DATABASE MANAGEMENT SYSTEMS

# PROJECT REPORT
# ZOO MANAGEMENT SYSTEM

**Risheek Nayak (B20AI058)**

**Suyog Gupta (B20AI059)**

**Shubham Solanki (B20AI040)**

## Introduction

- Our goal is to show an initial overview of how a zoo's data is managed centrally by using a database management system.

- An overview of our project:
    - Animal Details: Proper details about how, when, and from where animals have been imported, their basic type, and their information.
    - Staff Details: Used to store all the valid details of the staff i.e. salary, name, job type, shifts, etc.
    - Zoo Museum: An attraction site for the zoo. It includes different types of precious things from ancient to modern times.
    - Zoo's visitors: Details of the visitors of the zoo.

## Entities

**We have used 11 entities in our database. The detailed explanation of each one of them are given below:**

1. Animal Details:
   - This table contains all the basic information about the individual animal types that are kept in the cages.
   - Attributes: animal_id, cage_no, animal_type, scientific_name, gender, gestation_month, present_health, birth_date, class
   - Primary Key - animal_id
   - Foreign Key - animal_type

2. Animal Habitat:
   - This table keeps track of what time and what food the animals are fed by the name of the staff member. The animals are generally fed thrice a day (morning, noon, and evening).
   - Attributes: animal_type, staff_id, breakfast_timing, breakfast_food, breakfast_stock, lunch_timing, lunch_food, lunch_stock, dinner_timing, dinner_food, dinner_stock
   - Primary key - animal_type
   - Foreign key - a_name

3. Animal Import:
    - The animals are imported from different countries and the data regarding this is recorded such as from which country it is imported, the date of import, cost, and present age during exportation.
    - Attributes: Import_id, animal_id, animal_name, entry_date, import_cost, age_during_export
    - Primary Key - import_id
    - Foreign Key - animal_id

4. Expense:
    - This table tracks the expense of each animal which includes food, medicines, and other additional charges per cage.
    - Attributes: Expense_id, animal_id, food_cost, medicine_cost,cleaning_cost, service_charge
    - Primary key - Expense_id
    - Foreign Key - animal_id

5. Staff Details:
    - This table includes all the information about the employees working in the zoo.
    - This includes the type of job, duration, designation, salary, etc,
    - Attributes: Staff_id,Staff_name , Gender,Job_type,Salary, Email_address, shifting_time, date_of_joining,job_duration.

- Primary Key - Staff_id
- Foreign Key - None

6. Manager Details:
    - Managers are responsible for handling different sections of the zoo and staff.
    - Attributes: Manager_id, Staff_id, Manager_name, gender, job_type, salary, email_address, Managing_Area, Managing_Experience, General_Manager, admin_privilages
    - Primary Key - manager_id
    - Foreign Key - Staff_id

7. Vet Details:
    - The responsibilities of the vets are to monitor the health of the animals and diagnose them when they are sick.
    - Attributes: vet_id, Staff_id, vet_name, head_vet, specialty, veternity_experience
    - Primary Key - vet_id
    - Foreign key - staff_id

8. Visitor Details:
    - The visitors are the main motive behind a zoo. Three categories of tickets are available based on the customer's age. If the visitor wants to visit the museum, he needs an additional museum ticket.

- Attributes: visitor_id, age, ticket_type, zoo_ticket_code, museum_ticket_code, ticket_price, coupon_no
- Primary Key - Visitor_id
- Unique - coupon_no

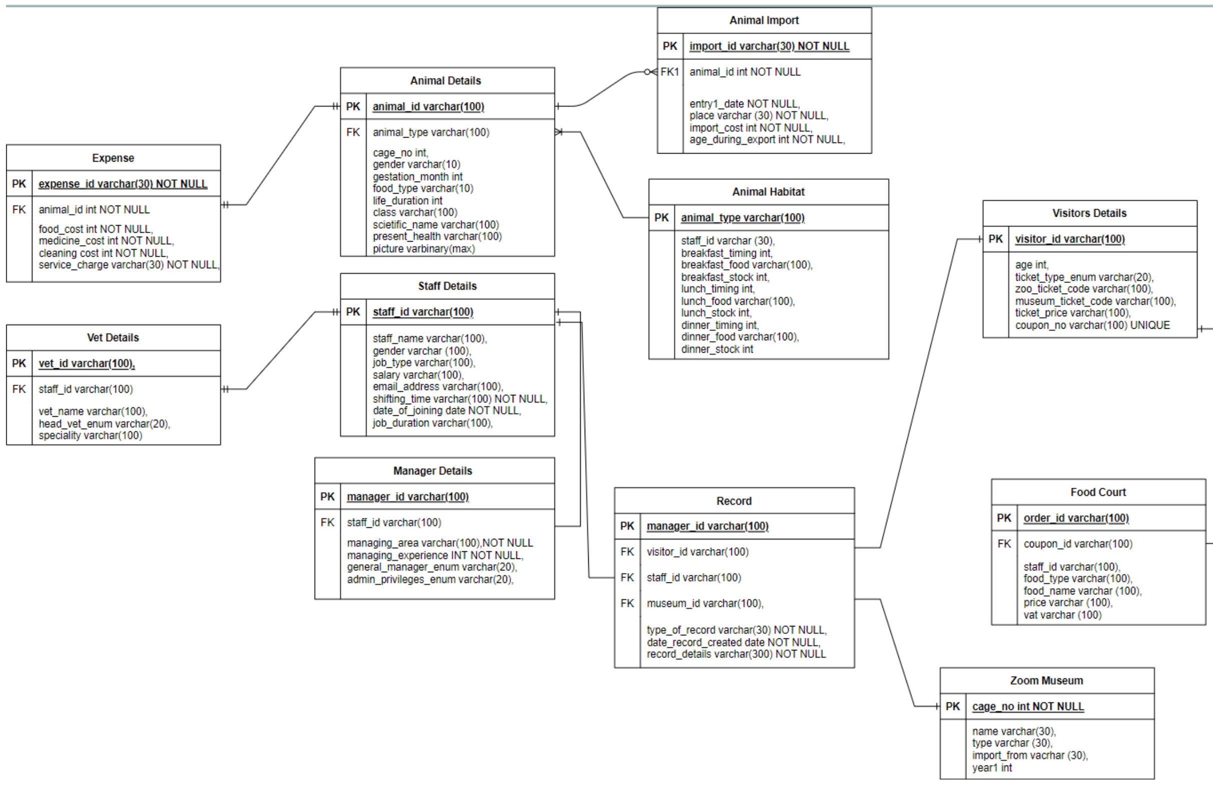
9. Food Court:
    - The zoo also has a food court for visitors. The visitors with a coupon only have access.
    - Attributes: coupon_no, staff_id ,order_id, food_type, food_name, price, vat
    - Primary Key - order_id
    - Foreign Key - coupon_no

10. Zoo Museum:
    - The zoo museum consists of living and non-living animals' information including their import place and year.
    - Attributes: meseum_id, name, Type, import_from, year
    - Primary Key - museum_id
    - Foreign Key - None

11. Record:
    - Keeps track of any unexpected situation occurred in the zoo such as death etc.

- Attributes: record_id, cage_no, staff_id, visitor_id, museum_id, type_of_record, date_record_created, record_details
- Primary Key - record_id
- Foreign key - animal_id, staff_id, visitor_id, museum_id

## **ER Model**



## Trigger:

We have also added a trigger to add to our functionality.

We are using this to increase the count whenever there is a new_entry.

```
CREATE TRIGGER decree

ON Animal_details

FOR INSERT

AS

BEGIN

  update species_count

  set species_count.count1=species_count.count1+1

  where species_count.scintific_name = new.scintific_name;

END;
```

---

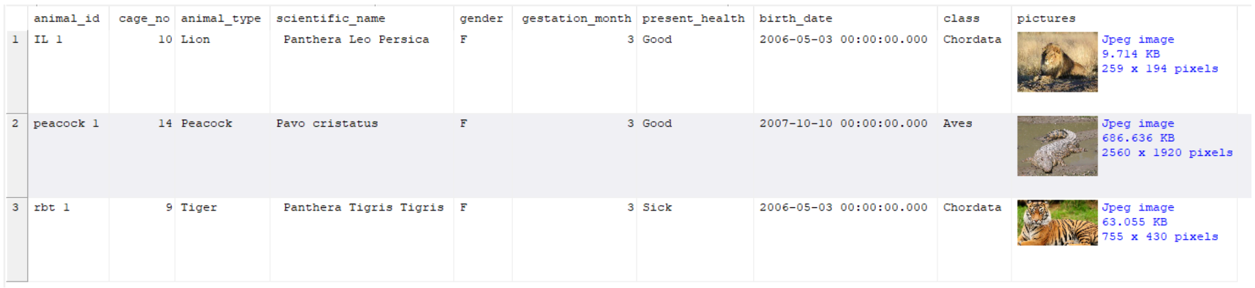
## SQL Queries

1. **Using WHERE Clause**

   Animals with Gestation Month less than 5 years select *

from Animal_details

where gestation_month < 5;



| | animal_id | cage_no | animal_type | scientific_name | gender | gestation_month | present_health | birth_date | class | pictures |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | IL 1 | 10 | Lion | Panthera Leo Persica | F | 3 | Good | 2006-05-03 00:00:00.000 | Chordata | Jpeg image 9.714 KB 259 x 194 pixels |
| 2 | peacock 1 | 14 | Peacock | Pavo cristatus | F | 3 | Good | 2007-10-10 00:00:00.000 | Aves | Jpeg image 686.636 KB 2560 x 1920 pixels |
| 3 | rbt 1 | 9 | Tiger | Panthera Tigris Tigris | F | 3 | Sick | 2006-05-03 00:00:00.000 | Chordata | Jpeg image 63.055 KB 755 x 430 pixels |

**<u>Relational Algebra:</u> $\pi(\sigma_{gestation\_month>5}$(animal details))**

## 2. Using BETWEEN Clause

visitor's age between 3 and 10.

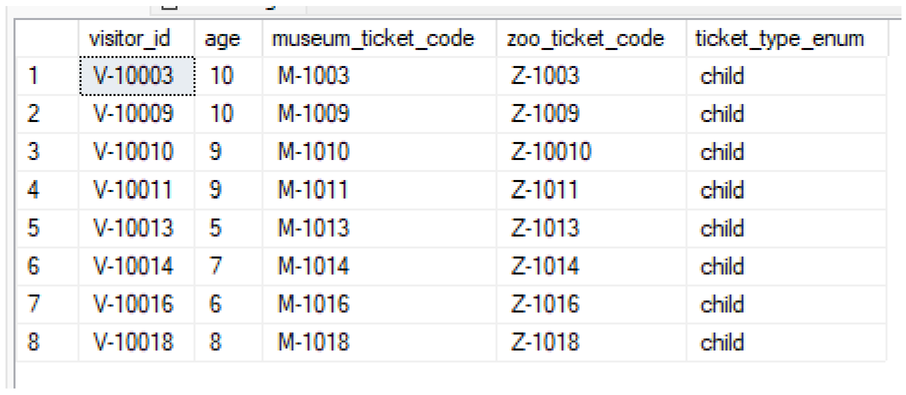
SELECT visitor_id, age, museum_ticket_code, zoo_ticket_code, ticket_type_enum

FROM visitor_details

where age BETWEEN 3 and 10;

**<u>Relational Algebra:</u>**

**$\pi_{vistor\_id,age,mueseum\_ticket\_code,zoo\_ticket\_code,ticker\_type\_enum}(\sigma_{age>=3 \text{ and } age=<10}$(animal details))**



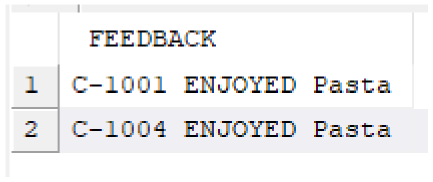
| | visitor_id | age | museum_ticket_code | zoo_ticket_code | ticket_type_enum |
|---|---|---|---|---|---|
| 1 | V-10003 | 10 | M-1003 | Z-1003 | child |
| 2 | V-10009 | 10 | M-1009 | Z-1009 | child |
| 3 | V-10010 | 9 | M-1010 | Z-10010 | child |
| 4 | V-10011 | 9 | M-1011 | Z-1011 | child |
| 5 | V-10013 | 5 | M-1013 | Z-1013 | child |
| 6 | V-10014 | 7 | M-1014 | Z-1014 | child |
| 7 | V-10016 | 6 | M-1016 | Z-1016 | child |
| 8 | V-10018 | 8 | M-1018 | Z-1018 | child |

## 3. Using CONCAT Clause

Feedback of the customer who had pasta

SELECT CONCAT(coupon_no ,' ENJOYED ',food_name) AS ' FEEDBACK '

FROM food_court WHERE food_name='Pasta';



| | FEEDBACK |
|---|---|
| 1 | C-1001 ENJOYED Pasta |
| 2 | C-1004 ENJOYED Pasta |

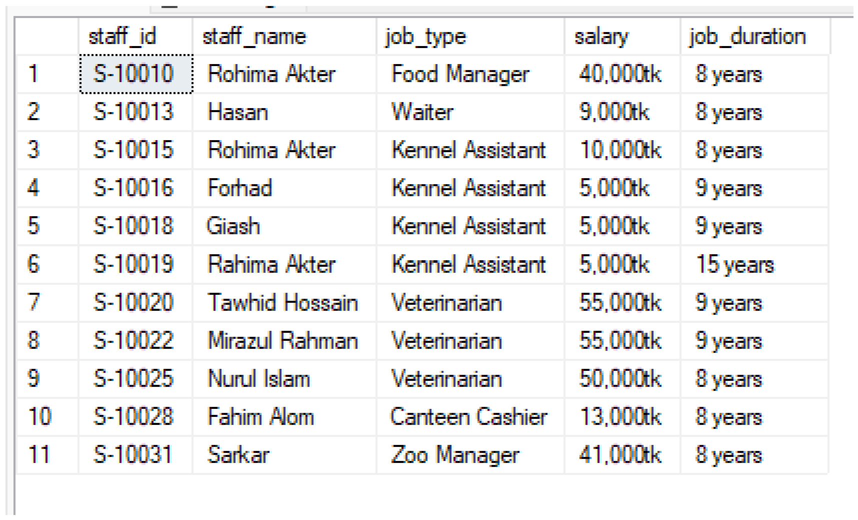
## 4. Using IN/NOT IN Clause

Show the details of the staff whose job duration is  8,9 &15 years

SELECT staff_id,staff_name,job_type,salary,job_duration

FROM staff_details

WHERE job_duration IN ('9 years','15 years','8 years');



| | staff_id | staff_name | job_type | salary | job_duration |
|---|---|---|---|---|---|
| 1 | S-10010 | Rohima Akter | Food Manager | 40,000tk | 8 years |
| 2 | S-10013 | Hasan | Waiter | 9,000tk | 8 years |
| 3 | S-10015 | Rohima Akter | Kennel Assistant | 10,000tk | 8 years |
| 4 | S-10016 | Forhad | Kennel Assistant | 5,000tk | 9 years |
| 5 | S-10018 | Giash | Kennel Assistant | 5,000tk | 9 years |
| 6 | S-10019 | Rahima Akter | Kennel Assistant | 5,000tk | 15 years |
| 7 | S-10020 | Tawhid Hossain | Veterinarian | 55,000tk | 9 years |
| 8 | S-10022 | Mirazul Rahman | Veterinarian | 55,000tk | 9 years |
| 9 | S-10025 | Nurul Islam | Veterinarian | 50,000tk | 8 years |
| 10 | S-10028 | Fahim Alom | Canteen Cashier | 13,000tk | 8 years |
| 11 | S-10031 | Sarkar | Zoo Manager | 41,000tk | 8 years |

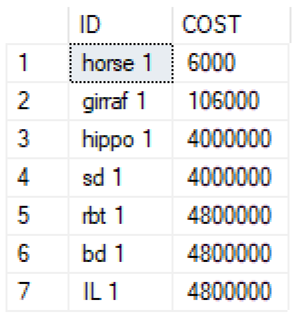
**Relational Algebra:**

**$\pi_{staff_id,staff_name,job_type_salary_job_duration}(\sigma_{job_duration='8'and'9'and'15'}($animal details$))$**

## 5. Using ORDER BY Clause

Import Cost in Ascending Order

```
SELECT animal_id 'ID', IMPORT_COST 'COST' FROM animal_import
ORDER BY COST ASC;
```



| | ID | COST |
|---|---|---|
| 1 | horse 1 | 6000 |
| 2 | girraf 1 | 106000 |
| 3 | hippo 1 | 4000000 |
| 4 | sd 1 | 4000000 |
| 5 | rbt 1 | 4800000 |
| 6 | bd 1 | 4800000 |
| 7 | IL 1 | 4800000 |

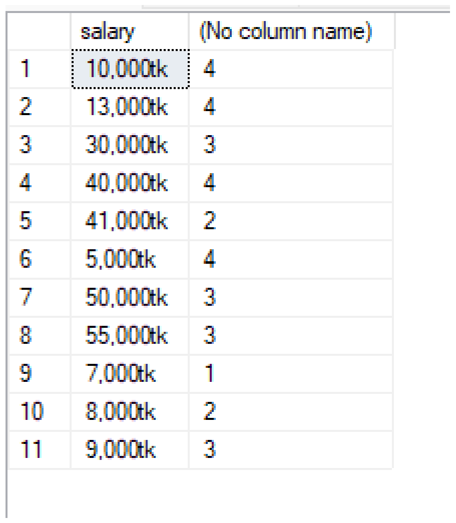
**Relational Algebra:** **$\pi_{id,cost}(\gamma_{orderby(cost)}($animal_import$))$**

## 6. Using GROUP BY and HAVING Clause

Group the staffs on the basis of their salary

```
SELECT salary, count(salary)

FROM staff_details GROUP BY salary;
```



|  | salary | (No column name) |
|---|---|---|
| 1 | 10,000tk | 4 |
| 2 | 13,000tk | 4 |
| 3 | 30,000tk | 3 |
| 4 | 40,000tk | 4 |
| 5 | 41,000tk | 2 |
| 6 | 5,000tk | 4 |
| 7 | 50,000tk | 3 |
| 8 | 55,000tk | 3 |
| 9 | 7,000tk | 1 |
| 10 | 8,000tk | 2 |
| 11 | 9,000tk | 3 |

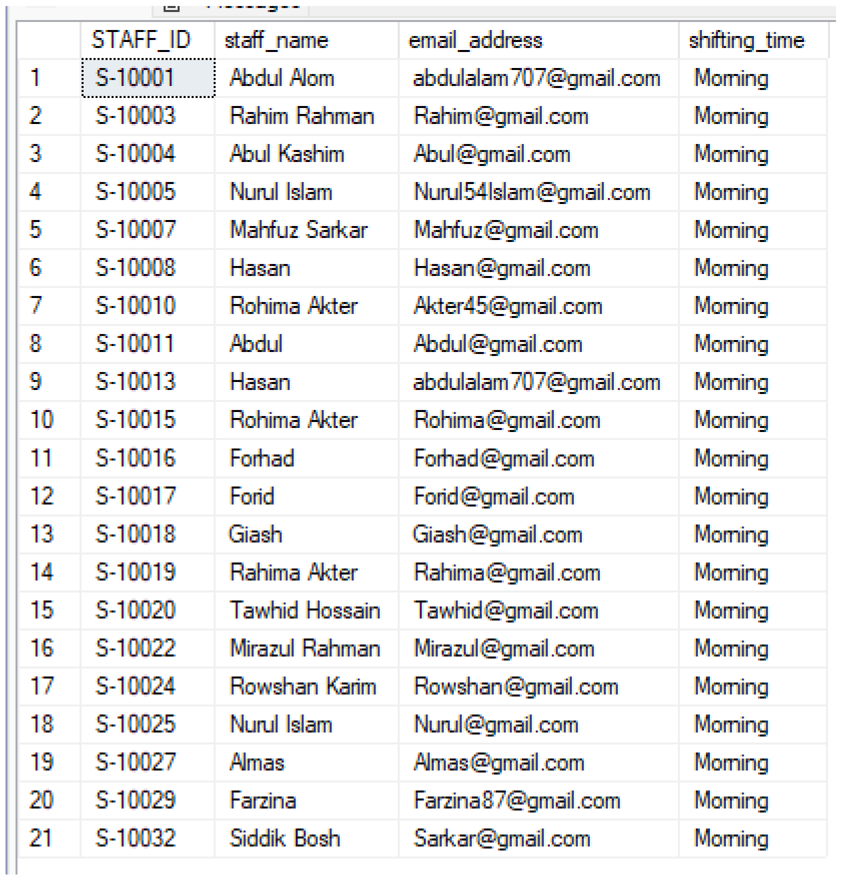
**<u>Relational Algebra:</u> $\pi_{salary,count(salary)}(\gamma_{groupby(salary)}(staff\_details))$**

## 7. Using LIKE Clause

Staff who had morning shift

```
SELECT STAFF_ID,staff_name,email_address,shifting_time

FROM staff_details WHERE shifting_time like '%MORNING%';
```



| | STAFF_ID | staff_name | email_address | shifting_time |
|---|---|---|---|---|
| 1 | S-10001 | Abdul Alom | abdulalam707@gmail.com | Morning |
| 2 | S-10003 | Rahim Rahman | Rahim@gmail.com | Morning |
| 3 | S-10004 | Abul Kashim | Abul@gmail.com | Morning |
| 4 | S-10005 | Nurul Islam | Nurul54Islam@gmail.com | Morning |
| 5 | S-10007 | Mahfuz Sarkar | Mahfuz@gmail.com | Morning |
| 6 | S-10008 | Hasan | Hasan@gmail.com | Morning |
| 7 | S-10010 | Rohima Akter | Akter45@gmail.com | Morning |
| 8 | S-10011 | Abdul | Abdul@gmail.com | Morning |
| 9 | S-10013 | Hasan | abdulalam707@gmail.com | Morning |
| 10 | S-10015 | Rohima Akter | Rohima@gmail.com | Morning |
| 11 | S-10016 | Forhad | Forhad@gmail.com | Morning |
| 12 | S-10017 | Forid | Forid@gmail.com | Morning |
| 13 | S-10018 | Giash | Giash@gmail.com | Morning |
| 14 | S-10019 | Rahima Akter | Rahima@gmail.com | Morning |
| 15 | S-10020 | Tawhid Hossain | Tawhid@gmail.com | Morning |
| 16 | S-10022 | Mirazul Rahman | Mirazul@gmail.com | Morning |
| 17 | S-10024 | Rowshan Karim | Rowshan@gmail.com | Morning |
| 18 | S-10025 | Nurul Islam | Nurul@gmail.com | Morning |
| 19 | S-10027 | Almas | Almas@gmail.com | Morning |
| 20 | S-10029 | Farzina | Farzina87@gmail.com | Morning |
| 21 | S-10032 | Siddik Bosh | Sarkar@gmail.com | Morning |

## 8. Using JOIN Clause

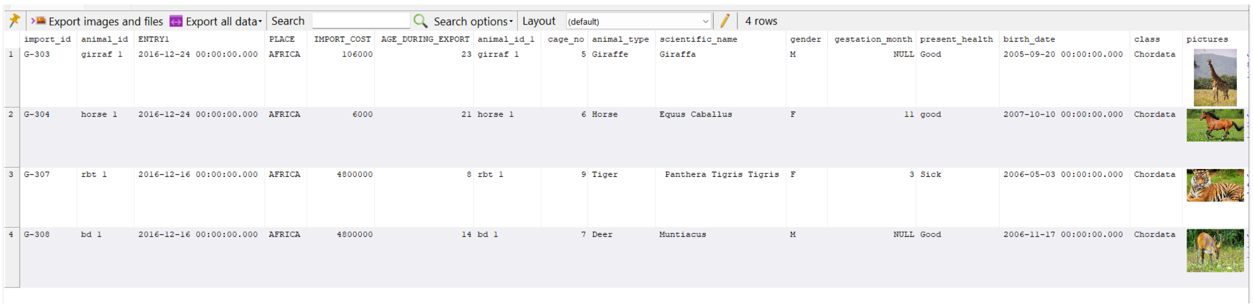
Show details of animals imported from africa.

```
SELECT *

FROM animal_import i JOIN Animal_details l

ON (i.animal_id = l.animal_id)

where i.PLACE = 'AFRICA' ;
```



Export images and files | Export all data | Search | Search options | Layout (default) | 4 rows

|  | import_id | animal_id | ENTRY1 | PLACE | IMPORT_COST | AGE_DURING_EXPORT | animal_id_1 | cage_no | animal_type | scientific_name | gender | gestation_month | present_health | birth_date | class | pictures |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | G-303 | girraf 1 | 2016-12-24 00:00:00.000 | AFRICA | 106000 | 23 | girraf 1 | 5 | Giraffe | Giraffa | M | NULL | Good | 2005-09-20 00:00:00.000 | Chordata | |
| 2 | G-304 | horse 1 | 2016-12-24 00:00:00.000 | AFRICA | 6000 | 21 | horse 1 | 6 | Horse | Equus Caballus | F | 11 | good | 2007-10-10 00:00:00.000 | Chordata | |
| 3 | G-307 | rbt 1 | 2016-12-16 00:00:00.000 | AFRICA | 4800000 | 8 | rbt 1 | 9 | Tiger | Panthera Tigris Tigris | F | 3 | Sick | 2006-05-03 00:00:00.000 | Chordata | |
| 4 | G-308 | bd 1 | 2016-12-16 00:00:00.000 | AFRICA | 4800000 | 14 | bd 1 | 7 | Deer | Muntiacus | M | NULL | Good | 2006-11-17 00:00:00.000 | Chordata | |

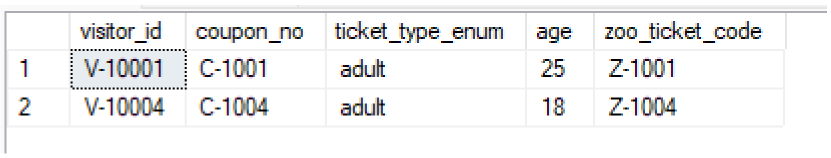
**<u>Relational Algebra:</u>**

$\pi(\sigma_{animal\_import.place='africa'}(animal\_import \bowtie animal\_details))$

## 9. NESTED SUBQUERIES

Visitors who had a coupon and had pasta

```
SELECT visitor_details.visitor_id, visitor_details.coupon_no,
visitor_details.ticket_type_enum,visitor_details.age,visitor_details.zoo_ticket_code
FROM visitor_details
WHERE visitor_details.coupon_no
IN(select food_court.coupon_no from food_court where food_name ='Pasta');
```

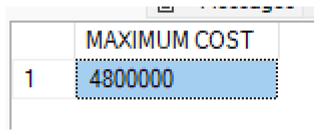


|  | visitor_id | coupon_no | ticket_type_enum | age | zoo_ticket_code |
|---|---|---|---|---|---|
| 1 | V-10001 | C-1001 | adult | 25 | Z-1001 |
| 2 | V-10004 | C-1004 | adult | 18 | Z-1004 |

## 10. AGGREGATE FUNCTIONS

Show maximum import cost

SELECT MAX(IMPORT_COST) 'MAXIMUM COST' FROM animal_import;



```
   MAXIMUM COST
1  4800000
```

**<u>Relational Algebra:</u> $\pi_{max(import\_cost)}$(animal_cost)**

## Contributions:

| **Group Member** | **Contributions** |
|---|---|
| Risheek Nayak (B20AI058) | 4 entity design (), SQL queries, Relational Algebra, Presentation Slides, report, |
| Suyog Gupta (B20AI059) | 4 entity design (), SQL queries, Relational Algebra, Presentation Slides, report, |
| Shubham Solanki (B20AI040) | 3 entity design (), SQL queries, Relational Algebra, Presentation Slides, images insertion., |