



NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

MDE

TP1 – Class 2

Database Modeling

2024 – 2025

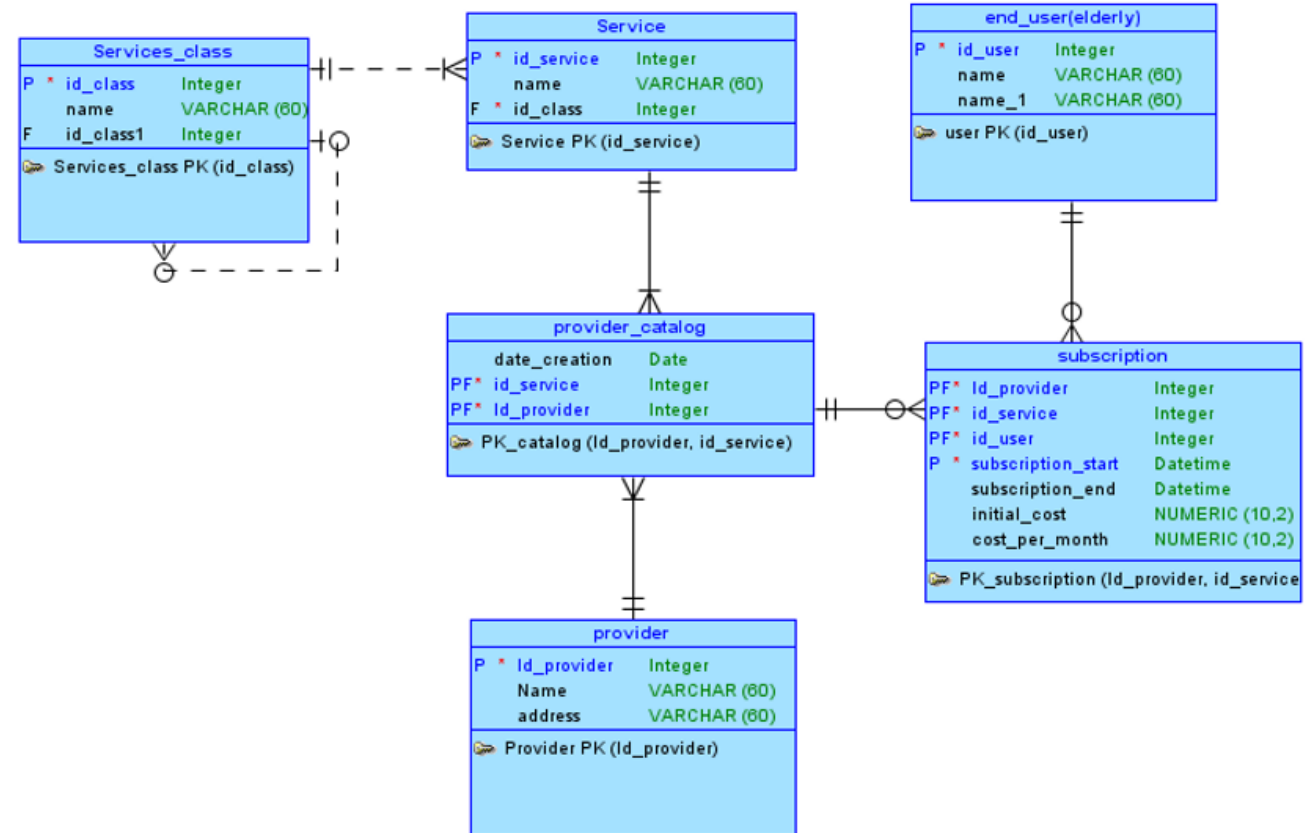
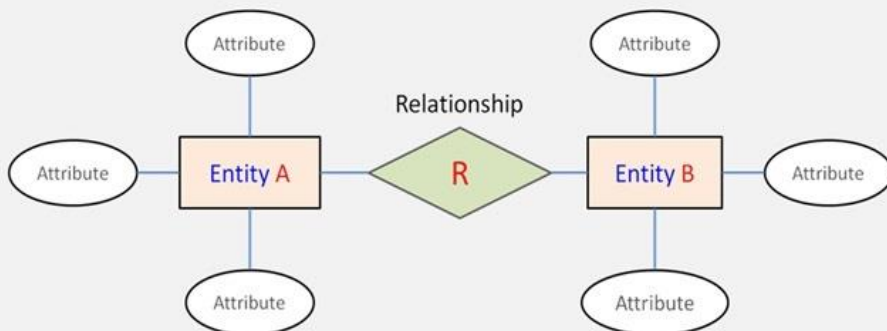
- ❑ ER Diagram overview
- ❑ Presentation of the Lab work 1 assignment
- ❑ Modeling the proposed system:
 - Identification of relevant entities and relationships
 - Characterization of each entity
 - Constraints and integrity aspects
- ❑ Model Creation in MySQL (Forward Engineering)
- ❑ Populating the database with testing data + Querying (SQL scripting)

ERD Review (from theoretical classes)

ERD

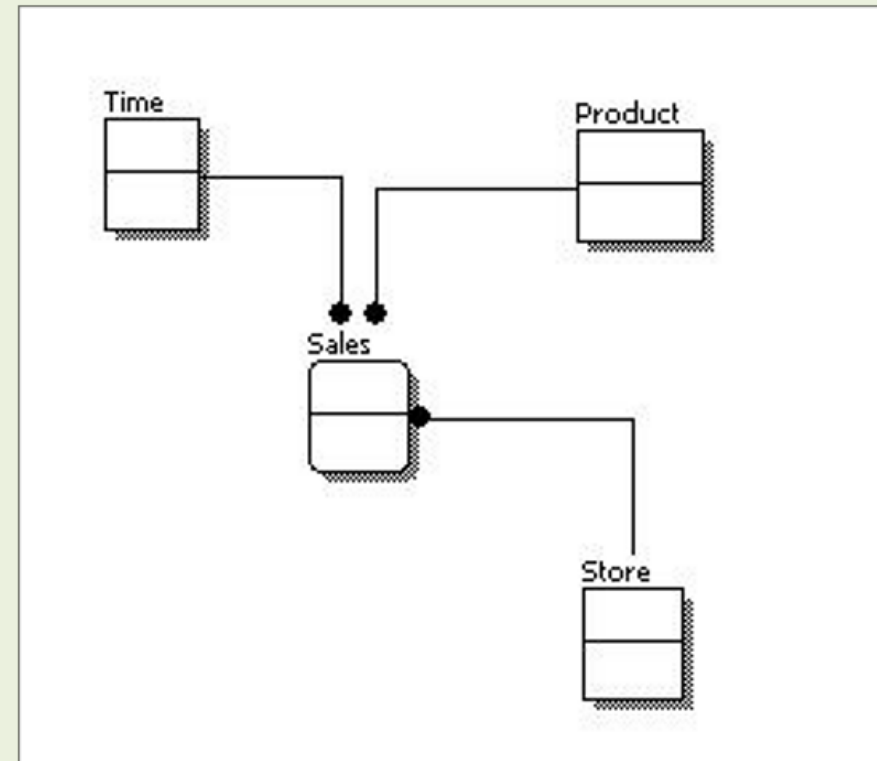
Entity-Relationship Diagram (ERD):

- is a graphical representation of **entities** and their **relationships** to each other.
- It is widely used for designing and modeling **relational databases**. In an ERD, entities are represented as rectangles, relationships as diamonds, and attributes as ovals.



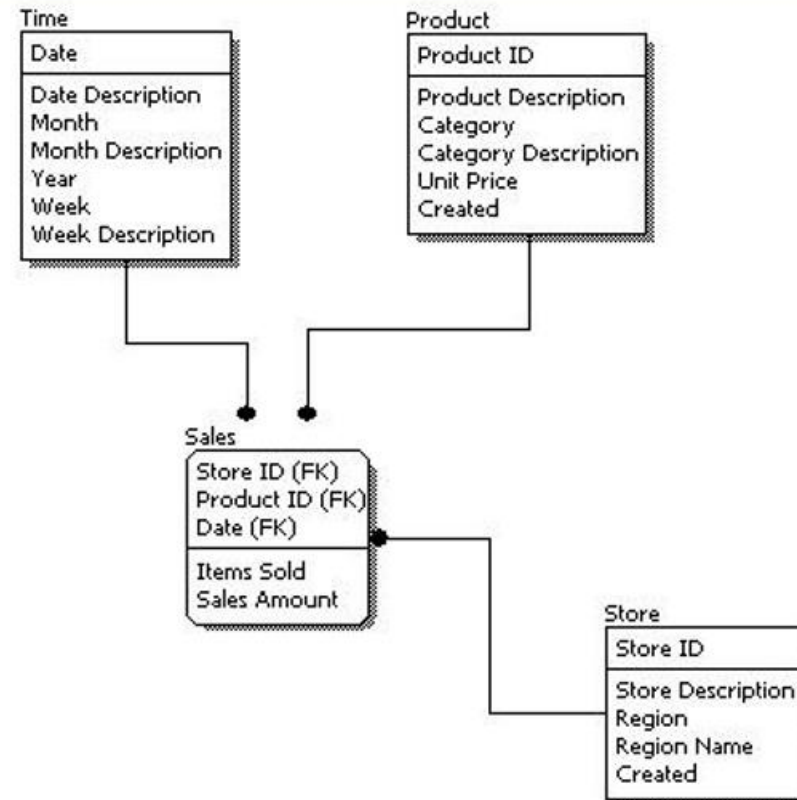
Conceptual Model

- At this level, the objective is to identify the classes/entities and their relationships/ interactions between these classes/entities.



Logical Model

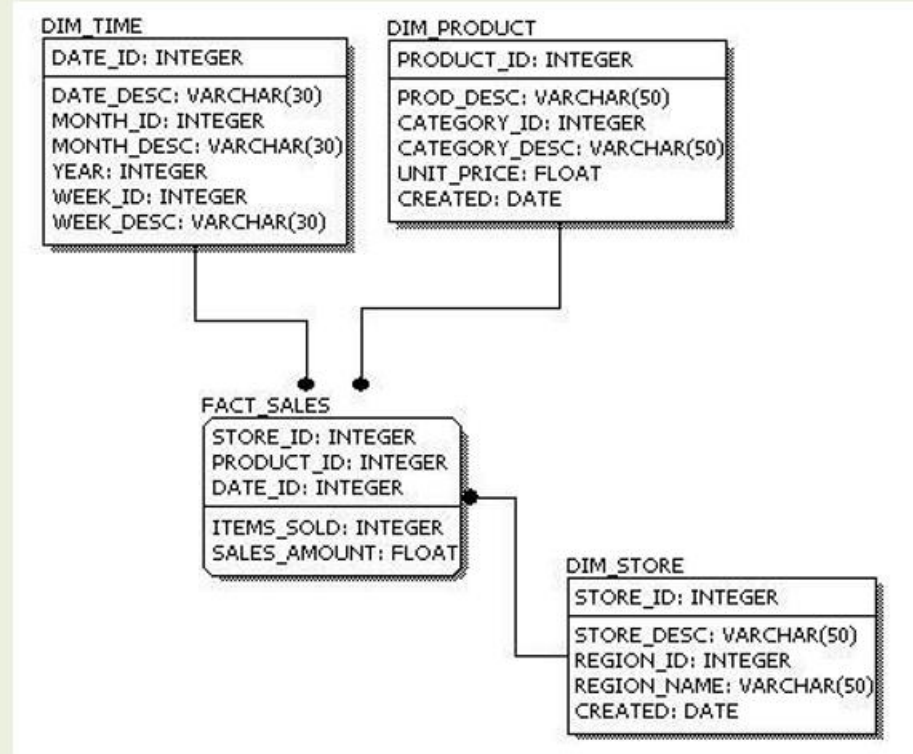
- Specify the elements of the entities, without concern about how they will be implemented.



<http://www.1keydata.com/datawarehousing/data-modeling-levels.html>

Physical model

- Specify exactly how the model will be implemented in a data base
- Definition of the fields (attributes) and their types, constraints, integrity rules.



<http://www.1keydata.com/datawarehousing/data-modeling-levels.html>

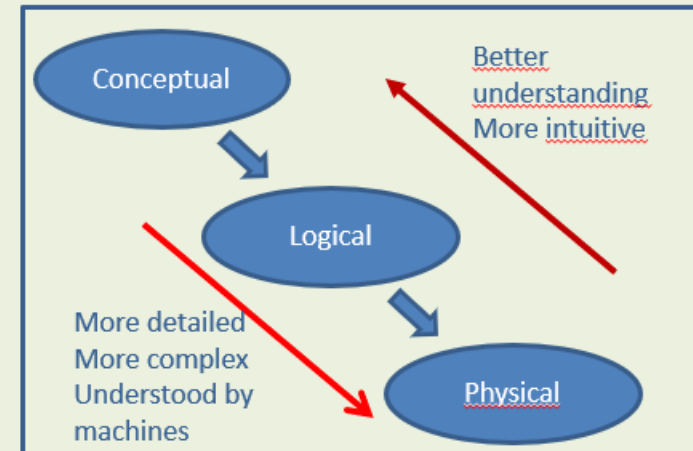
Abstraction levels

Conceptual (The **what** layer)

- More abstract
- Of the highest level
- Greater readability
- More intuitive
- Transmitting the idea/essence of the problem (even to people outside the engineering field, e.g., managers).

Logical and physical (The **how** layer)

- More concrete
- More details in order to define exactly how to implement the specified model.
- Machine software can read it.



Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

PRESENTATION OF WORK ASSIGNMENT



Descrição do trabalho a efetuar

Este trabalho destina-se a proporcionar aos alunos um contacto com os conceitos básicos acerca do projeto/modelação de bases de dados, tendo em vista a sua posterior utilização em atividades enquadradas no seio da Engenharia Eletrotécnica e de Computadores.

Concretamente, o estudo e a implementação do problema proposto neste trabalho servirão para fazer uma abordagem aos seguintes conceitos, que constituem as fases de um projeto de criação de uma base de dados:

- Estudo do problema apresentado.
- Modelação dum diagrama de Entidades e Relacionamentos (*DER*) que satisfaça os requisitos do problema apresentado.
- Transformação do diagrama num esquema/modelo relacional e implementação numa base-de-dados, utilizando *SQL*.
- Implementação de questões (*SQL queries*), vistas (*SQL views*), triggers (*SQL triggers*) e funções/procedimentos (*SQL functions/procedures*) de forma a implementar as funcionalidades pretendidas no problema.
- Valorizam-se (até um máximo de 1 valor, **com limite máximo de 20 valores**) os trabalhos que implementem um conjunto de interfaces amigáveis para a visualização e manipulação dos dados armazenados na BD (*JDBC*, *outros*).



Descrição Geral do Problema

Você foi contratado para desenvolver um sistema de base de dados para uma empresa especializada na gestão de pequenos produtores agrícolas e otimização do uso de recursos. Este sistema deve gerir informações sobre produtores, cultivos, tratamentos, distribuidores e registos de produção e comercialização, considerando os seguintes pontos:

- **Produtores:** A empresa atende pequenos agricultores que cultivam diferentes tipos de produtos. Cada produtor é identificado por informações como nome, endereço, NIF (Número de Identificação Fiscal), telefone, área cultivada, entre outros.
- **Cultivos:** Cada produtor pode ter vários cultivos, identificados por um código único, tipo de cultura, período de plantio e colheita, rendimento esperado e práticas agrícolas adotadas.
- **Tratamentos:** Os produtores aplicam tratamentos agrícolas, como fertilizantes e pesticidas. Cada tratamento deve ser registado com seu nome, fornecedor, quantidade utilizada e data de aplicação.
- **Distribuidores:** Os produtos colhidos são enviados para distribuidores, que devem ser identificados por nome, localização, capacidade de armazenamento e histórico de compras.
- **Registos de Produção:** Cada lote de produção deve ser registado no sistema, incluindo data de colheita, quantidade produzida, qualidade do produto e destino da produção.

- **Comercialização:** O sistema deve otimizar a venda dos produtos agrícolas, registrando informações sobre preços praticados, custos de logística e demanda de mercado.
- **Sensores IoT:** O sistema deve permitir a integração com sensores IoT para monitoramento em tempo real de variáveis agrícolas, como umidade do solo, temperatura, qualidade do ar e níveis de irrigação. Esses dados devem ser coletados, armazenados e analisados para otimizar a produção.
- **Criação de Alertas:** Criação de alertas para os produtores baseados nas condições atuais da produção, como por exemplo humidade do solo a baixo de um determinado valor, ou temperatura da produção acima de um valor aceitável.
- **Simulação de Dados:** Devido à ausência inicial de um sistema operacional real, os dados referentes à produção e comercialização deverão ser simulados para testar e demonstrar as funcionalidades do sistema.
- **Dados Gerados:** Posteriormente, será disponibilizada a integração com fontes de dados externos, como previsões climáticas e tendências de mercado. A solução deverá ser capaz de integrar esses dados e sugerir boas práticas aos produtores.
- **Gestão de Recursos:** O sistema deve incluir funcionalidades para monitorizar e otimizar o uso de água, fertilizantes e energia na produção, minimizando desperdícios e promovendo a sustentabilidade.
- **Faturação:** Deve haver um registo de transações comerciais, detalhando o número da fatura, data, volume comercializado, destino da produção e estado da fatura (emitida/paga). As faturas podem ser geradas automaticamente no final de cada ciclo de comercialização.

Este sistema de base de dados deve fornecer uma plataforma robusta para a gestão e operação eficiente de pequenas propriedades agrícolas, garantindo um melhor aproveitamento dos recursos e aumentando a rentabilidade dos produtores.



Requisitos funcionais para o problema apresentado

O desenvolvimento do projeto para o cenário apresentado requer a construção dum modelo para a base-de-dados, que consiga satisfazer um conjunto considerável de requisitos. No entanto, dado o tempo disponível para este trabalho, é apenas necessário que o modelo consiga satisfazer os requisitos funcionais definidos na tabela 1.

Tabela 1 – Requisitos funcionais (para a BD)

Req. Func.	Descrição
RF 1	Implementar as operações CRUD da BD para as entidades identificadas.
RF 2	Registar valores provenientes das produções (quantidade, qualidade, comercialização).
RF 3	Proceder ao registo de alertas para condições adversas
RF4	Visualizar o histórico de colheitas de um produtor num intervalo de tempo.
RF 5	Visualizar os produtores que mais comercializam produtos.
RF6	Visualizar o estado atual dos tratamentos e estoque de produtos.
RF7	Extraír e processar dados de sensores IoT no campo agrícola. (Temperatura/humidade/luminosidade máxima/média/mínima)
RF 8	Gerar relatórios sobre eficiência agrícola e uso de recursos.
RF 9	Proponha um requisito relevante e ainda por identificar que implique a especificação de uma ou mais entidades. Implemente.
RF 10	Proponha um requisito relevante ainda por identificar e que requeira uma query simples para o satisfazer (Query ou View usada numa Query). Implemente.
RF 11	Proponha um requisito relevante ainda por identificar e que requeira uma query com funções de agregação (sum, max, min, avg, etc) para o satisfazer. Implemente.
RF 12	Proponha um requisito relevante ainda por identificar e que requeira o desenvolvimento de functions / procedures para o satisfazer. Implemente.
RF13	Implementação de uma aplicação em Java que seja cliente da base de dados desenvolvida (Realizar os requisitos anteriores numa aplicação)
RF14	Extração de dados reais da instalação de testes (Extraír e guardar os dados na base de dados respeitando os requisitos anteriores)

Para que se consiga avaliar adequadamente a qualidade do modelo de dados, é necessário que a BD possua dados em quantidade razoável. Por exemplo, calcular a média dum valor com base numa lista que contenha apenas um (ou poucos valores) não é satisfatório. **É necessário que cada tabela possua uma quantidade suficiente de dados que permita ilustrar os requisitos.**

Proceed to develop the ERD for the problem in the lab work

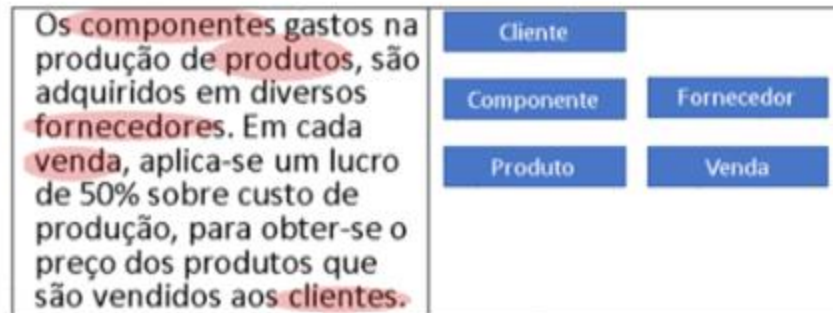
Do Iteratively:

1. Read the problem
2. Use the Peter Chan approach, described during TP classes, to develop the Entity-Relationship Diagram
 - a. Nouns -> Entities
 - b. Verbs -> Relationships
3. Create the ERD (or change it).
4. Develop it (forward engineering)
5. Create the schema
6. Insert testing data
7. Make queries (according to the needs of the problem).
8. Verify whether the model satisfies the problem (able to handle the requirements in the document).
9. If not, solve the identified issues/gaps (repeat the cycle).

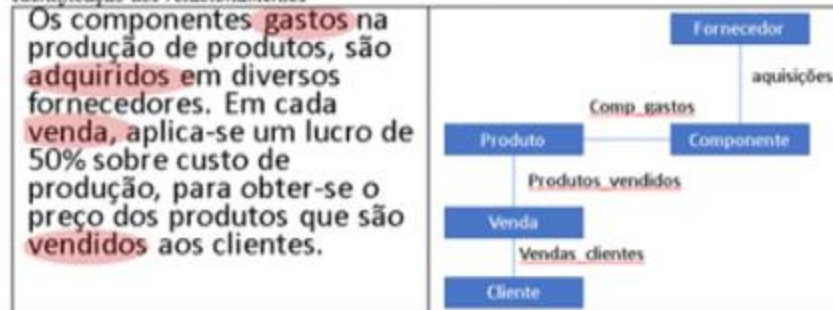
ENTITY-RELATIONSHIP DIAGRAM

Anexo 1 – Modelação: Criação do Diagrama de Entidades e Relacionamentos

Ilustrando como se identificam entidades a partir da descrição do problema. Para modelos de grandes dimensões, uma boa abordagem consiste em representar parcelas pequenas do modelo, ou seja, um fragmento de cada.



Identificação dos relacionamentos



Para uma representação mais compacta, pode-se fundir as duas tabelas anteriores numa só, conseguindo-se identificar em simultâneo as entidades e os relacionamentos, utilizando duas cores distintas, por exemplo, destacar com azul os verbos e a verde os substantivos.

Use the Peter Chan approach to identify the entities and relationships from the description of work

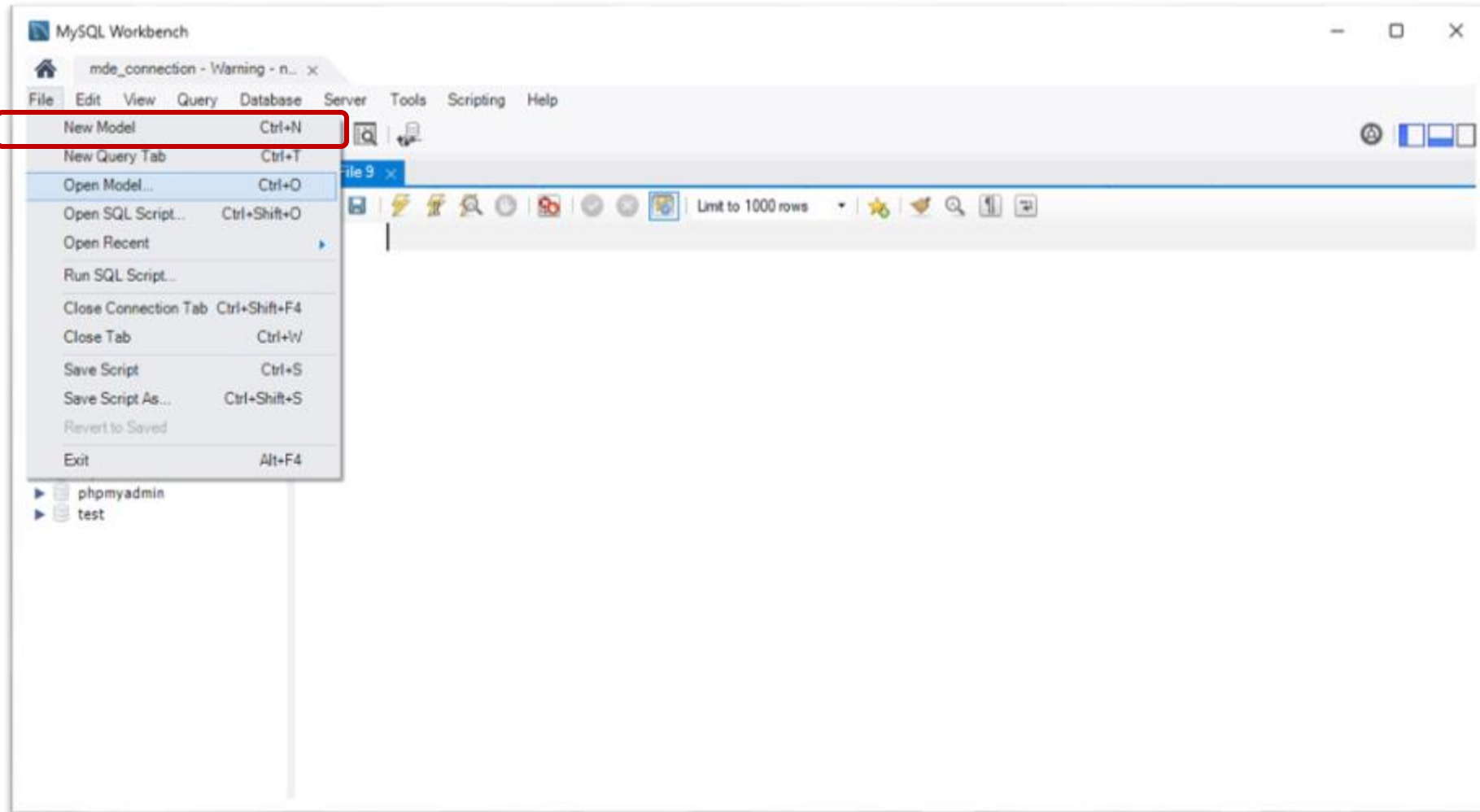
Use the template given in Annex 1 of the Work assignment document and the slides of theoretical classes

Caracterização dos relacionamentos

#	Necessidade	Entidade 1 (dominante)	Entidade 2 (dependente)	Cardinalidade
1	Sistema, registar , <u>faculdades</u>	Sistema (UNL)	Faculdades	1---N
2	<u>faculdade</u> possui <u>curros</u>	Faculdades	Cursos	1---N
3	<u>curros</u> , que " pode ser de diversos " <u>tipos</u> (licenciaturas, mestrados, doutoramentos)	Tipos_curros	Cursos	1---N
4	<u>Faculdade</u> agrega vários <u>departamentos</u>	Faculdades	Departamentos	1—N
5	sendo estes (os dep.) que prestam os <u>curros</u> referidos.	Departamentos	Cursos	1---N
6	Um <u>departamento</u> possui <u>docentes</u>	Departamentos	Docentes	1---N
7	<u>docentes</u> que " estão agrupados " por distintas <u>áreas científicas</u>	<u>áreas científicas</u>	Docents	1---N
8	Cada <u>curso</u> tem associado um perfil curricular	Curso	Perfil_curricular	1---N
9	<u>perfil curricular</u> que descreve as <u>disciplinas</u> respetivas	Perfil_curricular	disciplina	N---M
10	<u>alunos</u> matriculam-se num curso	Curso	Aluno	N---M
11	Cada <u>aluno</u> inscreve-se nas diversas <u>disciplinas</u>	Aluno	Disciplina	N-M

MODEL CREATION IN MySQL (forward engineering)

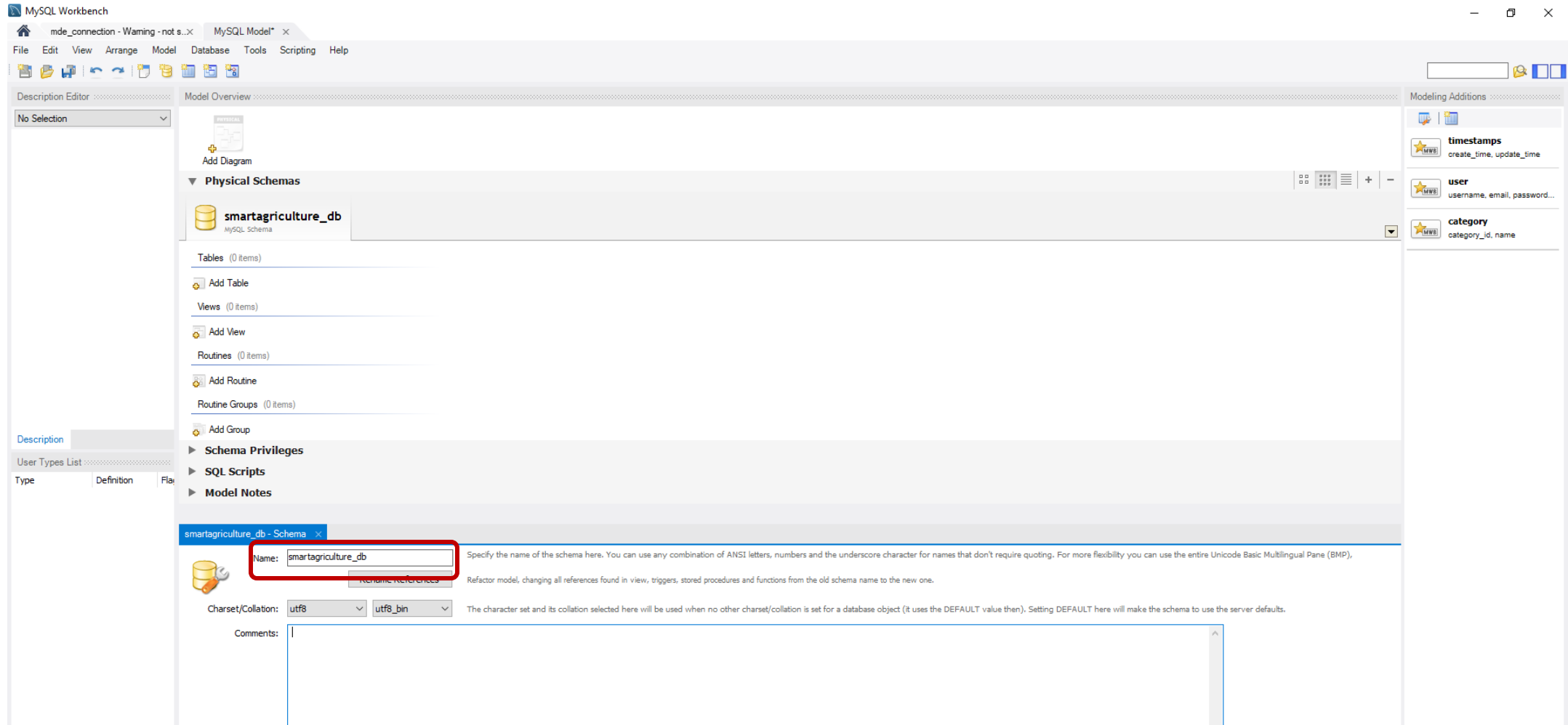
- Select File->New Model



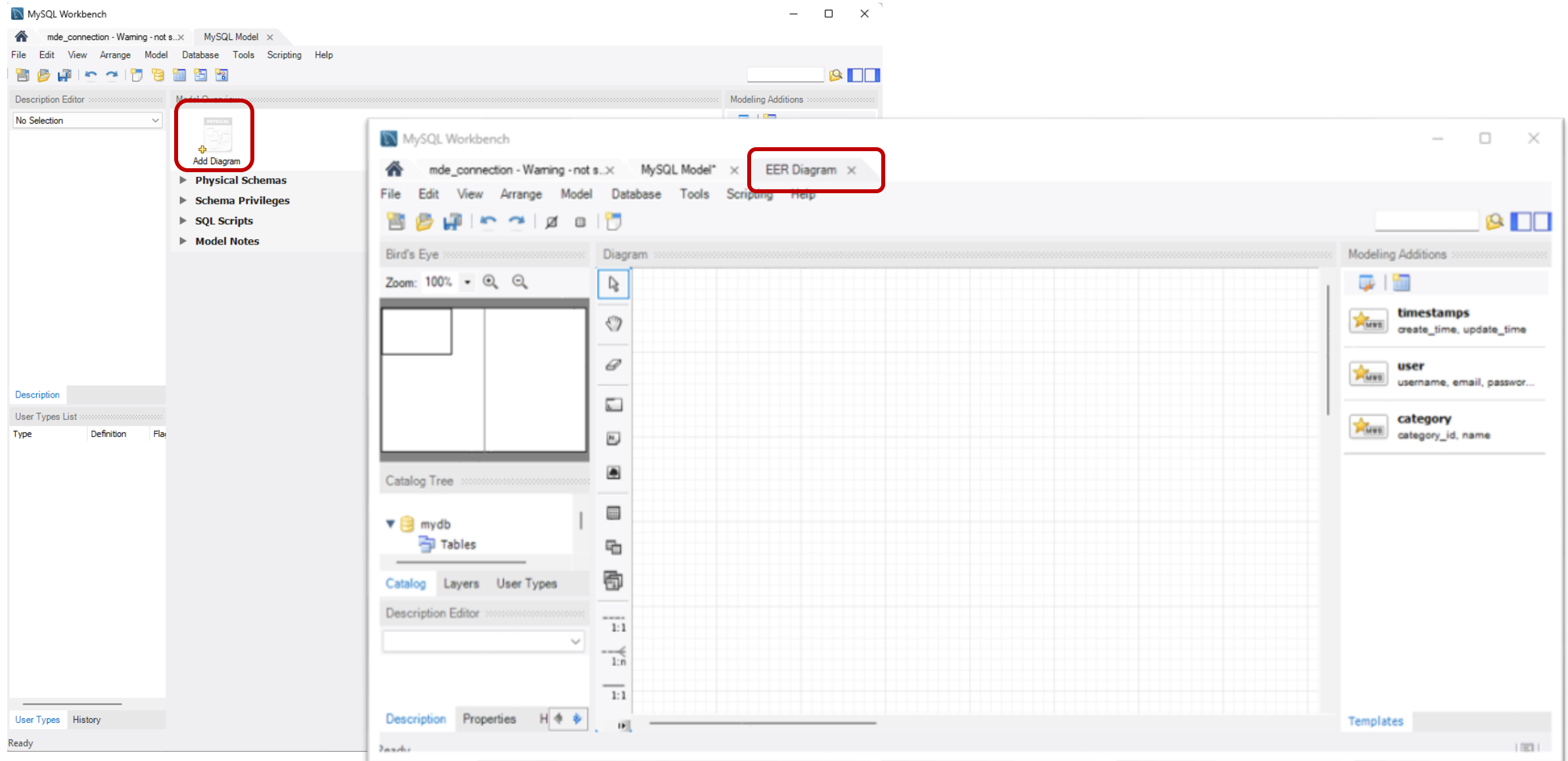
MODEL CREATION IN MySQL (forward engineering)



- Name it as *smartagriculture_db*



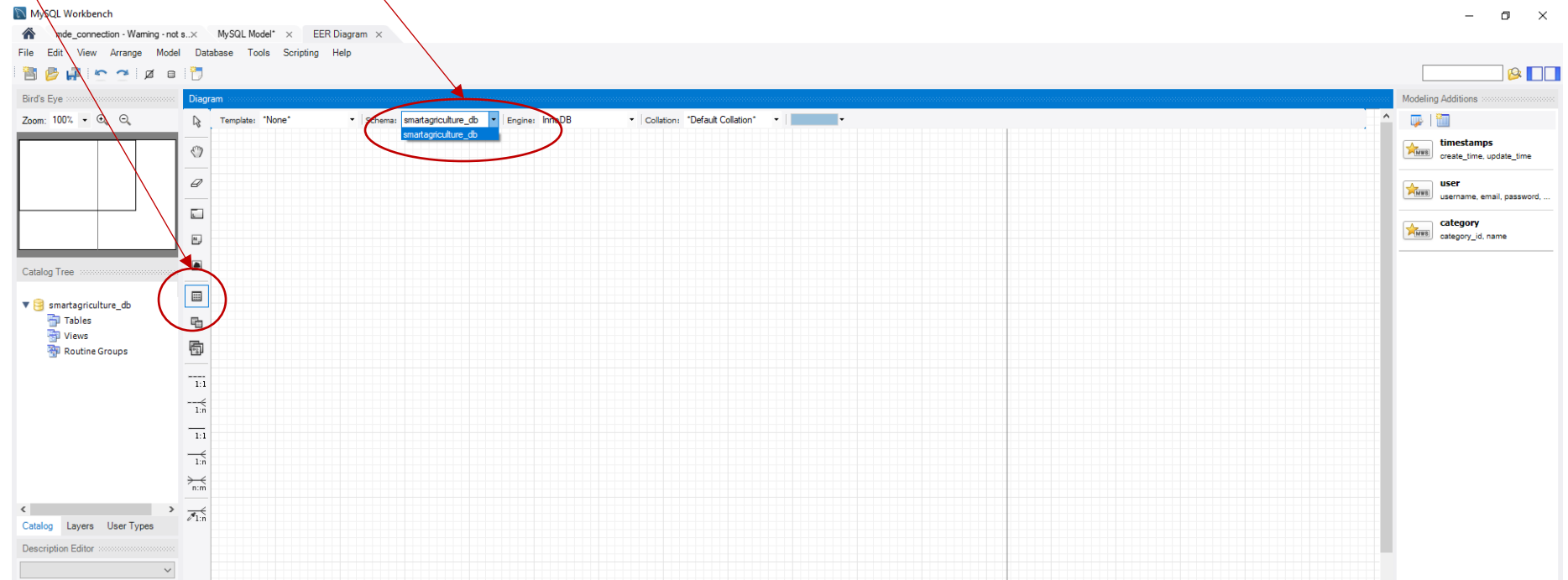
MODEL CREATION IN MySQL (forward engineering)



MODEL CREATION IN MySQL (forward engineering) – New Table

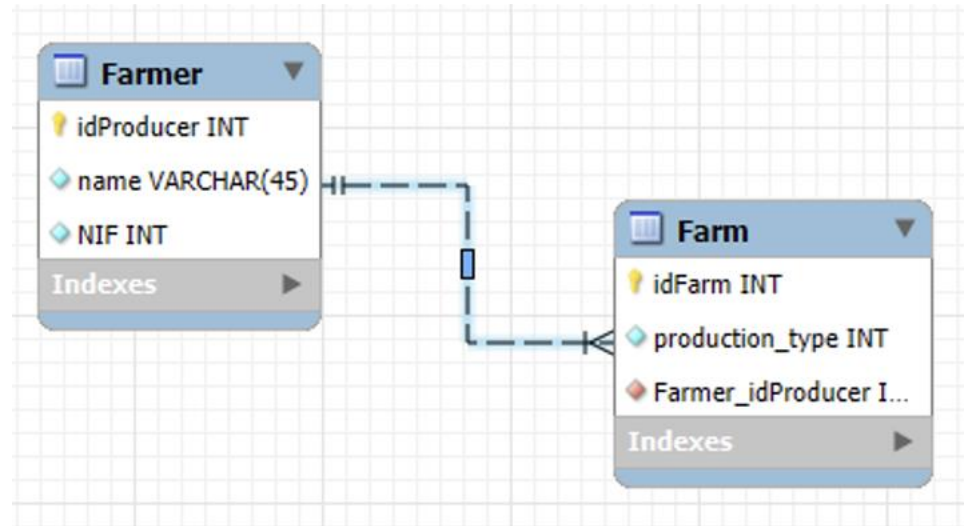


- Create new table
- Choose the appropriate schema



MODEL CREATION IN MySQL (forward engineering)

- Let's use a small model is just to illustrate the process
- Might need more attributes!



MODEL CREATION IN MySQL (forward engineering) – New Table



Zoom: 100%

Catalog Tree

- smartagriculture_db
 - Tables
 - Views
 - Routine Groups

Farmer - Table

Table Name: **Farmer** Schema: **smartagriculture_db**

Column Name	Datatype	PK	NN	UO	B	UN	ZF	AI	G	Default/Expression
idProducer	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NIF	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: **NIF**

Charset/Collation: **Default Charset** **Default Collation**

Comments:

Data Type: **INT**

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☒ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

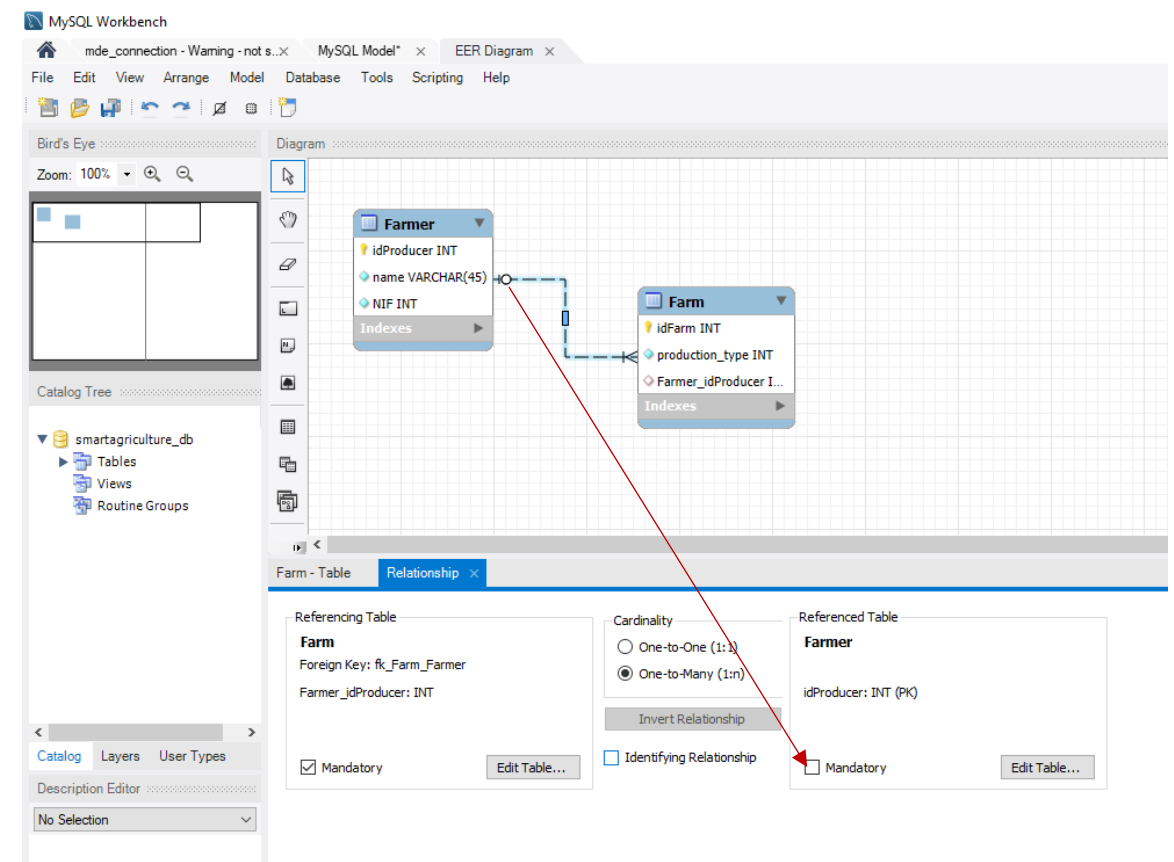
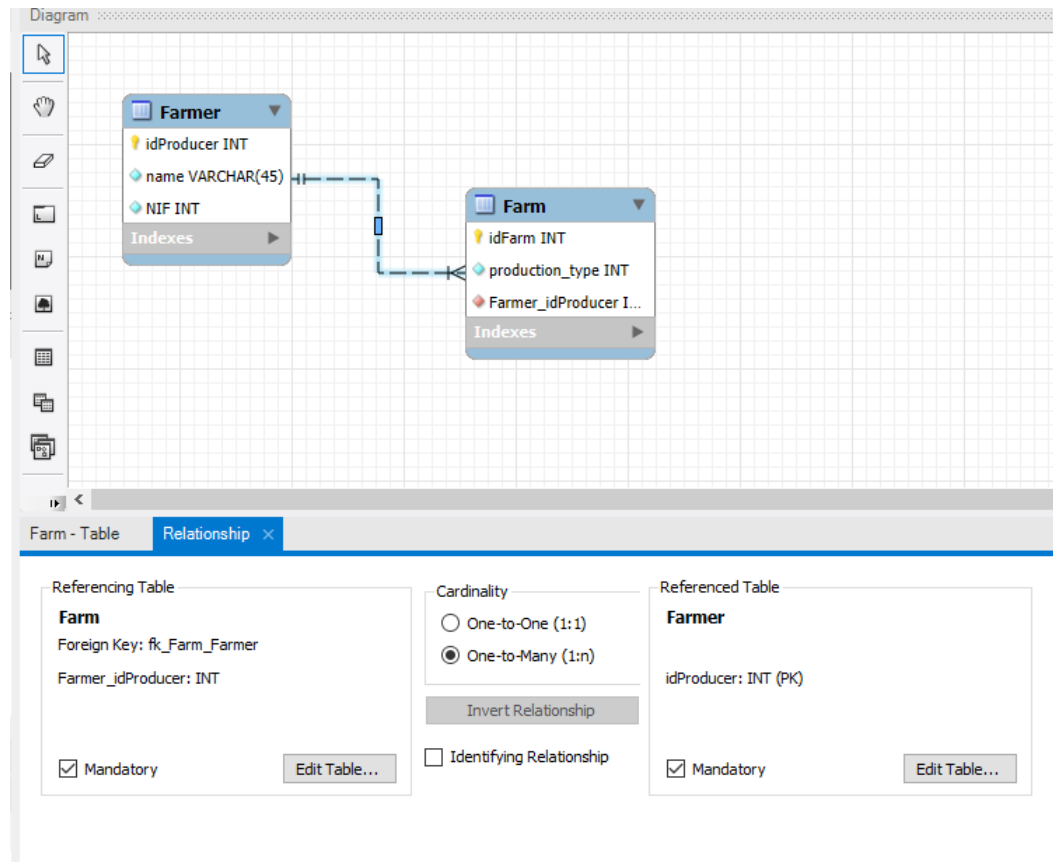
Attribute assignment

- Datatype
- Constraints

Check appropriate schema selection

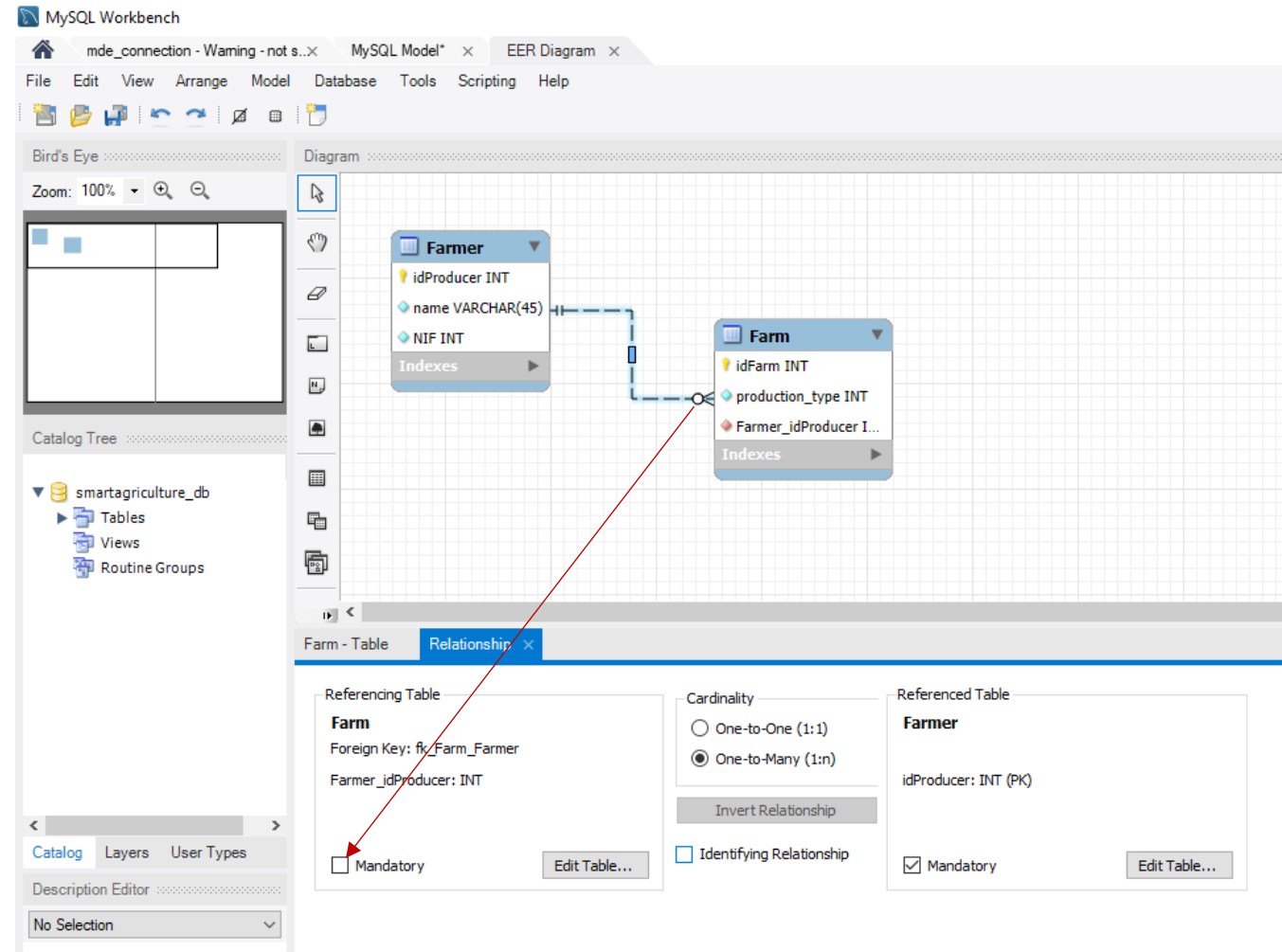
MODEL CREATION IN MySQL (forward engineering) – Relationships

- Default relationship: 1 → 1 or more



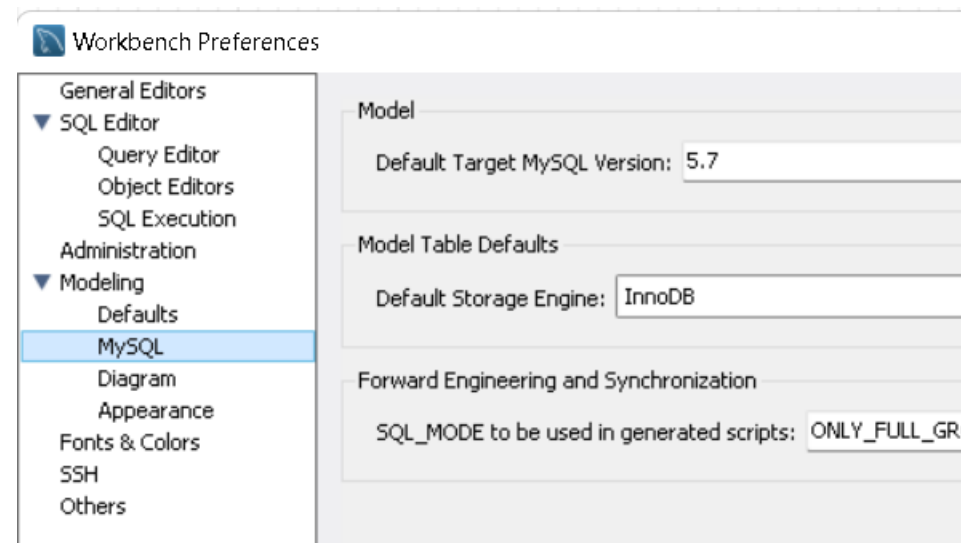
MODEL CREATION IN MySQL (forward engineering) – Relationships

- Default relationship: 1 → 0 or more

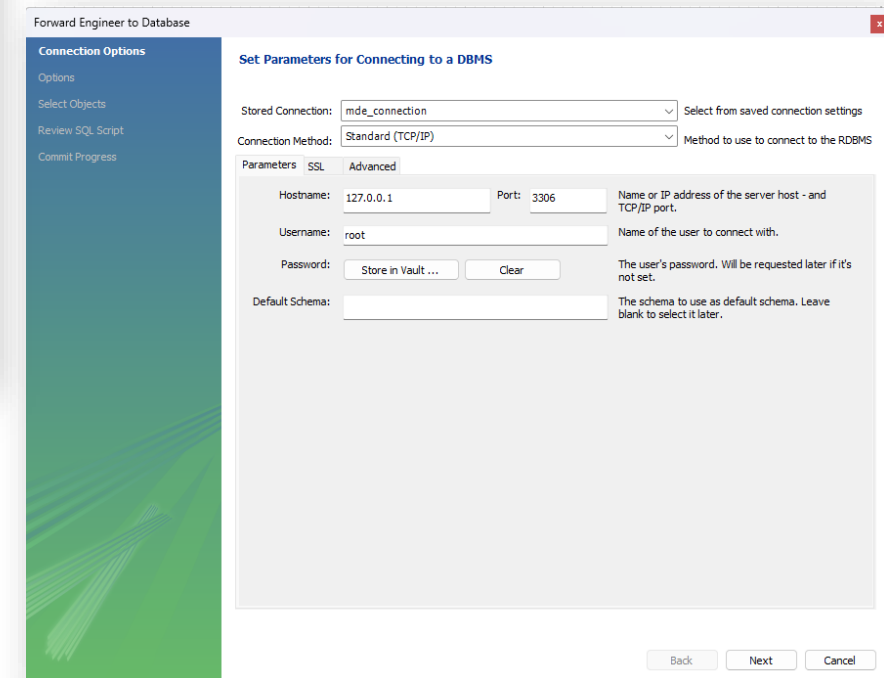
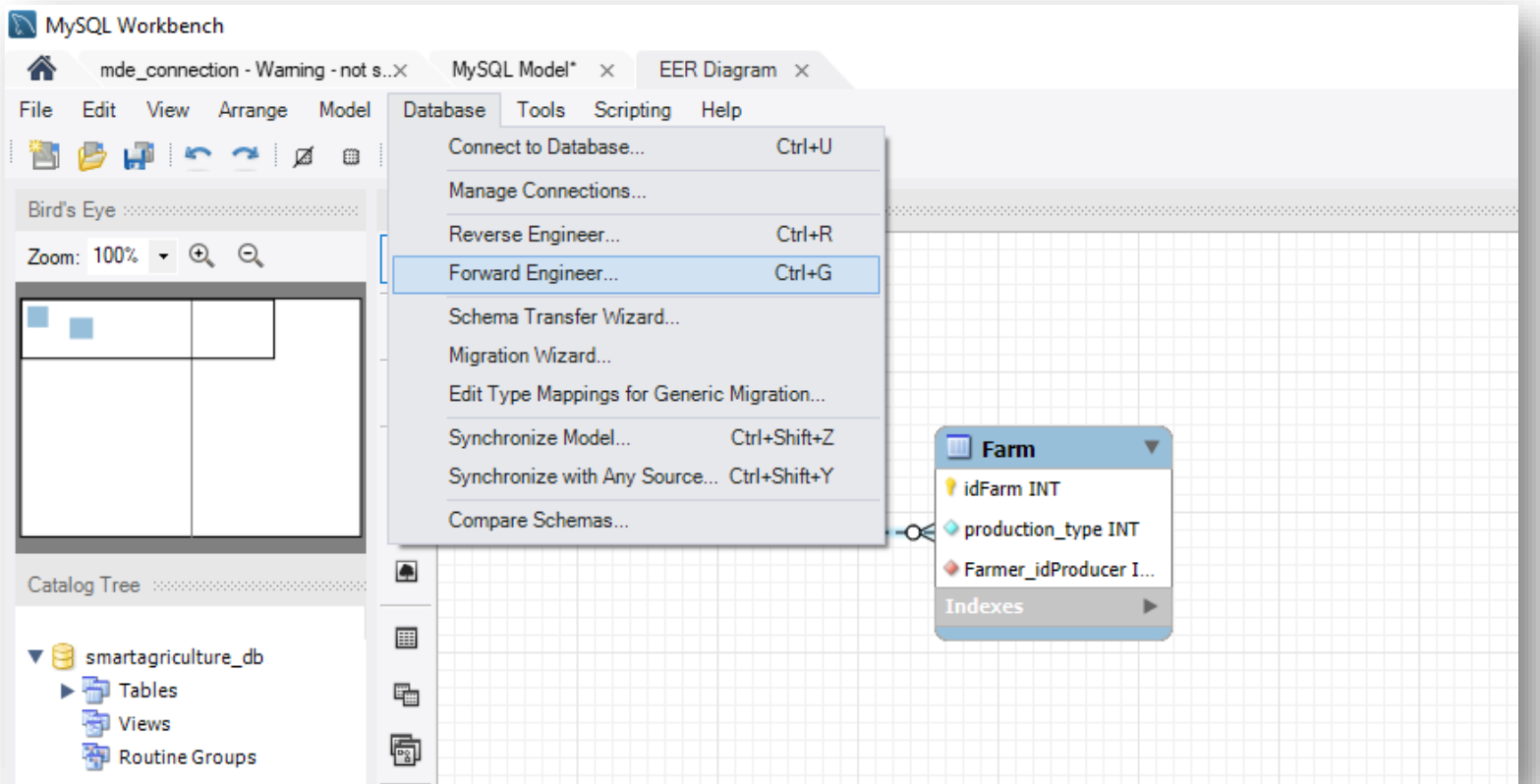


!!!! IMPORTANT -> CONFIGURATION

- In order to avoid a bug between MySQL versions, we need to go to menu Edit > Preferences > Modeling > MySQL.
- Change Default Target MySQL version to 5.7



FORWARD ENGINEERING



FORWARD ENGINEERING – SUGGESTED SETTINGS



Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Set Options for Database to be Created

Tables

- ☐ Skip creation of FOREIGN KEYS
- ☐ Skip creation of FK Indexes as well
- ☐ Generate separate CREATE INDEX statements
- ☐ Generate INSERT statements for tables
- ☐ Disable FK checks for INSERTs

Other Objects

- ☐ Don't create view placeholder tables
- ☐ Do not create users. Only create privileges (GRANTS)

Code Generation

- ☐ DROP objects before each CREATE object
- ☒ Generate DROP SCHEMA
- ☒ Omit schema qualifier in object names
- ☒ Generate USE statements
- ☐ Add SHOW WARNINGS after every DDL statement
- ☒ Include model attached scripts

Back Next Cancel

FORWARDING...



Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Select Objects to Forward Engineer

To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.

☒ Export MySQL Table Objects

2 Total Objects, 2 Selected

Show Filter

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Review the SQL Script to be Executed

This script will now be executed on the DB server to create your databases. You may make changes before executing.

```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY'
6
7  -----
8  -- Schema smartagriculture_db
9  -----
10 DROP SCHEMA IF EXISTS `smartagriculture_db` ;
11
12 -----
13 -- Schema smartagriculture_db
14 -----
15 CREATE SCHEMA IF NOT EXISTS `smartagriculture_db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
16 USE `smartagriculture_db` ;
17
18 -----
19 -- Table `Farmer`
20 -----
21 CREATE TABLE IF NOT EXISTS `Farmer` (
22   `idProducer` INT NOT NULL AUTO_INCREMENT,
23   `name` VARCHAR(45) NOT NULL,
24   `NIF` INT NOT NULL,
```

Save to File...

Copy to Clipboard

Back

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Forward Engineering Progress

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

☒ Connect to DBMS

☒ Execute Forward Engineered Script

☒ Read Back Changes Made by Server

☒ Save Synchronization State

Forward Engineer Finished Successfully

Show Logs

Back

Close

Cancel

BACK TO THE CONNECTION...



The screenshot shows the MySQL Workbench interface. At the top, a warning message says "Warning - n...". Below it, the "Navigator" panel on the left shows a tree of schemas. A red circle highlights the "smartagriculture_db" schema, which contains sub-items for "Tables", "Views", "Stored Procedures", and "Functions". The "Query" window in the center contains the following SQL commands:

```
1 • show databases;
2
3 • use evfleets_db
```

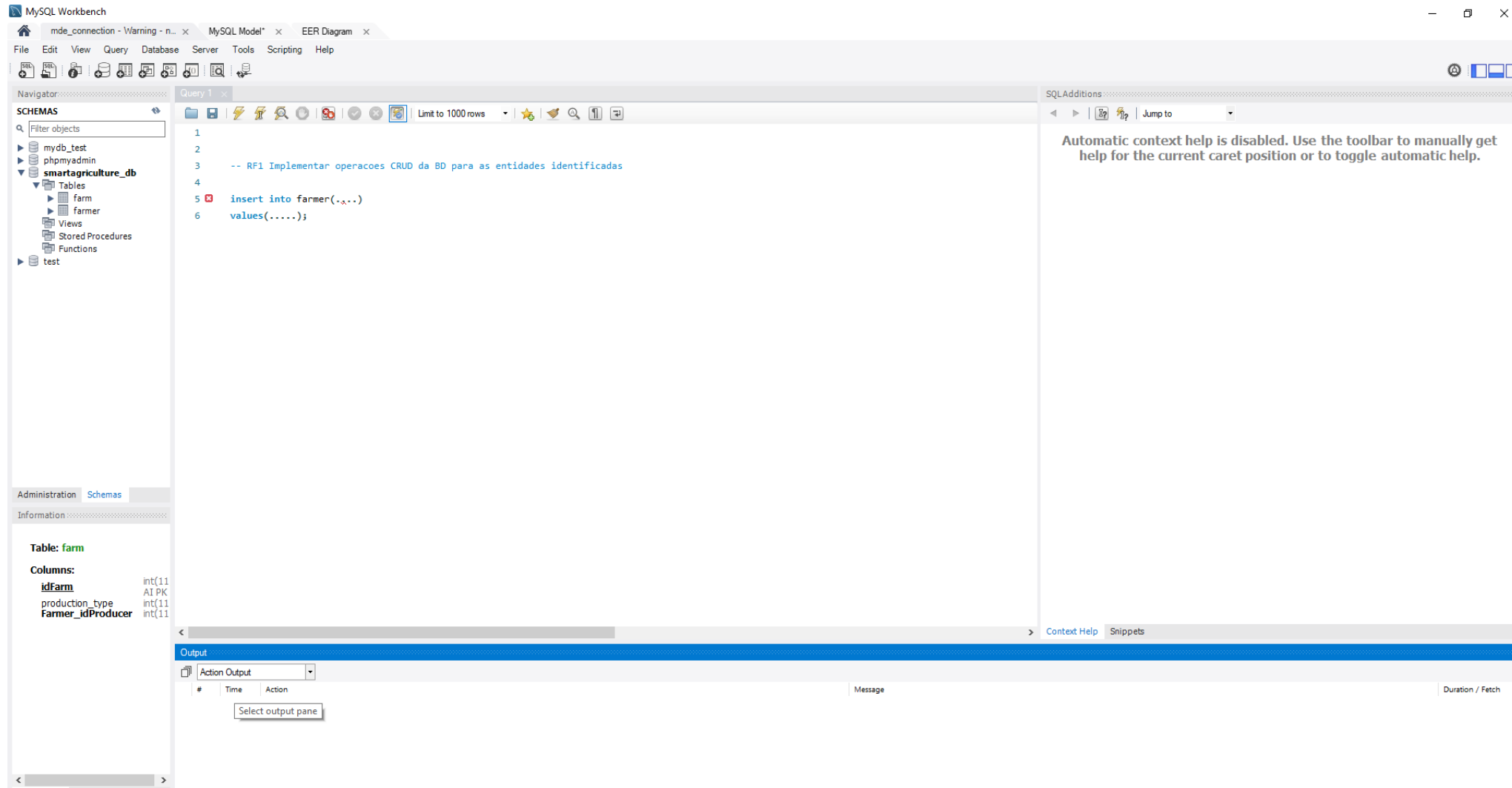
The "Output" panel at the bottom shows the results of the query execution:

#	Time	Action	Message	Duration / Fetch
1	17:51:00	DROP DATABASE 'evfleets_db'	1 row(s) affected	0.594 sec

At the bottom left, the status bar indicates "Query Completed".

- Check if the new schema is already in place (it might be needed a refresh...)

At this moment we are ready to start the development of the FRs!



- ❑ Implementation of the FRs: CRUD and Queries for consulting information.
- ❑ Joins and Views
- ❑ Aggregation functions.

Keep Up The
Good Work!