LAU
School of Arts and Sciences

Department of Computer Science & Mathematics

A DATABASE DESIGN FOR CALL OF THE WILD ZOO PARK

by

ZAINAB YAHYA
TALA JURDI
RANIA DAKROUB
HUSSEIN RAKKA
NOUR ONAISSI

A REPORT
Submitted to DR. RAMZI HARATY in partial fulfillment of the requirements
for the course Database Management Systems in Computer Science

The IT Crowd
November 23, 2020

I. Introduction:

We, “The IT Crowd” team, are pleasured to present our project “A DATABASE DESIGN FOR CALL OF THE WILD ZOO PARK”. We were able to gain the necessary experience to work with databases and designing an ER diagram.

The report sums up our work on the project throughout the four phases of designing a database for a zoo park. In general, the experience was beneficial as we gained crucial skills that will help us in our careers in the future, even though that we did not have the best tools due to the unfortunate event of COVID-19 and turning to online learning.

The most challenging phase was phase III as we discovered a whole lot of new mistakes we have in the design of the database and had to rebuild the design, report and edit everything again. Also, it was a crucial phase since it included the use of the Oracle server to implement the code.

Phase I covered the process of defining our entities and relationships, it was challenging as we started from scratch to design the database, with minimal experience regarding zoos. The phase held challenges since the design we chose at the phase defined our path for the rest of the project, even if we changed few things later. Phase II covered translating these entities and relationships into tables and relation schemas that we would use later in our database.

After phase II, we were finished with the theoretical part and started converting our design into a real concrete database. Phase III consisted of converting our schema into an SQL code, filling the database with sample data and then implementing few queries based on our database and the context of the zoo.

Phase IV, the database was normalized for the sake of optimization and removing redundancy.

We are very pleased to take part in the creation of this database. We also would like to also thank Dr. Ramzi R. Haraty for his support and guidance throughout every step of this project even with the difficulties of online learning. Without Dr. Haraty, this project would not have been successful.

II. System Description and Constraints:

Zoos were made to bring wildlife closer to humans, for them to see these beautiful creatures allows them to understand nature better, and connect to it. So not only do they help unite and educate the community, but they also may reintroduce extinct and endangered species back into their natural environment.

Zoos today are uniquely equipped to preserve the future of wild species on our planet. These institutions oversee robust and sophisticated breeding programs that nurture threatened species and sustain genetic biodiversity, aiming to ultimately reintroduce the animals back into the wild. And they also are an economic boost for the local community and produce helpful scientific researches.

The zoo which we are working on, Call of The Wild, is here to create a habitat for all species of animals to save the endangered species; while helping people to gain more knowledge about these animals through creating the suitable environment yet paying attention to the safety of the visitors.

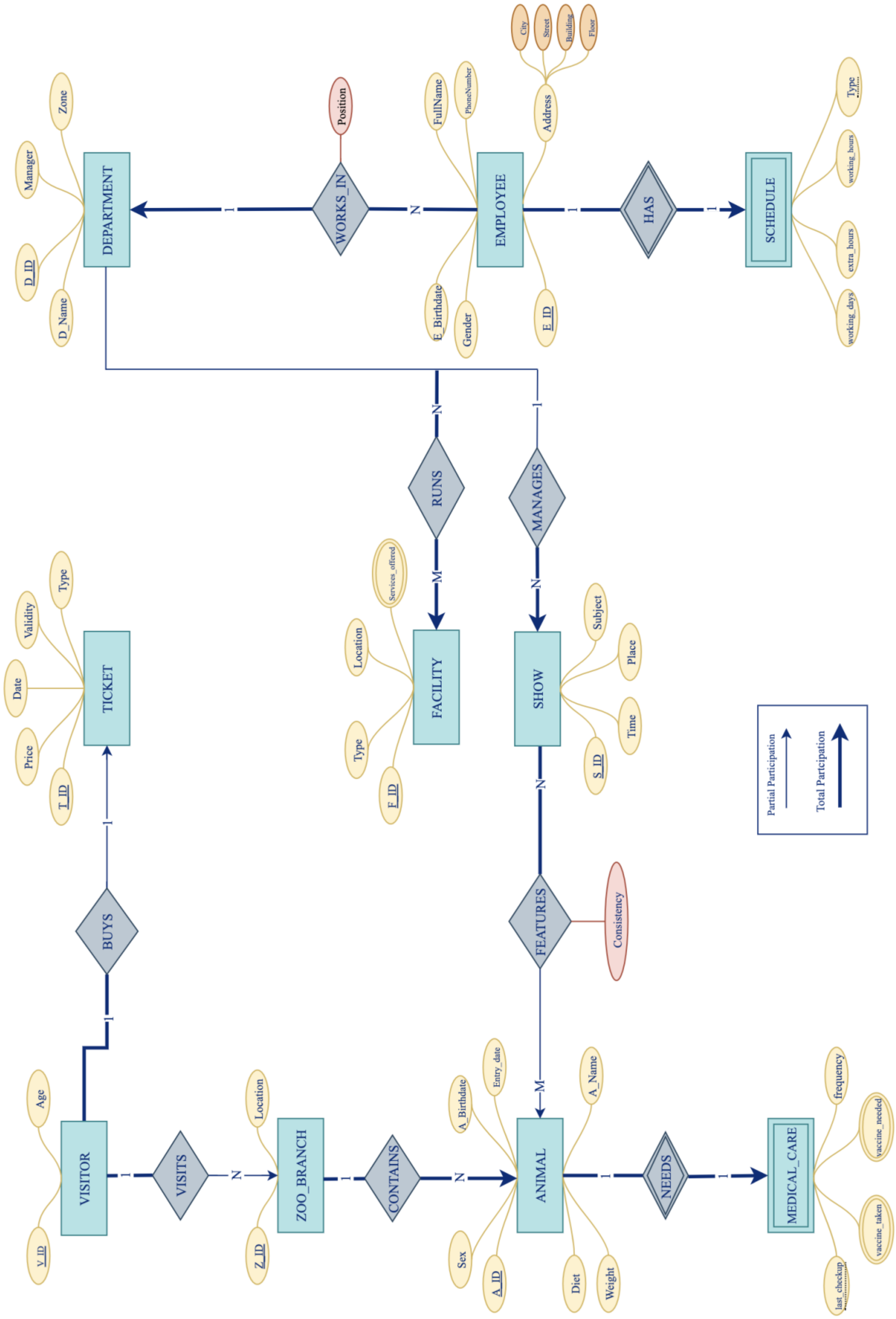
Since animals are the main concern of a zoo, the visitors must buy a ticket in order have access for the services of the zoo and to see the animals themselves. The zoo that we are dealing with, is a chain of zoos in fact; it has multiple branches across the country in more than one place. "Call of The Wild" zoo is a zoo that is spread over many countries across the middle east and expanding in Europe. However, our database will be concerned only about the branch of Beirut. We will be addressing the other branches in only relation only, which is VISITS.

Every department has its own employees working in it that have a certain schedule to follow, the department may be responsible for managing shows or running facilities . The visitor then buys the tickets for the zoo and shows featuring animals, those animals obviously need medical care to stay healthy.

These employees and zoo keepers are not only trained, but some also start out as interns or volunteers to learn more about the animals and what it takes to be a zookeeper. A tight schedule keeps employees on the right times to sell the tickets, and help produce animal shows that attract many visitors, and apart all that, animals need medical care most of the time which made each zoo provide a proper and educated medical team.

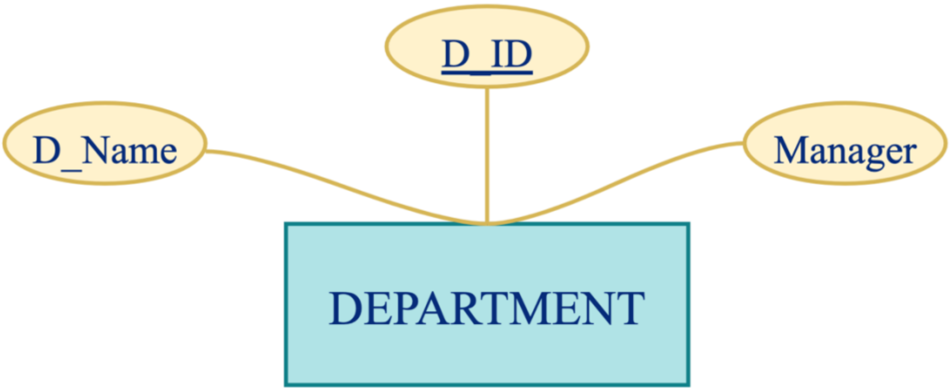
Zoos are actually more than animals, most of them have a full range of modern various facilities to make visits comfortable, easy and enjoyable like wheelchair hire, lockers, information hub, toilets and much more. The zoo has other facilities that are presented to provide the visitor with a better experience, these facilities include restaurants, cafes, giftshops, etc.

III. ER Diagram:



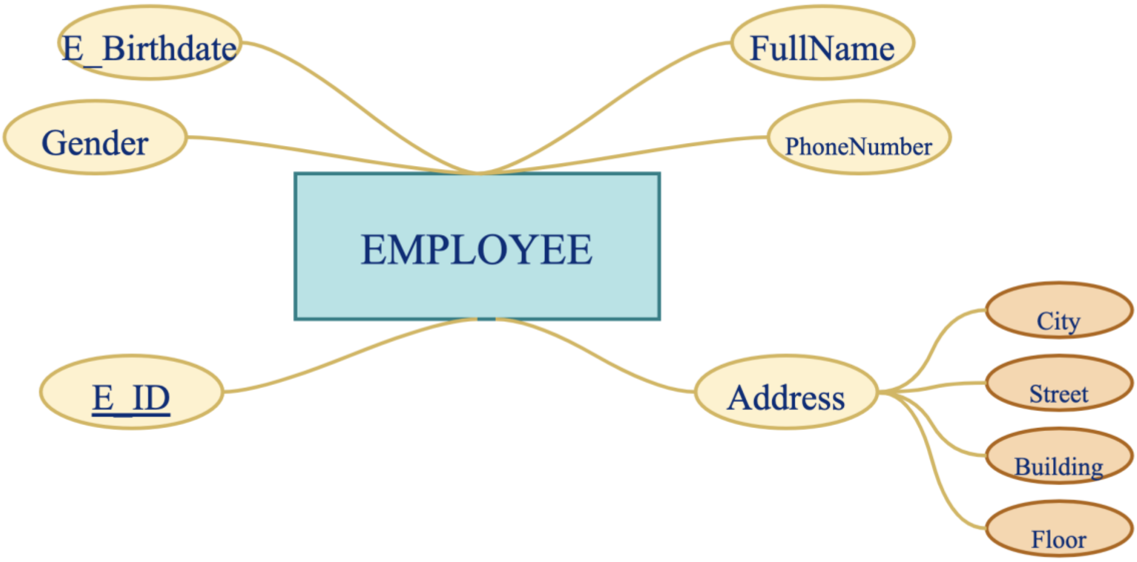
IV. Entities:

1. DEPARTMENT:



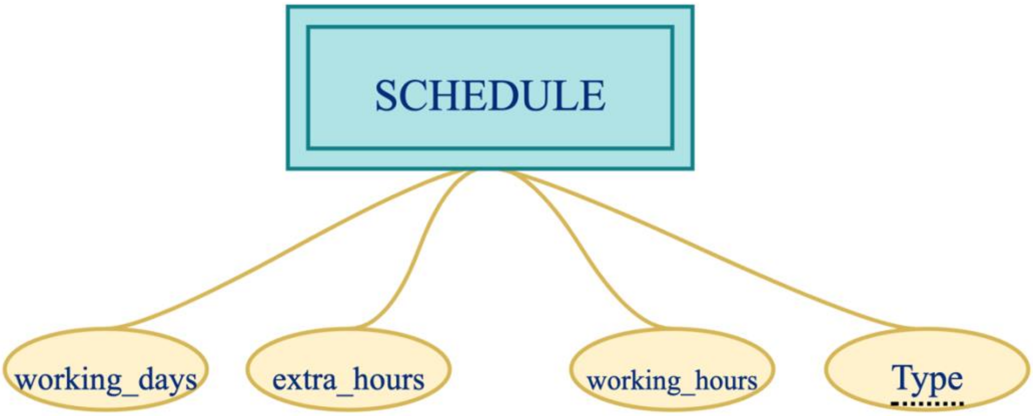
A zoo have more than the animals, there are many other things to manage. That's why we present DEPARTMENT, it could be the HR, finance, or security department. Every department is identified by a unique ID, D_ID which is the primary key, D_Name which represent its name, and a Manager to specify the name of the employee that is responsible for the department.

2. EMLOYEE:



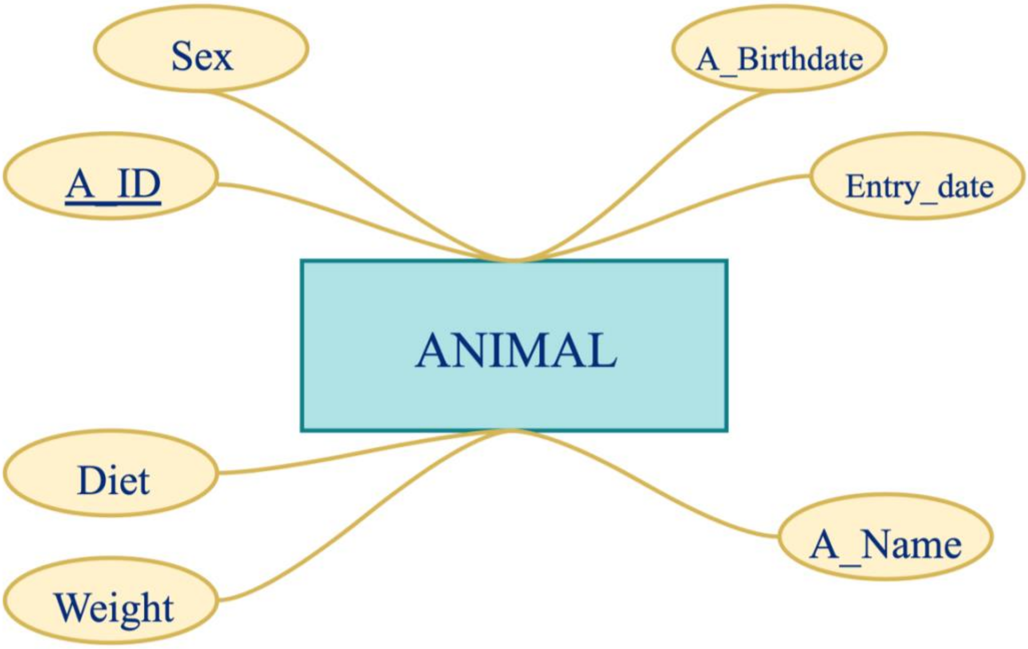
To run the zoo, employees are needed. Each EMPLOYEE is represented by a unique E_ID, which is the primary key. The employees have other attributes as well that are essential in their files; those are: FullName, Gender, PhoneNumber, B_Birthdate, and the Address. Address is a composite attribute made of 4 simple attributes: City, Street, Building and Floor.

3. SCHEDULE:



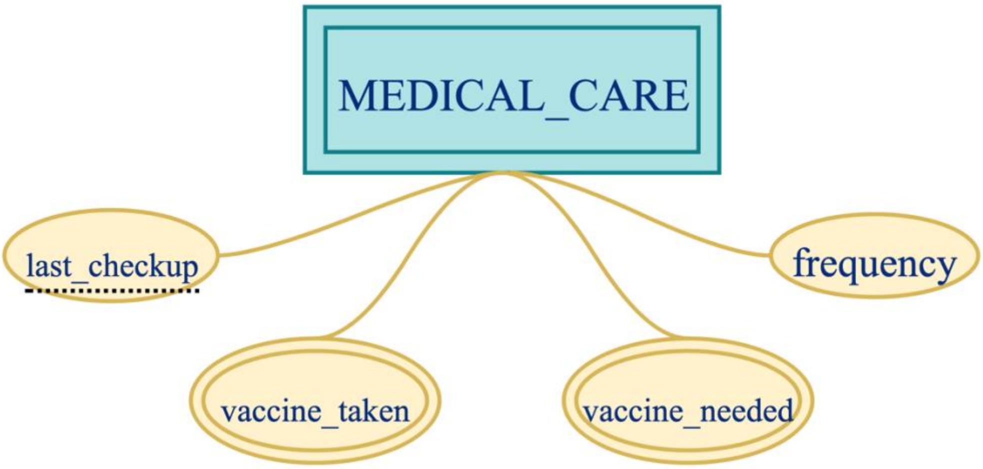
Employees cannot be assigned randomly, they need a schedule to follow; that's why we create a weak attribute that is named SCHEDULE. The SCHEDULE's attributes are Type: to specific whether it is full time or part time, and it is the partial key, working_days, working _hours, and extra_hours in case the employee is doing an extra shift.

4. ANIMAL:



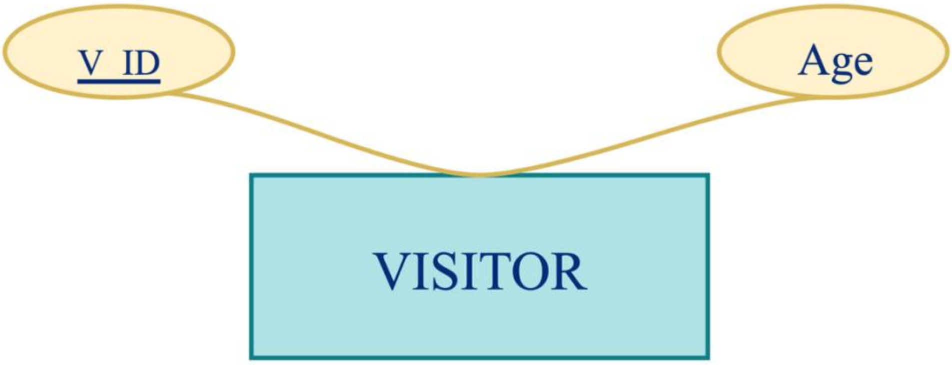
As the purpose of a zoo is featuring animals of different types for people, then the animal is the main concern of our zoo. When adding an ANIMAL to our database, we need to provide all the essential information of the animal. First, every animal has a unique A_ID to be differentiated from other animals, so its ID is the primary key. Also, there is A_Name, A_Birthdate and Entry_date. To be provided a better medical care, the Weight and Sex of the animal are also needed. Finally, each animal have a specific Diet that the zoo have to provide.

5. MIDECAL_CARE:



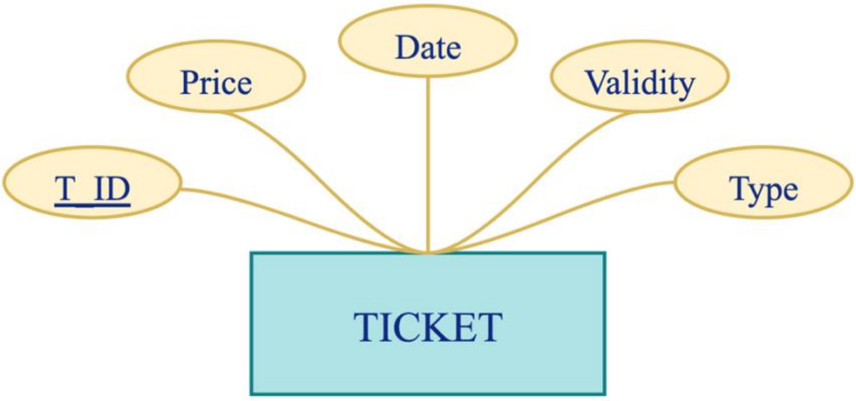
To keep the animals safe and provide a healthy environment in the zoo, medical care must be continuously provided for the animals. MEDICAL_CARE consists of the vaccinations needed for every animal; in order to distinguish between the needed vaccinations and the taken ones, separate attributes are provided for each. Thus, we have two attribute named vaccine_needed and another named vaccine_taken; both of these attributes are multivalued since an animal usually needs more than one vaccine. Also, we need the date of the last_checkup done for the animal, being the partial key, and frequency of the checkups needed to ensure the health of the animal.

6. VISITOR:



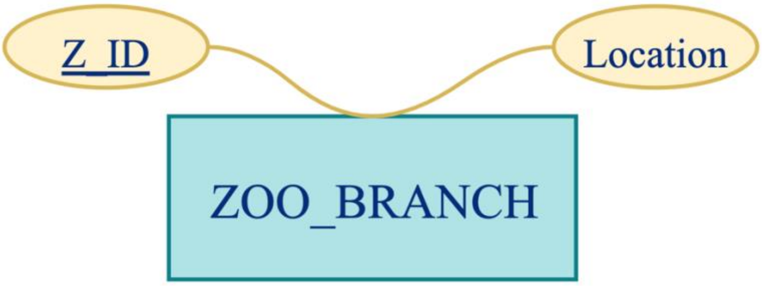
The visitor of the zoo is the main focus of creating a zoo; this visitor is the reason why the zoo keeps operating and offering services. Although the visitors are vital for a zoo, their personal information do not add a value for the database, so our database is satisfied with the main data about the visitor. The visitors will be asked only age because it affect the type of tickets they are getting. Also, the visitors will be given an entrance number V_ID automatically after giving their required information, this number is initialed at zero when the zoo opens every morning. V_ID is the primary key of VISITOR entity.

7. TICKET:



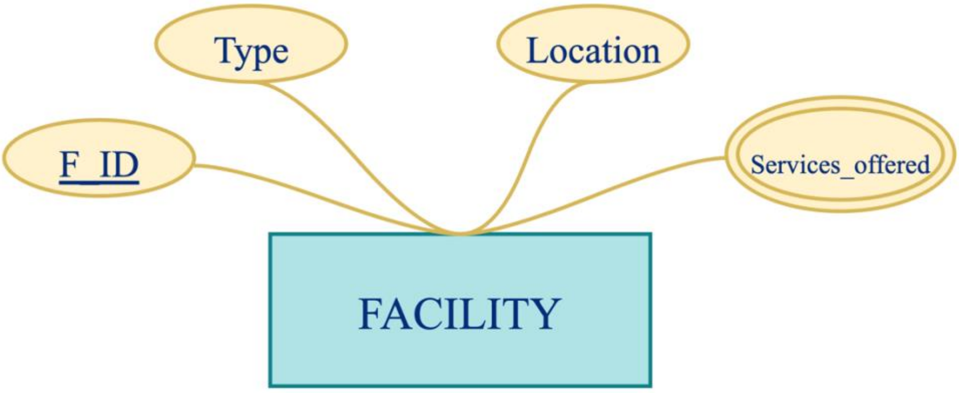
Every visitor has to buy a ticket in order to enter the zoo and have access for all its services. Each TICKET is identified by a unique T_ID which is the primary key; it has a Type that is determined by the services that the client wants, a Price that is related to the type of the ticket, the Validity of this ticket that determine the duration for which the ticket is usable, and the current Date.

8. ZOO_BRANCH:



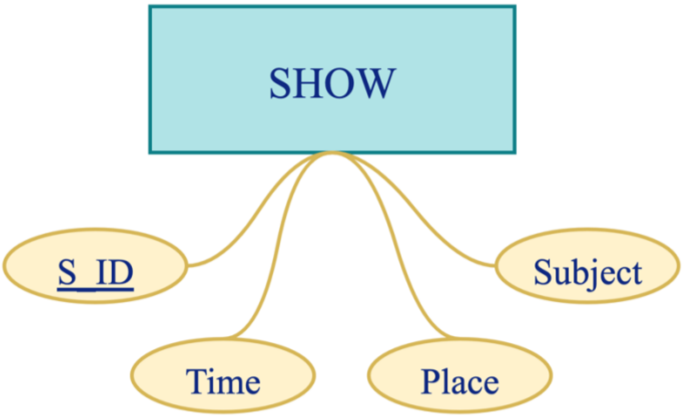
Since we are dealing with a chain of zoos that are spread across the country, then it will be wise to have ZOO_BRANCH entity to store the information of every branch separately. Each branch has a unique Z_ID which is the primary key and location which is the city it is located in.

9. FACILITY:



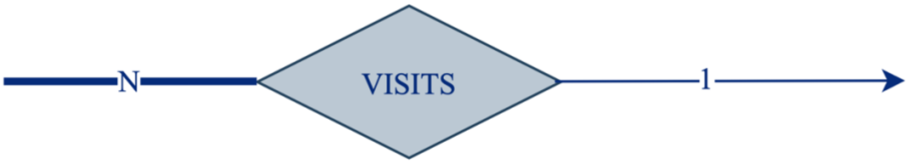
To keep the visitors entertained, a zoo must provide other services than the animals. The more services there are, the more pleased visitors we have. These services include the restaurants and cafes that are inside the zoo, giftshops, etc.. which are offered through certain facilities. Every FACILITY is presented through a number which is the primary key, it is named F_ID. Also, the service has other attributes such as Location, Type and services_offered.

10. SHOW:

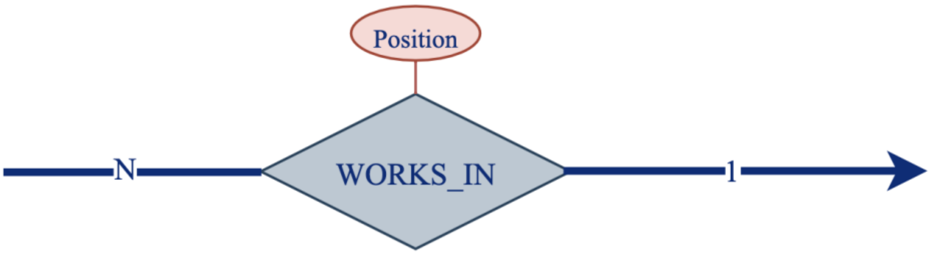


Featuring animals alone is boring, that's why zoos offers shows to entertain the visitors. As usual, a SHOW is identified by an exclusive S_ID, the show has Time and Place, and a Subject. as in the animals that it will be offering.

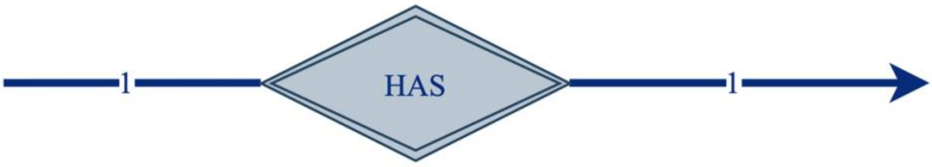
V. Relationships:



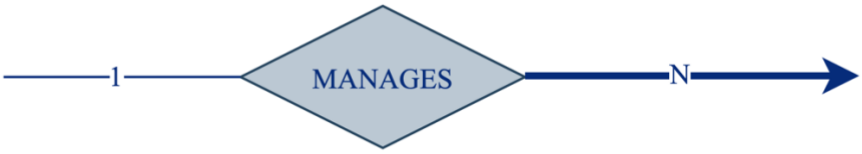
A **visitor** is related to a **zoo** by the act of visiting it. Thus, a "VISITS" relationship is created between **VISITOR Entity** and **ZOO_BRANCH Entity**. A visitor can visit only one zoo branch at a time, while a zoo branch may have many visitors so it is a N:1 relationship. The participation is total on the side of the visitor because otherwise we will not be concerned with the visitor; however, it is partial on the side of zoo branch because a visitor does not need to visit all branches in order to be considered as a visitor.



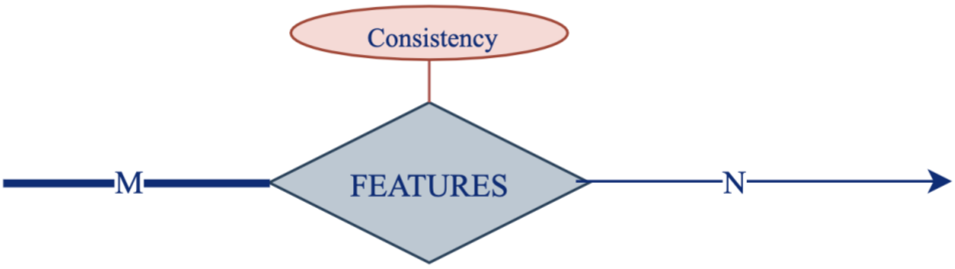
All **employees** have to work in the **department** for finalizing tasks. Thus, a "WORKS_IN" relationship has to be created between **EMPLOYEE Entity** and **DEPARTMENT Entity.** Every employee can work in one department but every department could have more than one employee, so it is a N:1 relationship. The participation is total on both sides of the relation because each department should have employees to operate and an employee has to belong to a department. . The relationship has an attribute "Position" to specify what does the employee do in that department.



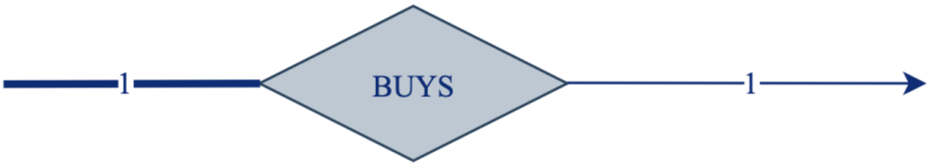
Each **employee** has a specific **schedule** to abide with. Thus, a "HAS" identifying relationship has to be created between the **EMPLOYEE ENTITY** and the **SCHEDULE ENTITY.** Every employee has a one schedule to follow and each schedule is associated by an employee, so it is a 1:1 relationship. The participation is in total since all employees must have a schedule to abide with such as specific time to take a break, tasks to be done etc.…



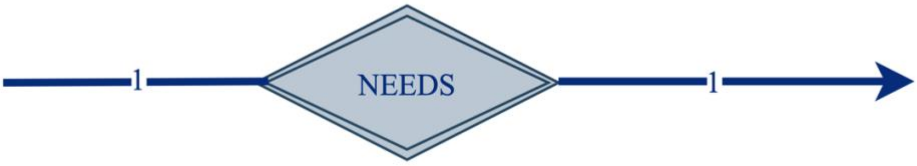
A **show** is run by a **department**. Thus, a “MANAGES” relationship has to be created between **DEPARTMENT Entity** and **SHOW Entity.** A department can run many shows but a show could be run by one department only, so it is a 1:N relationship. The participation is partial on the department side since not all departments need to run shows but total on the show side since a show must be run by a department.



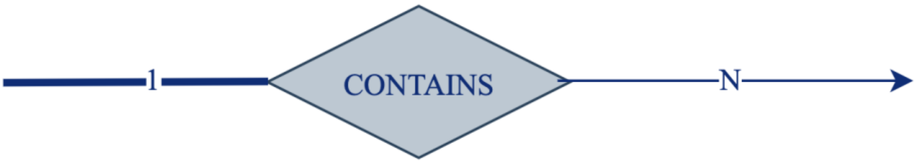
A **show** cannot exist without **animals**. Thus a “FEATURES” relationship is created between **SHOW Entity** and **ANIMAL Entity**. A show may have more than one animal and an animal can participate in more than one show, so it is an N:M relationship. The participation is total on the side of the show since a show needs animals to be run, but it is partial participation on the side of the animals since not all animals need to be part of a show. The relation has an attribute, Consistency, to specify how frequent the show is presented, whether it is daily, weekly or even monthly.



Every **visitor** must buy a **ticket** in order to have access for the zoo. Thus, a “BUYS” relationship is created between the **VISITOR Entity** and **TICKET Entity**. Each visitor needs only one ticket so it is 1:1 relationship. Every visitor must buy a ticket, so it is a total participation on the side of the visitor while it is partial on the side of the ticket because not all the ticket will be necessarily bought.

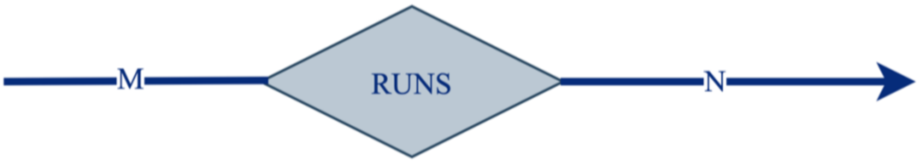


Each **animal** should be provided a **medical care**. Thus, the “NEEDS” identifying relationship has to be created between ANIMAL Entity and **weak MEDICAL_CARE Entity**. Each animals needs one medical record to be associated to it and there will not be any medical record that it is not assigned to an animal, so it is a 1:1 relationship. The participation is total on both sides since every animal needs medical care and medical care is associated to an animal.



We will use the same relationship CONTAINS twice since it has the same features in both cases.

Naturally, a **zoo** must have **animals**. Thus, a “CONTAINS” relationship is created between ZOO_BRANCH Entity and ANIMAL Entity. A zoo must have more than one animal while an animal can be present in one zoo only; thus, it is a 1:N relationship. The participation is total on both sides since a zoo must contain animals and an animal must be in the zoo to be registered in our database.



A **department** is what manages **facilities**. Thus, a “RUNS” relationship has to be created between **DEPARTMENT Entity** and **FACILITY Entity**. A department can run more than one facility and a facility could be run by more than a department according to what it offers, so it is a N:M relationship. The participation is total on the side of facility since every facility should be run by a department, but the participation is partial on the side of department because not all departments run a facility.

VI. ER to Relational Mapping Algorithms

After designing the ER schema and having displayed the database for the ZOO as a system of entities, attributes, and relationships, this high level design must be translated into a relational database design. In order to map the ER design to a relational database design, a seven-step algorithm needs to be followed. The following is a detailed description on applying the different steps to our database design.

STEP 1: Mapping of Regular Entity Types

In the first step, the regular entity types must be mapped into relations. Each regular entity is going to have its own relation that includes all of its simple attributes and a single primary key which is underlined. The regular (strong) entities in this database design for zoo are: VISITOR, TICKET, ANIMAL, ZOO_BRANCH, SHOW, FACILITY, EMPLOYEE and DEPARTMENT.

VISITOR

V_ID	Age

The VISITOR entity contains only simple attributes so we simply include in this relation the four attributes: *Age* and *V_number* which is underlined for being the primary key.

TICKET

T_ID	Price	Date	Validity	Type

The TICKET entity contains only simple attributes so we simply include in this relation the five attributes: *Price*, *Date*, *Validity*, *Type* and *T_ID* which is underlined for being the primary key.

ZOO_BRANCH

Z_ID	Location

The ZOO_BRANCH entity contains only simple attributes so we simply include in this relation the four attributes: *Location* and *Z_ID* which is underlined for being the primary key.

DEPARTMENT

D_ID	D_Name	Manager

The DEPARTMENT entity contains only simple attributes so we simply include in this relation the four attributes: *Manager*, *D_Name* and *D_ID* which is the primary key.

SHOW

S_ID	Time	Place	Subject

The SHOW entity contains only simple attributes so we simply include in this relation the four attributes: *Place*, *Time*, *Subject* and *S_ID* which is underlined for being the primary key.

FACILITY

F_ID	Type	Location

The FACILITY entity contains simple and multivalued attributes, the multivalued attribute *Services_offered* is not represented in the relation. The relation include the three simple attributes: *Type*, *Location* and *F_ID* which is underlined for being the primary key.

ANIMAL

A_ID	A_NAME	A_Birthdate	Entry_date	Sex
Weight	Diet			

The ANIMAL entity contains simple and multivalued attributes, the multivalued attribute *Environment* is not represented in the relation. The relation include the three simple attributes: *Sex*, *Birthdate*, *Entry_dat*e, *Diet*, *Weight* and *A_ID* which is underlined for being the primary key.

EMPLOYEE

E_ID	FullName	E_Birthdate	Gender	phoneNumber
City	Street	Building	Floor	

The EMPLOYEE entity contains both simple and composite attributes. We include the all the simple attributes: *E_ID* which is underlined for being the primary key, *FullName*, *Gender*, E_*Birthdate* and *PhoneNumber*. The EMPLOYEE entity has *Address* as a composite attribute out of which only the simple attributes *City*, *Street*, *Building* and *Floor* are included in the relation.

STEP 2: Mapping of Weak Entity Types

In this step, the weak entity types are mapped into relations. As in Step 1, only the simple attributes are included in the relations and not multivalued or derived attributes. Furthermore, weak entity relation has a foreign key attribute which is the primary key of the owner entity type. The combination of the foreign key added and the partial key of the weak entity type represent the primary key of the relation. The weak entities in our database design are: SCHEDULE and MEDICAL_CARE.

MEDICAL_CARE

FK_A_ID	Last_Checkup	Checkup_Duration

The weak entity MEDICAL_CARE includes multivalued attributes vaccine_taken and vaccine_needed, which are not going to be included in this relation. The weak entity has no derived attributes. The simple attributes are: last_checkup and checkup_duration which are included. Moreover, the Animal_ID , the primary of the owner entity ANIMAL , is included. Last_checkup and the primary key Animal_ID are combined to represent the primary key of this relation.

SCHEDULE

FK_E_ID	Type	Working_Days	Working_Hours	Extra_Hours

The weak entity SCHEDULE does not include a multivalued attribute. The weak entity does not have any derived attributes. The simple attributes are: working_days, extra_hours, working_hours and type. Moreover, the E_ID, the primary of owner entity EMPLOYEE, is included. E_ID and the partial key Type are combined to represent the primary key of this relation.

STEP 3: Mapping of Binary 1:1 Relationship Types

In this step, we are going to map the binary one-to-one relationships. In order to accomplish our goal we can follow one of three approaches. The first approach, called foreign key approach is where we choose the entity on the total participation side of the relation, then we add as a foreign key the primary key of the other entity participating in this relation. The second approach, called merged relation approach is where we merge the two entities participating in the relationship into a single relation. This is only used when both participations are total and thus not useful in our case. The third approach, called cross- reference or relationship relation approach is where we create a third relation which will include the primary keys of both entities participating in the relationship. We are going to follow the foreign key approach because it is the most useful in our case. The binary one-to-one relationship that need to be mapped is: BUYS.

VISITOR (BUYS)

V_ID	T_ID	Age

Every visitor must get a ticket to enter the zoo. The “BUYS” relationship links the VISITOR entity and the TICKET entity. On the VISITOR entity side, we have a total participation while on the TICKET entity side we have a partial participation. We choose the VISITOR relation in which we added, as a foreign key, the primary key of the Ticket which is T_ID.

STEP 4: Mapping of Binary 1:N Relationship Types

In this step, we are going to map the binary one-to-many relationships. We add a foreign key in the entity type at the many sides of the relationship. This foreign key is the primary key of the other entity type participating in this relationship. We must also include any other simple attribute of the one-to-many relationship. The one-to-many relationships that need to be mapped are: VISITS, CONTAINS, PRODUCES, MANAGES and WORKS_IN.

ZOO_BRANCH (VISITS)

V_Number	Z_ID	Location

A visitor may visit more than one zoo branch. The "VISITS" relationship links the ZOO_BRANCH entity and the VISITOR entity. The ZOO_BRANCH entity is on the many side so; we add to its relation the foreign key ID which is the primary key of the VISITOR entity named V_NUMBER.

ANIMAL (CONTAINS)

Z_Number	A_ID	A_Name	A_Birthdate	Entry_date
Sex	Weight	Diet		

A zoo branch may contain more than one animal. The "CONTAINS" relationship links the ANIMAL entity and the ZOO_BRANCH entity. The ANIMAL entity is on the many side so; we add to its relation the foreign key ID which is the primary key of the ZOO_BRANCH entity named Z_Number.

SHOW (MANAGES)

Dp_Number	S_ID	Place	Time
Subject			

A department may manage more than one show. The "MANAGES" relationship links the SHOW entity and the DEPARTEMENT entity. The SHOW entity is on the many side so; we add to its relation the foreign key ID which is the primary key of the DEPARTEMENT entity and named it DEPARTEMENT_NAME

EMPLOYEE (WORKS_IN)

Dprt_Number	E_ID	FullName	E_Birthdate	Gender
City	Street	Building	Floor	phoneNumber
Position				

Many employees may work in one department. The “WORKS_IN” relationship links the EMPLOYEE entity and the DEPARTEMENT entity. The EMPLOYEE entity is on the many side so; we add to its relation the foreign key ID which is the primary key of the DEPARTEMENT entity and named it DEPARTEMENT_NAME.

STEP 5: Mapping of M:N Relationship Types

In this step, we are going to map the binary many-to-many relationships. For each many-to-many relationship we are going to create a new relation which includes, as foreign keys, the primary keys of all participating relations. Their combination will form the primary key of this newly created relation. We must also include any other simple attribute of the many-to-many relationship. The many-to-many relationships needed to be mapped are:

RUNS

Dprt_Number	F_Number

There is an M: N relation between DEPARTMENT and FACILITY entities. A "WORKS_ON" relation, which contains as foreign keys the primary keys of both DEPARTMENT and FACILITY entities, is created. Hence, WORKS_ON relation will include the attributes Department-Name and FNumber. These attributes form together the primary key.

FEATURES

S_Number	A_Number	Consistency

There is an M:N relation between SHOW and ANIMAL entities. Thus, a FEATURES relation is formed, this relation includes the primary keys of both SHOW and ANIMAL entities as foreign keys. The FEATURES relation contains the attributes Show-ID and Animal-Identity, both these attributes form the primary key of this relation.

STEP 6: Mapping of Multivalued Attributes

In this step, we are going to map the multivalued attributes which we ignored before. For each multivalued attribute we create a new relation containing the related attribute and the primary key of the entity to which it belongs. Their combination will represent the primary key of the newly created relation. We have three multivalued attributes which are: the taken vaccines, needed vaccines, and facility services.

T_VACCINE

Animal_ID	Last_Checkup_Date	Vaccine_done

The Vaccine_Taken attribute is a multivalued attribute in MEDICAL_CARE entity. A T_VACCINE relation is formed, it includes the primary key A_ID and Last_Checkup as foreign keys and the corresponding attribute Vaccine_Taken_Date. Combined, they form a primary key.

N_VACCINE

Animal_Nu	Lastly_checked_up	Vaccine_to_be_taken

The Vaccine_needed attribute is a multivalued attribute in MEDICAL_CARE entity. A N_VACCINE relation, which contains the primary key A_ID and Last_Checkup as foreign keys and the corresponding attribute Needed_V, is formed. Combined, they form a primary key.

FACILITY_SERVICES

F_Number	Service

The FACILITY_SERVICES relation holds the primary key B_Number_of the entity FACILITY as a foreign. The primary key is formed of both F_Number and Service.

STEP 7: Mapping of N-ary Relationship Types

In this step, we are going to map the N-ary Relationship types. We should create a new relation containing the primary keys of all participating entities and any simple attributes of the relationship type. In our design we have no N-ary relationship types, so this step is not applicable here.

FINAL STEP: Final Displays

VISITOR

V_ID	Age

TICKET

T_ID	Price	Date	Validity	Type

ZOO_BRANCH

Z_ID	Location

DEPARTMENT

D_ID	D_Name	Manager

SHOW

S_ID	Time	Place	Subject

FACILITY

F_ID	Type	Location

ANIMAL

A_ID	A_NAME	A_Birthdate	Entry_date	Sex
Weight	Diet			

EMPLOYEE

E_ID	FullName	E_Birthdate	Gender	phoneNumber
City	Street	Building	Floor	

MEDICAL_CARE

FK_A_ID	Last_Checkup	Checkup_Duration

SCHEDULE

FK_E_ID	Type	Working_Days	Working_Hours	Extra_Hours

VISITOR (BUYS)

V_ID	T_ID	Age

ZOO_BRANCH (VISITS)

V_Number	Z_ID	Location

ANIMAL (CONTAINS)

Z_Number	A_ID	A_Name	A_Birthdate	Entry_date
Sex	Weight	Diet		

SHOW (MANAGES)

Dp_Number	S_ID	Place	Time
Subject			

EMPLOYEE (WORKS_IN)

Dprt_Number	E_ID	FullName	E_Birthdate	Gender
City	Street	Building	Floor	phoneNumber
Position				

RUNS

Dprt_Number	F_Number

FEATURES

S_Number	A_Number	Consistency

T_VACCINE

Animal_ID	Last_Checkup_Date	Vaccine_done

N_VACCINE

Animal_Nu	Lastly_checked_up	Vaccine_to_be_taken

FACILITY_SERVICES

F_Number	Service

VII. Table Structure for Promise Hospital:

After designing the ER diagram for Call of The Wild Zoo and mapping this diagram into relational database design, now it is time to start creating the actual tables for our database on the Oracle Database Server. We will start by creating all tables and then inserting data into these tables. Finally, we will execute some queries to display the importance of the database and especially in a zoo.

1. ANIMAL:

```
CREATE TABLE  ANIMAL
(
        A_ID VARCHAR(10) NOT NULL,
        A_Name VARCHAR2(15) NOT NULL,
        A_Birthdate INT NOT NULL,
        Entry_Date INT NOT NULL,
        Sex CHAR(1),
        Weight VARCHAR2(15),
        Diet CHAR(12),
        PRIMARY KEY(A_ID)
  ) ;
```

2. VISITOR:

```
CREATE TABLE VISITOR
 (
        V_ID VARCHAR(13),
        Age INT,
        PRIMARY KEY(V_ID)
);
```

3. ZOO_BRANCH:

```
CREATE TABLE ZOO_BRANCH
(
        Z_ID VARCHAR(4),
        Location VARCHAR(20),
        PRIMARY KEY (Z_ID)
);
```

4. TICKET:

```
CREATE TABLE TICKET
(
        T_ID VARCHAR(15) PRIMARY KEY,
        Price INT,
        T_Date DATE,
        Validity INT,
        T_Type CHAR(15) NOT NULL
);
```

5. DEPARTMENT:

```
CREATE TABLE DEPARTMENT
(
        D_ID CHAR(5) PRIMARY KEY ,
        D_NAME VARCHAR(30) NOT NULL ,
        MANAGER VARCHAR(20) NOT NULL
);
```

6. EMPLOYEE:

```
CREATE TABLE EMPLOYEE
(
        E_ID CHAR(8) PRIMARY KEY ,
        FullName VARCHAR(15) NOT NULL ,
        E_Birthdate DATE ,
        Gender CHAR NOT NULL CHECK(GENDER IN ('F', 'M')),
        PhoneNumber VARCHAR(15) ,
        City VARCHAR(9) NOT NULL ,
        Street VARCHAR(15) ,
        Building VARCHAR(20) ,
        Floor INT
);
```

7. FACILITY:

```
CREATE TABLE FACILITY
(
        F_ID CHAR(4) PRIMARY KEY,
        Type CHAR(20),
        Location CHAR(2)
);
```

8. SHOW:

```
CREATE TABLE SHOW
(
        S_ID VARCHAR(5) PRIMARY KEY,
        Place VARCHAR(15) ,
        Subject VARCHAR(15) ,
         S_Time VARCHAR(5)
);
```

9. SCHEDULE:

```
CREATE TABLE SCHEDULE
(
        FK_E_ID CHAR(9) ,
        Type VARCHAR(15) ,
        Working_Days VARCHAR(9) ,
        Working_Hours VARCHAR(9) ,
        Extra_Hours INT ,
FOREIGN KEY(FK_E_ID) REFERENCES EMPLOYEE(E_ID)
);
```

10. MEDICAL_CARE:

```
CREATE TABLE MEDICAL_CARE
(
        FK_A_ID VARCHAR(10),
        Last_Checkup DATE,
        CheckUp_Duration VARCHAR(10),
        FOREIGN KEY(FK_A_ID) REFERENCES ANIMAL(A_ID),
        PRIMARY KEY (FK_A_ID,Last_Checkup)
);
```

11. TAKEN VACCINES:

```
CREATE TABLE T_VACCINE
 (
        Animal_ID VARCHAR(40),
        Last_Checkup_Date DATE,
        Vaccine_done CHAR(40),
        PRIMARY KEY (Animal_ID, Last_Checkup_Date, Vaccine_done)
 ) ;
```

12. NEEDED VACCINES:

```
CREATE TABLE N_VACCINE
  (
        Animal_Number VARCHAR(40),
        Lastly_checked_up DATE,
        Vaccine_to_be_taken CHAR(40),
         PRIMARY KEY (Animal_Number, Lastly_checked_up, Vaccine_to_be_taken)
);
```

13. FACILITY SERVICES:

```
CREATE TABLE FACILITY_SERVICES
(
        F_Number VARCHAR(40) PRIMARY KEY,
        SERVICE VARCHAR(40)
);
```

14. VISITS:

```
CREATE TABLE VISITS
(
        V_Number VARCHAR(14) ,
        Z_ID VARCHAR(4) ,
        Location VARCHAR(20),
        FOREIGN KEY(V_Number) REFERENCES VISITOR(V_ID),
        FOREIGN KEY(Z_ID) REFERENCES ZOO_BRANCH(Z_ID),
        PRIMARY KEY(V_Number,Z_ID)
);
```

15. FEATURES:

```
CREATE TABLE FEATURES
(
        S_Number CHAR(40),
        A_Number CHAR(40),
        Consistency CHAR(40),
        PRIMARY KEY ( S_Number, A_Number)
);
```

16. CONTAINS:

```
CREATE TABLE CONTAINS
(
        Z_Number VARCHAR(4) NOT NULL,
        A_ID VARCHAR(10) NOT NULL,
        A_Name VARCHAR(15) ,
        A_Birthdate INT ,
        Entry_Date INT,
        Sex CHAR(1),
        Weight VARCHAR(15),
        Diet VARCHAR (12),
        FOREIGN KEY(Z_Number) REFERENCES ZOO_BRANCH(Z_ID),
        FOREIGN KEY(A_ID) REFERENCES ANIMAL(A_ID),
        PRIMARY KEY (Z_Number,A_ID)
);
```

17. WORKS IN:

```
CREATE TABLE WORKS_IN
(
        Dprt_Number CHAR(5) ,
        E_ID CHAR(8) ,
        FullName VARCHAR(15) NOT NULL ,
        E_Birthdate DATE ,
        Gender CHAR NOT NULL CHECK(GENDER IN ('F', 'M')),
        phoneNumber VARCHAR(15) ,
        City VARCHAR(9) NOT NULL ,
        Street VARCHAR(15) ,
        Building VARCHAR(12),
        Floor INT ,
        POSITION VARCHAR(30) ,
        PRIMARY KEY (Dprt_Number, E_ID),
         FOREIGN KEY(E_ID) REFERENCES EMPLOYEE(E_ID),
        FOREIGN KEY (Dprt_Number) REFERENCES DEPARTMENT(D_ID)
);
```

18. BUYS:

```
CREATE TABLE BUYS
(
        V_ID VARCHAR(10),
        T_ID VARCHAR(15),
        AGE INT,
        FOREIGN KEY(V_ID) REFERENCES VISITOR(V_ID),
        FOREIGN KEY (T_ID) REFERENCES TICKET(T_ID),
        PRIMARY KEY(V_ID, T_ID)
);
```

19. MANAGES:

```
CREATE TABLE MANAGES
(
        DP_NUMBER CHAR(9),
        S_ID VARCHAR(9),
        M_Time TIMESTAMP,
        Place CHAR(15),
        Subject CHAR(40),
        FOREIGN KEY (DP_NUMBER) REFERENCES DEPARTMENT (D_ID)
);
```

20. RUNS:

```
CREATE TABLE RUNS
(
        Dprt_Number CHAR(40),
        F_Number CHAR(40),
        PRIMARY KEY ( Dprt_Number, F_Number)
);
```

VIII. Table Descriptions:

After creating all the tables on the oracle database server we can view the description of each table in order to make sure everything is fine and no mistakes were made during creation of table.

In our database we have the following tables created on the oracle database server:

1. ANIMAL:

Query:

DESC ANIMAL;

Results:

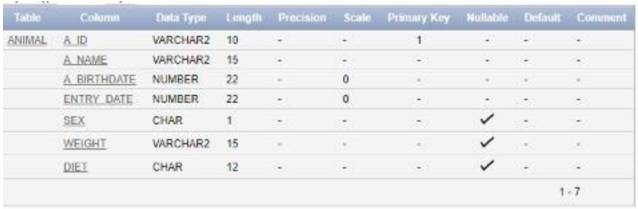


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ANIMAL	A_ID	VARCHAR2	10	-	-	1	-	-	-
	A_NAME	VARCHAR2	15	-	-	-	-	-	-
	A_BIRTHDATE	NUMBER	22	-	0	-	-	-	-
	ENTRY_DATE	NUMBER	22	-	0	-	-	-	-
	SEX	CHAR	1	-	-	-	✓	-	-
	WEIGHT	VARCHAR2	15	-	-	-	✓	-	-
	DIET	CHAR	12	-	-	-	✓	-	-

1 - 7

2. VISITOR:

Query:

DESC VISITOR;

Results:

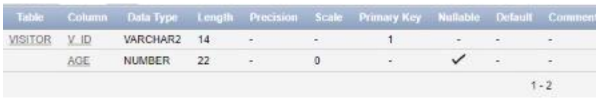


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Commen
VISITOR	V_ID	VARCHAR2	14	-	-	1	-	-	-
	AGE	NUMBER	22	-	0	-	✓	-	-

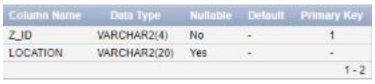
1 - 2

3. ZOO_BRANCH:

Query:

DESC ZOO_BRANCH;

Results:



Column Name	Data Type	Nullable	Default	Primary Key
Z_ID	VARCHAR2(4)	No	-	1
LOCATION	VARCHAR2(20)	Yes	-	-

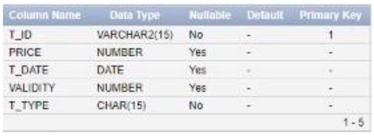
1 - 2

4. TICKET:

Query:

DESC TICKET;

Results:



Column Name	Data Type	Nullable	Default	Primary Key
T_ID	VARCHAR2(15)	No	-	1
PRICE	NUMBER	Yes	-	-
T_DATE	DATE	Yes	-	-
VALIDITY	NUMBER	Yes	-	-
T_TYPE	CHAR(15)	No	-	-

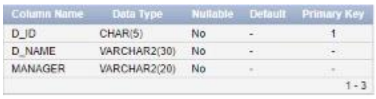
1 - 5

5. DEPARTMENT:

Query:

```
DESC DEPARTMENT;
```

Results:



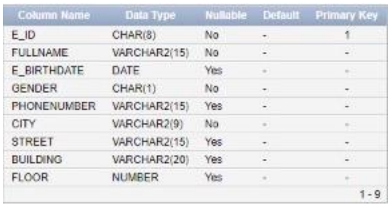
Column Name	Data Type	Nullable	Default	Primary Key
D_ID	CHAR(5)	No	-	1
D_NAME	VARCHAR2(30)	No	-	-
MANAGER	VARCHAR2(20)	No	-	-
				1 - 3

6. EMPLOYEE:

Query:

```
DESC EMPLOYEE;
```

Results:



Column Name	Data Type	Nullable	Default	Primary Key
E_ID	CHAR(8)	No	-	1
FULLNAME	VARCHAR2(15)	No	-	-
E_BIRTHDATE	DATE	Yes	-	-
GENDER	CHAR(1)	No	-	-
PHONENUMBER	VARCHAR2(15)	Yes	-	-
CITY	VARCHAR2(9)	No	-	-
STREET	VARCHAR2(15)	Yes	-	-
BUILDING	VARCHAR2(20)	Yes	-	-
FLOOR	NUMBER	Yes	-	-
				1 - 9

7. FACILITY:

Query:

```
DESC FACILITY;
```

Results:

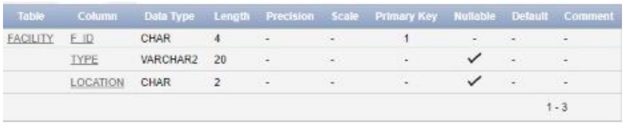


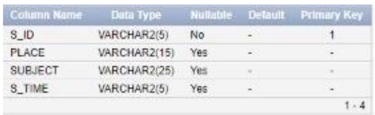
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACILITY	F_ID	CHAR	4	-	-	1	-	-	-
	TYPE	VARCHAR2	20	-	-	-	✓	-	-
	LOCATION	CHAR	2	-	-	-	✓	-	-
								1 - 3	

8. SHOW:

Query:

```
DESC SHOW;
```

Results:



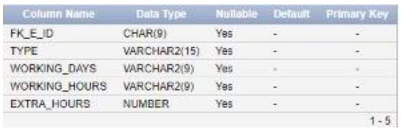
Column Name	Data Type	Nullable	Default	Primary Key
S_ID	VARCHAR2(5)	No	-	1
PLACE	VARCHAR2(15)	Yes	-	-
SUBJECT	VARCHAR2(25)	Yes	-	-
S_TIME	VARCHAR2(5)	Yes	-	-
				1 - 4

9. SCHEDULE:

Query:

```
DESC SCHEDULE;
```

Results:



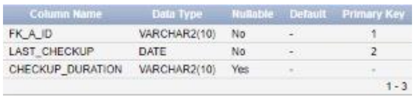
Column Name	Data Type	Nullable	Default	Primary Key
FK_E_ID	CHAR(9)	Yes	-	-
TYPE	VARCHAR2(15)	Yes	-	-
WORKING_DAYS	VARCHAR2(9)	Yes	-	-
WORKING_HOURS	VARCHAR2(9)	Yes	-	-
EXTRA_HOURS	NUMBER	Yes	-	-
				1 - 5

10. MEDICAL_CARE:

Query:

```
DESC MEDICAL_CARE;
```

Results:



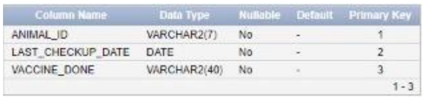
Column Name	Data Type	Nullable	Default	Primary Key
FK_A_ID	VARCHAR2(10)	No	-	1
LAST_CHECKUP	DATE	No	-	2
CHECKUP_DURATION	VARCHAR2(10)	Yes	-	-
				1 - 3

11. TAKEN VACCINES:

Query:

```
DESC T_VACCINE;
```

Results:



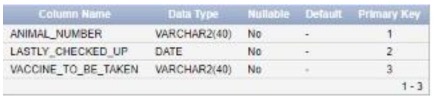
Column Name	Data Type	Nullable	Default	Primary Key
ANIMAL_ID	VARCHAR2(7)	No	-	1
LAST_CHECKUP_DATE	DATE	No	-	2
VACCINE_DONE	VARCHAR2(40)	No	-	3
				1 - 3

12. NEEDED VACCINES:

Query:

```
DESC N_VACCINE;
```

Results:



Column Name	Data Type	Nullable	Default	Primary Key
ANIMAL_NUMBER	VARCHAR2(40)	No	-	1
LASTLY_CHECKED_UP	DATE	No	-	2
VACCINE_TO_BE_TAKEN	VARCHAR2(40)	No	-	3
				1 - 3

13. FACILITY SERVICES:

Query:

```
DESC FACILITY_SERVICE;
```

Results:

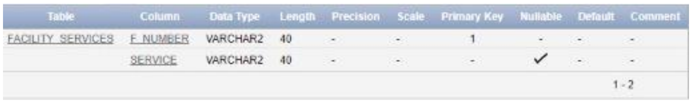


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACILITY_SERVICES	F_NUMBER	VARCHAR2	40	-	-	1	-	-	-
	SERVICE	VARCHAR2	40	-	-	-	✓	-	-
								1 - 2	

14. VISITS:

Query:

```
DESC VISITS;
```

Results:

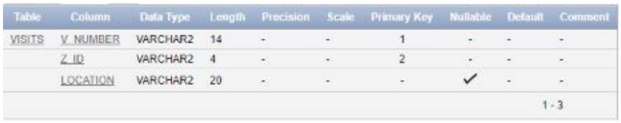


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VISITS	V_NUMBER	VARCHAR2	14	-	-	1	-	-	-
	Z_ID	VARCHAR2	4	-	-	2	-	-	-
	LOCATION	VARCHAR2	20	-	-	-	✓	-	-
								1 - 3	

15. FEATURES:

Query:

```
DESC FEATURES;
```

Results:

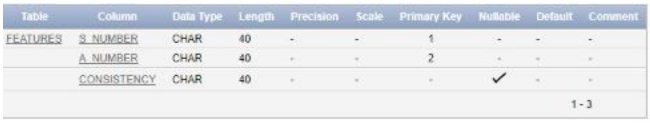


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FEATURES	S_NUMBER	CHAR	40	-	-	1	-	-	-
	A_NUMBER	CHAR	40	-	-	2	-	-	-
	CONSISTENCY	CHAR	40	-	-	-	✓	-	-

1 - 3

16. CONTAINS:

Query:

```
DESC CONTAINS;
```

Results:

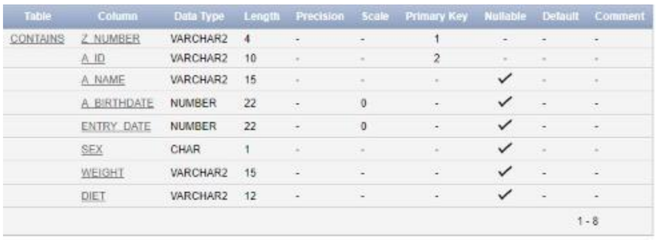


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CONTAINS	Z_NUMBER	VARCHAR2	4	-	-	1	-	-	-
	A_ID	VARCHAR2	10	-	-	2	-	-	-
	A_NAME	VARCHAR2	15	-	-	-	✓	-	-
	A_BIRTHDATE	NUMBER	22	-	0	-	✓	-	-
	ENTRY_DATE	NUMBER	22	-	0	-	✓	-	-
	SEX	CHAR	1	-	-	-	✓	-	-
	WEIGHT	VARCHAR2	15	-	-	-	✓	-	-
	DIET	VARCHAR2	12	-	-	-	✓	-	-

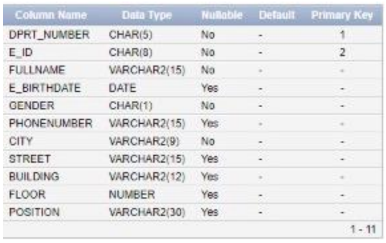
1 - 8

17. WORKS IN:

Query:

```
DESC WORKS_IN;
```

Results:



Column Name	Data Type	Nullable	Default	Primary Key
DPRT_NUMBER	CHAR(5)	No	-	1
E_ID	CHAR(8)	No	-	2
FULLNAME	VARCHAR2(15)	No	-	-
E_BIRTHDATE	DATE	Yes	-	-
GENDER	CHAR(1)	No	-	-
PHONENUMBER	VARCHAR2(15)	Yes	-	-
CITY	VARCHAR2(9)	No	-	-
STREET	VARCHAR2(15)	Yes	-	-
BUILDING	VARCHAR2(12)	Yes	-	-
FLOOR	NUMBER	Yes	-	-
POSITION	VARCHAR2(30)	Yes	-	-

1 - 11

18. BUYS:

Query:

```
DESC BUYS;
```

Results:

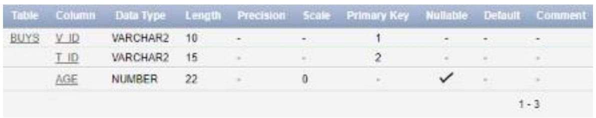


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BUYS	V_ID	VARCHAR2	10	-	-	1	-	-	-
	T_ID	VARCHAR2	15	-	-	2	-	-	-
	AGE	NUMBER	22	-	0	-	✓	-	-

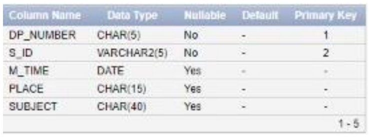
1 - 3

19. MANAGES:

Query:

DESC MANAGES;

Results:



Column Name	Data Type	Nullable	Default	Primary Key
DP_NUMBER	CHAR(5)	No	-	1
S_ID	VARCHAR2(5)	No	-	2
M_TIME	DATE	Yes	-	-
PLACE	CHAR(15)	Yes	-	-
SUBJECT	CHAR(40)	Yes	-	-
				1 - 5

20. RUNS:

Query:

DESC RUNS;

Results:

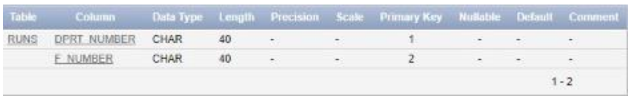


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RUNS	DPRT_NUMBER	CHAR	40	-	-	1	-	-	-
	F_NUMBER	CHAR	40	-	-	2	-	-	-
								1 - 2	

IX. Inserting Data:

1. ANIMAL:

```
INSERT INTO ANIMAL VALUES('A-001', 'Panda', 2010, 2015, 'M', '80kg', 'Herbivorous');
INSERT INTO ANIMAL VALUES('A-002', 'Panda', 2007, 2015, 'F', '75kg', 'Herbivorous');
INSERT INTO ANIMAL VALUES('A-003','Asian Elephant',2007,2015, 'F', '3500kg', 'Herbivorous');
INSERT INTO ANIMAL VALUES('A-004', 'Asian Elephant', 1990, 2010, 'M', '8000kg', 'Herbivorous' );
INSERT INTO ANIMAL VALUES('A-005', 'Lion', 2015, 2020, 'M','100kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-006', 'Lion', 2015, 2019, 'F','90kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-007', 'Penguin',2019, 2020,'F','22kg','Fish');
INSERT INTO ANIMAL VALUES('A-008', 'Penguin',2018, 2020,'F','20kg','Fish');
INSERT INTO ANIMAL VALUES('A-009', 'Monkey',2008, 2011,'M','20kg','Omnivore');
INSERT INTO ANIMAL VALUES('A-010', 'Monkey',2007, 2010,'M','27kg','Omnivore');
INSERT INTO ANIMAL VALUES('A-011', 'Dolphin',2000,2010,'F','3000kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-012', 'Dolphin',2002, 2009,'F','3700kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-013', 'Dolphin',1998,2009,'M','4000kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-014', 'Monkey',2007, 2010,'F','20kg','Omnivore');
INSERT INTO ANIMAL VALUES('A-015', 'Seal',2001, 2005,'F','50kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-016', 'Sloth',2006, 2008,'M','5kg','Omnivore');
INSERT INTO ANIMAL VALUES('A-017', 'Crocodile',1970, 2005,'M','250kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-018', 'Rabbit',2020, 2020,'M','1kg','Herbivore');
INSERT INTO ANIMAL VALUES('A-019', 'Rabbit',2020, 2020,'F','1kg','Herbivore');
INSERT INTO ANIMAL VALUES('A-020', 'Golden Eagle',2005,
2015,'F','4kg','Carnivore');
INSERT INTO ANIMAL VALUES('A-021', 'Buttefly',2020,2020,'F','','Herbivore');
```

2. VISITOR:

```
INSERT INTO VISITOR VALUES ('20201112-0120',31);
INSERT INTO VISITOR VALUES ('20201112-0123',33);
INSERT INTO VISITOR VALUES ('20201112-0124',25);
INSERT INTO VISITOR VALUES ('20201112-0125',5);
INSERT INTO VISITOR VALUES ('20201112-0126',40);
INSERT INTO VISITOR VALUES ('20201112-0127',35);
INSERT INTO VISITOR VALUES ('20201112-0128',14);
INSERT INTO VISITOR VALUES ('20201112-0129',10);
INSERT INTO VISITOR VALUES ('20201112-0130',7);
INSERT INTO VISITOR VALUES ('20201112-0131',5);
INSERT INTO VISITOR VALUES ('20201112-0132',31);
INSERT INTO VISITOR VALUES ('20201112-0133',20);
INSERT INTO VISITOR VALUES ('20201112-0134',24);
INSERT INTO VISITOR VALUES ('20201112-0135',2);
INSERT INTO VISITOR VALUES ('20201112-0136',60);
INSERT INTO VISITOR VALUES ('20201112-0137',53);
INSERT INTO VISITOR VALUES ('20201112-0138',41);
INSERT INTO VISITOR VALUES ('20201112-0139',10);
INSERT INTO VISITOR VALUES ('20201112-0140',9);
```

3. ZOO_BRANCH:

```
INSERT INTO ZOO_BRANCH VALUES('Z_0','Lebanon,Beirut');
INSERT INTO ZOO_BRANCH VALUES('Z_1','UAE,Dubai');
INSERT INTO ZOO_BRANCH VALUES('Z_2','UAE,Abu Dhabi');
INSERT INTO ZOO_BRANCH VALUES('Z_3','Iraq,Baghdad');
INSERT INTO ZOO_BRANCH VALUES('Z_4','Syria,Damascus');
INSERT INTO ZOO_BRANCH VALUES('Z_5','KSA,Riyadh');
INSERT INTO ZOO_BRANCH VALUES('Z_6','Morocco,Rabat');
INSERT INTO ZOO_BRANCH VALUES('Z_7','Egypt,Cairo');
INSERT INTO ZOO_BRANCH VALUES('Z_8','Jordan,Amman');
INSERT INTO ZOO_BRANCH VALUES('Z-9','Qatar,Doha');
```

4. TICKET:

```
INSERT INTO TICKET VALUES('11-0000001', 15, 12-11-2020, 12, 'Prime');
INSERT INTO TICKET VALUES('11-0000002', 15, 12-11-2020, 12, 'Prime');
INSERT INTO TICKET VALUES('11-0000003', 15, 12-11-2020, 12, 'Prime');
INSERT INTO TICKET VLAUES('11-0000004', 15, 12-11-2020, 12, 'Prime');
INSERT INTO TICKET VALUES('10-0000001', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000002', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000003', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000004', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000005', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000006', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('10-0000007', 10, 11-11-2020, 8, 'Regular');
INSERT INTO TICKET VALUES('01-0000001', 10, 12-11-2020, 12,'K-Prime');
INSERT INTO TICKET VALUES('01-0000002', 10, 12-11-2020, 12,'K-Prime');
INSERT INTO TICKET VALUES('01-0000003', 10, 12-11-2020, 12,'K-Prime');
INSERT INTO TICKET VALUES('01-0000004', 10, 12-11-2020, 12,'K-Prime');
INSERT INTO TICKET VALUES('00-0000001', 7,  11-11-2020, 8, 'K-Regular');
INSERT INTO TICKET VALUES('00-0000002', 7,  11-11-2020, 8, 'K-Regular');
INSERT INTO TICKET VALUES('00-0000003', 7,  11-11-2020, 8, 'K-Regular');
```

5. DEPARTMENT:

```
INSERT INTO DEPARTMENT VALUES('D-00','ADMINISTRATION','Hussein Muhsen');
INSERT INTO DEPARTMENT VALUES('D-01','HEALTH & SAFETY','Zeinab Yahya');
INSERT INTO DEPARTMENT VALUES('D-02','FINANCE','Tala Jurdi');
INSERT INTO DEPARTMENT VALUES('D-03','CONSERVATION & RESEARCH','Mia Mansour');
INSERT INTO DEPARTMENT VALUES('D-04','ANIMAL CARE','Israa Haider');
INSERT INTO DEPARTMENT VALUES('D-05','AQUATIC','Rania Dakroub');
INSERT INTO DEPARTMENT VALUES('D-06','VISITOR & CUSTOMER SERVICES','Natalie Nazzal');
INSERT INTO DEPARTMENT VALUES('D-07','MARKETING','Ascia Hakeem');
INSERT INTO DEPARTMENT VALUES('D-08','PUBLIC RELATIONS','Hussein Rakka');
INSERT INTO DEPARTMENT VALUES('D-09','HUMAN RESOURCES','Noor Onaisi');
```

6. EMPLOYEE:

```
INSERT INTO EMPLOYEE VALUES('E-00001','Noor Zaarour',DATE'1990-5-4','F','76735584','Baalbeck','Bk-638','Al Turath','5');
INSERT INTO EMPLOYEE VALUES('E-00002','Hussein Muhsen',DATE'1996-12-10','M','76296411','Beirut','B-861','666 Meters','6');
INSERT INTO EMPLOYEE VALUES('E-00003','Zeinab Yahya',DATE'1997-8-27','F','76863278','Beirut','B-843','77 Doors','6');
INSERT INTO EMPLOYEE VALUES('E-00004','Michelle Mourad',DATE'1988-10-10','M','76885377','Byblos','By-277','77 Doors','1');
INSERT INTO EMPLOYEE VALUES('E-00005','Tala Jurdi',DATE'1979-8-14','F','76196582','Joneih','J-945','Adonis 2','3');
INSERT INTO EMPLOYEE VALUES('E-00006','Israa Haider',DATE'1981-4-13','F','76996357','Baalbeck','Bk-647','Al Turath','2');
INSERT INTO EMPLOYEE VALUES('E-00007','Ibrahim kansour',DATE'1967-7-18','M','76015489','Tyre','T-190','OC 5','3');
INSERT INTO EMPLOYEE VALUES('E-00008','Rania Dakroub',DATE'1959-9-8','F','76551894','Beirut','B-682','the 990','1');
INSERT INTO EMPLOYEE VALUES('E-00009','Noor Onaisi',DATE'1980-5-6','F','76945177','Beirut','B-624','Warda','2');
INSERT INTO EMPLOYEE VALUES('E-000010','Hussein Rakka',DATE'1975-9-10','M','76268593','Tripoli','Ti-433','The 117','3');
INSERT INTO EMPLOYEE VALUES('E-000011','Jad Baydoun',DATE'1992-9-1','M','76892547','Byblos','By-287','The Abroad','1');
INSERT INTO EMPLOYEE VALUES('E-000012','Mariam Zaytouni',DATE'1988-2-16','F','76327854','Beirut','B-611','Building 37','4');
INSERT INTO EMPLOYEE VALUES('E-000013','Daniell Bannout',DATE'1983-9-9','M','76884138','Byblos','By-256','The Above','5');
INSERT INTO EMPLOYEE VALUES('E-000014','Natalie Nazzal',DATE'1989-1-30','F','76884138','Beirut','B-683','Al Manara','1');
INSERT INTO EMPLOYEE VALUES('E-000015','Josephine bazzi',DATE'1988-3-11','F','76763299','Beirut','B-649','sky Gate','1');
INSERT INTO EMPLOYEE VALUES('E-000016','Sarah Chouman',DATE'1992-6-22','F','76886312','Beirut','B-693','Sama Beirut','3');
INSERT INTO EMPLOYEE VALUES('E-000017','Jaber Soeid',DATE'1959-6-12','M','76184659','Kheyam','Kh-022','the 801','2');
INSERT INTO EMPLOYEE VALUES('E-000018','Mia Mansour',DATE'1966-8-13','F','76901145','Joneih','J-933','Adonis 7','4');
INSERT INTO EMPLOYEE VALUES('E-000019','Lia Ashkar',DATE'1993-12-15','F','76564888','Beirut','B-638','Middle Eastern 66','3');
INSERT INTO EMPLOYEE VALUES('E-000020','Ascia Hakeem',DATE'1998-5-5','F','76981437','Beirut','B-610','Mediterran 6','6');
```

7. FACILITY:

```
INSERT INTO FACILITY VALUES('F-00','Restrooms','A1');
INSERT INTO FACILITY VALUES('F-01','Restaurant','B1');
INSERT INTO FACILITY VALUES ('F-02','Cafe','B1');
INSERT INTO FACILITY VALUES ('F-03', 'Giftshop', 'A2');
INSERT INTO FACILITY VALUES ('F-04', 'Prayer Room', 'A1');
INSERT INTO FACILITY VALUES ('F-05', 'Children Care', 'A1');
INSERT INTO FACILITY VALUES ('F-06', 'Parking lot', 'A4');
INSERT INTO FACILITY VALUES ('F-07', 'The Animal Store', 'C1');
INSERT INTO FACILITY VALUES ('F-08', 'Tour Guide', 'A3');
INSERT INTO FACILITY VALUES ('F-09', 'Donation Center', 'A2');
```

8. SHOW:

```
INSERT INTO SHOW VALUES('S-000',  'D3', 'Dolphin's Show','11:00');
NSERT INTO SHOW VALUES('S-001', 'D4', 'Tour in the Aquarium','12:30');
INSERT INTO SHOW VALUES('S-002', 'C2', 'Monkeys Battle', '13:00');
INSERT INTO SHOW VALUES('S-003', 'D2', 'Penguins and beyond','14:00');
INSERT INTO SHOW VALUES('S-004', 'C4', 'Fly amongst Butterflies', '15:15');
INSERT INTO SHOW VALUES('S-005', 'D1', 'Circus', '16:00');
INSERT INTO SHOW VALUES('S-006', 'C3', 'Lions Attack','17:45');
INSERT INTO SHOW VALUES('S-007', 'C2','Elephant's Art','13:15');
INSERT INTO SHOW VALUES('S-008',  'C4', 'Birds in flight','14:30');
INSERT INTO SHOW VALUES('S-009',  'D3', 'The Seal Game','15:30');
```

9. SCHEDULE:

```
INSERT INTO SCHEDULE VALUES('E-00001','Part-time','MWF','12PM-5PM','4');
INSERT INTO SCHEDULE VALUES('E-00002','Fulltime','MTWRF','10AM-7PM','2');
INSERT INTO SCHEDULE VALUES('E-00003','Fulltime','MTWRF','10AM-7PM','1');
INSERT INTO SCHEDULE VALUES('E-00004','Part-time','TR','10AM-5PM','0');
INSERT INTO SCHEDULE VALUES('E-00005','Fulltime','MTWRF','10AM-7PM','0');
INSERT INTO SCHEDULE VALUES('E-00006','Fulltime','MTWRF','10AM-7PM','3');
INSERT INTO SCHEDULE VALUES('E-00007','Part-time','MW','2PM-7PM','4');
INSERT INTO SCHEDULE VALUES('E-00008','Fulltime','MTWRF','10AM-7PM','1');
INSERT INTO SCHEDULE VALUES('E-00009','Fulltime','MTWRF','10AM-7PM','1');
INSERT INTO SCHEDULE VALUES('E-00010','Fulltime','MTWRF','10AM-7PM','0');
INSERT INTO SCHEDULE VALUES('E-00011','Part-time','RF','10AM-6PM','2');
INSERT INTO SCHEDULE VALUES('E-00012','Fulltime','MTWRF','10AM-7PM','3');
INSERT INTO SCHEDULE VALUES('E-00013','Part-time','MWF','10AM-7PM','2');
INSERT INTO SCHEDULE VALUES('E-00014','Fulltime','MTWRF','10AM-7PM','0');
INSERT INTO SCHEDULE VALUES('E-00015','Part-time','MF','12AM-7PM','0');
INSERT INTO SCHEDULE VALUES('E-00016','Fulltime','MTWRF','10AM-7PM','3');
INSERT INTO SCHEDULE VALUES('E-00017','Part-time','MTF','11PM-6PM','4');
INSERT INTO SCHEDULE VALUES('E-00018','Fulltime','MTWRF','10AM-7PM','1');
INSERT INTO SCHEDULE VALUES('E-00019','Part-time','MTWR','10PM-5PM','1');
INSERT INTO SCHEDULE VALUES('E-00020','Fulltime','MTWRF','10AM-7PM','0');
```

10. MEDICAL_CARE:

11.

```
INSERT INTO MEDICAL_CARE VALUES ('A-001', DATE'2020-01-01', '3 months');
INSERT INTO MEDICAL_CARE VALUES ('A-002', DATE'2020-08-09', '3 months');
INSERT INTO MEDICAL_CARE VALUES ('A-003', DATE'2020-10-11', '6 months') ;
INSERT INTO MEDICAL_CARE VALUES ('A-004', DATE'2020-10-11', '6 months');
INSERT INTO MEDICAL_CARE VALUES ('A-005', DATE'2020-08-09', '2 months');
INSERT INTO MEDICAL_CARE VALUES ('A-006', DATE'2020-08-09', '2 months');
INSERT INTO MEDICAL_CARE VALUES ('A-008', DATE'2020-10-12', '3 weeks');
INSERT INTO MEDICAL_CARE VALUES ('A-009', DATE'2020-08-09', '6 weeks');
INSERT INTO MEDICAL_CARE VALUES ('A-010', DATE'2020-10-11', '6 weeks') ;
INSERT INTO MEDICAL_CARE VALUES ('A-011', DATE'2020-08-09', '6 weeks');
INSERT INTO MEDICAL_CARE VALUES ('A-012', DATE'2020-10-01', '1 month');
```

12. TAKEN VACCINES:

```
INSERT INTO T_VACCINE VALUES (‘A-001’, NULL, NULL);
INSERT INTO T_VACCINE VALUES (‘A-002’, ‘09-08-2020', ‘Rabies’);
INSERT INTO T_VACCINE VALUES (‘A-002’, ‘09-08-2020', ‘Rabies’);
INSERT INTO T_VACCINE VALUES (‘A-003’, ‘11-10-2020', ‘Distemper’);
INSERT INTO T_VACCINE VALUES (‘A-003’, ‘11-10-2020', ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-004’, ‘11-10-2020', ‘Distemper’);
INSERT INTO T_VACCINE VALUES (‘A-004’, ‘11-10-2020', ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-005’, ‘08-09-2020', ‘Chlamydia’);
INSERT INTO T_VACCINE VALUES (‘A-006’, “08-09-2020, ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-007’, NULL, NULL);
INSERT INTO T_VACCINE VALUES (‘A-008’, ‘10-12-2020', ‘Feline leukemia’);
INSERT INTO T_VACCINE VALUES (‘A-008’, ‘10-12-2020', ‘Bordetella’);
INSERT INTO T_VACCINE VALUES (‘A-009’, ‘09-08-2020', ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-010’, ‘11-10-2020', ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-011’, ‘08-09-2020', ‘Influenza’);
INSERT INTO T_VACCINE VALUES (‘A-012’, ‘01-10-2020', ‘DHPP’);
INSERT INTO T_VACCINE VALUES (‘A-013’, ‘11-03-2020', ‘DHPP’);
INSERT INTO T_VACCINE VALUES (‘A-014’, ‘05-10-2020', ‘DHPP’);
INSERT INTO T_VACCINE VALUES (‘A-015’, ‘12-11-2020', ‘Leptospira’);
INSERT INTO T_VACCINE VALUES (‘A-015’, ‘12-11-2020', ‘Feline leukemia’);
INSERT INTO T_VACCINE VALUES (‘A-016’, NULL, NULL);
INSERT INTO T_VACCINE VALUES (‘A-017’, ‘11-03-2020', ‘Bordetella’);
INSERT INTO T_VACCINE VALUES (‘A-018’, ‘05-10-2020', ‘Lyme’);
INSERT INTO T_VACCINE VALUES (‘A-019’, ‘10-12-2020', ‘Lyme’);
INSERT INTO T_VACCINE VALUES (‘A-020’, NULL, NULL);
INSERT INTO T_VACCINE VALUES (‘A-021’, NULL, NULL);
```

13. NEEDED VACCINES:

```
INSERT INTO N_VACCINE VALUES (‘A-001’, NULL, ‘Rabies’);
INSERT INTO N_VACCINE VALUES (‘A-001’, NULL, ‘Lyme’);
INSERT INTO N_VACCINE VALUES (‘A-001’, NULL, ‘Feline parvovirus’);
INSERT INTO N_VACCINE VALUES (‘A-002’, ‘11-11-2020', ‘Feline parvovirus’);
INSERT INTO N_VACCINE VALUES (‘A-002’, ‘11-11-2020', ‘Lyme’);
INSERT INTO N_VACCINE VALUES (‘A-005’, ‘08-10-2020', ‘Parainfluenza’);
INSERT INTO N_VACCINE VALUES (‘A-006’, ‘12-06-2020', ‘Parainfluenza’);
INSERT INTO N_VACCINE VALUES (‘A-007’, NULL, ‘Bordetella’);
INSERT INTO N_VACCINE VALUES (‘A-012’, ‘04-08-2020', ‘Feline leukemia’);
INSERT INTO N_VACCINE VALUES (‘A-013’, ‘08-10-2020', ‘Feline leukemia’);
INSERT INTO N_VACCINE VALUES (‘A-016’, NULL, ‘Leptospira’);
INSERT INTO N_VACCINE VALUES (‘A-018’, ‘04-08-2020', ‘Feline herpesvirus’);
INSERT INTO N_VACCINE VALUES (‘A-019’, ‘11-11-2020', ‘Feline parvovirus’);
INSERT INTO N_VACCINE VALUES (‘A-020’, NULL, ‘Chlamydia Felisa’);
```

14. FACILITY SERVICES:

```
INSERT INTO FACILITY_SERVICES VALUES ('F-00', 'Toilets');
INSERT INTO FACILITY_SERVICES VALUES ('F-01', 'Dine-in');
INSERT INTO FACILITY_SERVICES VALUES ('F-01', 'Host birthdays');
INSERT INTO FACILITY_SERVICES VALUES ('F-02', 'Null');
INSERT INTO FACILITY_SERVICES VALUES ('F-04', 'Sell souvenirs');
INSERT INTO FACILITY_SERVICES VALUES ('F-04', 'Quiet area');
INSERT INTO FACILITY_SERVICES VALUES ('F-05', 'Children activities');
INSERT INTO FACILITY_SERVICES VALUES ('F-05', 'Baby care room');
INSERT INTO FACILITY_SERVICES VALUES ('F-06', 'Null');
INSERT INTO FACILITY_SERVICES VALUES ('F-07', 'Pet shop');
INSERT INTO FACILITY_SERVICES VALUES ('F-07', 'Food for zoo animals');
INSERT INTO FACILITY_SERVICES VALUES ('F-08', 'Zoo tour');
INSERT INTO FACILITY_SERVICES VALUES ('F-08', 'Inquiry center');
INSERT INTO FACILITY_SERVICES VALUES ('F-09', 'Donate to Recycle');
```

15. VISITS:

```
INSERT INTO  VISITS VALUES('20201112-0123','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0124','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0125','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0126','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0127','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0128','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0129','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0130','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0131','Z_0','Lebanon,Beirut');
INSERT INTO VISITS VALUES('20201112-0132','Z_1','UAE,Dubai');
INSERT INTO VISITS VALUES('20201112-0133','Z_1','UAE,Dubai');
INSERT INTO VISITS VALUES('20201112-0134','Z_8','Jordan,Amman');
INSERT INTO VISITS VALUES('20201112-0135','Z_8','Jordan,Amman');
INSERT INTO VISITS VALUES('20201112-0136','Z_3','Iraq,Baghdad');
INSERT INTO VISITS VALUES('20201112-0137','Z_4','Syria,Damascus');
INSERT INTO VISITS VALUES('20201112-0138','Z_4','Syria,Damascus');
INSERT INTO VISITS VALUES('20201112-0139','Z_4','Syria,Damascus');
```

16. FEATURES:

```
INSERT INTO FEATURES VALUES ('S-000', 'A-012', 'daily');
INSERT INTO FEATURES VALUES ('S-000', 'A-013', 'daily');
INSERT INTO FEATURES VALUES ('S-001', 'A-014', 'daily');
INSERT INTO FEATURES VALUES ('S-002', 'A-009', 'weekly');
INSERT INTO FEATURES VALUES ('S-002', 'A-010', 'weekly');
INSERT INTO FEATURES VALUES ('S-002', 'A-011', 'weekly');
INSERT INTO FEATURES VALUES ('S-003', 'A-007', 'daily');
INSERT INTO FEATURES VALUES ('S-003', 'A-008', 'daily');
INSERT INTO FEATURES VALUES ('S-004', 'A-021', 'daily');
INSERT INTO FEATURES VALUES ('S-005', 'A-005', 'weekly');
INSERT INTO FEATURES VALUES ('S-005', 'A-003', 'weekly');
INSERT INTO FEATURES VALUES ('S-006', 'A-005', 'daily');
INSERT INTO FEATURES VALUES ('S-006', 'A-006', 'daily');
INSERT INTO FEATURES VALUES ('S-007', 'A-004', 'weekly');
INSERT INTO FEATURES VALUES ('S-008', 'A-020', 'weekly');
INSERT INTO FEATURES VALUES ('S-009', 'A-015', 'weekly');
```

17. CONTAINS:

```
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-001', 'Panda', 2010, 2015, 'M', '80kg', 'Herbivorous');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-002', 'Panda', 2007, 2015, 'F', '75kg', 'Herbivorous');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-003','Asian Elephant',2007,2015, 'F', '3500kg', 'Herbivorous');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-004', 'Asian Elephant', 1990, 2010, 'M',
'8000kg', 'Herbivorous' );
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-005', 'Lion', 2015, 2020,
'M','100kg','Carnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-006', 'Lion', 2015, 2019,
'F','90kg','Carnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-007', 'Penguin',2019,
2020,'F','22kg','Fish');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-008', 'Penguin',2018,
2020,'F','20kg','Fish');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-009', 'Monkey',2008,
2011,'M','20kg','Omnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-010', 'Monkey',2007,
2010,'M','27kg','Omnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-011',
'Dolphin',2000,2010,'F','3000kg','Carnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-012', 'Dolphin',2002,
2009,'F','3700kg','Carnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-013',
'Dolphin',1998,2009,'M','4000kg','Carnivore');
INSERT INTO CONTAINS VALUES(‘Z-0’,'A-014', 'Monkey',2007,
2010,'F','20kg','Omnivore');
INSERT INTO CONTAINS VALUES('Z_0','A-015', 'Seal',2001,
2005,'F','50kg','Carnivore');
INSERT INTO CONTAINS VALUES('Z_0','A-016', 'Sloth',2006,
2008,'M','5kg','Omnivore');
INSERT INTO CONTAINS VALUES('Z_0','A-017', 'Crocodile',1970,
2005,'M','250kg','Carnivore');
INSERT INTO CONTAINS VALUES('Z_0','A-018', 'Rabbit',2020,
2020,'M','1kg','Herbivore');
INSERT INTO CONTAINS VALUES('Z_0','A-019', 'Rabbit',2020,
2020,'F','1kg','Herbivore');
INSERT INTO CONTAINS VALUES('Z_0','A-020', 'Golden Eagle',2005,
2015,'F','4kg','Carnivore');
```

```
INSERT INTO CONTAINS VALUES('Z_0','A-021', 'Buttefly',2020,2020,'F','','Herbivore');
```

18. WORKS IN:

```
INSERT INTO WORKS_IN VALUES('D-04','E-00001','Noor Zaarour',DATE'1990-5-4','F','76735584','Baalbeck','Bk-638','Al Turath','5','Zookeeper');
INSERT INTO WORKS_IN VALUES('D-00','E-00002','Hussein Muhsen',DATE'1996-12-10','M','76296411','Beirut','B-861','666 Meters','6','Manager');
INSERT INTO WORKS_IN VALUES('D-01','E-00003','Zeinab Yahya',DATE'1997-8-27','F','76863278','Beirut','B-843','77 Doors','6','Manager');
INSERT INTO WORKS_IN VALUES('D-02','E-00004','Michelle Mourad',DATE'1988-10-10','M','76885377','Byblos','By-277','77 Doors','1','Accountant');
INSERT INTO WORKS_IN VALUES('D-02','E-00005','Tala Jurdi',DATE'1979-8-14','F','76196582','Joneih','J-945','Adonis 2','3','Manager');
INSERT INTO WORKS_IN VALUES('D-04','E-00006','Israa Haider',DATE'1981-4-13','F','76996357','Baalbeck','Bk-647','Al Turath','2','Manager');
INSERT INTO WORKS_IN VALUES('D-03','E-00007','Ibrahim kansour',DATE'1967-7-18','M','76015489','Tyre','T-190','OC 5','3','Researcher');
INSERT INTO WORKS_IN VALUES('D-05','E-00008','Rania Dakroub',DATE'1959-9-8','F','76551894','Beirut','B-682','the 990','1','Manager');
INSERT INTO WORKS_IN VALUES('D-09','E-00009','Noor Onaisi',DATE'1980-5-6','F','76945177','Beirut','B-624','Warda','2','Manager');
INSERT INTO WORKS_IN VALUES('D-08','E-000010','Hussein Rakka',DATE'1975-9-10','M','76268593','Tripoli','Ti-433','The 117','3','Manager');
INSERT INTO WORKS_IN VALUES('D-04','E-000011','Jad Baydoun',DATE'1992-9-1','M','76892547','Byblos','By-287','The Abroad','1','Zookeeper');
INSERT INTO WORKS_IN VALUES('D-01','E-000012','Mariam Zaytouni',DATE'1988-2-16','F','76327854','Beirut','B-611','Building 37','4','Vet');
INSERT INTO WORKS_IN VALUES('D-08','E-000013','Daniell Bannout',DATE'1983-9-9','M','76884138','Byblos','By-256','The Above','5','Fundraiser');
```

```sql
INSERT INTO WORKS_IN VALUES('D-06','E-000014','Natalie Nazzal',DATE'1989-1-30','F','76884138','Beirut','B-683','Al Manara','1','Manager');
INSERT INTO WORKS_IN VALUES('D-09','E-000015','Josephine bazzi',DATE'1988-3-11','F','76763299','Beirut','B-649','sky Gate','1','Training coordinator');
INSERT INTO WORKS_IN VALUES('D-05','E-000016','Sarah Chouman',DATE'1992-6-22','F','76886312','Beirut','B-693','Sama Beirut','3','Coordinator');
INSERT INTO WORKS_IN VALUES('D-04','E-000017','Jaber Soeid',DATE'1959-6-12','M','76184659','Kheyam','Kh-022','the 801','2','Inspector');
INSERT INTO WORKS_IN VALUES('D-03','E-000018','Mia Mansour',DATE'1966-8-13','F','76901145','Joneih','J-933','Adonis 7','4','Manager');
INSERT INTO WORKS_IN VALUES('D-00','E-000019','Lia Ashkar',DATE'1993-12-15','F','76564888','Beirut','B-638','Middle Eastern 66','3','Receptionist');
INSERT INTO WORKS_IN VALUES('D-07','E-000020','Ascia Hakeem',DATE'1998-5-5','F','76981437','Beirut','B-610','Mediterran 6','6','Manager');
```

19. BUYS:

```sql
INSERT INTO BUYS VALUES('11-0000001',  '20201112-0123', 33);
INSERT INTO BUYS VALUES('11-0000002',  '20201112-0124', 25);
INSERT INTO BUYS VALUES('01-0000001',  '20201112-0125',5);
INSERT INTO BUYS VALUES('10-0000001',  '20201112-0126',40);
INSERT INTO BUYS VALUES('10-0000002',  '20201112-0127',35);
INSERT INTO BUYS VALUES('10-0000003',  '20201112-0128',14);
INSERT INTO BUYS VALUES('01-0000002',  '20201112-0129',10);
INSERT INTO BUYS VALUES('01-0000003',  '20201112-0130',7);
INSERT INTO BUYS VALUES('01-0000004',  '20201112-0131',5);
INSERT INTO BUYS VALUES('11-0000003',  '20201112-0132',31);
INSERT INTO BUYS VALUES('11-0000004',  '20201112-0133',20);
INSERT INTO BUYS VALUS('10-0000004',    '20201112-0134',24);
INSERT INTO BUYS VALUES('00-0000001',  '20201112-0135',2);
INSERT INTO BUYS VALUES('10-0000005',  '20201112-0136',60);
INSERT INTO BUYS VALUES('10-0000006',  '20201112-0137',53);
INSERT INTO BUYS VALUES('10-0000007',  '20201112-0138',41);
INSERT INTO BUYS VALUES('00-0000002',  '20201112-0138',10);
INSERT INTO BUYS VALUES('00-0000003',    '20201112-0139',9);
```

20. MANAGES:

```
INSERT INTO MANAGES VALUES('D-05', 'S-000', '11-11-2020 11:00:00', 'D3',
'Dolphins Show');
INSERT INTO MANAGES VALUES('D-05',  'S-001', '10-11-2020 12:30:00', 'D4', 'Tour
in the Aquarium');
INSERT INTO MANAGES VALUES('D-04',    'S-002', '10-11-2020 13:00:00',
    'C2',     'Monkeys Battle' );
INSERT INTO MANAGES VALUES('D-05',    'S-003', '11-11-2020 14:00:00',
    'D2',     'Penguins and beyond' );
INSERT INTO MANAGES VALUES('D-04',    'S-004', '11-11-2020 15:15:00',
    'C4',     'Fly amongst Butterflies');
INSERT INTO MANAGES VALUES('D-04',    'S-005', '12-11-2020 16:00:00',
    'D1',     'Circus');
INSERT INTO MANAGES VALUES('D-04',    'S-006', '12-11-2020 17:45:00',
    'C3',     'Lions Attack');
INSERT INTO MANAGES VALUES('D-04',    'S-007', '13-11-2020 13:15:00',
    'C2',     'Elephant’s Art');
INSERT INTO MANAGES VALUES('D-04',    'S-008', '13-11-2020 14:30:00',
    'C4',     'Birds in flight') ;
INSERT INTO MANAGES VALUES('D-04',    'S-009', '14-11-2020 15:45:00',
    'D3',     'The Seal Game');
```

21. RUNS:

```
INSERT INTO RUNS VALUES ('D-06', 'F-00');
INSERT INTO RUNS VALUES ('D-06', 'F-01');
INSERT INTO RUNS VALUES ('D-06', 'F-02');
INSERT INTO RUNS VALUES ('D-06', 'F-03');
INSERT INTO RUNS VALUES ('D-06', 'F-04');
INSERT INTO RUNS VALUES ('D-04', 'F-05');
INSERT INTO RUNS VALUES ('D-06', 'F-06');
INSERT INTO RUNS VALUES ('D-04', 'F-07');
INSERT INTO RUNS VALUES ('D-06', 'F-08');
INSERT INTO RUNS VALUES ('D-00', 'F-09');
```

X. Final Tables State:

1. ANIMAL:

A_ID	A_NAME	A_Birthdate	Entry_date	Sex	Weight	Diet
A-001	Panda	2010	2015	M	80kg	Herbivorous
A-002	Panda	2007	2015	F	75Kg	Herbivorous
A-003	Asian Elephant	1980	2010	F	3500kg	Herbivorous
A-004	Asian Elephant	1990	2010	M	3400kg	Herbivorous
A-005	Lion	2015	2018	F	90kg	Carnivore
A-006	Lion	2014	2017	M	100Kg	Carnivore
A-007	Penguin	2019	2020	F	22kg	Fish
A-008	Penguin	2018	2020	F	20kg	Fish
A-009	Monkey	2010	2011	M	24kg	Omnivore
A-010	Monkey	2008	2011	M	27kg	Omnivore
A-011	Monkey	2007	2010	F	20kg	Omnivore
A-012	Dolphin	2000	2010	M	3000kg	Carnivore
A-013	Dolphin	2002	2009	M	3700kg	Carnivore
A-014	Dolphin	1998	2009	F	4000kg	Carnivore
A-015	Seal	2001	2005	F	50kg	Carnivore
A-016	Sloth	2006	2008	M	5kg	Omnivore
A-017	Crocodile	1970	2005	M	250kg	Carnivore
A-018	Rabbit	2020	2020	F	1kg	Herbivore

A-019	Rabbit	2020	2020	M	1 kg	Herbivore
A-020	Golden eagle	2005	2015	F	4kg	Carnivore
A-021	Butterfly	2020	2020	F	-	Herbivore

2. VISITOR:

V_ID	Age
20201112-0123	33
20201112-0124	25
20201112-0125	5
20201112-0126	40
20201112-127	35
20201112-0128	14
20201112-0129	10
20201112-0130	7
20201112-0131	5
20201112-0132	31
20201112-0133	20
20201112-0134	24
20201112-0135	2
20201112-0136	60
20201112-0137	53
20201112-0138	41
20201112-0138	10
20201112-0139	9

3. ZOO_BRANCH:

Z_ID	Location
Z-0	Lebanon, Beirut
Z-1	UAE, Dubai
Z-2	UAE, Abu Dhabi
Z-3	Iraq, Baghdad
Z-4	Syria, Damascus
Z-5	KSA, Riyad
Z-6	Morocco, Rabat

Z-7	Egypt, Cairo
Z-8	Jordan, Amman
Z-9	Qatar, Doha

4. TICKET:

T_ID	Price	Date	Validity	Type
11-0000001	15$	12 Nov. 2020	12h	Prime
11-0000002	15$	12 Nov. 2020	12h	Prime
11-0000003	15$	12 Nov. 2020	12h	Prime
11-0000004	15$	12 Nov. 2020	12h	Prime
10-0000001	10$	11 Nov. 2020	8h	Regular
10-0000002	10$	11 Nov. 2020	8h	Regular
10-0000003	10$	11 Nov. 2020	8h	Regular
10-0000004	10$	11 Nov. 2020	8h	Regular
10-0000005	10$	11 Nov. 2020	8h	Regular
10-0000006	10$	11 Nov. 2020	8h	Regular
10-0000007	10$	11 Nov. 2020	8h	Regular
01-0000001	10$	12 Nov. 2020	12h	K-Prime
01-0000002	10$	12 Nov. 2020	12h	K-Prime
01-0000003	10$	12 Nov. 2020	12h	K-Prime
01-0000004	10$	12 Nov. 2020	12h	K-Prime
00-0000001	7$	11 Nov. 2020	8h	K-Regular
00-0000002	7$	11 Nov. 2020	8h	K- Regular
00-0000003	7$	11 Nov. 2020	8h	K- Regular

5. DEPARTMENT:

D_ID	D_Name	Manager
D-00	Administration	Hussain Muhsen
D-01	Health & Safety	Zainab Yahya
D-02	Finance	Tala Jurdi
D-03	Conservation & Research	Mia Mansour
D-04	Animal Care	Israa Haidar
D-05	Aquatic	Rania Dakroub
D-06	Visitors & Customer Services	Nathalie Nazzal

D-07	Marketing	Ascia Hakeem
D-08	Public Relations	Hussein Rakka
D-09	Human Resources	Nour Onaissi

6. EMPLOYEE:

E_ID	Fullname	gender	E-birthdate	Phone number	city	street	Building	Floor
E-00001	Noor Zaarour	F	1990/5/4	76735584	Baalbeck	Bk-638	Al Turath	5
E-00002	Hussein Muhsen	M	1996/12/10	76296411	Beirut	B-861	666 Meters	6
E-00003	Zeinab Yahya	F	1997/8/27	76863278	Beirut	B-843	77 Doors	6
E-00004	Michelle Mourad	M	1988/10/10	76885377	Byblos	By-277	Angelina 51	1
E-00005	Tala Jurdi	F	1979/8/14	76196582	Joneih	J-945	Adonis 2	3
E-00006	Israa Haider	F	1981/4/13	76996357	Baalbek	Bk-647	Al Turath	2
E-00007	Ibrahim kansour	M	1967/7/18	76015489	Tyre	T-190	OC 5	3
E-00008	Rania Dakroub	F	1959/9/8	76551894	Beirut	B-682	the 990	1
E-00009	Noor Onaisi	F	1980/5/6	76945177	Beirut	B-624	Warda	2
E-00010	Hussein Rakka	M	1975/9/10	76268593	Tripoli	Ti-433	The 117	3
E-00011	Jad Baydoun	M	1992/9/1	76892547	Byblos	By-287	The Abroad	1
E-00012	Mariam Zaytouni	F	1988/2/16	76327854	Beirut	B-611	Building 37	4
E-00013	Daniell Bannout	M	1983/9/9	76884138	Byblos	By-256	The Above	5

E-00014	Natalie Nazzal	F	1989/1/30	76865148	Beirut	B-683	Al Manara	1
E-00015	Josephine bazzi	F	1988/3/11	76763299	Beirut	B-649	sky Gate	1
E-00016	Sarah Chouman	F	1992/6/22	76886312	Beirut	B-693	Sama Beirut	3
E-00017	Jaber Soeid	M	1959/6/12	76184659	Kheyam	Kh-022	the 801	2
E-00018	Mia Mansour	F	1966/8/13	76901145	Joneih	J-933	Adonis 7	4
E-00019	Lia Ashkar	F	1993/12/15	76564888	Beirut	B-638	Middle Eastern 66	3
E-00020	Ascia Hakeem	F	1998/5/5	76981437	Beirut	B-610	Mediterran 6	6

7. FACILITY:

F_ID	Type	Location
F-00	Restrooms	A1
F-01	Restaurant	B1
F-02	Café	B1
F-03	Giftshop	A2
F-04	Prayer Room	A1
F-05	Children Care	A1
F-06	Parking lot	A4
F-07	The Animal Store	C1
F-08	Tour Guide	A3
F-09	Donation Center	A2

8. SHOW:

S_ID	Time (24hrs system)	Place	Subject
S-000	11:00	D3	Dolphins’ show
S-001	12:30	D4	Tour in the Aquarium
S-002	13:00	C2	Monkeys Battle
S-003	14:00	D2	Penguins and beyond

S-004	15:15	C4	Fly amongst Butterflies
S-005	16:00	D1	Circus
S-006	17:45	C3	Lions Attack
S-007	13:15	C2	Elephant’s Art
S-008	14:30	C4	Birds in flight
S-009	15:45	D3	The Seal Game

9. SCHEDULE:

FK_E_ID	Type	Working_Days	Working_Hours	Extra_Hours
E-00001	Part-time	MWF	12PM-5PM	4
E-00002	Fulltime	MTWRF	10AM-7PM	2
E-00003	Fulltime	MTWRF	10AM-7PM	1
E-00004	Part-time	TR	10AM-5PM	0
E-00005	Fulltime	MTWRF	10AM-7PM	0
E-00006	Fulltime	MTWRF	10AM-7PM	3
E-00007	Part-time	MW	2PM-7PM	4
E-00008	Fulltime	MTWRF	10AM-7PM	1
E-00009	Fulltime	MTWRF	10AM-7PM	1
E-00010	Fulltime	MTWRF	10AM-7PM	0
E-00011	Part-time	RF	10AM-6PM	2
E-00012	Fulltime	MTWRF	10AM-7PM	3
E-00013	Part-time	MWF	10AM-7M	2
E-00014	Fulltime	MTWRF	10AM-7PM	0
E-00015	Part-time	MF	12PM-7PM	0
E-00016	Fulltime	MTWRF	10AM-7PM	3
E-00017	Part-time	MTF	11AM-6PM	4
E-00018	Fulltime	MTWRF	10AM-7PM	1
E-00019	Part-time	MTWR	10AM-5PM	1
E-00020	Fulltime	MTWRF	10AM-7PM	0

10. MEDICAL_CARE:

FK_A_ID	Last_Checkup	Checkup_Duration
A-001	NULL	3 months
A-002	09-08-2020	3 months

A-003	11-10-2020	6 months
A-004	11-10-2020	6 months
A-005	08-09-2020	2 months
A-006	08-09-2020	2 months
A-007	NULL	3 weeks
A-008	10-12-2020	3 weeks
A-009	09-08-2020	6 weeks
A-010	11-10-2020	6 weeks
A-011	08-09-2020	6 weeks
A-012	01-10-2020	1 month
A-013	11-03-2020	1 month
A-014	05-10-2020	1 month
A-015	12-11-2020	1 month
A-016	NULL	1 month
A-017	11-03-2020	3 months
A-018	05-10-2020	1 month
A-019	10-12-2020	1 month
A-020	NULL	6 months
A-021	NULL	NULL

11. TAKEN VACCINES:

Animal_ID	Last_Checkup_Date	Vaccine_done
A-001	NULL	NULL
A-002	09-08-2020	Rabies
A-002	09-08-2020	Rabies
A-003	11-10-2020	Distemper
A-003	11-10-2020	Influenza
A-004	11-10-2020	Distemper
A-004	11-10-2020	Influenza
A-005	08-09-2020	Chlamydia
A-006	08-09-2020	Influenza
A-007	NULL	NULL

A-008	10-12-2020	Feline leukemia
A-008	10-12-2020	Bordetella
A-009	09-08-2020	Influenza
A-010	11-10-2020	Influenza
A-011	08-09-2020	Influenza
A-012	01-10-2020	DHPP
A-013	11-03-2020	DHPP
A-014	05-10-2020	DHPP
A-015	12-11-2020	Leptospira
A-015	12-11-2020	Feline leukemia
A-016	NULL	NULL
A-017	11-03-2020	Bordetella
A-018	05-10-2020	Lyme
A-019	10-12-2020	Lyme
A-020	NULL	NULL
A-021	NULL	NULL

12. NEEDED VACCINES:

Animal_Nu	Lastly_checked_up	Vaccine_to_be_taken
A-001	NULL	Rabies
A-001	NULL	Lyme
A-001	NULL	Feline parvovirus
A-002	09-08-2020	Feline parvovirus
A-002	09-08-2020	Lyme
A-005	08-09-2020	Parainfluenza
A-006	08-09-2020	Parainfluenza
A-007	NULL	Bordetella
A-012	01-10-2020	Feline leukemia
A-013	11-03-2020	Feline leukemia
A-016	NULL	Leptospira
A-018	05-10-2020	Feline herpesvirus
A-019	10-12-2020	Feline herpesvirus

A-020	NULL	Chlamydia Felisa

13. FACILITY SERVICES:

F_Number	Service
F-00	Toilets
F-01	Dine-in
F-01	Host birthdays
F-02	Null
F-03	Sell souvenirs
F-04	Quiet area
F-05	Children activities
F-05	Baby care room
F-06	Null
F-07	Pet shop
F-07	Food for zoo animals
F-08	Zoo tour
F-08	Inquiry center
F-09	Donate to Recycle

14. VISITS:

V_Number	Z_ID	Location
20201112-0123	Z-0	Lebanon, Beirut
20201112-0124	Z-0	Lebanon, Beirut
20201112-0125	Z-0	Lebanon, Beirut
20201112-0126	Z-0	Lebanon, Beirut
20201112-127	Z-0	Lebanon, Beirut
20201112-0128	Z-0	Lebanon, Beirut
20201112-0129	Z-0	Lebanon, Beirut
20201112-0130	Z-0	Lebanon, Beirut
20201112-0131	Z-0	Lebanon, Beirut
20201112-0132	Z-1	UAE, Dubai

20201112-0133	Z-1	UAE, Dubai
20201112-0134	Z-8	Jordan, Amman
20201112-0135	Z-8	Jordan, Amman
20201112-0136	Z-3	Iraq, Baghdad
20201112-0137	Z-4	Syria, Damascus
20201112-0138	Z-4	Syria, Damascus
20201112-0138	Z-4	Syria, Damascus
20201112-0139	Z-4	Syria, Damascus

15. FEATURES:

T_ID	V_ID	Age
11-0000001	20201112-0123	33
11-0000002	20201112-0124	25
01-0000001	20201112-0125	5
10-0000001	20201112-0126	40
10-0000002	20201112-127	35
10-0000003	20201112-0128	14
01-0000002	20201112-0129	10
01-0000003	20201112-0130	7
01-0000004	20201112-0131	5
11-0000003	20201112-0132	31
11-0000004	20201112-0133	20
10-0000004	20201112-0134	24
00-0000001	20201112-0135	2
10-0000005	20201112-0136	60
10-0000006	20201112-0137	53
10-0000007	20201112-0138	41
00-0000002	20201112-0138	10
00-0000003	20201112-0139	9

16. CONTAINS:

Z_Number	A_ID	A_NAME	A_Birthdate	Entry_date	Sex	Weight	Diet
Z-0	A-001	Panda	2010	2015	M	80kg	Herbivore

Z-0	A-002	Panda	2007	2015	F	75Kg	Herbivore
Z-0	A-003	Asian Elephant	1980	2010	F	3500kg	Herbivore
Z-0	A-004	Asian Elephant	1990	2010	M	3400kg	Herbivore
Z-0	A-005	Lion	2015	2018	F	90kg	Carnivore
Z-0	A-006	Lion	2014	2017	M	100Kg	Carnivore
Z-0	A-007	Penguin	2019	2020	F	22kg	Fish
Z-0	A-008	Penguin	2018	2020	F	20kg	Fish
Z-0	A-009	Monkey	2010	2011	M	24kg	Omnivore
Z-0	A-010	Monkey	2008	2011	M	27kg	Omnivore
Z-0	A-011	Monkey	2007	2010	F	20kg	Omnivore
Z-0	A-012	Dolphin	2000	2010	M	3000kg	Carnivore
Z-0	A-013	Dolphin	2002	2009	M	3700kg	Carnivore
Z-0	A-014	Dolphin	1998	2009	F	4000kg	Carnivore
Z-0	A-015	Seal	2001	2005	F	50kg	Carnivore
Z-0	A-016	Sloth	2006	2008	M	5kg	Omnivore
Z-0	A-017	Crocodile	1970	2005	M	250kg	Carnivore
Z-0	A-018	Rabbit	2020	2020	F	1kg	Herbivore
Z-0	A-019	Rabbit	2020	2020	M	1 kg	Herbivore
Z-0	A-020	Golden eagle	2005	2015	F	4kg	Carnivore
Z-0	A-021	Butterfly	2020	2020	F	-	Herbivore

17. WORKS IN:

Dprt_Number	E_ID	Fullname	gender	E-birthdate	Phone number	city	street	Building	Floor	Position
D-04	E-00001	Noor Zaarour	F	1990/5/4	76735584	Baalbeck	Bk-638	Al Turath	5	Zookeeper
D-00	E-00002	Hussein Muhsen	M	1996/12/10	76296411	Beirut	B-861	666 Meters	6	Manager
D-01	E-00003	Zeinab Yahya	F	1997/8/27	76863278	Beirut	B-843	77 Doors	6	Manager
D-02	E-00004	Michelle Mourad	M	1988/10/10	76885377	Byblos	By-277	Angelina 51	1	Accountant
D-02	E-00005	Tala Jurdi	F	1979/8/14	76196582	Joneih	J-945	Adonis 2	3	Manager

D-04	E-00006	Israa Haider	F	1981/4/13	76996357	Baalbek	Bk-647	Al Turath	2	Manager
D-03	E-00007	Ibrahim kansour	M	1967/7/18	76015489	Tyre	T-190	OC 5	3	Researcher
D-05	E-00008	Rania Dakroub	F	1959/9/8	76551894	Beirut	B-682	the 990	1	Manager
D-09	E-00009	Noor Onnaisi	F	1980/5/6	76945177	Beirut	B-624	Warda	2	Manager
D-08	E-00010	Hussein Rakka	M	1975/9/10	76268593	Tripoli	Ti-433	The 117	3	Manager
D-04	E-00011	Jad Baydoun	M	1992/9/1	76892547	Byblos	By-287	The Abroad	1	Zookeeper
D-01	E-00012	Mariam Zaytouni	F	1988/2/16	76327854	Beirut	B-611	Building 37	4	Vet
D-08	E-00013	Daniell Bannout	M	1983/9/9	76884138	Byblos	By-256	The Above	5	Fundraiser
D-06	E-00014	Natalie Nazzal	F	1989/1/30	76865148	Beirut	B-683	Al Manara	1	Manager
D-09	E-00015	Josephine bazzi	F	1988/3/11	76763299	Beirut	B-649	sky Gate	1	Training coordinator
D-05	E-00016	Sarah Chouman	F	1992/6/22	76886312	Beirut	B-693	Sama Beirut	3	Coordinator
D-04	E-00017	Jaber Soeid	M	1959/6/12	76184659	Kheyam	Kh-022	the 801	2	Inspector
D-03	E-00018	Mia Mansour	F	1966/8/13	76901145	Joneih	J-933	Adonis 7	4	Manager
D-00	E-00019	Lia Ashkar	F	1993/12/15	76564888	Beirut	B-638	Middle Eastern 66	3	Receptionist
D-07	E-00020	Ascia Hakeem	F	1998/5/5	76981437	Beirut	B-610	Mediterran 6	6	Manager

18. BUYS:

T_ID	V_ID	Age
11-0000001	20201112-0123	33
11-0000002	20201112-0124	25

01-0000001	20201112-0125	5
10-0000001	20201112-0126	40
10-0000002	20201112-127	35
10-0000003	20201112-0128	14
01-0000002	20201112-0129	10
01-0000003	20201112-0130	7
01-0000004	20201112-0131	5
11-0000003	20201112-0132	31
11-0000004	20201112-0133	20
10-0000004	20201112-0134	24
00-0000001	20201112-0135	2
10-0000005	20201112-0136	60
10-0000006	20201112-0137	53
10-0000007	20201112-0138	41
00-0000002	20201112-0138	10
00-0000003	20201112-0139	9

19. MANAGES:

Dpr_Number	S_ID	Time(24hrs system)	Place	Subject
D-05	S-000	11:00	D3	Dolphins’ show
D-05	S-001	12:30	D4	Tour in the Aquarium
D-04	S-002	13:00	C2	Monkeys Battle
D-05	S-003	14:00	D2	Penguins and beyond
D-04	S-004	15:15	C4	Fly amongst Butterflies
D-04	S-005	16:00	D1	Circus
D-04	S-006	17:45	C3	Lions Attack
D-04	S-007	13:15	C2	Elephant’s Art
D-04	S-008	14:30	C4	Birds in flight
D-04	S-009	15:45	D3	The Seal Game

20. RUNS:

Dprt_Number	F_Number
D-06	F-00
D-06	F-01

D-06	F-02
D-06	F-03
D-06	F-04
D-04	F-05
D-06	F-06
D-04	F-07
D-06	F-08
D-00	F-09

XI. Queries:

Query one:

There was an accident related to one of the employees getting too close to the crocodile and now the zoo needs to find the person responsible for the aftereffects of this event.

Code:

```
SELECT WI.FullName, WI.phoneNumber
FROM WORKS_IN WI
WHERE WI.FullName = D.MANAGER IN;
(
SELECT D.MANAGER
FROM DEPARTMENT D
WHERE D.NAME=’HEALTH & SAFETY’;
```

Query two:

One of the sloths in the zoo was experiencing some side a effects after recent vaccination and the vet responsible ordered a check up and was in need of the type of vaccine administered so he can do a proper inspection.

Code:

```
SELECT V.Lastly_checked_up  , A.A_ID
FROM T_VACCINE V, ANIMAL A
WHERE A.A_Name = ‘sloth’ AND V.ANIMAL_ID = A.A_ID;
```

Query three:

Emergency at the zoo, retrieve the phone number of all managers od departments.

Code:

```
SELECT W.Phonenumber, W.Position
FROM WORKS_IN W
WHERE Position='Manager';
```

Query four:

A new group of animals is delivered

Code:

```
INSERT INTO ANIMAL
VALUES('A-022', 'Cat', 2015, 2020, 'M', '5kg', 'Carnivore')
VALUES('A-023', 'Cat', 2013, 2020, 'F', '4kg', 'Carnivore')
VALUES('A-023', 'Squirrel', 2014, 2020, 'M', '2kg', 'Omnivores')
VLAUES('A-024', 'Squirrel', 2012, 2020, 'F', '3kg', 'Omnivores')
VALUES('A-025', 'Llama', 2010, 2020, 'M', '150kg', 'Herbivores')
VALUES('A-026', 'Llama', 2019, 2020, 'F', '9kg', 'Herbivores')
VALUES('A-027', 'Hoatzin', 2017, 2020, 'M', '1kg', 'Herbivores');
```

Query five:

An animal cage is left open for the third time, get the names of the zoo keepers and kick one of them because it is his third time.

Code:

```
PRINT 'Names of all zookeepers: ';
SELECT W.Fullname, W.Position
FROM WORKS_IN W
WHERE Position='Zookeeper';
PRINT 'After investigation, one of the zookeepers got kicked out.';
DELETE FROM WORKS_IN
WHERE Fullname='Jad Baydoun';
```

Query six:

A school trip ,30 more visitors, we add more shows in collaboration with the school

Code:

```
INSERT INTO VISITOR(V_ID, Age)
VALUES('20201112-0122', '12')
VALUES('20201112-0140', '15')
VALUES('20201112-0141', '16')
VALUES('20201112-0142', '11')
VALUES('20201112-0143', '11')
VALUES('20201112-0144', '17')
VALUES('20201112-0145', '17')
VALUES('20201112-0146', '12')
VALUES('20201112-0147', '19')
VALUES('20201112-0148', '20')
VALUES('20201112-0149', '10')
VALUES('20201112-0150', '9')
VALUES('20201112-0151', '13')
VALUES('20201112-0152', '14')
VALUES('20201112-0153', '19')
VALUES('20201112-0154', '20')
VALUES('20201112-0155', '8')
VALUES('20201112-0156', '11')
VALUES('20201112-0157', '17')
VALUES('20201112-0158', '17')
VALUES('20201112-0159', '12')
VALUES('20201112-0160', '13')
VALUES('20201112-0161', '15')
VALUES('20201112-0162', '15')
VALUES('20201112-0163', '9')
VALUES('20201112-0164', '11')
VALUES('20201112-0165', '15')
VALUES('20201112-0166', '18')
VALUES('20201112-0167', '16')
VALUES('20201112-0168', '13')
VALUES('20201112-0169', '20')
VALUES('20201112-0170', '20')

SELECT FROM V.VISITOR
FROM VISITOR V
WHERE V = 'Age';

PRINT 'In collaboration with the school some new shows were added';

INSERT INTO MANAGES('Dpr_Number', 'S_ID', 'Time', 'Place', 'Subject')
VALUES('D-04', 'S-005', '16:00', 'D1', 'Elephant rides')
VALUES('D-05', 'S-001', '12:30', 'D4', 'Pandas' Day')

SELECT FROM M.Subject
FROM MANAGES M
WHERE Subject = 'Elephant rides', 'Pandas' Day';
```

Query seven:

A disease spread, add a vaccine for all herbivores

Code:

```
SELECT A.Diet, NV.N_VACCINE
FROM ANIMAL A, N_VACCINE NV
WHERE Diet = 'Herbivores', Vaccine_to_be_taken = 'Rabies';
PRINT 'All the animals got infected. Therefore, we need vaccine for all herbivores';
INSERT INTO N_VACCINE(Animal_Number, Lastly_checked_up, Vaccine_to_be_taken)
VALUES('A-001', 'NULL', 'Rabies')
```

Query eight:

A manager got infected with COVID-19 virus, so we have to call all the phone numbers of the managers that are in the same department to do a PCR test.

Code:

```
PRINT 'A manager in the department of the zoo tested positive for COVID-19, so all employees in the zoo must do the PCR test for safety purposes.'
SELECT W.Phonenumber, D.D_ID, W.Dprt_Number
FROM WORKS_IN AS W, DEPARTMENT AS D
WHERE D.D_ID=W.Dprt_Number AND W.Phonenumber;
```

Query nine:

A n employee has a new born and she should take a break

Code:

```
//GET THE EMPLOYEE ID
SELECT S.FK_E_ID
FROM SCHEDULE AS S
WHERE S.FKE_ID= ‘E-00012’;
// DELETE HER WORKING HOURS
DELETE FROM SCHEDULE
WHERE S.Working_hours= 10AM-7PM;
```

Query ten:

An explosion occurred and some animals passed away while others got injured

Code:

```
 DELETE FROM ANIMAL
WHERE Diet=’Fish’ AND ‘Omnivore’;
```

XII. Normalization Up to The BCNF Normal Form:

After creating all relations we should improve them by normalizing according to several normal forms. Here we are going to normalize our database up to the fourth normal form which is the Boyce-Codd Normal Form. On each relation we are going to apply the four normal forms. We start with the first then second then third and at last the BCNF normal form. Let us first start by a general description to each normal form.

First Normal Form:

This form does not allow multivalued attributes, composite attributes, and their combinations to exist in a relation.

1. Only attribute values permitted are single atomic values.
2. Domain of an attribute must only include atomic values and the value of an attribute in a tuple must be a single value from the domain of that attribute.
3. Disallows having a set of values as an attribute value for a single tuple.

Second Normal Form:

The Second normal form is based on the concept of full functional dependency. Before explaining the second form let us define some concepts used in this form and other forms also.

Functional Dependencies: A constraint between two sets of attributes from the database. The values of the Y component of a tuple in relation R depend on, or are determined by the values of an X component. We say that Y is functionally dependent on X.

Prime attribute: An attribute that is a member of a candidate key in a relation R. An attribute is called non-prime if it is not a prime attribute that is, if it is not a member of any candidate key.

Full functional dependency: A functional dependency X →Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore.

Partial Dependency: A functional dependency X →Y is a partial functional dependency if removal of any attribute A from X means that the dependency still holds.

A relation schema R is in the second normal form if every nonprime attribute in R is fully functionally dependent on the every key of R and every nonprime attribute A in R is not partially dependent on any key in R.

Third Normal Form:

The third normal form is based on the concept of transitive dependency. So let us first define a transitive dependency.

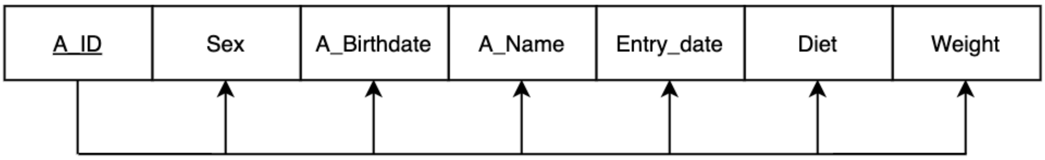
Transitive Dependency: A functional dependency X →Y in a relation schema R is a transitive dependency if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and both X →Z and Z →Y hold.

A relation schema R is in the third normal form if it satisfies the second normal form and no nonprime attribute of R is transitively dependent on the primary key. For every nontrivial functional dependency X →Y either X should be a super key or Y is a prime attribute.

Boycee-Codd Normal Form:

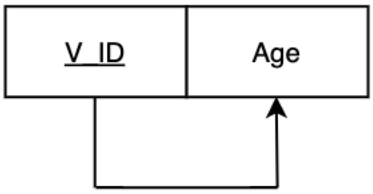
The BoyceeCodd normal form is a stricter form than the third normal form. The BCNF differs from the definition of the third normal form in only one condition. The third normal form allows the right hand side of the functional dependency to be a prime attribute while BCNF does not allow that.

1. ANIMAL:



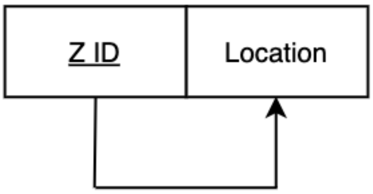
- 1NF is satisfied is ANIMAL entity since there is no multivalued or composite attributes in this entity.
- Since 1NF is satisfied and all the non-prime attributes are fully dependent on the primary key A_ID thus the 2NF is satisfied.
- 3NF is satisfied since there is not Transitive Functional Dependency in the entity ANIMAL, no non-prime attribute is transitively dependent on the primary key.
- The BCNF is satisfied since the 3NF is satisfied and there exists no FD X → A where X is not a super key.

2. VISITOR:



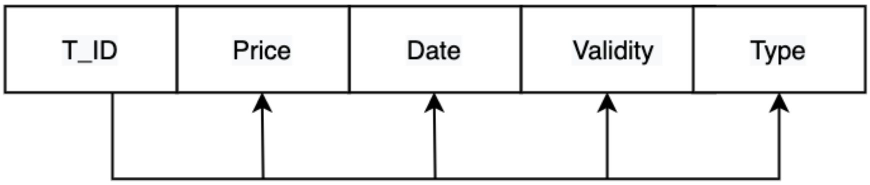
- 1NF is satisfied in VISITOR since we do not have a multivalued or composite attribute in this entity.
- 2NF is satisfied since the one and only attribute Age depends on the primary key V_ID. If the primary key V_ID is dropped then the FD cannot hold.
- 3NF is satisfied since we do not have a Transitive Functional Dependency in the entity VISITOR and no non-prime attribute is transitively dependent on the primary key V_ID.
- The entity VISITOR satisfies the 3NF and does not have FD X → A is not a super key therefore BCNF is satisfied.

3. ZOO_BRANCH:



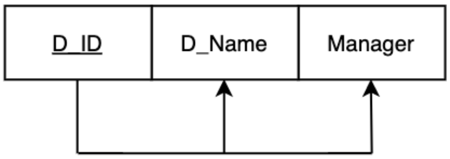
- 1 NF is satisfied in ZOO_BRANCH since there is not multivalued or composite attribute in this entity.
- As for the 2NF all attributes depend on the primary key Z_ID and if the primary key is dropped the FD cannot hold.
- 3NF is satisfied since there is no Transitive Functional Dependency in the entity ZOO_BRANCH, no non-prime attribute is transitively dependent on the primary key Z_ID.
- The entity satisfies the 3NF and has no FD X → A is not a super key thus BCNF is satisfied.

4. TICKET:



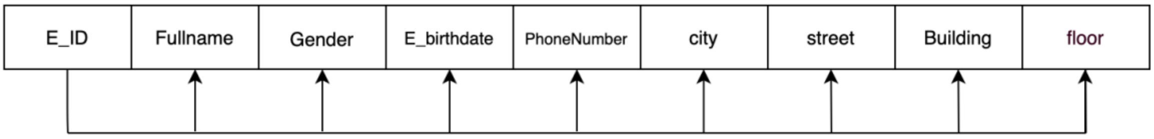
- 1NF is satisfied in TICKET since we do not have a multivalued or composite attribute in this entity.
- 2NF is satisfied since all attributes depend on the primary key T_ID. If the primary key T_ID is dropped then the FD cannot hold.
- 3NF is satisfied since we do not have a Transitive Functional Dependency in the entity TICKET and no non-prime attribute is transitively dependent on the primary key T_ID.
- The entity TICKET satisfies the 3NF and does not have FD X → A is not a super key therefore BCNF is satisfied.

5. DEPARTMENT:



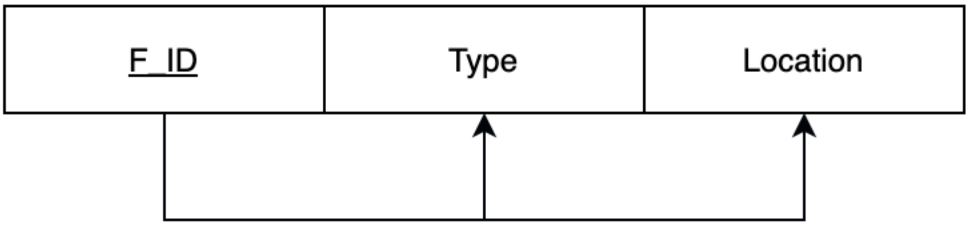
- 1NF is satisfied in DEPARTMENT since we do not have a multivalued or composite attribute in this entity.
- 2NF is satisfied since all attributes depend on the primary key D_ID. If the primary key D_ID is dropped then the FD cannot hold.
- 3NF is satisfied since we do not have a Transitive Functional Dependency in the entity DEPARTMENT and no non-prime attribute is transitively dependent on the primary key D_ID.
- The entity DEPARTMENT satisfies the 3NF and does not have FD X → A is not a super key therefore BCNF is satisfied.

6. EMPLOYEE:



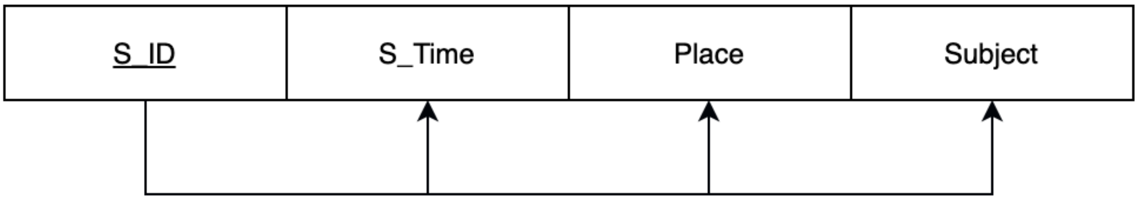
- The EMPLOYEE relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
- The EMPLOYEE relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “E_ID”.
- The EMPLOYEE relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “E_ID”.
- The EMPLOYEE relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key.

7. FACILITY:



- 1NF is satisfied in FACILITY since there is no multivalued or composite attribute in this entity.
- Since 1NF is satisfied and all Non-prime attributes are Fully Functional Dependent on the primary key F_ID then the 2NF is satisfied.
- In the FACILITY table the conditions are satisfied for the 3NF because it satisfies the 2NF and there are no nonprime attributes that are transitively dependent on the primary key F_ID.
- BCNF is satisfied since there is no FD X→ A such that X is a not a super key.

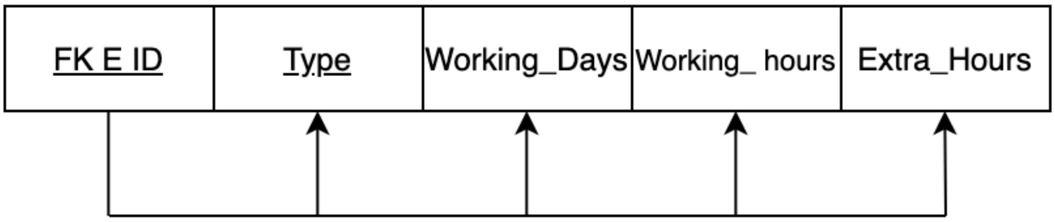
8. SHOW:



- 1NF is satisfied in SHOW since there is no multivalued or composite attribute in this entity.
- Since 1NF is satisfied and all Non-prime attributes are Fully Functional Dependent on the primary key S_ID then the 2NF is satisfied.

- In the SHOW table the conditions are satisfied for the 3NF because it satisfies the 2NF and the nonprime attributes are dependent on the primary key S_ID.
- BCNF is satisfied since there is no FD X → A such that X is a not a super key.

9. SCHEDULE



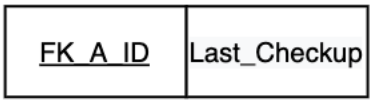
- The SCHEDULE relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.
- .The SCHEDULE relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “FK_E_ID and Type”.
- The SCHEDULE relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “FK_E_ID and Type”
- The SCHEDULE relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key

10. MEDICAL_CARE:

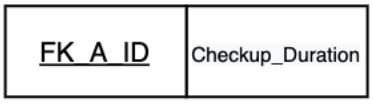
FK_A_ID	Last_Checkup	Checkup_Duration

- In the MEDICAL_CARE table there is not multivalued and composite attributes thus 1NF is satisfied.
- The 2NF is not satisfied since we hve two primary keys FK_A_ID and Last_Checkup suck that {FK_A_ID, Last_Checkup} → Checkup_Duration but if Last_Checkup dropped FK_A_ID → Checkup_Duration still holds.

MEDICAL_CARE-A

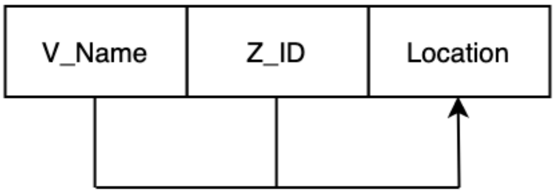


MEDICAL_CARE-B



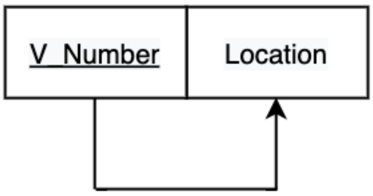
- After separation of the table the 2NF is satisfied and 3NF is satisfied too since every FD X→ A where X is a super key or A is prime attribute and there is no transitively dependent function in these tables
- BCNF is satisfied since there is no FD X→ A such that X is not a super key

11.VISITS:

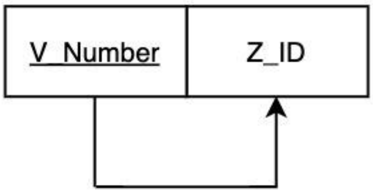


- 1NF is satisfied in VISITS since we do not have a multivalued or composite attribute in this entity.
- 2NF is not satisfied since we have two primary keys V_Number and Z_ID such that {V_Number, Z_ID} → Location however if Z_ID is dropped then V_Number → Location still holds.

VISITS-A



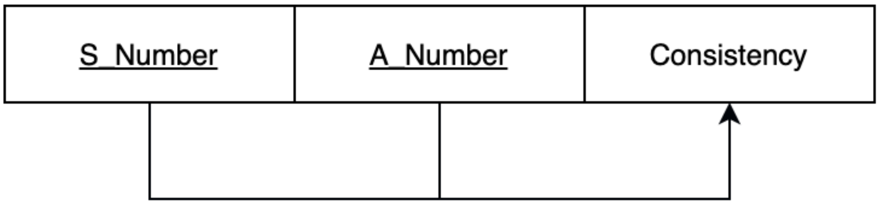
VISITS-B



- After separation of the table the 2NF is satisfied and 3NF is satisfied too since every FD X → A X is a super key or A is prime attribute and there is no transitively dependent function in these tables.

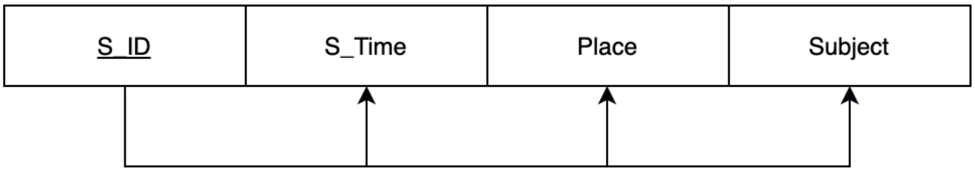
- BCNF is satisfied since there is no FD X → A such that X is a not a super key.

12. FEATURES:

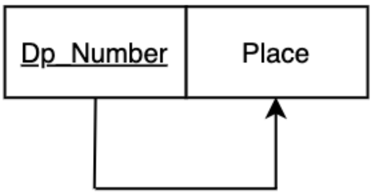


- 1NF is satisfied in FEAUTURES since there is no multivalued or composite attribute in this entity.
- Since 1NF is satisfied and all Non-prime attributes are Fully Functional Dependent on the primary keys S_Number and A_Number then the 2NF is satisfied.
- For the 3NF, there is a Functional Dependency between S_Number and A_Number → Consistency, moreover S_Number and A_Number are super thus 3NF is satisfied.
- BCNF is satisfied since as we can see, in this table the FD X → A where X is a super key is satisfied.

MANAGES-A

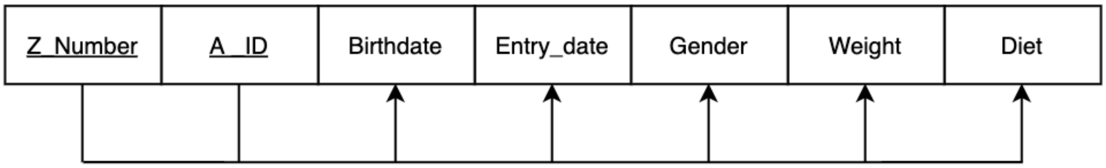


MANAGES-B



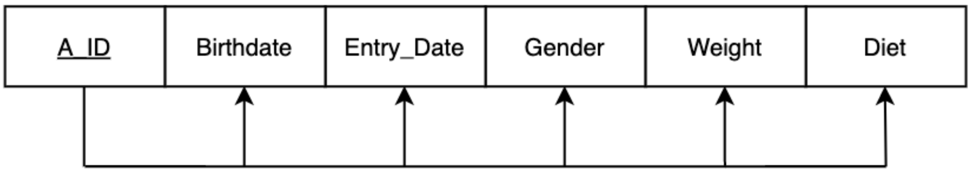
- After separation of the table the 2NF is satisfied and 3NF is satisfied too since every FD X → A, X is a super key or A is prime attribute and there is no transitively dependent function in these tables.
- BCNF is satisfied since there is no FD X→A such that X is a not a super key.

13. CONTAINS:

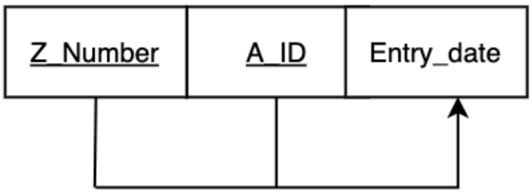


- In the CONTAINS table there is no multivalued and composite attribute thus 1NF is satisfied.
- The 2NF is not satisfied since we have two primary keys Z_Number and A_ID

CONTAINS-A

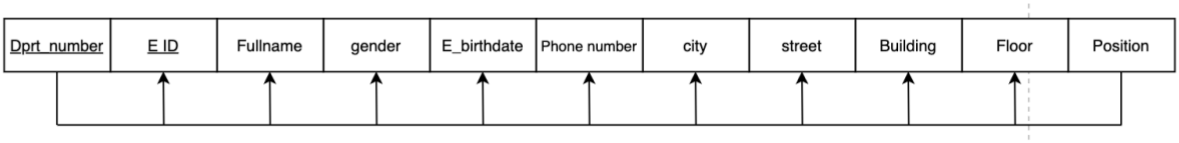


CONTAINS-B



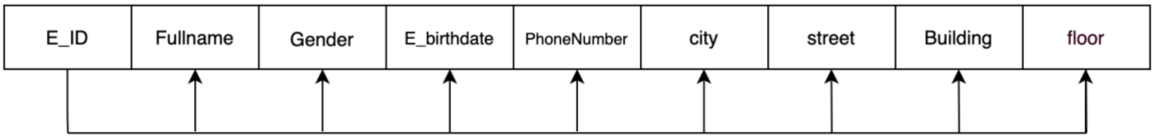
- After separation of the table, the 2NF is satisfied and 3NF is satisfied too since every FD X → A X is a super key or A is prime attribute and there is not transitively dependent function in these tables.
- BCNF is satisfied since there is no FD X → A such that X is not a super key.

14. WORKS IN:

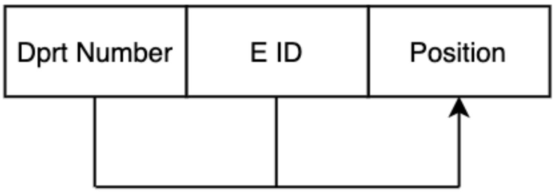


- The WORKS_IN relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic
- The 2NF is not satisfied since we have two primary keys “Dprt_Number” and “E_ID”

WORKS_IN-A:

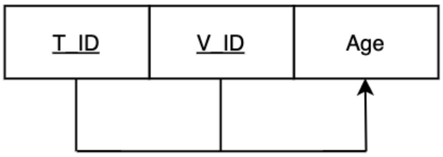


WORKS_IN-B:



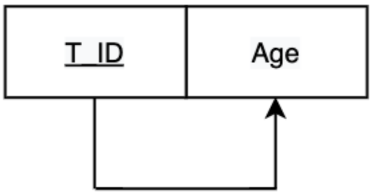
- After separation of the table, the 2NF is satisfied.
- The WORKS_IN relation schemas satisfiy all conditions of the 3NF because they satisfy the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “E_ID”
- The WORKS_IN relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key

15. BUYS:

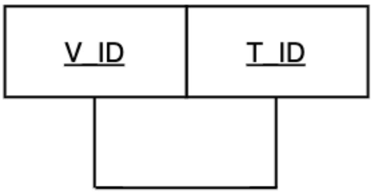


- 1NF is satisfied in BUYS since we do not have a multivalued or composite attribute in this entity.
- 2NF is not satisfied since we have two primary keys V_ID and T_ID such that {V_ID, T_ID} → Age however if V_ID is dropped then T_ID → Age still holds.

BUYS-A



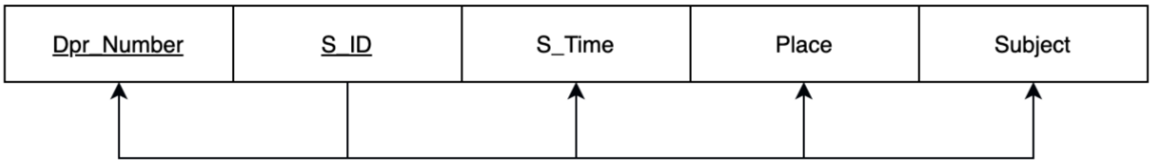
BUYS-B



- After separation of the table the 2NF is satisfied and 3NF is satisfied too since every FD X→A X is a super key or A is prime attribute and there is no transitively dependent function in these tables.

- BCNF is satisfied since there is no FD X→ A such that X is a not a super key.

16. MANAGES:



- 1NF is satisfied in MANAGES since there is no multivalued or composite attribute in this entity.
- 1NF is satisfied, however all Non-prime attributes are partially Functional Dependent only on the primary key S_ID, thus the table should be separated.

17. RUNS:

Dprt_Number	F_Number

- 1NF is satisfied in RUNS since there is no multivalued or composite attribute in this entity.
- Relation RUNS does not have non-prime attributes.

XIII. Conclusion:

In the above report, we have attempted to precisely model how to design and define a professional, and high quality zoo database.

Since databases are extremely fragile and complex, the whole procedure has to compartmentalized into clear stages. The first step, was to design the actual entities and relationships that build up this database. And despite the fact that these entities and relationships were built at a high level; the second step is where these higher level descriptions are changed into relation schemes. In stage 3, transform these relation schemes into SQL code and build the queries and tables on the oracle server. In the last, and fourth stage, we have normalized the database. Regardless of size, Databases (Merriam Webster Online, n.d) are extremely valuable for any business or organization.

Multiple factors, agents, and transactions are involved in any actual organization. It is crucial for these to be built with exact and well organized databases. They gave to be handled carefully to assure preventing any mistakes from occurring when used by future users and clients to ensure such great quality, the database has to be built slowly with such care that the mini-world of discourse; which the database represents, can be computationally represented in a very precise manner.

XIV. Report Checklist:

- ✓ Font Type: Times New Roman.
- ✓ Team of five: Yes.
- ✓ Plain English: Yes.
- ✓ Group named: Yes.
- ✓ Group leader: Yes.
- ✓ Real life database application: Yes.
- ✓ Challenging and interesting database application: Yes.
- ✓ Font Size: 11.
- ✓ 1.5 Spaced: Yes.
- ✓ Typed: Yes.
- ✓ At least 10 essential pages: Yes.
- ✓ At least 10 entities: Yes.
- ✓ At least 1 multivalued attribute: Yes.
- ✓ At least 1 composite relation: Yes.
- ✓ At least 1 multivalued relationship: Yes.
- ✓ At least 1 weak entity: Yes.
- ✓ ER Model built: Yes.
- ✓ Deadline respected: Yes.
- ✓ Corrected phase I and II: Yes.
- ✓ Improved phase I and II: Yes.
- ✓ Translated the ERdiagram into relational schemas: Yes.
- ✓ Build database on the Oracle Server: Yes.
- ✓ At least ten tuples per relation on the database server: Yes.
- ✓ 10 transactions to query database: Yes.
- ✓ Express queries in SQL: Yes.
- ✓ Execute SQL queries on the database implemented using ORACLE: Yes.
- ✓ Normalize the database up to the BCNF Normal Form: Yes
- ✓ Explain the normalization process: Yes