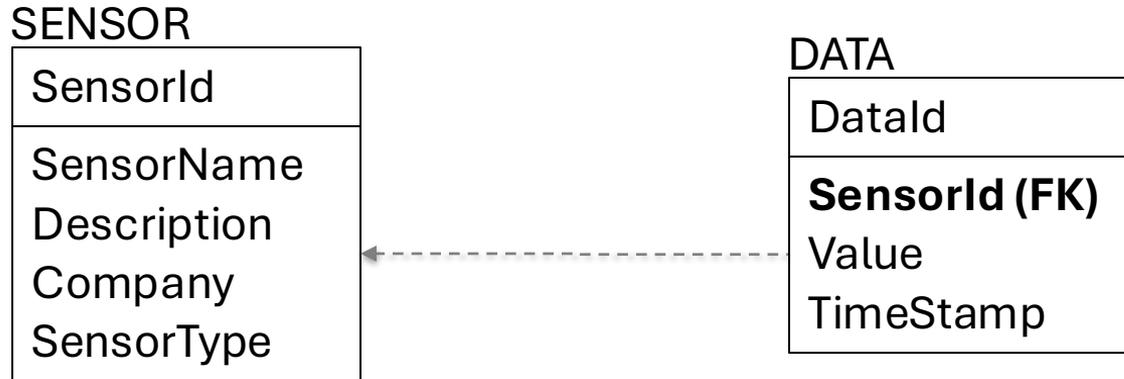
Design Fundamentals

Hans-Petter Halvorsen

Basic Example

SENSOR

SensorId
SensorName Description Company SensorType

DATA

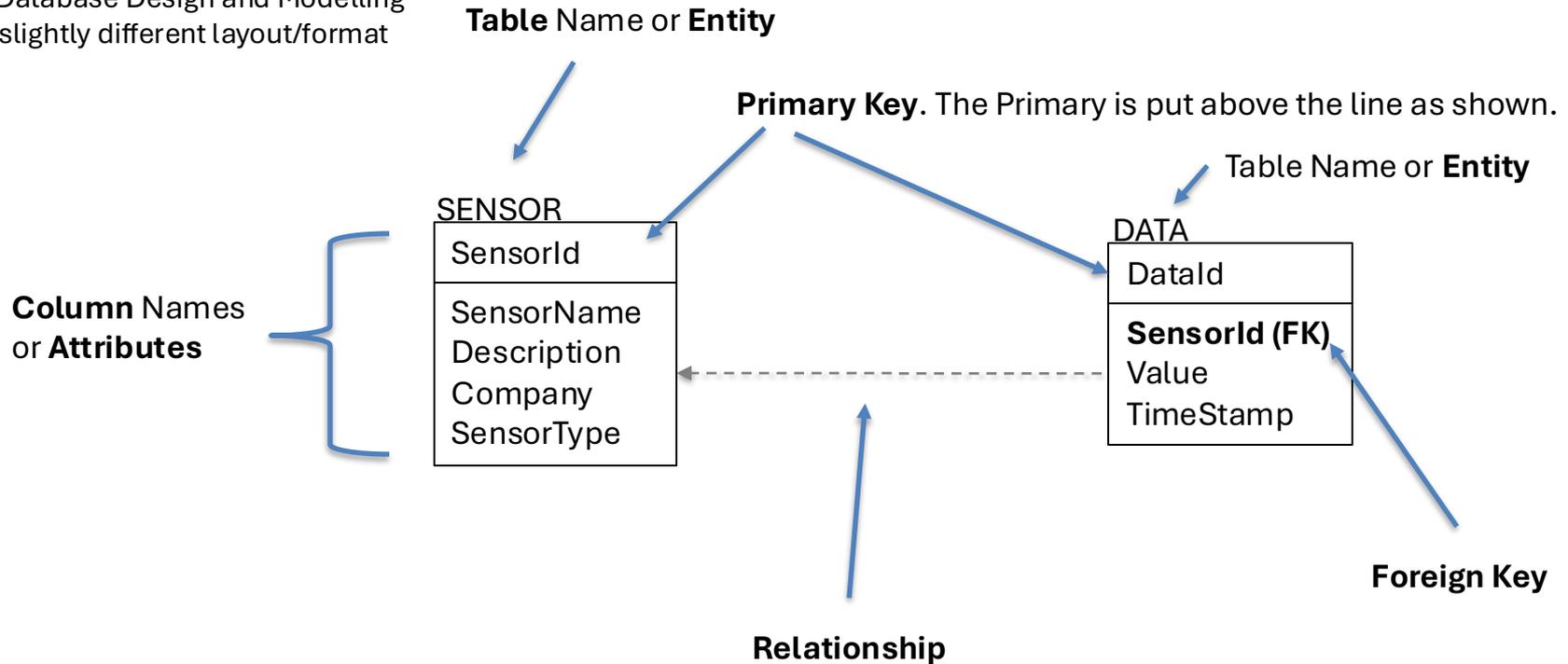| DataId |
| --- |
| **SensorId (FK)**
Value
TimeStamp |

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

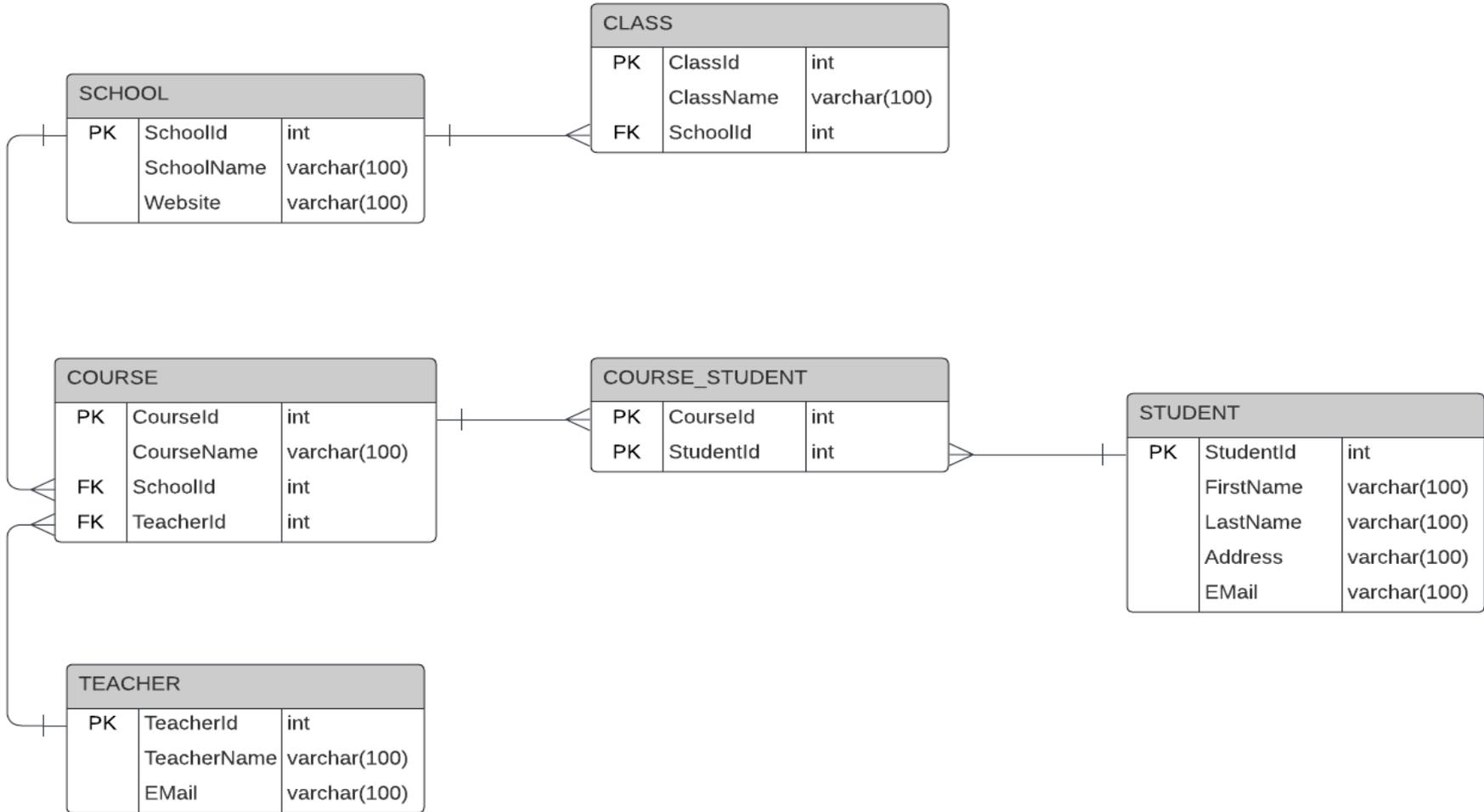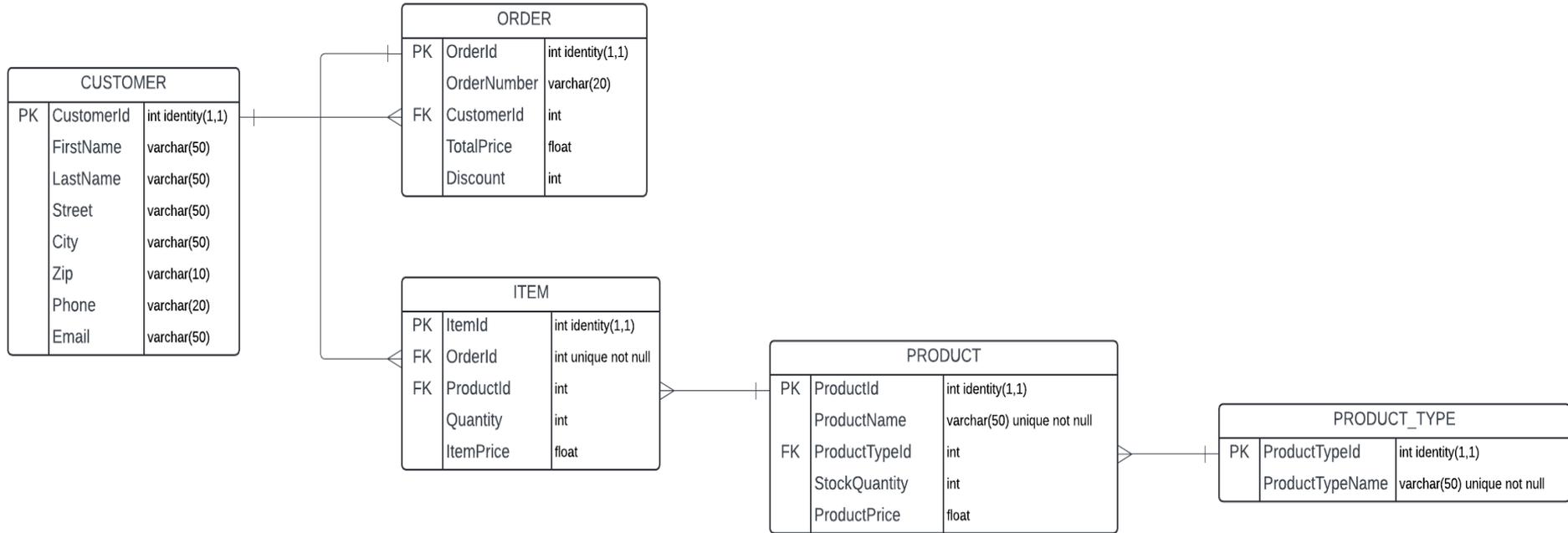Each Database Design and Modelling has a slightly different layout/format

Table Name or **Entity**

Primary Key. The Primary is put above the line as shown.

Table Name or **Entity**

SENSOR

SensorId
SensorName
Description
Company
SensorType

Column Names or **Attributes**

DATA

DataId
SensorId (FK)
Value
TimeStamp

Foreign Key

Relationship

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 like erwin Data Modeler, MySQL Workbench, etc.

- While other tools are just general-purpose diagram tools like Visio, Miro, Lucidchart, Draw.io, etc.

- The advantage with a specialized Database Design and Modelling tool is that you can generate SQL code for your tables.

- Lucidchart has also a simple feature for generating a SQL script.

SCHOOL

PK	SchoolId	int
	SchoolName	varchar(100)
	Website	varchar(100)

CLASS

PK	ClassId	int
	ClassName	varchar(100)
FK	SchoolId	int

COURSE

PK	CourseId	int
	CourseName	varchar(100)
FK	SchoolId	int
FK	TeacherId	int

COURSE_STUDENT

PK	CourseId	int
PK	StudentId	int

STUDENT

PK	StudentId	int
	FirstName	varchar(100)
	LastName	varchar(100)
	Address	varchar(100)
	EMail	varchar(100)

TEACHER

PK	TeacherId	int
	TeacherName	varchar(100)
	EMail	varchar(100)

CUSTOMER

PK	CustomerId	int identity(1,1)
	FirstName	varchar(50)
	LastName	varchar(50)
	Street	varchar(50)
	City	varchar(50)
	Zip	varchar(10)
	Phone	varchar(20)
	Email	varchar(50)

ORDER

PK	OrderId	int identity(1,1)
	OrderNumber	varchar(20)
FK	CustomerId	int
	TotalPrice	float
	Discount	int

ITEM

PK	ItemId	int identity(1,1)
FK	OrderId	int unique not null
FK	ProductId	int
	Quantity	int
	ItemPrice	float

PRODUCT

PK	ProductId	int identity(1,1)
	ProductName	varchar(50) unique not null
FK	ProductTypeId	int
	StockQuantity	int
	ProductPrice	float

PRODUCT_TYPE

PK	ProductTypeId	int identity(1,1)
	ProductTypeName	varchar(50) unique not null

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,

Lucidchart

Lucidchart

Hans-Petter Halvorsen

Lucidchart

- Lucidchart is an online multipurpose diagram tool.

- It can also be used to create Database diagrams.

- You can use it for free, but it also have paid options for more functionality and features.

ERD – Entity Relationship Diagram

- You can share and collaborate with others.

- It has a built in ERD template that you can use.

- Homepage: https://www.lucidchart.com

- Create an Entity Relationship Diagram in Lucidchart: https://help.lucid.co/hc/en-us/articles/1647565238292-Create-an-Entity-Relationship-Diagram-in-Lucidchart

Free Plan

Note! With the Free version you can only have 3 <u>editable</u> diagrams Lucidchart diagrams. This means you can only update your 3 recent diagrams.

This is typically good enough for a single project, student work or a student project

You can also create multiple Pages, etc, in one document. This means you can have one ER diagram in each Page inside a Lucidchart diagram

∞ PREMIUM

You're trying to access a premium feature. Upgrade to enjoy **Unlimited Editable Documents** and much more.

You're currently on a free account, which allows up to 3 editable documents. If you create a document anyway, one of your documents will become view only. **Upgrade now to get unlimited editable documents and more:**

- Visio Import
- Unlimited Editable Documents
- Revision History
- Unlimited Shapes
- Premium Templates
- And <u>much more</u>:

Start trial   **Create a document anyway**

Lucidchart – Create New ERD

Templates

You can also start from a Template:

Templates

You can also start from a Template:

Lucidchart – Crowfoot Notation

SCHOOL		
PK	SchoolId	int
	SchoolName	varchar(100)
	Website	varchar(100)

CLASS		
PK	ClassId	int
	ClassName	varchar(100)
FK	SchoolId	int
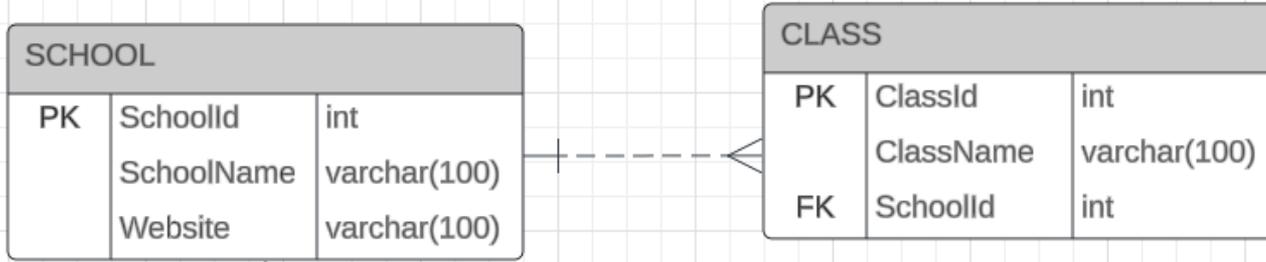
We have different types of relationships between 2 tables, but **most of the time we use a "One-to-many" (1:N) relationship**.

—————————————|— One

—————————————< Many

—————————————||— One (and only one)

—————————————O|— Zero or one

—————————————K One or many

—————————————OK Zero or many

	CLASS		
PK	Class...	nt
	ClassN	va
FK	School	nt
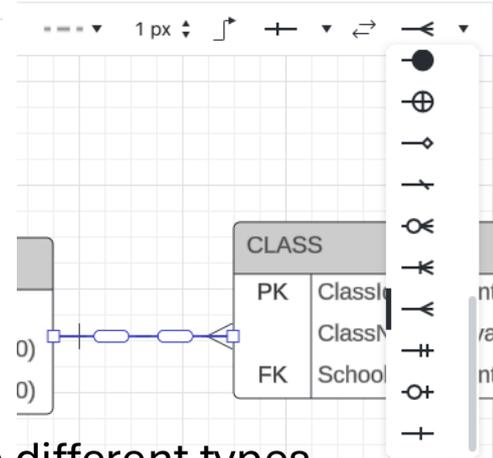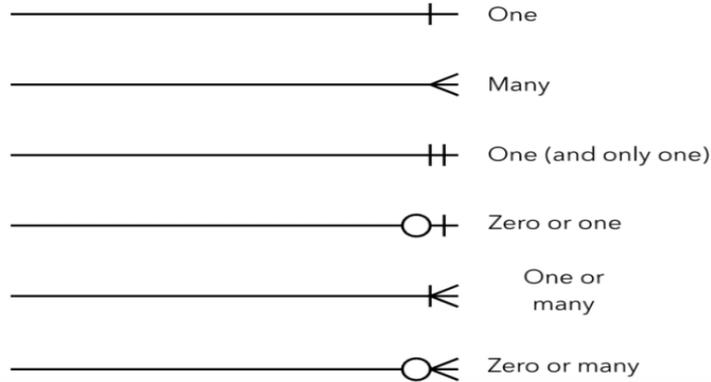
Lucidchart has support for all these different types

Lucidchart Example
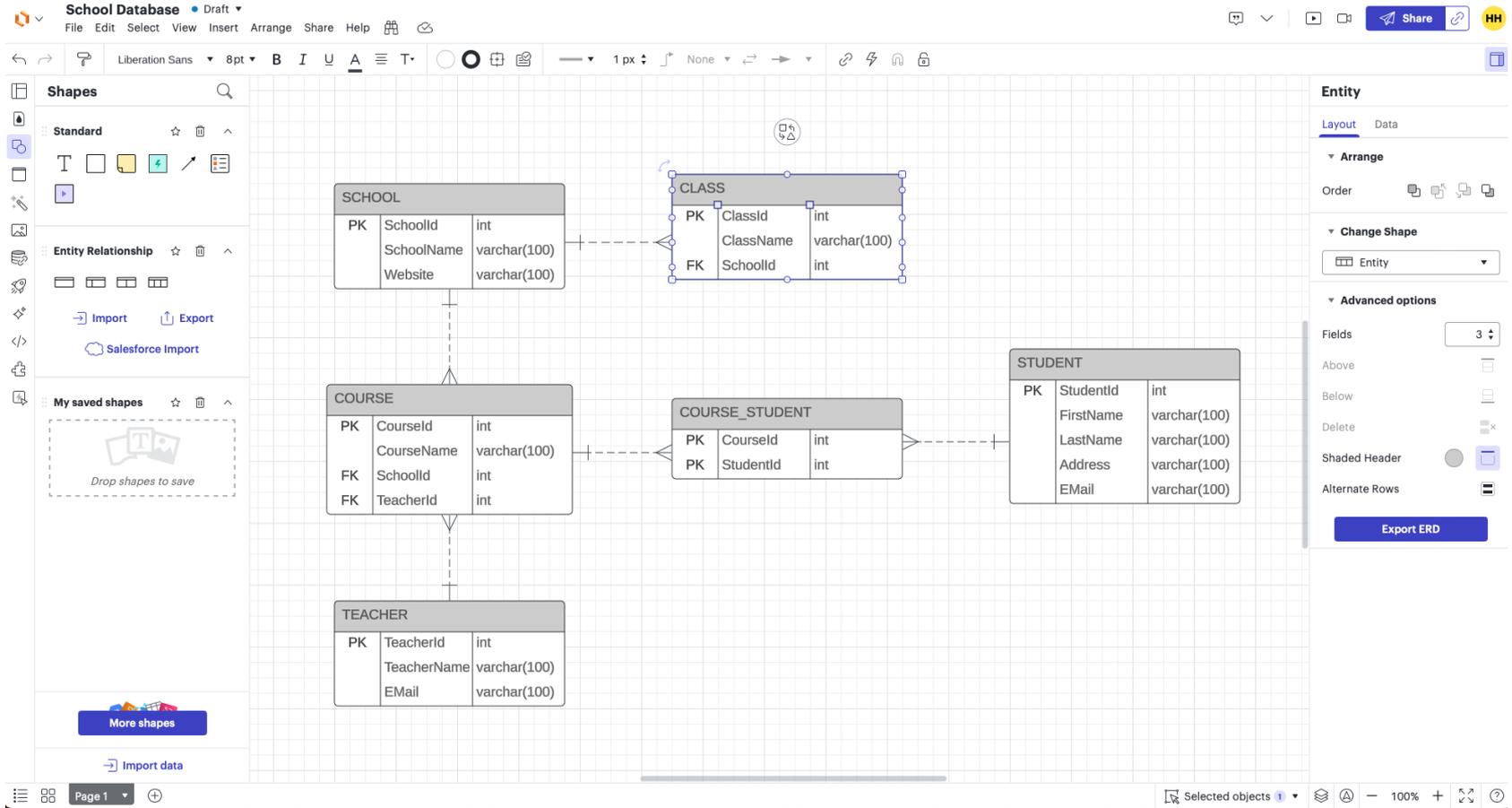
Export as Image

Typically, you want to include your ERD in a report or a PowerPoint, so then you can use the Export function.

SCHOOL

PK	SchoolId	int
	SchoolName	varchar(100)
	Website	varchar(100)

CLASS

PK	ClassId	int
	ClassName	varchar(100)
FK	SchoolId	int

COURSE

PK	CourseId	int
	CourseName	varchar(100)
FK	SchoolId	int
FK	TeacherId	int

COURSE_STUDENT

PK	CourseId	int
PK	StudentId	int

STUDENT

PK	StudentId	int
	FirstName	varchar(100)
	LastName	varchar(100)
	Address	varchar(100)
	EMail	varchar(100)

TEACHER

PK	TeacherId	int
	TeacherName	varchar(100)
	EMail	varchar(100)

Export to SQL

These Tables can now be easily exported as a SQL script and be inserted into SQL Server, MySQL, etc.
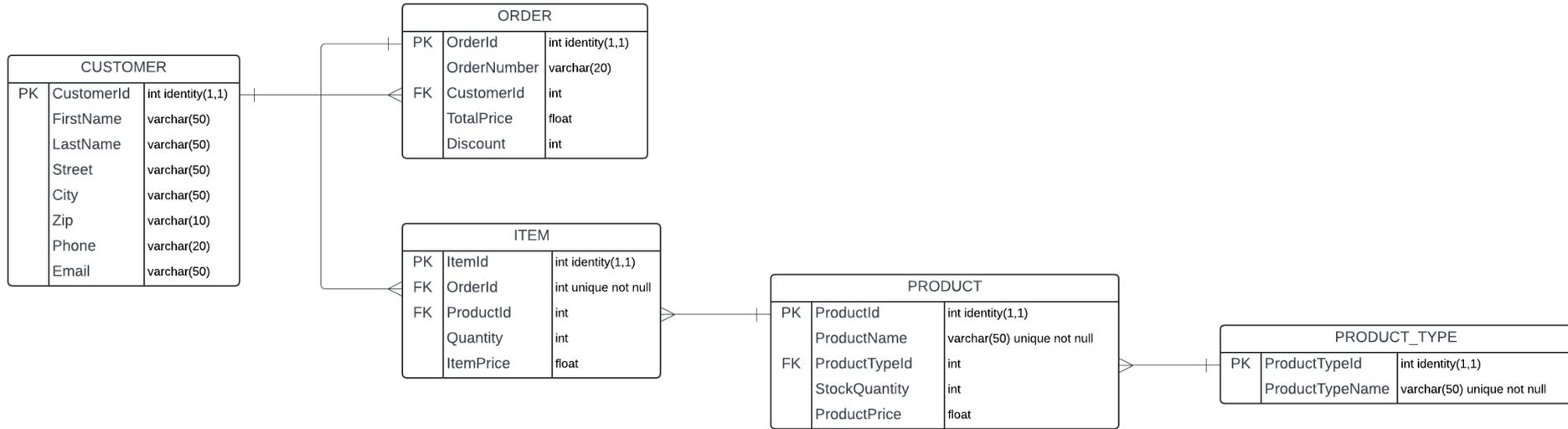
Order System Example

- Customers
- Products
- Orders
- ..
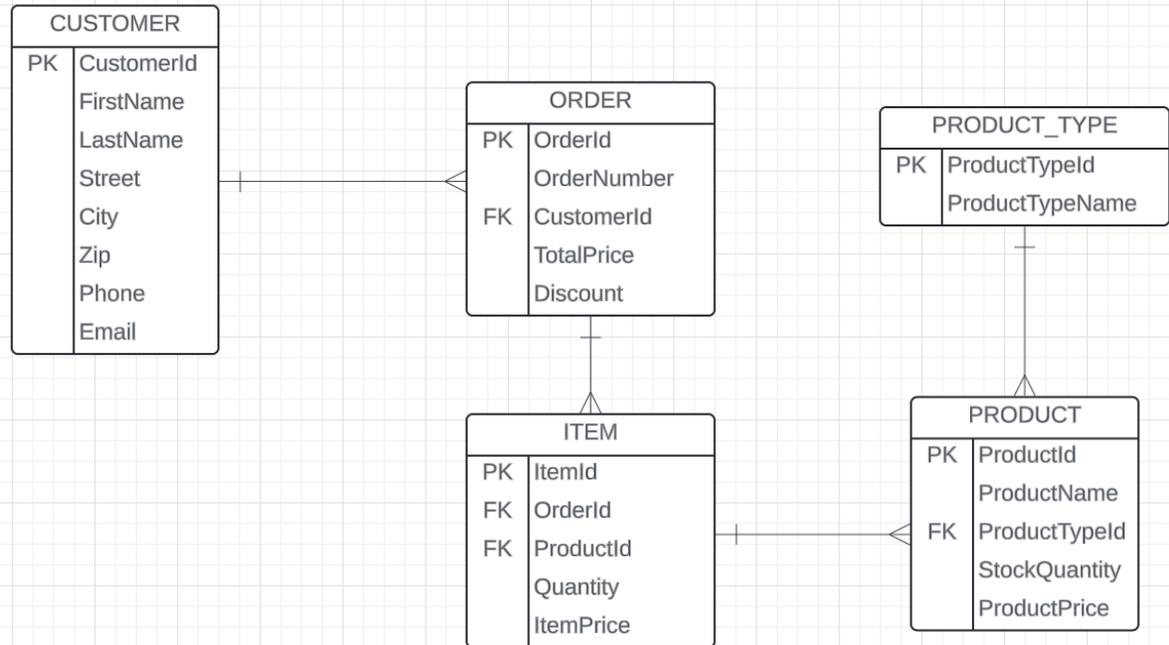
Order System

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

These Tables can now be easily 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