Group - 6 | Jay Patel(UCID: jrp54) | Krupa Lakhani (UCID: kgl) | Het Patel (UCID : hp578)

Final Report OF ZOO MANAGEMENT SYSTEM

CS 631 003 Data Management System Design

TABLE OF CONTENT

1. Introduction

The goal of this project is to develop a user-friendly online system for Turtleback Zoo, streamlining operations across Asset Management, Daily Zoo Activity, and Management Reporting. The focus is on simplicity, adaptability for future features, and ensuring accurate data for efficient decision-making. The project also emphasizes robust security measures and close collaboration with the zoo's team for effective integration and continuous improvement. Here are some key points.

1. Make an easy-to-use online system for Turtleback Zoo

2. Main menu system.

3. To Manage day-to-day operations.

4. Develop the Daily Zoo Activity part.

5. Build the Management and Reporting part.

6. Manage and ensure financial reports.

7. Security to protect sensitive zoo data, keeping it private and intact.

8. Make sure the system can grow and change easily. It should be ready for new features as Turtleback Zoo's needs evolve. Work closely with the zoo team for feedback and tweaks, ensuring the system smoothly becomes part of their daily routine.

2. Phase-1 Goal

- Capture the functional requirements by considering the provided specifications and assumptions.
- Define and solidify entities, their attributes, and the connections between them.
- Recognize key constraints in the system.
- Construct an extended Entity-Relationship (E-R) diagram encompassing all attributes, entities, and relationships.
- Identify significant challenges in the development of the conceptual design.

3. Entity Types

Employee (ID, StartDate, JobType, Name, Address)

Hourly Rate (HR_ID, Rate)

Species (S_ID, Food Cost, Name)

Animal (Animal_ID, Status, Birth Year)

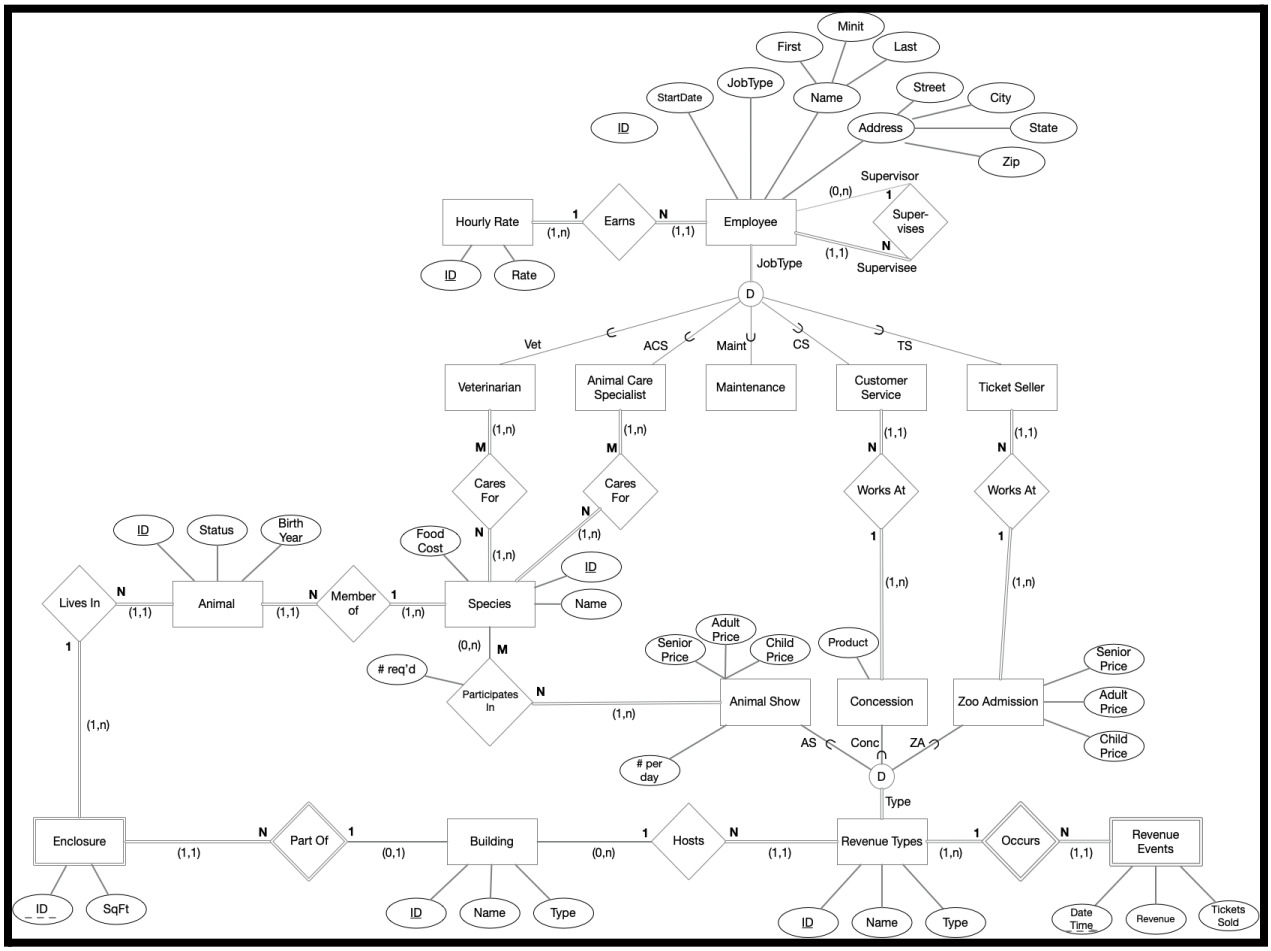
Enclosure- weak Entity (Enclosure_ID, SqFt)

Building (Building_ID, Name, Type)

Revenue Type (RevenueType_ID, Name, Type)

Revenue Events- weak Entity (Date_Time, Revenue, Tickets Sold)

4. EER Diagram



Pdf attached at the last.

5. Challenges on phase one.

- Deciding on the correct arrangement of attributes in the diagram involves carefully reviewing the instructions and making reasonable assumptions. It is essential to follow the instructions closely and progress step by step until the diagram is fully developed.
- Regarding the notation for the EER diagram, it is recommended to review the slides to gain a better understanding of the required notation.
- When it comes to selecting the platform for constructing the EER diagram, a brief demonstration was conducted on various platforms, ultimately leading to the choice of LucidChart for building the diagram.

6. Phase-2 Goal

- Demonstrate the process of translating from the EER diagram to the Relational schema, adhering to the EER to Relational algorithm.

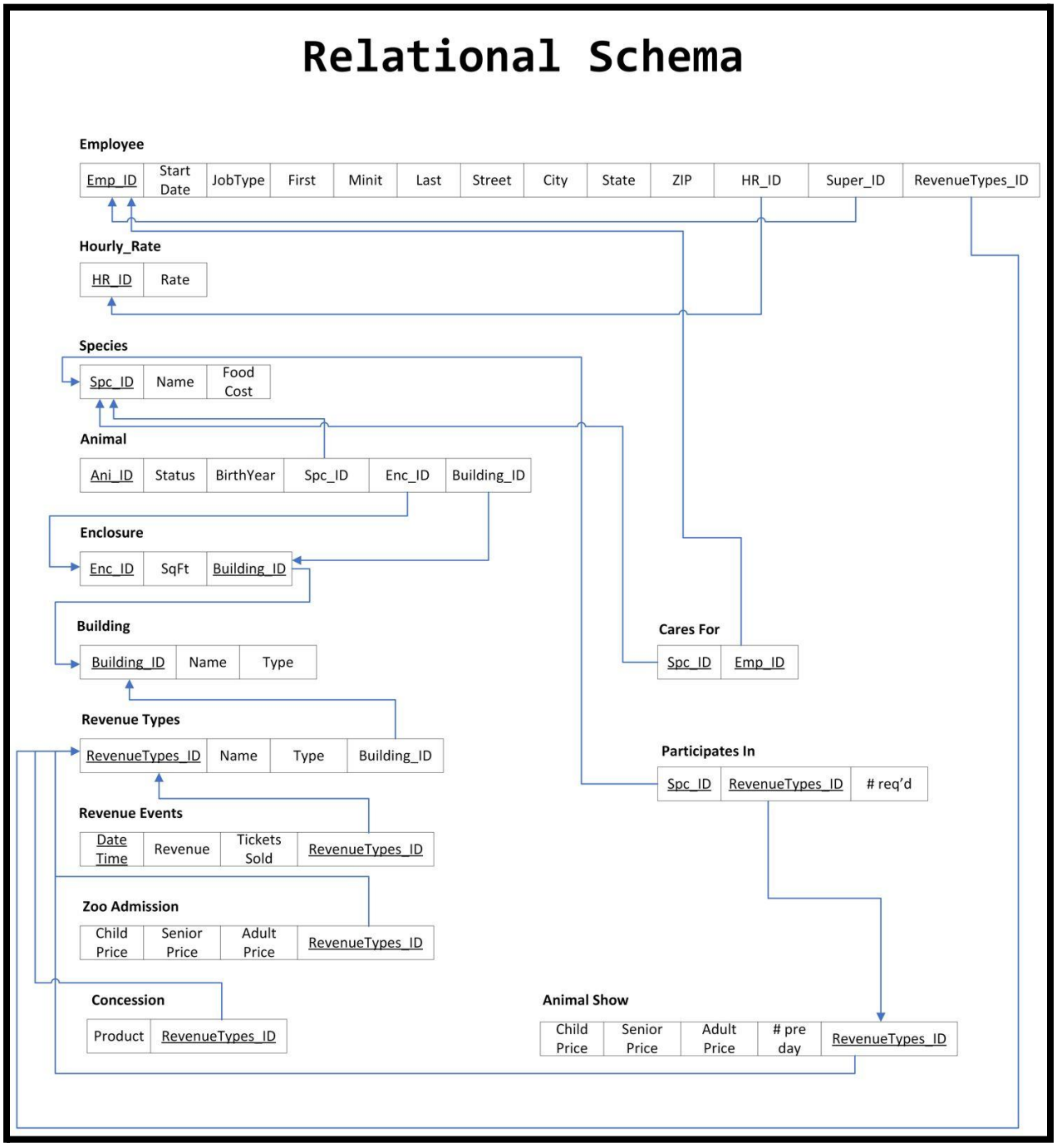
- Ensure accurate identification of primary keys and foreign keys in the relational database schema.

- Specify other keys in the text accompanying your schema.

- Indicate constraints, beyond referential integrity constraints, in the text accompanying each table.

- Highlight any challenges encountered during this phase of the project.

7. Relational Schema



Pdf attached at the last.

8. Primary key and Foreign Key

There are 13 Tables in our Relational Scheme and they are :

1. **According to step-1 map all strong entities.**

There are following strong entities in the given EER diagram:

- Employee
- Animal
- Hourly rate
- Species
- Building
- Revenue Type

So we created the Relational Schema of all the above mentioned entities with all their respective attributes.

2. **According to step-2 map all weak entities.**

There are following weak entities in the given EER diagram:

- Revenue Events
- Enclosure

So we created the Relational Schema of the above mentioned entities which has Primary Key as the combination of its partial key and Primary Key of Identifying Entity.

In Revenue Events partial key is Date Time and Primary key of Identifying entity is Revenue_type_ID. So, both of them combine to form the primary key of Revenue Events in Relational Schema.

In Enclosure partial key is Enc_ID and Primary key of Identifying entity is Building_ID. So, both of them combine to form the primary key of Enclosure in Relational Schema.

3. **According to step-3 map all 1:N Realtionship Type.**

In this primary key of the 1 side relation is given to the Entity on the N side of the relationship as a Foreign key.

There are following 1:N Relationship Type in the given EER diagram:

- Earns

  Earns is a relationship between Hourly Rate and Employee. HR_ID is given to employee as foreign key.
- Member of

  Member of is a relationship between Animal and Species. Spc_ID is given to Animal as foreign key.
- Lives In

  Lives In of is a relationship between Animal and Enclosure. Enc_ID and Building_ID is given to Animal as foreign key.
- Hosts

  Hosts of is a relationship between Building and Revenue Type. Building_ID is given to Revenue Type as foreign key.
- Works At

  Works At of is a relationship between sub class of Revenue Type and sub class of Employee. Revenue_Type_ID is given to Employee as foreign key.
- Supervises

  Supervises of is a relationship between Employee and Employee. Super_ID is given to Employee as foreign key.

4. **According to step-4 map all M:N Realtionship Type.**

In this a separate table for the relationship is created in relation schema using the attributes of the relation and the primary key of the entities related to it.

There are following M:N Relationship Type in the given EER diagram:

- Cares For

  Cares For is a relationship between sub class of Employee and species. A separate table is created with the attributes of the relationship and the primary key of the employee (EMP_ID) and the species (SPC_ID) entity.

- Participates In

  Participates In is a relationship between Sub class of Revenue Types and species. A separate table is created with the attributes of the relationship, that is #reg'd and the primary key of the Revenue Type (Revenue_Type_ID) and the species (SPC_ID) entity.

5. According to step-5 mapping Specialization or Generalization.

There are following Specialization or Generalizationin the given EER diagram:

- Employee and its Sub classes

  In this the rule 8C is applied, in which all the attributes of the sub class is given to the super class. As in this case there are no attributes in the sub class so there will be only attributes of the Employee.

- Revenue Type and its sub class

  In this the rule 8A is applied, in which the primary key of the Revenue type is given to the sub classes and different tables are created using the primary key of the super class and the attributes of the sub class for each sub class.
  So Animal Show will have Revenue_Type_ID as PK and attributes as Senior Price, Adult Price, Child Price and #Per day.
  Concession will have Revenue_Type_ID as PK and Product as its attributes.
  Zoo Admission will have Revenue_Type_ID as PK and Senior Price, Adult Price and Child Price as its attributes.
  In this relation schema all the Foreign Keys are pointing towards its respective primary key.

9. Challenges faced on phase 2

- Creating a Relational Schema poses challenges, stemming from complexities in both the data structure and the mapping process.
- Mapping the intricate Enhanced Entity-Relationship (EER) diagram of Turtleback Zoo to a relational schema is particularly challenging due to the diverse nature of the zoo's operations.
- The relationships between entities, including animals, employees, buildings, and attractions, demand careful consideration to accurately represent dependencies and cardinalities in the relational schema.
- The specifics of employee types, incorporating details like degree year and species specialties, introduce additional complexity to the mapping process.
- Balancing normalization to eliminate redundancy while ensuring practical query performance is a delicate challenge that needs attention.
- Capturing the dynamic nature of data elements such as ticket prices, employee hourly rates, and other periodically changing factors requires thoughtful schema design to maintain data consistency over time.

10. Phase 3 Goal

- Project's goal, detailing the creation of the database schema, instance, and application programs, along with any revisions to Phase 2 specifications, encountered problems and justify solutions.
- Include ample sample data in tables to effectively demonstrate required tasks.
- Execute SQL commands for table creation, incorporating primary, secondary, and foreign keys.
- Run SQL command files to populate each table with a sufficient number of tuples, considering relationships between tables with no integrity violations.
- Develop a menu-driven application system for the Turtleback Zoo database, aligning with the functional requirements outlined on the next page.

11. Creating Tables

```
CREATE TABLE BUILDING(

Building_ID VARCHAR(10) NOT NULL,

Build_Name VARCHAR(30) NOT NULL,

Buid_Type VARCHAR(50) NOT NULL,

PRIMARY KEY (Building_ID)

) ;

CREATE TABLE REVENUE_TYPES(

RevenueTypes_ID VARCHAR(10) NOT NULL,

Rev_Name VARCHAR(30) NOT NULL,

Rev_Type VARCHAR(30) NOT NULL,

Building_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (RevenueTypes_ID),

CONSTRAINT revt_fk_bui FOREIGN KEY (Building_ID) REFERENCES BUILDING(Building_ID)
```

```
) ;

CREATE TABLE HOURLY_RATE(

Hr_ID VARCHAR(10) NOT NULL,

Rate INT NOT NULL,

PRIMARY KEY (Hr_ID)

) ;

CREATE TABLE EMPLOYEE(

Emp_ID CHAR(9) NOT NULL,

Start_Date DATE NOT NULL,

JobType VARCHAR(15) NOT NULL,

Fname VARCHAR(15) NOT NULL,

Minit CHAR NOT NULL,

Lname VARCHAR(15) NOT NULL,

Street VARCHAR(30) NOT NULL,

City VARCHAR(15) NOT NULL,

State_Name VARCHAR(15) NOT NULL,

PinCode CHAR(5) NOT NULL,

Hr_ID VARCHAR(10) NOT NULL,

Super_ID CHAR(9),

RevenueTypes_ID VARCHAR(10),

PRIMARY KEY (Emp_ID),

CONSTRAINT emp_fk_hou FOREIGN KEY (Hr_ID) REFERENCES HOURLY_RATE(Hr_ID),

CONSTRAINT emp_fk_rty FOREIGN KEY (RevenueTypes_ID) REFERENCES
REVENUE_TYPES(RevenueTypes_ID)

) ;
```

```
CREATE TABLE SPECIES(

Spc_ID VARCHAR(10) NOT NULL,

Spc_Name VARCHAR(30) NOT NULL,

Food_Cost INT,

PRIMARY KEY (Spc_ID),

UNIQUE (Spc_Name)

) ;

CREATE TABLE ENCLOSURE(

Enc_ID VARCHAR(10) NOT NULL,

Sq_ft INT NOT NULL,

Building_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (Enc_ID,Building_ID),

CONSTRAINT enc_fk_bui FOREIGN KEY (Building_ID) REFERENCES BUILDING(Building_ID)

) ;

CREATE TABLE ANIMAL(

Ani_ID VARCHAR(10) NOT NULL,

Status VARCHAR(10) NOT NULL,

Birth_Year CHAR(4) NOT NULL,

Spc_ID VARCHAR(10) NOT NULL,

Enc_ID VARCHAR(10) NOT NULL,

Building_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (Ani_ID),

CONSTRAINT ani_fk_spc FOREIGN KEY (Spc_ID) REFERENCES SPECIES(Spc_ID),

CONSTRAINT ani_fk_enc FOREIGN KEY (Enc_ID,Building_ID) REFERENCES
ENCLOSURE(Enc_ID,Building_ID)
```

```
--CONSTRAINT ani_fk_encb FOREIGN KEY (Building_ID) REFERENCES ENCLOSURE(Building_ID)

) ;

CREATE TABLE REVENUE_EVENTS(

Date_Time DATE NOT NULL,

Revenue INT NOT NULL,

Tickets_Sold INT NOT NULL,

RevenueTypes_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (Date_Time,RevenueTypes_ID),

CONSTRAINT reve_fk_revt FOREIGN KEY (RevenueTypes_ID) REFERENCES
REVENUE_TYPES(RevenueTypes_ID)

) ;

CREATE TABLE ZOO_ADMISSION(

Child_Price INT NOT NULL,

Adult_Price INT NOT NULL,

Senior_Price INT NOT NULL,

RevenueTypes_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (RevenueTypes_ID),

CONSTRAINT zoo_fk_revt FOREIGN KEY (RevenueTypes_ID) REFERENCES
REVENUE_TYPES(RevenueTypes_ID)

) ;

CREATE TABLE CONCESSION(

Product VARCHAR(10) NOT NULL,

RevenueTypes_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (RevenueTypes_ID),
```

```
CONSTRAINT con_fk_revt FOREIGN KEY (RevenueTypes_ID) REFERENCES
REVENUE_TYPES(RevenueTypes_ID)

) ;

CREATE TABLE CARES_FOR(

Emp_ID CHAR(9) NOT NULL,

Spc_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (Emp_ID,Spc_ID),

CONSTRAINT cf_fk_emp FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE(Emp_ID),

CONSTRAINT cf_fk_spc FOREIGN KEY (Spc_ID) REFERENCES SPECIES(Spc_ID)

) ;

CREATE TABLE ANIMAL_SHOW(

Child_Price INT NOT NULL,

Adult_Price INT NOT NULL,

Senior_Price INT NOT NULL,

Shows_Per_Day INT NOT NULL,

RevenueTypes_ID VARCHAR(10) NOT NULL,

PRIMARY KEY (RevenueTypes_ID),

CONSTRAINT anis_fk_revt FOREIGN KEY (RevenueTypes_ID) REFERENCES
REVENUE_TYPES(RevenueTypes_ID)

) ;

CREATE TABLE PARTICIPATES_IN(

Spc_ID VARCHAR(10) NOT NULL,

RevenueTypes_ID VARCHAR(10) NOT NULL,

Num_spc_req INT NOT NULL,

PRIMARY KEY (Spc_ID,RevenueTypes_ID),
```

```
CONSTRAINT pi_fk_spc FOREIGN KEY (Spc_ID) REFERENCES SPECIES(Spc_ID),

CONSTRAINT pi_fk_anis FOREIGN KEY (RevenueTypes_ID) REFERENCES
ANIMAL_SHOW(RevenueTypes_ID)

) ;
```

12. Populating Tables

INSERT INTO HOURLY_RATE(Hr_ID , Rate) VALUES(1,50);

INSERT INTO HOURLY_RATE(Hr_ID , Rate) VALUES(2,45);

INSERT INTO HOURLY_RATE(Hr_ID , Rate) VALUES(3,40);

INSERT INTO HOURLY_RATE(Hr_ID , Rate) VALUES(4,35);

INSERT INTO HOURLY_RATE(Hr_ID , Rate) VALUES(5,30);

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1001','Kings Landing','Animal Exhibits');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1002','Feathered Flyers Terminal','Aviaries');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1003','Fishy Business Plaza','Aquariums');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1004','Slither Inn','Reptile House');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1005','Bug Bistro','Insectariums');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1006','Conservaplex','Conservation Centers');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1007','Penguin Paradise Pub','Penguin exhibit');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1008','Turtleville Towers','Turtle habitat');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1009','Chomp and Stomp Caf&#65533;','Restaurants and Cafes');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1010','Zooper Souvenirs Emporium','Gift Shops');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1011','Flutterby Fun Zone','Butterfly garden');

INSERT INTO BUILDING(Building_ID , Build_Name, Buid_Type) VALUES('B1012','Zoo-topia Theater','Theaters and Show Arenas');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT101', 'Ticket Sales', 'General Admission', 'B1001');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT102', 'Bird House Revenue', 'Bird House Admission', 'B1002');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT103', 'Aquariums Revenue', 'Aquarium Admission', 'B1003');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT104', 'Reptiles Revenue', 'Reptile House Admission', 'B1004');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT105', 'Insectariums Revenue', 'Insectariums Admission', 'B1005');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT106', 'Conservation Revenue', 'Conservation Admission', 'B1006');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT107', 'Penguin Revenue', 'Penguin Admission', 'B1007');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT108', 'Turtle habitat Revenue', 'Turtle habitat Admission', 'B1008');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT109', 'Food Court', 'Food and Beverage', 'B1009');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT110', 'Gift Shop', 'Retail', 'B1010');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT111', 'Butterfly garden Ticket Sales', 'Butterfly garden Admission', 'B1011');

INSERT INTO REVENUE_TYPES (RevenueTypes_ID, Rev_Name, Rev_Type, Building_ID) VALUES('RT112', 'Event Tickets', 'Events and Shows', 'B1012');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('563214789', TO_DATE('2022-01-01', 'YYYY-MM-DD'), 'Veterinarian', 'John', 'D', 'Doe', '123 Main St', 'New York', 'NY', '10001', 1, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('890456132', TO_DATE('2022-02-01', 'YYYY-MM-DD'), 'Animal Care', 'Jane', 'A', 'Smith', '456 Oak St', 'Los Angeles', 'CA', '90001', 2, '563214789', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('123987654', TO_DATE('2022-03-01', 'YYYY-MM-DD'), 'Veterinarian', 'Bob', 'B', 'Johnson', '789 Pine St', 'Chicago', 'IL', '60601', 1, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('657890432', TO_DATE('2022-04-01', 'YYYY-MM-DD'), 'Maintanance', 'Alice', 'C', 'Williams', '101 Maple St', 'Houston', 'TX', '77001', 3, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('214356789',

TO_DATE('2022-05-01', 'YYYY-MM-DD'), 'Ticket Seller', 'David', 'E', 'Jones', '202 Cedar St', 'Phoenix', 'AZ', '85001', 4, NULL, 'RT101');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('987654321', TO_DATE('2022-06-01', 'YYYY-MM-DD'), 'Veterinarian', 'Eva', 'F', 'Brown', '303 Elm St', 'Philadelphia', 'PA', '19101', 1, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('345678901', TO_DATE('2022-07-01', 'YYYY-MM-DD'), 'Maintanance', 'Frank', 'G', 'Miller', '404 Birch St', 'San Antonio', 'TX', '78201', 3, '657890432', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('876543210', TO_DATE('2022-08-01', 'YYYY-MM-DD'), 'CustomerService', 'Grace', 'H', 'Davis', '505 Oak St', 'San Diego', 'CA', '92101', 5, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('109876543', TO_DATE('2022-09-01', 'YYYY-MM-DD'), 'Animal Care', 'Henry', 'I', 'Taylor', '606 Pine St', 'Dallas', 'TX', '75201', 2, '563214789', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('432109876', TO_DATE('2022-10-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Ivy', 'J', 'Anderson', '707 Cedar St', 'San Jose', 'CA', '95101', 4, NULL, 'RT102');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('765432109', TO_DATE('2022-11-01', 'YYYY-MM-DD'), 'Veterinarian', 'Jack', 'K', 'Wilson', '808 Elm St', 'Austin', 'TX', '73301', 1, NULL, NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('210987654',

TO_DATE('2022-12-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Katie', 'L', 'Moore', '909 Birch St', 'Jacksonville', 'FL', '32201', 4, NULL, 'RT103');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('543210987', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Animal Care', 'Leo', 'M', 'Martin', '1010 Oak St', 'Indianapolis', 'IN', '46201', 2, '123987654', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('678901234', TO_DATE('2023-02-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Mia', 'N', 'Clark', '1111 Pine St', 'San Francisco', 'CA', '94101', 4, NULL,'RT104');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('321098765', TO_DATE('2023-03-01', 'YYYY-MM-DD'), 'CustomerService', 'Mike', 'O', 'Connor', '333 Pine St', 'Columbus', 'OH', '43201', 5, '876543210', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('876543219', TO_DATE('2023-04-01', 'YYYY-MM-DD'), 'Animal Care', 'Natalie', 'P', 'Perez', '444 Cedar Dr', 'Charlotte', 'NC', '28201', 2, '987654321', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('432109875', TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'CustomerService', 'Oscar', 'Q', 'Quinn', '555 Elm Ln', 'San Francisco', 'CA', '94101', 5, '876543210', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('987654310', TO_DATE('2023-06-01', 'YYYY-MM-DD'), 'Maintanance', 'Paula', 'R', 'Ramirez', '666 Maple Rd', 'Seattle', 'WA', '98101', 3, '657890432', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('234567890',

TO_DATE('2023-07-01', 'YYYY-MM-DD'), 'Animal Care', 'Quincy', 'S', 'Sullivan', '777 Pine Ave', 'Denver', 'CO', '80201', 2, '765432109', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('890123456', TO_DATE('2023-08-01', 'YYYY-MM-DD'), 'Maintanance', 'Rachel', 'T', 'Thomas', '888 Cedar Blvd', 'Las Vegas', 'NV', '89101', 3, '657890432', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('871123219', TO_DATE('2024-09-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Sam', 'U', 'Smith', '999 Oak St', 'Miami', 'FL', '33101', 4, NULL, 'RT105');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('876543112', TO_DATE('2024-10-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Tina', 'V', 'Taylor', '123 Elm Ave', 'Minneapolis', 'MN', '55401', 4, NULL, 'RT106');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('112543219', TO_DATE('2024-11-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Ulysses', 'W', 'Williams', '456 Pine Ln', 'Portland', 'OR', '97201', 4, NULL, 'RT107');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('811243219', TO_DATE('2024-12-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Victoria', 'X', 'Jones', '789 Cedar Dr', 'Atlanta', 'GA', '30301', 4, NULL, 'RT108');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('432100006', TO_DATE('2025-01-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Walter', 'Y', 'Young', '101 Maple Blvd', 'Detroit', 'MI', '48201', 4, NULL, 'RT109');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('411109876',

TO_DATE('2025-02-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Xena', 'Z', 'Zhang', '202 Oak Ave', 'San Francisco', 'CA', '94101', 4, NULL, 'RT110');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('111109876', TO_DATE('2025-03-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Yvonne', 'A', 'Adams', '303 Pine St', 'Houston', 'TX', '77001', 4, NULL, 'RT111');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('432111876', TO_DATE('2025-04-01', 'YYYY-MM-DD'), 'Ticket Seller', 'Zachary', 'B', 'Brown', '404 Elm Ln', 'Chicago', 'IL', '60601', 4, NULL, 'RT112');

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('120987654', TO_DATE('2025-05-01', 'YYYY-MM-DD'), 'Maintanance', 'Olivia', 'C', 'Chen', '505 Cedar Rd', 'Phoenix', 'AZ', '85001', 3, '657890432', NULL);

INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID, RevenueTypes_ID) VALUES('123981112', TO_DATE('2025-06-01', 'YYYY-MM-DD'), 'CustomerService', 'Elijah', 'D', 'Davis', '606 Pine Ct', 'Seattle', 'WA', '98101', 5, '876543210', NULL);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S001', 'Lion', 500);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S002', 'Tiger', 550);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S003', 'Elephant', 700);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S004', 'Giraffe', 400);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S005', 'Zebra', 350);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S006', 'Gorilla', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S007', 'Kangaroo', 250);

```
INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S008', 'Cheetah', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S009', 'Direwolf', 600);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S010', 'Koala', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S011', 'Hippopotamus', 450);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S012', 'Rhinoceros', 480);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S013', 'Camel', 550);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S014', 'Leopard', 400);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S015', 'Wolf', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S016', 'Peacock', 500);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S017', 'Eagle', 350);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S018', 'Sparrow', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S019', 'Flamingo', 600);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S020', 'Hawk', 480);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S021', 'Hornbill', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S022', 'Kookaburra', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S023', 'Owl', 280);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S024', 'Macaw', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S025', 'Parrot', 520);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S026', 'Sloth', 220);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S027', 'Woodpecker', 320);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S028', 'KingFisher', 200);
```

```
INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S029', 'Pelican', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S030', 'Swan', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S031', 'Octopus', 250);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S032', 'Lobster', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S033', 'Jellyfish', 400);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S034', 'Starfish', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S035', 'Goldfish', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S036', 'Dolphin', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S037', 'Shark', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S038', 'Crab', 520);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S039', 'Stingray ', 450);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S040', 'Eel', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S041', 'Beetles', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S042', 'Ants', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S043', 'Honeybees', 80);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S044', 'Dragonflies', 90);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S045', 'Mantises', 280);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S046', 'Cockroaches', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S047', 'Stick Insects', 70);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S048', 'Walking Sticks', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S049', 'Ladybugs', 220);
```

```
INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S050', 'Grasshoppers', 350);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S051', 'Tarantulas', 280);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S052', 'Scorpions', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S053', 'Moths', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S054', 'Chameleon', 90);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S055', 'Cicadas', 480);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S056', 'Amur Leopard', 80);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S057', 'Vaquita ', 250);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S058', 'Javan Rhino', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S059', 'Blue Whale', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S060', 'Vulture', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S061', 'Red Wolf ', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S062', 'Kakapo ', 400);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S063', 'Houston Toad ', 70);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S064', 'Golden Lion Tamarin',
150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S065', 'Bali Mynah', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S066', 'Mud Turtle', 80);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S067', 'Alligator Snapping
Turtle ', 70);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S068', 'Red-Eared Slider',
200);
```

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S069', 'Hawksbill Sea Turtle', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S070', 'Box Turtle ', 350);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S071', 'Leatherback Sea Turtle', 280);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S072', 'Snapping Turtle', 90);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S073', 'Green Sea Turtle', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S074', 'Painted Turtle', 220);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S075', 'Loggerhead Sea Turtle', 200);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S076', 'Emperor Penguin', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S077', 'King Penguin', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S078', 'Ad�lie Penguin ', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S079', 'Chinstrap Penguin', 450);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S080', 'Gentoo Penguin', 520);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S081', 'Monarch Butterfly', 70);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S082', 'Swallowtail Butterfly', 220);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S083', 'Painted Lady Butterfly', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S084', 'Blue Morpho Butterfly', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S085', 'Peacock Butterfly', 250);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S086', 'Red Admiral Butterfly', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S087', 'Common Tiger Butterfly', 80);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S088', 'Eastern Tailed-Blue Butterfly', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S089', 'Black Swallowtail Butterfly', 350);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S090', 'Orange-tip Butterfly', 80);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S091', 'Galapagos Tortoise', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S092', 'Nile Crocodile ', 70);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S093', 'Tuatara', 100);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S094', 'Gopher Snake', 180);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S095', 'Komodo Dragon', 120);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S096', 'Leopard Gecko', 250);

```
INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S097', 'American Alligator', 300);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S098', 'Ball Python', 220);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S099', 'Green Iguana', 150);

INSERT INTO SPECIES (Spc_ID, Spc_Name, Food_Cost) VALUES('S100', 'Viper', 500);


INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC001', 350, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC002', 450, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC003', 550, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC004', 650, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC005', 750, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC006', 850, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC007', 950, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC008', 550, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC009', 650, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC010', 750, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC011', 850, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC012', 950, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC013', 550, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC014', 650, 'B1001');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC015', 750, 'B1001');
```

```
INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC016', 850, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC017', 950, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC018', 550, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC019', 650, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC020', 750, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC021', 350, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC022', 450, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC023', 550, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC024', 650, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC025', 750, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC026', 850, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC027', 950, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC028', 550, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC029', 650, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC030', 750, 'B1002');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC031', 850, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC032', 950, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC033', 550, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC034', 650, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC035', 750, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC036', 850, 'B1003');
```

```
INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC037', 950, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC038', 550, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC039', 650, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC040', 750, 'B1003');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC041', 350, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC042', 450, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC043', 550, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC044', 650, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC045', 750, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC046', 850, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC047', 950, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC048', 550, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC049', 650, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC050', 750, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC051', 850, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC052', 950, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC053', 550, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC054', 650, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC055', 750, 'B1005');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC056', 850, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC057', 950, 'B1006');
```

```
INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC058', 550, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC059', 650, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC060', 750, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC061', 350, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC062', 450, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC063', 550, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC064', 650, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC065', 750, 'B1006');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC066', 850, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC067', 950, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC068', 550, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC069', 650, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC070', 750, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC071', 850, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC072', 950, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC073', 550, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC074', 650, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC075', 750, 'B1008');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC076', 850, 'B1007');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC077', 950, 'B1007');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC078', 550, 'B1007');
```

```
INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC079', 650, 'B1007');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC080', 750, 'B1007');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC081', 350, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC082', 450, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC083', 550, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC084', 650, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC085', 750, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC086', 850, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC087', 950, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC088', 550, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC089', 650, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC090', 750, 'B1011');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC091', 850, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC092', 950, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC093', 550, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC094', 650, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC095', 750, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC096', 850, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC097', 950, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC098', 550, 'B1004');

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC099', 650, 'B1004');
```

INSERT INTO ENCLOSURE (Enc_ID, Sq_ft, Building_ID) VALUES ('EC100', 750, 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID ,Enc_ID , Building_ID) VALUES ('A001', 'Healthy', '2001','S001', 'EC001', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID ,Enc_ID , Building_ID) VALUES ('A002', 'Healthy', '2004','S001', 'EC001', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID ,Enc_ID , Building_ID) VALUES ('A003', 'Ill', '2005','S001', 'EC001', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID ,Enc_ID , Building_ID) VALUES ('A004', 'Healthy', '2007','S001', 'EC001', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID ,Enc_ID , Building_ID) VALUES ('A005', 'Healthy', '2002','S001', 'EC001', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A006', 'Healthy', 2009, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A007', 'Ill', 2016, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A008', 'Healthy', 2000, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A009', 'Ill', 2020, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A010', 'Healthy', 2014, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A011', 'Ill', 2005, 'S002', 'EC002', 'B1001');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A012', 'Healthy', 2023, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A013', 'Ill', 2012, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A014', 'Healthy', 2018, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A015', 'Ill', 2007, 'S002', 'EC002', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A016', 'Healthy', 2008, 'S003', 'EC003', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A017', 'Ill', 2005, 'S003', 'EC003', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A018', 'Healthy', 2002, 'S003', 'EC003', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A019', 'Ill', 2010, 'S003', 'EC003', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A020', 'Healthy', 2012, 'S003', 'EC003', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A021', 'Healthy', 2009, 'S004', 'EC004', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A022', 'Ill', 2004, 'S004', 'EC004', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A023', 'Healthy', 2003, 'S004', 'EC004', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A024', 'Ill', 2011, 'S004', 'EC004', 'B1001');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A025', 'Healthy', 2010, 'S004', 'EC004', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A026', 'Healthy', 2013, 'S005', 'EC005', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A027', 'Ill', 2010, 'S005', 'EC005', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A028', 'Healthy', 2005, 'S005', 'EC005', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A029', 'Ill', 2008, 'S006', 'EC006', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A030', 'Healthy', 2006, 'S006', 'EC006', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A031', 'Ill', 2012, 'S006', 'EC006', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A032', 'Healthy', 2009, 'S007', 'EC007', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A033', 'Ill', 2004, 'S007', 'EC007', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A034', 'Healthy', 2011, 'S007', 'EC007', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A035', 'Healthy', 2013, 'S008', 'EC008', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A036', 'Ill', 2010, 'S008', 'EC008', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A037', 'Healthy', 2005, 'S008', 'EC008', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A038', 'Healthy', 2008, 'S009', 'EC009', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A039', 'Ill', 2006, 'S009', 'EC009', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A040', 'Healthy', 2012, 'S009', 'EC009', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A041', 'Ill', 2009, 'S010', 'EC010', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A042', 'Healthy', 2004, 'S010', 'EC010', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A043', 'Ill', 2011, 'S010', 'EC010', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A044', 'Healthy', 2013, 'S011', 'EC011', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A045', 'Ill', 2010, 'S011', 'EC011', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A046', 'Healthy', 2005, 'S011', 'EC011', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A047', 'Ill', 2008, 'S012', 'EC012', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A048', 'Healthy', 2006, 'S012', 'EC012', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A049', 'Ill', 2012, 'S012', 'EC012', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A050', 'Healthy', 2009, 'S013', 'EC013', 'B1001');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A051', 'Ill', 2004, 'S013', 'EC013', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A052', 'Healthy', 2011, 'S013', 'EC013', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A053', 'Healthy', 2014, 'S014', 'EC014', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A054', 'Ill', 2011, 'S014', 'EC014', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A055', 'Healthy', 2006, 'S014', 'EC014', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A056', 'Ill', 2009, 'S015', 'EC015', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A057', 'Healthy', 2007, 'S015', 'EC015', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A058', 'Maternity', 2013, 'S015', 'EC015', 'B1001');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A059', 'Healthy', 2015, 'S016', 'EC016', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A060', 'Maternity', 2017, 'S016', 'EC016', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A061', 'Healthy', 2019, 'S016', 'EC016', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A062', 'Maternity', 2013, 'S017', 'EC017', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A063', 'Healthy', 2016, 'S017', 'EC017', 'B1002');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A064', 'Maternity', 2018, 'S017', 'EC017', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A065', 'Healthy', 2012, 'S018', 'EC018', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A066', 'Maternity', 2014, 'S018', 'EC018', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A067', 'Maternity', 2014, 'S018', 'EC018', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A068', 'Healthy', 2015, 'S019', 'EC019', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A069', 'Maternity', 2016, 'S019', 'EC019', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A070', 'Healthy', 2012, 'S019', 'EC019', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A071', 'Healthy', 2018, 'S019', 'EC019', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A072', 'Maternity', 2014, 'S019', 'EC019', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A073', 'Healthy', 2016, 'S020', 'EC020', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A074', 'Maternity', 2012, 'S020', 'EC020', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A075', 'Healthy', 2018, 'S020', 'EC020', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A076', 'Maternity', 2015, 'S020', 'EC020', 'B1002');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A077', 'Healthy', 2020, 'S020', 'EC020', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A078', 'Healthy', 2015, 'S021', 'EC021', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A079', 'Maternity', 2016, 'S021', 'EC021', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A080', 'Healthy', 2017, 'S021', 'EC021', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A081', 'Maternity', 2018, 'S021', 'EC021', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A082', 'Healthy', 2019, 'S021', 'EC021', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A083', 'Maternity', 2014, 'S022', 'EC022', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A084', 'Healthy', 2015, 'S022', 'EC022', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A085', 'Maternity', 2016, 'S022', 'EC022', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A086', 'Healthy', 2017, 'S022', 'EC022', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A087', 'Maternity', 2018, 'S022', 'EC022', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A088', 'Healthy', 2019, 'S023', 'EC023', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A089', 'Maternity', 2020, 'S023', 'EC023', 'B1002');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A090', 'Healthy', 2021, 'S024', 'EC024', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A091', 'Maternity', 2022, 'S024', 'EC024', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A092', 'Healthy', 2020, 'S025', 'EC025', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A093', 'Maternity', 2021, 'S025', 'EC025', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A094', 'Healthy', 2018, 'S026', 'EC026', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A095', 'Maternity', 2019, 'S026', 'EC026', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A096', 'Healthy', 2022, 'S027', 'EC027', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A097', 'Ill', 2023, 'S027', 'EC027', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A098', 'Healthy', 2020, 'S028', 'EC028', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A099', 'Ill', 2021, 'S028', 'EC028', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A100', 'Healthy', 2023, 'S029', 'EC029', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A101', 'Ill', 2024, 'S029', 'EC029', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A102', 'Healthy', 2019, 'S030', 'EC030', 'B1002');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A103', 'Healthy', 2013, 'S030', 'EC030', 'B1002');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A104', 'Ill', 2015, 'S031', 'EC031', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A105', 'Healthy', 2017, 'S031', 'EC031', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A106', 'Ill', 2019, 'S031', 'EC031', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A107', 'Healthy', 2021, 'S031', 'EC031', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A108', 'Ill', 2023, 'S031', 'EC031', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A109', 'Healthy', 2012, 'S032', 'EC032', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A110', 'Ill', 2014, 'S032', 'EC032', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A111', 'Healthy', 2016, 'S032', 'EC032', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A112', 'Ill', 2018, 'S032', 'EC032', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A113', 'Healthy', 2020, 'S032', 'EC032', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A114', 'Ill', 2011, 'S033', 'EC033', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A115', 'Healthy', 2013, 'S033', 'EC033', 'B1003');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A116', 'Ill', 2015, 'S033', 'EC033', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A117', 'Healthy', 2017, 'S033', 'EC033', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A118', 'Healthy', 2017, 'S033', 'EC033', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A119', 'Healthy', 2016, 'S034', 'EC034', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A120', 'Ill', 2019, 'S034', 'EC034', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A121', 'Healthy', 2018, 'S034', 'EC034', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A122', 'Ill', 2020, 'S034', 'EC034', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A123', 'Healthy', 2022, 'S034', 'EC034', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A124', 'Healthy', 2015, 'S035', 'EC035', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A125', 'Ill', 2020, 'S035', 'EC035', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A126', 'Healthy', 2018, 'S035', 'EC035', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A127', 'Ill', 2021, 'S035', 'EC035', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A128', 'Healthy', 2023, 'S035', 'EC035', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A129', 'Healthy', 2017, 'S036', 'EC036', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A130', 'Ill', 2022, 'S036', 'EC036', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A131', 'Healthy', 2020, 'S036', 'EC036', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A132', 'Ill', 2023, 'S036', 'EC036', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A133', 'Healthy', 2019, 'S037', 'EC037', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A134', 'Ill', 2017, 'S037', 'EC037', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A135', 'Healthy', 2021, 'S037', 'EC037', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A136', 'Healthy', 2019, 'S037', 'EC037', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A137', 'Healthy', 2019, 'S037', 'EC037', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A138', 'Ill', 2017, 'S038', 'EC038', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A139', 'Healthy', 2021, 'S038', 'EC038', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A140', 'Healthy', 2019, 'S038', 'EC038', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A141', 'Healthy', 2019, 'S039', 'EC039', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A142', 'Ill', 2017, 'S039', 'EC039', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A143', 'Healthy', 2021, 'S039', 'EC039', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A144', 'Healthy', 2019, 'S039', 'EC039', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A145', 'Healthy', 2019, 'S040', 'EC040', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A146', 'Ill', 2017, 'S040', 'EC040', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A147', 'Healthy', 2021, 'S040', 'EC040', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A148', 'Healthy', 2019, 'S040', 'EC040', 'B1003');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A149', 'Maternity', 2017, 'S041', 'EC041', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A150', 'Healthy', 2015, 'S041', 'EC041', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A151', 'Maternity', 2013, 'S041', 'EC041', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A152', 'Healthy', 2011, 'S041', 'EC041', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A153', 'Maternity', 2018, 'S042', 'EC042', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A154', 'Healthy', 2016, 'S042', 'EC042', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A155', 'Maternity', 2014, 'S042', 'EC042', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A156', 'Healthy', 2012, 'S042', 'EC042', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A157', 'Maternity', 2017, 'S043', 'EC043', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A158', 'Healthy', 2015, 'S043', 'EC043', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A159', 'Maternity', 2013, 'S043', 'EC043', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A160', 'Healthy', 2011, 'S043', 'EC043', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A161', 'Maternity', 2008, 'S044', 'EC044', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A162', 'Healthy', 2006, 'S044', 'EC044', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A163', 'Maternity', 2004, 'S044', 'EC044', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A164', 'Healthy', 2002, 'S044', 'EC044', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A165', 'Maternity', 2009, 'S045', 'EC045', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A166', 'Healthy', 2007, 'S045', 'EC045', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A167', 'Maternity', 2005, 'S045', 'EC045', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A168', 'Healthy', 2003, 'S045', 'EC045', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A169', 'Maternity', 2010, 'S046', 'EC046', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A170', 'Healthy', 2008, 'S046', 'EC046', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A171', 'Maternity', 2006, 'S046', 'EC046', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A172', 'Healthy', 2004, 'S046', 'EC046', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A173', 'Maternity', 1998, 'S047', 'EC047', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A174', 'Healthy', 1993, 'S047', 'EC047', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A175', 'Maternity', 1990, 'S047', 'EC047', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A176', 'Healthy', 1988, 'S047', 'EC047', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A177', 'Healthy', 2001, 'S048', 'EC048', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A178', 'Maternity', 1995, 'S048', 'EC048', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A179', 'Healthy', 1991, 'S048', 'EC048', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A180', 'Maternity', 1987, 'S048', 'EC048', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A181', 'Healthy', 1998, 'S049', 'EC049', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A182', 'Maternity', 1992, 'S049', 'EC049', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A183', 'Healthy', 1989, 'S049', 'EC049', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A184', 'Maternity', 1985, 'S049', 'EC049', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A185', 'Healthy', 1980, 'S050', 'EC050', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A186', 'Maternity', 1975, 'S050', 'EC050', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A187', 'Healthy', 1970, 'S050', 'EC050', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A188', 'Maternity', 1965, 'S050', 'EC050', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A189', 'Healthy', 1982, 'S051', 'EC051', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A190', 'Maternity', 1977, 'S051', 'EC051', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A191', 'Healthy', 1972, 'S051', 'EC051', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A192', 'Maternity', 1967, 'S051', 'EC051', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A193', 'Healthy', 1984, 'S052', 'EC052', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A194', 'Maternity', 1979, 'S052', 'EC052', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A195', 'Healthy', 1974, 'S052', 'EC052', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A196', 'Maternity', 1969, 'S052', 'EC052', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A197', 'Healthy', 2008, 'S053', 'EC053', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A198', 'Maternity', 2000, 'S053', 'EC053', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A199', 'Healthy', 1992, 'S053', 'EC053', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A200', 'Maternity', 1984, 'S053', 'EC053', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A201', 'Healthy', 2007, 'S054', 'EC054', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A202', 'Maternity', 1999, 'S054', 'EC054', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A203', 'Healthy', 1991, 'S054', 'EC054', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A204', 'Maternity', 1983, 'S054', 'EC054', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A205', 'Healthy', 2006, 'S055', 'EC055', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A206', 'Maternity', 1998, 'S055', 'EC055', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A207', 'Healthy', 1990, 'S055', 'EC055', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A208', 'Maternity', 1982, 'S055', 'EC055', 'B1005');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A209', 'Healthy', 2005, 'S056', 'EC056', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A210', 'Maternity', 1997, 'S056', 'EC056', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A211', 'Healthy', 2013, 'S056', 'EC056', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A212', 'Maternity', 1989, 'S056', 'EC056', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A213', 'Healthy', 2002, 'S057', 'EC057', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A214', 'Maternity', 1994, 'S057', 'EC057', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A215', 'Healthy', 2010, 'S057', 'EC057', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A216', 'Ill', 1986, 'S057', 'EC057', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A217', 'Healthy', 1999, 'S058', 'EC058', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A218', 'Ill', 1991, 'S058', 'EC058', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A219', 'Healthy', 2007, 'S058', 'EC058', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A220', 'Ill', 1983, 'S058', 'EC058', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A221', 'Healthy', 1990, 'S059', 'EC059', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A222', 'Ill', 1978, 'S059', 'EC059', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A223', 'Healthy', 1987, 'S059', 'EC059', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A224', 'Ill', 1975, 'S059', 'EC059', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A225', 'Healthy', 1992, 'S060', 'EC060', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A226', 'Ill', 1980, 'S060', 'EC060', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A227', 'Healthy', 1989, 'S060', 'EC060', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A228', 'Ill', 1977, 'S060', 'EC060', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A229', 'Healthy', 1980, 'S061', 'EC061', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A230', 'Ill', 1975, 'S061', 'EC061', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A231', 'Healthy', 1983, 'S061', 'EC061', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A232', 'Ill', 1972, 'S061', 'EC061', 'B1006');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A233', 'Healthy', 1981, 'S062', 'EC062', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A234', 'Ill', 1978, 'S062', 'EC062', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A235', 'Healthy', 1984, 'S062', 'EC062', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A236', 'Ill', 1974, 'S062', 'EC062', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A237', 'Healthy', 1982, 'S063', 'EC063', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A238', 'Ill', 1979, 'S063', 'EC063', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A239', 'Healthy', 1985, 'S063', 'EC063', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A240', 'Healthy', 1980, 'S064', 'EC064', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A241', 'Ill', 1975, 'S064', 'EC064', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A242', 'Healthy', 1970, 'S064', 'EC064', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A243', 'Ill', 1965, 'S064', 'EC064', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A244', 'Healthy', 1985, 'S065', 'EC065', 'B1006');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A245', 'Ill', 1980, 'S065', 'EC065', 'B1006');
```

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A246', 'Healthy', 2005, 'S066', 'EC066', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A247', 'Ill', 1998, 'S067', 'EC067', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A248', 'Healthy', 2008, 'S067', 'EC067', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A249', 'Ill', 1995, 'S068', 'EC068', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A250', 'Healthy', 2010, 'S068', 'EC068', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A251', 'Ill', 1992, 'S069', 'EC069', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A252', 'Healthy', 2012, 'S069', 'EC069', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A253', 'Ill', 1989, 'S070', 'EC070', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A254', 'Healthy', 2015, 'S070', 'EC070', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A255', 'Healthy', 2021, 'S071', 'EC071', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A256', 'Ill', 2018, 'S071', 'EC071', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A257', 'Healthy', 2020, 'S072', 'EC072', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A258', 'Ill', 2017, 'S072', 'EC072', 'B1008');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A259', 'Healthy', 2019, 'S073', 'EC073', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A260', 'Ill', 2016, 'S073', 'EC073', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A261', 'Healthy', 2018, 'S074', 'EC074', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A262', 'Ill', 2015, 'S074', 'EC074', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A263', 'Healthy', 2017, 'S075', 'EC075', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A264', 'Ill', 2014, 'S075', 'EC075', 'B1008');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A265', 'Healthy', 2019, 'S076', 'EC076', 'B1007');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A266', 'Ill', 2016, 'S077', 'EC077', 'B1007');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A267', 'Healthy', 2014, 'S078', 'EC078', 'B1007');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A268', 'Ill', 2013, 'S079', 'EC079', 'B1007');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A269', 'Healthy', 2012, 'S080', 'EC080', 'B1007');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A270', 'Healthy', 2019, 'S081', 'EC081', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A271', 'Ill', 2017, 'S081', 'EC081', 'B1011');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A272', 'Healthy', 2015, 'S081', 'EC081', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A273', 'Ill', 2020, 'S082', 'EC082', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A274', 'Healthy', 2018, 'S082', 'EC082', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A275', 'Ill', 2016, 'S082', 'EC082', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A276', 'Healthy', 2021, 'S083', 'EC083', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A277', 'Ill', 2019, 'S083', 'EC083', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A278', 'Healthy', 2017, 'S083', 'EC083', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A279', 'Ill', 2015, 'S084', 'EC084', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A280', 'Healthy', 2013, 'S084', 'EC084', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A281', 'Ill', 2011, 'S084', 'EC084', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A282', 'Healthy', 2018, 'S085', 'EC085', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A283', 'Ill', 2015, 'S085', 'EC085', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A284', 'Healthy', 2014, 'S086', 'EC086', 'B1011');
```

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A285', 'Ill', 2013, 'S087', 'EC087', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A286', 'Healthy', 2012, 'S088', 'EC088', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A287', 'Ill', 2011, 'S089', 'EC089', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A288', 'Healthy', 2010, 'S090', 'EC090', 'B1011');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A289', 'Ill', 2012, 'S091', 'EC091', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A290', 'Healthy', 2017, 'S091', 'EC091', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A291', 'Ill', 2014, 'S091', 'EC091', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A292', 'Healthy', 2016, 'S092', 'EC092', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A293', 'Maternity', 2019, 'S092', 'EC092', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A294', 'Healthy', 2013, 'S092', 'EC092', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A295', 'Maternity', 2015, 'S093', 'EC093', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A296', 'Healthy', 2010, 'S093', 'EC093', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A297', 'Maternity', 2018, 'S093', 'EC093', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A298', 'Healthy', 2011, 'S094', 'EC094', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A299', 'Maternity', 2014, 'S094', 'EC094', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A300', 'Healthy', 2019, 'S094', 'EC094', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A301', 'Healthy', 2020, 'S095', 'EC095', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A302', 'Maternity', 2018, 'S095', 'EC095', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A303', 'Healthy', 2016, 'S095', 'EC095', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A304', 'Maternity', 2020, 'S096', 'EC096', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A305', 'Healthy', 2019, 'S096', 'EC096', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A306', 'Maternity', 2017, 'S096', 'EC096', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A307', 'Healthy', 2022, 'S097', 'EC097', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A308', 'Maternity', 2021, 'S097', 'EC097', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A309', 'Healthy', 2018, 'S097', 'EC097', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES ('A310', 'Maternity', 2017, 'S098', 'EC098', 'B1004');

```
INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A311', 'Healthy', 2016, 'S098', 'EC098', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A312', 'Maternity', 2014, 'S098', 'EC098', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A313', 'Healthy', 2015, 'S099', 'EC099', 'B1004');

INSERT INTO ANIMAL (Ani_ID, Status, Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES
('A314', 'Healthy', 2015, 'S100', 'EC100', 'B1004');


INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-01', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-02', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-03', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-04', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-05', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-06', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-07', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-08', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-09', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-10', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-11', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-12', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-13', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-14', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-16', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-17', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-18', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-19', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-20', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-21', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-22', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-23', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FFLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-24', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-25', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-26', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-27', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-28', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-29', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-30', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(100000,
1000000)) + 1, FLOOR(dbms_random.value(1000, 10000)) + 1, 'RT101');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT102');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT103');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT104');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT105');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT106');
```

```
INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT107');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT108');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT109');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT110');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT111');

INSERT INTO REVENUE_EVENTS (Date_Time, Revenue, Tickets_Sold, RevenueTypes_ID)
VALUES (TO_DATE('2023-01-31', 'YYYY-MM-DD'), FLOOR(dbms_random.value(1, 100000)) +
1, FLOOR(dbms_random.value(1, 1000)) + 1, 'RT112');


INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (20, 30, 25, 'RT101');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (7, 10, 8, 'RT102');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (7, 10, 8, 'RT103');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (7, 10, 8, 'RT104');
```

```
INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (7, 10, 8, 'RT105');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (7, 10, 8, 'RT106');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (4, 10, 4, 'RT107');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (4, 10, 4, 'RT108');

INSERT INTO ZOO_ADMISSION (Child_Price, Adult_Price, Senior_Price, RevenueTypes_ID)
VALUES (4, 10, 4, 'RT111');


INSERT INTO CONCESSION (Product, RevenueTypes_ID) VALUES (15, 'RT109');

INSERT INTO CONCESSION (Product, RevenueTypes_ID) VALUES (15, 'RT110');


INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S001');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S002');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S003');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S004');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S005');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S006');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S007');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S008');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S009');
```

```
INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S010');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S011');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S012');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S013');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S014');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S015');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S016');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S017');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S018');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S019');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (890456132, 'S020');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S021');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S022');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S023');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S024');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S025');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S026');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S027');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S028');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S029');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S030');
```

```
INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S031');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S032');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S033');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S034');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S035');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S036');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S037');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S038');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S039');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (109876543, 'S040');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S041');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S042');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S043');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S044');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S045');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S046');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S047');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S048');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S049');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S050');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S051');
```

```
INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S052');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S053');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S054');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S055');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S056');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S057');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S058');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S059');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (543210987, 'S060');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S061');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S062');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S063');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S064');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S065');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S066');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S067');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S068');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S069');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S070');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S071');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S072');
```

```
INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S073');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S074');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S075');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S076');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S077');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S078');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S079');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (876543219, 'S080');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S081');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S082');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S083');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S084');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S085');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S086');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S087');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S088');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S089');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S090');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S091');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S092');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S093');
```

```
INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S094');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S095');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S096');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S097');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S098');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S099');

INSERT INTO CARES_FOR (Emp_ID, Spc_ID) VALUES (234567890, 'S100');


INSERT INTO ANIMAL_SHOW
(Child_Price,Adult_Price,Senior_Price,Shows_Per_Day,RevenueTypes_ID) VALUES
(10,20,15,3,'RT112');


INSERT INTO PARTICIPATES_IN (Spc_ID,RevenueTypes_ID,Num_spc_req) VALUES
('S037','RT112',2);

INSERT INTO PARTICIPATES_IN (Spc_ID,RevenueTypes_ID,Num_spc_req) VALUES
('S076','RT112',1);

INSERT INTO PARTICIPATES_IN (Spc_ID,RevenueTypes_ID,Num_spc_req) VALUES
('S092','RT112',2);

INSERT INTO PARTICIPATES_IN (Spc_ID,RevenueTypes_ID,Num_spc_req) VALUES
('S016','RT112',2);

INSERT INTO PARTICIPATES_IN (Spc_ID,RevenueTypes_ID,Num_spc_req) VALUES
('S036','RT112',2);
```

13. Implementation and Challenge Faced

- Project phase poses challenges in establishing seamless connections between the database and Tkinter framework.
- Integration of Asset Management, Daily Zoo Activity, and Management Reporting functionalities requires overcoming hurdles in implementing complex SQL commands within Tkinter.
- Linking database operations with the user interface adds complexity, demanding a focused effort to master intricate SQL commands for effective integration into Tkinter-based application.
- Learning curve associated with these technical aspects emphasizes the need for a strategic approach for successful project implementation.
- Challenges encountered in navigating Tkinter library nuances, requiring a thorough understanding of its functionalities for seamless database incorporation.
- Complexity extends to mastering intricate SQL commands tailored for Tkinter, posing a learning curve for the development team.
- Overcoming challenges demands a meticulous approach to address technical intricacies of Tkinter library and nuances of SQL commands for robust integration with Tkinter-based application.

14. User Guide

There are few steps in our Application and they are :

1. Introduction:

Welcome to the Zoo Management System! This application is designed to help zookeepers and administrators efficiently manage and track various aspects of the zoo, including animals, buildings, attractions, employees, and hourly wages.

2. Main Tabs:

The application consists of three main tabs:

A. Asset Management

Allows you to manage assets such as animals, buildings, attractions, employees, and hourly wages. Here you can Add, update, view these assets. There are **5 subtabs.** When you click the asset subtab, A table of the asset will display infront of you.

- When you select a row in the table, its value will automatically fit into the place holder of the fields.
- For adding that asset, you need to enter the value of the attributes and then click on the add button to successfully add the asset.
- For updating the asset you need to again enter the asset details which you want to update. After entering, click the update button to successfully update the asset.

B. Daily Zoo Activity

Provides features for tracking daily activities within the zoo. There are **3 subtabs.**

a. Attraction

Here the user can view the details of revenue of the particular attraction, also the total number of tickets with the date sold. To add the Attraction tickets, enter the date, type of attraction, type of ticket(child, Senior, and Adult).

b. Concession

Here the user can view the details of revenue, product location, tickets sold on a particular date. To add the Concession tickets, enter the date, type of Product.

c. Attendance.

Here the user can view the details of total revenue, attendance on a particular date for each activity. To add the numbers of attendee and revenue for each type of activity enter date, Type of attraction, and type of ticket(child, Senior, and Adult).

C. Management and Reporting

Enables management and reporting functionalities for better zoo administration. Further it has **3 subtabs:**

- **Total revenue**

User needs to enter the date for which he wants the revenue for all the different revenue sources with the details of no of tickets sold, revenue source name and total revenue for that source.

- **Animal report**

To retrieve the details of animal population by species, including totals by status, total monthly food cost and costs for assigned veterinarians and animal care specialists assuming a 40 hour work week. For this you need to click Get Total Report.

- **Revenue by time period**

This includes **3 subtops:**

a. Top 3 attractions

User need to enter start and end date to retrieve the top 3 attractions (in terms of total revenue) in descending order.

b. Best 5 days for month

User needs to enter the month to compute the 5 best days (in terms of total revenue.

c. Average Revenue

User need to enter start and end date to compute the average revenue for each attraction, concession, and total attendance.

Here are some screenshot of application usage.

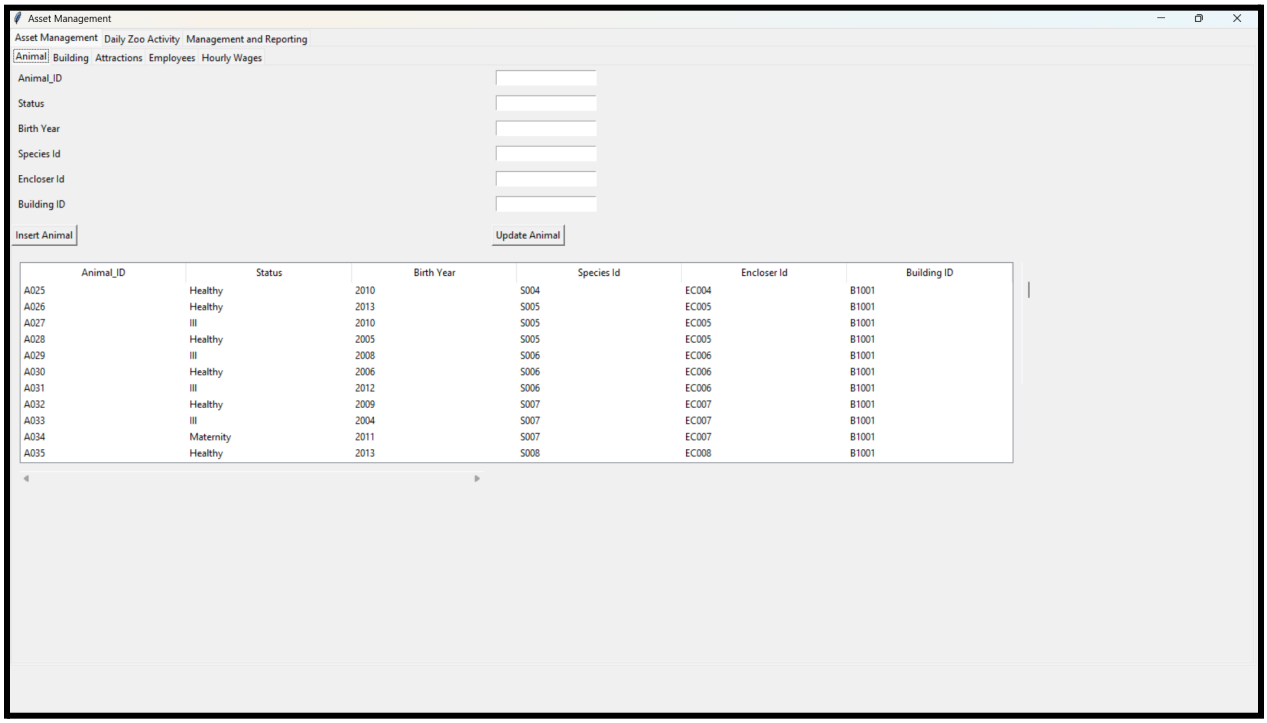


Fig-1: Adding and updating Animal details

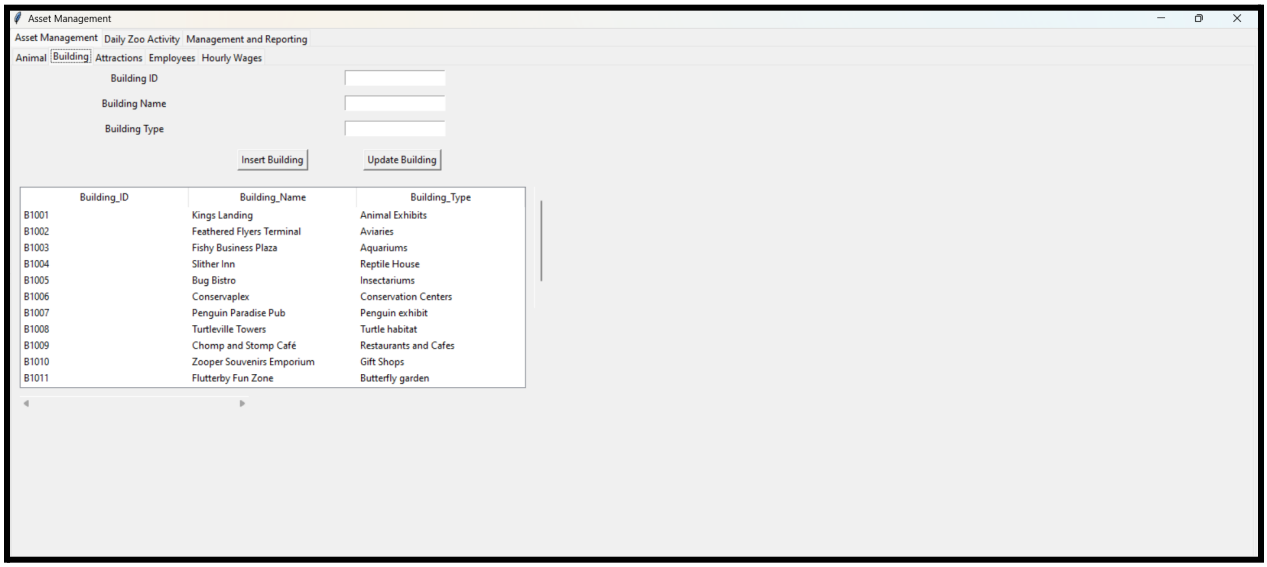


Fig-2: Adding and updating Building details

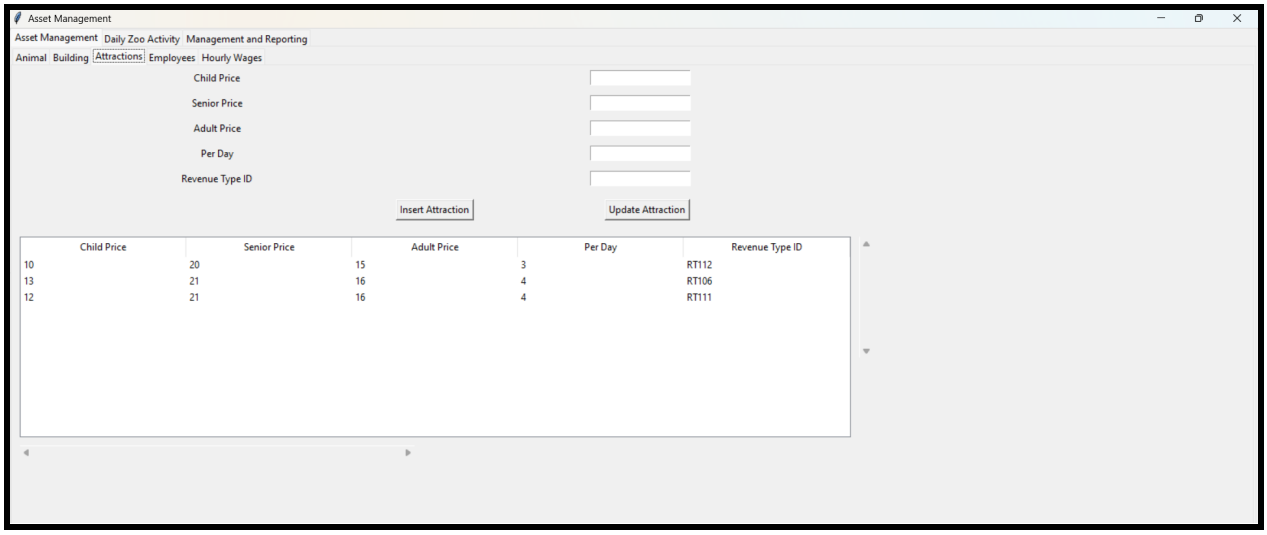


Fig-3: Adding and updating Attraction details

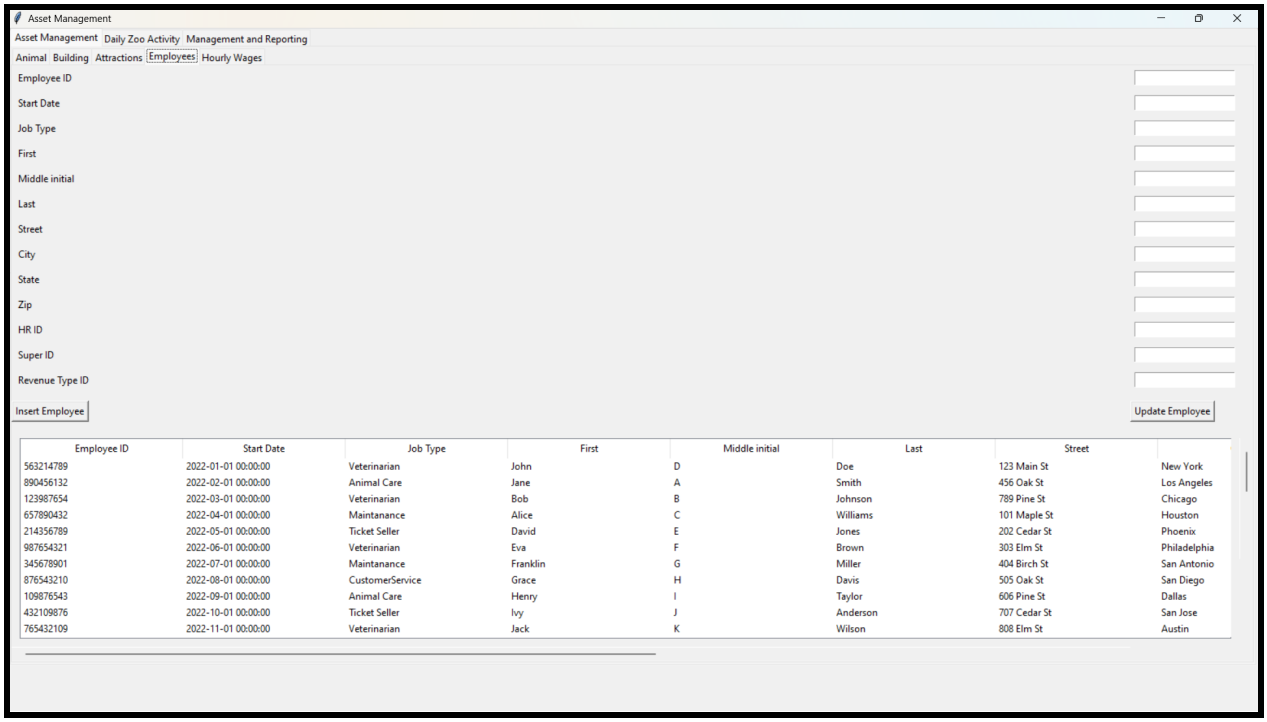


Fig-4: Adding and updating Employee details

Fig-5: Adding and updating Hourly wages details

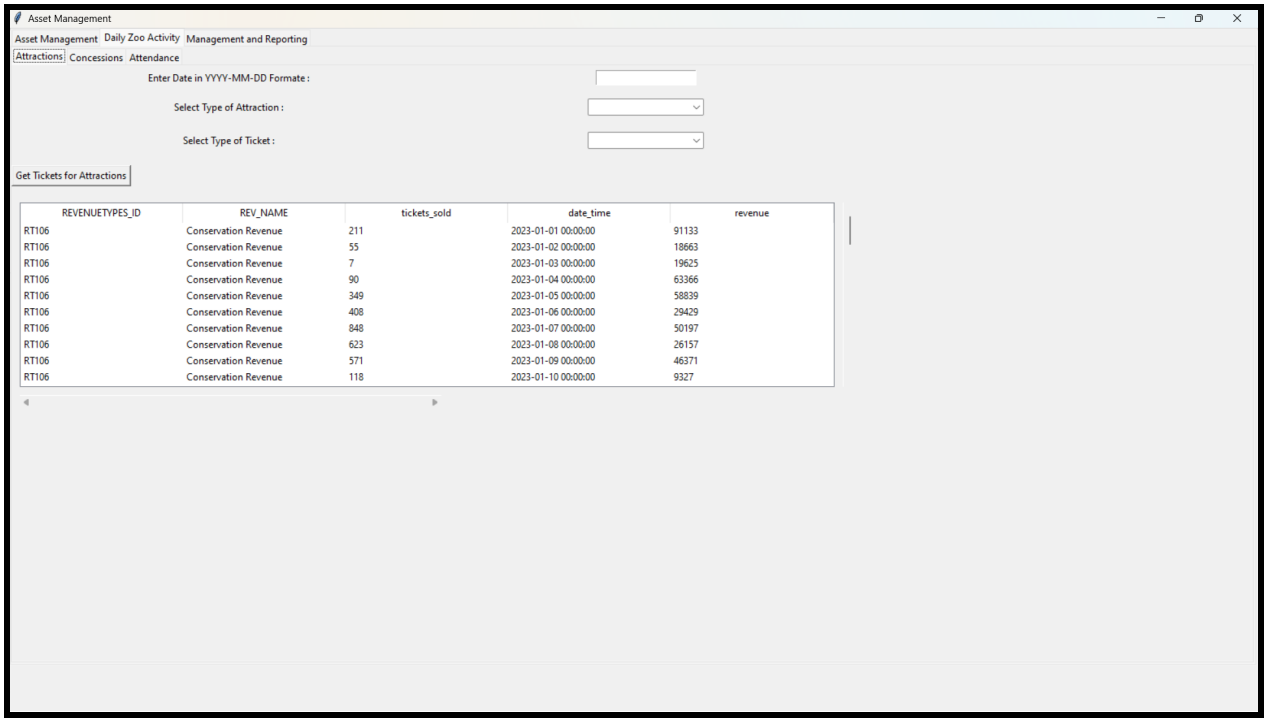


Fig-6: Adding Attraction Ticket and view Attraction Details

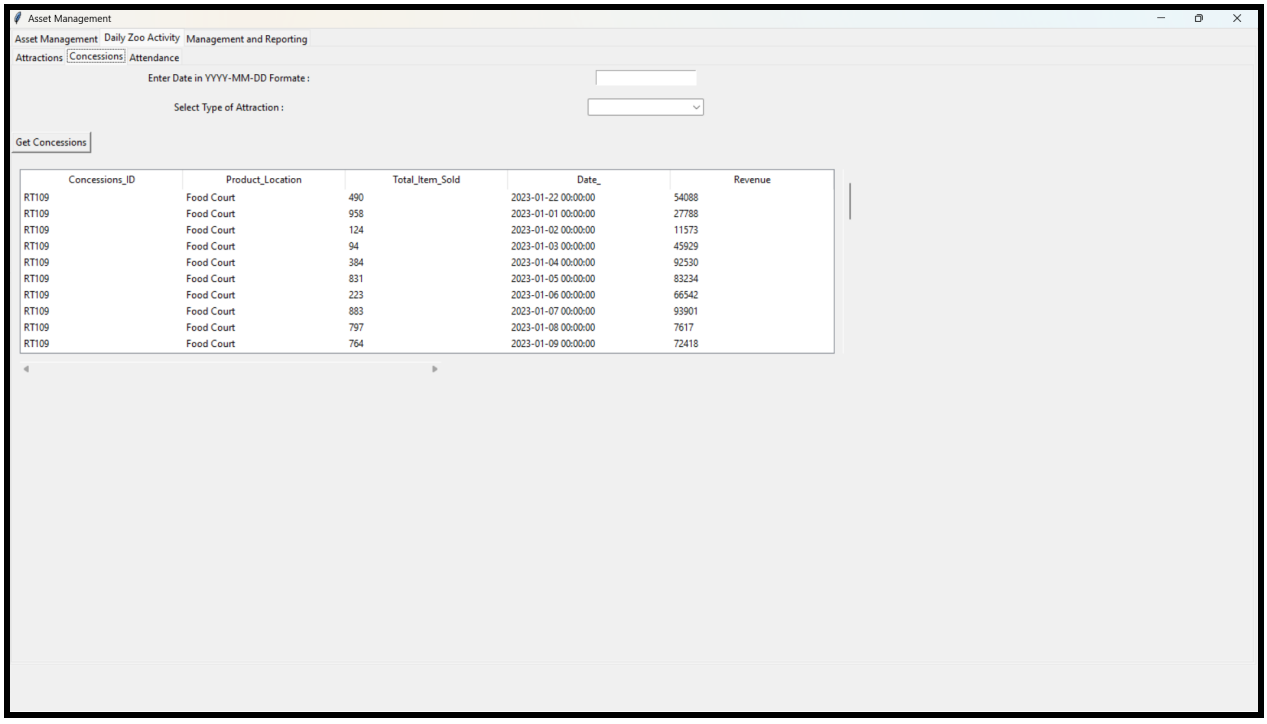


Fig-7: Adding Concession and view Concession Details

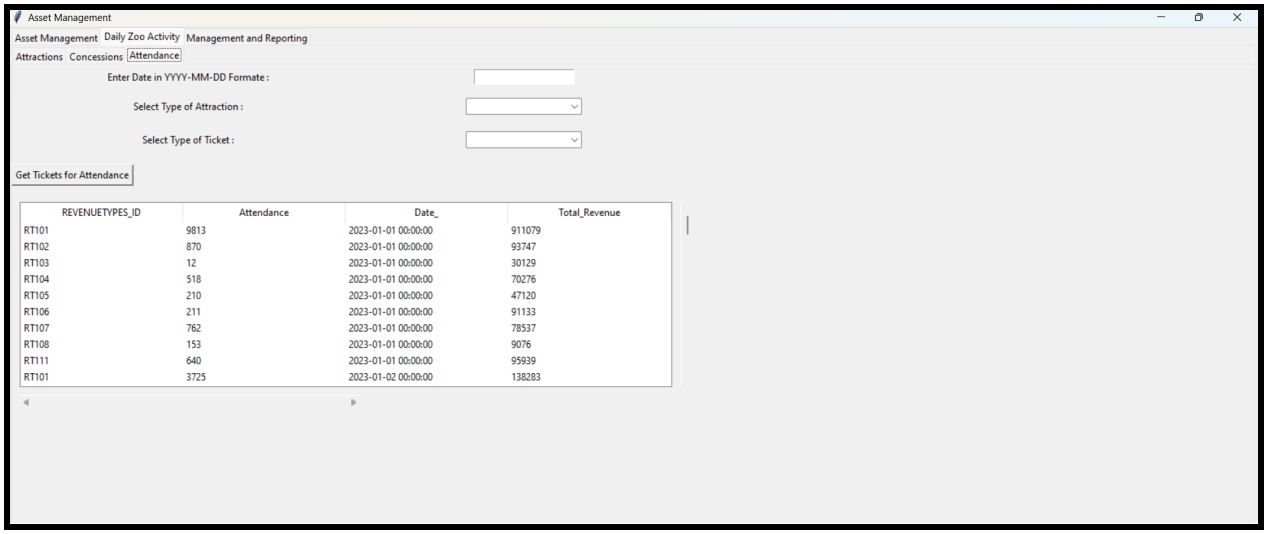


Fig-8: Adding Attendee ticket and Attendance Details

Fig-9: Get Total Revenue details for particular date

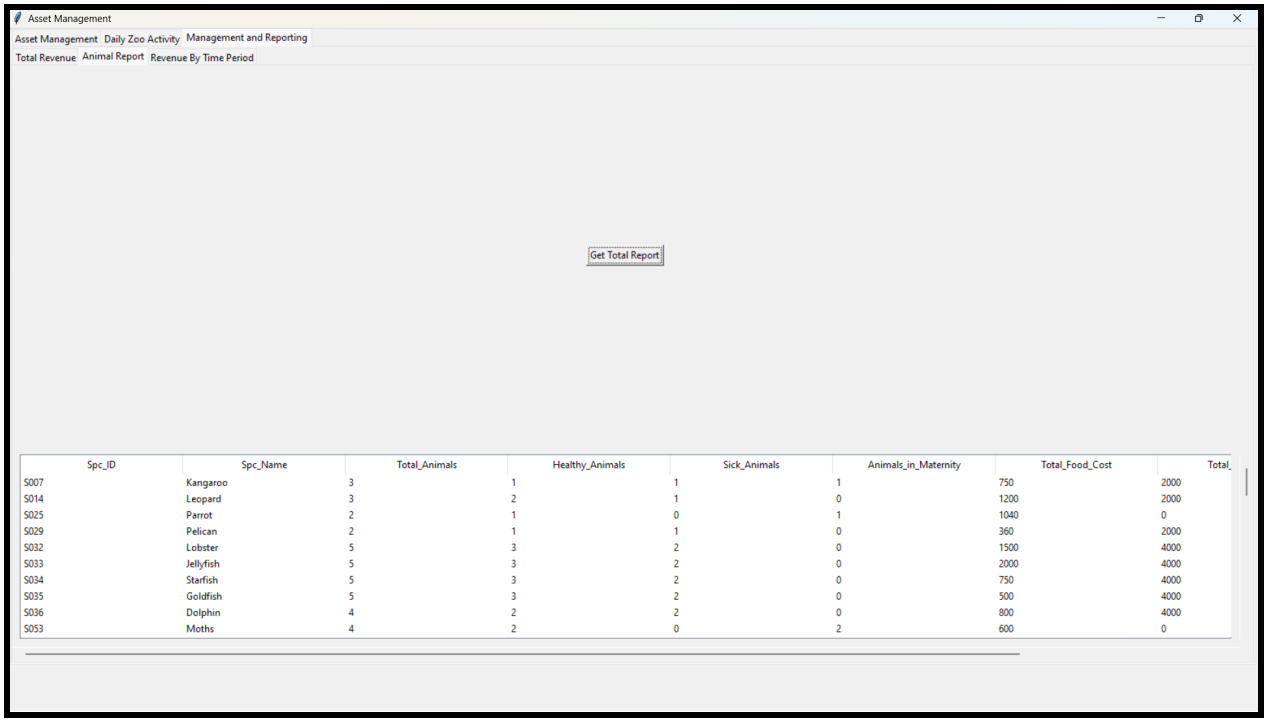


Fig-10: Get Animal Report details

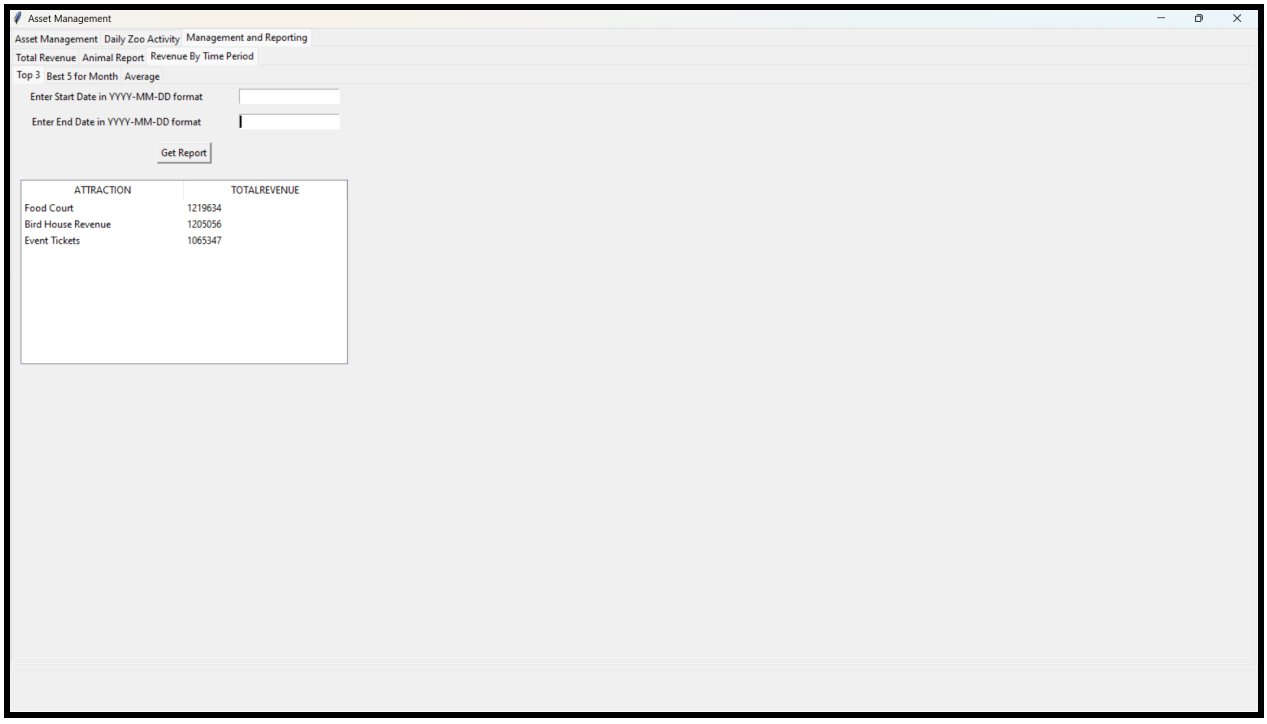


Fig-11: Get Top 3 Attraction for a particular period

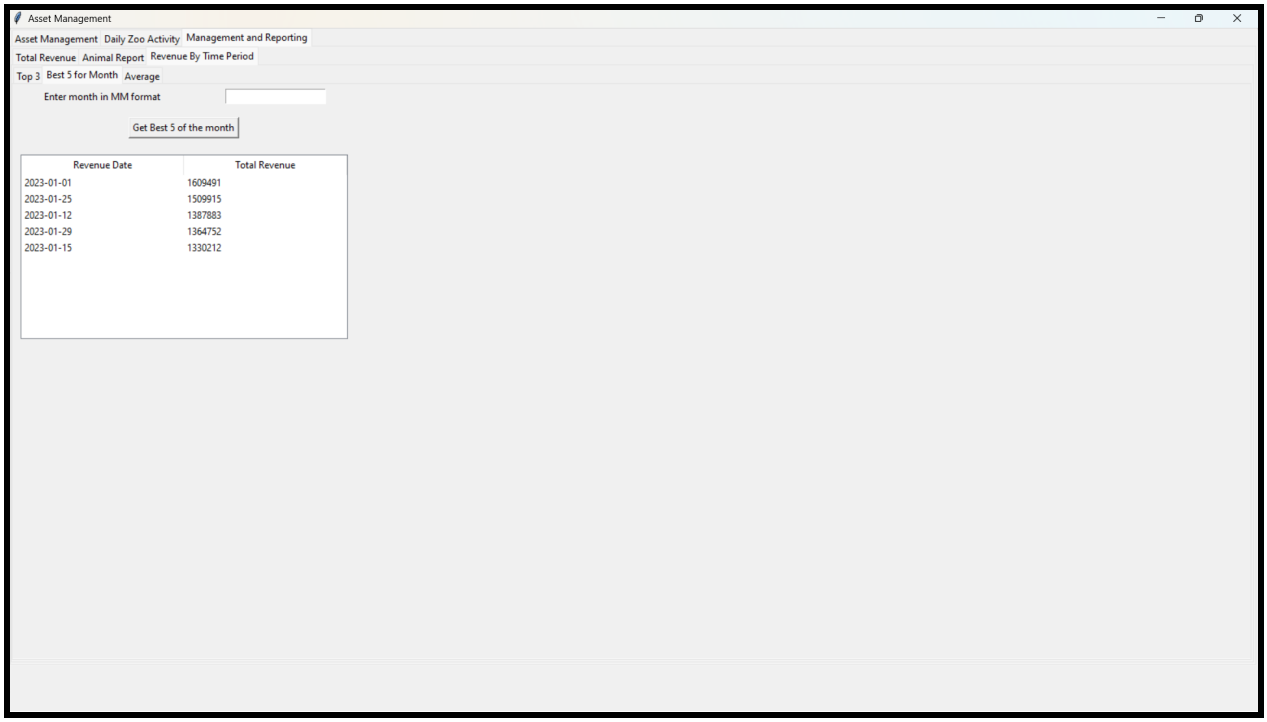


Fig-12: Get Best 5 Days Revenue for a particular Month

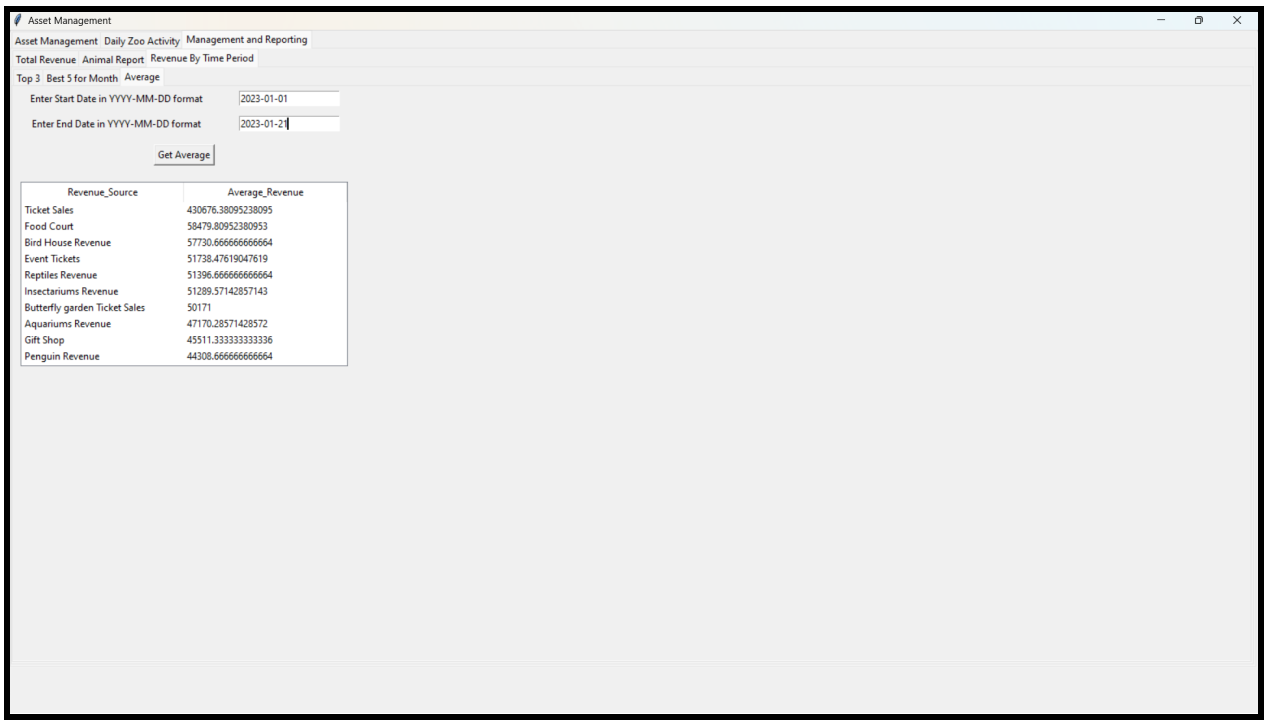


Fig-13: Get Average Revenue for a particular period

15. Source Code

```
import tkinter as tk

from tkinter import ttk

from tkcalendar import Calendar

import cx_Oracle

import datetime


#---------------------------------------Connection-------------------
---------------------------------


username = 'hp578'

password = '11_Oraclesql'

host = 'prophet.njit.edu'

port = 1521

sid = 'course'

dsn = cx_Oracle.makedsn(host, port, sid)

try:

    connection = cx_Oracle.connect(username,password,dsn)

    print("Connection successful!")

    # Add your code to work with the database here

except cx_Oracle.DatabaseError as e:

    print(f"Error connecting to the Oracle database: {e}")


#---------------------------------------Connection-----------------
-----------------------------------
```

```
class zoo:

    def __init__(self, root):

        self.root = root

        self.root.title("Asset Management")


        # ------------------------------------------------Creating
main tabs-----------------------------------------


        self.tab_control = ttk.Notebook(root)

        self.asset_management = ttk.Frame(self.tab_control)

        self.daily_zoo_activity = ttk.Frame(self.tab_control)

        self.management_reporting = ttk.Frame(self.tab_control)


        #------------------------------------------------------ Adding
main tabs ----------------------------------------------


        self.tab_control.add(self.asset_management, text='Asset
Management')

        self.tab_control.add(self.daily_zoo_activity, text='Daily Zoo
Activity')

        self.tab_control.add(self.management_reporting,
text='Management and Reporting')


        self.tab_control.pack(expand=1, fill='both')


        #------------------------------------------------- Adding
sub-tabs under Tab Asset-----------------------------------------
```

```
        self.sub_tab_control_asset =
ttk.Notebook(self.asset_management)

        self.sub_tab_animal = ttk.Frame(self.sub_tab_control_asset)

        self.sub_tab_building = ttk.Frame(self.sub_tab_control_asset)

        self.sub_tab_attraction =
ttk.Frame(self.sub_tab_control_asset)

        self.sub_tab_employees =
ttk.Frame(self.sub_tab_control_asset)

        self.sub_tab_hourly = ttk.Frame(self.sub_tab_control_asset)




        self.sub_tab_control_asset.add(self.sub_tab_animal,
text='Animal')

        self.sub_tab_control_asset.add(self.sub_tab_building,
text='Building')

        self.sub_tab_control_asset.add(self.sub_tab_attraction,
text='Attractions')

        self.sub_tab_control_asset.add(self.sub_tab_employees,
text='Employees')

        self.sub_tab_control_asset.add(self.sub_tab_hourly,
text='Hourly Wages')


        self.sub_tab_control_asset.pack(expand=1, fill='both')


        #------------------------------------------------- Adding
sub-tabs under Tab Animal ----------------------------------------
```

```
        self.view_animal_id_label = tk.Label(self.sub_tab_animal,
text='Animal_ID')

        self.view_animal_id_entry = tk.Entry(self.sub_tab_animal)

        self.view_animal_id_label.grid(row=0, column=0, padx=5,
pady=5, sticky="w")

        self.view_animal_id_entry.grid(row=0, column=1, padx=5,
pady=5, sticky="w")


        self.view_animal_status_label = tk.Label(self.sub_tab_animal,
text='Status')

        self.view_animal_status_entry = tk.Entry(self.sub_tab_animal)

        self.view_animal_status_label.grid(row=1, column=0, padx=5,
pady=5, sticky="w")

        self.view_animal_status_entry.grid(row=1, column=1, padx=5,
pady=5, sticky="w")


        self.view_animal_birth_label = tk.Label(self.sub_tab_animal,
text='Birth Year')

        self.view_animal_birth_entry = tk.Entry(self.sub_tab_animal)

        self.view_animal_birth_label.grid(row=2, column=0, padx=5,
pady=5, sticky="w")

        self.view_animal_birth_entry.grid(row=2, column=1, padx=5,
pady=5, sticky="w")


        self.view_animal_spc_id_label = tk.Label(self.sub_tab_animal,
text='Species Id')

        self.view_animal_spc_id_entry = tk.Entry(self.sub_tab_animal)

        self.view_animal_spc_id_label.grid(row=3, column=0, padx=5,
pady=5, sticky="w")
```

```
        self.view_animal_spc_id_entry.grid(row=3, column=1, padx=5,
pady=5, sticky="w")


        self.view_animal_enc_id_label = tk.Label(self.sub_tab_animal,
text='Encloser Id')

        self.view_animal_enc_id_entry = tk.Entry(self.sub_tab_animal)

        self.view_animal_enc_id_label.grid(row=4, column=0, padx=5,
pady=5, sticky="w")

        self.view_animal_enc_id_entry.grid(row=4, column=1, padx=5,
pady=5, sticky="w")


        self.view_animal_build_id_label =
tk.Label(self.sub_tab_animal, text='Building ID')

        self.view_animal_build_id_entry =
tk.Entry(self.sub_tab_animal)

        self.view_animal_build_id_label.grid(row=5, column=0, padx=5,
pady=5, sticky="w")

        self.view_animal_build_id_entry.grid(row=5, column=1, padx=5,
pady=5, sticky="w")


        self.insert_animal_button = tk.Button(self.sub_tab_animal,
text = "Insert Animal", command=self.add_animal)

        self.insert_animal_button.grid(row=6, column=0, columnspan=2,
pady=10, sticky="w")


        self.upd_animal_button = tk.Button(self.sub_tab_animal, text
= "Update Animal", command=self.update_animal)

        self.upd_animal_button.grid(row=6, column=1, columnspan=2,
pady=10, sticky="w")
```

```
        self.message_label = tk.Label(root, text="")

        self.message_label.pack()


        columns = ("Animal_ID", "Status", "Birth Year", "Species Id",
"Encloser Id", "Building ID")

        tree = ttk.Treeview(self.sub_tab_animal, columns=columns,
show="headings")


        # Set column headings

        for col in columns:

            tree.heading(col, text=col)


        # Grid the Treeview

        tree.grid(row=7, column=0, pady=10, padx=10, sticky="nsew" ,
rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb = ttk.Scrollbar(self.sub_tab_animal, orient="vertical",
command=tree.yview)

        vsb.grid(row=7, column=2, pady=10, sticky="ns")

        tree.configure(yscrollcommand=vsb.set)


        cursor2 = connection.cursor()

        query = "SELECT * FROM HP578.ANIMAL"

        cursor2.execute(query)
```

```
        # Fetch all rows from the result
        rows = cursor2.fetchall()
        # Insert data into the Treeview
        for row in rows:
            tree.insert("", "end", values=row)

        def on_treeview_select(event):
    # Get the selected item
            selected_item = tree.selection()

            # Check if any item is selected
            if selected_item:
                # Clear the entries
                self.view_animal_id_entry.delete(0, 'end')
                self.view_animal_status_entry.delete(0, 'end')
                self.view_animal_birth_entry.delete(0, 'end')
                self.view_animal_spc_id_entry.delete(0, 'end')
                self.view_animal_enc_id_entry.delete(0, 'end')
                self.view_animal_build_id_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree.item(selected_item)['values']
                if values:
                    self.view_animal_id_entry.insert(0, values[0])  #
Assuming name is the first column
```

```
                    self.view_animal_status_entry.insert(0,
values[1])

                    self.view_animal_birth_entry.insert(0, values[2])

                    self.view_animal_spc_id_entry.insert(0,
values[3])

                    self.view_animal_enc_id_entry.insert(0,
values[4])

                    self.view_animal_build_id_entry.insert(0,
values[5])

        tree.bind('<ButtonRelease-1>', on_treeview_select)


        #------------------------------------------------- Adding
sub-tabs under Tab Building
-----------------------------------------


        self.view_building_id_label = tk.Label(self.sub_tab_building,
text='Building ID')

        self.view_building_id_entry = tk.Entry(self.sub_tab_building)

        self.view_building_id_label.grid(row=0, column=0, padx=5,
pady=5)

        self.view_building_id_entry.grid(row=0, column=1, padx=5,
pady=5)



        self.view_building_name_label =
tk.Label(self.sub_tab_building, text='Building Name')

        self.view_building_name_entry =
tk.Entry(self.sub_tab_building)

        self.view_building_name_label.grid(row=1, column=0, padx=5,
pady=5)
```

```
        self.view_building_name_entry.grid(row=1, column=1, padx=5,
pady=5)


        self.view_building_type_label =
tk.Label(self.sub_tab_building, text='Building Type')

        self.view_building_type_entry =
tk.Entry(self.sub_tab_building)

        self.view_building_type_label.grid(row=2, column=0, padx=5,
pady=5)

        self.view_building_type_entry.grid(row=2, column=1, padx=5,
pady=5)


        self.view_building_button = tk.Button(self.sub_tab_building,
text='Insert Building', command=self.add_building)

        self.view_building_button.grid(row=3, column=0, columnspan=2,
pady=10)


        self.view_building_button = tk.Button(self.sub_tab_building,
text='Update Building', command=self.upd_building)

        self.view_building_button.grid(row=3, column=1, columnspan=2,
pady=10)


        self.message_label = tk.Label(root, text="")

        self.message_label.pack()


        columns_build = ("Building_ID", "Building_Name",
"Building_Type")

        tree_build = ttk.Treeview(self.sub_tab_building,
columns=columns_build, show="headings")
```

```
        # Set column headings
        for col in columns_build:
            tree_build.heading(col, text=col)

        # Grid the Treeview
        tree_build.grid(row=4, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )

        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="vertical", command=tree_build.yview)
        vsb_build.grid(row=4, column=2, pady=10, sticky="ns")
        tree_build.configure(yscrollcommand=vsb_build.set)

        hsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="horizontal", command=tree_build.xview)
        hsb_build.grid(row=6, column=0, padx=10, sticky="ew")
        tree_build.configure(xscrollcommand=hsb_build.set)
        cursor3 = connection.cursor()
        query1 = "SELECT * FROM HP578.BUILDING"
        cursor3.execute(query1)

        # Fetch all rows from the result
        rows_build = cursor3.fetchall()
        # Insert data into the Treeview
```

```
        for row in rows_build:
            tree_build.insert("", "end", values=row)

        def on_treeview_select_b(event):
    # Get the selected item
            selected_item_build = tree_build.selection()

            # Check if any item is selected
            if selected_item_build:
                # Clear the entries
                self.view_building_id_entry.delete(0, 'end')
                self.view_building_name_entry.delete(0, 'end')
                self.view_building_type_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values =
tree_build.item(selected_item_build)['values']
                if values:
                    self.view_building_id_entry.insert(0, values[0])
# Assuming name is the first column
                    self.view_building_name_entry.insert(0,
values[1])
                    self.view_building_type_entry.insert(0,
values[2])
        tree_build.bind('<ButtonRelease-1>', on_treeview_select_b)
```

```
        #------------------------------------------------ Adding
sub-tabs under Tab Attraction
-----------------------------------------


        self.view_attraction_c_p_label =
tk.Label(self.sub_tab_attraction, text='Child Price')

        self.view_attraction_c_p_entry =
tk.Entry(self.sub_tab_attraction)

        self.view_attraction_c_p_label.grid(row=0, column=0, padx=5,
pady=5)

        self.view_attraction_c_p_entry.grid(row=0, column=1, padx=5,
pady=5)


        self.view_attraction_s_p_label =
tk.Label(self.sub_tab_attraction, text='Senior Price')

        self.view_attraction_s_p_entry =
tk.Entry(self.sub_tab_attraction)

        self.view_attraction_s_p_label.grid(row=1, column=0, padx=5,
pady=5)

        self.view_attraction_s_p_entry.grid(row=1, column=1, padx=5,
pady=5)


        self.view_attraction_a_p_label =
tk.Label(self.sub_tab_attraction, text='Adult Price')

        self.view_attraction_a_p_entry =
tk.Entry(self.sub_tab_attraction)

        self.view_attraction_a_p_label.grid(row=2, column=0, padx=5,
pady=5)

        self.view_attraction_a_p_entry.grid(row=2, column=1, padx=5,
pady=5)
```

```
        self.view_attraction_per_day_label =
tk.Label(self.sub_tab_attraction, text='Per Day')

        self.view_attraction_per_day_entry =
tk.Entry(self.sub_tab_attraction)

        self.view_attraction_per_day_label.grid(row=3, column=0,
padx=5, pady=5)

        self.view_attraction_per_day_entry.grid(row=3, column=1,
padx=5, pady=5)



        self.view_attraction_rev_id_label =
tk.Label(self.sub_tab_attraction, text='Revenue Type ID')

        self.view_attraction_rev_id_entry =
tk.Entry(self.sub_tab_attraction)

        self.view_attraction_rev_id_label.grid(row=4, column=0,
padx=5, pady=5)

        self.view_attraction_rev_id_entry.grid(row=4, column=1,
padx=5, pady=5)



        self.view_attraction_button =
tk.Button(self.sub_tab_attraction, text='Insert Attraction',
command=self.add_attraction)

        self.view_attraction_button.grid(row=5, column=0,
columnspan=2, pady=10)



        self.view_attraction_button =
tk.Button(self.sub_tab_attraction, text='Update Attraction',
command=self.upd_attraction)

        self.view_attraction_button.grid(row=5, column=1,
columnspan=2, pady=10)



        self.message_label = tk.Label(root, text="")
```

```
        self.message_label.pack()


        columns_att = ("Child Price", "Senior Price", "Adult Price",
"Per Day", "Revenue Type ID")

        tree_att = ttk.Treeview(self.sub_tab_attraction,
columns=columns_att, show="headings")


        # Set column headings

        for col in columns_att:

            tree_att.heading(col, text=col)


        # Grid the Treeview

        tree_att.grid(row=6, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="vertical", command=tree_att.yview)

        vsb_att.grid(row=6, column=2, pady=10, sticky="ns")

        tree_att.configure(yscrollcommand=vsb_att.set)


        hsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="horizontal", command=tree_att.xview)

        hsb_att.grid(row=8, column=0, padx=10, sticky="ew")

        tree_att.configure(xscrollcommand=hsb_att.set)

        cursor4 = connection.cursor()
```

```
        query2 = "SELECT * FROM HP578.ANIMAL_SHOW"
        cursor4.execute(query2)

        # Fetch all rows from the result
        rows_att = cursor4.fetchall()
        # Insert data into the Treeview
        for row in rows_att:
            tree_att.insert("", "end", values=row)

        def on_treeview_select_b(event):
    # Get the selected item
            selected_item_att = tree_att.selection()

            # Check if any item is selected
            if selected_item_att:
                # Clear the entries
                self.view_attraction_rev_id_entry.delete(0, 'end')
                self.view_attraction_c_p_entry.delete(0, 'end')
                self.view_attraction_s_p_entry.delete(0, 'end')
                self.view_attraction_a_p_entry.delete(0, 'end')
                self.view_attraction_per_day_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_att.item(selected_item_att)['values']
```

```
                if values:
                    self.view_attraction_rev_id_entry.insert(0,
values[4])  # Assuming name is the first column
                    self.view_attraction_c_p_entry.insert(0,
values[0])
                    self.view_attraction_s_p_entry.insert(0,
values[1])
                    self.view_attraction_a_p_entry.insert(0,
values[2])
                    self.view_attraction_per_day_entry.insert(0,
values[3])
        tree_att.bind('<ButtonRelease-1>', on_treeview_select_b)


        #------------------------------------------------- Adding
sub-tabs under Tab Employee
-----------------------------------------


        self.view_employee_id_label =
tk.Label(self.sub_tab_employees, text='Employee ID')
        self.view_employee_id_entry =
tk.Entry(self.sub_tab_employees)
        self.view_employee_id_label.grid(row=0, column=0, padx=5,
pady=5,sticky='w' )
        self.view_employee_id_entry.grid(row=0, column=1, padx=5,
pady=5,sticky='w')


        self.view_employee_start_label =
tk.Label(self.sub_tab_employees, text='Start Date')
        self.view_employee_start_entry =
tk.Entry(self.sub_tab_employees)
```

```
        self.view_employee_start_label.grid(row=1, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_start_entry.grid(row=1, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_job_label =
tk.Label(self.sub_tab_employees, text='Job Type')

        self.view_employee_job_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_job_label.grid(row=2, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_job_entry.grid(row=2, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_f_label = tk.Label(self.sub_tab_employees,
text='First')

        self.view_employee_f_entry = tk.Entry(self.sub_tab_employees)

        self.view_employee_f_label.grid(row=3, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_f_entry.grid(row=3, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_m_label = tk.Label(self.sub_tab_employees,
text='Middle initial')

        self.view_employee_m_entry = tk.Entry(self.sub_tab_employees)

        self.view_employee_m_label.grid(row=4, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_m_entry.grid(row=4, column=1, padx=5,
pady=5,sticky='w')
```

```
        self.view_employee_l_label = tk.Label(self.sub_tab_employees,
text='Last')

        self.view_employee_l_entry = tk.Entry(self.sub_tab_employees)

        self.view_employee_l_label.grid(row=5, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_l_entry.grid(row=5, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_st_label =
tk.Label(self.sub_tab_employees, text='Street')

        self.view_employee_st_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_st_label.grid(row=6, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_st_entry.grid(row=6, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_city_label =
tk.Label(self.sub_tab_employees, text='City')

        self.view_employee_city_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_city_label.grid(row=7, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_city_entry.grid(row=7, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_state_label =
tk.Label(self.sub_tab_employees, text='State')

        self.view_employee_state_entry =
tk.Entry(self.sub_tab_employees)
```

```
        self.view_employee_state_label.grid(row=8, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_state_entry.grid(row=8, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_zip_label =
tk.Label(self.sub_tab_employees, text='Zip')

        self.view_employee_zip_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_zip_label.grid(row=9, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_zip_entry.grid(row=9, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_hr_id_label =
tk.Label(self.sub_tab_employees, text='HR ID')

        self.view_employee_hr_id_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_hr_id_label.grid(row=10, column=0, padx=5,
pady=5,sticky='w')

        self.view_employee_hr_id_entry.grid(row=10, column=1, padx=5,
pady=5,sticky='w')



        self.view_employee_sup_id_label =
tk.Label(self.sub_tab_employees, text='Super ID')

        self.view_employee_sup_id_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_sup_id_label.grid(row=11, column=0,
padx=5, pady=5,sticky='w')

        self.view_employee_sup_id_entry.grid(row=11, column=1,
padx=5, pady=5,sticky='w')
```

```
        self.view_employee_rev_id_label =
tk.Label(self.sub_tab_employees, text='Revenue Type ID')

        self.view_employee_rev_id_entry =
tk.Entry(self.sub_tab_employees)

        self.view_employee_rev_id_label.grid(row=12, column=0,
padx=5, pady=5,sticky='w')

        self.view_employee_rev_id_entry.grid(row=12, column=1,
padx=5, pady=5,sticky='w')


        self.add_employee_button = tk.Button(self.sub_tab_employees,
text='Insert Employee', command=self.add_employee)

        self.add_employee_button.grid(row=13, column=0, columnspan=2,
pady=10 ,sticky='w')


        self.upd_employee_button = tk.Button(self.sub_tab_employees,
text='Update Employee', command=self.upd_employee)

        self.upd_employee_button.grid(row=13, column=1, columnspan=2,
pady=10,sticky='w')


        self.message_label = tk.Label(root, text="")

        self.message_label.pack()


        columns_em = ("Employee ID", "Start Date", "Job Type",
"First","Middle initial", "Last", "Street","City","State","Zip","HR
ID","Super ID","Revenue Type ID")

        tree_em = ttk.Treeview(self.sub_tab_employees,
columns=columns_em, show="headings")
```

```
        # Set column headings

        for col in columns_em:

            tree_em.heading(col, text=col)


        # Grid the Treeview

        tree_em.grid(row=14, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar


        def on_h(*args):

            tree_em.xview(*args)


        def on_v(*args):

            tree_em.yview(*args)


        vsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="vertical", command=on_v)

        vsb_em.grid(row=14, column=2, pady=10, sticky="ns")

        tree_em.configure(yscrollcommand=vsb_em.set)


        hsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="horizontal", command=on_h)

        tree_em.configure(xscrollcommand=hsb_em.set)

        hsb_em.grid(row=16, column=0, padx=0, sticky="ew")
```

```
        self.sub_tab_employees.grid_rowconfigure(0, weight=1)
        self.sub_tab_employees.grid_columnconfigure(0, weight=1)

        cursor6 = connection.cursor()
        query4 = "SELECT * FROM HP578.EMPLOYEE"
        cursor6.execute(query4)

        # Fetch all rows from the result
        rows = cursor6.fetchall()
        # Insert data into the Treeview
        for row in rows:
            tree_em.insert("", "end", values=row)

        def on_treeview_select_em(event):
    # Get the selected item
            selected_item = tree_em.selection()

            # Check if any item is selected
            if selected_item:
                # Clear the entries
                self.view_employee_id_entry.delete(0, 'end')
                self.view_employee_start_entry.delete(0, 'end')
                self.view_employee_job_entry.delete(0, 'end')
                self.view_employee_f_entry.delete(0, 'end')
                self.view_employee_m_entry.delete(0, 'end')
```

```
                self.view_employee_l_entry.delete(0, 'end')
                self.view_employee_st_entry.delete(0, 'end')
                self.view_employee_city_entry.delete(0, 'end')
                self.view_employee_state_entry.delete(0, 'end')
                self.view_employee_zip_entry.delete(0, 'end')
                self.view_employee_hr_id_entry.delete(0, 'end')
                self.view_employee_sup_id_entry.delete(0, 'end')
                self.view_employee_rev_id_entry.delete(0, 'end')



                # Get values from the selected item and fill the
entries
                values = tree_em.item(selected_item)['values']
                if values:
                    self.view_employee_id_entry.insert(0, values[0])
                    self.view_employee_start_entry.insert(0,
values[1])
                    self.view_employee_job_entry.insert(0, values[2])
                    self.view_employee_f_entry.insert(0, values[3])
# Assuming name is the first column
                    self.view_employee_m_entry.insert(0, values[4])
                    self.view_employee_l_entry.insert(0, values[5])
                    self.view_employee_st_entry.insert(0, values[6])
                    self.view_employee_city_entry.insert(0,
values[7])
                    self.view_employee_state_entry.insert(0,
values[8])
```

```
                    self.view_employee_zip_entry.insert(0, values[9])
# Assuming name is the first column
                    self.view_employee_hr_id_entry.insert(0,
values[10])
                    self.view_employee_sup_id_entry.insert(0,
values[11])
                    self.view_employee_rev_id_entry.insert(0,
values[12])
        tree_em.bind('<ButtonRelease-1>', on_treeview_select_em)


        # #------------------------------------------------ Adding
sub-tabs under Tab Hourly ------------------------------------------


        self.view_hourly_id_label = tk.Label(self.sub_tab_hourly,
text='HR ID')
        self.view_hourly_id_entry = tk.Entry(self.sub_tab_hourly)
        self.view_hourly_id_label.grid(row=0, column=0, padx=5,
pady=5)
        self.view_hourly_id_entry.grid(row=0, column=1, padx=5,
pady=5)


        self.view_hourly_rate_label = tk.Label(self.sub_tab_hourly,
text='Rate')
        self.view_hourly_rate_entry = tk.Entry(self.sub_tab_hourly)
        self.view_hourly_rate_label.grid(row=1, column=0, padx=5,
pady=5)
        self.view_hourly_rate_entry.grid(row=1, column=1, padx=5,
pady=5)
```

```
        self.add_hourly_button = tk.Button(self.sub_tab_hourly,
text='Insert Hourly Wages', command=self.add_hourly)

        self.add_hourly_button.grid(row=2, column=0, columnspan=2,
pady=10, sticky='w')


        self.upd_hourly_button = tk.Button(self.sub_tab_hourly,
text='Update Hourly Wages', command=self.upd_hourly)

        self.upd_hourly_button.grid(row=2, column=1, columnspan=2,
pady=10)


        self.message_label = tk.Label(root, text="")

        self.message_label.pack()


        columns_hw = ("HR ID", "Rate")

        tree_hw = ttk.Treeview(self.sub_tab_hourly,
columns=columns_hw, show="headings")


        # Set column headings

        for col in columns_hw:

            tree_hw.heading(col, text=col)


        # Grid the Treeview

        tree_hw.grid(row=3, column=0, pady=10, padx=10, sticky="nsew"
, rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar
```

```
        vsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="vertical", command=tree_hw.yview)

        vsb_hw.grid(row=3, column=2, pady=10, sticky="ns")

        tree_hw.configure(yscrollcommand=vsb_hw.set)


        hsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="horizontal", command=tree_hw.xview)

        hsb_hw.grid(row=5, column=0, padx=10, sticky="ew")

        tree_hw.configure(xscrollcommand=hsb_hw.set)

        cursor5 = connection.cursor()

        query3 = "SELECT * FROM HP578.HOURLY_RATE"

        cursor5.execute(query3)


        # Fetch all rows from the result

        rows_hw = cursor5.fetchall()

        # Insert data into the Treeview

        for row in rows_hw:

            tree_hw.insert("", "end", values=row)


        def on_treeview_select_hw(event):

    # Get the selected item

            selected_item_hw = tree_hw.selection()


            # Check if any item is selected

            if selected_item_hw:

                # Clear the entries
```

```
                self.view_hourly_id_entry.delete(0, 'end')
                self.view_hourly_rate_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_hw.item(selected_item_hw)['values']
                if values:
                    self.view_hourly_id_entry.insert(0, values[0])  #
Assuming name is the first column
                    self.view_hourly_rate_entry.insert(0, values[1])
        tree_hw.bind('<ButtonRelease-1>', on_treeview_select_hw)

        #------------------------------------------------- Adding
sub-tabs under Tab Daily Zoo Activity
-----------------------------------------

        self.sub_tab_control_daily =
ttk.Notebook(self.daily_zoo_activity)
        self.sub_tab_daily_attractions =
ttk.Frame(self.sub_tab_control_daily)
        self.sub_tab_daily_concessions =
ttk.Frame(self.sub_tab_control_daily)
        self.sub_tab_daily_attendance =
ttk.Frame(self.sub_tab_control_daily)



self.sub_tab_control_daily.add(self.sub_tab_daily_attractions,
text='Attractions')
```

```
self.sub_tab_control_daily.add(self.sub_tab_daily_concessions,
text='Concessions')

        self.sub_tab_control_daily.add(self.sub_tab_daily_attendance,
text='Attendance')



        self.sub_tab_control_daily.pack(expand=1, fill='both')



        #------------------------------------------------- Adding
sub-tabs under Tab Daily Attractions
-----------------------------------------



        self.date_attractions_label =
tk.Label(self.sub_tab_daily_attractions, text='Enter Date in
YYYY-MM-DD Formate : ')

        self.date_attractions_entry =
tk.Entry(self.sub_tab_daily_attractions)

        self.date_attractions_label.grid(row=0, column=0, padx=5,
pady=5)

        self.date_attractions_entry.grid(row=0, column=1, padx=5,
pady=5)



        self.selected_option_var1 = tk.StringVar()

        options1=["Conservation Admission", "Butterfly garden
Admission", "Events and Shows"]

        self.name_attractions_label =
tk.Label(self.sub_tab_daily_attractions, text='Select Type of
Attraction : ')

        self.name_attractions_label.grid(row=1, column=0, padx=5,
pady=5)
```

```
        self.dropdown1 = ttk.Combobox(self.sub_tab_daily_attractions,
values=options1, textvariable=self.selected_option_var1)

        self.dropdown1.grid(row=1, column=1, padx=10, pady=10)



        self.selected_option_var2 = tk.StringVar()

        options=["Child", "Adult", "Senior"]

        self.type_attractions_label =
tk.Label(self.sub_tab_daily_attractions, text='Select Type of Ticket
: ')

        self.type_attractions_label.grid(row=2, column=0, padx=5,
pady=5)

        self.dropdown2 = ttk.Combobox(self.sub_tab_daily_attractions,
values=options, textvariable=self.selected_option_var2)

        self.dropdown2.grid(row=2, column=1, padx=10, pady=10)



        self.get_attractions_button =
tk.Button(self.sub_tab_daily_attractions, text='Get Tickets for
Attractions', command=self.attraction_report)

        self.get_attractions_button.grid(row=3, column=0,
columnspan=2, pady=10, sticky='w')



        columns_a = ("REVENUETYPES_ID", "REV_NAME", "tickets_sold",
"date_time", "revenue")

        tree_a = ttk.Treeview(self.sub_tab_daily_attractions,
columns=columns_a, show="headings")



        # Set column headings

        for col in columns_a:

            tree_a.heading(col, text=col)
```

```
        # Grid the Treeview
        tree_a.grid(row=4, column=0, columnspan=2,padx=10, pady=10)


        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_a = ttk.Scrollbar(self.sub_tab_daily_attractions,
orient="vertical", command=tree_a.yview)
        vsb_a.grid(row=4, column=2, pady=10, sticky="ns")
        tree_a.configure(yscrollcommand=vsb_a.set)


        hsb_a = ttk.Scrollbar(self.sub_tab_daily_attractions,
orient="horizontal", command=tree_a.xview)
        hsb_a.grid(row=6, column=0, padx=10, sticky="ew")
        tree_a.configure(xscrollcommand=hsb_a.set)
        cursor_a = connection.cursor()
        query_a = '''
        SELECT
            re.REVENUETYPES_ID AS Attraction_ID,
            rt.REV_NAME AS Attraction_Location,
            re.tickets_sold AS Ticket_Sold,
            re.date_time AS Date_Of_Attraction,
            re.revenue
        FROM
            HP578.revenue_events re
            JOIN
```

```
            HP578.revenue_types rt ON re.REVENUETYPES_ID =
rt.REVENUETYPES_ID

            WHERE

            re.REVENUETYPES_ID IN (SELECT DISTINCT REVENUETYPES_ID
FROM animal_show)
        '''

        cursor_a.execute(query_a)


        # Fetch all rows from the result

        rows_a = cursor_a.fetchall()

        # Insert data into the Treeview

        for row in rows_a:

            tree_a.insert("", "end", values=row)


        #------------------------------------------------- Adding
sub-tabs under Tab Daily Concessions
-----------------------------------------


        self.date_concession_label =
tk.Label(self.sub_tab_daily_concessions, text='Enter Date in
YYYY-MM-DD Formate : ')

        self.date_concession_entry =
tk.Entry(self.sub_tab_daily_concessions)

        self.date_concession_label.grid(row=0, column=0, padx=5,
pady=5)

        self.date_concession_entry.grid(row=0, column=1, padx=5,
pady=5)


        self.selected_option_var_con = tk.StringVar()
```

```
        options_con=["Food Court", "Gift Shop"]

        self.name_concession_label =
tk.Label(self.sub_tab_daily_concessions, text='Select Type of
Attraction : ')

        self.name_concession_label.grid(row=1, column=0, padx=5,
pady=5)

        self.dropdown_con =
ttk.Combobox(self.sub_tab_daily_concessions, values=options_con,
textvariable=self.selected_option_var_con)

        self.dropdown_con.grid(row=1, column=1, padx=10, pady=10)



        self.get_concession_button =
tk.Button(self.sub_tab_daily_concessions, text='Get Concessions',
command=self.concession_report)

        self.get_concession_button.grid(row=2, column=0,
columnspan=2, pady=10, sticky='w')



        columns_con = ("Concessions_ID", "Product_Location",
"Total_Item_Sold","Date_", "Revenue")

        tree_con = ttk.Treeview(self.sub_tab_daily_concessions,
columns=columns_con, show="headings")



        # Set column headings

        for col in columns_con:

            tree_con.heading(col, text=col)



        # Grid the Treeview

        tree_con.grid(row=3, column=0, columnspan=2,padx=10, pady=10)
```

```
        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_con = ttk.Scrollbar(self.sub_tab_daily_concessions,
orient="vertical", command=tree_con.yview)
        vsb_con.grid(row=3, column=2, pady=10, sticky="ns")
        tree_con.configure(yscrollcommand=vsb_con.set)


        hsb_con = ttk.Scrollbar(self.sub_tab_daily_concessions,
orient="horizontal", command=tree_con.xview)
        hsb_con.grid(row=5, column=0, padx=10, sticky="ew")
        tree_con.configure(xscrollcommand=hsb_con.set)
        cursor_con = connection.cursor()
        query_con = '''
        SELECT
            c.RevenueTypes_ID AS Concessions_ID,
            rt.rev_name AS Product_Location,
            re.tickets_sold AS Total_Item_Sold,
            re.date_time AS Date_,
            re.revenue AS Revenue
        FROM
            CONCESSION c
        JOIN
            revenue_events re on c.RevenueTypes_ID=re.RevenueTypes_ID
        JOIN
            revenue_types rt on rt.revenuetypes_id=re.revenuetypes_id
        '''
```

```
        cursor_con.execute(query_con)


        # Fetch all rows from the result

        rows_con = cursor_con.fetchall()

        # Insert data into the Treeview

        for row in rows_con:

            tree_con.insert("", "end", values=row)


        #------------------------------------------------- Adding
sub-tabs under Tab Daily Attendance
-----------------------------------------


        self.date_attendance_label =
tk.Label(self.sub_tab_daily_attendance, text='Enter Date in
YYYY-MM-DD Formate : ')

        self.date_attendance_entry =
tk.Entry(self.sub_tab_daily_attendance)

        self.date_attendance_label.grid(row=0, column=0, padx=5,
pady=5)

        self.date_attendance_entry.grid(row=0, column=1, padx=5,
pady=5)


        self.selected_option_var_atd_1 = tk.StringVar()

        options_atd_1=["General Admission", "Bird House Admission",
"Aquarium Admission", "Reptile House Admission", "Insectariums
Admission", "Conservation Admission", "Penguin Admission", "Turtle
habitat Admission", "Butterfly garden Admission"]

        self.name_attendance_label =
tk.Label(self.sub_tab_daily_attendance, text='Select Type of
Attraction : ')
```

```
        self.name_attendance_label.grid(row=1, column=0, padx=5,
pady=5)

        self.dropdown_atd =
ttk.Combobox(self.sub_tab_daily_attendance, values=options_atd_1,
textvariable=self.selected_option_var_atd_1)

        self.dropdown_atd.grid(row=1, column=1, padx=10, pady=10)



        self.selected_option_var_atd_2 = tk.StringVar()

        options_atd_2=["Child", "Adult", "Senior"]

        self.type_attendance_label =
tk.Label(self.sub_tab_daily_attendance, text='Select Type of Ticket :
')

        self.type_attendance_label.grid(row=2, column=0, padx=5,
pady=5)

        self.dropdown_atd_2 =
ttk.Combobox(self.sub_tab_daily_attendance, values=options_atd_2,
textvariable=self.selected_option_var_atd_2)

        self.dropdown_atd_2.grid(row=2, column=1, padx=10, pady=10)



        self.get_attendance_button =
tk.Button(self.sub_tab_daily_attendance, text='Get Tickets for
Attendance', command=self.attendance_report)

        self.get_attendance_button.grid(row=3, column=0,
columnspan=2, pady=10, sticky='w')



        columns_atd = ("REVENUETYPES_ID", "Attendance", "Date_",
"Total_Revenue")

        tree_atd = ttk.Treeview(self.sub_tab_daily_attendance,
columns=columns_atd, show="headings")
```

```
        # Set column headings
        for col in columns_atd:
            tree_atd.heading(col, text=col)


        # Grid the Treeview
        tree_atd.grid(row=4, column=0, columnspan=2,padx=10, pady=10)


        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_atd = ttk.Scrollbar(self.sub_tab_daily_attendance,
orient="vertical", command=tree_atd.yview)
        vsb_atd.grid(row=4, column=2, pady=10, sticky="ns")
        tree_atd.configure(yscrollcommand=vsb_atd.set)


        hsb_atd = ttk.Scrollbar(self.sub_tab_daily_attendance,
orient="horizontal", command=tree_atd.xview)
        hsb_atd.grid(row=6, column=0, padx=10, sticky="ew")
        tree_atd.configure(xscrollcommand=hsb_atd.set)
        cursor_atd = connection.cursor()
        query_atd = '''
        SELECT
            re.RevenueTypes_ID AS Revenue_ID,
            re.tickets_sold AS Attendance,
            re.date_time AS Date_,
            re.revenue AS Total_Revenue
        FROM
```

```
            HP578.zoo_admission za

        JOIN

            HP578.revenue_events re ON za.RevenueTypes_ID =
re.RevenueTypes_ID

        '''

        cursor_atd.execute(query_atd)


        # Fetch all rows from the result

        rows_atd = cursor_atd.fetchall()

        # Insert data into the Treeview

        for row in rows_atd:

            tree_atd.insert("", "end", values=row)


        #------------------------------------------------- Adding
sub-tabs under Tab Management
-----------------------------------------


        # Adding sub-tabs under Tab Management

        self.sub_tab_control_manage =
ttk.Notebook(self.management_reporting)

        self.sub_tab_total_rev =
ttk.Frame(self.sub_tab_control_manage)

        self.sub_tab_animal_report =
ttk.Frame(self.sub_tab_control_manage)

        self.sub_tab_revenue_by_time =
ttk.Frame(self.sub_tab_control_manage)
```

```
        self.sub_tab_control_manage.add(self.sub_tab_total_rev,
text='Total Revenue')

        self.sub_tab_control_manage.add(self.sub_tab_animal_report,
text='Animal Report')

        self.sub_tab_control_manage.add(self.sub_tab_revenue_by_time,
text='Revenue By Time Period')



        self.sub_tab_control_manage.pack(expand=1, fill='both')



        #------------------------------------------------- Adding
sub-tabs under Tab Total Revenue
-----------------------------------------



        self.cal_total_revenue_label =
tk.Label(self.sub_tab_total_rev, text='Enter Date in YYYY-MM-DD
Formate : ')

        self.cal_total_revenue_entry =
tk.Entry(self.sub_tab_total_rev)

        self.cal_total_revenue_label.grid(row=0, column=0, padx=5,
pady=5)

        self.cal_total_revenue_entry.grid(row=0, column=1, padx=5,
pady=5)



        self.get_total_revenue_button =
tk.Button(self.sub_tab_total_rev, text='Get Total Report',
command=self.total_rev)

        self.get_total_revenue_button.grid(row=2, column=0,
columnspan=2, pady=10)
```

```
        #------------------------------------------------- Adding
sub-tabs under Tab Animal Report
-----------------------------------------


        self.get_total_revenue_button =
tk.Button(self.sub_tab_animal_report, text='Get Total Report',
command=self.animal_roport)

        self.get_total_revenue_button.grid(row=0, column=0,
columnspan=2, pady=10)


        #------------------------------------------------- Adding
sub-tabs under Tab Revenue By Time Period
-----------------------------------------


        self.sub_tab_control_revenue_time =
ttk.Notebook(self.sub_tab_revenue_by_time)

        self.sub_tab_revenue_time_top_3 =
ttk.Frame(self.sub_tab_control_revenue_time)

        self.sub_tab_revenue_time_best_5 =
ttk.Frame(self.sub_tab_control_revenue_time)

        self.sub_tab_revenue_time_avg =
ttk.Frame(self.sub_tab_control_revenue_time)



self.sub_tab_control_revenue_time.add(self.sub_tab_revenue_time_top_3
, text='Top 3')


self.sub_tab_control_revenue_time.add(self.sub_tab_revenue_time_best_
5, text='Best 5 for Month')


self.sub_tab_control_revenue_time.add(self.sub_tab_revenue_time_avg,
text='Average')
```

```
        self.sub_tab_control_revenue_time.pack(expand=1, fill='both')


        #------------------------------------------------- Adding
sub-tabs under Tab Top-3
-------------------------------------------------------------


        self.top_3_report_sd_label =
tk.Label(self.sub_tab_revenue_time_top_3, text='Enter Start Date in
YYYY-MM-DD format ')

        self.top_3_report_sd_entry =
tk.Entry(self.sub_tab_revenue_time_top_3)

        self.top_3_report_sd_label.grid(row=0, column=0, padx=5,
pady=5)

        self.top_3_report_sd_entry.grid(row=0, column=1, padx=5,
pady=5)


        self.top_3_report_ed_label =
tk.Label(self.sub_tab_revenue_time_top_3, text='Enter End Date in
YYYY-MM-DD format ')

        self.top_3_report_ed_entry =
tk.Entry(self.sub_tab_revenue_time_top_3)

        self.top_3_report_ed_label.grid(row=1, column=0, padx=5,
pady=5)

        self.top_3_report_ed_entry.grid(row=1, column=1, padx=5,
pady=5)


        self.get_top_3_button =
tk.Button(self.sub_tab_revenue_time_top_3, text='Get Report',
command=self.top_3)
```

```
        self.get_top_3_button.grid(row=2, column=0, columnspan=2,
pady=10)


        #------------------------------------------------- Adding
sub-tabs under Tab Best-5
-----------------------------------------------------------


        self.best_5_report_id_label =
tk.Label(self.sub_tab_revenue_time_best_5, text='Enter month in MM
format ')

        self.best_5_report_id_entry =
tk.Entry(self.sub_tab_revenue_time_best_5)

        self.best_5_report_id_label.grid(row=0, column=0, padx=5,
pady=5)

        self.best_5_report_id_entry.grid(row=0, column=1, padx=5,
pady=5)




        self.get_best_5_button =
tk.Button(self.sub_tab_revenue_time_best_5, text='Get Best 5 of the
month', command=self.best_5)

        self.get_best_5_button.grid(row=2, column=0, columnspan=2,
pady=10)


        #------------------------------------------------- Adding
sub-tabs under Tab Average
-----------------------------------------------------------


        self.avg_report_sd_label =
tk.Label(self.sub_tab_revenue_time_avg, text='Enter Start Date in
YYYY-MM-DD format ')
```

```
        self.avg_report_sd_entry =
tk.Entry(self.sub_tab_revenue_time_avg)

        self.avg_report_sd_label.grid(row=0, column=0, padx=5,
pady=5)

        self.avg_report_sd_entry.grid(row=0, column=1, padx=5,
pady=5)



        self.avg_report_ed_label =
tk.Label(self.sub_tab_revenue_time_avg, text='Enter End Date in
YYYY-MM-DD format ')

        self.avg_report_ed_entry =
tk.Entry(self.sub_tab_revenue_time_avg)

        self.avg_report_ed_label.grid(row=1, column=0, padx=5,
pady=5)

        self.avg_report_ed_entry.grid(row=1, column=1, padx=5,
pady=5)



        self.get_avg_button =
tk.Button(self.sub_tab_revenue_time_avg, text='Get Average',
command=self.avg)

        self.get_avg_button.grid(row=2, column=0, columnspan=2,
pady=10)



#------------------------------------------------------- FUNCTIONS
--------------------------------------------------------------------
--------------------------------------------------------------------
-

    def add_animal(self):

        ani_id = self.view_animal_id_entry.get()

        print(ani_id)

        ani_status = self.view_animal_status_entry.get()
```

```
        print(ani_status)

        ani_birth = self.view_animal_birth_entry.get()

        print(ani_birth)

        ani_spc = self.view_animal_spc_id_entry.get()

        print(ani_spc)

        ani_enc = self.view_animal_enc_id_entry.get()

        print(ani_enc)

        ani_bid = self.view_animal_build_id_entry.get()

        print(ani_bid)

        cursor = connection.cursor()

        query = 'INSERT INTO HP578.ANIMAL (Ani_ID , Status,
Birth_Year, Spc_ID, Enc_ID, Building_ID) VALUES(:1, :2, :3, :4, :5,
:6)'

        cursor.execute(query , (ani_id,
ani_status,ani_birth,ani_spc,ani_enc,ani_bid))

        connection.commit()

        cursor.close()

        # connection.close()

        # Clear input fields

        self.view_animal_id_entry.delete(0, 'end')

        self.view_animal_status_entry.delete(0, 'end')

        self.view_animal_birth_entry.delete(0, 'end')

        self.view_animal_spc_id_entry.delete(0, 'end')

        self.view_animal_enc_id_entry.delete(0, 'end')

        self.view_animal_build_id_entry.delete(0, 'end')
```

```
        # Display a message in the label
        message = "Animal added successfully!"
        self.message_label.config(text=message)


        columns = ("Animal_ID", "Status", "Birth Year", "Species Id",
"Encloser Id", "Building ID")
        tree = ttk.Treeview(self.sub_tab_animal, columns=columns,
show="headings")


        # Set column headings
        for col in columns:
            tree.heading(col, text=col)


        # Grid the Treeview
        tree.grid(row=7, column=0, pady=10, padx=10, sticky="nsew" ,
rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb = ttk.Scrollbar(self.sub_tab_animal, orient="vertical",
command=tree.yview)
        vsb.grid(row=7, column=2, pady=10, sticky="ns")
        tree.configure(yscrollcommand=vsb.set)


        hsb = ttk.Scrollbar(self.sub_tab_animal, orient="horizontal",
command=tree.xview)
        hsb.grid(row=10, column=0, padx=10, sticky="ew")
```

```
        tree.configure(xscrollcommand=hsb.set)
        cursor2 = connection.cursor()
        query = "SELECT * FROM HP578.ANIMAL"
        cursor2.execute(query)

        # Fetch all rows from the result
        rows = cursor2.fetchall()
        # Insert data into the Treeview
        for row in rows:
            tree.insert("", "end", values=row)

        def on_treeview_select(event):
    # Get the selected item
            selected_item = tree.selection()

            # Check if any item is selected
            if selected_item:
                # Clear the entries
                self.view_animal_id_entry.delete(0, 'end')
                self.view_animal_status_entry.delete(0, 'end')
                self.view_animal_birth_entry.delete(0, 'end')
                self.view_animal_spc_id_entry.delete(0, 'end')
                self.view_animal_enc_id_entry.delete(0, 'end')
                self.view_animal_build_id_entry.delete(0, 'end')
```

```
                # Get values from the selected item and fill the
entries
                values = tree.item(selected_item)['values']
                if values:
                    self.view_animal_id_entry.insert(0, values[0])  #
Assuming name is the first column
                    self.view_animal_status_entry.insert(0,
values[1])
                    self.view_animal_birth_entry.insert(0, values[2])
                    self.view_animal_spc_id_entry.insert(0,
values[3])
                    self.view_animal_enc_id_entry.insert(0,
values[4])
                    self.view_animal_build_id_entry.insert(0,
values[5])
        tree.bind('<ButtonRelease-1>', on_treeview_select)


    def update_animal(self):
        ani_id = self.view_animal_id_entry.get()
        print(ani_id)
        ani_status = self.view_animal_status_entry.get()
        print(ani_status)
        ani_birth = self.view_animal_birth_entry.get()
        print(ani_birth)
        ani_spc = self.view_animal_spc_id_entry.get()
        print(ani_spc)
        ani_enc = self.view_animal_enc_id_entry.get()
```

```
        print(ani_enc)

        ani_bid = self.view_animal_build_id_entry.get()

        print(ani_bid)

        cursor = connection.cursor()

        query = 'UPDATE HP578.ANIMAL SET Status = :1, Birth_Year =
:2, Spc_ID = :3, Enc_ID = :4, Building_ID = :5 WHERE Ani_ID = :6'

        cursor.execute(query, (ani_status, ani_birth, ani_spc,
ani_enc, ani_bid, ani_id))

        connection.commit()

        cursor.close()

        # Clear input fields

        self.view_animal_id_entry.delete(0, 'end')

        self.view_animal_status_entry.delete(0, 'end')

        self.view_animal_birth_entry.delete(0, 'end')

        self.view_animal_spc_id_entry.delete(0, 'end')

        self.view_animal_enc_id_entry.delete(0, 'end')

        self.view_animal_build_id_entry.delete(0, 'end')


        # Display a message in the label

        message = "Animal data Updated successfully!"

        self.message_label.config(text=message)


        columns = ("Animal_ID", "Status", "Birth Year", "Species Id",
"Encloser Id", "Building ID")

        tree = ttk.Treeview(self.sub_tab_animal, columns=columns,
show="headings")
```

```
        # Set column headings
        for col in columns:
            tree.heading(col, text=col)

        # Grid the Treeview
        tree.grid(row=7, column=0, pady=10, padx=10, sticky="nsew" ,
rowspan=2, columnspan=2, ipadx=10, ipady=10 )

        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb = ttk.Scrollbar(self.sub_tab_animal, orient="vertical",
command=tree.yview)
        vsb.grid(row=7, column=2, pady=10, sticky="ns")
        tree.configure(yscrollcommand=vsb.set)

        hsb = ttk.Scrollbar(self.sub_tab_animal, orient="horizontal",
command=tree.xview)
        hsb.grid(row=10, column=0, padx=10, sticky="ew")
        tree.configure(xscrollcommand=hsb.set)
        cursor2 = connection.cursor()
        query = "SELECT * FROM HP578.ANIMAL"
        cursor2.execute(query)

        # Fetch all rows from the result
        rows = cursor2.fetchall()
        # Insert data into the Treeview
        for row in rows:
```

```
            tree.insert("", "end", values=row)


        def on_treeview_select(event):
    # Get the selected item
            selected_item = tree.selection()


            # Check if any item is selected
            if selected_item:
                # Clear the entries
                self.view_animal_id_entry.delete(0, 'end')
                self.view_animal_status_entry.delete(0, 'end')
                self.view_animal_birth_entry.delete(0, 'end')
                self.view_animal_spc_id_entry.delete(0, 'end')
                self.view_animal_enc_id_entry.delete(0, 'end')
                self.view_animal_build_id_entry.delete(0, 'end')


                # Get values from the selected item and fill the
entries
                values = tree.item(selected_item)['values']
                if values:
                    self.view_animal_id_entry.insert(0, values[0])  #
Assuming name is the first column
                    self.view_animal_status_entry.insert(0,
values[1])
                    self.view_animal_birth_entry.insert(0, values[2])
                    self.view_animal_spc_id_entry.insert(0,
values[3])
```

```
                    self.view_animal_enc_id_entry.insert(0,
values[4])
                    self.view_animal_build_id_entry.insert(0,
values[5])
        tree.bind('<ButtonRelease-1>', on_treeview_select)


    def add_building(self):
        building_id = self.view_building_id_entry.get()
        print(building_id)
        building_name = self.view_building_name_entry.get()
        print(building_name)
        building_type = self.view_building_type_entry.get()
        print(building_type)
        cursor = connection.cursor()
        query = 'INSERT INTO HP578.BUILDING (Building_ID ,
Build_Name, Buid_Type) VALUES(:1, :2, :3)'
        cursor.execute(query , (building_id,
building_name,building_type))
        connection.commit()
        cursor.close()


        self.view_building_id_entry.delete(0, 'end')
        self.view_building_name_entry.delete(0, 'end')
        self.view_building_type_entry.delete(0, 'end')


        # Display a message in the label
        message = "Building added successfully!"
```

```
        self.message_label.config(text=message)


        columns_build = ("Building_ID", "Building_Name",
"Building_Type")
        tree_build = ttk.Treeview(self.sub_tab_building,
columns=columns_build, show="headings")


        # Set column headings
        for col in columns_build:
            tree_build.heading(col, text=col)


        # Grid the Treeview
        tree_build.grid(row=4, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="vertical", command=tree_build.yview)
        vsb_build.grid(row=4, column=2, pady=10, sticky="ns")
        tree_build.configure(yscrollcommand=vsb_build.set)


        hsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="horizontal", command=tree_build.xview)
        hsb_build.grid(row=6, column=0, padx=10, sticky="ew")
        tree_build.configure(xscrollcommand=hsb_build.set)
        cursor3 = connection.cursor()
```

```
        query1 = "SELECT * FROM HP578.BUILDING"
        cursor3.execute(query1)

        # Fetch all rows from the result
        rows_build = cursor3.fetchall()
        # Insert data into the Treeview
        for row in rows_build:
            tree_build.insert("", "end", values=row)

        def on_treeview_select_b(event):
    # Get the selected item
            selected_item_build = tree_build.selection()

            # Check if any item is selected
            if selected_item_build:
                # Clear the entries
                self.view_building_id_entry.delete(0, 'end')
                self.view_building_name_entry.delete(0, 'end')
                self.view_building_type_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values =
tree_build.item(selected_item_build)['values']
                if values:
```

```
                    self.view_building_id_entry.insert(0, values[0])
# Assuming name is the first column
                    self.view_building_name_entry.insert(0,
values[1])
                    self.view_building_type_entry.insert(0,
values[2])
        tree_build.bind('<ButtonRelease-1>', on_treeview_select_b)


    def upd_building(self):
        building_id = self.view_building_id_entry.get()
        print(building_id)
        building_name = self.view_building_name_entry.get()
        print(building_name)
        building_type = self.view_building_type_entry.get()
        print(building_type)
        cursor = connection.cursor()
        query = 'UPDATE HP578.BUILDING SET Build_Name = :1, Buid_Type
= :2 WHERE Building_ID = :3'
        cursor.execute(query , (building_name,building_type,
building_id))
        connection.commit()
        cursor.close()


        self.view_building_id_entry.delete(0, 'end')
        self.view_building_name_entry.delete(0, 'end')
        self.view_building_type_entry.delete(0, 'end')
```

```
        # Display a message in the label
        message = "Building Updated successfully!"
        self.message_label.config(text=message)

        columns_build = ("Building_ID", "Building_Name",
"Building_Type")
        tree_build = ttk.Treeview(self.sub_tab_building,
columns=columns_build, show="headings")

        # Set column headings
        for col in columns_build:
            tree_build.heading(col, text=col)

        # Grid the Treeview
        tree_build.grid(row=4, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )

        # Configure Treeview to allow vertical and horizontal
scrollbar
        vsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="vertical", command=tree_build.yview)
        vsb_build.grid(row=4, column=2, pady=10, sticky="ns")
        tree_build.configure(yscrollcommand=vsb_build.set)

        hsb_build = ttk.Scrollbar(self.sub_tab_building,
orient="horizontal", command=tree_build.xview)
        hsb_build.grid(row=6, column=0, padx=10, sticky="ew")
```

```
        tree_build.configure(xscrollcommand=hsb_build.set)
        cursor3 = connection.cursor()
        query1 = "SELECT * FROM HP578.BUILDING"
        cursor3.execute(query1)

        # Fetch all rows from the result
        rows_build = cursor3.fetchall()
        # Insert data into the Treeview
        for row in rows_build:
            tree_build.insert("", "end", values=row)

        def on_treeview_select_b(event):
    # Get the selected item
            selected_item_build = tree_build.selection()

            # Check if any item is selected
            if selected_item_build:
                # Clear the entries
                self.view_building_id_entry.delete(0, 'end')
                self.view_building_name_entry.delete(0, 'end')
                self.view_building_type_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values =
tree_build.item(selected_item_build)['values']
```

```
                if values:
                    self.view_building_id_entry.insert(0, values[0])
# Assuming name is the first column
                    self.view_building_name_entry.insert(0,
values[1])
                    self.view_building_type_entry.insert(0,
values[2])
        tree_build.bind('<ButtonRelease-1>', on_treeview_select_b)


    def add_attraction(self):
        attraction_rev_id = self.view_attraction_rev_id_entry.get()
        print(attraction_rev_id)
        attraction_c_p = self.view_attraction_c_p_entry.get()
        print(attraction_c_p)
        attraction_s_p = self.view_attraction_s_p_entry.get()
        print(attraction_s_p)
        attraction_a_p = self.view_attraction_a_p_entry.get()
        print(attraction_a_p)
        attraction_per_day = self.view_attraction_per_day_entry.get()
        print(attraction_per_day)
        cursor = connection.cursor()
        query = 'INSERT INTO HP578.ANIMAL_SHOW (Child_Price ,
Adult_Price, Senior_Price, Shows_Per_Day, RevenueTypes_ID) VALUES(:1,
:2, :3, :4, :5)'
        cursor.execute(query , (attraction_c_p, attraction_a_p,
attraction_s_p, attraction_per_day, attraction_rev_id))
        connection.commit()
```

```
        cursor.close()


        self.view_attraction_rev_id_entry.delete(0, 'end')

        self.view_attraction_c_p_entry.delete(0, 'end')

        self.view_attraction_s_p_entry.delete(0, 'end')

        self.view_attraction_a_p_entry.delete(0, 'end')

        self.view_attraction_per_day_entry.delete(0, 'end')


        # Display a message in the label

        message = "Animal Attraction added successfully!"

        self.message_label.config(text=message)


        columns_att = ("Child Price", "Senior Price", "Adult Price",
"Per Day", "Revenue Type ID")

        tree_att = ttk.Treeview(self.sub_tab_attraction,
columns=columns_att, show="headings")


        # Set column headings

        for col in columns_att:

            tree_att.heading(col, text=col)


        # Grid the Treeview

        tree_att.grid(row=6, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar
```

```
        vsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="vertical", command=tree_att.yview)

        vsb_att.grid(row=6, column=2, pady=10, sticky="ns")

        tree_att.configure(yscrollcommand=vsb_att.set)


        hsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="horizontal", command=tree_att.xview)

        hsb_att.grid(row=8, column=0, padx=10, sticky="ew")

        tree_att.configure(xscrollcommand=hsb_att.set)

        cursor4 = connection.cursor()

        query2 = "SELECT * FROM HP578.ANIMAL_SHOW"

        cursor4.execute(query2)


        # Fetch all rows from the result

        rows_att = cursor4.fetchall()

        # Insert data into the Treeview

        for row in rows_att:

            tree_att.insert("", "end", values=row)


        def on_treeview_select_b(event):

    # Get the selected item

            selected_item_att = tree_att.selection()


            # Check if any item is selected

            if selected_item_att:

                # Clear the entries
```

```
                self.view_attraction_rev_id_entry.delete(0, 'end')
                self.view_attraction_c_p_entry.delete(0, 'end')
                self.view_attraction_s_p_entry.delete(0, 'end')
                self.view_attraction_a_p_entry.delete(0, 'end')
                self.view_attraction_per_day_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_att.item(selected_item_att)['values']
                if values:
                    self.view_attraction_rev_id_entry.insert(0,
values[4])  # Assuming name is the first column
                    self.view_attraction_c_p_entry.insert(0,
values[0])
                    self.view_attraction_s_p_entry.insert(0,
values[1])
                    self.view_attraction_a_p_entry.insert(0,
values[2])
                    self.view_attraction_per_day_entry.insert(0,
values[3])
        tree_att.bind('<ButtonRelease-1>', on_treeview_select_b)

    def upd_attraction(self):
        attraction_rev_id = self.view_attraction_rev_id_entry.get()
        print(attraction_rev_id)
        attraction_c_p = self.view_attraction_c_p_entry.get()
        print(attraction_c_p)
```

```
        attraction_s_p = self.view_attraction_s_p_entry.get()

        print(attraction_s_p)

        attraction_a_p = self.view_attraction_a_p_entry.get()

        print(attraction_a_p)

        attraction_per_day = self.view_attraction_per_day_entry.get()

        print(attraction_per_day)

        cursor = connection.cursor()

        query = 'UPDATE HP578.ANIMAL_SHOW SET Child_Price = :1,
Adult_Price = :2, Senior_Price = :3, Shows_Per_Day = :4 WHERE
RevenueTypes_ID = :5'

        cursor.execute(query , (attraction_c_p, attraction_a_p,
attraction_s_p, attraction_per_day, attraction_rev_id))

        connection.commit()

        cursor.close()


        self.view_attraction_rev_id_entry.delete(0, 'end')

        self.view_attraction_c_p_entry.delete(0, 'end')

        self.view_attraction_s_p_entry.delete(0, 'end')

        self.view_attraction_a_p_entry.delete(0, 'end')

        self.view_attraction_per_day_entry.delete(0, 'end')


        # Display a message in the label

        message = "Animal Attraction added successfully!"

        self.message_label.config(text=message)


        columns_att = ("Child Price", "Senior Price", "Adult Price",
"Per Day", "Revenue Type ID")
```

```
        tree_att = ttk.Treeview(self.sub_tab_attraction,
columns=columns_att, show="headings")


        # Set column headings

        for col in columns_att:

            tree_att.heading(col, text=col)


        # Grid the Treeview

        tree_att.grid(row=6, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="vertical", command=tree_att.yview)

        vsb_att.grid(row=6, column=2, pady=10, sticky="ns")

        tree_att.configure(yscrollcommand=vsb_att.set)


        hsb_att = ttk.Scrollbar(self.sub_tab_attraction,
orient="horizontal", command=tree_att.xview)

        hsb_att.grid(row=8, column=0, padx=10, sticky="ew")

        tree_att.configure(xscrollcommand=hsb_att.set)

        cursor4 = connection.cursor()

        query2 = "SELECT * FROM HP578.ANIMAL_SHOW"

        cursor4.execute(query2)


        # Fetch all rows from the result
```

```
        rows_att = cursor4.fetchall()
        # Insert data into the Treeview
        for row in rows_att:
            tree_att.insert("", "end", values=row)

        def on_treeview_select_b(event):
    # Get the selected item
            selected_item_att = tree_att.selection()

            # Check if any item is selected
            if selected_item_att:
                # Clear the entries
                self.view_attraction_rev_id_entry.delete(0, 'end')
                self.view_attraction_c_p_entry.delete(0, 'end')
                self.view_attraction_s_p_entry.delete(0, 'end')
                self.view_attraction_a_p_entry.delete(0, 'end')
                self.view_attraction_per_day_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_att.item(selected_item_att)['values']
                if values:
                    self.view_attraction_rev_id_entry.insert(0,
values[4])  # Assuming name is the first column
                    self.view_attraction_c_p_entry.insert(0,
values[0])
```

```
                    self.view_attraction_s_p_entry.insert(0,
values[1])
                    self.view_attraction_a_p_entry.insert(0,
values[2])
                    self.view_attraction_per_day_entry.insert(0,
values[3])
        tree_att.bind('<ButtonRelease-1>', on_treeview_select_b)


    def add_employee(self):
        emp_id = self.view_employee_id_entry.get()
        print(emp_id)
        sd_p = self.view_employee_start_entry.get()
        print(sd_p)
        jt = self.view_employee_job_entry.get()
        print(jt)
        fname = self.view_employee_f_entry.get()
        print(fname)
        minit = self.view_employee_m_entry.get()
        lname = self.view_employee_l_entry.get()
        print(lname)
        street = self.view_employee_st_entry.get()
        print(street)
        city = self.view_employee_city_entry.get()
        print(city)
        state = self.view_employee_state_entry.get()
        print(state)
```

```
        zip = self.view_employee_zip_entry.get()

        print(zip)

        hrid = self.view_employee_hr_id_entry.get()

        print(hrid)

        sup_id = self.view_employee_sup_id_entry.get()

        print(sup_id)

        rev_id = self.view_employee_rev_id_entry.get()

        print(rev_id)

        cursor = connection.cursor()

        # #INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname,
Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID,
RevenueTypes_ID) VALUES('563214789', TO_DATE('2022-01-01',
'YYYY-MM-DD'), 'Veterinarian', 'John', 'D', 'Doe', '123 Main St',
'New York', 'NY', '10001', 1, NULL, NULL);

        query = "INSERT INTO HP578.EMPLOYEE (Emp_ID, Start_Date,
JobType, Fname, Minit, Lname, Street ,City, State_Name, PinCode,
Hr_ID, Super_ID, RevenueTypes_ID) VALUES(:1, TO_DATE(:2,
'YYYY-MM-DD'), :3, :4, :5, :6, :7, :8, :9, :10, :11, :12, :13)"

        cursor.execute(query ,
(emp_id,sd_p,jt,fname,minit,lname,street,city,state,zip,hrid,sup_id,r
ev_id))

        connection.commit()

        cursor.close()


        message = "Employee inserted successfully!"

        self.message_label.config(text=message)
```

```
        columns_em = ("Employee ID", "Start Date", "Job Type",
"First","Middle initial", "Last", "Street","City","State","Zip","HR
ID","Super ID","Revenue Type ID")

        tree_em = ttk.Treeview(self.sub_tab_employees,
columns=columns_em, show="headings")


        # Set column headings

        for col in columns_em:

            tree_em.heading(col, text=col)


        # Grid the Treeview

        tree_em.grid(row=14, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="vertical", command=tree_em.yview)

        vsb_em.grid(row=14, column=1, pady=10, sticky="ns")

        tree_em.configure(yscrollcommand=vsb_em.set)


        hsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="horizontal", command=tree_em.xview)

        hsb_em.grid(row=16, column=0, padx=10, sticky="ew")

        tree_em.configure(xscrollcommand=hsb_em.set)

        cursor6 = connection.cursor()

        query4 = "SELECT * FROM HP578.EMPLOYEE"

        cursor6.execute(query4)
```

```
    # Fetch all rows from the result
    rows = cursor6.fetchall()
    # Insert data into the Treeview
    for row in rows:
        tree_em.insert("", "end", values=row)

    def on_treeview_select_em(event):
# Get the selected item
        selected_item = tree_em.selection()

        # Check if any item is selected
        if selected_item:
            # Clear the entries
            self.view_employee_id_entry.delete(0, 'end')
            self.view_employee_start_entry.delete(0, 'end')
            self.view_employee_job_entry.delete(0, 'end')
            self.view_employee_f_entry.delete(0, 'end')
            self.view_employee_m_entry.delete(0, 'end')
            self.view_employee_l_entry.delete(0, 'end')
            self.view_employee_st_entry.delete(0, 'end')
            self.view_employee_city_entry.delete(0, 'end')
            self.view_employee_state_entry.delete(0, 'end')
            self.view_employee_zip_entry.delete(0, 'end')
            self.view_employee_hr_id_entry.delete(0, 'end')
```

```
                self.view_employee_sup_id_entry.delete(0, 'end')
                self.view_employee_rev_id_entry.delete(0, 'end')



                # Get values from the selected item and fill the
entries
                values = tree_em.item(selected_item)['values']
                if values:
                    self.view_employee_id_entry.insert(0, values[0])
                    self.view_employee_start_entry.insert(0,
values[1])
                    self.view_employee_job_entry.insert(0, values[2])
                    self.view_employee_f_entry.insert(0, values[3])
# Assuming name is the first column
                    self.view_employee_m_entry.insert(0, values[4])
                    self.view_employee_l_entry.insert(0, values[5])
                    self.view_employee_st_entry.insert(0, values[6])
                    self.view_employee_city_entry.insert(0,
values[7])
                    self.view_employee_state_entry.insert(0,
values[8])
                    self.view_employee_zip_entry.insert(0, values[9])
# Assuming name is the first column
                    self.view_employee_hr_id_entry.insert(0,
values[10])
                    self.view_employee_sup_id_entry.insert(0,
values[11])
                    self.view_employee_rev_id_entry.insert(0,
values[12])
```

```
        tree_em.bind('<ButtonRelease-1>', on_treeview_select_em)


    def upd_employee(self):
        emp_id = self.view_employee_id_entry.get()
        print(emp_id)
        sd_p = self.view_employee_start_entry.get()
        sd_p=sd_p[0:10]
        print(sd_p)
        jt = self.view_employee_job_entry.get()
        print(jt)
        fname = self.view_employee_f_entry.get()
        print(fname)
        minit = self.view_employee_m_entry.get()
        lname = self.view_employee_l_entry.get()
        print(lname)
        street = self.view_employee_st_entry.get()
        print(street)
        city = self.view_employee_city_entry.get()
        print(city)
        state = self.view_employee_state_entry.get()
        print(state)
        zip = self.view_employee_zip_entry.get()
        print(zip)
        hrid = self.view_employee_hr_id_entry.get()
        print(hrid)
```

```
        sup_id = self.view_employee_sup_id_entry.get()

        if sup_id is None or sup_id=="None" :

            sup_id=None

        print(sup_id)

        rev_id = self.view_employee_rev_id_entry.get()

        if rev_id is None or rev_id=="None" :

            rev_id=None

        print(rev_id)

        cursor = connection.cursor()

        # #INSERT INTO EMPLOYEE (Emp_ID, Start_Date, JobType, Fname,
Minit, Lname, Street ,City, State_Name, PinCode, Hr_ID, Super_ID,
RevenueTypes_ID) VALUES('563214789', TO_DATE('2022-01-01',
'YYYY-MM-DD'), 'Veterinarian', 'John', 'D', 'Doe', '123 Main St',
'New York', 'NY', '10001', 1, NULL, NULL);

        query = "UPDATE HP578.EMPLOYEE SET Start_Date=TO_DATE(:1,
'YYYY-MM-DD') , JobType=:2, Fname=:3, Minit=:4, Lname=:5, Street=:6
,City=:7, State_Name=:8, PinCode=:9, Hr_ID=:10, Super_ID=:11,
RevenueTypes_ID=:12 WHERE Emp_ID =:13 "

        cursor.execute(query ,
(sd_p,jt,fname,minit,lname,street,city,state,zip,hrid,sup_id,rev_id,e
mp_id))

        connection.commit()

        # cursor.close()


        message = "Employee inserted successfully!"

        self.message_label.config(text=message)
```

```
        columns_em = ("Employee ID", "Start Date", "Job Type",
"First","Middle initial", "Last", "Street","City","State","Zip","HR
ID","Super ID","Revenue Type ID")

        tree_em = ttk.Treeview(self.sub_tab_employees,
columns=columns_em, show="headings")


        # Set column headings

        for col in columns_em:

            tree_em.heading(col, text=col)


        def on_treeview_configure(event):

            # Update the scroll region to cover the entire treeview

            tree_em.update_idletasks()

            hsb_em.configure(command=tree_em.xview,
scrollregion=tree_em.bbox("all"))


        # Grid the Treeview

        tree_em.grid(row=14, column=0, pady=10, padx=10,
sticky="nsew" , rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="vertical", command=tree_em.yview)

        vsb_em.grid(row=14, column=2, pady=10, sticky="ns")

        tree_em.configure(yscrollcommand=vsb_em.set)
```

```
        hsb_em = ttk.Scrollbar(self.sub_tab_employees,
orient="horizontal", command=tree_em.xview)
        tree_em.configure(xscrollcommand=hsb_em.set)
        hsb_em.grid(row=16, column=0, padx=10, sticky="ew")


        tree_em.bind("<Configure>", on_treeview_configure)
        cursor6 = connection.cursor()
        query4 = "SELECT * FROM HP578.EMPLOYEE"
        cursor6.execute(query4)


        # Fetch all rows from the result
        rows = cursor6.fetchall()
        # Insert data into the Treeview
        for row in rows:
            tree_em.insert("", "end", values=row)


        def on_treeview_select_em(event):
    # Get the selected item
            selected_item = tree_em.selection()


            # Check if any item is selected
            if selected_item:
                # Clear the entries
                self.view_employee_id_entry.delete(0, 'end')
                self.view_employee_start_entry.delete(0, 'end')
```

```
                self.view_employee_job_entry.delete(0, 'end')
                self.view_employee_f_entry.delete(0, 'end')
                self.view_employee_m_entry.delete(0, 'end')
                self.view_employee_l_entry.delete(0, 'end')
                self.view_employee_st_entry.delete(0, 'end')
                self.view_employee_city_entry.delete(0, 'end')
                self.view_employee_state_entry.delete(0, 'end')
                self.view_employee_zip_entry.delete(0, 'end')
                self.view_employee_hr_id_entry.delete(0, 'end')
                self.view_employee_sup_id_entry.delete(0, 'end')
                self.view_employee_rev_id_entry.delete(0, 'end')



                # Get values from the selected item and fill the
entries
                values = tree_em.item(selected_item)['values']
                if values:
                    self.view_employee_id_entry.insert(0, values[0])
                    self.view_employee_start_entry.insert(0,
values[1])
                    self.view_employee_job_entry.insert(0, values[2])
                    self.view_employee_f_entry.insert(0, values[3])
# Assuming name is the first column
                    self.view_employee_m_entry.insert(0, values[4])
                    self.view_employee_l_entry.insert(0, values[5])
                    self.view_employee_st_entry.insert(0, values[6])
```

```
                    self.view_employee_city_entry.insert(0,
values[7])

                    self.view_employee_state_entry.insert(0,
values[8])

                    self.view_employee_zip_entry.insert(0, values[9])
# Assuming name is the first column

                    self.view_employee_hr_id_entry.insert(0,
values[10])

                    self.view_employee_sup_id_entry.insert(0,
values[11])

                    self.view_employee_rev_id_entry.insert(0,
values[12])

        tree_em.bind('<ButtonRelease-1>', on_treeview_select_em)


    def add_hourly(self):

        hr_id = self.view_hourly_id_entry.get()

        print(hr_id)

        hr_rate = self.view_hourly_rate_entry.get()

        print(hr_rate)

        cursor = connection.cursor()

        query = 'INSERT INTO HP578.HOURLY_RATE (Hr_ID , Rate)
VALUES(:1, :2)'

        cursor.execute(query , (hr_id, hr_rate))

        connection.commit()

        cursor.close()

        # connection.close()


        self.view_hourly_id_entry.delete(0, 'end')
```

```
        self.view_hourly_rate_entry.delete(0, 'end')


        # Display a message in the label

        message = "Hourly Rate added successfully!"

        self.message_label.config(text=message)


        columns_hw = ("HR ID", "Rate")

        tree_hw = ttk.Treeview(self.sub_tab_hourly,
columns=columns_hw, show="headings")


        # Set column headings

        for col in columns_hw:

            tree_hw.heading(col, text=col)


        # Grid the Treeview

        tree_hw.grid(row=3, column=0, pady=10, padx=10, sticky="nsew"
, rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="vertical", command=tree_hw.yview)

        vsb_hw.grid(row=3, column=2, pady=10, sticky="ns")

        tree_hw.configure(yscrollcommand=vsb_hw.set)


        hsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="horizontal", command=tree_hw.xview)
```

```
        hsb_hw.grid(row=5, column=0, padx=10, sticky="ew")
        tree_hw.configure(xscrollcommand=hsb_hw.set)
        cursor4 = connection.cursor()
        query2 = "SELECT * FROM HP578.HOURLY_RATE"
        cursor4.execute(query2)

        # Fetch all rows from the result
        rows_hw = cursor4.fetchall()
        # Insert data into the Treeview
        for row in rows_hw:
            tree_hw.insert("", "end", values=row)

        def on_treeview_select_hw(event):
    # Get the selected item
            selected_item_hw = tree_hw.selection()

            # Check if any item is selected
            if selected_item_hw:
                # Clear the entries
                self.view_hourly_id_entry.delete(0, 'end')
                self.view_hourly_rate_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_hw.item(selected_item_hw)['values']
```

```
                if values:
                    self.view_hourly_id_entry.insert(0, values[0])  # Assuming name is the first column
                    self.view_hourly_rate_entry.insert(0, values[1])
        tree_hw.bind('<ButtonRelease-1>', on_treeview_select_hw)


    def upd_hourly(self):
        hr_id = self.view_hourly_id_entry.get()
        print(hr_id)
        hr_rate = self.view_hourly_rate_entry.get()
        print(hr_rate)
        cursor = connection.cursor()
        query = 'UPDATE HP578.HOURLY_RATE SET Rate=:1 WHERE Hr_Id=:2'
        cursor.execute(query , ( hr_rate,hr_id))
        connection.commit()
        cursor.close()
        # connection.close()


        self.view_hourly_id_entry.delete(0, 'end')
        self.view_hourly_rate_entry.delete(0, 'end')


        # Display a message in the label
        message = "Hourly Rate added successfully!"
        self.message_label.config(text=message)
```

```
        columns_hw = ("HR ID", "Rate")

        tree_hw = ttk.Treeview(self.sub_tab_hourly,
columns=columns_hw, show="headings")


        # Set column headings

        for col in columns_hw:

            tree_hw.heading(col, text=col)


        # Grid the Treeview

        tree_hw.grid(row=3, column=0, pady=10, padx=10, sticky="nsew"
, rowspan=2, columnspan=2, ipadx=10, ipady=10 )


        # Configure Treeview to allow vertical and horizontal
scrollbar

        vsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="vertical", command=tree_hw.yview)

        vsb_hw.grid(row=3, column=2, pady=10, sticky="ns")

        tree_hw.configure(yscrollcommand=vsb_hw.set)


        hsb_hw = ttk.Scrollbar(self.sub_tab_hourly,
orient="horizontal", command=tree_hw.xview)

        hsb_hw.grid(row=5, column=0, padx=10, sticky="ew")

        tree_hw.configure(xscrollcommand=hsb_hw.set)

        cursor4 = connection.cursor()

        query2 = "SELECT * FROM HP578.HOURLY_RATE"

        cursor4.execute(query2)
```

```
        # Fetch all rows from the result
        rows_hw = cursor4.fetchall()
        # Insert data into the Treeview
        for row in rows_hw:
            tree_hw.insert("", "end", values=row)

        def on_treeview_select_hw(event):
    # Get the selected item
            selected_item_hw = tree_hw.selection()

            # Check if any item is selected
            if selected_item_hw:
                # Clear the entries
                self.view_hourly_id_entry.delete(0, 'end')
                self.view_hourly_rate_entry.delete(0, 'end')

                # Get values from the selected item and fill the
entries
                values = tree_hw.item(selected_item_hw)['values']
                if values:
                    self.view_hourly_id_entry.insert(0, values[0])  #
Assuming name is the first column
                    self.view_hourly_rate_entry.insert(0, values[1])
        tree_hw.bind('<ButtonRelease-1>', on_treeview_select_hw)

    def view_hourly(self):
```

```
        print("Hello")


    def best_5(self):
        month = self.best_5_report_id_entry.get()
        print(month)
        cursor = connection.cursor()
        query = '''
        WITH DailyRevenue AS (
        SELECT
            TO_CHAR(Date_Time, 'YYYY-MM-DD') AS RevenueDate,
            SUM(Revenue) AS TotalRevenue
        FROM
            HP578.REVENUE_EVENTS
            WHERE
            EXTRACT(MONTH FROM Date_Time) = :1
        GROUP BY
            TO_CHAR(Date_Time, 'YYYY-MM-DD')
        )


        SELECT
            RevenueDate,
            TotalRevenue
        FROM (
            SELECT
                RevenueDate,
```

```
                TotalRevenue,
                RANK() OVER (ORDER BY TotalRevenue DESC) AS
RevenueRank
            FROM
                DailyRevenue
        )
        WHERE
            RevenueRank <= 5
        ORDER BY
        TotalRevenue DESC
        '''
        cursor.execute(query, {'1': month})
        connection.commit()
        #cursor.close()
        # connection.close()


        self.best_5_report_id_entry.delete(0, 'end')


        # Display a message in the label
        message = "Report made successfully!"
        self.message_label.config(text=message)


        self.tree = ttk.Treeview(self.sub_tab_revenue_time_best_5)
        self.tree['show']='headings'
        self.tree["columns"] = ("RevenueDate", "TotalRevenue")
```

```
        self.tree.heading("RevenueDate", text="Revenue Date")
        self.tree.heading("TotalRevenue", text="Total Revenue")
        self.tree.grid(row=4, column=0, columnspan=2,padx=10,
pady=10)



        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)


    def top_3(self):
        st_d = self.top_3_report_sd_entry.get()
        print(st_d)
        ed_d = self.top_3_report_ed_entry.get()
        print(ed_d)
        cursor = connection.cursor()
        query = '''
        WITH AttractionRevenue AS (
            SELECT
                rt.Rev_Name AS Attraction,
                SUM(re.Revenue) AS TotalRevenue
            FROM
                HP578.REVENUE_EVENTS re
            JOIN REVENUE_TYPES rt ON re.RevenueTypes_ID =
rt.RevenueTypes_ID
            WHERE
```

```
                re.Date_Time BETWEEN TO_DATE(:1, 'YYYY-MM-DD') AND
TO_DATE(:2, 'YYYY-MM-DD')
            GROUP BY
                rt.Rev_Name
        )
        SELECT
            Attraction,
            TotalRevenue
        FROM
            (
                SELECT
                    Attraction,
                    TotalRevenue,
                    RANK() OVER (ORDER BY TotalRevenue DESC) AS rnk
                FROM
                    AttractionRevenue
            )
        WHERE
            rnk <= 4 and rnk>1
        '''
        cursor.execute(query, (st_d,ed_d))
        connection.commit()


        self.top_3_report_sd_entry.delete(0, 'end')
        self.top_3_report_ed_entry.delete(0, 'end')
```

```
        # Display a message in the label
        message = "Report generated successfully!"
        self.message_label.config(text=message)

        self.tree = ttk.Treeview(self.sub_tab_revenue_time_top_3)
        self.tree['show']='headings'
        self.tree["columns"] = ("ATTRACTION", "TOTALREVENUE")
        self.tree.heading("ATTRACTION", text="ATTRACTION")
        self.tree.heading("TOTALREVENUE", text="TOTALREVENUE")
        self.tree.grid(row=4, column=0, columnspan=2,padx=10,
pady=10)


        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)

    def attraction_report(self):
        att_date = self.date_attractions_entry.get()
        print(att_date)
        att_name=self.dropdown1.get()
        print(att_name)
        att_type=self.dropdown2.get()
        print(att_type)
        att_type=att_type+'_Price'
```

```
        print(att_type)


        cursor = connection.cursor()
        if att_type=='Adult_Price':
            query = """
                UPDATE HP578.revenue_events
                SET REVENUE = REVENUE + (SELECT ADULT_PRICE FROM
HP578.ANIMAL_SHOW WHERE revenuetypes_id = (SELECT revenuetypes_id
FROM HP578.revenue_types WHERE revenue_types.rev_type=:2)),
                    tickets_sold = tickets_sold + 1
                WHERE date_time = TO_DATE(:1, 'YYYY-MM-DD') AND
revenuetypes_id = (SELECT revenuetypes_id FROM HP578.revenue_types
WHERE revenue_types.rev_type=:2)
                        """
        elif att_type=='Child_Price':
            query = """
                UPDATE HP578.revenue_events
                SET REVENUE = REVENUE + (SELECT CHILD_PRICE FROM
HP578.ANIMAL_SHOW WHERE revenuetypes_id = (SELECT revenuetypes_id
FROM HP578.revenue_types WHERE revenue_types.rev_type=:2)),
                    tickets_sold = tickets_sold + 1
                WHERE date_time = TO_DATE(:1, 'YYYY-MM-DD') AND
revenuetypes_id = (SELECT revenuetypes_id FROM HP578.revenue_types
WHERE revenue_types.rev_type=:2)
                        """
        elif att_type=='Senior_Price':
            query = """
                UPDATE HP578.revenue_events
```

```
                SET REVENUE = REVENUE + (SELECT SENIOR_PRICE FROM
HP578.ANIMAL_SHOW WHERE revenuetypes_id = (SELECT revenuetypes_id
FROM HP578.revenue_types WHERE revenue_types.rev_type=:2)),

                    tickets_sold = tickets_sold + 1

                WHERE date_time = TO_DATE(:1, 'YYYY-MM-DD') AND
revenuetypes_id = (SELECT revenuetypes_id FROM HP578.revenue_types
WHERE revenue_types.rev_type=:2)

                        """


        cursor.execute(query,{'1': att_date,'2':att_name})

        connection.commit()

        cursor.close()


        # Display a message in the label

        message = "Report generated successfully!"

        self.message_label.config(text=message)


        self.tree = ttk.Treeview(self.sub_tab_daily_attractions)

        self.tree['show']='headings'

        self.tree["columns"] = ("REVENUETYPES_ID", "REV_NAME",
"tickets_sold", "date_time", "revenue")

        self.tree.heading("REVENUETYPES_ID", text="REVENUETYPES_ID")

        self.tree.heading("REV_NAME", text="REV_NAME")

        self.tree.heading("tickets_sold", text="tickets_sold")

        self.tree.heading("date_time", text="date_time")

        self.tree.heading("revenue", text="revenue")
```

```
        self.tree.grid(row=4, column=0, columnspan=2,padx=10,
pady=10)


        cursor = connection.cursor()

        query = '''

        SELECT

            re.REVENUETYPES_ID AS Attraction_ID,

            rt.REV_NAME AS Attraction_Location,

            re.tickets_sold AS Ticket_Sold,

            re.date_time AS Date_Of_Attraction,

            re.revenue

        FROM

            HP578.revenue_events re

            JOIN

            HP578.revenue_types rt ON re.REVENUETYPES_ID =
rt.REVENUETYPES_ID

            WHERE

            re.REVENUETYPES_ID IN (SELECT DISTINCT REVENUETYPES_ID
FROM animal_show)

        '''

        cursor.execute(query)


        for row in cursor.fetchall():

            self.tree.insert("", "end", values=row)


        connection.commit()
```

```
        cursor.close()
        # connection.close()


    def attendance_report(self):
        atd_date = self.date_attendance_entry.get()
        print(atd_date)
        atd_name=self.dropdown_atd.get()
        print(atd_name)
        atd_type=self.dropdown_atd_2.get()
        print(atd_type)
        atd_type=atd_type+'_Price'
        print(atd_type)


        cursor = connection.cursor()


        if atd_type=='Adult_Price':
            query = """
                UPDATE HP578.revenue_events
SET revenue = revenue+ (SELECT ADULT_PRICE FROM HP578.zoo_admission
WHERE revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)),
tickets_sold=tickets_sold + 1 WHERE date_time = TO_DATE(:1,
'YYYY-MM-DD') AND revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)
                        """
        elif atd_type=='Child_Price':
            query = """
```

```
                UPDATE HP578.revenue_events
SET revenue = revenue+ (SELECT CHILD_PRICE FROM HP578.zoo_admission
WHERE revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)),
tickets_sold=tickets_sold + 1 WHERE date_time = TO_DATE(:1,
'YYYY-MM-DD') AND revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)
                        """
        elif atd_type=='Senior_Price':
            query = """
                UPDATE HP578.revenue_events
SET revenue = revenue+ (SELECT SENIOR_PRICE FROM HP578.zoo_admission
WHERE revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)),
tickets_sold=tickets_sold + 1 WHERE date_time = TO_DATE(:1,
'YYYY-MM-DD') AND revenuetypes_id=(SELECT revenuetypes_id FROM
HP578.revenue_types WHERE revenue_types.rev_type=:2)
                        """

        # cursor.execute(query)
        cursor.execute(query,{'1': atd_date,'2':atd_name})
        connection.commit()
        cursor.close()
        # connection.close()

        # Display a message in the label
        message = "Report generated successfully!"
        self.message_label.config(text=message)
```

```
        self.tree = ttk.Treeview(self.sub_tab_daily_attendance)
        self.tree['show']='headings'
        self.tree["columns"] = ("Revenue_ID", "Attendance", "Date_",
"Total_Revenue")
        self.tree.heading("Revenue_ID", text="Revenue_ID")
        self.tree.heading("Attendance", text="Attendance")
        self.tree.heading("Date_", text="Date_")
        self.tree.heading("Total_Revenue", text="Total_Revenue")
        self.tree.grid(row=4, column=0, columnspan=2,padx=10,
pady=10)



        cursor = connection.cursor()
        query = '''
        SELECT
            re.RevenueTypes_ID AS Revenue_ID,
            re.tickets_sold AS Attendance,
            re.date_time AS Date_,
            re.revenue AS Total_Revenue
        FROM
            HP578.zoo_admission za
        JOIN
            HP578.revenue_events re ON za.RevenueTypes_ID =
re.RevenueTypes_ID
```

```
        '''
        cursor.execute(query)

        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)

        connection.commit()
        cursor.close()

    def concession_report(self):
        con_date = self.date_concession_entry.get()
        print(con_date)
        con_name=self.dropdown_con.get()
        print(con_name)

        cursor = connection.cursor()
        query = '''
        UPDATE REVENUE_EVENTS
    SET revenue=revenue+(SELECT product from CONCESSION WHERE
revenuetypes_id=(SELECT revenuetypes_id FROM HP578.revenue_types
WHERE revenue_types.rev_name=:2)),
    tickets_sold=tickets_sold + 1
    WHERE date_time = TO_DATE(:1, 'YYYY-MM-DD') AND
revenuetypes_id=(SELECT revenuetypes_id FROM HP578.revenue_types
WHERE revenue_types.rev_name=:2)
        '''
        cursor.execute(query,{'1': con_date,'2':con_name})
```

```
        connection.commit()

        cursor.close()


        # Display a message in the label

        message = "Report generated successfully!"

        self.message_label.config(text=message)


        self.tree = ttk.Treeview(self.sub_tab_daily_concessions)

        self.tree['show']='headings'

        self.tree["columns"] = ("Concessions_ID", "Product_Location",
"Total_Item_Sold","Date_", "Revenue")

        self.tree.heading("Concessions_ID", text="Concessions_ID")

        self.tree.heading("Product_Location",
text="Product_Location")

        self.tree.heading("Total_Item_Sold", text="Total_Item_Sold")

        self.tree.heading("Date_", text="Date_")

        self.tree.heading("Revenue", text="Revenue")

        self.tree.grid(row=3, column=0, columnspan=2,padx=10,
pady=10)




        cursor = connection.cursor()

        query = '''

        SELECT

            c.RevenueTypes_ID AS Concessions_ID,

            rt.rev_name AS Product_Location,
```

```
            re.tickets_sold AS Total_Item_Sold,
            re.date_time AS Date_,
            re.revenue AS Revenue
        FROM
            CONCESSION c
        JOIN
            revenue_events re on c.RevenueTypes_ID=re.RevenueTypes_ID
        JOIN
            revenue_types rt on rt.revenuetypes_id=re.revenuetypes_id
        '''
        cursor.execute(query)

        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)

        connection.commit()
        cursor.close()

    def total_rev(self):
        date_total = self.cal_total_revenue_entry.get()
        print(date_total)
        cursor = connection.cursor()
        query = '''
        SELECT
            re.date_time AS DATE_,
```

```
            rt.rev_name AS Revenue_Name,
            re.tickets_sold AS Ticket_Sold,
            re.revenue AS Revenue
        FROM
            HP578.revenue_events re
        JOIN
            HP578.revenue_types rt on
re.revenuetypes_id=rt.revenuetypes_id and re.date_time=TO_DATE(:1,
'YYYY-MM-DD')
        '''
        cursor.execute(query,{'1': date_total})
        connection.commit()

        # Display a message in the label
        message = "Report generated successfully!"
        self.message_label.config(text=message)

        self.tree = ttk.Treeview(self.sub_tab_total_rev)
        self.tree['show']='headings'
        self.tree["columns"] = ("DATE_", "Revenue_Name",
"Ticket_Sold", "Revenue")
        self.tree.heading("DATE_", text="DATE_")
        self.tree.heading("Revenue_Name", text="Revenue_Name")
        self.tree.heading("Ticket_Sold", text="Ticket_Sold")
        self.tree.heading("Revenue", text="Revenue")
        self.tree.grid(row=3, column=0, columnspan=2,padx=10,
pady=10)
```

```
        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)


    def animal_roport(self):
        cursor = connection.cursor()
        query = '''
        SELECT
            S.Spc_ID,
            S.Spc_Name,
            COUNT(A.Ani_ID) AS Total_Animals,
            COUNT(CASE WHEN A.Status = 'Healthy' THEN 1 END) AS
Healthy_Animals,
            COUNT(CASE WHEN A.Status = 'Ill' THEN 1 END) AS
Sick_Animals,
            COUNT(CASE WHEN A.Status = 'Maternity' THEN 1 END) AS
Animals_in_Maternity,
            SUM(S.Food_Cost) AS Total_Food_Cost,
            NVL((select rate FROM hourly_rate where Hr_ID=1) * 40 *
COUNT(CASE WHEN A.Status = 'Ill' THEN 1 END),0) AS Total_Vet_Cost,
            NVL(((select rate FROM hourly_rate where Hr_ID=2) * 40 *
COUNT(CASE WHEN A.Status = 'Maternity' THEN 1 END)),0) AS
Total_Care_Specialist_Cost
        FROM
            SPECIES S
        JOIN
```

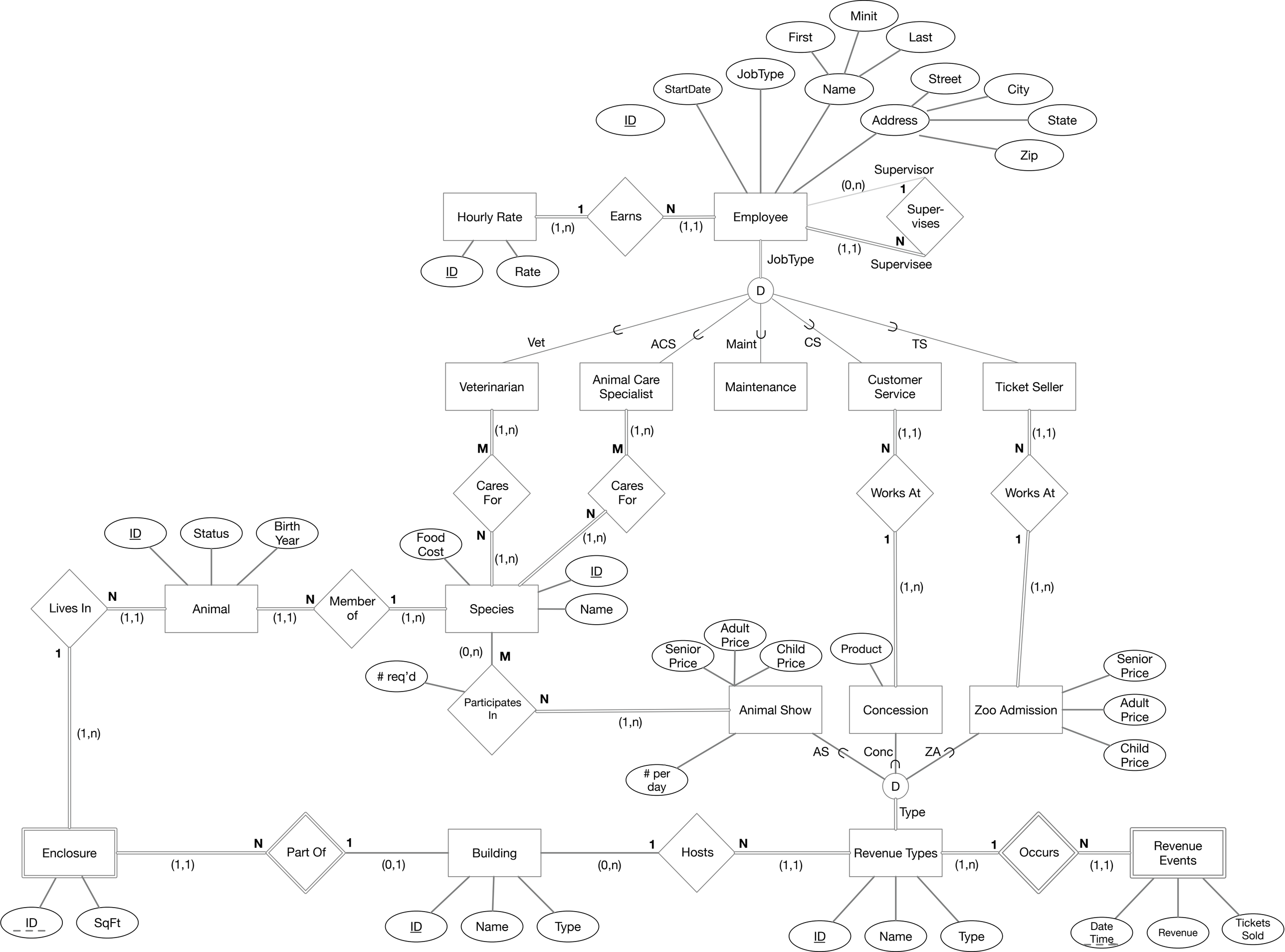
```
            ANIMAL A ON S.Spc_ID = A.Spc_ID

        LEFT JOIN

            CARES_FOR CF ON S.Spc_ID = CF.Spc_ID

        LEFT JOIN

            EMPLOYEE E ON CF.Emp_ID = E.Emp_ID

        LEFT JOIN

            HOURLY_RATE HR_Vet ON E.Hr_ID = HR_Vet.Hr_ID AND
E.JobType = 'Veterinarian'

        LEFT JOIN

            HOURLY_RATE HR_Spec ON E.Hr_ID = HR_Spec.Hr_ID AND
E.JobType = 'Animal Care'

        GROUP BY

            S.Spc_ID,S.Spc_Name

        '''

        cursor.execute(query)

        connection.commit()


        # Display a message in the label

        message = "Report generated successfully!"

        self.message_label.config(text=message)


        self.tree = ttk.Treeview(self.sub_tab_animal_report)

        self.tree['show']='headings'

        self.tree["columns"] = ("Spc_ID", "Spc_Name",
"Total_Animals", "Healthy_Animals", "Sick_Animals",
"Animals_in_Maternity", "Total_Food_Cost", "Total_Vet_Cost",
"Total_Care_Specialist_Cost")
```

```
        self.tree.heading("Spc_ID", text="Spc_ID")

        self.tree.heading("Spc_Name", text="Spc_Name")

        self.tree.heading("Total_Animals", text="Total_Animals")

        self.tree.heading("Healthy_Animals", text="Healthy_Animals")

        self.tree.heading("Sick_Animals", text="Sick_Animals")

        self.tree.heading("Animals_in_Maternity",
text="Animals_in_Maternity")

        self.tree.heading("Total_Food_Cost", text="Total_Food_Cost")

        self.tree.heading("Total_Vet_Cost", text="Total_Vet_Cost")

        self.tree.heading("Total_Care_Specialist_Cost",
text="Total_Care_Specialist_Cost")

        self.tree.grid(row=1, column=0, columnspan=2,padx=10,
pady=10)


        def on_h(*args):

            self.tree.xview(*args)


        def on_v(*args):

            self.tree.yview(*args)


        vsb_em = ttk.Scrollbar(self.sub_tab_animal_report,
orient="vertical", command=on_v)

        vsb_em.grid(row=1, column=2, pady=10, sticky="ns")

        self.tree.configure(yscrollcommand=vsb_em.set)


        hsb_em = ttk.Scrollbar(self.sub_tab_animal_report,
orient="horizontal", command=on_h)
```

```
        self.tree.configure(xscrollcommand=hsb_em.set)
        hsb_em.grid(row=2, column=0, padx=0, sticky="ew")

        self.sub_tab_animal_report.grid_rowconfigure(0, weight=1)
        self.sub_tab_animal_report.grid_columnconfigure(0, weight=1)



        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)

    def avg(self):
        st_d = self.avg_report_sd_entry.get()
        print(st_d)
        ed_d = self.avg_report_ed_entry.get()
        print(ed_d)
        cursor = connection.cursor()
        query = '''
        SELECT
            rt.REV_NAME AS Revenue_Source,
            AVG(re.revenue) AS Average_Revenue
        FROM
            HP578.revenue_events re
        LEFT JOIN
            HP578.revenue_types rt ON re.revenuetypes_id =
rt.revenuetypes_id
```

```
        WHERE
            re.date_time BETWEEN TO_DATE(:1, 'YYYY-MM-DD') AND
TO_DATE(:2, 'YYYY-MM-DD')
        GROUP BY
            rt.rev_name
        ORDER BY
            AVG(re.revenue) DESC
        '''
        cursor.execute(query, (st_d,ed_d))
        connection.commit()


        self.top_3_report_sd_entry.delete(0, 'end')
        self.top_3_report_ed_entry.delete(0, 'end')


        # Display a message in the label
        message = "Report generated successfully!"
        self.message_label.config(text=message)
        self.tree = ttk.Treeview(self.sub_tab_revenue_time_avg)
        self.tree['show']='headings'
        self.tree["columns"] = ("Revenue_Source", "Average_Revenue")
        self.tree.heading("Revenue_Source", text="Revenue_Source")
        self.tree.heading("Average_Revenue", text="Average_Revenue")
        self.tree.grid(row=4, column=0, columnspan=2,padx=10,
pady=10)
```

```
        for row in cursor.fetchall():
            self.tree.insert("", "end", values=row)




if __name__ == "__main__":
    root = tk.Tk()
    app = zoo(root)
    root.mainloop()
```


Minit
First
Last
JobType
StartDate
Name
Street
City
ID
Address
State
Zip
Supervisor
(0,n)
1
Hourly Rate
1
Earns
N
Employee
Super-
vises
(1,n)
(1,1)
N
ID
Rate
JobType
(1,1)
Supervisee
D
Vet
ACS
Maint
CS
TS
Veterinarian
Animal Care
Specialist
Maintenance
Customer
Service
Ticket Seller
(1,n)
(1,n)
(1,1)
(1,1)
M
M
N
N
Cares
For
Cares
For
Works At
Works At
N
(1,n)
1
1
(1,n)
(1,n)
ID
Status
Birth
Year
Food
Cost
ID
Lives In
N
Animal
N
Member
of
1
Species
Name
(1,1)
(1,1)
(1,n)
1
(0,n)
M
Adult
Price
Senior
Price
Child
Price
Product
Senior
Price
req'd
Participates
In
N
Animal Show
Concession
Zoo Admission
Adult
Price
(1,n)
(1,n)
Child
Price
AS
Conc
ZA
per
day
D
Type
Enclosure
N
Part Of
1
Building
1
Hosts
N
Revenue Types
1
Occurs
N
Revenue
Events
(1,1)
(0,1)
(0,n)
(1,1)
(1,n)
(1,1)
ID
SqFt
ID
Name
Type
ID
Name
Type
Date
Time
Revenue
Tickets
Sold

Relational Schema