

# SPORTIFY IIITD

Team - 63



Samyak Jain | Kshitij Mohan | Harshal Dev  
Satwik Tiwari | Hitesh Garg

# **Week 1:**

We first collectively brainstormed to find out some real world problems that we could solve through this opportunity of developing a DBMS project. The first whole week was spent in collecting ideas, discussing them amongst ourselves. We finally decided that we want to target the audience of our college and college environment in general and started to find out problems that are faced by students. After brainstorming, thinking, discussing our personal experiences and thoughts we arrived at our problem statement.

## **Problem Statement:**

Sport is an essential component of a students' life. There is an important need for a management system for all the activities related to the same. In a busy routine of student life, it is challenging to find people to play along with. Also, it is a hassle to connect with people with the same sporting interests as you and adjust according to their schedule to play with them. Combining the hard-hitting nature of our college's (IIIT Delhi) schedule and crunch of sporting facilities like the small size of the ground, badminton court being occupied most of the time, it is difficult for one to play sports of their choice. There is a lack of a system that fills in this gap and promotes a healthy sporting culture in our college, and helps everyone get a chance to play the game of their choice with people sharing similar interests and skills. Similarly, if one needs to play a sport, but he doesn't have the required equipment or does the college, he has to "ask" his "friend/acquaintance" about it. It is possible that an unknown person might have the desired equipment, but he might not get the equipment because he doesn't know that person. So to bridge this gap and our problem mentioned above, we thought of working in this area.

## Our Motivation and the idea :

Our motivation stems from our want to bring about a sporting perception and revolution and culture in our college . We wanted to create a platform that solves all the problems listed above in an efficient manner. The platform will facilitate, systematic booking of events like a cricket match, a football match etc. by firstly checking for availabilities of resources like equipments(if one wants to book any) and also the venues.It will also connect the people who share the same sporting interest so that we do not face the hassle of finding people to play with.This will also aim to bring the sporting community together which will help in forming good bonds and promote sport culture in general.Also, one can host events be it large scale league matches, or personal matches on the system easily. Our platform would also facilitate the facility of peer borrowing and lending.This way, people can share their equipments and play different sports.This facility of borrowing will also be done by users from the college sporting equipment inventory as well.This system will ensure efficient functioning of the same as the users can easily check the availability of the same easily and much more people can get the chance to use it and play.Also, the feature of checking availability of venues will be provided and a booking system for the same is also provided.The application will also feature various vendors to which either the user or the college sports administration can approach to either buy or repair their products. So in all, our application is a wide covering one which aims to promote sporting culture and encourage more people to play by efficiently managing the sporting system in the college.

## Week 2 :

We started this phase of the project by identifying the stakeholders for our project. We aimed to identify all such groups and people who would be affected by our application.

### Stakeholders :

- ❑ **Students (aka Users):** They will be the ones that will mainly be the users of our application, use it to connect to other individuals, host events i.e - matches and find people to play with
- ❑ **Sports Equipment Vendor/Repairer:** The local sport shop owners which can take new equipment or repair orders from users as well as the Resource manager.
- ❑ **Coaches:** Users will be able to book coaches to help them train for a particular sport.
- ❑ **Referees:** While hosting an event, they will be booked by the host for conducting the game.
- ❑ **Facilitator:** They will be hosting big events like sports tournaments and inter-college sports events.
- ❑ **College Sports Administration:** It is the administrative owners of all the resources like the venues, grounds and all the facilities including the inventory materials. They will also manage the college inventory and handle activities like - lending, recovery, repair, new equipment order, and returning of equipment.
- ❑ **Admin:** They will be able to view the usage statistics of the app as well as user analysis.

After identification of stakeholders, we decided that we would also like to feature the following extra features in our application.

## More ideas to implement :

- Lending borrowing system among the users also apart from the college inventory.
- Live status tracking of the venues
- Interaction amongst the users through which they can discuss various things, ask for equipment suggestions and build a community.
- Adding friends based on similar interest tags.
- Rating feature also will be included
- Notifying the participants for the event
- Login and signup for making a record of users to the database
- Wallets for users which allows them to place repair or new orders.
- High skilled Students can act as a coach also and become referees.

## Queries for Stakeholders :

### ❑ Student/ User:

- Book Venue
- Host Public Event
- Lend Equipment
- Borrow Equipment
- Order Equipment (Repairs also)
- Invite Friends
- Check venue availability
- Join Events
- Ask for coaches

- Give ratings
- View event statistics

**❑ College Sports Administration:**

- Place Equipment orders
- Place repair orders
- Lend Equipment
- Check equipment availability
- Check equipment status
- Request equipment/repairs
- Add penalties

**❑ Sports Equipment Vendor/Repairer:**

- Check buy orders
- Check repair orders
- Request payment
- Send delivery order
- Check delivery status
- Update repair status

**❑ Coaches:**

- Update booking status
- Update availability
- Request payments (from users)
- View ratings

**❑ Referee:**

- Get event details
- Update availability (his/her own availability status & timings)
- Add event results
- Request payments (from facilitators)

❑ **Facilitator (basically students who can host official matches):**

- Host Events
- Update event statistics (player details)
- Book referee
- Check venue availability
- Check equipment availability
- Add event notice

❑ **Application Admin:**

- Get user statistics
- Get application reviews

**So, the Sports we plan to include are :**

Badminton Tennis, Basketball, Table Tennis, Squash, Cricket, Football, Volleyball, Foosball, Billiards, Chess, Swimming  
Because majorly these are the ones that are commonly played in the college.

## Week 3 :

After deciding the stakeholders, roughly writing their queries and commands as such. We now move on to planning and deciding the entities and their corresponding attributes.

## **ENTITIES AND ATTRIBUTES :**

❑ **Students / Users :**

- Name (varchar)
  - First Name (varchar)
  - Last Name (varchar)
- Student/User ID (int)

- Gender(varchar)
- Contact Details
  - E-Mail (varchar)
  - Phone number (varchar)
- Date of Birth (Date)
- Date of Joining(Date)
- Address Location (varchar)

#### ❑ **College Sports Administration :**

- Administration ID (int)
- Name(varchar)
  - First Name(varchar)
  - Last Name(varchar)
- Office Location(varchar)
- Contact Details (varchar)
  - Email (varchar)
  - Phone Number (varchar)

#### ❑ **Sports Equipment Vendor/Repairer :**

- Vendor ID(int)
- Shop Name(varchar)
- Location(varchar)
- Equipment repaired (int)
- Contact Details (str)
  - Phone number(varchar)
  - Email (varchar)

#### ❑ **Coaches :**

- Name (varchar)
  - First Name(varchar)
  - Last Name(varchar)
- Coach Id (int)
- Gender(varchar)
- Date of Birth(Date)
- Number of Players Coached(int)



- Ratings (float)
- Availability Status (TinyInt)
- Contact Details(varchar)
  - Phone Number(varchar)
  - Email id(varchar)

#### ❑ Referee :

- Referee Id (int)
- Gender(varchar)
- Referee Name (varchar)
  - First Name(varchar)
  - Last Name(varchar)
- Date of Birth (Date)
- No of Games refereed (int)
- Availability Status (TinyInt)
- Contact Details(varchar)
  - Email id(varchar)
  - Phone number(varchar)

#### ❑ Venue :

- Venue ID (int)
- Opening Time(DateTime)
- Closing Time(DateTime)
- Venue Type (varchar)
- Location (varchar)
- Occupancy Status (TinyInt)
- No of Current team/players (int)

#### ❑ Equipment :

- Equipment ID (int)
- Equipment Name(varchar)
- Condition(varchar)
- Availability Status (TinyInt)

### ❑ College Equipment

- Equipment Name(varchar)
- Condition(varchar)
- Availability Status(TinyInt)
- Manufacturer(varchar)

### ❑ Sport Event :

- Event ID (int)
- Event Name(varchar)
- EventDate (Date)
- Start Time (Date)
- End Time (Date)
- Event Duration(float)
- Participation Limit(int)

### ❑ Order:

- Order Status (Pending / Complete) (varchar)
- Order ID (int)
- Type (which sport's equipment) (varchar)

### ❑ Sport :

- Sport ID (int)
- Name (varchar)
- Max Players (int)

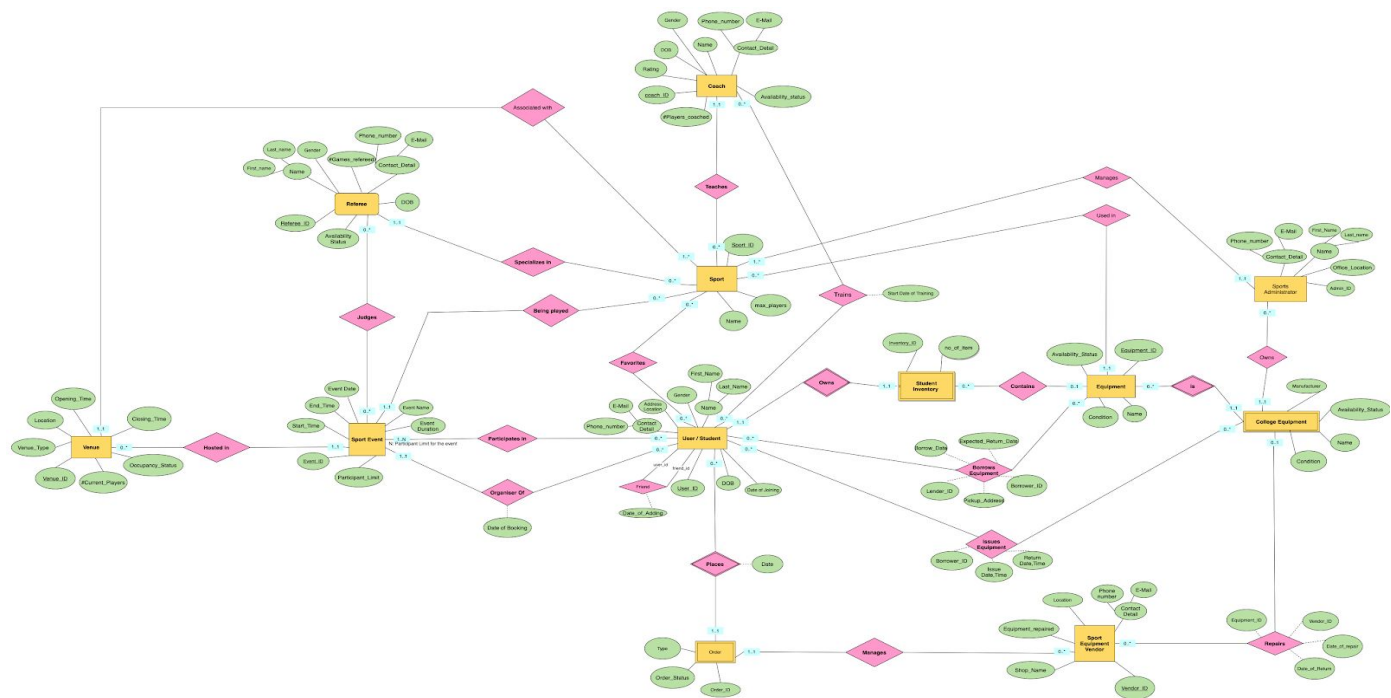
Majorly, these entities are identified that would be needed for the first development phase of the project. Further, we proceed to identify relationships among these entities.

- User *organises* Sport Event
- User *participates in* Sport Event
- User *favorites* Sport (preferred, favorite sports)
- Coach *trains* User
- User *places* Orders (like for repair, buy orders)

- Order is *managed* by the Sport Equipment Vendor
- Sport Equipment Vendor *repairs* college equipments
- User *owns* a Student Inventory
- Student Inventory *contains* equipments
- College equipments *are also* equipments
- College equipment is *owned* by Sports Administrators.
- A Sport Administrator *manages* a sport.
- Equipment is *used in* a sport.
- User *borrow*s Equipment (peer to peer)
- User *makes friends* with other Users (who have similar interests)
- Coach *teaches* a Sport
- A Sport event is *hosted in* a Venue
- Referee *judges* a Sport Event
- Referee *specializes in* a Sport
- A Venue is *associated with* some Sport
- A Sport is being *played in* a Sport event
- User *issues* College Equipments

## Week 4 :

Till now we have decided the stakeholders, what they expect the system to do, the major entities, their attributes and how they will be related. Now we move on to design the ER diagram for the project.



## Week 5 :

We have made the ER diagram for our project. Now we proceed to decide the schema for the tables.

### **SCHEMA :**

PK : Denotes the primary key present directly in the table.

FK : Denotes the foreign key present directly in the table.

2nd column : columns for that table

3rd column : data type

4th column : domain, constraints, checks required if any.

### **USER**

<b>PK</b>	User_ID	int	Not null, unsigned, unique
	First_name	varchar(25)	Not null
	Last_name	varchar(25)	Not null
	Gender	varchar(6)	Not null, check(Gender in ('Male', 'Female', 'Others'))
	Email	varchar(100)	Not null, unique, check for invalid characters, multiple '@' symbols etc.
	Phone_number	varchar(15)	Not null, unique, length should not exceed 15 characters(including code)

	Date_of_Birth (DOB)	Date	Not null , should be ideally less than or equal to current date.
	Age	int	Not Null, unsigned, Age >= 18
	Date_of_Joining (DOJ)	Date	Not null, check DOJ > DOB
	Address_Location	varchar(200)	Not null

## SPORT\_EVENT

<b>PK</b>	Event_ID	int	Not null, unsigned, unique
	Event_Name	varchar(60)	Not null
	Event_Date	Date	Not null, should ideally greater than equal to current date
	Start_Time	DateTime	Not null, should ideally greater than equal to current time.
	End_Time	DateTime	Not null, check End_time > start time
	Duration	float	Not null, unsigned

	Participant_Limit	int	Not null, unsigned, default = max size{sport played} else entered by user
	Date_of_Booking	Date	Not null, should ideally greater than equal to current date
<b>FK</b>	Organizer_ID	int	Not null, unique, unsigned
<b>FK</b>	Venue_ID	int	Not null, unique, unsigned
<b>FK</b>	Sport_ID	int	Not null, unique, unsigned

Referential Integrity Constraints :

- Foreign Key Organizer\_ID references to User(User\_ID).
- Foreign Key Venue\_ID references to Venue(Venue\_ID).
- Foreign Key Sport\_ID references to Sport(Sport\_ID).

## **VENUE**

<b>PK</b>	Venue_ID	int	Not null, unsigned, unique
	Venue_Name	varchar(100)	Not null
	Venue_Type	varchar(6)	Not null, check(Venue_Type in ('Indoor', 'Outdoor'))
	Opening_Time	DateTime	Not null

	Closing_Time	DateTime	Not null, check Closing_time > Opening_Time
	Location	varchar(150)	Not null
	Occupancy_Status	TinyInt	Not null, value lies in {0,1} works as kind of a boolean check
	No_of_current_players_in_venue	int	Not null, greater than equal to 0
<b>FK</b>	Sport_ID	int	Not null, unsigned, unique,

Referential Integrity Constraints :

- Foreign Key Sport\_ID references to Sport(Sport\_ID).

## **REFEREE**

<b>PK</b>	Referee_ID	int	Not null, unsigned, unique
	First_name	varchar(25)	Not null
	Last_name	varchar(25)	Not null
	Gender	varchar(6)	Not Null, check(Gender in ('Male', 'Female','Others'))
	Email	varchar(100)	Not null, unique, check for invalid characters, multiple '@' symbols etc.



	Phone_number	varchar(15)	Not null, unique, length should not exceed 15 characters(including code)
	Date_of_Birth (DOB)	Date	Not null, should be ideally less than or equal to current date.
	Availability_Status	TinyInt	Not null, value lies in {0,1} works as kind of a boolean check
	no_of_Games_refereed	int	Not null, unsigned, default starting value = 0
<b>FK</b>	Specialization_Sport_ID	int	Not null, unsigned, unique,

Referential Integrity Constraints :

- Foreign Key Specialization\_Sport\_ID references to Sport(Sport\_ID).

## COACH

<b>PK</b>	Coach_ID	int	Not null, unsigned, unique
	First_name	varchar(25)	Not null
	Last_name	varchar(25)	Not null
	Gender	varchar(6)	Not Null, check(Gender in ('Male', 'Female', 'Others'))
	Email	varchar(100)	Not null, unique, check for invalid characters, multiple '@' symbols etc.

	Phone_number	varchar(15)	Not null, unique, length should not exceed 15 characters(including code)
	Date_of_Birth (DOB)	Date	Not null, should be ideally less than or equal to current date.
	Availability_Status	TinyInt	Not null, value lies in {0,1} works as kind of a boolean check
	no_of_player_coached	int	Not null, unsigned, default starting value = 0
	Rating	float	Not null, unsigned, lies b/w [1,5]
<b>FK</b>	Specialization_Sport_ID	int	Not null, unsigned, unique

Referential Integrity Constraints :

- Foreign Key Specialization\_Sport\_ID references to Sport(Sport\_ID).

## **SPORT**

<b>PK</b>	Sport_ID	int	Not null, unsigned, unique
	Name	varchar(100)	Not null
	Maximum_Players	int	Not null, unsigned, greater than zero

## **EQUIPMENT**

<b>PK</b>	Equipment_ID	int	Not null, unique, unsigned
	Name	varchar(100)	Not null
	Availability_Status	TinyInt	Not null, value lies in {0,1} works as kind of a boolean check
	Condition	varchar(7)	Not null, check(Condition in (‘Poor’, ‘Average’, ‘Good’))
<b>FK</b>	Sport_ID	int	Not null, unique, unsigned
<b>FK</b>	Owner_ID	int	Can be null, (due to many to one, partial participation)

Referential Integrity Constraints :

- Foreign Key Sport\_ID references to Sport(Sport\_ID).
- Foreign Key Owner\_ID references the Owner\_ID part of primary key(formed from conjunction of attributes with foreign key) of Student Inventory which further references to User(User\_ID). This helps us to uniquely identify the owner of the inventory.

## **VENDOR**

<b>PK</b>	Vendor_ID	int	Not null, unsigned, unique
	Shop_name	varchar(100)	Not null
	Email	varchar(100)	Not null, unique, check for invalid characters, multiple ‘@’ symbols

			etc.
	Phone_number	varchar(15)	Not null, unique, length should not exceed 15 characters(including code)
	no_of_equipment_repaired	int	Not null, unsigned, default = 0
	Address_Location	varchar(200)	Not null

## **SPORT ADMINISTRATOR**

<b>PK</b>	Admin_ID	int	Not Null, unique, unsigned
	First_name	varchar(25)	Not null
	Last_name	varchar(25)	Not null
	Email	varchar(100)	Not null, unique, check for invalid characters, multiple '@' symbols etc.
	Phone_number	varchar(15)	Not null, unique, length should not exceed 15 characters(including code)
	Office_Location	varchar(200)	Not null

<b>FK</b>	Managing_Sport_ID	int	Not Null, unique, unsigned
-----------	-------------------	-----	----------------------------

Referential Integrity Constraints :

- Foreign Key Managing\_Sport\_ID references to Sport(Sport\_ID).

## **STUDENT\_INVENTORY**

<b>Discriminator</b>	Inventory_ID	int	Not null, unsigned, unique
	Item_Count	int	Not Null, unsigned, default starting value equal to 0
	Owner_id	int	Not null, unsigned, unique

Primary Key : (Owner\_ID, Inventory\_ID)

Referential Integrity Constraints :

- Foreign Key User\_ID references to User(User\_ID)

## **COLLEGE\_EQUIPMENTS**

<b>Discriminator</b>	Name	varchar(100)	Not null
	Availability_Status	TinyInt	Not null
	Condition	varchar(7)	Not null, check(Condition in ('Poor','Average', 'Good'))
	Equipment_ID	int	Not Null, unique, unsigned, references EQUIPMENT(Equipment_ID)

	Manufacturer	varchar(20)	Not null
	Owner_Admin_ID	int	Not null, unsigned, unique

Primary Key : (Name, Equipment\_ID, Owner\_Admin\_ID)

Referential Integrity Constraints :

- Foreign Key Owner\_Admin\_ID references to Sport Administrator(Admin\_ID).
- Foreign Key Equipment\_ID references to Equipment(Equipment\_ID).

## **ORDER**

<b>Discriminator</b>	Order_ID	int	Not Null, unique, unsigned
	Type	varchar(100)	Not null
	Order_Status	varchar(8)	Not null, check(Order_Status in ('COMPLETE', 'PENDING'))
	User_ID	int	Not null, unique, unsigned, references USER(User_ID)
	Equipment_ID	int	Not Null, unique, unsigned, references EQUIPMENT(Equipment_ID)
	Vendor_ID	int	Not null, unsigned, unique

Primary Key : (User\_ID, Order\_ID)

Referential Integrity Constraints :

- Foreign Key Vendor\_ID references to Vendor(Vendor\_ID)
- Foreign Key Equipment\_ID references to Equipment(Equipment\_ID).
- Foreign Key User\_ID references to User(User\_ID)

## **FAVORITES**

<b>FK</b>	Sport_ID	int	Not null, unique, unsigned
<b>FK</b>	User_ID	int	Not null, unique, unsigned

- Primary Key : (User\_ID, Sport\_ID)
- Referential Integrity Constraints :
  - Foreign Key Sport\_ID references to Sport(Sport\_ID)
  - Foreign Key User\_ID references to User(User\_ID)

## **TRAINS**

<b>FK</b>	Coach_ID	int	Not null, unique, unsigned
<b>FK</b>	User_ID	int	Not null, unique, unsigned
	Start_Date	Date	not null

Primary Key : (User\_ID, Sport\_ID, Start\_Date)

- Referential Integrity Constraints :
  - Foreign Key Coach\_ID references to Sport(Coach\_ID)
  - Foreign Key User\_ID references to User(User\_ID)

## **FRIEND**

<b>FK</b>	User_ID	int	Not null, unique, unsigned
<b>FK</b>	Friend_ID	int	Not null, unique, unsigned

	Date_of_Adding	Date	not null
--	----------------	------	----------

Primary Key : (User\_ID, Sport\_ID, Date\_of\_Adding)

- Referential Integrity Constraints :
  - Foreign Key Friend\_ID references to User(User\_ID)
  - Foreign Key User\_ID references to User(User\_ID)

## **ISSUES EQUIPMENTS**

<b>FK</b>	Borrower_ID	int	Not null, unique, unsigned
<b>FK</b>	Equipment_ID	int	Not null, unique, unsigned
	Issue_Date_Time	DateTime	not null
	Expected_Return_Date_Time	DateTime	Not null, Expected_Return_Date_Time <= Issue_Date_Time

Primary Key : (Borrower\_ID, Equipment\_ID)

- Referential Integrity Constraints :
  - Foreign Key Borrower\_ID references to User(User\_ID)
  - Foreign Key Equipment\_ID references to Equipment(Equipment\_ID).

## **JUDGES**

<b>FK</b>	Event_ID	int	Not null, unique, unsigned
<b>FK</b>	Referee_ID	int	Not null, unique, unsigned

Primary Key : (Event\_ID, Referee\_ID)



- Referential Integrity Constraints :
  - Foreign Key Event\_ID references to Sport\_Event(Event\_ID)
  - Foreign Key Referee\_ID references to Referee(Referee\_ID)

## **REPAIRS (COLLEGE EQUIPMENTS)**

<b>FK</b>	Vendor_ID	int	Not null, unsigned, unique
<b>FK</b>	College_Equipment_ID	int	Not Null, unsigned, unique
	Date_of_Repair	Date	Not null
	Date_of_Return	Date	Not null, Date_of_Return >= Date_of_Repair

- Primary Key : (Vendor\_ID, College\_Equipment\_ID)
- Referential integrity Constraints :
  - Foreign Key Vendor\_ID references to Vendor(Vendor\_ID)
  - Foreign Key College\_Equipment\_ID references Equipment\_ID part of the primary key (formed from conjunction of attributes with foreign key) of College Equipment which further references Equipment(Equipment\_ID).

## **PARTICIPATES\_IN**

<b>FK</b>	User_ID	int	Not null, unique, unsigned
<b>FK</b>	Event_ID	int	Not null, unique, unsigned

- Primary Key : (User\_ID, Event\_ID)
- Referential Integrity Constraints :
  - Foreign Key Event\_ID references to Sport\_Event(Event\_ID)
  - Foreign Key User\_ID references to User(User\_ID)

## **BORROWS EQUIPMENTS**

<b>FK</b>	Lender_ID	int	Not null, unique, unsigned
<b>FK</b>	Borrower_ID	int	Not null, unique, unsigned
<b>FK</b>	Equipment_ID	int	Not null, unique, unsigned
	Issue_Date_Time	DateTime	Not null
	Pickup_Address	Varchar(200)	Not null
	Expected_Return_Date_Time	DateTime	Not null, Expected_Return_Date_Time <= Issue_Date_Time

Primary Key : (Lender\_ID, Borrower\_ID, Equipment\_ID)

- Referential Integrity Constraints :
  - Foreign Key Lender\_ID references to User(User\_ID)
  - Foreign Key Borrower\_ID references to User(User\_ID)
  - Foreign Key Equipment\_ID references to Equipment(Equipment\_ID)

## **Week 6 :**

After creating the ER and designing the schema, we move on to make the tables and populate them with meaningful and sizable data.

These are some of the screenshots of the tables formed by us.

# 1. Administrator\_DB

MySQL Workbench

DBMS Test x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

dbms\_project

Tables

administrator\_db

Views

Stored Procedures

Functions

sys

Query 1 x

```
9      ,Office_Location VARCHAR(50) NOT NULL
10      ,Managing_Sport_ID INTEGER NOT NULL
11  );
12
13  SELECT * FROM Administrator_DB;
```

Result Grid

Admin_ID	First_name	Last_name	Email	Phone_number	Office_Location	Managing_Sport_ID
1	Lalo	Mounfield	lmounfield0@home.pl	627-238-6613	6 McGuire Avenue	10
2	Helsa	Petow	hpetow1@squarespace.com	323-869-8613	20 Vera Parkway	7
3	Virginia	Lyles	vlyles2@un.org	209-778-4226	841 Cambridge Parkway	10
4	Chevalier	Dwyr	cdwyr3@deliciousdays.com	481-539-2266	14989 Glendale Street	3
5	Althea	Powton	apowton4@feedburner.com	604-651-1059	478 Lakeland Crossing	7
6	Beryle	Gresch	bgresch5@umich.edu	321-203-0328	7855 Swallow Crossing	2
7	Annecorinne	O'Cudde	aocudde6@livejournal.com	120-455-3594	211 Cascade Trail	12
8	Rubetta	Pregel	rpregel7@don.ne.jp	334-958-8293	78 McBride Lane	6
9	Tessie	Packington	tpackington8@youku.com	949-554-6909	713 Packers Crossing	9
10	Cairistiona	Meininking	cmenninking9@typepad.com	523-349-4385	92 Milwaukee Crossing	8

Table: administrator\_db

Columns:

- Admin\_ID
- First\_name
- Last\_name
- Email
- Phone\_number

Action Output

#	Time	Action	Message	Duration / Fetch
9	22:43:17	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.administrator_db' ('Admin_ID...	OK	0.000 sec
10	22:43:17	DEALLOCATE PREPARE stmt	OK	0.000 sec
11	22:43:45	SELECT * FROM Administrator_DB LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

100%

ENG 10:43 PM 21-02-2021

## 2. Borrow\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'Borrow\_DB' table is highlighted in the Schemas pane. The query editor shows a query to select all data from the 'Borrow\_DB' table. The result grid displays the following data:

Lender_ID	Borrower_ID	Equipment_ID	issue_Date_Time	Expected_Return_Date_Time	Pickup_Address
54	74	96	2020-10-23 12:20:32	2020-10-23 15:20:32	410 Annamark Pass
56	544	38	2020-03-30 0:53:48	2020-03-30 1:53:48	74390 Truax Lane
91	147	78	2020-01-29 3:48:09	2020-01-29 4:48:09	9 Crowley Terrace
163	756	139	2020-04-26 22:08:36	2020-04-27 0:08:36	77909 Logan Pass
183	518	105	2020-05-29 21:26:38	2020-05-30 0:26:38	7968 Nancy Trail
188	749	138	2020-12-01 8:40:35	2020-12-01 12:40:35	196 Jackson Pass
214	216	67	2020-11-04 3:44:23	2020-11-04 4:44:23	468 McCormick Place
260	711	53	2020-05-10 1:50:24	2020-05-10 2:50:24	95 Eastlawn Way
307	694	98	2020-04-02 13:01:37	2020-04-02 15:01:37	6 Boyd Trail
332	184	66	2020-10-25 15:09:11	2020-10-25 18:09:11	44325 Garrison Way

The output pane shows the execution of the query, indicating that 80 row(s) were returned.

## 3. CollegeEquip\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'CollegeEquip\_DB' table is highlighted in the Schemas pane. The query editor shows a query to select all data from the 'CollegeEquip\_DB' table. The result grid displays the following data:

Equipment_ID	Name	Availability_Status	Conditions	Manufacturer	Owner_Admin_ID
161	Goggles	0	Poor	Dasya	19
162	Bathing caps	1	Good	Florinda	8
163	Flotation devices	0	Average	Margaretha	8
164	Kickboards	1	Poor	Florenza	19
165	Bathing caps	1	Good	Tandie	19
166	Flotation devices	0	Good	Fleurette	19
167	Kickboards	1	Average	Boyce	8
168	Badminton Racquet	0	Good	Winnie	13
169	Shuttlecock	1	Good	Winnie	18
170	Shuttlecock	1	Average	Chase	18

The output pane shows the execution of the query, indicating that 40 row(s) were returned.

## 4. Equipment\_DB

MySQL Workbench

DBMS Test x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

dbms\_project

Tables

administrator\_db

borrow\_db

coach\_db

collegeequip\_db

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Schema: dbms\_project

Query 1 x

Limit to 1000 rows

57 ,Manufacturer VARCHAR(20) NOT NULL

58 ,Owner\_Admin\_ID INTEGER NOT NULL

59 );

60

61 • SELECT \* FROM Equipment\_DB;

Result Grid

Equipment\_ID Name Availability\_Status Conditions Manufacturer Owner\_Admin\_ID

1	Badminton Racquet	1	Good	1	18
2	Shuttlecock	1	Average	1	37
3	Badminton Shoes	1	Average	1	60
4	Badminton Net	0	Poor	1	75
5	Tennis Racquet	1	Good	2	76
6	Tennis Ball	0	Poor	2	79
7	Basketball	1	Good	3	87
8	Squash Racquet	1	Good	5	122
9	Squash Ball	0	Average	5	159
10	Swimsuit	1	Poor	6	164

Equipment\_DB 6 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
52	22:50:57	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.equipment_db' ('Equipment_ID', 'Name', 'Availability_Status', 'Conditions', 'Manufacturer', 'Owner_Admin_ID') VALUES ('1', 'Badminton Racquet', '1', 'Good', '1', '18')'	OK	0.000 sec
53	22:50:58	DEALLOCATE PREPARE stmt	OK	0.000 sec
54	22:51:16	SELECT * FROM Equipment_DB LIMIT 0, 1000	157 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Type here to search

100%

10:51 PM 21-02-2021

## 5. Event\_DB

MySQL Workbench

DBMS Test x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

dbms\_project

Tables

administrator\_db

borrow\_db

coach\_db

collegeequip\_db

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Schema: dbms\_project

Query 1 x

Limit to 1000 rows

73 ,Venue\_ID INTEGER NOT NULL

74 ,Sport\_ID INTEGER NOT NULL

75 );

76

77 • SELECT \* FROM Event\_DB;

Result Grid

Event_ID	Event_Name	Event_Date	Duration	Start_Time	End_Time	Participant_Limit	Organizer_ID	Date_of_Booking	Venue_ID	Sport_ID
1	Cricket Match	2020-05-22	1	20:12:06	21:12:06	4	871	2020-05-19	8	7
2	Badminton Duels	2020-10-10	1	19:31:06	20:31:06	3	616	2020-10-08	1	1
3	Volleyball Selections	2020-12-11	3	0:07:49	3:07:49	2	134	2020-12-07	2	11
4	Basketball Trials	2020-05-27	2	11:22:58	13:22:58	3	900	2020-05-25	12	3
5	Football	2021-01-12	3	6:58:27	9:58:27	4	172	2021-01-08	19	9
6	Football Match	2020-12-15	4	13:14:35	17:14:35	3	254	2020-12-13	7	8
7	Swimming Practice	2020-08-25	4	14:53:14	18:53:14	2	738	2020-08-22	6	6
8	Billiards	2020-04-01	3	4:43:20	7:43:20	3	932	2020-03-30	7	8
9	Badminton Men Singles	2020-07-20	3	16:46:41	19:46:41	4	655	2020-07-18	4	9
10	Cricket Match	2020-01-10	4	2:17:36	6:17:36	1	97	2020-01-08	8	7

Event\_DB 7 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
60	22:52:18	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.event_db' ('Event_ID', 'Event_Name', 'Event_Date', 'Duration', 'Start_Time', 'End_Time', 'Participant_Limit', 'Organizer_ID', 'Date_of_Booking', 'Venue_ID', 'Sport_ID') VALUES ('1', 'Cricket Match', '2020-05-22', '1', '20:12:06', '21:12:06', '4', '871', '2020-05-19', '8', '7')'	OK	0.000 sec
61	22:52:18	DEALLOCATE PREPARE stmt	OK	0.000 sec
62	22:52:24	SELECT * FROM Event_DB LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

100%

10:52 PM 21-02-2021

## 6.Favourite\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'Favourite\_DB' table is highlighted in the Navigator. The table structure is defined as follows:

```
81      ,Sport_ID INTEGER NOT NULL
82  );
83
84  • SELECT * FROM Favourite_DB;
85
```

The Result Grid displays the following data:

User_ID	Friend_ID	Date_of_Adding
1	1	11/21/2020
2	2	7/5/2020
3	3	1/25/2020
4	4	8/31/2020
5	5	2/5/2020
6	6	10/18/2020
7	7	2/18/2020
8	8	2/13/2020
9	9	8/8/2020
10	10	4/7/2020

The Output pane shows the execution of the query, with a message indicating that 700 row(s) were returned.

## 7.Judge\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'Judge\_DB' table is highlighted in the Navigator. The table structure is defined as follows:

```
1 • SELECT * FROM dbms_project.judge_db;
```

The Result Grid displays the following data:

Event_ID	Referee_ID
41	11
79	15
11	32
96	10
25	35
47	49
93	2
187	6

The Output pane shows the execution of the query, with a message indicating that 20 row(s) were returned. An error message is also visible: 'Error Code: 1146: Table 'dbms\_project.judge\_db' doesn't exist'.



## 8. Participation\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'participatesin\_db' table is highlighted in the Schemas pane. The main query window displays the following SQL query:

```
1 • SELECT * FROM dbms_project.participatesin_db;
```

The Result Grid shows the following data:

User_ID	Event_ID
466	75
754	26
940	60
575	70
57	74
464	16
328	23
424	31
716	5
64	3

The Output pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
178	23:08:44	CREATE TABLE 'dbms_project'.participatesin_db ('User_ID' int, 'Event_ID' int)	OK	0.000 sec
179	23:08:44	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.participatesin_db ('User_ID', ...	OK	0.000 sec
180	23:08:45	DEALLOCATE PREPARE stmt	OK	0.000 sec

## 9. Sports\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'sports\_db' table is highlighted in the Schemas pane. The main query window displays the following SQL query:

```
1 • SELECT * FROM dbms_project.sports_db;
```

The Result Grid shows the following data:

Sport_ID	Name	Max_Players
1	Badminton	4
2	Tennis	4
3	Basketball	10
4	Table Tennis	4
5	Squash	4
6	Swimming	10
7	Cricket	22
8	Football	22
9	Football	4
10	Chess	2

The Output pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
128	23:01:05	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.sports_db ('Sport_ID', Nam...	OK	0.000 sec
129	23:01:05	DEALLOCATE PREPARE stmt	OK	0.000 sec
130	23:01:17	SELECT * FROM dbms_project.sports_db LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec

## 10. Trains\_DB

MySQL Workbench interface showing the **Trains\_DB** schema. The **Navigator** pane on the left lists the schema structure, including tables like **colleeequip\_db**, **equipment\_db**, **event\_db**, **favourite\_db**, **friend\_db**, **issueequip\_db**, **judges\_db**, **order\_db**, **referee\_db**, **sports\_db**, **studentinventory\_db**, **trains\_db**, **views**, **stored procedures**, **functions**, and **sys**. The **Table: trains\_db** is selected, showing columns: **Coach\_ID** (int), **User\_ID** (int), and **Start\_Date** (text).

The **Query 1** window shows the following SQL query:

```
SELECT * FROM dbms_project.trains_db;
```

The **Result Grid** displays the following data:

Coach_ID	User_ID	Start_Date
27	294	2020-02-10
3	52	2020-03-17
45	552	2020-06-07
8	961	2020-03-03
7	530	2020-07-21
18	14	2020-11-26
46	236	2020-06-10
26	95	2020-12-21
32	69	2021-02-07
14	105	2020-02-25

The **Output** pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
142	23:02:59	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.trains_db' ('Coach_ID','User...'	OK	0.000 sec
143	23:03:00	DEALLOCATE PREPARE stmt	OK	0.000 sec
144	23:03:10	SELECT * FROM dbms_project.trains_db LIMIT 0, 1000	200 row(s) returned	0.000 sec / 0.000 sec

## 11. Vendor\_DB

MySQL Workbench interface showing the **Vendor\_DB** schema. The **Navigator** pane on the left lists the schema structure, including tables like **equipment\_db**, **event\_db**, **favourite\_db**, **friend\_db**, **issueequip\_db**, **judges\_db**, **order\_db**, **referee\_db**, **sports\_db**, **studentinventory\_db**, **trains\_db**, **user\_db**, **vendor\_db**, **views**, **stored procedures**, **functions**, and **sys**. The **Table: vendor\_db** is selected, showing columns: **Vendor\_ID** (int), **Shop\_name** (text), **Email** (text), **Phone\_number** (text), **no\_of\_equipment\_repaired** (int), and **Address\_Location** (text).

The **Query 1** window shows the following SQL query:

```
SELECT * FROM dbms_project.vendor_db;
```

The **Result Grid** displays the following data:

Vendor_ID	Shop_name	Email	Phone_number	no_of_equipment_repaired	Address_Location
1	Dasya	dclampe0@arizona.edu	729-204-8647	4	1 Springs Junction
2	Florinda	femoney1@foxnews.com	960-980-1099	2	7 Clyde Gallagher Hill
3	Margaretha	mchoak2@army.mil	973-800-6224	4	764 Haas Parkway
4	Cindra	cdryburgh3@sun.com	777-820-4525	3	36 Hauk Circle
5	Marcelle	mmomme4@pen.io	833-950-0778	3	0486 Hovde Lane
6	Tandie	tcurlis5@biblegateway.com	402-189-8508	3	99 Elka Terrace
7	Fleurette	fdlyde5@bluehost.com	676-964-1587	1	37 Lakewood Trail
8	Boyce	bashbey7@godaddy.com	437-922-7143	2	1107 Monument Center
9	Winnie	wbremner8@acquirethisname.com	970-344-0806	4	40121 Hanson Pass
10	Chase	cregus9@del.com	118-294-2665	5	53002 Carioca Parkway

The **Output** pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
156	23:04:37	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.vendor_db' ('Vendor_ID','S...'	OK	0.000 sec
157	23:04:37	DEALLOCATE PREPARE stmt	OK	0.000 sec
158	23:04:47	SELECT * FROM dbms_project.vendor_db LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec



## 12. Coach\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'Coach\_DB' table is highlighted in the Schemas pane. The query editor shows a query to select all data from the Coach\_DB table. The result grid displays 10 rows of data with columns: Coach\_ID, First\_name, Last\_name, Gender, Email, Phone\_number, Date\_of\_Birth, Availability\_Status, Ratings, and Specialization\_Sport\_ID.

Coach_ID	First_name	Last_name	Gender	Email	Phone_number	Date_of_Birth	Availability_Status	Ratings	Specialization_Sport_ID
1	Jileen	Ketchen	Female	jfletcher0@chron.com	830-719-6261	1973-06-16	1	4	8
2	Vicki	Barnham	Male	vbarnham1@ask.com	291-769-6974	1986-01-23	0	1	9
3	Gonzalo	Fern	Male	gferrn2@lund1.de	965-672-0496	1974-12-24	0	5	8
4	Tisha	Maddern	Female	tmaddern3@marketwatch.com	131-938-3953	1978-03-31	0	4	11
5	Ardine	Camelli	Female	acamelli4@hstats.com	802-263-2441	1988-07-02	0	1	12
6	Korie	Huish	Female	khuish5@posterous.com	167-395-5531	1971-07-15	0	4	11
7	Roxi	Swithenby	Female	rswithenby6@phoca.cz	859-899-1714	1976-06-16	1	5	2
8	Ephrem	Ingolotti	Female	eingolotti7@sbwire.com	880-938-8392	1984-08-11	0	5	1
9	Dennison	MacGiany	Male	dmacgiany8@bbc.co.uk	111-361-9404	1984-11-29	1	2	7
10	Jere	Slucock	Male	jslucock9@myu.edu	354-486-8351	1973-12-01	0	5	11

The output pane shows the execution of the query, with a message indicating that 50 rows were returned.

## 13. IssueEquip\_DB

The screenshot shows the MySQL Workbench interface with the 'dbms\_project' schema selected. The 'IssueEquip\_DB' table is highlighted in the Schemas pane. The query editor shows a query to select all data from the IssueEquip\_DB table. The result grid displays 10 rows of data with columns: Borrower\_ID, Equipment\_ID, Issue\_Date\_Time, and Expected\_Return\_Date\_Time.

Borrower_ID	Equipment_ID	Issue_Date_Time	Expected_Return_Date_Time
914	194	2020-04-21 3:33:07	2020-04-21 5:33:07
673	161	2020-04-25 23:35:00	2020-04-26 0:35:00
572	192	2020-03-01 10:11:37	2020-03-01 11:11:37
834	199	2020-06-22 10:49:36	2020-06-22 14:49:36
727	169	2020-03-19 21:16:32	2020-03-20 1:16:32
634	178	2020-08-24 16:14:20	2020-08-24 18:14:20
941	194	2020-05-18 4:48:17	2020-05-18 8:48:17
779	162	2020-09-26 19:21:41	2020-09-26 21:21:41
614	191	2020-06-03 2:01:49	2020-06-03 4:01:49
656	195	2020-05-16 19:24:57	2020-05-16 20:24:57

The output pane shows the execution of the query, with a message indicating that 20 rows were returned.

## 14. Referee\_DB

MySQL Workbench DBMS Test

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 judges\_db order\_db referee\_db sports\_db studentinventory\_db

SCHEMAS

Filter objects

- coach\_db
- collegetequip\_db
- equipment\_db
- event\_db
- favourite\_db
- friend\_db
- issueequip\_db
- judges\_db
- order\_db
- referee\_db
- sports\_db
- studentinventory\_db
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas

Information

Table: studentinventory\_db

Columns:

- Inventory\_ID int
- Item Count int
- Owner\_ID int

Object Info Session

Query 1

```
SELECT * FROM dbms_project.studentinventory_db;
```

Result Grid

Inventory_ID	Item Count	Owner_ID
71	0	71
72	0	72
73	0	73
74	0	74
75	1	75
76	1	76
77	0	77
78	0	78
79	1	79

Output

Action Output

#	Time	Action	Message	Duration / Fetch
135	23.01.49	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.studentinventory_db' ('Inven...	OK	0.000 sec
136	23.01.54	DEALLOCATE PREPARE stmt	OK	0.000 sec
137	23.02.05	SELECT * FROM dbms_project.studentinventory_db LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.000 sec

Windows Taskbar: Type here to search, 100%, 11:02 PM 21-02-2021

## 15. Venue\_DB

MySQL Workbench DBMS Test

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 judges\_db order\_db referee\_db sports\_db studentinventory\_db trains\_db user\_db vendor\_db venue\_db

SCHEMAS

Filter objects

- event\_db
- favourite\_db
- friend\_db
- issueequip\_db
- judges\_db
- order\_db
- referee\_db
- sports\_db
- studentinventory\_db
- trains\_db
- user\_db
- vendor\_db
- venue\_db
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

Table: venue\_db

Columns:

- Venue\_ID int
- Venue\_Name text
- Venue\_Type text
- Opening\_Time text
- Closing\_Time text
- Location text
- Occupancy\_Status int
- Sport\_ID int

Object Info Session

Query 1

```
SELECT * FROM dbms_project.venue_db;
```

Result Grid

Venue_ID	Venue_Name	Venue_Type	Opening_Time	Closing_Time	Location	Occupancy_Status	Sport_ID
1	Badminton Court	Outdoor	6:52	11:42	Inside Sports Complex	0	1
2	Volleyball Court	Outdoor	10:45	6:36	In Ground	0	11
3	Tennis Court	Outdoor	11:03	6:45	Inside Sports Complex	1	2
4	Foosball Table	Indoor	3:03	22:49	Boys Common Room	0	9
5	Squash Court	Outdoor	6:41	14:58	Inside Sports Complex	0	5
6	Swimming Pool	Indoor	12:54	19:07	Inside Sports Complex	0	6
7	Field	Outdoor	6:05	19:18	Inside Ground	0	8
8	Cricket Pitch	Outdoor	15:18	3:21	Inside Ground	1	7
9	Chess club	Indoor	9:03	20:28	Club Room	1	10
10	Library	Indoor	19:20	19:41	Near Old Academics Building	0	10

Output

Action Output

#	Time	Action	Message	Duration / Fetch
163	23.05.26	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.venue_db' ('Venue_ID','Ve...	OK	0.000 sec
164	23.05.26	DEALLOCATE PREPARE stmt	OK	0.000 sec
165	23.05.35	SELECT * FROM dbms_project.venue_db LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec

Windows Taskbar: Type here to search, 100%, 11:05 PM 21-02-2021

## 16. Friend\_DB

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'dbms\_project' schema with various databases like 'administrator\_db', 'borrow\_db', 'coach\_db', 'colleeequip\_db', 'friend\_db', 'issueequip\_db', 'judges\_db', 'order\_db', 'referee\_db', 'sports\_db', and 'studentinventory\_db'. The main window shows a query editor with the following SQL code:

```
81      ,Sport_ID INTEGER NOT NULL
82    );
83
84 • SELECT * FROM Favourite_DB;
85
```

The 'Result Grid' shows the output of the query, displaying columns 'User\_ID', 'Friend\_ID', and 'Date\_of\_Adding' with 10 rows of data.

User_ID	Friend_ID	Date_of_Adding
1	1	11/21/2020
2	2	7/5/2020
3	3	1/25/2020
4	4	8/31/2020
5	5	2/5/2020
6	6	10/18/2020
7	7	2/18/2020
8	8	2/13/2020
9	9	8/8/2020
10	10	4/7/2020

The 'Output' pane shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
89	22:55:33	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.friend_db' ('User_ID','Friend...	OK	0.000 sec
90	22:55:37	DEALLOCATE PREPARE stmt	OK	0.000 sec
91	22:55:44	SELECT * FROM Friend_DB LIMIT 0, 1000	700 row(s) returned	0.000 sec / 0.000 sec

## 17. StudentInventory\_DB

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'dbms\_project' schema with various databases like 'coach\_db', 'colleeequip\_db', 'equipment\_db', 'event\_db', 'favourite\_db', 'friend\_db', 'issueequip\_db', 'judges\_db', 'order\_db', 'referee\_db', 'sports\_db', and 'studentinventory\_db'. The main window shows a query editor with the following SQL code:

```
1 • SELECT * FROM dbms_project.studentinventory_db;
```

The 'Result Grid' shows the output of the query, displaying columns 'Inventory\_ID', 'Item Count', and 'Owner\_ID' with 9 rows of data.

Inventory_ID	Item Count	Owner_ID
71	0	71
72	0	72
73	0	73
74	0	74
75	1	75
76	1	76
77	0	77
78	0	78
79	1	79

The 'Output' pane shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
135	23:01:49	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.studentinventory_db' ('Inven...	OK	0.000 sec
136	23:01:54	DEALLOCATE PREPARE stmt	OK	0.000 sec
137	23:02:05	SELECT * FROM dbms_project.studentinventory_db LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.000 sec

## 18. User\_DB

MySQL Workbench

DBMS Test x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 judges\_db order\_db referee\_db sports\_db studentinventory\_db trains\_db user\_db x

SCHEMAS

Filter objects

collegetequip\_db  
equipment\_db  
event\_db  
favourite\_db  
friend\_db  
issueequip\_db  
judges\_db  
order\_db  
referee\_db  
sports\_db  
studentinventory\_db  
trains\_db  
user\_db

Views  
Stored Procedures  
Functions

Administration Schemas

Information

Schema: dbms\_project

Query 1

```
SELECT * FROM dbms_project.user_db;
```

Result Grid

User_ID	First_name	Last_name	Email	Phone_Number	Date_of_Birth	Gender	Date_of_Joining	Address
1	Clark	Sproat	csproat0@behance.net	383-550-8477	1993-01-31	Male	2019-06-10	8112 Elmside Avenue
2	Bobby	Duligall	bduligall1@delicious.com	231-428-5523	1997-11-20	Female	2019-04-03	2302 Rusk Plaza
3	Tammy	Prichtet	tprichtet2@boston.com	490-227-7278	1994-02-22	Male	2019-03-06	1 2nd Parkway
4	Fritz	Rickeard	frickeard3@gnu.org	765-807-5573	1991-11-05	Female	2019-12-02	7824 Kingsford Junction
5	Loni	Andrichak	landrichak4@skyrock.com	327-182-3708	1995-12-20	Male	2019-06-24	87704 Lukken Road
6	Celle	Bordiss	cbordiss5@psu.edu	363-407-8233	1996-05-21	Female	2019-08-26	91 Towne Way
7	Ario	Bickerdyke	abickerdyke6@dyndns.org	933-155-8949	1990-01-22	Male	2019-04-30	48793 Larry Lane
8	Dorian	Eadie	deadie7@nytimes.com	771-225-2321	1991-06-29	Female	2019-02-12	10 Sundown Drive
9	Marcela	Barnbrook	mbarnbrook8@mail.ru	964-438-7447	1990-06-01	Female	2019-07-26	066 Sullivan Road
10	Trude	MacDearmid	tmacdearmid9@thetimes.co.uk	625-485-5160	1995-05-27	Female	2019-09-01	7566 Fairfield Point

Output

#	Time	Action	Message	Duration / Fetch
149	23.03.48	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.user_db' ('User_ID','First_na...	OK	0.000 sec
150	23.03.54	DEALLOCATE PREPARE stmt	OK	0.000 sec
151	23.04.05	SELECT * FROM dbms_project.user_db LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Windows taskbar: Type here to search, 100%, 11:04 PM 21-02-2021

## 19. Order\_DB

MySQL Workbench

DBMS Test x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 judges\_db order\_db x

SCHEMAS

Filter objects

dbms\_project

Tables

administrator\_db  
borrow\_db  
coach\_db  
collegetequip\_db  
equipment\_db  
event\_db  
favourite\_db  
friend\_db  
issueequip\_db  
judges\_db  
order\_db

Columns  
Indexes  
Foreign Keys

Administration Schemas

Information

Table: order\_db

Columns:

Column	int	text	int	int
Order_ID				
Type				
Order_Status				
User_ID				
Equipment_ID				
Vendor_ID				

Query 1

```
SELECT * FROM dbms_project.order_db;
```

Result Grid

Order_ID	Type	Order_Status	User_ID	Equipment_ID	Vendor_ID
1	Repair	Complete	466	139	19
2	Buy	Complete	754	41	13
3	Buy	Pending	940	102	6
4	Buy	Pending	575	17	18
5	Repair	Pending	57	51	8
6	Buy	Pending	464	30	11
7	Repair	Complete	328	48	8
8	Buy	Pending	424	60	9
9	Repair	Complete	716	2	6
10	Repair	Pending	64	105	11

Output

#	Time	Action	Message	Duration / Fetch
114	22.59.20	PREPARE stmt FROM 'INSERT INTO 'dbms_project'.order_db' ('Order_ID','Type'...	OK	0.000 sec
115	22.59.21	DEALLOCATE PREPARE stmt	OK	0.000 sec
116	22.59.28	SELECT * FROM dbms_project.order_db LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Windows taskbar: Type here to search, 100%, 10:59 PM 21-02-2021

## Individual Contribution :

As a team, we gelled pretty well together. We regularly met on Google Meet and Discord and had timely meetings. We all worked together with inputs coming from all the members equally. Almost all the work done has some or the other part of everyone's contribution. We divided the work equally, made the tables together. All the work was reviewed by every other teammate and thorough discussion was done on the same too.

Samyak Jain : Active participation in the ideation process, Coming up with Stakeholder queries, decided on some part of entities and attributes, Forming the ER Diagram, designing the schema and making the schema, Creation of tables, filling up some of tables.

Kshitij Mohan : Active participation in the ideation process, Coming up with Stakeholder queries, decided on the part of entities and attributes, making the schema, Creation of tables, Filling up the tables

Harshal Dev : Active participation in the ideation process, Forming and Writing the problem statement, decided on the part of entities and attributes, Forming the ER Diagram, designing the schema, creation of some tables, filling up the tables.

Satwik Tiwari : Active participation in the ideation process, Forming and Writing the problem statement, decided on part of entities and attributes, making the schema, creation of some tables, filling up the tables

Hitesh Garg : Active participation in the ideation process, Coming up with Stakeholder queries, decided on spart of entities and attributes, Forming the ER diagram, making the schema, creation of some tables, filling up tables.



## Code :

```
CREATE TABLE Administrator_DB(  
    Admin_ID      INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY  
KEY  
    ,First_name   VARCHAR(20) NOT NULL  
    ,Last_name    VARCHAR(20) NOT NULL  
    ,Email        VARCHAR(50) NOT NULL  
    ,Phone_number VARCHAR(12) NOT NULL  
    ,Office_Location VARCHAR(50) NOT NULL  
    ,Managing_Sport_ID INTEGER NOT NULL  
);
```

```
SELECT * FROM Administrator_DB;
```

```
CREATE TABLE Borrow_DB(  
    Lender_ID      INTEGER UNSIGNED NOT NULL UNIQUE  
PRIMARY KEY  
    ,Borrower_ID   INTEGER NOT NULL  
    ,Equipment_ID  INTEGER NOT NULL  
    ,issue_Date_Time VARCHAR(19) NOT NULL  
    ,Expected_Return_Date_Time VARCHAR(19) NOT NULL  
    ,Pickup_Address VARCHAR(50) NOT NULL  
);
```

```
SELECT * FROM Borrow_DB;
```

```
CREATE TABLE Coach_DB(  
    Coach_ID      INTEGER UNSIGNED NOT NULL UNIQUE  
PRIMARY KEY  
    ,First_name   VARCHAR(20) NOT NULL  
    ,Last_name    VARCHAR(20) NOT NULL  
    ,Gender       VARCHAR(10) NOT NULL
```

```
,Email          VARCHAR(50) NOT NULL
,Phone_number   VARCHAR(12) NOT NULL
,Date_of_Birth  DATE NOT NULL
,Availability_Status TINYINT NOT NULL
,Ratings        INTEGER NOT NULL
,Specialization_Sport_ID INTEGER NOT NULL
);
```

```
SELECT * FROM Coach_DB;
```

```
CREATE TABLE CollegeEquip_DB(
    Equipment_ID    INTEGER UNSIGNED NOT NULL UNIQUE
PRIMARY KEY
,Name              VARCHAR(20) NOT NULL
,Availability_Status TINYINT NOT NULL
,Conditions        VARCHAR(10) NOT NULL
,Manufacturer      VARCHAR(20) NOT NULL
,Owner_Admin_ID    INTEGER NOT NULL
);
```

```
SELECT * FROM CollegeEquip_DB;
```

```
CREATE TABLE Equipment_DB(
    Equipment_ID    INTEGER UNSIGNED NOT NULL UNIQUE
PRIMARY KEY
,Name              VARCHAR(20) NOT NULL
,Availability_Status TINYINT NOT NULL
,Conditions        VARCHAR(10) NOT NULL
,Manufacturer      VARCHAR(20) NOT NULL
,Owner_Admin_ID    INTEGER NOT NULL
);
```

```
SELECT * FROM Equipment_DB;
```

```
CREATE TABLE Event_DB(
```

```

    Event_ID      INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY
KEY
    ,Event_Name    VARCHAR(50) NOT NULL
    ,Event_Date    DATE NOT NULL
    ,Duration      INTEGER NOT NULL
    ,Start_Time    VARCHAR(8) NOT NULL
    ,End_Time      VARCHAR(8) NOT NULL
    ,Participant_Limit INTEGER NOT NULL
    ,Organizer_ID  INTEGER NOT NULL
    ,Date_of_Booking DATE NOT NULL
    ,Venue_ID      INTEGER NOT NULL
    ,Sport_ID      INTEGER NOT NULL
);

```

```

SELECT * FROM Event_DB;

```

```

CREATE TABLE Favourite_DB(
    User_ID INTEGER NOT NULL PRIMARY KEY
    ,Sport_ID INTEGER NOT NULL
);

```

```

SELECT * FROM Favourite_DB;

```

```

CREATE TABLE Friend_DB(
    User_ID      INTEGER NOT NULL PRIMARY KEY
    ,Friend_ID    INTEGER NOT NULL
    ,Date_of_Adding TEXT NOT NULL
);

```

```

DROP TABLE Friend_DB;
SELECT * FROM Friend_DB;

```

```

CREATE TABLE IssueEquip_DB(
    Borrower_ID      INTEGER NOT NULL PRIMARY KEY
    ,Equipment_ID     INTEGER NOT NULL

```



```
,Issue_Date_Time      VARCHAR(19) NOT NULL
,Expected_Return_Date_Time VARCHAR(19) NOT NULL
);
```

```
CREATE TABLE Judge_DB(
    Event_ID  INTEGER NOT NULL PRIMARY KEY
,Referee_ID INTEGER NOT NULL
);
```

```
CREATE TABLE Order_DB(order_db
    Order_ID  INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY KEY
,Order_Type  VARCHAR(10) NOT NULL
,Order_Status VARCHAR(10) NOT NULL
,User_ID    INTEGER NOT NULL
,Equipment_ID INTEGER NOT NULL
,Vendor_ID  INTEGER NOT NULL
);
```

```
CREATE TABLE Referee_DB(
    Referee_ID      INTEGER UNSIGNED NOT NULL UNIQUE
PRIMARY KEY
,First_name        VARCHAR(20) NOT NULL
,Last_name         VARCHAR(20) NOT NULL
,Gender            VARCHAR(10) NOT NULL
,Email             VARCHAR(50) NOT NULL
,Phone_number      VARCHAR(12) NOT NULL
,Date_of_Birth     DATE NOT NULL
,Availability_Status BIT NOT NULL
,Games_refereed    INTEGER NOT NULL
,Specialization_Sport_ID INTEGER NOT NULL
);
```

```
CREATE TABLE Sports_DB(
    Sport_ID  INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY KEY
,Name       VARCHAR(20) NOT NULL
```

```
,Max_Players INTEGER NOT NULL  
);
```

```
CREATE TABLE StudentInventory_DB(  
    Inventory_ID INTEGER NOT NULL PRIMARY KEY  
    ,Item_Count BIT NOT NULL  
    ,Owner_ID INTEGER NOT NULL  
);
```

```
CREATE TABLE Trains_DB(  
    Coach_ID INTEGER NOT NULL PRIMARY KEY  
    ,User_ID INTEGER NOT NULL  
    ,Start_Date DATE NOT NULL  
);
```

```
CREATE TABLE User_DB(  
    User_ID INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY  
KEY  
    ,First_name VARCHAR(20) NOT NULL  
    ,Last_name VARCHAR(20) NOT NULL  
    ,Email VARCHAR(50) NOT NULL UNIQUE  
    ,Phone_Number VARCHAR(12) NOT NULL UNIQUE  
    ,Date_of_Birth DATE NOT NULL  
    ,Gender VARCHAR(6) NOT NULL  
    ,Date_of_Joining DATE NOT NULL  
    ,Address VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Vendor_DB(  
    Vendor_ID INTEGER UNSIGNED NOT NULL UNIQUE  
PRIMARY KEY  
    ,Shop_name VARCHAR(20) NOT NULL  
    ,Email VARCHAR(50) NOT NULL  
    ,Phone_number VARCHAR(12) NOT NULL  
    ,no_of_equipment_repaired INTEGER NOT NULL
```

```
,Address_Location    VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Venue_DB(  
    Venue_ID          INTEGER UNSIGNED NOT NULL UNIQUE PRIMARY  
KEY  
    ,Venue_Name       VARCHAR(20) NOT NULL  
    ,Venue_Type       VARCHAR(10) NOT NULL  
    ,Opening_Time     VARCHAR(5) NOT NULL  
    ,Closing_Time     VARCHAR(5) NOT NULL  
    ,Location         VARCHAR(50) NOT NULL  
    ,Occupancy_Status BIT NOT NULL  
    ,Sport_ID         INTEGER NOT NULL  
);
```