BIODIVERSITY PARK SYSTEM

DBMS Project

Course Instructor: Dr.Debanjan Sadhya

Atal Bihari Vajpayee
Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior
(An Institute of National Importance, Ministry of Education, Government of India)

BIODIVERSITY PARK SYSTEM

Group Members

• Avantsa Raja Yashwanth	2021BCS-016
• Mekala Bhavana	2021BCS-041
• Muskan Debnath	2021BCS-046
• Narra Abhigna	2021BCS-048
• Nelluri Pavithra Sai Lakshmi	2021BCS-049
• Ravi Jwalana	2021BCS-056

Table of contents

Overview

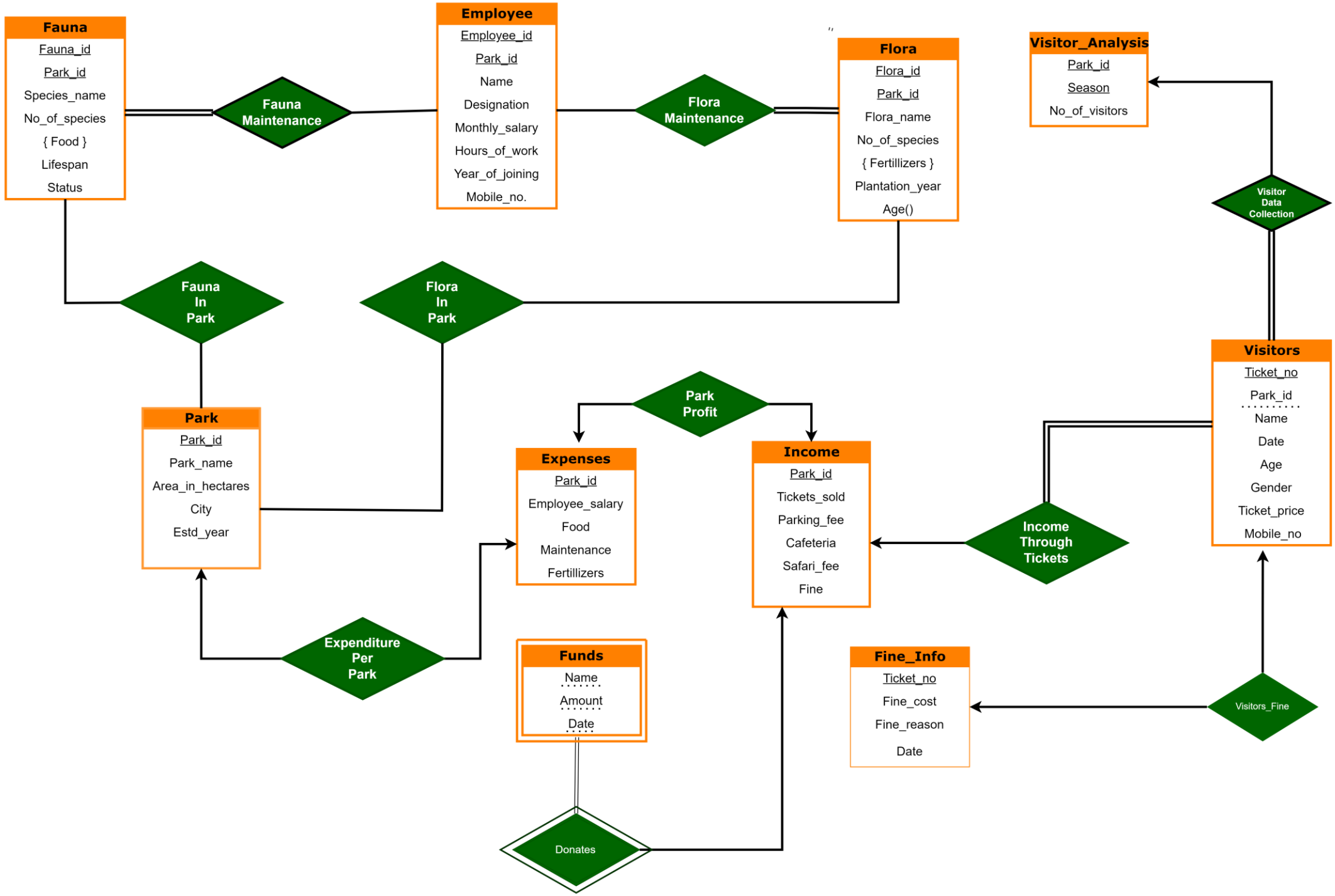
- Biodiversity park is a place where all the different kinds of life can be found in one area. It is an enrichment to our society as each of the organisms work together in the ecosystem to sustain balance of nature.
- But nowadays, many of the helpless beings are on the verge of extinction due to lack of healthy environment to live in.
- Hence, we have come up with the concept of aiding the endangered species by providing them with proper habitats via a database system.

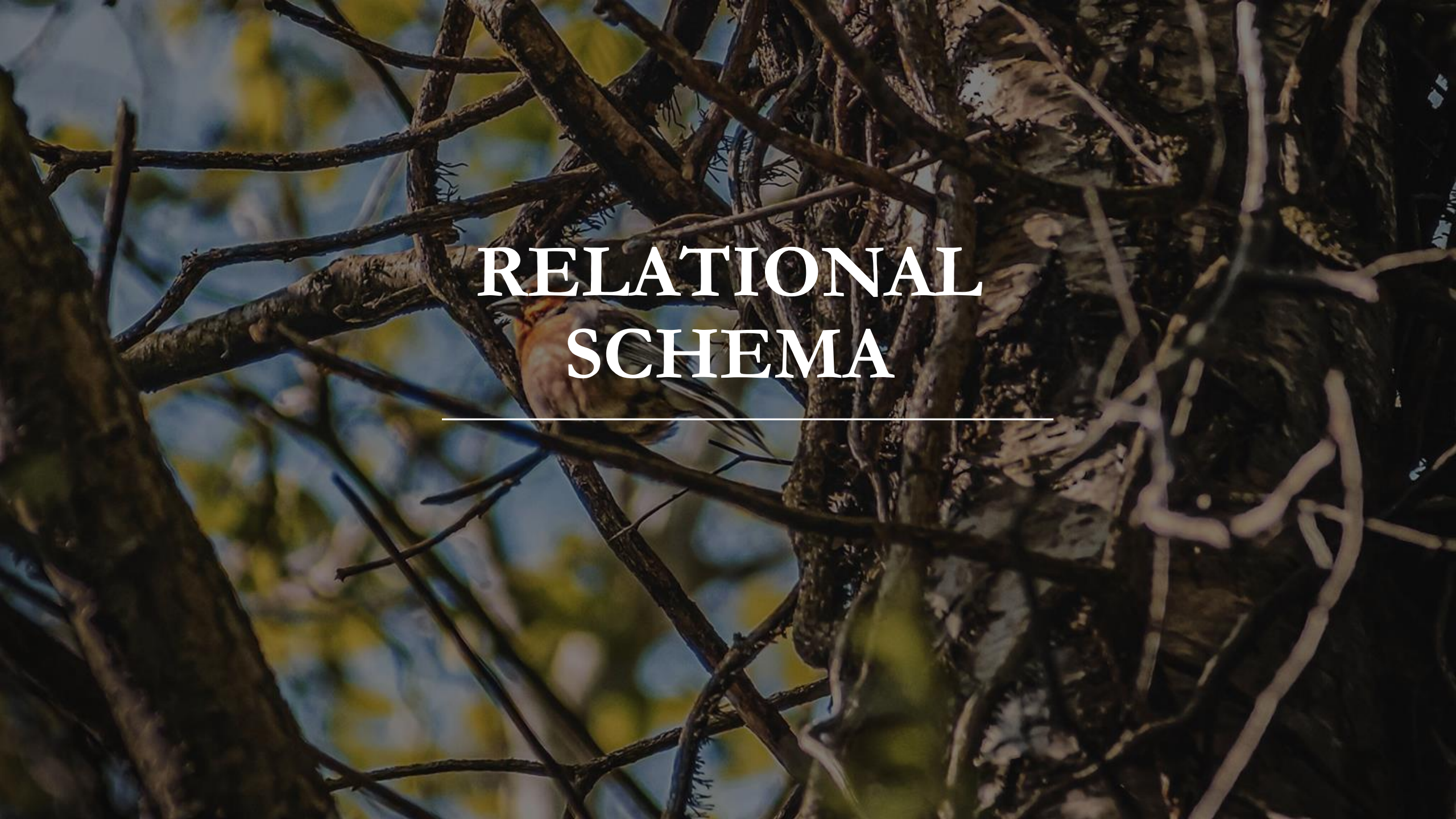
ENTITIES AND ATTRIBUTES

ENTITIES	ATTRIBUTES
Park	**Park_id**,Park_name, Area_in_hectares,City,Estd_year
Fauna	**Fauna_id**, **Park_id**,Species_name, No_of_species, Food, Lifespan, Status
Flora	**Flora_id**,**Park_id**,Flora_name, No_of_species,Fertilizers, Plantation_year,Age
Employee	**Employee_id, Park_id**,Name,Designation, Monthly_salary, Hours_of_work, Year_of_joining,Mobile_no
Visitors	**Ticket_no**,**Park_id**,Date,Name,Age,Gender,Ticket_price,Mobile_no
Expenses	**Park_id**,Employee_salary, Food, Fertilizers,Maintenance
Income	**Park_id**,Tickets_sold,Parking_fee,Cafeteria,Safari_fee,Fine
Fine_info	**Ticket_no**,Fine_reason,Fine_Cost,Date
Visitor_analysis	**Park_id**,**Season**,No_of_visitors
Funds	Name,Amount,Date

RELATION-SHIP	ENTITIES INVOLVED	CARDINALITY
Fauna Maintenance	Fauna , Employee	Many to Many, Total on Fauna Side
Flora Maintenance	Flora, Employee	Many to Many, Total on Flora Side
Visitor Data Collection	Visitor_Analysis,Visitors	Many to One, Total on Visitors side
Flora in Park	Flora, Park	Many to Many
Fauna in Park	Fauna, Park	Many to Many
Park Profit	Expenses, Income	One to One
Expenditure per Park	Park, Expenses	One to One
Income Through Tickets	Visitors,Income	Many to One, Total on visitors side
Visitors_Fine	Fine_Info, Visitors	One to one
Donates	Funds,Income	Many to One, Total on Funds Side


ER
DIAGRAM


Fauna
Fauna_id
Park_id
Species_name
No_of_species
{ Food }
Lifespan
Status
Fauna Maintenance
Employee
Employee_id
Park_id
Name
Designation
Monthly_salary
Hours_of_work
Year_of_joining
Mobile_no.
Flora Maintenance
Flora
Flora_id
Park_id
Flora_name
No_of_species
{ Fertillizers }
Plantation_year
Age()
Visitor_Analysis
Park_id
Season
No_of_visitors
Visitor Data Collection
Fauna In Park
Flora In Park
Park
Park_id
Park_name
Area_in_hectares
City
Estd_year
Park Profit
Expenses
Park_id
Employee_salary
Food
Maintenance
Fertillizers
Income
Park_id
Tickets_sold
Parking_fee
Cafeteria
Safari_fee
Fine
Visitors
Ticket_no
Park_id
Name
Date
Age
Gender
Ticket_price
Mobile_no
Income Through Tickets
Expenditure Per Park
Funds
Name
Amount
Date
Fine_Info
Ticket_no
Fine_cost
Fine_reason
Date
Visitors_Fine
Donates



RELATIONAL SCHEMA

Park	
Park_id(Primary)	Char(6)
Park_name	Varchar(45)
Area_in_hectares	Int
City	Varchar(45)
Estd_year	int

Employee	
Employee_id(Primary)	char(10)
Park_id(Primary)	char(6)
Name	varchar(45)
Designation	varchar(45)
Monthly_salary	numeric(5,0)
Hours_of_work	int
Year_of_joining	int
Mobile_no	numeric(10,0)

Fauna	
Fauna_id(Primary)	Char(5)
Park_id(Primary)	Char(6)
Species_name	Varchar(45)
No_of_species	Int
Lifespan	Varchar(45)
Status	Varchar(20)

Income	
Park_id(Primary)	Char(6)
Tickets_sold	Numeric(7,0)
Parking_fee	Numeric(6,0)
Cafeteria	Numeric(6,0)
Safari_fee	Numeric(6,0)
Fine	Numeric(6,0)

Flora_fertilizer	
Flora_id(Primary)	Char(5)
Park_id(primary)	Char(6)
Fertilizers(Primary)	Varchar(45)

Visitors	
Ticket_no(Primary)	Varchar(25)
Park_id(Foreign)	Char(6)
Date	Char()
Name	Varchar(45)
Age	Int
Gender	Char(6)
Ticket_price	Numeric(4,0)
Mobile_no	Numeric(10,0)

Flora	
Flora_id(Primary)	Char(5)
Park_id(Primary)	char(6)
Flora_name	Varchar(45)
No_of_species	Int
Plantation_year	Int
Age	Int

Fauna_food	
Fauna_id(Primary)	Char(5)
Park_id(primary)	Char(6)
Food(Primary)	Varchar(45)

Expenses	
Employee_salary	numeric(6,0)
Food	numeric(6,0)
Fertilizers	numeric(6,0)
Maintainence	numeric(6,0)
Park_id(Primary)	char(6)

Visitor analysis	
Park_id(Primary)	char(6)
Season(Primary)	varchar(45)
No_of_visitors	int

Fine_info	
Ticket_no(Primary)	varChar(20)
Fine_reason	Varchar(100)
Fine_cost	Int
Date	Char(5)

Funds	
Park_id(Primary)	char(6)
Name(Primary)	Varchar(35)
Amount(primary)	Numeric(6,0)
Date(Primary)	Char(5)

Relationship Set Schemas
Many to many

Fauna_maintenance(Fauna , Employee)	
Fauna_id(primary)	Char(5)
Park_id(primary)	Char(6)
Employee_id(primary)	Char(10)

Flora_maintenance(Flora, Employee)	
Flora_id(primary)	Char(5)
Park_id(primary)	Char(6)
Employee_id(primary)	Char(10)

Fauna_in_park(Fauna, Park)	
Park_id(primary)	Char(6)
Fauna_id(primary)	Char(5)

Flora_in_park(Flora, Park)	
Park_id(primary)	Char(6)
Flora_id(primary)	Char(5)

One to one

Park(Expenditure per Park) (Park, Expenses)
Park_id(primary)
Park_name
Area_in_hectares
City
Estd_year

Expenses(Park Profit)(Expenses, Income)
Employee_salary
Food
Fertilizers
Maintainence
Park_id(Primary)

Fine_info(Visitors_Fine)(Fine_Info, Visitors)
Ticket_no(Primary)
Fine_reason
Fine_cost
Date

Many to one

Visitor analysis(Visitor Data Collection)(Visitor_Analysis,Visitors)
Park_id(Primary)
Season(Primary)
No_of_visitors
Ticket_no

Visitors(Income Through Tickets)(Visitors,Income)
Ticket_no(Primary)
Park_id(Foreign)
Date
Name
Age
Gender
Ticket_price
Mobile_no

Funds(Donates)(Funds,Income)
Park_id(Primary)
Name(Primary)
Amount(primary)
Date(Primary)

FUNCTIONAL DEPENDENCIES

R1.**Park(Park_id,Park_name,Area_in_hectares,City,Estd_year)**

Functional Dependencies :
Park_id -> Park_name,Area_in_hectares,City,Estd_year

Assumptions :
All attributes are atomic
Two park names can be same
There can be two parks in same City, ocupied in same area or Year established.

Park
Park_id
Park_name
Area_in_hectares
City
Estd_year

R2.**Fauna(Fauna_id,Park_id,Species_name,No_of_species,Lifespan,Status)**

Functional Dependencies :
Fauna_id, Park_id -> Species_name, No_of_species, Lifespan, Status
Fauna_id -> Species_name, Lifespan, Status
Species_name -> Lifespan, Status

Assumptions :
All attributes are atomic.
No two species have same name i.e,(species_name is unique)
Fauna_id implies Species_name but Species_name cant imply Fauna_id

Fauna
Fauna_id
Park_id
Species_name
No_of_species
Lifespan
Status

R3.**Flora(Flora_id,Park_id,Flora_name,No_of_species,Plantation_year,Age)**

Functional Dependencies :

Flora_id,Park_id -> Flora_name,No_of_species,Plantation_year,Age

Flora_id ->Flora_name

Assumptions :

All attributes are atomic.

Flora_id implies Species_name but Species_name cant imply Flora_id

Flora
Flora_id
Park_id
Flora_name
No_of_species
Plantation_year
Age( )

R4.**Employee(Employee_id, Park_id,Name,Designation, Monthly_salary, Hours_of_work,Year_of_joining ,Mobile_no)**

Functional Dependencies :

Employee_id,Park_id ->Name, Designation, Monthly_salary, Hours_of_works, Year_of_joining, Mobile_no.

Assumptions :

All attributes are atomic. , Employee.Name is a simple attribute , Mobile_no is single valued attribute.

Two employees from different parks can have same employee_id.

Employee
Employee_id
Park_id
Name
Designation
Monthly_salary
Hours_of_work
Year_of_joining
Mobile_no

R5.**Visitors(Ticket_no,Park_id, Date, Name, Age, Gender, Ticket_price, Mobile_no)**

Functional Dependencies :

Ticket_no. ->Park_id, Date, Name, Age, Gender, Ticket_price, Mobile_no.

Age -> Ticket_price

Assumptions :

All attributes are atomic. Name is simple attribute. Mobile_no is single valued attribute.

Ticket_no attribute includes date and park id so, no two park can have same Ticket_no.

In all parks ticket price is same and is based on Age.

Visitors
Ticket_no
Park_id
Date
Name
Age
Gender
Ticket_price
Mobile_no

R6.**Expenses(Park_id,Employee_salary, Food, Fertilizers,Maintenance)**

Functional Dependencies :

Park_id ->Employee_salary, Food, Fertilizers,Maintenance

Assumptions :

All attributes are atomic.

Expenses
Park_id
Employee_salary
Food
Fertilizers
Maintenance

R7. **Income(<u>Park_id</u>,Tickets_sold,Parking_fee,Cafeteria,Safari_fee,Fine)**

Functional Dependencies :

Park_id -> Tickets_sold,Parking_fee,Cafeteria,Safari_fee,Fine

Assumptions :

All attributes are atomic.

Income
<u>Park_id</u>
Tickets_sold
Parking_fee
Cafeteria
Safari_fee
Fine

R8. **Fine_info(<u>Ticket_no</u>,Fine_reason,Fine_Cost,Date)**

Functional Dependencies :

Ticket_no->Fine_reason,Fine_Cost,Date

Assumptions :

All attributes are atomic.

Ticket_no attribute includes date and park id so, no two parks can have same Ticket_no.

In different parks Fine_cost may be different for same Fine_reason

One Ticket_no (Visitor) can be fined atmost once

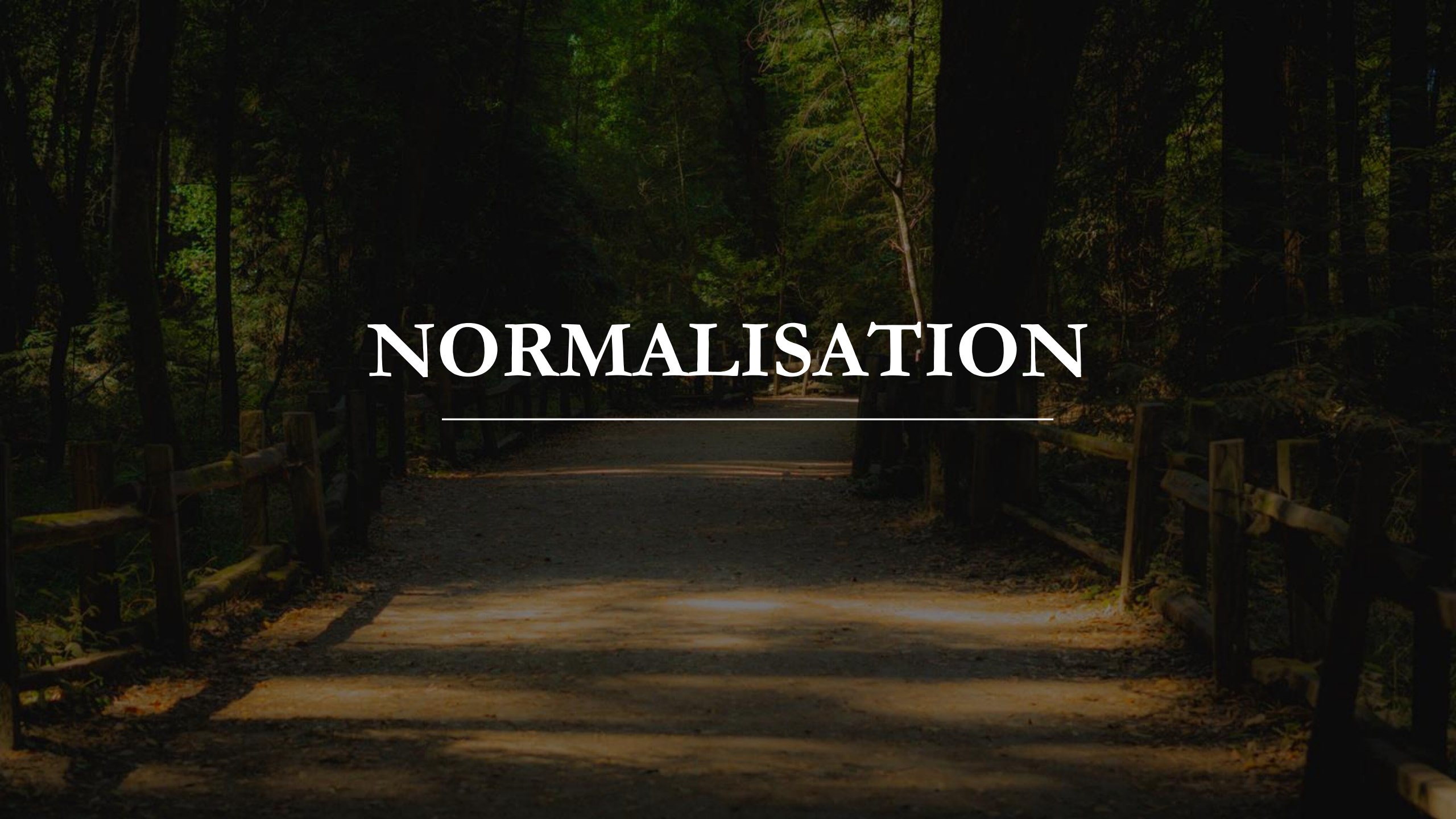
Fine_info
<u>Ticket_no</u>
Fine_reason
Fine_cost
Date

R9. **Visitor_analysis(<u>Park_id,Season</u>,No_of_visitors)**

Functional Dependencies :
Park_id,Season -> No_of_visitors

Assumptions :
All attributes are atomic.

Visitor analysis
<u>Park_id</u>
<u>Season</u>
No_of_visitors



NORMALISATION

R1.**Park(<u>Park_id</u>, Park_name, Area_in_hectares, City, Estd_year)**

Primary key: Park_id

Functional Dependencies :

Park_id -> Park_name,Area_in_hectares,City,Estd_year

Normalisation:

Park entity is in **1NF** as we assume all the attributes are atomic.
There is no partial dependencies, so it is in **2NF**.
There are no Transitive dependencies also, so it in **3NF**.
This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R2.**Fauna(Fauna_id,Park_id,Species_name,No_of_species,Lifespan,Status)**

Primary key: Fauna_id, Park_id

Functional Dependencies :

Fauna_id, Park_id -> Species_name, No_of_species, Lifespan, Status
Fauna_id -> Species_name, Lifespan, Status
Species_name -> Lifespan, Status

Normalisation:

Fauna entity is in **1NF** as we assume all the attributes are atomic.
There is partial dependencies, so it is **not in 2NF**.
So, after doing normalization to 2NF:
R2.1 Fauna_1(Fauna_id, park_id, No_of_species)
R2.2 Fauna_2(Fauna_id, Species_name, Lifespan, Status)

R2.1 **Fauna_1(Fauna_id, park_id, No_of_species)**
Functional Dependencies :
Fauna_id, Park_id -> No_of_species
There are no transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X ->Y, X is the super key of the entity.

R2.2 **Fauna_2(Fauna_id, Species_name, Lifespan, Status)**
Functional Dependencies :
Fauna_id -> Species_name, Lifespan, Status
Species_name -> Lifespan, Status

In R2.2 there is transitive dependency as
Fauna_id -> Species_name
Species_name-> Lifespan,Status
Fauna_id-> Lifespan,Status

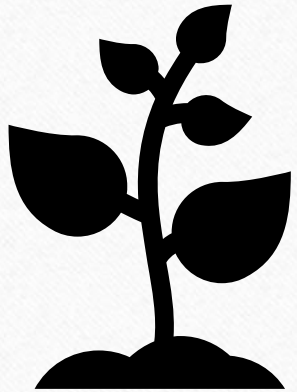
So, after doing normalization to 3NF: R2.2.1 Fauna_2_1(Fauna_id, Species_name)
R2.2.2 Fauna_2_2(Species_name, Lifespan, Status)

There are no transitive dependencies also, so it in **3NF** .
These two tables are in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R3. **Flora(Flora_id, Park_id, Flora_name, No_of_species, Plantation_year, Age)**

Primary key: Flora_id, Park_id
Functional Dependencies :
Flora_id,Park_id -> Flora_name, No_of_species, Plantation_year, Age
Flora_id ->Flora_name

Normalisation:
Fauna entity is in **1NF** as we assume all the attributes are atomic.
There is partial dependencies, so it is **not in 2NF**.
So, after doing normalization to 2NF:
R3.1 Flora_1(Flora_id, Park_id, No_of_species, Plantation_year, Age)
R3.2 Flora_2(Flora_id, Flora_name)
R3.1 Flora_1(Flora_id, Park_id, No_of_species, Plantation_year,Age)



Functional Dependencies :
Flora_id,Park_id -> No_of_species, Plantation_year, Age

There are no transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R3.2 **Flora_2**(Flora_id, Flora_name)
Functional Dependencies :
Flora_id-> Flora_name

There are no transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R4.**Employee(Employee_id, Park_id, Name, Designation, Monthly_salary, Hours_of_work, Year_of_joining , Mobile_no)**

Primary key: Employee_id, Park_id

Functional Dependencies :

Employee_id,Park_id -> Name, Designation, Monthly_salary, Hours_of_works, Year_of_joining, Mobile_no.

Normalisation:

Employee entity is in **1NF** as we assume all the attributes are atomic.
There is no partial dependencies, so it is in **2NF**.
There are no transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R5. **Visitors(Ticket_no, Park_id, Date, Name, Age, Gender, Ticket_price, Mobile_no)**

Primary key: Ticket_no

Functional Dependencies :

Ticket_no ->Park_id, Date, Name, Age, Gender, Ticket_price, Mobile_no

Age -> Ticket_price

Normalisation:

Visitors entity is in **1NF** as we assume all the attributes are atomic.

There is no partial dependencies, so it is in **2NF**.

There is a transitive dependency as so this is **not in 3NF**

Ticket_no-> Age

Age-> Ticket_price

Ticket_no->Ticket_price

So, after doing normalization to 3NF: R5.1 Visitors_1(Ticket_no ,Park_id, Date, Name, Age, Gender, Mobile_no)

R5.2 Visitors_2(Age , Ticket_price)

R5.1 **Visitors_1(Ticket_no ,Park_id, Date, Name, Age, Gender, Mobile_no)**

Functional Dependency:

Ticket_no->Park_id, Date, Name, Age, Gender, Mobile_no

There are no Transitive dependencies also, so it in **3NF** .

This table is in **BCNF** also as for every functional dependency X ->Y, X is the super key of the entity

R5.2 **Visitors_2(Age , Ticket_price)**

Functional Dependency:

Age->Ticket_price

There are no Transitive dependencies also, so it in **3NF** .

This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity

R6.**Expenses(<u>Park_id</u>, Employee_salary, Food, Fertilizers, Maintenance)**

Primary key : Park_id

Functional Dependencies :

Park_id ->Employee_salary, Food, Fertilizers,Maintenance

Normalisation:

Expenses entity is in **1NF** as we assume all the attributes are atomic.
There is no partial dependencies, so it is in **2NF**.
There are no Transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X ->Y, X is the super key of the entity.

R7. **Income(Park_id, Tickets_sold, Parking_fee, Cafeteria, Safari_fee, Fine)**

Primary key: Park_id

Functional Dependencies :
Park_id -> Tickets_sold, Parking_fee, Cafeteria, Safari_fee, Fine

Normalisation:

Income entity is in **1NF** as we assume all the attributes are atomic.
There is no partial dependencies, so it is in **2NF**.
There are no Transitive dependencies also, so it in **3NF** .
This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R8. **Fine_info(Ticket_no, Fine_reason, Fine_Cost, Date)**

Primary key: Ticket_no

Functional Dependencies :

Ticket_no->Fine_reason, Fine_Cost, Date

Normalisation:

Fine_info entity is in **1NF** as we assume all the attributes are atomic.

There are no partial dependencies, so it is in **2NF**.

There are no Transitive dependencies also, so it in **3NF** .

This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

R9. **Visitor_analysis(Park_id, Season, No_of_visitors)**

Primary key: Park_id,Season

Functional Dependencies :

Park_id,Season -> No_of_visitors

Normalisation:

Visitors_analysis entity is in **1NF** as we assume all the attributes are atomic.

There is no partial dependencies, so it is in **2NF**.

There are no Transitive dependencies also, so it in **3NF** .

This table is in **BCNF** also as for every functional dependency X -> Y, X is the super key of the entity.

RELATIONAL SCHEMA(BCNF)

Park	
Park_id(Primary)	char(6)
Park_name	Varchar(45)
Area_in_hectares	Int
City	Varchar(45)
Estd_year	Int

Flora_2	
Flora_id(Primary)	Char(5)
Flora_name	Varchar(45)

Flora_1	
Flora_id(Primary)	Char(5)
Park_id(Primary)	Char(6)
No_of_species	Int
Plantation_year	Int
Age	Int

Fauna_1	
Fauna_id(Primary)	Char(5)
Park_id(Primary)	Char(6)
No_of_species	Int

Fauna_2_1	
Fauna_id(Primary)	Char(5)
Species_name	Varchar(45)

Fauna_2_2	
Species_name	Varchar(45)
Lifespan	Varchar(45)
Status	Varchar(20)

Employee	
Employee_id(Primary)	char(10)
Park_id(Primary)	char(6)
Name	varchar(45)
Designation	varchar(45)
Monthly_salary	numeric(5,0)
Hours_of_work	int
Year_of_joining	int
Mobile_no.	numeric(10,0)

Visitors_2	
Age	Varchar(5)
Ticket_price	Int

Visitors_1	
Ticket_no(Primary)	Varchar(25)
Park_id(Foreign)	Char(6)
Date	Char(5)
Name	Varchar(45)
Age	Int
Gender	Char(6)
Mobile_no	Numeric(10,0)

Expenses	
Employee_salary	numeric(6,0)
Food	numeric(6,0)
Fertilizers	numeric(6,0)
Maintainence	numeric(6,0)
Park_id(Primary)	char(6)

Fine_info	
Ticket_no(Primary)	varChar(20)
Fine_reason	Varchar(100)
Fine_cost	Int
Date	Char(5)

Income	
Park_id(Primary)	Char(6)
Tickets_sold	Numeric(7,0)
Parking_fee	Numeric(6,0)
Cafeteria	Numeric(6,0)
Safari_fee	Numeric(6,0)
Fine	Numeric(6,0)

Visitor_analysis	
Park_id(Primary)	char(6)
Season(Primary)	varchar(45)
No_of_visitors	int

Flora_fertilizer	
Flora_id(Primary)	Char(5)
Park_id(primary)	Char(6)
Fertilizers(Primary)	Varchar(45)

Fauna_food	
Fauna_id(Primary)	Char(5)
Park_id(primary)	Char(6)
Food(Primary)	Varchar(45)

Funds	
Park_id(primary)	char(6)
Name(primary)	Varchar(35)
Amount(primary)	Numeric(6, 0)
Date(primary)	Char(5)


TABLES

Park

Park_id	Park_name	Area_in_hectares	City	Estd_year
UD-102	Udaan biodiversity park	200	Pune	2012
VA-103	Vanaparvam biodiversity park	111	Kozhikode	2010
YA-101	Yamuna biodiversity park	9770	Delhi	2015

Fauna_2_1

Fauna_id	Species_name
EL-03	Elephant
LI-01	Lion
MO-05	Monkey
PE-04	Peacock
TI-02	Tiger

Fauna_1

Fauna_id	Park_id	No_of_species
EL-03	UD-102	8
EL-03	YA-101	9
LI-01	UD-102	3
LI-01	VA-103	4
LI-01	YA-101	5
MO-05	UD-102	12
MO-05	VA-103	18
MO-05	YA-101	15
PE-04	UD-102	7
PE-04	VA-103	5
TI-02	VA-103	6
TI-02	YA-101	3

Fauna_food

Fauna_id	Food
EL-03	bark
EL-03	grasses
LI-01	birds
LI-01	turtles
MO-05	banana
MO-05	bird egg
PE-04	berries
PE-04	insects
TI-02	deer
TI-02	goat

Fauna_2_2

Species_name	Lifespan	Status
Elephant	60-70	endangered
Lion	15-20	vulnerable
Monkey	20-25	least concern
Peacock	10-25	least concern
Tiger	8-10	endangered

Flora_1

Flora_id	Park_id	No_of_species	Plantation_year	Age
BA-01	UD-102	25	2014	8
BA-01	VA-103	30	2012	10
BA-01	YA-101	35	2016	6
EU-03	UD-102	20	2013	9
EU-03	YA-101	30	2018	4
LI-05	UD-102	10	2014	8
LI-05	VA-103	22	2012	10
LO-02	UD-102	14	2017	5
LO-02	VA-103	17	2014	8
LO-02	YA-101	25	2012	10
RU-04	VA-103	12	2020	2
RU-04	YA-101	18	2019	3

Flora_fertilizers

Flora_id	Fertilizers
BA-01	Cattle manure
BA-01	Mushroom compost
EU-03	High potassium
EU-03	Low nitrogen
LI-05	NPK 10-30-20
LI-05	NPK 15-9-12
LO-02	NPK 18-18-21
LO-02	NPK 4-8-4
RU-04	NPK 10-10-10
RU-04	Rotted cow manure

Flora_2

Flora_id	Flora_name
BA-01	Bamboo
EU-03	Eucalyptus
LI-05	Lily
LO-02	Lotus
RU-04	Rudraksha

Fine_info

Ticket_no.	Fine_reason	Fine_cost	Date
UD-102-12/08-1	feeding food to animals	2000	10/11
UD-102-19/07-1000	wrong parking	1000	19/07
VA-103-30/07-956	throwing plastic bottles in the park	500	30/07
YA-101-10/11-83	feeding food to animals	2000	10/11

Visitors_1

Ticket_no.	Park_id	Date	Name	Age	Gender	Mobile_no.
UD-102-10/11-1	UD-102	10/11	Swapna	21	female	9184750253
UD-102-10/11-2	UD-102	10/11	Rishi	9	male	9184750253
UD-102-19/07-1000	UD-102	19/07	Karan	45	male	9372804567
VA-103-18/02-8	VA-103	18/02	Madhav	24	male	8283374925
VA-103-21/06-67	VA-103	21/06	Riya	5	female	6194638583
VA-103-30/07-956	VA-103	30/07	Rithwik	36	male	9930027856
YA-101-06/02-603	YA-101	06/02	Pranav	19	male	9736659927
YA-101-08/04-909	YA-101	08/04	Hanu	20	male	6933758926
YA-101-10/11-83	YA-101	10/11	Divya	51	female	9983374460
YA-101-19/04-9	YA-101	19/04	Manas	8	male	8376629887

Visitors_2

Age	Ticket_price
<=10	100
>10	200

Expenses

Park_id	Employee_salary	Food	Fertilizers	Maintenance
UD-102	52000	300000	70000	150000
VA-103	49000	200000	30000	100000
YA-101	95000	400000	100000	500000

Income

Park_id	Tickets_sold	Parking_fee	Cafeteria	Safari_fee	Fine
UD-102	600000	50000	70000	300000	30000
VA-103	400000	25000	45000	100000	10000
YA-101	1000000	80000	150000	475000	45000

Employee

Employee_id	Park_id	Name	Designation	Monthly_salary	Hours_of_work	Year_of_joining	Mobile_no
UD-102-M1	UD-102	Aarav	manager	25000	7	2015	8173562982
UD-102-S1	UD-102	Shiva	security	12000	10	2014	7193740360
UD-102-W1	UD-102	Harshith	worker	15000	8	2012	6193740631
VA-103-M3	VA-103	Suriya	manager	24000	7	2012	9281763942
VA-103-S3	VA-103	Charan	security	11000	10	2010	8162736471
VA-103-W3	VA-103	Arjun	worker	14000	8	2010	9183746509
YA-101-D2	YA-101	Swathi	doctor	40000	5	2022	9927746638
YA-101-M2	YA-101	Deekshit	manager	26000	7	2020	8244164779
YA-101-S2	YA-101	Shiva	security	13000	10	2015	9937746620
YA-101-W2	YA-101	Ram	worker	16000	8	2016	8176635429

Funds

Name	Amount	Date	Park_id
Aishwarya Rai Bachchan	700000	29/10	UD-102
Animal aid	75000	24/07	VA-103
Blue cross	225000	12/02	YA-101
Ghazal Alagh	200000	05/04	VA-103
Posh foundation	125000	17/06	YA-101
Ramesh Babu	5000	12/11	UD-102
Sita Rajput	7050	27/02	YA-101
Sita Rajput	10250	27/03	YA-101
Sonu Sood	500000	12/11	VA-103
Vishal K Reddy	100000	10/01	UD-102

Visitor_analysis

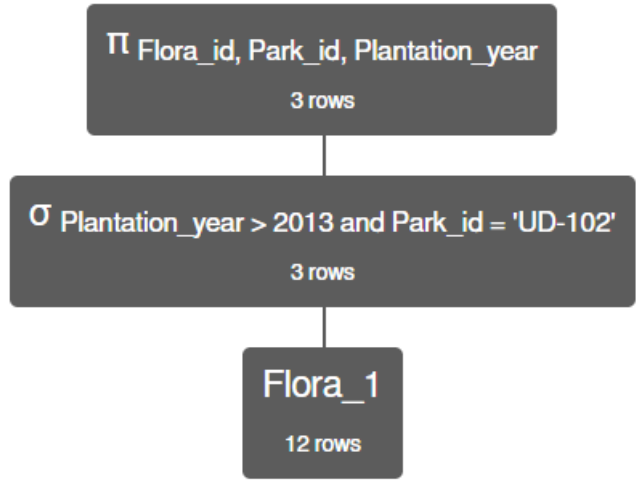
Park_id	Season	No_of_visitors
UD-102	Autumn	16000
UD-102	Monsoon	12000
UD-102	Spring	18000
UD-102	Summer	20000
UD-102	Winter	15000
VA-103	Autumn	9000
VA-103	Monsoon	7000
VA-103	Spring	5000
VA-103	Summer	11000
VA-103	Winter	6000
YA-101	Autumn	18000
YA-101	Monsoon	19000
YA-101	Spring	21000
YA-101	Summer	24000
YA-101	Winter	16000

RELATIONAL ALGEBRA EXPRESSIONS

Find flora_id, Park_id, Plantation_year from park where plantation_year >2013 and park_id = UD-102

```
1 π Flora_id,Park_id,Plantation_year (σPlantation_year > 2013 ∧ Park_id ='UD-
  102' (Flora_1))
```

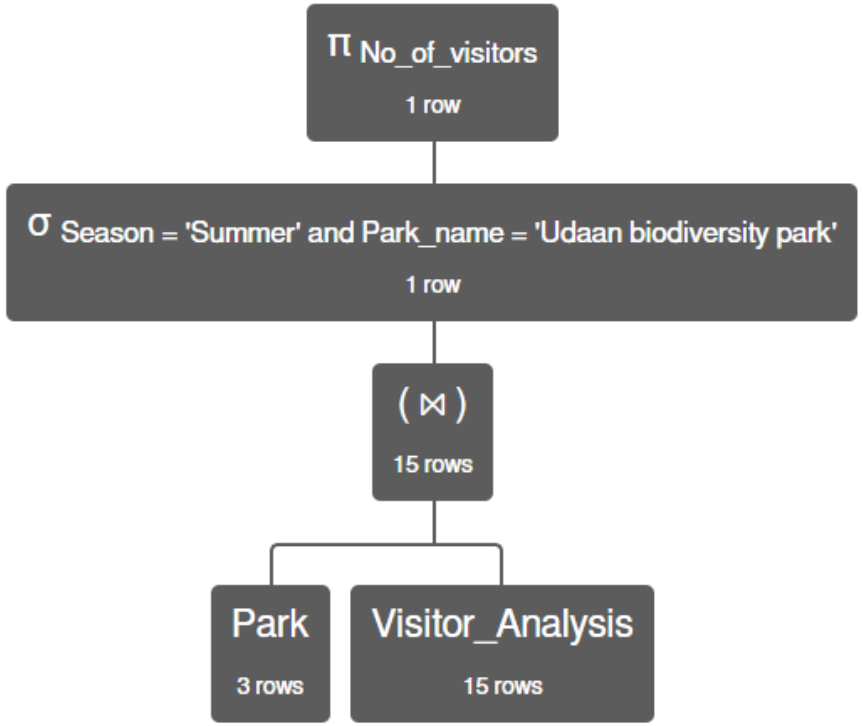
Flora_1.Flora_id	Flora_1.Park_id	Flora_1.Plantation_year
'BA-01'	'UD-102'	2014
'LI-05'	'UD-102'	2014
'LO-02'	'UD-102'	2017



Find no. of visitors who visited in season summer in Udaan biodiversity park

```
1 π No_of_visitors (σ Season = 'Summer' ∧ Park_name ='Udaan biodiversity
  park' (Park ⋈ Visitor_Analysis))
```

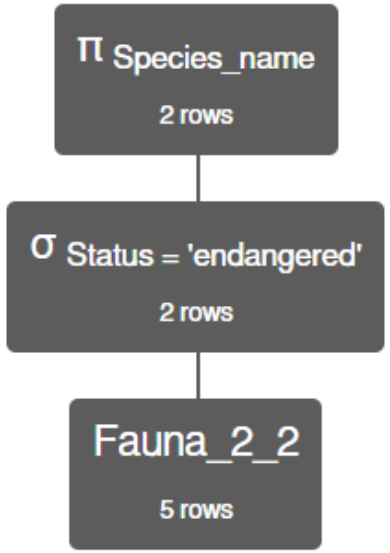
Visitor_Analysis.No_of_visitors
20000



Find species name where status is endangered

```
1 π Species_name (σ Status = 'endangered'  (Fauna_2_2))
```

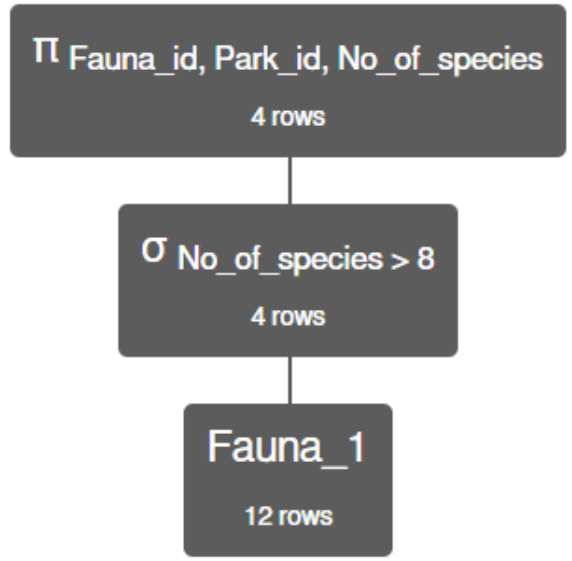
Fauna_2_2.Species_name
'Elephant'
'Tiger'



Find fauna_id, park_id and no of species where no of species is greater than 8

```
1 π Fauna_id,Park_id,No_of_species (σ No_of_species > 8  (Fauna_1))
```

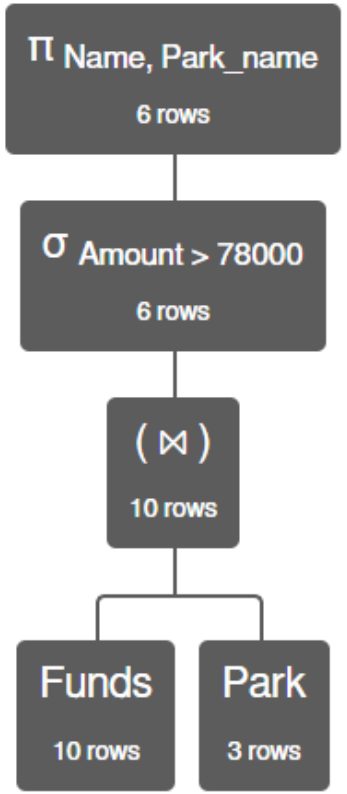
Fauna_1.Fauna_id	Fauna_1.Park_id	Fauna_1.No_of_species
'EL-03'	'YA-101'	9
'MO-05'	'UD-102'	12
'MO-05'	'VA-103'	18
'MO-05'	'YA-101'	15



Find name of donor and park_name where donation amount is greater than 78000

```
1 π Name,Park_name (σ Amount > 78000  (Funds⋈Park))
```

Funds.Name	Park.Park_name
'Aishwarya Rai Bachan'	'Udaan biodiversity park'
'Blue cross'	'Yamuna biodiversity park'
'Ghazal Alagh'	'Vanaparvam biodiversity park'
'Posh foundation'	'Yamuna biodiversity park'
'Sonu Sood'	'Vanaparvam biodiversity park'
'Vishal K Reddy'	'Udaan biodiversity park'



SQL QUERIES

Find number of visitors with age less than or equal to "20".

```
1 •    SELECT count(Name)
2      FROM visitors_1
3      where Age<=20;
```

	count(Name)
▶	5

Find the employee with fifth highest salary

```
1  select Employee_id,Name,Monthly_salary
2  from employee as e1
3  where 4=(select count(distinct Monthly_salary)
4           from employee as e2
5           where e2.Monthly_salary > e1.Monthly_salary);
```

	Employee_id	Name	Monthly_salary
▶	YA-101-W2	Ram	16000

Find the maximum number of visitors per season

```
1 •   select   max(No_of_visitors),season
2     from Visitor_analysis
3     group by season;
```

max(No_of_visitors)	season
18000	Autumn
19000	Monsoon
21000	Spring
24000	Summer
16000	Winter

Find fauna whose food starts with b and no of species is less than 10

```
1 •   Select Fauna_id,Food,Park_id
2     from Fauna_1 natural join Fauna_food
3     Where Food like "b%" and No_of_species<10;
```

Fauna_id	Food	Park_id
EL-03	bark	UD-102
EL-03	bark	YA-101
LI-01	birds	UD-102
LI-01	birds	VA-103
LI-01	birds	YA-101
PE-04	berries	UD-102
PE-04	berries	VA-103

Increase income earned by safari fee by 10% for park whose park_id starts with Y

```
1 •  update income
2    set Safari_fee= Safari_fee*1.1
3    where Park_id like 'Y%';
4 •  select *
5    from income;
```

Park_id	Tickets_sold	Parking_fee	Cafeteria	Safari_fee	Fine
UD-102	600000	50000	70000	300000	30000
VA-103	400000	25000	45000	100000	10000
YA-101	1000000	80000	150000	574750	45000


THANK YOU