

# Übungsprotokoll

## INSY - Informationssysteme

	<b>Übungsdatum:</b> KW 40/2021 – KW 41/2021	<b>Klasse:</b> 3AHIT	<b>Name:</b> Felix Schneider
	<b>Abgabedatum:</b> 11.10.2021	<b>Gruppe:</b> INSY_2	<b>Note:</b>
<b>Leitung:</b> DI (FH) Alexander MESTL	<b>Mitübende:</b> -		
<b>Übungsbezeichnung:</b>  MySQL Radwerkstatt + Süßigkeitenfabrik - Modell			

**Inhaltsverzeichnis:**

1	Aufgabenstellung.....	3
2	Theoretische Grundlagen .....	4
3	Ergebnisse .....	5
4	Code.....	6
4.1	Radwerkstatt .....	6
4.2	Süßigkeitenfabrik.....	9
5	Kommentar.....	13

# 1 Aufgabenstellung

## 1.Beispiel) Radwerkstatt - 8.Punkte

Für eine Radwerkstatt soll zur Verwaltung eine Datenbank entwickelt werden.

- **EMPLOYEES:** Mitarbeiter sind eindeutig durch eine ID **EMPLOYEE\_ID** (int - not null, unique). Zusätzlich wird die Personalnummer **EMPLOYEE\_CODE** (varchar(12) - not null, unique), der Nachname **LAST\_NAME** (varchar(50) - not null), der Vorname **FIRST\_NAME** (varchar(50) - not null) und das Gehalt **SALARY** (int - not null) gespeichert.

- **BIKES:** Fahrräder sind eindeutig durch eine ID **BIKE\_ID** (int - not null, unique). Außerdem werden Marke **BRAND** (varchar(100) - not null) und Bezeichnung **NAME** (varchar(100) - not null) gespeichert. Jeder Mitarbeiter betreut mehrere Fahrräder. Jedes Fahrrad wird von einem Mitarbeiter betreut.

- **REPAIR\_SERVICE:** Jedes Service wird eindeutig durch eine ID **SERVICE\_ID** (int - not null, unique) identifiziert.

Jedes Service besteht mindestens aus einem Schritt (**WORKING\_PHASES**). Ein Arbeitsschritt wird immer einem einzelnen Service zugeordnet. Jeder Schritt hat eine eindeutige Nummer **PHASE\_ID** (int - not null, unique) und es werden eine Beschreibung **DESCRIPTION** (varchar(255) - not null) und die Kosten **PRICE** (int - not null) für den Schritt gespeichert. Im Zuge eines Services wird ein Fahrrad wieder in Stand gesetzt.

- **SPARE\_PARTS:** Es kann sein, dass für manche Arbeitsschritte Ersatzteile benötigt werden. Jedes Ersatzteil ist eindeutig durch eine Produktnummer **PART\_ID** (int - not null, unique). Außerdem werden der Lagerbestand **STORED\_UNITS** (int - not null) und der Preis **PRICE** (int - not null) gespeichert.



## 8.) Aufgabenblatt - Gruppe A(8.Punkte)



Kompetenzen ▾

**Relationale Modellierung:** Entwurf der Struktur einer relationalen Datenbank.

### 1.Beispiel) Süßigkeitenfabrik - 8.Punkte

Entwerfen Sie für die folgende Aufgabenstellung ein relationales Diagramm.

- **SWEETS:** Süßigkeiten werden eindeutig identifiziert durch eine ID **SWEET\_ID** (int - not null, unique). Zusätzlich wird ein Verkaufsnamen **NAME** (varchar(100) - not null, unique) und eine Verpackungsgröße **WRAPPING\_TYPE** (varchar(20) - not null, {DIN\_A, DIN\_A2, DIN\_B2, DIN\_B4}) (**E\_WRAPPING\_TYPES**) gespeichert. Außerdem sind die Kalorien **CALORIFIC\_VALUE** (int - not null) und die Produktionskosten **COST\_OF\_PRODUCTION** (int - not null) bekannt.

- **PRODUCT\_REALISATION\_PROCESSES:** Produktionsprozesse sind eindeutig durch eine ID **PROCESS\_ID** (int - not null, unique). Zusätzlich wird eine Bezeichnung **PROCESS\_NAME** (varchar(50) - not null, unique) gespeichert.

Es ist bekannt welche Süßigkeiten mit welchen Produktionsprozessen hergestellt werden können. Dabei kann ein Produktionsprozess nur einer einzigen Süßigkeit zugeordnet sein. Für eine Süßigkeit können jedoch mehrere Produktionsprozesse bekannt sein.

Produktionsprozesse bestehen aus mehreren Teilschritten (**SUBSTEPS**). Ein Teilschritt ist eindeutig durch eine ID **STEP\_ID** (int - not null, unique). Für jeden Teilschritt wird ebenfalls die Dauer **TIME\_DURATION** (int - not null) gespeichert. Ein Teilschritt kann in mehreren Produktionsprozessen enthalten sein. Für jeden Teilschritt wird ebenfalls gespeichert welcher Teilschritt **STEP\_INDEX** (int - not null) er im entsprechenden Produktionsprozess ist.

- **RAW\_MATERIALS:** Rohstoffe sind eindeutig durch eine ID **MATERIAL\_ID** (int - not null, unique). Es ist auch eine Bezeichnung **NAME** (varchar(100) - not null, unique) und die Qualitätsstufe **QUALITY** (varchar(2) - not null, {Q1, Q2, Q3})

(**E\_QUALITY\_TYPES**) bekannt. Manche Rohstoffe können durch andere Rohstoffe substituiert werden. Dabei kann ein Rohstoff immer durch mehrere andere Rohstoffe ersetzt werden. Es ist auch bekannt welche Menge **AMOUNT** (int - not null) von welchem Rohstoffen in einem Teilschritt eines Produktionsprozesses verbraucht werden.

- **MACHINES:** Maschinen sind eindeutig durch eine ID **MACHINE\_ID** (int - not null, unique). Für Maschinen wird ebenfalls der Hersteller **PRODUCER** (varchar(100) - not null) und eine Bezeichnung **NAME** (varchar(255) - not null, unique) gespeichert.

Es ist bekannt welche Maschine für welchen Teilschritt eines Verarbeitungsprozesses benötigt wird. Zur Durchführung eines Teilschritts können dabei mehrere Maschinen eingesetzt werden. Eine Maschine kann für die Durchführung mehrerer Teilschritte zum Einsatz kommen.

- **EMPLOYEES:** Mitarbeiter sind eindeutig durch eine ID **EMPLOYEE\_ID** (int - not null, unique). Für Mitarbeiter wird ebenfalls der Vorname **FIRST\_NAME** (varchar(50) - not null) und Nachname **LAST\_NAME** (varchar(50) - not null) sowie das Gehalt **SALARY** (int - not null) gespeichert. Arbeiter und Techniker sind Mitarbeiter. Für Mitarbeiter ist bekannt welche Maschine, welchen Teilschritts welches Produktionsprozesses sie betreuen.

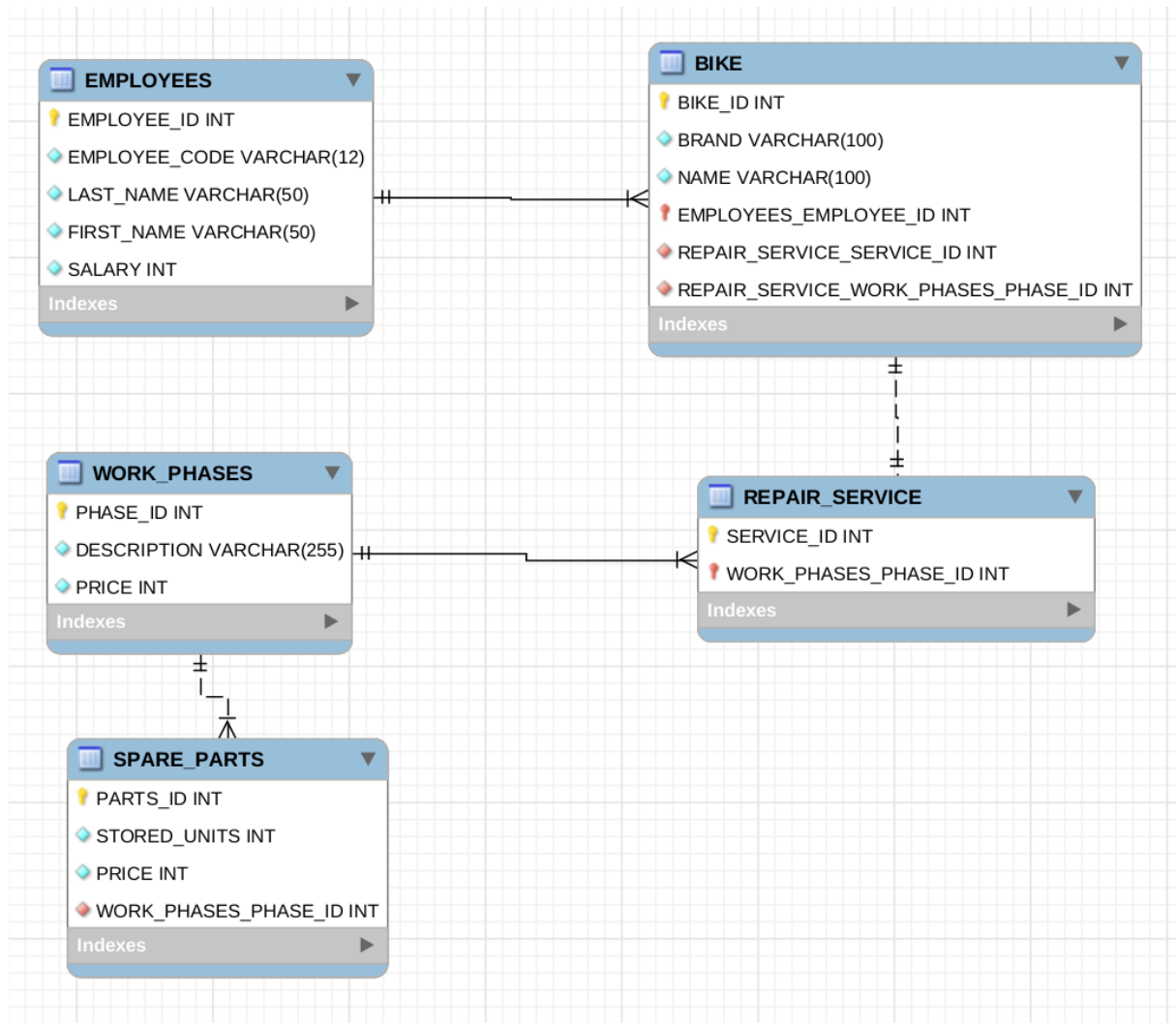
**Hinweis:** Verwenden Sie im Falle der Mitarbeiter Singletable Vererbung.

□

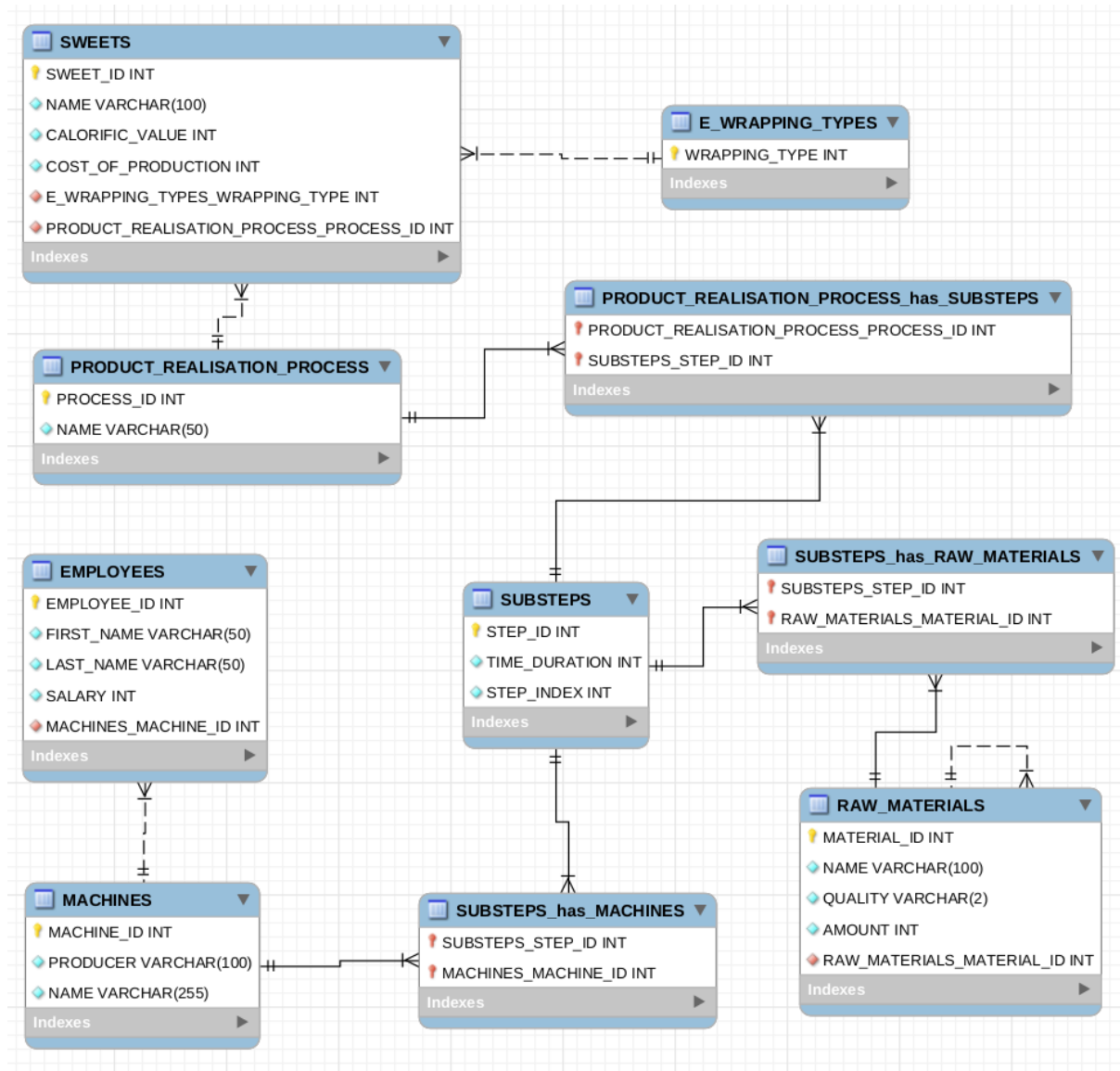
## 2 Theoretische Grundlagen

➔ siehe Datenmodellierung Informationssysteme – Dipl.-Ing. Msc. Paul Panhofer BSc.

### 3 Ergebnisse







## 4 Code

### 4.1 Radwerkstatt

```
-- MySQL Script generated by MySQL Workbench
-- Mo 11 Okt 2021 18:29:42 CEST
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,
ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema mydb
-- -----
```

```
-- -----  
-- Schema mydb  
-- -----  
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;  
USE `mydb` ;  
  
-- -----  
-- Table `mydb`.`EMPLOYEES`  
-- -----  
CREATE TABLE IF NOT EXISTS `mydb`.`EMPLOYEES` (  
  `EMPLOYEE_ID` INT NOT NULL,  
  `EMPLOYEE_CODE` VARCHAR(12) NOT NULL,  
  `LAST_NAME` VARCHAR(50) NOT NULL,  
  `FIRST_NAME` VARCHAR(50) NOT NULL,  
  `SALARY` INT NOT NULL,  
  PRIMARY KEY (`EMPLOYEE_ID`),  
  UNIQUE INDEX `EMPLOYEE_ID_UNIQUE` (`EMPLOYEE_ID` ASC) VISIBLE,  
  UNIQUE INDEX `EMPLOYEE_CODE_UNIQUE` (`EMPLOYEE_CODE` ASC) VISIBLE)  
ENGINE = InnoDB;  
  
-- -----  
-- Table `mydb`.`WORK_PHASES`  
-- -----  
CREATE TABLE IF NOT EXISTS `mydb`.`WORK_PHASES` (  
  `PHASE_ID` INT NOT NULL,  
  `DESCRIPTION` VARCHAR(255) NOT NULL,  
  `PRICE` INT NOT NULL,  
  PRIMARY KEY (`PHASE_ID`),  
  UNIQUE INDEX `PHASE_ID_UNIQUE` (`PHASE_ID` ASC) VISIBLE)  
ENGINE = InnoDB;  
  
-- -----  
-- Table `mydb`.`REPAIR_SERVICE`  
-- -----  
CREATE TABLE IF NOT EXISTS `mydb`.`REPAIR_SERVICE` (  
  `SERVICE_ID` INT NOT NULL,  
  `WORK_PHASES_PHASE_ID` INT NOT NULL,  
  PRIMARY KEY (`SERVICE_ID`, `WORK_PHASES_PHASE_ID`),  
  UNIQUE INDEX `SERVICE_ID_UNIQUE` (`SERVICE_ID` ASC) VISIBLE,  
  INDEX `fk_REPAIR_SERVICE_WORK_PHASES1_idx` (`WORK_PHASES_PHASE_ID` ASC)  
VISIBLE,  
  CONSTRAINT `fk_REPAIR_SERVICE_WORK_PHASES1`  
    FOREIGN KEY (`WORK_PHASES_PHASE_ID`)  
    REFERENCES `mydb`.`WORK_PHASES` (`PHASE_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

-----
-- Table `mydb`.`BIKE`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`BIKE` (
  `BIKE_ID` INT NOT NULL,
  `BRAND` VARCHAR(100) NOT NULL,
  `NAME` VARCHAR(100) NOT NULL,
  `EMPLOYEES_EMPLOYEE_ID` INT NOT NULL,
  `REPAIR_SERVICE_SERVICE_ID` INT NOT NULL,
  `REPAIR_SERVICE_WORK_PHASES_PHASE_ID` INT NOT NULL,
  PRIMARY KEY (`BIKE_ID`, `EMPLOYEES_EMPLOYEE_ID`),
  UNIQUE INDEX `BIKE_ID_UNIQUE` (`BIKE_ID` ASC) VISIBLE,
  INDEX `fk_BIKE_EMPLOYEES_idx` (`EMPLOYEES_EMPLOYEE_ID` ASC) VISIBLE,
  INDEX `fk_BIKE_REPAIR_SERVICE1_idx` (`REPAIR_SERVICE_SERVICE_ID` ASC,
  `REPAIR_SERVICE_WORK_PHASES_PHASE_ID` ASC) VISIBLE,
  CONSTRAINT `fk_BIKE_EMPLOYEES`
    FOREIGN KEY (`EMPLOYEES_EMPLOYEE_ID`)
      REFERENCES `mydb`.`EMPLOYEES` (`EMPLOYEE_ID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_BIKE_REPAIR_SERVICE1`
    FOREIGN KEY (`REPAIR_SERVICE_SERVICE_ID`,
  `REPAIR_SERVICE_WORK_PHASES_PHASE_ID`)
      REFERENCES `mydb`.`REPAIR_SERVICE` (`SERVICE_ID`, `WORK_PHASES_PHASE_ID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`SPARE_PARTS`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`SPARE_PARTS` (
  `PARTS_ID` INT NOT NULL,
  `STORED_UNITS` INT NOT NULL,
  `PRICE` INT NOT NULL,
  `WORK_PHASES_PHASE_ID` INT NOT NULL,
  PRIMARY KEY (`PARTS_ID`),
  UNIQUE INDEX `PARTS_ID_UNIQUE` (`PARTS_ID` ASC) VISIBLE,
  INDEX `fk_SPARE_PARTS_WORK_PHASES1_idx` (`WORK_PHASES_PHASE_ID` ASC)
VISIBLE,
  CONSTRAINT `fk_SPARE_PARTS_WORK_PHASES1`
    FOREIGN KEY (`WORK_PHASES_PHASE_ID`)
      REFERENCES `mydb`.`WORK_PHASES` (`PHASE_ID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## 4.2 Süßigkeitenfabrik

```
-- MySQL Script generated by MySQL Workbench  
-- Mo 11 Okt 2021 18:03:24 CEST  
-- Model: New Model    Version: 1.0  
-- MySQL Workbench Forward Engineering  
  
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;  
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,  
ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';  
  
--  
-- Schema mydb  
--  
--  
-- Schema mydb  
--  
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;  
USE `mydb` ;  
  
--  
-- Table `mydb`.`table1`  
--  
CREATE TABLE IF NOT EXISTS `mydb`.`table1` (  
)  
ENGINE = InnoDB;  
  
--  
-- Table `mydb`.`E_WRAPPING_TYPES`  
--  
CREATE TABLE IF NOT EXISTS `mydb`.`E_WRAPPING_TYPES` (  
  `WRAPPING_TYPE` INT NOT NULL,  
  PRIMARY KEY (`WRAPPING_TYPE`))  
ENGINE = InnoDB;  
  
--  
-- Table `mydb`.`PRODUCT_REALISATION_PROCESS`  
--  
CREATE TABLE IF NOT EXISTS `mydb`.`PRODUCT_REALISATION_PROCESS` (  
  `PROCESS_ID` INT NOT NULL,  
  `NAME` VARCHAR(50) NOT NULL,
```

```

PRIMARY KEY (`PROCESS_ID`),
UNIQUE INDEX `PROCESS_ID_UNIQUE` (`PROCESS_ID` ASC) VISIBLE,
UNIQUE INDEX `NAME_UNIQUE` (`NAME` ASC) VISIBLE)
ENGINE = InnoDB;

-----
-- Table `mydb`.`SWEETS`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`SWEETS` (
  `SWEET_ID` INT NOT NULL,
  `NAME` VARCHAR(100) NOT NULL,
  `CALORIFIC_VALUE` INT NOT NULL,
  `COST_OF_PRODUCTION` INT NOT NULL,
  `E_WRAPPING_TYPES_WRAPPING_TYPE` INT NOT NULL,
  `PRODUCT_REALISATION_PROCESS_PROCESS_ID` INT NOT NULL,
  PRIMARY KEY (`SWEET_ID`),
  UNIQUE INDEX `SWEET_ID_UNIQUE` (`SWEET_ID` ASC) VISIBLE,
  UNIQUE INDEX `NAME_UNIQUE` (`NAME` ASC) VISIBLE,
  INDEX `fk_SWEETS_E_WRAPPING_TYPES_idx` (`E_WRAPPING_TYPES_WRAPPING_TYPE`
ASC) VISIBLE,
  INDEX `fk_SWEETS_PRODUCT_REALISATION_PROCESS1_idx`
(`PRODUCT_REALISATION_PROCESS_PROCESS_ID` ASC) VISIBLE,
  CONSTRAINT `fk_SWEETS_E_WRAPPING_TYPES`
    FOREIGN KEY (`E_WRAPPING_TYPES_WRAPPING_TYPE`)
    REFERENCES `mydb`.`E_WRAPPING_TYPES` (`WRAPPING_TYPE`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_SWEETS_PRODUCT_REALISATION_PROCESS1`
    FOREIGN KEY (`PRODUCT_REALISATION_PROCESS_PROCESS_ID`)
    REFERENCES `mydb`.`PRODUCT_REALISATION_PROCESS` (`PROCESS_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`SUBSTEPS`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`SUBSTEPS` (
  `STEP_ID` INT NOT NULL,
  `TIME_DURATION` INT NOT NULL,
  `STEP_INDEX` INT NOT NULL,
  PRIMARY KEY (`STEP_ID`),
  UNIQUE INDEX `STEP_ID_UNIQUE` (`STEP_ID` ASC) VISIBLE)
ENGINE = InnoDB;

-----
-- Table `mydb`.`RAW_MATERIALS`

```

```
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`RAW_MATERIALS` (  
  `MATERIAL_ID` INT NOT NULL,  
  `NAME` VARCHAR(100) NOT NULL,  
  `QUALITY` VARCHAR(2) NOT NULL,  
  `AMOUNT` INT NOT NULL,  
  `RAW_MATERIALS_MATERIAL_ID` INT NOT NULL,  
  PRIMARY KEY (`MATERIAL_ID`),  
  UNIQUE INDEX `MATERIAL_ID_UNIQUE` (`MATERIAL_ID` ASC) VISIBLE,  
  UNIQUE INDEX `NAME_UNIQUE` (`NAME` ASC) VISIBLE,  
  INDEX `fk_RAW_MATERIALS_RAW_MATERIALS1_idx` (`RAW_MATERIALS_MATERIAL_ID`  
ASC) VISIBLE,  
  CONSTRAINT `fk_RAW_MATERIALS_RAW_MATERIALS1`  
    FOREIGN KEY (`RAW_MATERIALS_MATERIAL_ID`)  
    REFERENCES `mydb`.`RAW_MATERIALS` (`MATERIAL_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`MACHINES`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`MACHINES` (  
  `MACHINE_ID` INT NOT NULL,  
  `PRODUCER` VARCHAR(100) NOT NULL,  
  `NAME` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`MACHINE_ID`),  
  UNIQUE INDEX `MACHINE_ID_UNIQUE` (`MACHINE_ID` ASC) VISIBLE,  
  UNIQUE INDEX `NAME_UNIQUE` (`NAME` ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`EMPLOYEES`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`EMPLOYEES` (  
  `EMPLOYEE_ID` INT NOT NULL,  
  `FIRST_NAME` VARCHAR(50) NOT NULL,  
  `LAST_NAME` VARCHAR(50) NOT NULL,  
  `SALARY` INT NOT NULL,  
  `MACHINES_MACHINE_ID` INT NOT NULL,  
  PRIMARY KEY (`EMPLOYEE_ID`),  
  UNIQUE INDEX `EMPLOYEE_ID_UNIQUE` (`EMPLOYEE_ID` ASC) VISIBLE,  
  INDEX `fk_EMPLOYEES_MACHINES1_idx` (`MACHINES_MACHINE_ID` ASC) VISIBLE,  
  CONSTRAINT `fk_EMPLOYEES_MACHINES1`  
    FOREIGN KEY (`MACHINES_MACHINE_ID`)  
    REFERENCES `mydb`.`MACHINES` (`MACHINE_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;

-----
-- Table `mydb`.`PRODUCT_REALISATION_PROCESS_has_SUBSTEPS`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`PRODUCT_REALISATION_PROCESS_has_SUBSTEPS` (
  `PRODUCT_REALISATION_PROCESS_PROCESS_ID` INT NOT NULL,
  `SUBSTEPS_STEP_ID` INT NOT NULL,
  PRIMARY KEY (`PRODUCT_REALISATION_PROCESS_PROCESS_ID`, `SUBSTEPS_STEP_ID`),
  INDEX `fk_PRODUCT_REALISATION_PROCESS_has_SUBSTEPS_SUBSTEPS1_idx`
(`SUBSTEPS_STEP_ID` ASC) VISIBLE,
  INDEX `fk_PRODUCT_REALISATION_PROCESS_has_SUBSTEPS_PRODUCT_REALISA_idx`
(`PRODUCT_REALISATION_PROCESS_PROCESS_ID` ASC) VISIBLE,
  CONSTRAINT `fk_PRODUCT_REALISATION_PROCESS_has_SUBSTEPS_PRODUCT_REALISATI1`
  FOREIGN KEY (`PRODUCT_REALISATION_PROCESS_PROCESS_ID`)
  REFERENCES `mydb`.`PRODUCT_REALISATION_PROCESS` (`PROCESS_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_PRODUCT_REALISATION_PROCESS_has_SUBSTEPS_SUBSTEPS1`
  FOREIGN KEY (`SUBSTEPS_STEP_ID`)
  REFERENCES `mydb`.`SUBSTEPS` (`STEP_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`SUBSTEPS_has_MACHINES`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`SUBSTEPS_has_MACHINES` (
  `SUBSTEPS_STEP_ID` INT NOT NULL,
  `MACHINES_MACHINE_ID` INT NOT NULL,
  PRIMARY KEY (`SUBSTEPS_STEP_ID`, `MACHINES_MACHINE_ID`),
  INDEX `fk_SUBSTEPS_has_MACHINES_MACHINES1_idx` (`MACHINES_MACHINE_ID` ASC)
VISIBLE,
  INDEX `fk_SUBSTEPS_has_MACHINES_SUBSTEPS1_idx` (`SUBSTEPS_STEP_ID` ASC)
VISIBLE,
  CONSTRAINT `fk_SUBSTEPS_has_MACHINES_SUBSTEPS1`
  FOREIGN KEY (`SUBSTEPS_STEP_ID`)
  REFERENCES `mydb`.`SUBSTEPS` (`STEP_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_SUBSTEPS_has_MACHINES_MACHINES1`
  FOREIGN KEY (`MACHINES_MACHINE_ID`)
  REFERENCES `mydb`.`MACHINES` (`MACHINE_ID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`SUBSTEPS_has_RAW_MATERIALS`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`SUBSTEPS_has_RAW_MATERIALS` (  
  `SUBSTEPS_STEP_ID` INT NOT NULL,  
  `RAW_MATERIALS_MATERIAL_ID` INT NOT NULL,  
  PRIMARY KEY (`SUBSTEPS_STEP_ID`, `RAW_MATERIALS_MATERIAL_ID`),  
  INDEX `fk_SUBSTEPS_has_RAW_MATERIALS_RAW_MATERIALS1_idx`  
  (`RAW_MATERIALS_MATERIAL_ID` ASC) VISIBLE,  
  INDEX `fk_SUBSTEPS_has_RAW_MATERIALS_SUBSTEPS1_idx` (`SUBSTEPS_STEP_ID` ASC)  
  VISIBLE,  
  CONSTRAINT `fk_SUBSTEPS_has_RAW_MATERIALS_SUBSTEPS1`  
    FOREIGN KEY (`SUBSTEPS_STEP_ID`)  
    REFERENCES `mydb`.`SUBSTEPS` (`STEP_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_SUBSTEPS_has_RAW_MATERIALS_RAW_MATERIALS1`  
    FOREIGN KEY (`RAW_MATERIALS_MATERIAL_ID`)  
    REFERENCES `mydb`.`RAW_MATERIALS` (`MATERIAL_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## 5 Kommentar

Was ist eine Singletable Vererbung (Süßigkeitenfabrik – Employee)?