

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.

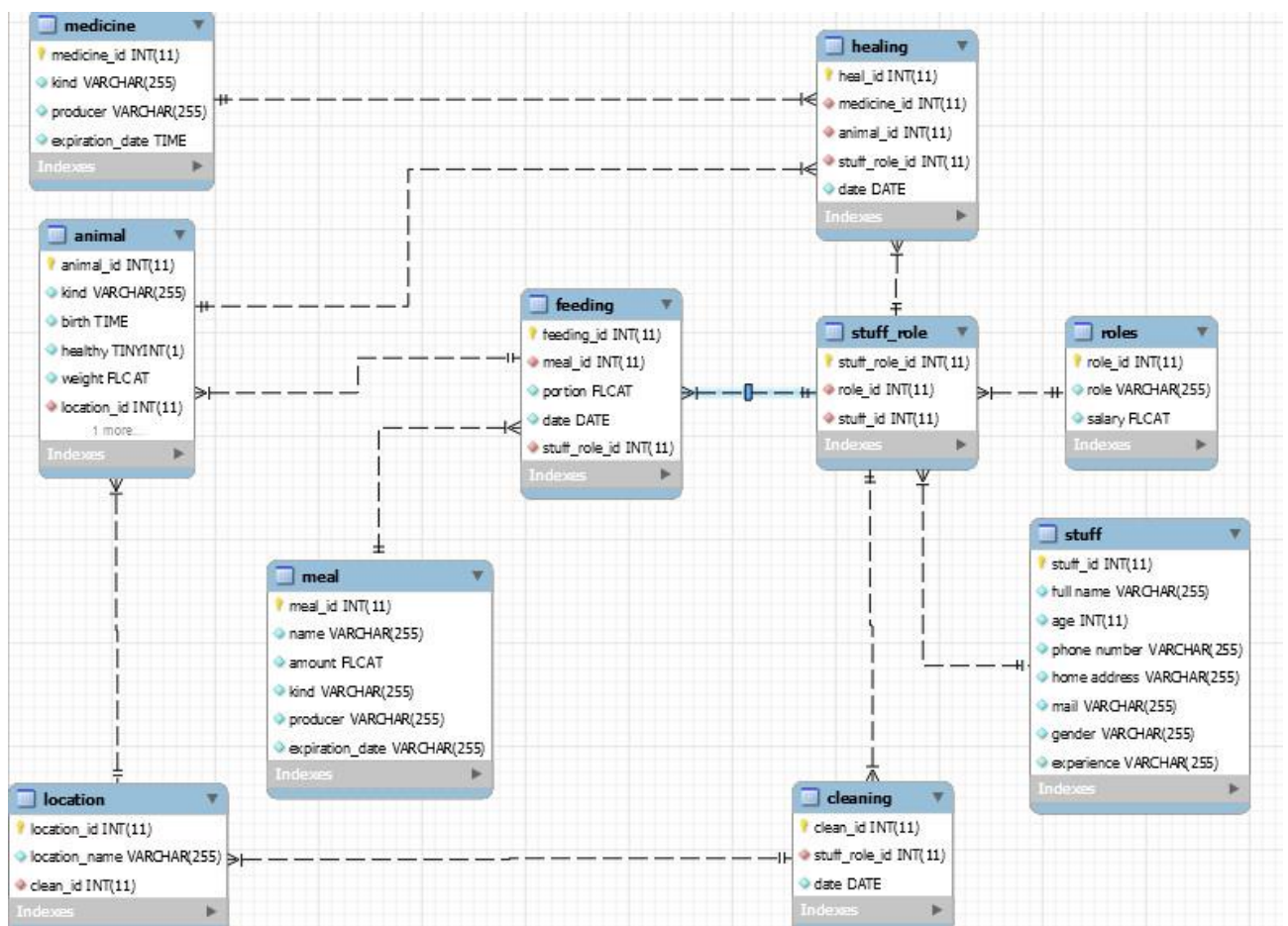
Тема:

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_name` VARCHAR(255) NOT NULL,  
    PRIMARY KEY (`role_id`)  
);
```

```
`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.