МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №2

із дисципліни

Організація баз даних та знань

Виконав:

Ст. групи КН-207

Древницький Ю. А.

Прийняв:

Мельникова Н.І.

Львів – 2019 р.

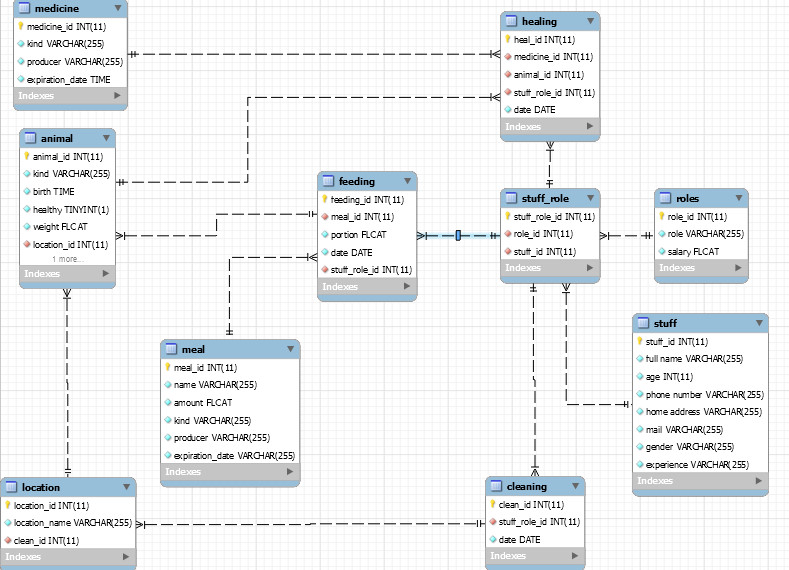
Тема:

Створення таблиць бази даних засобами SQL

Мета:

Побудувати даталогічну модель бази даних; визначити типи, розмірності та обмеження полів; визначити обмеження таблиць; розробити SQL запити для створення спроектованих таблиць.

Скріншот моделі:



Код SQL:

CREATE SCHEMA `zoo` DEFAULT CHARACTER SET utf8 COLLATE utf8\_bin ;

CREATE TABLE `zoo`.`medicine` (

`medicine\_id` INT NOT NULL AUTO\_INCREMENT,

`kind` VARCHAR(255) NOT NULL,

`producer` VARCHAR(255) NOT NULL,

`expiration\_date` TIME NOT NULL,

PRIMARY KEY (`medicine\_id`)

);

CREATE TABLE `zoo`.`meal` (

`meal\_id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(255) NOT NULL,

`amount` FLOAT NOT NULL,

`kind` VARCHAR(255) NOT NULL,

`producer` VARCHAR(255) NOT NULL,

`expiration\_date` VARCHAR(255) NOT NULL,

PRIMARY KEY (`meal\_id`)

);

CREATE TABLE `zoo`.`roles` (

`role\_id` INT NOT NULL AUTO\_INCREMENT,

`role` VARCHAR(255) NOT NULL,

`salary` FLOAT NOT NULL,

PRIMARY KEY (`role\_id`)

);

CREATE TABLE `zoo`.`stuff` (

`stuff\_id` INT NOT NULL AUTO\_INCREMENT,

`full name` VARCHAR(255) NOT NULL,

`age` INT NOT NULL,

`phone number` VARCHAR(255) NOT NULL,

`home address` VARCHAR(255) NOT NULL,

`mail` VARCHAR(255) NOT NULL,

`gender` VARCHAR(255) NOT NULL,

`experience` VARCHAR(255) NOT NULL,

PRIMARY KEY (`stuff\_id`)

);

CREATE TABLE `zoo`.`stuff\_role` (

`stuff\_role\_id` INT NOT NULL AUTO\_INCREMENT,

`role\_id` INT NOT NULL,

`stuff\_id` INT NOT NULL,

PRIMARY KEY (`stuff\_role\_id`)

,

constraint role\_fk foreign key (role\_id)

references zoo.roles (role\_id) on delete cascade on update cascade

,

constraint stuff\_fk foreign key (stuff\_id)

references zoo.stuff (stuff\_id) on delete cascade on update cascade

);

CREATE TABLE `zoo`.`cleaning` (

`clean\_id` INT NOT NULL AUTO\_INCREMENT,

`stuff\_role\_id` INT NOT NULL,

`date` DATE NOT NULL,

PRIMARY KEY (`clean\_id`)

,

constraint stuff\_role\_fk\_3 foreign key (stuff\_role\_id)

references zoo.stuff\_role (stuff\_role\_id) on delete cascade on update cascade

);

CREATE TABLE `zoo`.`location` (

`location\_id` INT NOT NULL AUTO\_INCREMENT,

`location\_name` VARCHAR(255) NOT NULL,

`clean\_id` INT NOT NULL,

PRIMARY KEY (`location\_id`)

,

constraint clean\_fk\_1 foreign key (clean\_id)

references zoo.cleaning (clean\_id) on delete cascade on update cascade

);

CREATE TABLE `zoo`.`feeding` (

`feeding\_id` INT NOT NULL AUTO\_INCREMENT,

`meal\_id` INT NOT NULL,

`portion` FLOAT NOT NULL,

`date` DATE NOT NULL,

`stuff\_role\_id` INT NOT NULL UNIQUE,

PRIMARY KEY (`feeding\_id`)

,

constraint meal\_fk foreign key (meal\_id)

references zoo.meal (meal\_id) on delete no action on update cascade

,

constraint stuff\_role\_fk\_2 foreign key (stuff\_role\_id)

references zoo.stuff\_role (stuff\_role\_id) on delete cascade on update cascade

);

CREATE TABLE `zoo`.`animal` (

`animal\_id` INT NOT NULL AUTO\_INCREMENT,

`kind` VARCHAR(255) NOT NULL,

`birth` TIME NOT NULL,

`healthy` BOOLEAN NOT NULL,

`weight` FLOAT NOT NULL,

`location\_id` INT NOT NULL,

`feeding\_id` INT NOT NULL,

PRIMARY KEY (`animal\_id`)

,

constraint location\_fk\_1 foreign key (location\_id)

references zoo.location (location\_id) on delete cascade on update cascade

,

constraint feeding\_fk\_1 foreign key (feeding\_id)

references zoo.feeding (feeding\_id) on delete cascade on update cascade

);

CREATE TABLE `zoo`.`healing` (

`heal\_id` INT NOT NULL AUTO\_INCREMENT,

`medicine\_id` INT NOT NULL,

`animal\_id` INT NOT NULL UNIQUE,

`stuff\_role\_id` INT NOT NULL,

`date` DATE NOT NULL,

PRIMARY KEY (`heal\_id`)

,

constraint medicine\_fk foreign key (medicine\_id)

references zoo.medicine (medicine\_id) on delete no action on update cascade

,

constraint animal\_fk\_2 foreign key (animal\_id)

references zoo.animal (animal\_id) on delete cascade on update cascade

,

constraint stuff\_role\_fk\_1 foreign key (stuff\_role\_id)

references zoo.stuff\_role (stuff\_role\_id) on delete cascade on update cascade

);

Висновок:

Я спроектував базу даних для управління зоопарком.