Database Management Systems Case Study:

Tour de France

Manav Madhukar Dalvi

M.Sc. Big Data Analytics , St. Xavier's College, Mumbai

Database Management Systems Case Study

Name: Manav Madhukar Dalvi

UID no.: 229031

Section: M.Sc. Big Data Analytics

Class: Part 1

Roll no.: 23

Topic: **Entity Relationship Case Study on Tour de France**

Index

Abstract

Tour de France is world's most prestigious and most difficult bicycle race. It is held primarily in France while also occasionally passing through neighbouring countries. Of the three foremost races; the others being Giro d'Italia and the Vuelta a Espana, this event attracts the world's best riders. The top cyclists from around the world compete in a gruelling three-week 3,500 km journey each July that takes competitors through the mountain chains of the Pyrenees and the Alps and on to the finish on the Champs-Élysées in Paris. The Tour de France is broadcast worldwide on 121 different television channels in over 190 countries with huge audiences watching each stage, including a record 44 million viewers who took in one of the final stages in the 2009 race. It takes a massive effort to broadcast all the excitement to the millions of fans watching around the globe.

The objective of this case study to understand the database structure of this mega cycling event, Tour de France. Due to the sheer scale of the event, the database for this, also needs to be large and intricate. We can understand the structure of it with the help of the Entity-Relationship diagram (pg. no. 6). The structure can be seen as a bulb-shaped diagram, where the base portrays the entities on the participation side of the event and the top of the bulb shows the organizational side of it.

We will go ahead with this analogy for the whole of this case study.

Introduction

We will start this case study by understanding what an entity is. An entity is anything that has a physical existence like an object, person, place, etc. Each entity has a set of attributes. For understanding, entities can be visualized as tables where attributes will be columns.

This case study involves a total of 25 entities, 11 on the participation side and 14 on organizational side. We will first look at the entities in participation side:

1. Cyclists (CYC_id, First_Name, Last_Name, Age, Height, Weight, Team_id, Type_, Country)
2. Team (Team_id, Sponsor_id, Gen_Mag_id, Race_Dir_ id, Press_Officer_id, Tech_Dir_id, Hosp_Mag_id, Doc_id, Soigneur_1_id, Soigneur_2_id, Soigneur_3_id, Soigneur_4_id, Mechanic_1_id, Mechanic_2_id, Mechanic_3_id, Mechanic_4_id)
3. Team_Vehicles (Vehicle_Identification_Number_T, Type_of_vehicle, Capacity_, Purpose, Num_of_seats)
4. Hospitality_Managers (Hosp_Mag_id, HM_F_name, HM_L_name, Years_of_experience, Num_of_teams_managed, Age)
5. General_Managers (Gen_Mag_id, M_F_name, M_L_name, Years_of_experience, Num_of_events_managed, Age)
6. Race_Directors (Race_Dir_id, RD_F_name, RD_L_name, Years_of_experience, Num_of_races_directed, Age)
7. Press_Officers (Press_officer_id, PO_F_name, PO_L_name, Years_of_experience, Num_of_events_handled, Age)
8. Doctors (Doc_id, D_F_name, D_L_name, Years_of_experience, Specialization, Age)
9. Soigneurs (Soigneur_id, SG_F_name, SG_L_name, Years_of_experience, Specialization, Age)
10. Mechanics (Mechanic_id, MC_F_name, MC_L_name, Years_of_experience, Specialization, Age)

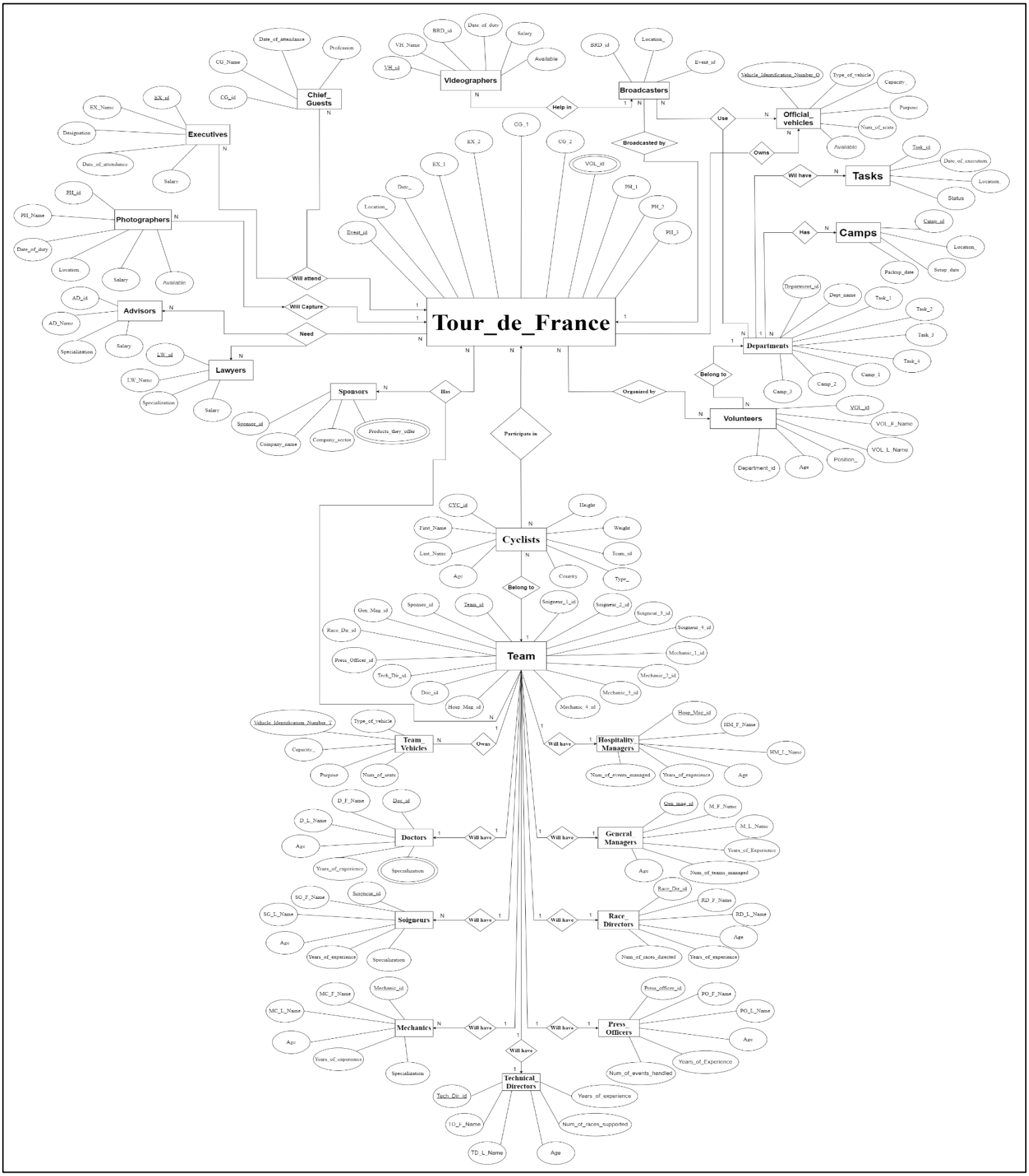
11. Technical_Directors (Tech_Dir_id, TD_F_name, TD_L_name, Years_of_experience, Num_of_races_supported, Age)

The above were the entities on the participation side, now we will look at the organizational side of this database. The entities are as follows:

1. Tour_de_France (Event_id, Location_, Date_, EX_1, EX_2, CG_1, CG_2, VOL_id, PH_1, PH_2, PH_3)
2. Sponsors (Sponsor_id, Company_name, Company_sector, Products_they_offer)
3. Lawyers (LW_id, LW_Name, Specialization, Salary)
4. Advisors (AD_id, AD_Name, Specialization, Salary)
5. Photographers (PH_id, PH_Name, Date_of_duty, Location_, Salary, Available)
6. Videographers (VH_id, VH_Name, Date_of_duty, BRD_id, Salary, Available)
7. Broadcasters (BRD_id, Location_, Event_id)
8. Executives (EX_id, EX_Name, Designation, Date_of_attendance, Salary)
9. Chief_Guests (CG_id, CG_Name, Profession, Date_of_attendance)
10. Volunteers (VOL_id, VOL_F_Name, VOL_L_Name, Position_, Age, Department_id)
11. Departments (Department_id, Dept_name, Task_1, Task_2, Task_3, Task_4, Camp_1, Camp_2, Camp_3)
12. Camps (Camp_id, Location_, Setup_date, Packup_date)
13. Tasks (Task_id, Date_of_execution, Location_, Status)
14. Official_Vehicles (Vehicle_Identification_Number_O, Type_of_vehicle, Capacity_, Purpose, Num_of_seats, Available)

This sums up all the entities in our Tour de France database. These entities together have intricate relationships amongst themselves. We can understand these entities and their relationships better when it is visualized for us. Hence, the next section, we view the Entity-Relationship Diagram.

Entity-Relationship Diagram



Entity Description

The Entity Relationship diagram gives us a beautiful overview of the database, including the entities within along with their relationships with each other. But the diagram does not do justice to each individual entities, their attribute, and their types. Hence, we take a detailed look into the entities as follows:

1. General_Managers

TABLE GENERAL_MANAGERS

Column	Null?	Type
GEN_MAG_ID	NOT NULL	NUMBER
M_F_NAME	NOT NULL	VARCHAR2(20)
M_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
NUM_OF_EVENTS_MANAGED	NOT NULL	NUMBER
AGE	-	NUMBER

Here, Gen_Mag_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Gen\_Mag\_id)^+$ = {Gen_Mag_id, M_F_name, M_L_name, Years_of_experience, Num_of_events_managed, Age}

2. Hospitality_Managers

TABLE HOSPITALITY_MANAGERS

Column	Null?	Type
HOSP_MAG_ID	NOT NULL	NUMBER
HM_F_NAME	NOT NULL	VARCHAR2(20)
HM_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
NUM_OF_TEAMS_MANAGED	NOT NULL	NUMBER
AGE	-	NUMBER

Here, Hosp_Mag_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Hosp\_Mag\_id)^+$ = {Hosp_Mag_id, HM_F_name, HM_L_name, Years_of_experience, Num_of_teams_managed, Age }

3. Race_Directors

TABLE RACE_DIRECTORS

Column	Null?	Type
RACE_DIR_ID	NOT NULL	NUMBER
RD_F_NAME	NOT NULL	VARCHAR2(20)
RD_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
NUM_OF_RACES_DIRECTED	NOT NULL	NUMBER
AGE	-	NUMBER

Here, Race_Dir_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Race\_Dir\_id)^+$ = {Race_Dir_id, RD_F_name, RD_L_name, Years_of_experience, Num_of_races_directed, Age }

4. Press_Officers

TABLE PRESS_OFFICERS

Column	Null?	Type
PRESS_OFFICER_ID	NOT NULL	NUMBER
PO_F_NAME	NOT NULL	VARCHAR2(20)
PO_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
NUM_OF_EVENTS_HANDLED	NOT NULL	NUMBER
AGE	-	NUMBER

Here, Press_Officer_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(\text{Press\_Officer\_id})^+$ = {Press_Officer_id, PO_F_name, PO_L_name, Years_of_experience, Num_of_events_handled, Age }

5. Doctors

TABLE DOCTORS

Column	Null?	Type
DOC_ID	NOT NULL	NUMBER
D_F_NAME	NOT NULL	VARCHAR2(20)
D_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
SPECIALIZATION	NOT NULL	VARCHAR2(20)
AGE	-	NUMBER

Here, Doc_id is the primary key through which all the other attributes can be obtained. The attribute Specialization is a multivalued attribute here.

Functional Dependency:

$(\text{Doc\_id})^+$ = { Doc_id, D_F_name, D_L_name, Years_of_experience, Specialization, Age}

6. Soigneurs

TABLE SOIGNEURS

Column	Null?	Type
SOIGNEUR_ID	NOT NULL	NUMBER
SG_F_NAME	NOT NULL	VARCHAR2(20)
SG_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
SPECIALIZATION	NOT NULL	VARCHAR2(20)
AGE	-	NUMBER

Here, Soigneur_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(\text{Soigneur\_id})^+$ = { Soigneur_id, SG_F_name, SG_L_name, Years_of_experience, Specialization, Age}

7. Mechanics

TABLE MECHANICS

Column	Null?	Type
MECHANIC_ID	NOT NULL	NUMBER
MC_F_NAME	NOT NULL	VARCHAR2(20)
MC_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
SPECIALIZATION	NOT NULL	VARCHAR2(20)
AGE	-	NUMBER

Here, Mechanic_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(\text{Mechanic\_id})^+$ = { Mechanic_id, MC_F_name, MC_L_name, Years_of_experience, Specialization, Age}

8. Technical_Directors

TABLE TECHNICAL_DIRECTORS

Column	Null?	Type
TECH_DIR_ID	NOT NULL	NUMBER
RD_F_NAME	NOT NULL	VARCHAR2(20)
RD_L_NAME	NOT NULL	VARCHAR2(20)
YEARS_OF_EXPERIENCE	NOT NULL	NUMBER
NUM_OF_RACES_DIRECTED	NOT NULL	NUMBER
AGE	-	NUMBER

Here, Tech_Dir_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Tech\_Dir\_id)^+$ = { Tech_Dir_id, TD_F_name, TD_L_name, Years_of_experience, Num_of_races_supported, Age}

9. Cyclists

TABLE CYCLISTS

Column	Null?	Type
CYC_ID	NOT NULL	NUMBER
FIRST_NAME	NOT NULL	VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(20)
AGE	NOT NULL	NUMBER
HEIGHT	NOT NULL	FLOAT(126)
WEIGHT	NOT NULL	FLOAT(126)
TEAM_ID	NOT NULL	NUMBER
TYPE_	NOT NULL	VARCHAR2(20)
COUNTRY	NOT NULL	VARCHAR2(20)

Here, CYC_id is the primary key through which all the other attributes can be obtained. We also have a foreign key Team_id referencing to the Team entity.

Functional Dependency:

$(CYC\_id)^+$ = { CYC_id, First_Name, Last_Name, Age, Height, Weight, Team_id, Type_, Country}

10. Team

TABLE TEAM

Column	Null?	Type
TEAM_ID	NOT NULL	NUMBER
SPONSOR_ID	NOT NULL	NUMBER
GEN_MAG_ID	NOT NULL	NUMBER
RACE_DIR_ID	NOT NULL	NUMBER
PRESS_OFFICER_ID	NOT NULL	NUMBER
TECH_DIR_ID	NOT NULL	NUMBER
HOSP_MAG_ID	NOT NULL	NUMBER
DOC_ID	NOT NULL	NUMBER
SOIGNEUR_1_ID	NOT NULL	NUMBER
SOIGNEUR_2_ID	NOT NULL	NUMBER
SOIGNEUR_3_ID	NOT NULL	NUMBER
SOIGNEUR_4_ID	NOT NULL	NUMBER
MECHANIC_1_ID	NOT NULL	NUMBER
MECHANIC_2_ID	NOT NULL	NUMBER
MECHANIC_3_ID	NOT NULL	NUMBER
MECHANIC_4_ID	NOT NULL	NUMBER

Here, Team_id is the primary key through which all the other attributes can be obtained. We also have a set of foreign keys like Sponsor_id referencing to the Sponsor entity, Gen_Mag_id referencing to the General_Managers entity, Race_Dir_id referencing to the Race_Directors entity, Press_Officer_id referencing to the Press_Officers entity, Tech_Dir_id referencing to the Technical_Directors entity, Hosp_Mag_id referencing to the Hospitality_Managers entity, Doc_id referencing to the Doctors entity and a bunch or Soigneur_id's and Mechanic_id's referencing to the Soigneurs and Mechanics entity respectively.

Functional Dependency:

$(Team\_id)^{+}$ = { Team_id, Sponsor_id, Gen_Mag_id, Race_Dir_ id, Press_Officer_id, Tech_Dir_id, Hosp_Mag_id, Doc_id, Soigneur_1_id, Soigneur_2_id, Soigneur_3_id, Soigneur_4_id, Mechanic_1_id, Mechanic_2_id, Mechanic_3_id, Mechanic_4_id}

11. Team_Vehicles

TABLE TEAM_VEHICLES

Column	Null?	Type
VEHICLE_IDENTIFICATION_NUMBER_T	NOT NULL	VARCHAR2(9)
TYPE_OF_VEHICLE	NOT NULL	VARCHAR2(20)
CAPACITY_	-	NUMBER
PURPOSE	-	VARCHAR2(20)
NUM_OF_SEATS	-	NUMBER

Here, Vehicle_Identification_Number_T is the primary key through which all the other attributes can be obtained.

Functional Dependency:
$(\text{Vehicle\_Identification\_Number\_T})^{+}$ =
{Vehicle_Identification_Number_T, Type_of_vehicle, Capacity_, Purpose, Num_of_seats }

12. Lawyers

TABLE LAWYERS

Column	Null?	Type
LW_ID	NOT NULL	NUMBER
LW_NAME	NOT NULL	VARCHAR2(20)
SPECIALIZATION	NOT NULL	VARCHAR2(20)
SALARY	NOT NULL	FLOAT(126)

Here, LW_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:
$(\text{LW\_id})^{+}$ = { LW_id, LW_Name, Specialization, Salary }

13. Advisors

TABLE ADVISORS

Column	Null?	Type
AD_ID	NOT NULL	NUMBER
AD_NAME	NOT NULL	VARCHAR2(20)
SPECIALIZATION	NOT NULL	VARCHAR2(20)
SALARY	NOT NULL	FLOAT(126)

Here, AD_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(AD\_id)^+$ = { AD_id, AD_Name, Specialization, Salary }

14. Photographers

TABLE PHOTOGRAPHERS

Column	Null?	Type
PH_ID	NOT NULL	NUMBER
PH_NAME	NOT NULL	VARCHAR2(20)
DATE_OF_DUTY	NOT NULL	DATE
LOCATION_	NOT NULL	VARCHAR2(20)
SALARY	NOT NULL	FLOAT(126)
AVAILABLE	NOT NULL	NUMBER(1,0)

Here, PH_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(PH\_id)^+$ = { PH_id, PH_Name, Date_of_duty, Location_, Salary, Available }

15. Broadcasters

TABLE BROADCASTERS

Column	Null?	Type
BRD_ID	NOT NULL	NUMBER
LOCATION_	NOT NULL	VARCHAR2(20)
EVENT_ID	NOT NULL	NUMBER

Here, BRD_id is the primary key through which all the other attributes can be obtained. We also have a foreign key Event_id referencing to the Tour_de_France entity.

Functional Dependency:

$(BRD\_id)^+$ = { BRD_id, Location_, Event_id}

16. Videographers

TABLE VIDEOGRAPHERS

Column	Null?	Type
VH_ID	NOT NULL	NUMBER
VH_NAME	NOT NULL	VARCHAR2(20)
DATE_OF_DUTY	NOT NULL	DATE
BRD_ID	NOT NULL	NUMBER
SALARY	NOT NULL	FLOAT(126)
AVAILABLE	NOT NULL	NUMBER(1,0)

Here, VH_id is the primary key through which all the other attributes can be obtained. We also have a foreign key BRD_id referencing to the Broadcasters entity.

Functional Dependency:

$(VH\_id)^+$ = { VH_id, VH_Name, Date_of_duty, BRD_id, Salary, Available }

17. Executives

TABLE EXECUTIVES

Column	Null?	Type
EX_ID	NOT NULL	NUMBER
EX_NAME	NOT NULL	VARCHAR2(20)
DESIGNATION	NOT NULL	VARCHAR2(20)
DATE_OF_ATTENDANCE	NOT NULL	DATE
SALARY	NOT NULL	FLOAT(126)

Here, EX_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(EX\_id)^+$ = { EX_id, EX_Name, Date_of_attendance, Designation, Salary}

18. Chief_Guests

TABLE CHIEF_GUESTS

Column	Null?	Type
CG_ID	NOT NULL	NUMBER
CG_NAME	NOT NULL	VARCHAR2(20)
PROFESSION	-	VARCHAR2(20)
DATE_OF_ATTENDANCE	NOT NULL	DATE

Here, CG_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(CG\_id)^+$ = { CG_id, CG_Name, Date_of_attendance, Profession}

19. Tour_de_France

TABLE TOUR_DE_FRANCE

Column	Null?	Type
EVENT_ID	NOT NULL	NUMBER
LOCATION_	NOT NULL	VARCHAR2(20)
DATE_	NOT NULL	DATE
EX_1	-	NUMBER
EX_2	-	NUMBER
CG_1	-	NUMBER
CG_2	-	NUMBER
VOL_ID	-	NUMBER
PH_1	NOT NULL	NUMBER
PH_2	-	NUMBER
PH_3	-	NUMBER

Here, Event_id is the primary key through which all the other attributes can be obtained. The attribute VOL_id is a multivalued attribute here. We also have a set of foreign keys like EX_id's referencing to the Executive entity, CG_id's referencing to the Chief_Guests entity, VOL_id referencing to the Volunteers entity and PH_id's referencing to the Photographers entity.

Functional Dependency:

$(Event\_id)^+$ = { Event_id, Location, Date, EX_1, EX_2, CG_1, CG_2, VOL_id, PH_1, PH_2, PH_3 }

20. Official_Vehicles

TABLE OFFICIAL_VEHICLES

Column	Null?	Type
VEHICLE_IDENTIFICATION_NUMBER_O	NOT NULL	VARCHAR2(9)
TYPE_OF_VEHICLE	-	VARCHAR2(20)
CAPACITY_	-	NUMBER
PURPOSE	-	VARCHAR2(20)
NUM_OF_SEATS	-	NUMBER
AVAILABLE	NOT NULL	NUMBER(1,0)

Here, Vehicle_Identification_Number_O is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Vehicle\_Identification\_Number\_O)^+$ = {Vehicle_Identification_Number_O, Type_of_vehicle, Capacity_, Purpose, Num_of_seats, Available}

21. Tasks

TABLE TASKS

Column	Null?	Type
TASK_ID	NOT NULL	NUMBER
DATE_OF_EXECUTION	-	DATE
LOCATION_	-	VARCHAR2(30)
STATUS	-	NUMBER(1,0)

Here, Task_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Task\_id)^+$ = { Task_id, Location_, Date_of_execution, Status}

22. Camps

TABLE CAMPS

Column	Null?	Type
CAMP_ID	NOT NULL	NUMBER
LOCATION_	-	VARCHAR2(30)
SETUP_DATE	-	DATE
PACKUP_DATE	-	DATE

Here, Camp_id is the primary key through which all the other attributes can be obtained.

Functional Dependency:

$(Camp\_id)^+$ = { Camp_id, Location_, Setup_date, Packup_date}

23. Departments

TABLE DEPARTMENTS

Column	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER
DEPT_NAME	NOT NULL	VARCHAR2(20)
TASK_1	-	NUMBER
TASK_2	-	NUMBER
TASK_3	-	NUMBER
TASK_4	-	NUMBER
CAMP_1	NOT NULL	NUMBER
CAMP_2	-	NUMBER
CAMP_3	-	NUMBER

Here, Department_id is the primary key through which all the other attributes can be obtained. We also have a set of foreign keys like Task_id's referencing to the Tasks entity and Camp_id's referencing to the Camps entity.

Functional Dependency:

$(Department\_id)^+$ = { Department_id, Dept_name, Task_1, Task_2, Task_3, Task_4, Camp_1, Camp_2, Camp_3}

24. Volunteers

TABLE VOLUNTEERS

Column	Null?	Type
VOL_ID	NOT NULL	NUMBER
VOL_F_NAME	NOT NULL	VARCHAR2(20)
VOL_L_NAME	NOT NULL	VARCHAR2(20)
POSITION_	NOT NULL	VARCHAR2(20)
AGE	-	NUMBER
DEPARTMENT_ID	-	NUMBER

Here, VOL_id is the primary key through which all the other attributes can be obtained. We also have a foreign key Department_id referencing to the Departments entity.

Functional Dependency:

$(VOL\_id)^+$ = { VOL_id, VOL_F_Name, VOL_L_Name, Position_, Age, Department_id }

25. Sponsors

TABLE SPONSORS

Column	Null?	Type
SPONSOR_ID	NOT NULL	NUMBER
COMPANY_NAME	NOT NULL	VARCHAR2(20)
COMPANY_SECTOR	NOT NULL	VARCHAR2(20)
PRODUCTS_THEY_OFFER	-	VARCHAR2(20)

Here, Sponsor_id is the primary key through which all the other attributes can be obtained. The attribute Products_they_offer is a multivalued attribute here.

Functional Dependency:

$(Sponsor\_id)^+$ = { Sponsor_id, Company_name, Company_sector, Products_they_offer }

Relationships among Entities

Understanding the relationships between entities is an essential step in interpreting the structure of a particular database. Until now, we have discussed about the overall structure of the database, understood how the entities and its attributes are defined and now, we need to look at how these entities relate to each other.

We will first look at our main entity, ***Tour_de_France***, it is connected to many entities by a relationship and we will look at them one by one. Firstly, due to the sheer scale of the event, it needs some ***Lawyers*** and ***Advisors*** in case of any legal action. Speaking of the scale, this event needs a lot of man power to make this happen as smoothly as it can. For that, we have ***Volunteers*** that help organizing and contributing to this event. Every volunteer belongs to a ***Department***. These ***Departments*** are located at different ***Camps*** all over the event locations and must complete some set of assigned ***Tasks*** on an assigned date.

The main entity owns a set of ***Official_Vehicles*** which can be used by ***Departments*** to complete their tasks and the ***Broadcasters*** for help in videography based on their availability of the vehicle. There are ***Videographers*** present who help the ***Broadcasters*** for capturing the moments live.

This ***Tour_de_France*** events are held for 23 days involving 21 different stages. These events are attended by variety of ***Executives*** from the company and ***Chief_Guests*** from different professions. And for capturing all these events, we have set of ***Photographers*** assigned for each event at different locations around the map.

Sponsors have a big role in making this event live up to its name. They not only help in funding the events but also sponsor some ***Teams*** of ***Cyclists***. These ***Teams*** involve not only different types of ***Cyclists*** from around the world but also a set of professionals who help in running this team efficiently. These professionals include a set of ***General_Managers, Hopitality_Managers, Race_directors, Doctors, Soigneurs and Mechanics***. These ***Teams*** own ***Team_Vehicles*** which they use to assist the ***Cyclists*** in whichever manner they require.

Now that we have completely understood the entities and their relationships among each other, we can try to make it more efficient.

Normalization

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies. Normalization divides the larger table into smaller and links them using relationships. The normal form is used to reduce redundancy from the database table.

The main reason for normalizing the relations is removing these anomalies. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.

For our case study, we have 3 entities which have multivalued attributes. These entities are Doctors, Sponsors and Tour_de_France, each having multivalued attributes namely, Specializations, Products_they_offer and VOL_id respectively. In order to convert the database to the 1st Normal Form (1NF), we need to deal with these attributes.

1. For Doctors entity, we create a weak entity named DOC_Special which will have foreign key referencing to the Doc_id in the Doctors table and store its specializations in a different attribute. The structure of that entity will be as follows:

TABLE DOC_SPECIAL

Column	Null?	Type
DOC_ID	-	NUMBER
SPECIALIZATIONS	-	VARCHAR2(20)

2. For Sponsors entity, we create a weak entity named SPO_Products which will have foreign key referencing to the Sponsor_id in the Sponsors table and store its products in a different attribute. The structure of that entity will be as follows:

TABLE SPO_PRODUCTS

Column	Null?	Type
SPONSOR_ID	-	NUMBER
PRODUCTS_THEY_OFFER	-	VARCHAR2(20)

3. For Tour_de_France entity, we create a weak entity named TDF_Vols which will have foreign key referencing to the Event_id in the Tour_de_France table and store its VOL_id's in a different attribute. The structure of that entity will be as follows:

TABLE TDF_VOLS

Column	Null?	Type
EVENT_ID	-	NUMBER
VOL_ID	-	VARCHAR2(20)

After creation of these weak entities, we will have a database that is confirmed to achieve 1NF as there will be no multivalued attributes present in the database.

For achieving the 2nd Normal Form (2NF), we need to check if there exists any partial dependency among the attributes of our entities. But, in the **Entity Description** section, we already checked and confirmed for each entity that there lies no partial dependency amongst them.

Hence, we can conclude that, the Tour de France database has achieved its 2nd Normal Form.

Appendix

1. Tour de France
2. Entity Relationship Diagram
3. Entity Descriptions [Code]
4. Entity Descriptions [Snippets]

[If the pdf links do not work, Click here]