# SQL script bilag

CREATE SCHEMA IF NOT EXISTS `Polygon` DEFAULT CHARACTER SET utf8 ;

USE `Polygon` ;

CREATE TABLE IF NOT EXISTS `Polygon`.`Zipcode` (

`zipId` INT NOT NULL AUTO\_INCREMENT,

`zip` INT NULL,

`city` VARCHAR(45) NULL,

PRIMARY KEY (`zipId`))

CREATE TABLE IF NOT EXISTS `Polygon`.`userAddress` (

`addressId` INT NOT NULL AUTO\_INCREMENT,

`addressLine` VARCHAR(200) NOT NULL,

`Zipcode\_zipId` INT NOT NULL,

PRIMARY KEY (`addressId`),

INDEX `fk\_userAddress\_Zipcode1\_idx` (`Zipcode\_zipId` ASC),

CONSTRAINT `fk\_userAddress\_Zipcode1`

FOREIGN KEY (`Zipcode\_zipId`)

REFERENCES `Polygon`.`Zipcode` (`zipId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`User` (

`userId` INT NOT NULL AUTO\_INCREMENT,

`firstname` VARCHAR(45) NOT NULL,

`lastname` VARCHAR(45) NOT NULL,

`phone` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`userAddress\_addressId` INT NOT NULL,

PRIMARY KEY (`userId`),

UNIQUE INDEX `idUser\_UNIQUE` (`userId` ASC),

INDEX `fk\_User\_userAddress1\_idx` (`userAddress\_addressId` ASC),

CONSTRAINT `fk\_User\_userAddress1`

FOREIGN KEY (`userAddress\_addressId`)

REFERENCES `Polygon`.`userAddress` (`addressId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Address` (

`addressId` INT NOT NULL AUTO\_INCREMENT,

`addressline` VARCHAR(200) NOT NULL,

`zipcode\_addressId` INT NOT NULL,

PRIMARY KEY (`addressId`),

INDEX `fk\_Address\_zipcode\_idx` (`zipcode\_addressId` ASC),

CONSTRAINT `fk\_Address\_zipcode`

FOREIGN KEY (`zipcode\_addressId`)

REFERENCES `Polygon`.`Zipcode` (`zipId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Building` (

`buildingId` INT NOT NULL AUTO\_INCREMENT,

`hidden` INT NULL DEFAULT 0,

`Address\_addressId` INT NOT NULL,

`rapportURL` VARCHAR(250) NOT NULL,

`User\_userId` INT NOT NULL,

`buildingName` VARCHAR(150) NULL,

PRIMARY KEY (`buildingId`),

UNIQUE INDEX `buildingId\_UNIQUE` (`buildingId` ASC),

INDEX `fk\_Building\_Address1\_idx` (`Address\_addressId` ASC),

INDEX `fk\_Building\_User1\_idx` (`User\_userId` ASC),

CONSTRAINT `fk\_Building\_Address1`

FOREIGN KEY (`Address\_addressId`)

REFERENCES `Polygon`.`Address` (`addressId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Building\_User1`

FOREIGN KEY (`User\_userId`)

REFERENCES `Polygon`.`User` (`userId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Picture` (

`pictureId` INT NOT NULL AUTO\_INCREMENT,

`pictureURL` VARCHAR(250) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`pictureId`),

INDEX `fk\_picture\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_picture\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Document` (

`documentId` INT NOT NULL AUTO\_INCREMENT,

`file` MEDIUMBLOB NOT NULL,

`note` VARCHAR(250) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`documentId`),

INDEX `fk\_Docs\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_Docs\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Damage` (

`damageId` INT NOT NULL AUTO\_INCREMENT,

`room` VARCHAR(150) NULL,

`comments` INT NULL,

`roomDamaged` INT NULL,

`when` VARCHAR(150) NULL,

`where` VARCHAR(150) NULL,

`whatHappend` VARCHAR(150) NULL,

`whatRepaired` VARCHAR(150) NULL,

`damageNr` INT NULL,

`other` VARCHAR(150) NULL,

`Building\_buildingId` INT NOT NULL,

`categorized` INT NULL,

PRIMARY KEY (`damageId`),

UNIQUE INDEX `damageId\_UNIQUE` (`damageId` ASC),

INDEX `fk\_Damage\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_Damage\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Request` (

`requestId` INT NOT NULL AUTO\_INCREMENT,

`type` VARCHAR(100) NULL,

PRIMARY KEY (`requestId`),

UNIQUE INDEX `requestId\_UNIQUE` (`requestId` ASC))

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `Polygon`.`Request\_has\_Building` (

`Request\_requestId` INT NOT NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`Request\_requestId`, `Building\_buildingId`),

INDEX `fk\_Request\_has\_Building\_Building1\_idx` (`Building\_buildingId` ASC),

INDEX `fk\_Request\_has\_Building\_Request1\_idx` (`Request\_requestId` ASC),

CONSTRAINT `fk\_Request\_has\_Building\_Request1`

FOREIGN KEY (`Request\_requestId`)

REFERENCES `Polygon`.`Request` (`requestId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Request\_has\_Building\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Newsletter` (

`newsletterId` INT NOT NULL AUTO\_INCREMENT,

`User\_userId` INT NOT NULL,

PRIMARY KEY (`newsletterId`),

UNIQUE INDEX `newsletterId\_UNIQUE` (`newsletterId` ASC),

INDEX `fk\_Newsletter\_User1\_idx` (`User\_userId` ASC),

CONSTRAINT `fk\_Newsletter\_User1`

FOREIGN KEY (`User\_userId`)

REFERENCES `Polygon`.`User` (`userId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Login` (

`loginId` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(30) NOT NULL,

`password` VARCHAR(30) NOT NULL,

`rank` INT NULL,

`User\_userId` INT NOT NULL,

PRIMARY KEY (`loginId`),

UNIQUE INDEX `loginId\_UNIQUE` (`loginId` ASC),

UNIQUE INDEX `username\_UNIQUE` (`username` ASC),

UNIQUE INDEX `password\_UNIQUE` (`password` ASC),

INDEX `fk\_Login\_User1\_idx` (`User\_userId` ASC),

CONSTRAINT `fk\_Login\_User1`

FOREIGN KEY (`User\_userId`)

REFERENCES `Polygon`.`User` (`userId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`BuildingInfo` (

`buildingInfoId` INT NOT NULL AUTO\_INCREMENT,

`buildYear` VARCHAR(100) NULL,

`area` VARCHAR(100) NULL,

`use` VARCHAR(150) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`buildingInfoId`),

INDEX `fk\_BuildingInfo\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_BuildingInfo\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`BuildingExamination` (

`buildingExaminationId` INT NOT NULL AUTO\_INCREMENT,

`reviewing` VARCHAR(100) NULL,

`description` VARCHAR(1000) NULL,

`comments` INT NULL,

`picture` INT NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`buildingExaminationId`),

INDEX `fk\_BuildingExamination\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_BuildingExamination\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Humidity` (

`humidityId` INT NOT NULL AUTO\_INCREMENT,

`scanning` INT NULL,

`humidityScan` VARCHAR(150) NULL,

`point` VARCHAR(150) NULL,

`description` VARCHAR(450) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`humidityId`),

INDEX `fk\_Humidity\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_Humidity\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`Conclusion` (

`conclusionId` INT NOT NULL AUTO\_INCREMENT,

`room` VARCHAR(150) NULL,

`recommendations` VARCHAR(350) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`conclusionId`),

INDEX `fk\_Conclusion\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_Conclusion\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `Polygon`.`RapportInfo` (

`rapportInfoId` INT NOT NULL AUTO\_INCREMENT,

`date` DATE NULL,

`author` VARCHAR(150) NULL,

`cooperation` VARCHAR(150) NULL,

`rapportNr` VARCHAR(45) NULL,

`Building\_buildingId` INT NOT NULL,

PRIMARY KEY (`rapportInfoId`),

INDEX `fk\_RapportInfo\_Building1\_idx` (`Building\_buildingId` ASC),

CONSTRAINT `fk\_RapportInfo\_Building1`

FOREIGN KEY (`Building\_buildingId`)

REFERENCES `Polygon`.`Building` (`buildingId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)