



Project Report
On
Database Design

Zoo Management System

Section A
INT205 Database Management System

Professor KITTIPONG WARASUP
And
Professor SUNISA SATHAPORNVAJANA
November 30, 2024

Table of Contents

Title	Page
Table of Contents	1
Member List	2
Member's Responsibility	2
Business Requirement	3
Logical Database Design	5
SQL Statements	6
Data Dictionary	12
DDL Script for creating the database	16
Export Data in the format of Insert Statement	28

Member List

66130500004 Kongphob Kongsan

66130500012 Charana Sukrot

66130500015 Jirapat Sompronratphun

66130500024 Chicha Aksawasewe

66130500045 Theerakan Choochat

Member's Responsibility

66130500004 (20%)

- ดูแล ตรวจสอบ Business Requirement ให้สอดคล้องตลอดงาน

66130500012 (20%)

- ออกแบบและจัดการในส่วนของ Logical Database

66130500015 (20%)

- ดูแลการ Mockup Data และการออกแบบสไลด์นำเสนอ และตรวจสอบเอกสาร

66130500024 (20%)

- ดูแล Query Statement และการออกแบบสไลด์นำเสนอ

66130500045 (20%)

- ตรวจสอบ ตรวจทาน และดูแลองค์ประกอบต่างๆ ของ เอกสาร Report

หมายเหตุ : ส่วนของการออกแบบ Business Requirement และการเลือกข้อมูล ตัดสินใจต่างๆ เป็นการตัดสินใจร่วมกันผ่านการประชุม (จึงไม่ได้มีการเขียนลงไปในงานเฉพาะตัว เพราะทุกคนทำร่วมกัน)

Business Requirement

Objective

- โดยมีจุดประสงค์หลักเพื่อการดูแลสัตว์ให้มีประสิทธิภาพสูงสุด

Scope and Overview

ระบบจัดการสวนสัตว์ของเราส่วนนี้จะ เน้นไปที่ระบบหลังบ้านของสวนสัตว์ โดยแบ่งออกเป็น

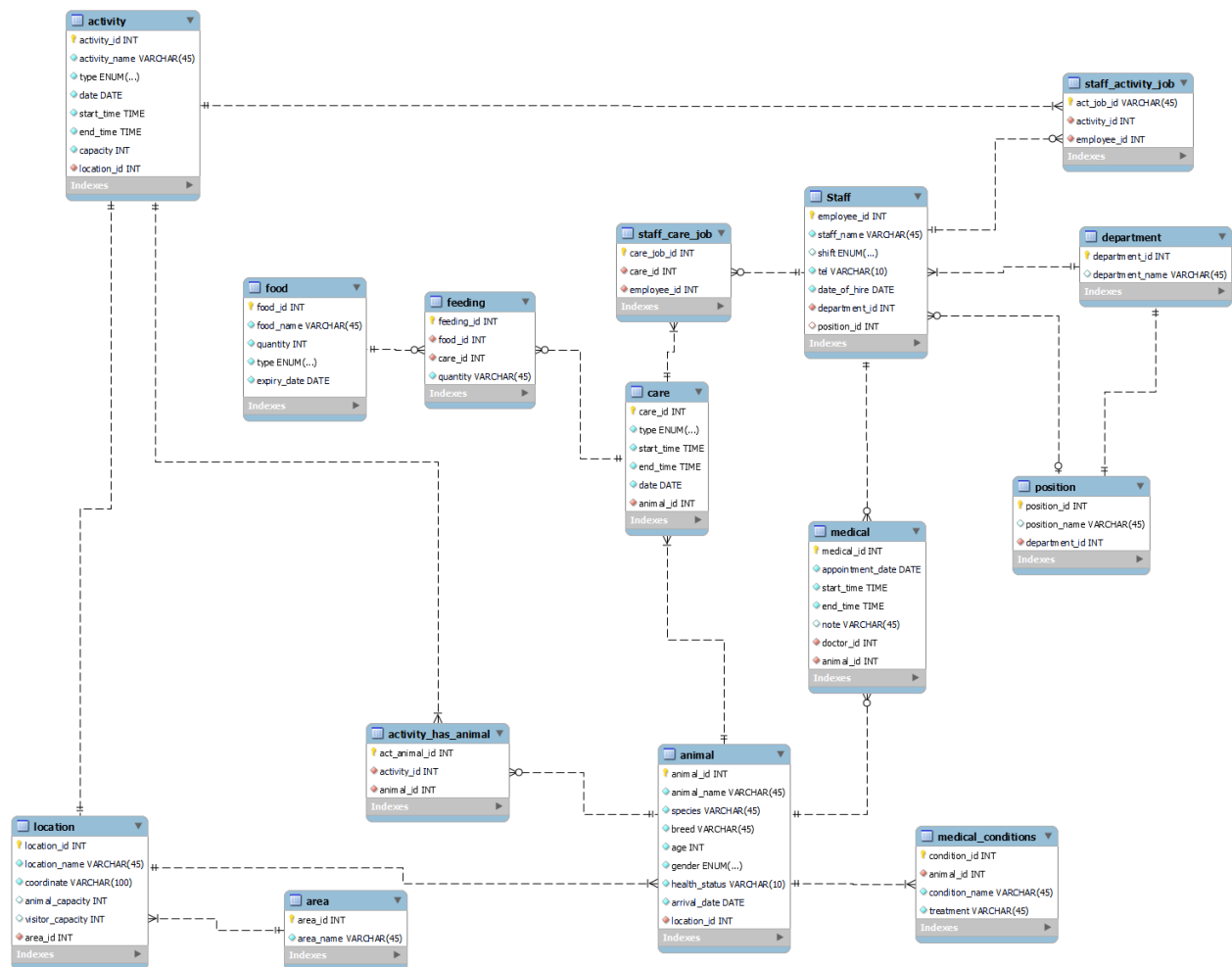
- ส่วนการดูแลสัตว์
- ส่วนของกิจกรรมของสัตว์
- ส่วนของพนักงาน
- ส่วนจัดการสถานที่

Requirements

- **พนักงาน** เป็นบุคลากรทำหน้าที่ดูแลและจัดการเรื่องต่างๆ ภายในสวนสัตว์ โดยพนักงานเองมีข้อกำหนดดังนี้
 - พนักงานทุกคนจำเป็นต้องอยู่ประจำในแผนกใดแผนกหนึ่งเท่านั้น
 - พนักงานทุกคนจะมีตำแหน่งหรือไม่ก็ได้ แต่ถ้ามีจะมีได้แค่ตำแหน่งเดียวเท่านั้น
 - พนักงานคนนึงจะดูแลสัตว์ได้หรือไม่ดูแลก็ได้
 - ในทุกครั้งที่มีการรักษาเกิดขึ้นต้องมีพนักงานที่เป็นแพทย์ไปดูแลสัตว์ 1 คน
- **สัตว์** ในสวนสัตว์จะต้องมีการเก็บข้อมูลหลายๆ อย่างของพวกเขโดยละเอียดเพื่อให้เราสามารถดูแลพวกเขาได้อย่างมีประสิทธิภาพ
 - สัตว์ทุกตัวจะต้องมีที่อยู่อย่างน้อย 1 ที่อยู่
 - สัตว์ทุกตัวจะต้องมีข้อมูลการดูแลในตารางอย่างน้อย 1 ครั้ง
 - สัตว์จะสามารถได้รับการรักษาหรือพบแพทย์ได้กี่ครั้งก็ได้
- **การดูแลสัตว์** ในแต่ละวันจำเป็นต้องมีการที่ พนักงาน เข้าไปดูสัตว์ ทั้งการทำความสะอาด การให้อาหาร และอื่นๆ ดังนั้นเราต้องเก็บข้อมูลส่วนนั้นไว้ด้วย
 - ในการให้อาหารสัตว์ในวันนึงอาจจะให้หรือไม่ให้อาหารสัตว์ก็ได้ตามแต่คู่มือการดูแล
 - ในการดูแลสัตว์สามารถใช้พนักงานมากกว่า 1 คนได้

- สามารถให้อาหารสัตว์มากกว่า 1 ชนิดได้หากมีการระบุในคู่มือ
- **การรักษาสัตว์** เป็นเรื่องปกติที่สัตว์ของเราอาจจะป่วยได้ และถ้าหากเกิดขึ้นเราจะต้องมีการจัดคิวให้สัตวแพทย์เข้าไปดูแล
 - ในการดูแลสัตว์จะต้องมีการระบุว่ามีแพทย์คนใดไปดูแลสัตว์ตัวใด
 - และจะต้องมีการเก็บอาการของสัตว์ที่ป่วยไว้ด้วยว่าตัวไหนป่วยเป็นอะไรในตอนนั้นๆ
- **กิจกรรมพิเศษ** ในสวนสัตว์มักจะจัดกิจกรรมพิเศษต่างๆ ขึ้นมาเสมอ และกิจกรรมเหล่านั้นต้องมีการดูแลและควบคุม
 - ในการจัดกิจกรรมพิเศษขึ้นมาจะต้องมี Staff อย่างน้อย 1 คนมาดูแลกิจกรรมดังกล่าว
 - ในการแสดงสามารถนำสัตว์มาแสดงก็ตัวก็ได้ แต่ต้องมีอย่างน้อย 1 ตัว
- **สถานที่** ในสวนสัตว์ประกอบไปด้วยสถานที่หลายที่มากมาย รวมถึงกรงสัตว์ด้วย ดังนั้นเราจำเป็นต้องมีการจัดการ
 - ทุกสถานที่ที่จะอยู่ใน Area ไตลัก Area
 - สถานที่แต่ละที่อาจจะมีการจำกัดจำนวนคนหรือไม่ก็ได้
 - สถานที่ที่เป็นกรงสัตว์จะมีจำนวนสัตว์สูงสุดไว้

Logical Database Design



SQL Statements

SQL Statement 1

```
SELECT s.employee_id , s.staff_name , d.department_name
```

```
FROM staff s
```

```
JOIN department a
```

```
ON s.department_id = d.department_id
```

```
ORDER BY s.staff_name;
```

Description

ให้แสดง เลขพนักงาน รายชื่อ และ สาขา ของพนักงาน ผลลัพธ์เรียงตาม ชื่อพนักงาน

Query Result

	employee_id	staff_name	department_name
▶	2	Malee Bunmee	Animal Care Department
	1	Somchai Pongdee	Veterinary Department
	4	Supawadee Jansri	Animal Care Department
	3	Wichai Intorn	Special Events Department

SQL Statement 2

```
SELECT a.animal_name , COUNT(aha.activity_id) AS activity_count  
  
FROM activity_has_animal aha  
  
JOIN animal a ON aha.animal_id = a.animal_id  
  
GROUP BY a.animal_name  
  
ORDER BY animal_name;
```

Description

แสดงสัตว์ที่เข้าร่วมกิจกรรมต่างๆ พร้อมแสดงจำนวนกิจกรรมที่สัตว์

แต่ละตัวเข้าร่วม ผลลัพธ์เรียงตามชื่อสัตว์

Query Result

animal_name	activity_count
Chompoo	1
Chor	1
Ploy	1
Thongdee	1

SQL Statement 3

```

SELECT a.animal_name,m.appointment_date,mc.condition_name,s.staff_name AS doctor

FROM animal a

JOIN medical_conditions mc ON a.animal_id = mc.animal_id

JOIN medical m ON a.animal_id = m.animal_id

JOIN staff s ON m.doctor_id = s.employee_id

ORDER BY m.appointment_date ASC;

```

Description

ให้ดึงข้อมูลเกี่ยวกับสัตว์ที่ต้องการการดูแลทางการแพทย์ โดยให้แสดงชื่อสัตว์ วันที่นัดหมาย เหตุผลในการดูแล และพนักงานที่เป็นผู้ดูแล ผลลัพธ์เรียงตามวันนัดหมายจากน้อยไปมาก

Query Result

animal_name	appointment_date	condition_name	doctor
Thongdee	2023-11-20	Parvovirus	Somchai Pongdee
Chompoo	2023-11-21	Arthritis	Malee Bunmee
Chor	2023-11-22	Feline Leukemia	Wichai Intorn
Ploy	2023-11-23	Respiratory Infection	Supawadee Jansri

SQL Statement 4

```

SELECT ar.area_name,l.location_name,l.coordinate,COUNT(DISTINCT a.animal_id) AS

total_animals,SUM(act.capacity) AS total_capacity

```

```
FROM Area ar JOIN Location l ON ar.area_id = l.area_id
```

```
JOIN Animal a ON l.location_id = a.location_id
```

```
LEFT JOIN Activity act ON l.location_id = act.location_id
```

```
GROUP BY ar.area_name, l.location_name, l.coordinate
```

```
ORDER BY total_animals DESC, total_capacity DESC;
```

Description

แสดงพื้นที่ทั้งหมดที่สัตว์ถูกดูแล พร้อมจำนวนนักท่องเที่ยวที่พื้นที่นั้นรองรับได้สูงสุด

Query Result

	area_name	location_name	coordinate	total_animals	total_capacity
▶	Savannah Area	Tiger Zone	13.7367,100.5232	1	100
	Mountain Area	Monkey Zone	13.7355,100.5220	1	80
	Wetlands Area	Elephant Zone	13.7370,100.5245	1	50
	Tropical Area	Parrot Zone	13.7380,100.5250	1	40

SQL Statement 5

```
SELECT
```

```
s.staff_name,s.position_id,d.department_name,act.activity_name,act.date,act.start_time,act.end_time,
```

```
GROUP_CONCAT(a.animal_name SEPARATOR ', ') AS involved_animals
```

```
FROM Staff s JOIN Staff_Activity_job saj ON s.employee_id = saj.employee_id

JOIN Activity act ON saj.activity_id = act.activity_id

LEFT JOIN Activity_Has_Animal aha ON act.activity_id = aha.activity_id

LEFT JOIN Animal a ON aha.animal_id = a.animal_id

JOIN Department d ON s.department_id = d.department_id

GROUP BY s.staff_name, s.position_id, d.department_name, act.activity_name, act.date, act.start_time,
act.end_time

ORDER BY act.date, act.start_time;
```

Description

แสดงตารางงานของพนักงานในกิจกรรมต่าง ๆ พร้อมรายละเอียดสัตว์ที่เกี่ยวข้อง รวมถึงกิจกรรมที่อาจจะไม่มีสัตว์ด้วย (เพื่อในกรณีที่เพิ่มสัตว์เข้าไป)

Query Result

	area_name	location_name	coordinate	total_animals	total_capacity
►	Savannah Area	Tiger Zone	13.7367,100.5232	1	100
	Mountain Area	Monkey Zone	13.7355,100.5220	1	80
	Wetlands Area	Elephant Zone	13.7370,100.5245	1	50
	Tropical Area	Parrot Zone	13.7380,100.5250	1	40

Data dictionary

Table: activity

No	Column Name	Description	Data Type	Constraint	References
1	activity_id	รหัสประจำกิจกรรม	INT	Primary Key	None
2	activity_name	ชื่อกิจกรรม	VARCHAR(45)	NOT NULL	None
3	type	ประเภทของกิจกรรม	VARCHAR(45)	NOT NULL	None
4	date	วันที่จัดกิจกรรม	DATE	NOT NULL	None
5	start_time	เวลาเริ่มกิจกรรม	TIME	NOT NULL	None
6	end_time	เวลาสิ้นสุดกิจกรรม	TIME	NOT NULL	None
7	capacity	จำนวนที่รองรับได้	INT	NOT NULL	None
8	location_id	รหัสสถานที่จัดกิจกรรม	INT	NOT NULL, Foreign Key	References location

Table: staff

No	Column Name	Description	Data Type	Constraint	References
1	employee_id	รหัสประจำตัวพนักงาน	INT	Primary Key	None
2	staff_name	ชื่อพนักงาน	VARCHAR(45)	NOT NULL	None
3	shift	กะทำงาน	ENUM	NULLABLE	None
4	tel	เบอร์โทรศัพท์	VARCHAR(10)	NOT NULL	None
5	date_of_hire	วันที่เริ่มทำงาน	DATE	NOT NULL	None
6	department_id	รหัสแผนก	INT	NOT NULL, Foreign Key	References department
7	position_id	รหัสตำแหน่ง	INT	NULLABLE , Foreign Key	References position

Table: staff_activity_job

No	Column Name	Description	Data Type	Constraint	References
1	act_job_id	รหัสประจำการปฏิบัติงาน	VARCHAR(45)	Primary Key	None
2	activity_id	รหัสกิจกรรม	INT	NOT NULL, Foreign Key	References activity
3	employee_id	รหัสพนักงาน	INT	NOT NULL, Foreign Key	References staff

Table: department

No	Column Name	Description	Data Type	Constraint	References
1	department_id	รหัสแผนก	INT	Primary Key	None
2	department_name	ชื่อแผนก	VARCHAR(45)	NULLABLE	None

Table: care

No	Column Name	Description	Data Type	Constraint	References
1	care_id	รหัสการดูแล	INT	Primary Key	None
2	type	ประเภทของการดูแล	ENUM	NOT NULL	None
3	start_time	เวลาเริ่มต้นการดูแล	TIME	NOT NULL	None
4	end_time	เวลาสิ้นสุดการดูแล	TIME	NOT NULL	None
5	date	วันที่ทำการดูแล	DATE	NOT NULL	None
6	animal_id	รหัสสัตว์	INT	NOT NULL, Foreign Key	References animal

Table: staff_care_job

No	Column Name	Description	Data Type	Constraint	References
1	care_job_id	รหัสงานดูแล	INT	Primary Key	None
2	care_id	รหัสการดูแล	INT	NOT NULL, Foreign Key	References care
3	employee_id	รหัสพนักงาน	INT	NOT NULL, Foreign Key	References

Table: food

No	Column Name	Description	Data Type	Constraint	References
1	food_id	รหัสประจำตัวอาหาร	INT	Primary Key	None
2	food_name	ชื่ออาหาร	VARCHAR(45)	NOT NULL	None
3	quantity	จำนวนอาหาร	INT	NOT NULL	None
4	type	ประเภทของอาหาร	ENUM	NOT NULL	None
5	expiry_date	วันหมดอายุ	DATE	NOT NULL	None

Table: feeding

No	Column Name	Description	Data Type	Constraint	References
1	feeding_id	รหัสประจำการให้อาหาร	INT	Primary Key	None
2	food_id	รหัสอาหาร	INT	NOT NULL, Foreign Key	References food
3	care_id	รหัสการดูแล	INT	NOT NULL, Foreign Key	References care
4	quantity	ปริมาณอาหารที่ให้	VARCHAR(45)	NOT NULL	None

Table: medical

No	Column Name	Description	Data Type	Constraint	References
1	medical_id	รหัสประวัติการรักษา	INT	Primary Key	None
2	appointment_date	วันนัดหมาย	DATE	NOT NULL	None
3	start_time	เวลาเริ่มการรักษา	TIME	NOT NULL	None
4	end_time	เวลาสิ้นสุดการรักษา	TIME	NOT NULL	None
5	note	หมายเหตุ	VARCHAR(45)	NULLABLE	None
6	doctor_id	รหัสสัตวแพทย์	INT	NOT NULL, Foreign Key	References staff
7	animal_id	รหัสสัตว์	INT	NOT NULL, Foreign Key	References animal

Table: position

No	Column Name	Description	Data Type	Constraint	References
1	position_id	รหัสตำแหน่ง	INT	Primary Key	None
2	position_name	ชื่อตำแหน่ง	VARCHAR(45)	NULLABLE	None
3	department_id	รหัสแผนก	INT	NOT NULL, Foreign Key	References department

Table: animal

No	Column Name	Description	Data Type	Constraint	References
1	animal_id	รหัสประจำสัตว์	INT	Primary Key	None
2	animal_name	ชื่อสัตว์	VARCHAR(45)	NOT NULL	None
3	species	ชนิดสัตว์	VARCHAR(45)	NOT NULL	None
4	breed	พันธุ์สัตว์	VARCHAR(45)	NOT NULL	None
5	age	อายุ	INT	NOT NULL	None
6	gender	เพศ	ENUM	NOT NULL	None
7	health_status	สถานะสุขภาพ	VARCHAR(10)	NOT NULL	None
8	arrival_date	วันที่เข้ามา	DATE	NOT NULL	None
9	location_id	รหัสสถานที่	INT	NOT NULL, Foreign Key	References location

Table: activity_has_animal

No	Column Name	Description	Data Type	Constraint	References
1	act_animal_id	รหัสประจำกิจกรรมที่เกี่ยวข้องกับสัตว์	INT	Primary Key	None
2	activity_id	รหัสกิจกรรม	INT	NOT NULL, Foreign Key	References activity
3	animal_id	รหัสสัตว์	INT	NOT NULL, Foreign Key	References animal

Table: medical_conditions

No	Column Name	Description	Data Type	Constraint	References
1	condition_id	รหัสสภาวะทางการแพทย์	INT	Primary Key	None
2	animal_id	รหัสสัตว์	INT	NOT NULL, Foreign Key	References animal
3	condition_name	ชื่อสภาวะทางการแพทย์	VARCHAR(45)	NOT NULL	None
4	treatment	การรักษา	VARCHAR(45)	NOT NULL	None

Table: area

No	Column Name	Description	Data Type	Constraint	References
1	area_id	รหัสพื้นที่	INT	Primary Key	None
2	area_name	ชื่อพื้นที่	VARCHAR(45)	NOT NULL	None

Table: location

No	Column Name	Description	Data Type	Constraint	References
1	location_id	รหัสประจำสถานที่	INT	Primary Key	None
2	location_name	ชื่อสถานที่	VARCHAR(45)	NOT NULL	None
3	coordinate	พิกัด	VARCHAR(100)	NOT NULL	None
4	animal_capacity	ความจุของสัตว์	INT	NULLABLE	None
5	visitor_capacity	ความจุของผู้เข้าชม	INT	NULLABLE	None
6	area_id	รหัสพื้นที่	INT	NOT NULL, Foreign Key	References area

DDL Script for creating the database

Schema logistics

```
CREATE SCHEMA IF NOT EXISTS `zoodb` DEFAULT CHARACTER SET utf8 ;  
USE `zoodb` ;
```

Table: area

```
CREATE TABLE IF NOT EXISTS `zoodb`.`area` (  
  `area_id` INT NOT NULL,  
  `area_name` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`area_id`))  
ENGINE = InnoDB;
```

Table: location

```
CREATE TABLE IF NOT EXISTS `zoodb`.`location` (  
  `location_id` INT NOT NULL,  
  `location_name` VARCHAR(45) NOT NULL,  
  `coordinate` VARCHAR(100) NOT NULL,  
  `animal_capacity` INT NULL,  
  `visitor_capacity` INT NULL,  
  `area_id` INT NOT NULL,  
  PRIMARY KEY (`location_id`),  
  INDEX `fk_location_area1_idx` (`area_id` ASC) VISIBLE,  
  CONSTRAINT `fk_location_area1`  
    FOREIGN KEY (`area_id`)  
    REFERENCES `zoodb`.`area` (`area_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Table: animal

```
CREATE TABLE IF NOT EXISTS `zoodb`.`animal` (  
  `animal_id` INT NOT NULL,  
  `animal_name` VARCHAR(45) NOT NULL,  
  `species` VARCHAR(45) NOT NULL,  
  `breed` VARCHAR(45) NOT NULL,  
  `age` INT NOT NULL,  
  `gender` ENUM("MALE", "FEMALE") NOT NULL,  
  `health_status` VARCHAR(10) NOT NULL,  
  `arrival_date` DATE NOT NULL,  
  `location_id` INT NOT NULL,  
  PRIMARY KEY (`animal_id`),  
  INDEX `fk_animal_location_idx` (`location_id` ASC) VISIBLE,  
  CONSTRAINT `fk_animal_location`  
    FOREIGN KEY (`location_id`)  
      REFERENCES `zoodb`.`location` (`location_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Table: department

```
CREATE TABLE IF NOT EXISTS `zoodb`.`department` (  
  `department_id` INT NOT NULL,  
  `department_name` VARCHAR(45) NULL,  
  PRIMARY KEY (`department_id`))  
ENGINE = InnoDB;
```

Table: position

```
CREATE TABLE IF NOT EXISTS `zoodb`.`position` (  
  `position_id` INT NOT NULL,  
  `position_name` VARCHAR(45) NULL,  
  `department_id` INT NOT NULL,  
  PRIMARY KEY (`position_id`),  
  INDEX `fk_position_department1_idx` (`department_id` ASC) VISIBLE,  
  CONSTRAINT `fk_position_department1`  
    FOREIGN KEY (`department_id`)  
      REFERENCES `zoodb`.`department` (`department_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Table: staff

```
CREATE TABLE IF NOT EXISTS `zoodb`.`Staff` (  
  `employee_id` INT NOT NULL,  
  `staff_name` VARCHAR(45) NOT NULL,  
  `shift` ENUM('Morning', 'Afternoon', 'Evening', 'Night') NULL,  
  `tel` VARCHAR(10) NOT NULL,  
  `date_of_hire` DATE NOT NULL,  
  `department_id` INT NOT NULL,  
  `position_id` INT NULL,  
  PRIMARY KEY (`employee_id`),  
  INDEX `fk_Staff_department1_idx` (`department_id` ASC) VISIBLE,  
  INDEX `fk_Staff_position1_idx` (`position_id` ASC) VISIBLE,  
  CONSTRAINT `fk_Staff_department1`  
    FOREIGN KEY (`department_id`)  
    REFERENCES `zoodb`.`department` (`department_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Staff_position1`  
    FOREIGN KEY (`position_id`)  
    REFERENCES `zoodb`.`position` (`position_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Table: activity

```

CREATE TABLE IF NOT EXISTS `zoodb`.`activity` (
  `activity_id` INT NOT NULL,
  `activity_name` VARCHAR(45) NOT NULL,
  `type` ENUM('AnimalInteraction', 'EducationAndLearning', 'PhotoOpportunities', 'EntertainmentAndShows',
'ExplorationAndAdventure', 'DiningAndRelaxation', 'CreativeActivities', 'GamesAndChallenges', 'KidsZone',
'SpecialEvents', 'ShoppingAndSouvenirs') NOT NULL,
  `date` DATE NOT NULL,
  `start_time` TIME NOT NULL,
  `end_time` TIME NOT NULL,
  `capacity` INT NOT NULL,
  `location_id` INT NOT NULL,
  PRIMARY KEY (`activity_id`),
  INDEX `fk_activity_location1_idx` (`location_id` ASC) VISIBLE,
  CONSTRAINT `fk_activity_location1`
    FOREIGN KEY (`location_id`)
      REFERENCES `zoodb`.`location` (`location_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: care

```

CREATE TABLE IF NOT EXISTS `zoodb`.`care` (
  `care_id` INT NOT NULL,
  `type` ENUM("CLEAN", "FEED", "CHECK") NOT NULL,
  `start_time` TIME NOT NULL,
  `end_time` TIME NOT NULL,
  `date` DATE NOT NULL,
  `animal_id` INT NOT NULL,
  PRIMARY KEY (`care_id`),
  INDEX `fk_care_animal1_idx` (`animal_id` ASC) VISIBLE,
  CONSTRAINT `fk_care_animal1`
    FOREIGN KEY (`animal_id`)
      REFERENCES `zoodb`.`animal` (`animal_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: food

```

CREATE TABLE IF NOT EXISTS `zoodb`.`food` (
  `food_id` INT NOT NULL,
  `food_name` VARCHAR(45) NOT NULL,
  `quantity` INT NOT NULL,
  `type` ENUM("Meat", "Plants") NOT NULL,
  `expiry_date` DATE NOT NULL,
  PRIMARY KEY (`food_id`))
ENGINE = InnoDB;

```

Table: medical

```

CREATE TABLE IF NOT EXISTS `zoodb`.`medical` (
  `medical_id` INT NOT NULL,
  `appointment_date` DATE NOT NULL,
  `start_time` TIME NOT NULL,
  `end_time` TIME NOT NULL,
  `note` VARCHAR(45) NULL,
  `doctor_id` INT NOT NULL,
  `animal_id` INT NOT NULL,
  PRIMARY KEY (`medical_id`),
  INDEX `fk_medical_Staff1_idx` (`doctor_id` ASC) VISIBLE,
  INDEX `fk_medical_animal1_idx` (`animal_id` ASC) VISIBLE,
  CONSTRAINT `fk_medical_Staff1`
    FOREIGN KEY (`doctor_id`)
      REFERENCES `zoodb`.`Staff` (`employee_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_medical_animal1`
    FOREIGN KEY (`animal_id`)
      REFERENCES `zoodb`.`animal` (`animal_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: staff_activity_job

```

CREATE TABLE IF NOT EXISTS `zoodb`.`staff_activity_job` (
  `act_job_id` VARCHAR(45) NOT NULL,
  `activity_id` INT NOT NULL,
  `employee_id` INT NOT NULL,
  PRIMARY KEY (`act_job_id`),
  INDEX `fk_activity_has_Staff_Staff1_idx` (`employee_id` ASC) VISIBLE,
  INDEX `fk_activity_has_Staff_activity1_idx` (`activity_id` ASC) VISIBLE,
  CONSTRAINT `fk_activity_has_Staff_activity1`
    FOREIGN KEY (`activity_id`)
      REFERENCES `zoodb`.`activity` (`activity_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_activity_has_Staff_Staff1`
    FOREIGN KEY (`employee_id`)
      REFERENCES `zoodb`.`Staff` (`employee_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```


Table: activity_has_animal

```

CREATE TABLE IF NOT EXISTS zoodb.`activity_has_animal` (
  `act_animal_id` INT NOT NULL,
  `activity_id` INT NOT NULL,
  `animal_id` INT NOT NULL,
  PRIMARY KEY (`act_animal_id`),
  INDEX `fk_activity_has_animal_animal1_idx` (`animal_id` ASC) VISIBLE,
  INDEX `fk_activity_has_animal_activity1_idx` (`activity_id` ASC) VISIBLE,
  CONSTRAINT `fk_activity_has_animal_activity1`
    FOREIGN KEY (`activity_id`)
      REFERENCES zoodb.`activity` (`activity_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_activity_has_animal_animal1`
    FOREIGN KEY (`animal_id`)
      REFERENCES zoodb.`animal` (`animal_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: staff_care_job

```

CREATE TABLE IF NOT EXISTS `zoodb`.`staff_care_job` (
  `care_job_id` INT NOT NULL,
  `care_id` INT NOT NULL,
  `employee_id` INT NOT NULL,
  PRIMARY KEY (`care_job_id`),
  INDEX `fk_care_has_Staff_Staff1_idx` (`employee_id` ASC) VISIBLE,
  INDEX `fk_care_has_Staff_care1_idx` (`care_id` ASC) VISIBLE,
  CONSTRAINT `fk_care_has_Staff_care1`
    FOREIGN KEY (`care_id`)
      REFERENCES `zoodb`.`care` (`care_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_care_has_Staff_Staff1`
    FOREIGN KEY (`employee_id`)
      REFERENCES `zoodb`.`Staff` (`employee_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: **feeding**

```

CREATE TABLE IF NOT EXISTS `zoodb`.`feeding` (
  `food_id` INT NOT NULL,
  `care_id` INT NOT NULL,
  `feeding_id` INT NOT NULL,
  `quantity` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`feeding_id`),
  INDEX `fk_food_has_care_care1_idx` (`care_id` ASC) VISIBLE,
  INDEX `fk_food_has_care_food1_idx` (`food_id` ASC) VISIBLE,
  CONSTRAINT `fk_food_has_care_food1`
    FOREIGN KEY (`food_id`)
      REFERENCES `zoodb`.`food` (`food_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_food_has_care_care1`
    FOREIGN KEY (`care_id`)
      REFERENCES `zoodb`.`care` (`care_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table: **medical_conditions**

```
CREATE TABLE IF NOT EXISTS zoodb.`medical_conditions` (  
  `condition_id` INT NOT NULL,  
  `animal_id` INT NOT NULL,  
  `condition_name` VARCHAR(45) NOT NULL,  
  `treatment` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`condition_id`),  
  INDEX `fk_medical_conditions_animal1_idx` (`animal_id` ASC) VISIBLE,  
  CONSTRAINT `fk_medical_conditions_animal1`  
    FOREIGN KEY (`animal_id`)  
      REFERENCES zoodb.`animal` (`animal_id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Export Data in the format of INSERT statement

```
USE `zoodb` ;
```

Department

```
INSERT INTO Department (department_id, department_name) VALUES  
(1, 'Veterinary Department'),  
(2, 'Animal Care Department'),  
(3, 'Special Events Department'),  
(4, 'Customer Service Department');
```

Position

```
INSERT INTO Position (position_id, position_name, department_id) VALUES  
(1, 'Manager', 1),  
(2, 'Veterinarian', 2),  
(3, 'Technician', 3),  
(4, 'Nurse', 4 );
```

Staff

```
INSERT INTO Staff (employee_id, staff_name, position_id, department_id, shift, tel, date_of_hire) VALUES  
(1, 'Somchai Pongdee', 1, 1, 'Morning', '0811234567', '2023-01-15'),  
(2, 'Malee Bunmee', 2, 2, 'Afternoon', '0923456789', '2022-11-20'),  
(3, 'Wichai Intorn', 3, 3, 'Night', '0834567890', '2023-05-01'),  
(4, 'Supawadee Jansri', 4, 2, 'Evening', '0895678901', '2023-07-10');
```

Area

```
INSERT INTO Area (area_id, area_name) VALUES
```

```
(1, 'Savannah Area'),  
(2, 'Wetlands Area'),  
(3, 'Mountain Area'),  
(4, 'Tropical Area');
```

Location

```
INSERT INTO Location (location_id, location_name, coordinate, animal_capacity, area_id) VALUES
```

```
(1, 'Tiger Zone', '13.7367,100.5232', 3, 1),  
(2, 'Elephant Zone', '13.7370,100.5245', 4, 2),  
(3, 'Monkey Zone', '13.7355,100.5220', 6, 3),  
(4, 'Parrot Zone', '13.7380,100.5250', 5, 4);
```

Animal

```
INSERT INTO Animal (animal_id, animal_name, species, breed, age, gender, health_status, arrival_date,  
location_id) VALUES
```

```
(1, 'Thongdee', 'Tiger', 'Indochinese Tiger', 8, 'Male', 'Healthy', '2021-05-12', 1),  
(2, 'Chompoo', 'Elephant', 'Thai Elephant', 12, 'Female', 'Healthy', '2020-03-15', 2),  
(3, 'Chor', 'Monkey', 'Long-tailed Macaque', 5, 'Male', 'Healthy', '2022-06-10', 3),  
(4, 'Ploy', 'Parrot', 'Macaw', 3, 'Female', 'Healthy', '2023-02-01', 4);
```

Care

```
INSERT INTO Care (care_id, type, start_time, end_time, date, animal_id) VALUES
```

```
(1, CLEAN, '08:00:00', '08:30:00', '2023-11-23', 1),  
(2, FEED, '10:00:00', '10:30:00', '2023-11-23', 2),  
(3, CHECK, '14:00:00', '14:30:00', '2023-11-23', 3),  
(4, 'CHECK', '15:00:00', '15:30:00', '2023-11-23', 4);
```

Food

```
INSERT INTO Food (food_id, food_name, quantity, type, expiry_date) VALUES
(1, 'Raw Meat', 20, 'Meat','2025-11-11'),
(2, 'Fresh Grass', 50, 'Meat','2025-11-11'),
(3, 'Banana', 30, 'Plants','2025-11-11'),
(4, 'Seeds', 15, 'Plants','2025-11-11');
```

Feeding

```
INSERT INTO Feeding (feeding_id, care_id, food_id, quantity) VALUES
(1, 1, 1, 5),
(2, 2, 2, 20),
(3, 3, 3, 10),
(4, 4, 4, 5);
```

Medical

```
INSERT INTO Medical (medical_id, appointment_date, start_time, end_time, note,doctor_id, animal_id) VALUES
(1, '2023-11-20', '09:00:00', '09:30:00', 'General health check',1, 1),
(2, '2023-11-21', '10:00:00', '10:30:00', 'Annual vaccination',2, 2),
(3, '2023-11-22', '11:00:00', '11:30:00', 'Treating leg injury',3, 3),
(4, '2023-11-23', '13:00:00', '13:30:00', 'Nutritional assessment',4, 4);
```

Activity

```
INSERT INTO Activity (activity_id, activity_name, type, date, start_time,end_time, capacity, location_id) VALUES
(1, 'Tiger Feeding Show', 'AnimalInteraction', '2023-11-25', '14:00:00','14:30:00', 100, 1),
(2, 'Elephant Walk', 'AnimalInteraction', '2023-11-26', '16:00:00','16:45:00', 50, 2),
(3, 'Monkey Show', 'EntertainmentAndShows', '2023-11-27', '11:00:00','11:30:00', 80, 3),
(4, 'Parrot Talk', 'PhotoOpportunities', '2023-11-28', '10:00:00','10:20:00', 40, 4);
```

Staff_Activity_job

```
INSERT INTO Staff_Activity_job (act_job_id,activity_id ,employee_id) VALUES
(1, 1,1),
(2, 2,2),
(3, 3,3),
(4, 4,4);
```

Activity_Has_Animal

```
INSERT INTO Activity_Has_Animal (act_animal_id, activity_id,animal_id) VALUES
(1, 1,1),
(2, 2,2),
(3, 3,3),
(4, 4,4);
```

Department

```
INSERT INTO Department (department_id, department_name)
VALUES
(100, 'Veterinary Department'),
(200, 'Animal Care Department'),
(300, 'Special Events Department'),
```


(400, 'Customer Service Department');

Medical Conditions

INSERT INTO medical_conditions (condition_id , condition_name, treatment, animal_id) VALUES

(001,'Parvovirus', 'Antiviral medication', 1),

(002,'Arthritis', 'Pain relief and supplements', 2),

(003,'Feline Leukemia', 'Supportive care', 3),

(004,'Respiratory Infection', 'Antibiotics and nebulization', 4);